An Algebraic Framework for the Solution of Inverse Problems in Cyber-Physical Systems

Dissertation by

Dipl.-Ing. Christoph Gugg, B.Sc.

Chair of Automation
Department Product Engineering
Montanuniversitaet Leoben, Austria

supervised by

O.Univ.-Prof. Dipl.-Ing. Dr.techn. Paul O'Leary, B.A., B.A.I, M.E.E.
Chair of Automation, Department Product Engineering, Montanuniversitaet Leoben, Austria

and

Prof. Dr. Peter Lee, B.Sc.
School of Engineering and Digital Arts, University of Kent, Canterbury, United Kingdom

Chair of Automation
Department Product Engineering
Montanuniversitaet Leoben
Peter-Tunner Straße 25
8700 Leoben, Austria

W: automation.unileoben.ac.at
E: automation@unileoben.ac.at
T: +43(0)3842/402-5301
F: +43(0)3842/402-5302

Cite this thesis as

```
@phdthesis{Gugg2015PhD,
 Author = {Gugg, Christoph},
 School = {Montanuniversitaet Leoben,
           Department Product Engineering,
           Chair of Automation},
 Title  = {An Algebraic Framework for the Solution of
           Inverse Problems in Cyber-Physical Systems},
 Year   = {2015}}
```

Last compiled on February 11, 2015; the document has been compiled 2459 times in total.

This thesis was typeset using LaTeX (PDFLatex, BibTex and MakeIndex from MacTeX-2013). Formulas and equations are formatted according to ISO80000-2. Computations were performed using The MathWorks MATLAB (R2012a, R2013b, R2014b), Maplesoft Maple (V16.0) and Wolfram Mathematica (10.0). Illustrations were designed using The Omni Group OmniGraffle (V5.3) and GNU Image Manipulation Program (GIMP V2.8). All registered trademarks and trade names are the property of their respective holders.

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich diese Arbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Christoph Gugg

Leoben, February 16, 2015 Christoph Gugg

Preamble

The time it took me to write my dissertation was a time well spent, I would not like to miss this period in my life. As his apprentice, Paul introduced me to the art of scientific writing and he showed me the value of good engineering practice. We've spent countless hours discussing novel approaches to find solutions for problems most people are not even aware they exist. It makes me proud, that over the years we not only developed a professional partnership, but also established a deep personal friendship. Paul enlightened me, that scientific documents such as this very dissertation will always remain in a work-in-progress state. If you are completely satisfied with your work, you did something terribly wrong. This formulation may sounds frustrating, but on second thought it enables us to reinvent ourselves, exploit unused potential and to leave predefined paths. Nevertheless, it is important to be proud of your achievements. At this point, I want to thank Paul for his support and the numerous lessons he taught me as friend and teacher during the last years.

One of these lessons was to always scrutinize even state-of-the-art literature. It is important to maintain a critical view onto all sources of knowledge. Especially in the young field of cyber-physical systems (CPS), there is no established body of knowledge. Numerous possible applications are described in literature, yet actual solutions to solve the associated problems are virtually non-existent. The vague and competing definitions of CPS pose a real threat to this young field of research. CPS are an interdisciplinary field, which makes it hard to identify suitable conferences to present our work. The importance of adequate physical models is often underrated in practical applications. New design methodologies such as model based design help scientists and engineers to formulate the required models on an abstract level and directly deploy them on a target system without detailed knowledge about the hardware platform. Assisting technologies of this kind may reduce the discrepancies between theory and practice. The problem class of inverse problems is ubiquitous in our daily life, e.g., our consciousness is based on our sensory experience. Cognition is based on perception, reasoning, awareness and judgment of physical phenomena. Our senses only provide us with the perturbed effect, yet we are searching for the associated cause. This introduces the concept of uncertainty, which is vital for any technical application incorporating measurements.

This thesis is structured from a bottom-up approach w.r.t. the information flow: from sensors to the supervisory system. The industrial projects and scientific publications in Part III form the core of this work. The related theory is explained in Part II while Part I summarizes the design methodologies. The chapter about machine vision is not to be seen in a CPS context, it has been included because the majority of projects in Part III uses optical cameras as the sensory input. A vast outlook is given in Part IV, because the described technology bears so much more potential for countless future applications. In particular, the topic of temporal data mining will play an important role for real-time condition monitoring of machines and knowledge discovery in machine data. A future thesis may describes CPS deductively from a top-down data mining point of view.

Nature is bound to the irrevocable laws of physics, even if we do not understand all of them yet. On the other hand, people follow their own rules, which are even harder to understand. I'm fortunate that I met special people along my journey whom I could rely on in every situation and I never had to doubt their support. I want to thank my family, friends and colleagues; all of them contributed to this work in their own ways. A special thanks goes to Peter for his constructive criticism as my second supervisor.

Abstract

A cyber-physical system (CPS) is defined as a distributed network of collaborating hybrid, dynamic devices that operate in real-time and abide the by laws of physics. CPS incorporate sensors to acquire data from their environment as well as actuators to influence physical processes. The concept encompasses intercommunicating mechatronic systems; popular civil implementations of this concept are wireless sensor actuator networks (WSAN) and cyber-physical production systems (CPPS).

The combination of sensors and actuators within the physical domain effectively forms a hierarchy of operational/reactive and strategic/predictive closed control loops within the cyber and socio domains respectively. The feedback loop is fundamentally a measurement system. In a mathematical sense, the evaluation of a measurement is an inverse problem, whereby the system's perturbed output, i.e., the effect, is observed and the system's original input, i.e., the cause, is sought. The acquired data has a given significance depending on the context it is related to. An analytically correct solution requires adequate mathematical models of the physical phenomena, whereby models are simplified abstractions of reality. Incorporation of a-priori knowledge about the system enables the solution of the problem with respect to a maximum likelihood estimation in the presence of noise. Using model based design (MBD), the equations are formulated on abstract model level without the need of detailed knowledge about the intended target hardware platform or programming language.

The dissertation focuses on the formulation of a robust algebraic framework for the description of physical models using discrete orthogonal polynomial (DOP) basis functions as numerical linear operators in regression analysis. Furthermore, a linear differential operator for the solution of perturbed ordinary and partial differential equations (ODE and PDE) has been derived. In this vein, inverse problems are solved using spectral regularization in a least squares sense with high numerical quality and stability. The use of linear operators is advantageous in terms of estimating the error propagation as well as their potential to be automatically translated to platform specific target code for embedded systems using MBD. In-the-loop verification techniques ensure the functional and numerical equivalence between model code and target code. The generic DOP concept has been expanded to weighted approximation, constrained basis functions and bivariate transformations to cover a wider range of possible applications.

The theoretical framework has been implemented in CPS applications on heavy machinery from the mining and tunneling industry utilizing the presented system design approach. This encompasses the use of DOP basis functions for system level calibration in machine vision together with a-priori estimation of confidence intervals, uncertainty weighted multi-source data fusion as well as the automatic generation and deployment of target code on various embedded processor platforms. Extensive experimental verification has been carried out during these projects. The new methods are completely general and fully scalable. They bear immense potential for future applications, especially in temporal data mining where multi-channel streaming data emerging from large-scale CPS has to be analyzed in real-time for adaptive/predictive control of physical processes within the socio domain.

Index Terms

cyber-physical system; inverse problem; model based design; embedded system; automatic programming; discrete orthogonal polynomials; machine vision;

Kurzfassung

Ein cyber-physisches System (CPS) ist definiert als ein verteiltes Netzwerk von zusammenarbeitenden hybriden, dynamischen Geräten welche in Echtzeit betrieben werden und den Gesetzen der Physik unterliegen. CPS verwenden Sensoren um Umgebungsdaten zu erfassen und Aktuatoren um physikalische Prozesse zu beeinflussen. Das Konzept umfasst kommunizierende mechatronische Systeme, bekannte zivile Anwendungen sind drahtlose Sensor-/Aktuatornetze (WSAN) sowie cyber-physische Produktionssysteme (CPPS).

Durch die Kombination von Sensoren und Aktuatoren im physischen Umfeld wird eine Hierarchie von operativen/reaktiven und strategischen/prädiktiven geschlossenen Regelkreisen im cyber bzw. in der socio Domäne aufgebaut. Die Rückkopplungsschleife im Regelkreis ist ein Messsystem, wobei die Auswertung von Messergebnissen ein inverses Problem im mathematischen Sinn darstellt. Das verrauschte Ausgangssignal, d.h. die Wirkung, wird beobachtet um auf das ursprüngliche Eingangssignal, d.h. die Ursache, zu schließen. Die erfassten Daten haben eine kontextabhängige Signifikanz. Eine analytisch korrekte Lösung benötigt adäquate mathematische Modelle der auftretenden physikalischen Phänomene, wobei Modelle immer vereinfachte Abstraktionen der Realität darstellen. Das Einbringen von Vorabwissen über das betrachtete System schafft die Voraussetzungen für die Problemlösung im Sinne eines Maximum-Likelihood Schätzers in Anwesenheit von Rauschen. Modellbasiertem Entwurf (MBD) ermöglicht die Formulierung der notwendigen Gleichungen auf abstrakter Modellebene ohne Detailwissen der verwendeten Zielhardware oder der verwendeten Programmiersprache.

Der Schwerpunkt dieser Dissertation liegt auf der robusten algebraischen Formulierung physikalischer Modelle durch den Einsatz von diskreten orthogonalen Polynomen (DOP) als numerische lineare Operatoren in der Regressionsanalyse. Ein linearer Differenzialoperator wurde entwickelt um verrauschte gewöhnliche sowie partielle Differenzialgleichungen (ODE und PDE) zu lösen. Ein Ergebnis mit hoher numerischer Qualität und Stabilität wird durch Regularisierung im Spektralbereich und mithilfe der Methode der kleinsten Quadrate erreicht. Lineare Operatoren ermöglichen die verlässliche Abschätzung der Fehlerfortpflanzung und mit MBD kann plattformspezifischer Zielcode für eingebettete Systeme erzeugt werden. Die funktionale sowie numerische Äquivalenz zwischen Modell- und Zielcode wird durch In-the-Loop Verifizierung sichergestellt. Das generische DOP Konzept wurde um gewichtete Näherungsverfahren, Basisfunktionen mit Nebenbedingungen sowie bivariate Transformation erweitert.

Die Implementierung der Theorie erfolge auf Schwermaschinen im Berg- und Tunnelbau unter Verwendung der vorgeschlagenen Entwurfsprinzipien. Dies umfasst den Einsatz von DOP Basisfunktionen für die Kalibration von Bildverarbeitungssystemen auf Systemebene inklusive Vertrauensintervallabschätzung sowie analytische, mit der Messunsicherheit gewichtete, Datenfusion aus mehreren Signalquellen. Aus den Modellen wurde automatisch Zielcode generiert und auf unterschiedlichen eingebetteten Prozessorplattformen verteilt. Umfassende experimentelle Verifizierung wurde während dieser Projekte durchgeführt. Die neuen Methoden sind komplett generisch und voll skalierbar, daher bieten sie Potential für vielfältige Anwendungen. Zeitabhängigen Daten, welche von großräumig angelegten CPS stammen und durch kontinuierliche parallele Datenflüsse gekennzeichnet sind, werden durch Data-Mining in Echtzeit analysiert.

Schlagwörter

cyber-physikalisches System; inverses Problem; modellbasierter Entwurf; eingebettetes System; automatisches Programmieren; diskrete orthogonale Polynome; maschinelles Sehen;

Glossary

x, y	...	scalar
$\hat{x}$, $\hat{y}$	...	estimator for scalar
$\boldsymbol{x}$, $\boldsymbol{y}$	...	vector
$\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$	...	estimator for vector
X, Y	...	matrix
$\boldsymbol{x}^{\mathrm{T}}$, X^{T}	...	transposed vector, transposed matrix
X^{-1}, X^{+}, X^{-}	...	inverse, pseudoinverse, generalized inverse matrix
$\mathsf{X} \circ \mathsf{Y}$	...	Hadamard product
$\mathsf{X} \otimes \mathsf{Y}$	...	Kronecker product
$\{x, y\}$	...	a set of variables
x_i	...	i^{th} element of $\boldsymbol{x}$
x_{ij}	...	i^{th} row / j^{th} column element of X
i, j, k	...	indexing variables
m, n	...	number of rows, columns, samples, etc.
$\boldsymbol{0}$	...	vector of zeros
$\boldsymbol{1}$	...	vector of ones
$\boldsymbol{e}_i$	...	unit vector in i^{th} dimension
I	...	identity matrix
W	...	weighting matrix
J	...	Jacobian matrix
L	...	linear operator
B	...	general basis
G	...	discrete orthogonal polynomial basis
D	...	differentiating matrix
$\mu_{\boldsymbol{x}}$	...	expectation value of $\boldsymbol{x}$
$\bar{x}$	...	sample mean of $\boldsymbol{x}$
$\sigma^2_{\boldsymbol{x}}$	...	variance of $\boldsymbol{x}$
$s^2_{\boldsymbol{x}}$	...	sample variance of $\boldsymbol{x}$
$\mathsf{\Lambda}_{\boldsymbol{x}}$	...	covariance matrix of $\boldsymbol{x}$
ν	...	degrees of freedom (d.f.)
$t_{(\nu)}$	...	Student distribution with ν d.f.
$\mathcal{F}_{(\nu_1,\nu_2)}$	...	Fisher distribution with ν_1 and ν_2 d.f.
$\chi^2_{(\nu)}$	...	Chi-squared distribution with ν d.f.
i.i.d.	...	independent and identically distributed

Contents

1 Introduction

Cyber-physical systems (CPS) are a relatively young field of study, the first publications are dating back to the year 1998[1], whereby the topic got widely accepted by the scientific community in the mid 2000s[2] [172]; as a consequence, numerous ambiguous definitions of CPS have been given, e.g., in [59, 60, 108, 194]. These existing definitions will be discussed and refined in Chapter 2 for the context of this thesis.

CPS are designed as a network of interacting computational elements controlling physical entities, i.e., a consequent advancement of mechatronic systems. The implementation of a CPS is an interdisciplinary task, resulting in co-design of hardware, software, networks and associated mathematical models of the physical phenomena. The goal of the thesis is to propose a stringent algebraic framework for models satisfying the demands of CPS in monitoring and control applications. The models are generally applicable in high-level supervisory as well as low-level embedded systems. A special focus is laid onto the efficient solution of inverse problems. This class of problems naturally occurs in measurements with perturbed observations, and, therefore, in CPS.

1.1 Synopsis

Part I: System Design. The idea of CPS is introduced and existing approaches to the topic are discussed. Fields of application are listed and a generic CPS architecture is described. A wireless sensor actuator network (WSAN) has numerous individual sensor/actuator nodes, called *motes*, whereby each mote is composed of an embedded system with sensors and actuators connected to it. The acquired and processed data is transmitted to a supervisory system, which consolidates the extracted information from multiple sources via temporal data mining. This approach effectively forms a large-scale multi-purpose monitoring and control system. A cyber-physical production system (CPPS) uses the concept of model based production for adaptive control of production processes.

Model based design (MBD) is an attempt to handle the design complexity and to ensure the fulfillment of the system requirements. Physical models are formulated in an abstract mathematical sense, which enables the deployment on embedded systems via automatic programming. An overview of embedded system architecture is given along with a description of in-the-loop verification techniques for embedded targeting. The supervisory system uses the very same models in a parallel processing environment to accelerate data mining. Additionally, a summary of optical principles and image processing techniques for machine vision is provided.

[1] Thomson Reuters: Web of Knowledge, search term *cyber-physical system*, `www.webofknowledge.com`

[2] Elsevier: Scopus, search term *cyber-physical system*, `www.scopus.com`

Part II: Algebraic Models. The established theoretical framework is completely general and facilitates the solution of inverse problems with uncertainty analysis. The context of the occurring physical effects in CPS must be fully understood in order to formulate suitable models during system identification. Models utilizing only fundamental linear algebra are executable on embedded systems while fulfilling the strict energy and communication constraints of a CPS.

The concept of basis functions is introduced, accompanied by the synthesis of discrete orthogonal polynomials (DOP) and a discussion of their numerical stability on different platforms. Spectral regularization allows the solution of ill-posed inverse problems w.r.t. a maximum likelihood estimation. Furthermore, weighted approximation, constrained basis functions and bivariate transformations are derived; combining these principles with a covariance estimation yields the capability for analytic multi-source data fusion based on measurement uncertainty.

The theory on basis functions is expanded to numerically solve ordinary differential equations (ODE) and partial differential equations (PDE) from data acquired from perturbed measurements in an efficient and robust manner with high numerical quality.

Part III: Projects and Publications. A total of four projects are presented along with their corresponding publications. The design approach presented in Part I and the theoretical framework formulated in Part II are applied in these specific applications. All projects are settled in a measurement/control/condition monitoring domain for underground constructions and heavy machinery from the mining and tunneling industry. The conference papers and journal articles are accompanied by additional information that did not fit into the documents when they were published. The first and second project are monitoring applications while the third and fourth project are control applications.

1. The *opto-electronic perpendicular* (OEP) demonstrates the possibility of using a bivariate DOP basis to implement a mapping between a distorted camera space and the metric real space for an optical measurement system. The concept is applied as a structural health monitoring system for large-scale underground constructions, e.g., shafts and tunnels, using electro-active glass targets;

2. The *embedded geo-sensing* (EGS) deploys the mathematical models on motes for measuring geological phenomena such as ground subsidence. The established sensor network is utilized for safety relevant condition monitoring of structural integrity in underground constructions. This includes the repeated real-time solution of ODE from perturbed measurement data on embedded systems. Utilizing a time series of measurement values supports the solution of PDE in a post-processing step on a supervisory system;

3. The *laser guided mining* (LGM) uses uncertainty weighted multi-source sensor fusion to facilitate a compact hardware design for a large-scale optical position sensitive detector (PSD) with extended measurement range. The device is used to remotely guide a continuous miner via a reference laser plane during its operation;

4. The *active laser target* (ALTv2) is a fully integrated optical measurement device for controlling a tunnel boring machine (TBM) in a harsh environmental setting. A system level calibration process is presented, whereby an estimation of the expected measurement uncertainty can be given a-priori.

Part IV: Discussion and Appendices. The results and insights acquired from the projects are interpreted and it is concluded how future work can be improved using this knowledge. An outlook is given for potential fields of applications for the design framework as well as the corresponding mathematical models. Thereby, a special focus is laid on temporal data mining techniques for real-time analytics of machine operating data, which is the consequent enhancement of the presented theory in a CPS environment. The appendix contains a list of common terms in metrology, the list of figures, the list of the author's publications and the bibliography.

1.2 Contributions

(1) Instrumentation theory. In contrast to classical digital signal processing, the proposed approach handles discrete, non-periodic signals of finite length which have been sampled from a continuous signal. A sound measurement and calibration model is formulated which enables the solution of forward and associated inverse problems in a least squares sense. The presented linear transformations are advantageous w.r.t. noise behavior, i.e., the measurement uncertainty can be estimated during the calibration procedure. The mathematical model facilitates analytic uncertainty weighted multi-source data fusion. The goodness-of-fit parameters enable the computation of confidence and prediction intervals during regression analysis. The nature of random error and the existence of a systematic error is determined via non-parametric statistical validation techniques. The method is in compliance with the guide to uncertainty in measurements [105] and relevant European Union directives.

(2) Mathematical framework. The verbose theoretical derivation proves, that there exists one and only one unitary discrete orthogonal basis which can by synthesized from a sum of monomials. The basis is numerically stable, provides excellent noise behavior and supports arbitrary placement of nodes, which is especially useful for the numeric solution of differential equations from perturbed measurement data. The continuous measurement model is discretized and its solution is efficient and stable, because the regularizing differential operator works in the presence of noise. The solver is implemented using only fundamental linear algebra; consequently, the propagation of uncertainty can be determined analytically and automatic programming for low-level languages becomes possible.

(3) Design methodologies. A survey of definitions for cyber-physical systems is given together with an overview of structured design methodologies for model based design. The architectural constraints of cyber-physical systems are described, whereby special focus was laid onto wireless sensor actuator networks. Thanks to its algebraic formulation, the mathematical framework is suitable for automatic generation of target code for various embedded processor platforms, i.e., the models are fully scalable. The types of embedded system architectures are listed together with relevant standards.

(4) Practical applications. The design methodologies and the mathematical framework are applied on laboratory prototypes. The devices are engineered to operate in a harsh environmental setting provided by the mining and tunneling industries. The presented theory is completely general, e.g., it can be applied for structural deformation measurements as well as for optical measurements. The concepts of machine vision are explained in this application-oriented context.

Part I

System Design

2 | Cyber-Physical Systems

CPS have been described in various ways since their emergence; however, there is still no unified body of knowledge. A selection of various definitions from institutions, universities and governments are cited in the upcoming section. The most general definition is:

> A cyber-physical system is an agent-based network of spatially distributed, collaborating, hybrid, computing devices and a supervisory intelligence. A CPS dynamically interacts with its environment in real-time and abides by the laws of physics.

The chapter's goal is to refine the topic to fit the thesis' scope. Fields of application are listed and a generic CPS architecture is derived with characterization of its main components. A CPS manifests itself in three distinct domains: the *physical*, the *cyber* and the *socio* space. Models form the architectural foundation of a CPS, see Fig. 2.1.

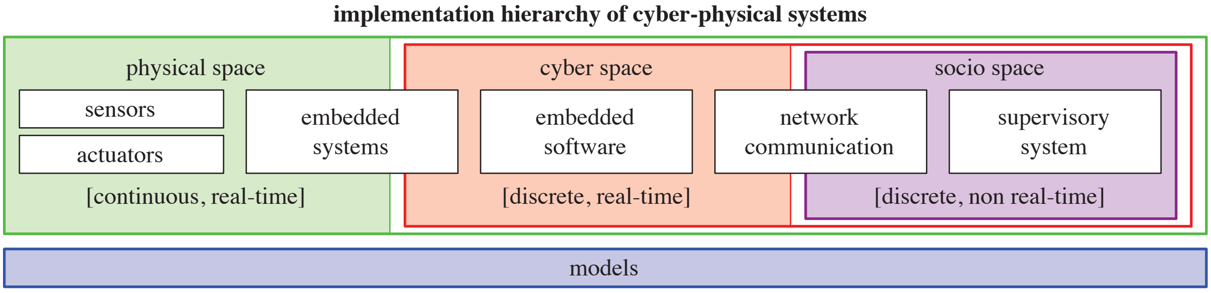


Fig. 2.1 A CPS is hierarchically divided into three layers. The physical system to be monitored and controlled resides in the continuous real space. Sensors and actuators interface with the physical space (green). They are connected to embedded systems, which integrate the required embedded software in the discrete cyber space (red). The data of multiple embedded systems is accumulated over network communications by a supervisory system in the temporally decoupled socio space (purple). Models (blue) unite the individual layers.

2.1 Definitions

According to the agendaCPS report [59, 108] by the German National Academy of Science and Engineering (acatech), which has been assigned by the German government (Federal Ministry of Education and Research), CPS are composed of connected embedded systems, which directly acquire physical data via sensors and control actuators to actively influence their surroundings. CPS possess multi-modal human-machine interfaces (HMI) and they are capable of utilizing globally available data and services; i.e., CPS can be seen as open social-technical networks and cover corresponding logistic, coordination and management processes.

The Fraunhofer Institute [57, 194, 195] defines CPS as self-organizing, connected, intelligent objects, which are controlled decentralized. They capture data of the real world via sensors, process them with software from embedded controllers, use the internet and cloud computing for mutual communication between the connectors and interact with the real world by means of mechatronic actuators. The institute sees much potential in CPS cooperating with humans in order to achieve lean production.

The United States National Institute of Standards and Technology (NIST) [44, 196] and the National Science Fund (NSF)[1] describe CPS as systems with integrated intelligence which have computational processes that interact with physical components in changing environments. The computational and physical processes of such systems are tightly interconnected and coordinated to work together effectively, often with humans-in-the-loop.

The Institute of Electrical and Electronics Engineers (IEEE) [10] as well as the Association for Computing Machinery (ACM) define[2] CPS as systems with a coupling of the cyber aspects of computing and communications with the physical aspects of dynamics and engineering that must abide by the laws of physics. This includes sensor networks, real-time and hybrid systems.

Another description has been given in the Elsevier Computer Communications [172] journal: in CPS, various embedded devices with computational components are networked to monitor, sense, and actuate physical elements in the real world. This is a transformation of traditional embedded systems into CPS, which are characterized by tight integration and coordination between computation and physical processes by means of networking.

Lee [119] from the University of California at Berkeley[3] and Tabuada [200] from University of California at Los Angeles define CPS as a network of physically distributed embedded sensors and actuators equipped with computing and communicating capabilities, usually with feedback loops where physical processes affect computations and vice versa.

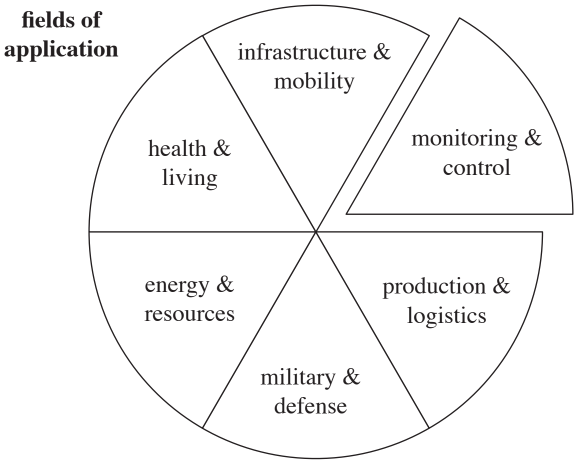


Fig. 2.2 CPS can be categorized into six fields of application. A number of scenarios are described in the agendaCPS report [59] with special focus on infrastructure and mobility, energy and resources as well as health and living applications. The field of production and logistics is described in [108]. *Smart dust* and *unmanned aerial vehicles* (UAV) [14], a.k.a. *drones*, are the weaponized agent-based form of CPS. The author wants to strictly dissociate with CPS utilized in a military context. The focus of this thesis is laid on the monitoring and control aspect of CPS, which also supports production and logistics.

[1] Cyber-Physical Systems Virtual Organization, `cps-vo.org`

[2] ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), `iccps.acm.org`

[3] A list of publications can be found at `cyberphysicalsystems.org`

2.2 Applications

In literature [59, 108], CPS claim a vast field of applications. Fig. 2.2 gives an overview of a possible classification, whereby the *production and logistics* as well as the *monitoring and control* aspect will be reviewed in more detail. Fig. 2.3 illustrates a CPS in a tunneling application, bringing together multiple collaborating measurement and control systems; some of them are introduced in Part III.

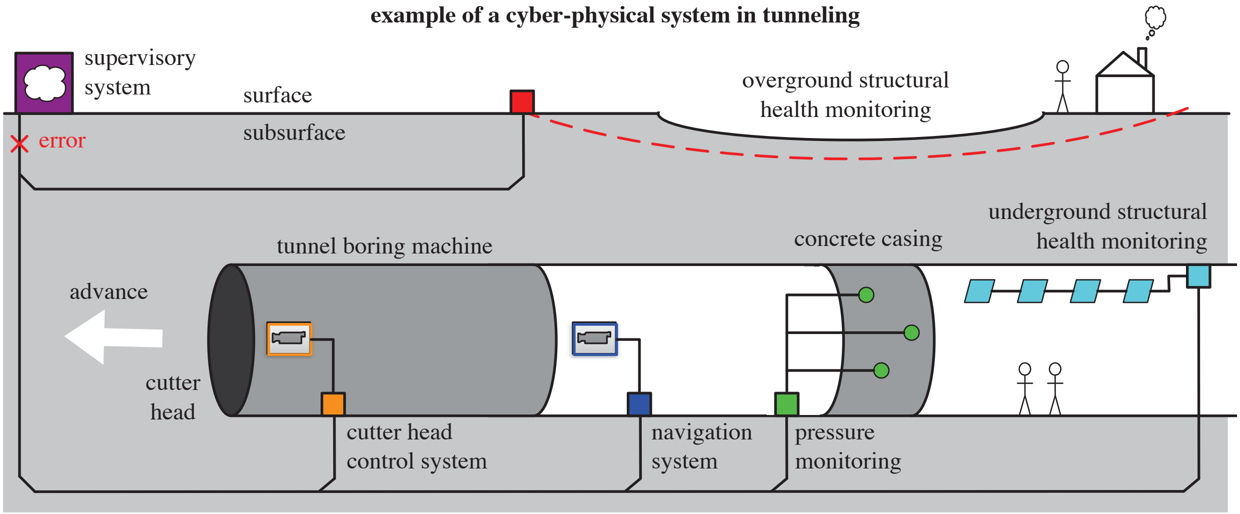

example of a cyber-physical system in tunneling
supervisory system
surface
subsurface
error
overground structural health monitoring
tunnel boring machine
concrete casing
underground structural health monitoring
advance
cutter head
cutter head control system
navigation system
pressure monitoring

Fig. 2.3 The example shows a heterogeneous CPS in a tunneling application. The motes (□) act as the agents within the CPS; they are connected to the supervisory system. The optical cutter head control system (orange) [79] and the optical navigation system (blue) [74] are used to control the tunnel boring machine. The tunnel is encased with a concrete casing, the pressure during the concreting process is monitored (green). Structural health monitoring is implemented either as an underground optical measurement system (cyan) [71] as well as an overground inclinometer based system (red) [88]. The agents are able to operate independently even when an error of the network connection occurs.

2.2.1 Cyber-Physical Production System

Condition monitoring and reactive control of facilities enables one of the major technological advances associated to CPS: the *cyber-physical production system* (CPPS) [60, 61]. In Europe and especially in Germany[4], these new manufacturing techniques are referred to as *Industrie* 4.0 [108]; within the United States a similar program is called *Smart Manufacturing* [44, 196]. CPPS mainly focus onto the *internet of things and services* to enable highly individualized rapid manufacturing in the self-organizing, resilient factory.

The concept of mass customization, or *batch size* 1, and the dynamic adaption of production processes on manufacturing execution system (MES) level is presented in [7, 21, 154, 215]. The combination of flexible production modules allows to have a mix of high and low volume products on the same production line. The *smart workpiece* becomes an active part of the production, notifying the machines via radio frequency identification (RFID) what operations to perform on it. In a CPS sense, products are the agents and the MES is the supervisory system. CPS are integrated horizontally in the value chain and vertically in production systems.

[4] Plattform Industrie 4.0, www.plattform-i40.de

Consequently, continuous life-cycle analysis with predictive maintenance and self-diagnostics of machinery becomes possible [68].

The need of human-machine collaboration for lean production is described by the Fraunhofer Institute [195] in order to even out fluctuating work loads in the production process. CPS are seen as assistance system for organizing work capacities within the factory. As stated in [57], the main challenges in establishing CPPS are: interoperability between legacy and new systems as well as communication interfaces between different production installations; the amount of complex data with the need of automatic anomaly recognition; the need of a user-centric approach with intelligent information conditioning; and a holistic security concept, which takes care of data interception and manipulation.

McKinsey identified twelve potentially economically disruptive technologies [130], a majority of them is CPS related: mobile internet, automation of knowledge work, the internet of things, cloud technology, advanced robotics, autonomous and near-autonomous vehicles as well as 3D printing. The last-mentioned is of particular usefulness for *rapid prototyping* [52], a methodology for quickly fabricating small numbers of physical parts via generative model based production.

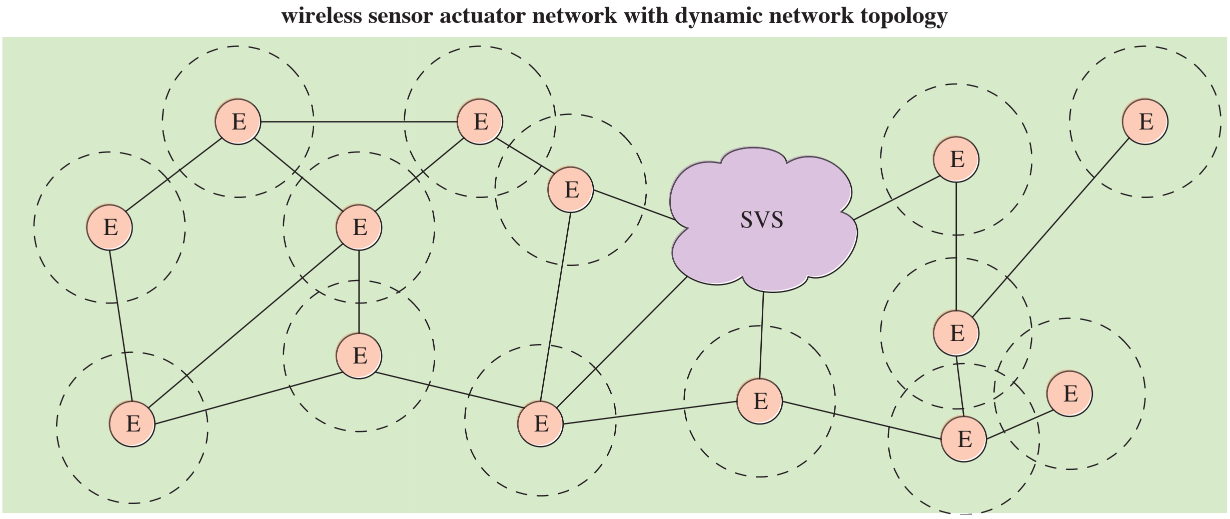

wireless sensor actuator network with dynamic network topology
E
E
E
E
E
E
E
E
SVS
E
E
E
E
E
E
E
E

Fig. 2.4 A generic WSAN is composed of a (variable) number of spatially distributed, autonomous sensor nodes (*motes*). The location of each mote is not known a-priori. Each mote consists of an embedded system (E) and sensor(s)/actuator(s) attached to it, see Fig. 2.5. The motes are dynamically connected adhoc to each other in a mesh-like, adaptable topology. Each mote covers a physical subsystem with its sensor(s)/actuator(s). These subsystems can overlap, the resulting redundancy is usable for verification purposes. Physical processes are intrinsically concurrent. A supervisory system (SVS) harvests the data from all motes for analytics in data mining.

2.2.2 Wireless Sensor Actuator Network

The term *wireless sensor actuator network* (WSAN) is the most generic description of a CPS incorporating the *monitoring and control* aspect [3], leading to the term of *ubiquitous computing* [59, 172]. Other common terms are *wireless sensor network* (WSN) [81] for implementations without actuators or simply *sensor network* as a generic description. The *wireless* property emphasizes the adaptive structure of a WSAN's network topology. WSAN have been originally developed for autonomous battlefield observation [120], coining the term *smart dust* [207];

fortunately, WSAN are now used in a more meaningful way, e.g., in monitoring civil infrastructure [212]. Fig. 2.5 shows the system environment of a WSAN in a CPS sense. The goal is to implement WSAN motes, which efficiently perform autonomous monitoring and reactive control, i.e., certain computations are carried out locally and the results are transmitted to a supervisory system, see Fig. 2.6. The bandwidth demands are effectively reduced by increasing the information density. Only the processed spectra/coefficients/moments are transmitted instead of the raw data. The wireless infrastructure poses a number of resource constraints:

1. The **power supply** must be provided via batteries or power scavenging units. A possible solution is the implementation of a stand-by or sleep state, which must not interfere with the global time synchronization within the network. Energy efficient communication protocols and models are required [207];

2. The self-organizing **network topology** is multi agent-based and scalable: motes can dynamically enter and leave the network. Each mote requires a unique identifier, e.g., via IPv6 [156]. The data is forwarded in multi-hop infrastructureless architecture between the motes until it reaches the supervisory system upstream or the actuators downstream respectively, leading to a cumulative data aggregation. Therefore, the data one mote has to carry increases with the size of the WSAN and motes near the sink suffer the most from communication overhead [81]. The network path is optimized either by the available power, minimum energy or minimum hop distance [3];

3. State-of-the-art **wireless communication** technologies on data link level are WiFi (IEEE 802.11b), ZigBee (IEEE 802.11.5) and Bluetooth (IEEE 802.15.1) [212], which are compatible to the most promising protocol stack IEEE 1451 [101] for WSAN. A stable network connection cannot be guaranteed. When the number of participating motes increases, the available bandwidth decreases while the latency increases. The sampled data must be buffered when the connection is lost and, as a consequence, the sample rate must be slowed down because of the limited local storage of embedded systems;

4. Each measurement acquired by a mote must have a **unique identifier** and a **time stamp**. The data can be corrupted or lost physically during measurement, computation and transmission as well as logically during organization and interpretation. Sensors and actuators are most commonly based on microelectromechanical systems (MEMS).

2.3 Architecture

Building a unified framework of algebraic models for CPS by carefully considering the consistency among all parts of the system, especially the agreement with the underlying physical principles, is the main issue of this thesis. From a design point of view, all domains must be planned and implemented simultaneously in order to yield a fully integrated CPS [47, 54], see Fig. 2.6. Systems analysis and engineering [16, 36, 63, 80, 173] deliver methods to handle the complexity of interdisciplinary cooperation. A top-down approach is most practical to ensure the fulfillment of the system requirements: the techniques associated with model based design (MBD) are presented in Chapter 3, the associated models are discussed in Part III. An overview of CPS system architecture is illustrated in Fig. 2.6 with a more detailed description in Fig. 2.7.

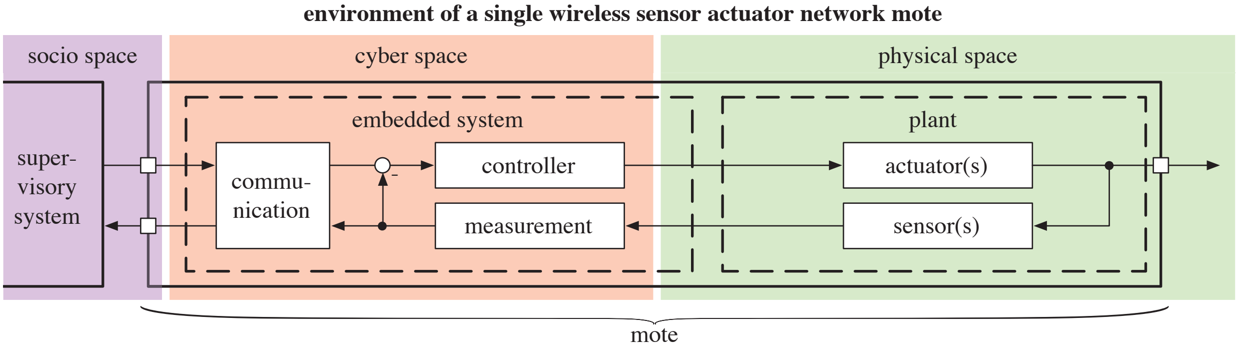


Fig. 2.5 A mote resided in the physical and the cyber space. Sensor(s) are used to acquire the current state of the plant, whereas actuator(s) influence the plant's state. The physical system's characteristics must be known to formulate adequate models and contextual constraints in order to give the acquired data significance. The embedded system integrates the software for the control and measurement processes, the generated information is forwarded to adjacent motes and ultimately to the supervisory system via the communication module.

The architectural components of a CPS are:

(1) System. A system is an integrated set of interacting components with a structured architecture and a clear functionality. The boundaries of a real system are within the spatial and temporal domain. Defined interfaces enable the communication with the system's surrounding environment, which also includes other adjacent systems. CPS are real-time, hybrid, dynamic systems, i.e., they exhibit both continuous and discrete time-dependent behavior. CPS are often related to as *systems of systems* with non-static boundaries, i.e., the modular architecture facilitates self-organization of spatially distributed subsystems. Each subsystem operates autonomously, subsystem failures do not lead to a complete system breakdown. System requirements are specified on system level; however, networked systems are never completely testable under all possible conditions and must be verified via cross-validation techniques.

(2) Physical space. The underlying physical system is bound to the laws of physics and situated within the continuous, real-time domain. Physical components always incorporate a certain level of uncertainty caused by randomness in the environment. The occurring physical phenomena are best expressed as ordinary differential equation (ODE) in one dimension or as partial differential equation (PDE) in multiple dimensions. Coupled physical subspaces interact with each other non-deterministically over many spatial and temporal scales, because each physical subspace is part of the real world. Sensors detect the current state of the physical subsystem, whereas actuators are changing it, i.e., the behavior of the physical system has a causal relationship with the control system. Sensors and actuators are interfacing with a subset of the physical space, whereby these components are connected to embedded systems.

(3) Cyber space. The cyber space is the discretized representation of the continuous physical space. Discretizing ODE or PDE yields difference equations, which are solved numerically. The on-site (local) embedded system has two main functions: quantitative measurement of the physical subsystem via sensors and control of the actuators. This forms an operative level-1 real-time control loop, i.e., a reactive systems in the field. Conventional programming paradigms can only be applied in a limited manner, as time cannot be hidden in the abstraction [150]. The composite of an embedded system and its peripherals form a local sensor/actuator node, the

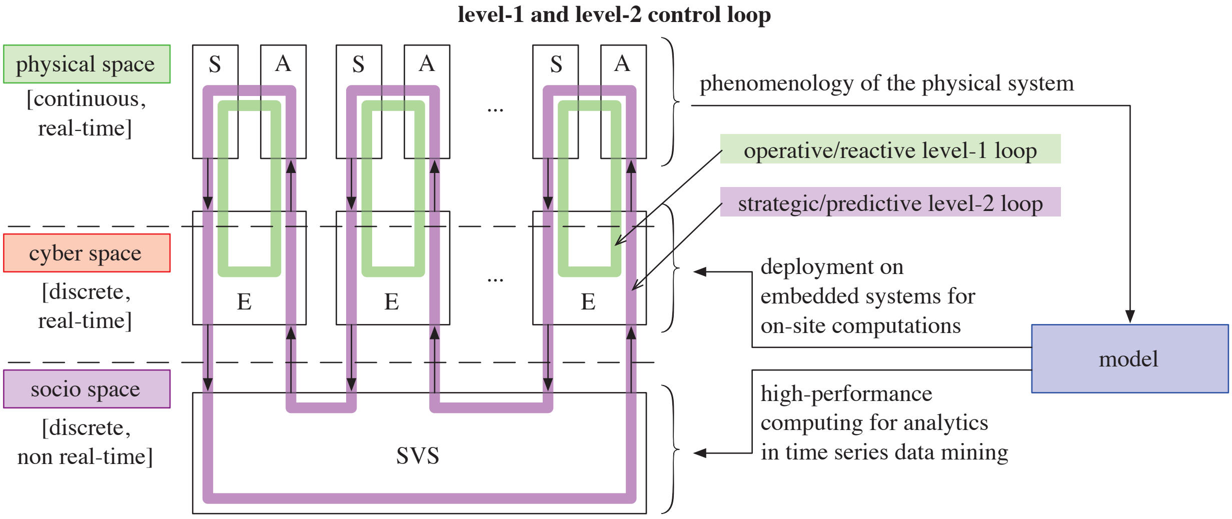

level-1 and level-2 control loop
physical space
[continuous, real-time]
cyber space
[discrete, real-time]
socio space
[discrete, non real-time]
S
A
E
SVS
...
phenomenology of the physical system
operative/reactive level-1 loop
strategic/predictive level-2 loop
deployment on embedded systems for on-site computations
model
high-performance computing for analytics in time series data mining

Fig. 2.6 Each locally embedded system (E) and its sensor(s)/actuator(s) (S/A) form an operative level-1 reactive system (green). Critical computations are carried out on-site for adaptive control in real-time. The processed data of each mote is forwarded to the supervisory system (SVS); using the information from all available motes forms a strategic level-2 predictive system. The physics behind the observed system defines the model. The models in this thesis are generically applicable and fully scalable, i.e., targeting of embedded systems is possible; the models are also suitable for high-performance computing (HPC).

mote. The intermediate computation of critical parameters results in significant reduction in communication overhead and energy consumption. Embedded software is written in low-level languages such as `C`.

(4) Socio space. The off-site (global) supervisory system collects the quantitative data streamed from multiple individual motes and fuses it to qualitative information with significance considering all available devices in the network. The information facilitates a decision-making process, i.e., a strategic level-2 control loop for predictions utilizing time series data mining techniques according to the cross industry standard process for data mining (CRISP-DM) framework [25], see Section 15.4. The goal of data mining and data analytics is to find events within the accumulated data, whereby strategic decisions are based on this events. Actions based on decisions bear a risk. Besides this functionality, the global supervisory system coordinates and synchronizes the locally distributed embedded systems.

(5) Models. A model is a simplified abstraction of a real physical system while still describing its characteristic behavior. Contextual understanding of the physical system such as constraints and a-priori knowledge are necessary to acquire unique solutions from deterministic models. Measurements incorporate uncertainty by definition, probabilistic models can compensate some of these effects. Higher-order ODE describe changes in the temporal or spatial domain, whereas conventional PDE describe changes in both domains of physical systems. A model's computational traits are: (1) quality of the results, i.e., how well does the model suit the problem; (2) cost, i.e., how much time, memory and energy does the computation require; and (3) complexity, i.e., is the model deployable on embedded systems and/or is it suitable for high-performance computing (HPC), see Fig. 2.6.

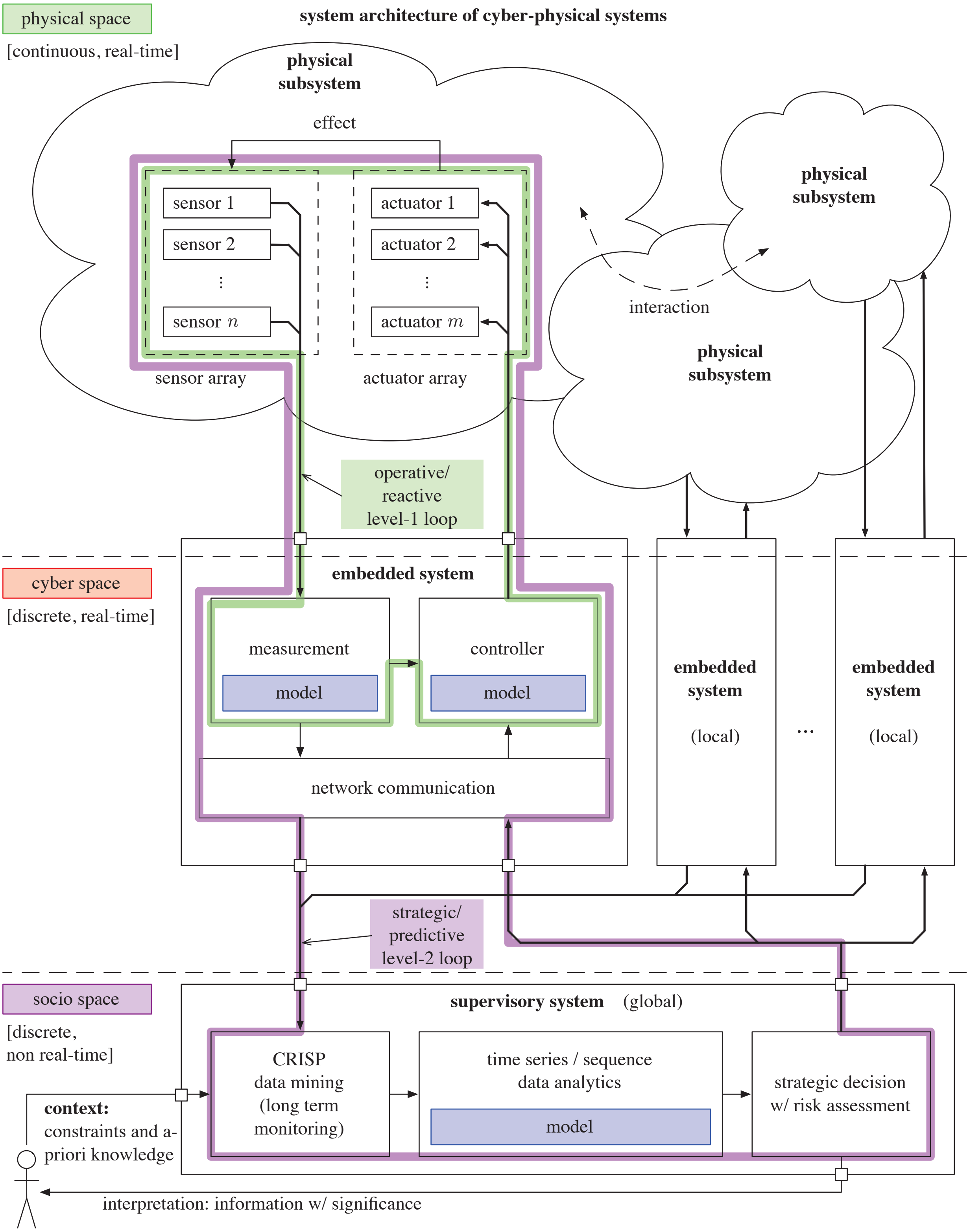

physical space
[continuous, real-time]
system architecture of cyber-physical systems
physical subsystem
effect
sensor 1
sensor 2
⋮
sensor n
actuator 1
actuator 2
⋮
actuator m
sensor array
actuator array
physical subsystem
interaction
physical subsystem
operative/ reactive level-1 loop
cyber space
[discrete, real-time]
embedded system
measurement
model
controller
model
network communication
embedded system
(local)
…
embedded system
(local)
strategic/ predictive level-2 loop
socio space
[discrete, non real-time]
supervisory system (global)
CRISP data mining (long term monitoring)
time series / sequence data analytics
model
strategic decision w/ risk assessment
context: constraints and a-priori knowledge
interpretation: information w/ significance

Fig. 2.7 This is a detail illustration of Fig. 2.6: the physical phenomenon determines the formulation of the model. The model is deployed on the embedded system's measurement and control routines as well as on the supervisory system's data analytics process. The reactive level-1 control loop and the predictive level-2 control loop are characteristic traits of a CPS.

3 | Model Based Design

Model based design (MBD) is a methodology for iterative system analysis, modeling and validation on an abstract level while ensuring the consistency of the system's physical representation [150, 174]. In literature [23, 32, 107, 147], MBD is described as a top-down approach, where software components are replaced by executable models, that represent the system's behavior and deliver immediate feedback. A system is divided in the description of its functionality and its architectural implementation [35, 189].

MBD supports a workflow that incorporates system modeling, controller design, simulation, code generation and rapid prototyping; engineering problems are handled on system level, not code level, which significantly reduces development time; detailed knowledge about the target system is not required [47]. Abstract models with well designed interfaces are adaptable for various projects. MBD improves the efficiency of system development that unites multiple physical domains, especially in the automotive [41, 148] and aerospace industry [211]. For instance, the NASA [201] developed a guidance, navigation and control system, where even the specification of requirements took several years. Using MBD, the requirements are executable and enable the errors identification early in the design process.

The consistency between models and their implementation is given at any point during the development process thanks to the integrated documentation [41]. Full traceability is needed to match requirements with functionality; furthermore, multiple engineering teams can work on the same project simultaneously [8]. The core idea of MBD is continuous verification and validation at each design stage to ensure that the (sub)system's requirements are satisfied. MBD utilizes the standardized V-model framework for product development [173, 204], see Fig. 3.1. It was shown in literature [1, 83, 107, 175, 183], that MBD is best-practice for designing CPS as well as safety relevant applications in compliance with the IEC 61508 standard on functional safety [32, 98].

Model driven architecture (MDA) is a standardized technique to automatically synthesize usable code from visualizations such as class diagrams; it is part of the OMG[1] UML[2] 2.x specification [160]. SysML [159] is an extension of UML to describe requirements on system level. Recent studies [27, 34, 35, 109] recommend a UML based *platform based design* (PBD) approach, which lays the focus on connecting subsystem via standardized interfaces. The difference to a conventional MBD approach is that parts of the development are carried out bottom-up. The goal is to establish a library of reusable soft- and hardware components.

[1] Object Management Group, `www.omg.org`

[2] Unified Modeling Language, `www.uml.org`

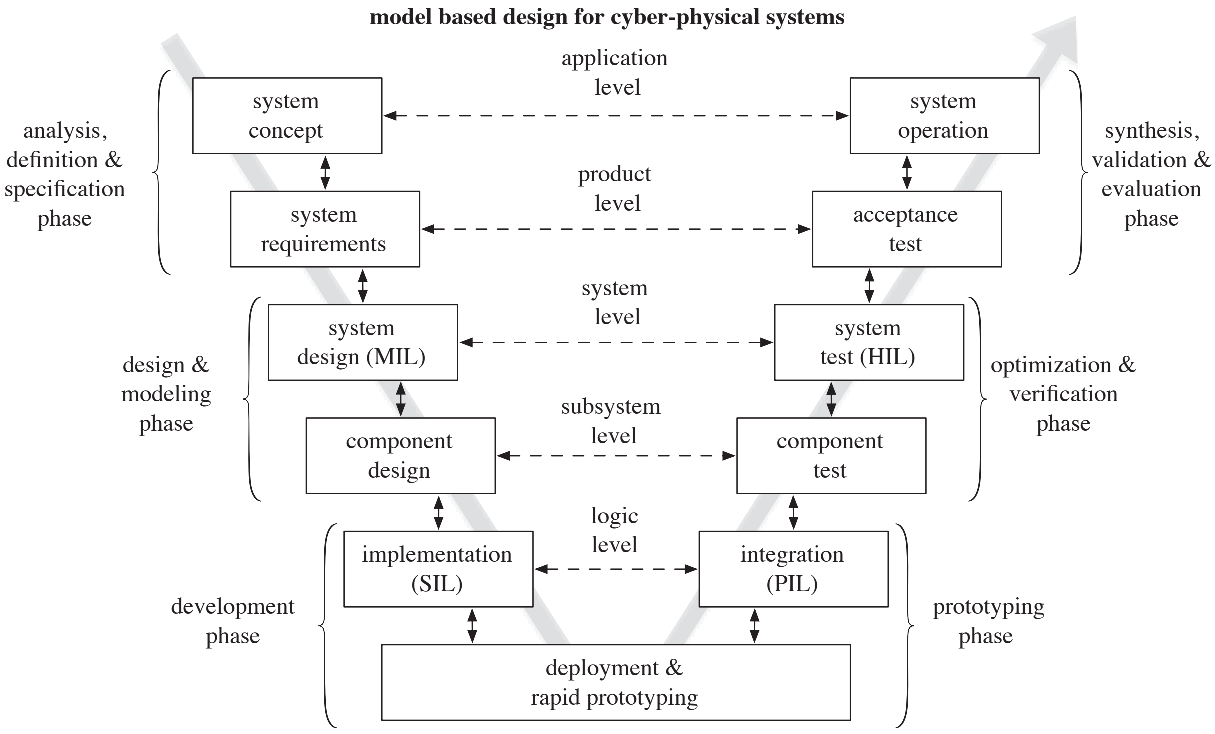


Fig. 3.1 MBD is based on the V-model for software/product development [147, 204].

(1) Application level: system concept and operation. Ideally, a product is the result of a distinct business plan with freedom to operate w.r.t. intellectual properties. A product is a self-containing system, which is defined by its functionality on application level within a given context. According to UML 2.x [160], the required functionality is described via exemplary use cases w.r.t. the system's operation. The goal is to identify numerous use cases to unambiguously define the system concept. It is important to ensure consistency, so that different use cases do not contradict each other or physical principles. The problems to be solved must be characterized and communicated explicitly. Test cases are generated from these use cases to verify the functionality during the system operation via cross validation. Post-production updates are possible using MBD.

(2) Product level: system requirements and acceptance. A product has functional (quality and performance) and non-functional (costs, safety, security, robustness, availability, etc.) system requirements (German: *Lastenheft*) [147]. The product properties are specified on product level. Technically spoken, requirements are constraints on the system which reduce the solution space. The underlying models are formulated during system identification and must abide by the laws of physics. Executable models enable full traceability of the requirements and simplifies acceptance testing, i.e., if the system is working as intended [41]. In contrast, textual descriptions of requirements always bear the potential to be misinterpreted. MBD reduces the need for dedicated requirement tracking tools such as IBM DOORS [94].

(3) System level: system design and test. A system is composed of a number of heterogeneous subsystems (or components/modules) within a structured architecture, this is defined in the system specification (German: *Pflichtenheft*) and part of the project's deliverables. Functional decomposition of the whole system is required to reduce the design complexity [148]. Therefore, an ideally designed system is the integration of these subsystems via defined interfaces.

Model-in-the-loop (MIL) simulation verifies the correctness of the mathematical framework on system level via executable models [32]. This is an efficient way for system identification and model parameter estimation with varying configurations; it is made possible through simulation on abstract model level in an artificial development-environment. Parameter sweeping during simulation allows to measure the system's performance/behavior under varying conditions. A design of experiments test plan helps to find the optimal set of parameters for a system incorporating multiple physical domains. Hardware-in-the-loop (HIL) simulation establishes a verification environment for the hardware's system test utilizing the models [54, 91, 149, 174], see also Fig.3.2.

(4) Subsystem level: component design and test. Each component is designed to fulfill a specific task within its domain and implements standardized interfaces on subsystem level. The challenge is to minimize the dependencies between the individual components. Most commonly, a component itself is partitioned into further (possibly already existing) subsystems. This leads to a natural form of recursion within in the design process and emphasizes the need of functional decomposition in order to handle the design complexity in means of size and complication [147, 148]. Existing subsystems, which have originally been written in other languages such as `C`/`C++`/Fortran, are integrated by wrapping the functions to match the interfaces. Each component must be tested individually [32].

(5) Logic level: implementation and integration. Automatic programming converts the implemented hardware-independent model code into the target machine code, both executable programs must be functionally and numerically equivalent on logic level: a certain input must deliver the same output on all abstraction layers [107]. Software-in-the-loop (SIL) verification is used to proof the functional/numerical equivalence of the model and the generated code on the development system [32]. The results computed by the model are compared with the results computed by the compiled program in the target language. This is of particular importance when the target language misses certain features of the model language. Note, that code is compiled differently depending on the operating system (most commonly Windows or UNIX based) and the CPU architecture (x86 or x64). Processor-in-the-loop (PIL) verifies the correct computation of the integrated code on the embedded system, see Fig. 3.2 (c). The results computed by the embedded system are fed back to the host system for comparison with the results computed by the aforementioned. The outcome is not necessarily the same, because the hardware platform of the host and target system are architecturally different (see Section 3.2).

(6) Deployment and rapid prototyping. The best set of parameters is deployed onto the target hardware. Depending of the process' degree of automation, this step requires manual configuration, e.g., tying the synthesized code to hand-written, hardware specific code. Automatic targeting facilitates rapid prototyping using HIL simulation [91], see Fig. 3.2 (d).

3.1 Automatic Programming

An embedded system is a dedicated computer system designed to perform specific functions, usually as part of a complete device or system. In a CPS sense, it is characterized by tight integration and coordination between computation and physical processes by means of networking. Embedded systems directly interact with the physical space; as a matter of fact, physical components are qualitatively different from conventional object-oriented software components. There are three relevant low-level target languages for embedded systems [122]: multi-purpose ANSI-C code, which is also the focus of this thesis; hardware description languages (HDL) such as VHDL [99] and Verilog [100] for field programmable gate arrays (FPGAs) or application specific integrated circuits (ASICs); and IEC 61131-3 compliant languages [97] such as structured text for programmable logic controllers (PLC). These hardware platforms are explained in Section 3.2.

Automatic programming is the task of converting the abstract models and their algorithms to usable machine code, effectively automating the time-consuming and error-prone process of low-level programming, i.e., the time-to-market is shortened significantly [121, 123, 147, 148]. The OMG defines this procedure as model driven architecture. The concept of automatic code generation, software synthesis and targeting has been described for various problems, such as convex optimization [141], agent-oriented software [83] and control units [183]. There has been a study [24] with control groups trying to write model-equivalent code manually in C, but they never got anywhere near as far as those who use MBD. The suitable model remains fully scalable during embedded targeting, because standard C code can be deployed on various platform. It is not possible to map every model function into an equivalent C function; however, mathematical models using only linear algebra are always translatable.

When using a certified code generator, the generated code complies to common programming paradigms such as the MISRA[3] C guidelines [145, 148]. Embedded systems are traditionally integrated in many safety critical industrial applications [205]. Consequently, the quality of the generated code must be in compliance with existing standards, e.g., ISO 26262 [31, 33, 104] or AUTOSAR[4] [110, 189] for the automotive and DO-178C [184] for the aerospace industry. MathWorks Polyspace [31, 140] is a verification tool to test C code against these standards.

Production code is generated after finishing the verification and validation process. All parameters needed during testing/debugging are stripped out and the code is optimized for performance (low memory demands, high computational speed) or safety (data consistency, robust algorithms) [32, 147]. Modern C/C++ compilers such as those from Microsoft Visual Studio [143], Intel Composer [103] with Math Kernel Library (MKL) or ARM Compiler [6] recognize linear operations and manual pragmas and replace those hand-written lines with suitable processor instructions [122]. These automatic compiler optimization improves the computation speed by a magnitude of up to 1000; furthermore, computations supported processor instructions are more energy efficient. The code is revised statically on soft- and hardware levels [122]:

1. The C code is optimized on source-code level in a **machine-independent** manner, which is beneficial for most programs. Standard optimization techniques include constant folding, loop transformations and function inlining;

2. Special operations are mapped to the controller's corresponding **machine-specific** instruction sets. The set of used techniques includes vectorization for continuous addressing in memory, permutation of operations and parallelization of computation tasks [198]. It was shown, that this mapping is an NP-hard partitioning problem [47, 122] and its solvable with a meta-heuristic approach. The assembly code is improved w.r.t. memory access organization and instruction scheduling.

[3] Motor Industry Software Reliability Association, www.misra.org.uk

[4] AUTomotive Open System ARchitecture, www.autosar.org

The LLVM[5] compilation framework introduces *code-lifecycle optimization*, which improves the code dynamically during run- and idle-times depending on the application/user. Furthermore, LLVM enables cross compilation on the host system via processor virtualization using a hardware-independent intermediate language [117]. The code can be compiled directly on the target system if an operating system (OS) and an appropriate compiler is present. The production code is verified with HIL. The complete workflow is visualized in Fig. 3.3.

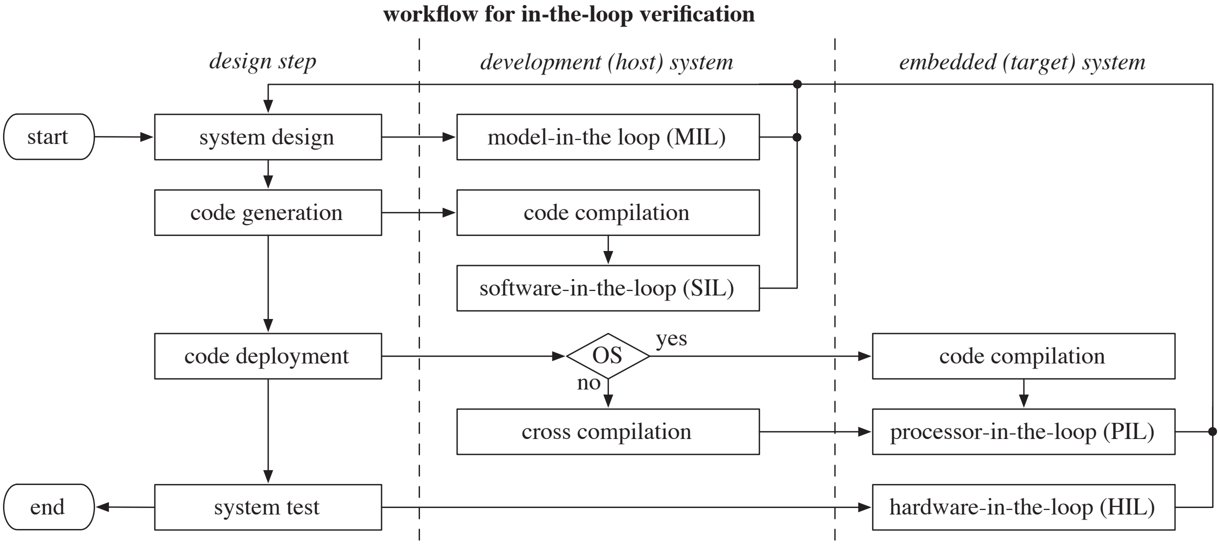


Fig. 3.3 MBD is an iterative development process with multiple verification steps. The correctness of the model-code translation must be tested for discrepancies, i.e., if the generated code is functionally and numerically equivalent to the model on code/software (SIL), processor (PIL) and hardware (HIL) level [32]. The code is either compiled on the target system if an operating system (OS) with a compiler is available or it is cross compiled on the development system.

Most industrial controllers are programmable with `C`. Table 3.1 gives an overview of engineering and scientific software for designing mathematical models and the functionality to automatically generate standard ANSI-`C` code from their application-specific syntax. A short survey on MBD approaches has been given in [34, 35, 148], tools for automatic code generation are summarized in [179]. SystemC [102] is a library for `C++` which introduces classes for event-driven simulation, but lacks the required abstraction of a MBD tool [54].

This thesis is focused on the use of MATLAB and its Coder Toolbox, because it is the standard software in industry for mathematical MBD. Although MATLAB is a commercial programming language, it is widely used[6] for engineering and scientific applications. MATLAB Coder features the ability to replace model function calls with calls for MATLAB executables (*.mex) for SIL verification. Such functions are wrappers around compiled `C` code, which can be directly called from the MATLAB development environment. Code generation fully supports linear algebra. Alternatively, the code for SIL can be compiled by any integrated development environment (IDE) of choice supporting `C`, e.g., Microsoft Visual Studio, Eclipse or Apple XCode.

[5] The LLVM Compiler Infrastructure, `llvm.org`

[6] TIOBE Programming Community Index, `www.tiobe.com`

software package	**toolbox**
MathWorks MATLAB [133]	MATLAB Coder
MathWorks Simulink/Stateflow [134]	Simulink Coder, Embedded Coder
National Instruments LabVIEW [155]	C Generator
Maplesoft Maple [131]	CodeGeneration Package
Wolfram Mathematica [209]	CCodeGenerator

Tab. 3.1 A selection of engineering tools and corresponding toolboxes supporting automatic C code generation directly from their IDE. Simulink Coder was formerly known as Real-Time Workshop (RTW).

3.2 Embedded Systems

The development (host) system is significantly different from the embedded (targeted) system. Modern central processing units (CPU) have a complex instruction set computer (CISC) von Neumann architecture with 32-bit (x86) or 64-bit (x64), where most microcontrollers employ a modified Harvard architecture. This section gives a very short overview in order to understand the significance for automatic code generation [122]. Embedded systems are roughly categorized into two groups:

1. A **microcontroller unit (MCU)** is a general purpose 8-, 16-, 32- or 64-bit reduced instruction set computer (RISC); MCUs are commonly programmable in C. The most popular designs are ARM's Cortex A/R/M (www.arm.com), Atmel's AVR (www.atmel.com) and Microchip's PIC (www.microchip.com) microcontrollers. MCUs are available with and without operating system, the most popular choice is a (real-time) embedded Linux distribution;

2. A **field programmable gate array (FPGA)** is a special purpose logical unit (programmable hardware). An FPGA contains reconfigurable logic blocks and interconnects (i.e., gate arrays), whereby functions are implemented by connecting these blocks (*netlist*). FPGAs are used for specialized hardware accelerated computationally intensive applications which utilize parallel architecture, e.g., real-time image processing [199]. FPGAs are reprogrammable directly on hardware level to change the functionality (i.e., field programmable). Market leaders are Xilinx (www.xilinx.com) and Altera (www.altera.com). An application specific integrated circuit (ASIC) is basically a migrated FPGA design customized for a particular use. The programming is burned-into the hardware, which makes the chip cheaper in mass production. ASICs are faster and more energy efficient, but cannot be modified after the design is committed. Bug fixing in post-production is not possible.

An *application processor* describes a heterogeneous computing system, i.e., a highly integrated system-on-chip (SoC). A general purpose MCU may be paired with a specialized co-processor, for instance a digital signal processors (DSPs) for filtering streaming data or graphic processing units (GPUs) for parallel computing. These special instruction sets must be known to the compiler for optimization w.r.t. automatic function replacement. Xilinx offers the possibility to integrate a fully functional ARM Cortex-A processor into its Zynq FPGA[7]; this facilitates the fact, that the boundaries between processor architectures are not intrinsically clear.

The OS, if one is present, must be considered. An OS is an abstraction layer for resource management between the hardware and the applications. Application programming interfaces (API) offered by the OS avoid the need of reprogramming low-level functions. The non-deterministic behavior of an OS can be a problem for embedded systems in real-time applications. Real-time operating systems (RTOS) possess predictable scheduling policies, but do only offer a limited range of services.

Rapid prototyping is simplified using development platforms such as Arduino (`www.arduino.cc`), Raspberry Pi (`www.raspberrypi.org`) and BeagleBoard (`www.beagleboard.org`). The employed MCUs are based on ARM and/or Atmel AVR designs. The boards feature various general purpose input/output (GPIO) interfaces such as I2C/TWI, SPI, UART and/or USB. The models described in this thesis are deployed on these development boards for PIL testing as shown in Fig. 3.2 (c).

The algebraic models presented in Part II are completely general and are meant to be deployed on embedded systems, see Fig. 3.4. The number of required floating point operations is known a-priori to the execution of the linear operations; consequently, these operations are capable for real-time computations by definition[8].

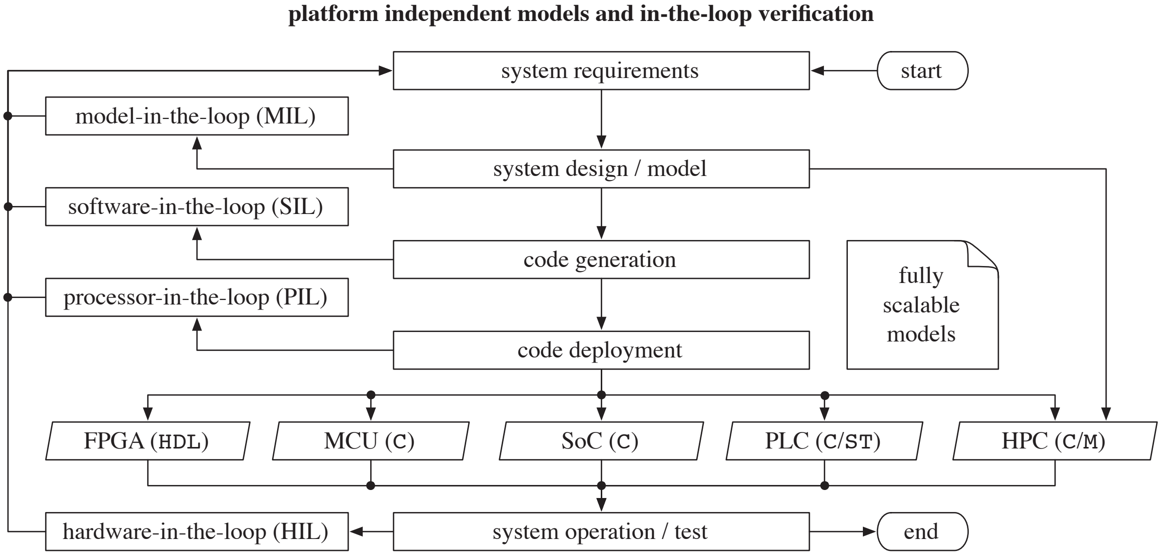


Fig. 3.4 Generic models can be deployed on low level FPGAs (field programmable gate arrays) using `HDL` (hardware description level) code, MCUs (microcontroller units) using `C` code, system on a chip (SoC) application systems using `C` code, industrial PLCs (programmable logic controllers) using `C` code or `ST` (structured text), and HPC (high-performance computing) systems using `C/C++` code and/or high-level model code (`M`).

[7] Zynq-7000 SoC, `www.xilinx.com/products/silicon-devices/soc/zynq-7000/`

[8] A real-time system is defined as any information processing activity or system which has to respond to externally generated input stimuli within a finite and specified period [214].

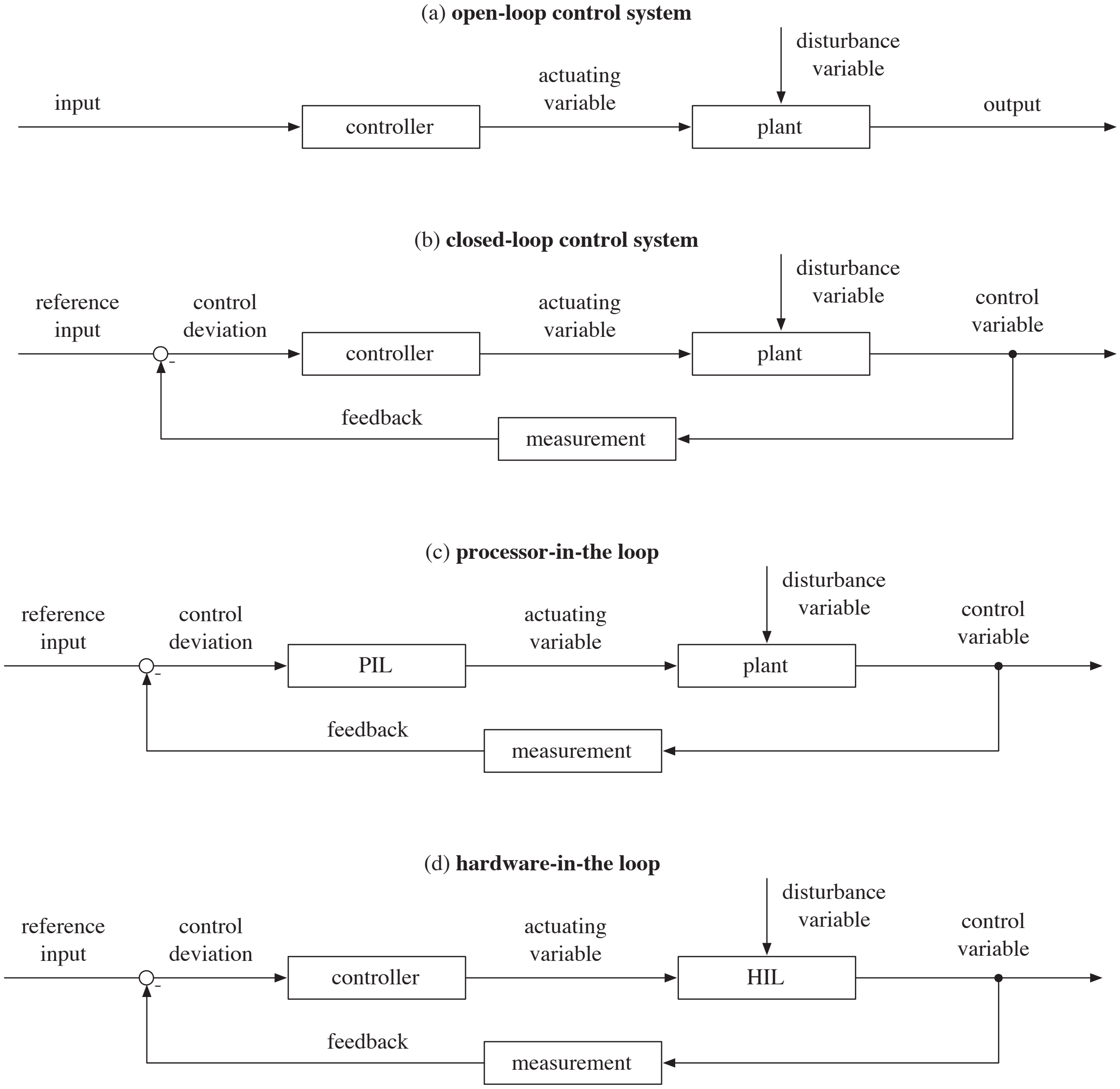


Fig. 3.2 The output of an open-loop system (a) has a direct functional relation to the input, this architecture is equivalent to a finite impulse response (FIR) filter. A closed-loop system (b) considers previous outputs of the system, this architecture is referred to as the classical *control-loop*. During processor-in-the-loop (c) verification, the embedded system is targeted with the automatically generated code, i.e., the controller's correct behavior w.r.t. its functionality and numerical behavior is verified. Hardware-in-the-loop (d) replaces the plant with a functionally equivalent model in order to test the complete control loop with real data emerging from the physical system.

4 | Machine Vision

This chapter provides the summarized background knowledge for the projects involving optical measurements presented in Part III. The Chair of Automation has done extensive research in machine vision in the past, mainly with applications derived from the metal processing industry [53, 69, 87, 111, 185, 188, 199, 203], quality assurance and inspection via infrared thermography [171, 182, 190] and condition monitoring for underground tunneling/mining operations [70, 71, 157]. The subtopic of metric vision covers the optical measurement of quantitative information for geometric objects, such as position, orientation, dimensions and shape. In a CPS context, machine vision systems detect physical phenomena, process the data on a(n) (embedded) computing system and transmit the information to a supervisory system.

4.1 Optical Arrangement

The quality of the acquired raw imaging data strongly depends on the arrangements of the optical components and the environmental conditions. The setup of the measurement scene is the origin of most systematic errors, which must be compensated during image processing. It is recommended to establish the best physical conditions for image acquisition within the application's constraints, i.e., cost and time, before processing the data.

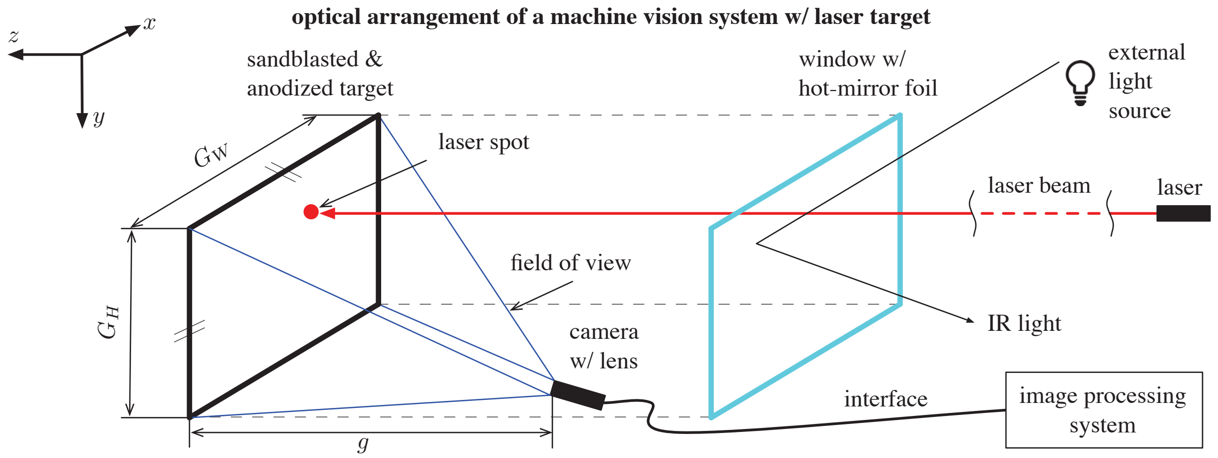


Fig. 4.1 The measurement chain of a general purpose machine vision system consists of hardware (optical) and software (image processing) components. The thesis focuses on optical measurement systems which are based on the use of targets and laser light. The laser beam passes through a glass window and impinges upon the specially prepared target; the resulting laser spot is then captured by a camera, the raw data is sent to an embedded or remote image processing system.

(1) Target. The target is the object of interest observed by the optical measurement system; most commonly, it is attached to the controlled machine. The physical size G of the target must be specified in the system requirements. Following measures increase the detectability of a laser spot or line on a metal target and reduce the influence of external disturbing light sources:

1. **Sandblasting** the target is the attempt to diffuse the surface's reflectance, which ideally has the same radiance when viewed from any angle. This desired property is called *Lambertian* and exhibits Lambertian reflectance according to Lambert's emission (cosine) law [127, 187]. This means the radiance is constant for the observer, because the emitted light and the corresponding observed area are reduced by the same factor; this effect is shown in Fig. 4.2. Consequently, the scattered light generated by a laser beam is angle invariant for the camera observing the laser spot/line;

2. **Anodizing** the metal target's surface via electrolytic passivation thickens its natural oxide layer, which makes it more durable against corrosion, especially when the target's surface has been roughened via sandblasting. Furthermore, anodizing aluminum facilitated dying [19]. A black target incorporates the highest contrast for an imaging system, whereas a colored target provides the highest reflectance rate within the associated spectrum of visible light;

3. **Hot mirrors** are dielectric mirrors which reflect the radiation energy of infrared light (IR) while allowing visible light to pass, effectively implementing a high-pass filter. The lower bound of reflected wavelengths is in the range of approximately $750 \ldots 1250\,[\text{nm}]$. The spectrum of white light contains large portions of IR light, which can negatively influence an optical measurement. If the instrument employs a window, a hot mirror foil can be applied to the glass. Due to the reflection rather than absorption of the radiation, the hot mirror and the housing are not critically heated. The optical system is then effectively protected from overheating [185] and the disruptive IR portions of white light. A comparison of hot mirror coatings for protecting camera equipment has been given in [185].

In certain applications, e.g., when multiple targets are set up as a consecutive chain, semi-transparent targets are required. A possible solution is electro-active glass [70, 71], which can switch from opaque to translucent, or screen-printed patterns on a glass plate [157, 158], see also the project description in Chapter 11 and 14.

(2) Lighting and/or laser light. Appropriate and stable lighting of the environment ensures a robust segmentation of objects. In measurement applications, the lighting is often put into effect with a laser. A laser is characterized by its power P and its wavelength λ. The emitted light's wavelength and frequency ν are related via $\lambda\,\nu = c$, i.e., the product between both factors is constant, whereby c denotes the speed of light[1]. This means shorter wavelengths cause higher frequencies. According to quantum mechanics [126, 127], the emitted energy E of a single photon is $E = \hbar\,\nu$, where $\hbar$ denotes the Planck constant[2]. The energy and power are related via $E = P\,t$, where t is time. As a consequence, lasers with shorter wavelengths (higher frequency) are more energetic than those with longer wavelengths (lower frequency). This must be considered when using lasers in working environments: the European Union Council directive

[1] speed of light $c = 299792458\,[\text{m/s}]$

[2] Planck constant $\hbar = 6.62606957(29)\,10^{-34}\,[\text{J s}]$

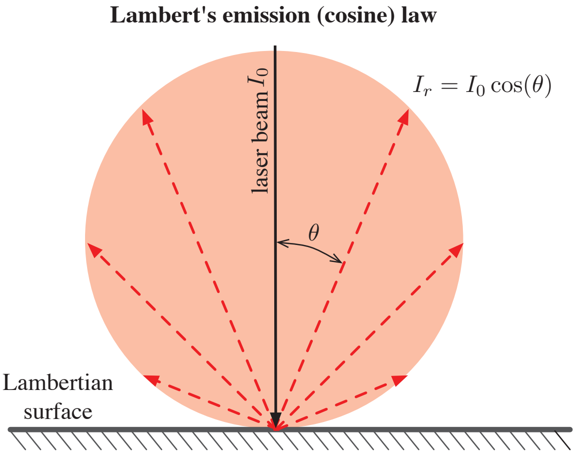


Fig. 4.2 When the target's Lambertian surface is illuminated by an external source I_0, i.e., the laser beam, then the irradiance I_r (power density) is proportional to the cosine of the angle θ.

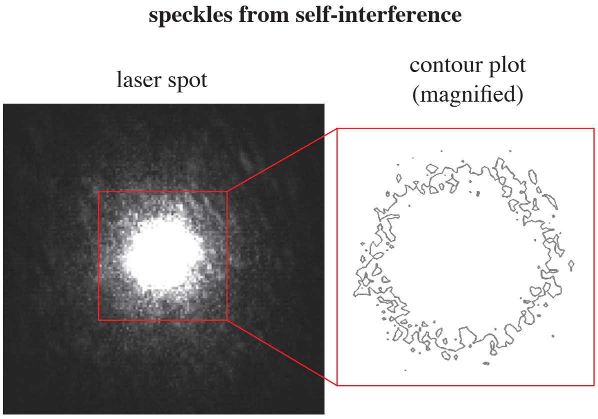


Fig. 4.3 Laser light is a coherent optical radiation. Parts of the laser light are reflected from the target, leading to self-interference. Speckles are the visible result. (Image from [70].)

2006/25/EC [50] on artificial optical radiation clearly defines thresholds for exposure limit values in dependence of exposure time, wavelength and area[3]. The directive uses the guideline for the *maximum permissible exposure* (MPE) value established in the IEC 60825 − 1 on safety of laser products [96], which also defines the classes of laser products. Considering these safety issues, optical instruments should employ visible red lasers with relatively short wavelengths if possible, see Fig. 4.4.

Unfortunately, the wavelength directly corresponds to the measurable objects size. In order to avoid diffraction[4] from wave effects [13, 126, 128] in practical applications, a the minimal wavelength λ_{min} which is at least five-times larger than the object size G should be chosen, i.e., $\lambda_{\text{min}} \geq 5\,G$ can be used as a rule of thumb.

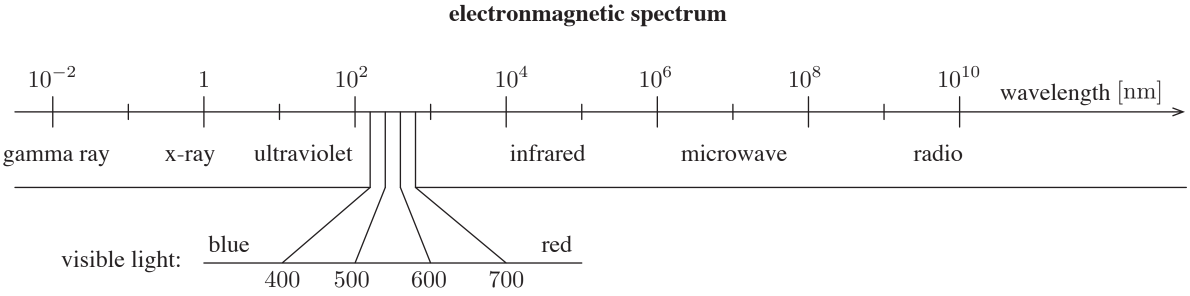


Fig. 4.4 The electromagnetic spectrum is the range of all frequencies of electromagnetic radiation. The laser light's color directly corresponds to its wavelength, whereby shorter wavelengths bear higher energy [181, 185]. Humans see light of in the range of approximately $\lambda = 400 \ldots 800$ [mm], whereas most cameras reach into the near IR range, covering $\lambda = 400 \ldots 1000$ [mm], which justifies the employment of hot mirrors in optical measurement applications.

[3] The critical area is a circle with ⌀7 [mm], which is approximately the size of a human eye's pupil.

[4] This is known as the Airy disc effect [2].

Emitted laser light is coherent; thus, self-interference (or *autocorrelation*) in the form of speckles [126] occurs when the laser beam is reflected from a normal surface, i.e., some spectral components are amplified and some are eradicated (see Fig. 4.3). This noisy behavior must be considered during measurement. Coherent light allows the use of interference filters. The filter acts as a bandpass, which transmits only a certain range of wavelengths, whereas wavelengths not in the range are blocked, effectively filtering polychromatic light to monochromatic light. As a side effect, interference filters are angle-dependent and will not work properly when mounted on wide-angle lenses.

(3) Lens. The employed lens is the main influencing factor of the camera's field of view. The lens transmits and refracts light, converging the beam of light to the imaging sensor. Commonly, the width G_{W} and the height G_{H} of the target are specified in the system requirements as the object size G. The object distance g is a function of the lens' focal length f and the size B of the imaging sensor. Common sensors are rectangular, leading to a sensor width B_{W} and height B_{H}. Be aware, that the height and width of the object and the sensor are aligned to each other. Fig. 4.5 shows the simplified model of a lens.

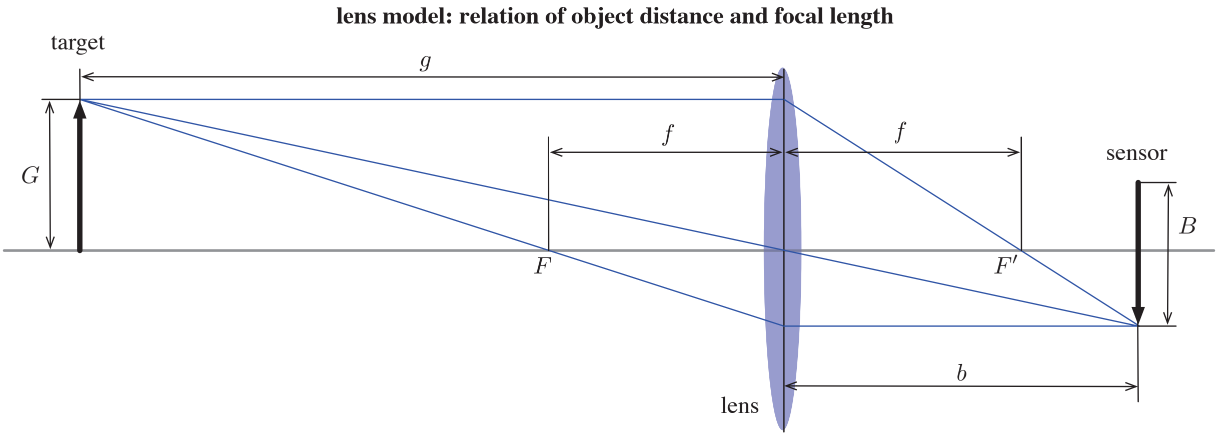


Fig. 4.5 This simplified lens model consists of a single optical element. The object distance g depends on the lens' focal length f, the size G of the target and the size B of the imaging sensor. The sensor distance b is determined by the camera mount's flange focal distance, see Table 4.2.

According to [126, 181, 185], the parameters are related via

$$g_{\mathrm{H}} = f\left(\frac{G_{\mathrm{H}} + B_{\mathrm{H}}}{B_{\mathrm{H}}}\right) \quad \text{and} \quad g_{\mathrm{W}} = f\left(\frac{G_{\mathrm{W}} + B_{\mathrm{W}}}{B_{\mathrm{W}}}\right), \quad \text{whereby} \quad g = \max(g_{\mathrm{H}}, g_{\mathrm{W}}). \tag{4.1}$$

The lens' angle of view ϕ is calculated via

$$\phi_{\mathrm{H}} = 2\tan^{-1}\left(\frac{B_{\mathrm{H}}}{2f}\right) \quad \text{and} \quad \phi_{\mathrm{W}} = 2\tan^{-1}\left(\frac{B_{\mathrm{W}}}{2f}\right), \quad \text{whereby} \quad \phi = \max(\phi_{\mathrm{H}}, \phi_{\mathrm{W}}). \tag{4.2}$$

Considering Eqn. (4.1) and (4.2), the achievable values for the object distance g and angle of view ϕ do not only depend on the focal length f, but also on the size B of the imaging sensor! The dimensionless focal ratio (or F-number) is commonly denoted by $1/\#$; it describes the ratio between the focal length f and the diameter d of the aperture, i.e., $1/\# = d/f$. The lower this value gets, the more light can pass through the lens; best image results will appear

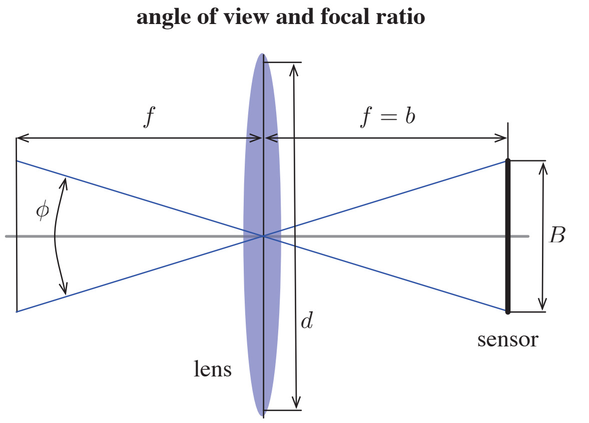


Fig. 4.6 The angle of view ϕ depends on the focal length f of the lens and the size B of the sensor. The calculation is based on the presumption, that f is equal to the sensor distance b. In order to avoid vignetting effects, d must be greater than the diagonal of the imaging sensor.

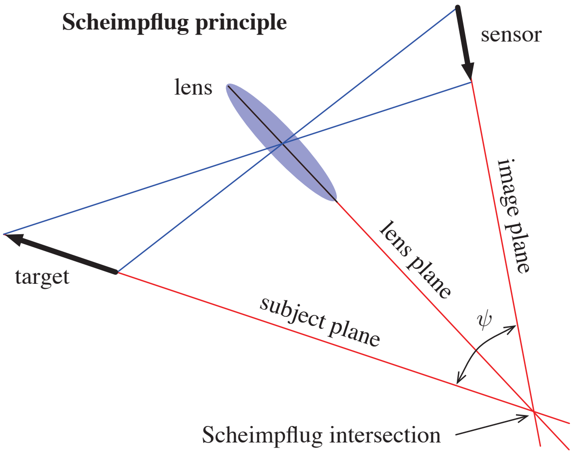


Fig. 4.7 The Scheimpflug principle ensures a correct depth of field for a non-parallel orientation between the subject plane and the image plane, i.e., both planes enclose the angle ψ. A practical example is given in Chapter 13.

with $f/\# = \{4, 5.6, 8\}$. With lower F-numbers, the image is blurred by aberrations; for high focal ratio, it is blurred by diffraction [13]. A special lens-sensor setup is based on the so called Scheimpflug principle [128, 181], see Fig. 4.7. This geometric rule describes the orientation of the plane of focus of a camera when the subject plane is not parallel to the image plane, i.e., the depth of field encloses an angle ψ with the image plane.

An advanced compound lens is composed of an array of simple lenses with a common axis, which allows correction of optical aberrations; however, such lenses are very cost intensive. For instance, telecentric lenses produce orthographic views of the subject which are free of perspective. Often lenses with a simple setup are chosen because of economic reasons. Fig. 4.8 gives an overview of optical distortion effects. To summarize, a shorter object distance g requires a shorter focal length f and/or a sensor with a bigger size B, which causes a greater angle of view ϕ, but also incorporates a greater optical distortion.

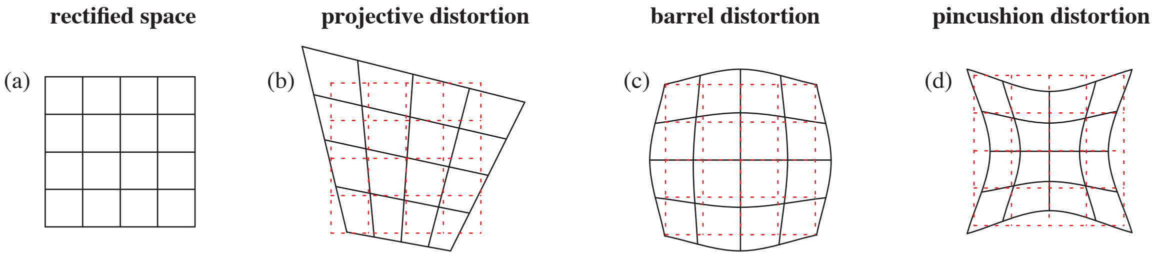


Fig. 4.8 Optical distortion is mainly caused by the radial components, i.e., distortion is a function of the displacement from the center. Rectification is the process of transforming the non-uniform lattice to a Cartesian grid (a). Projective distortion (b) is linear, whereas barrel (c) and pincushion (d) distortions are non-linear. Practical applications most commonly employ a combination of (b) and (c) when using wide-angle lenses.

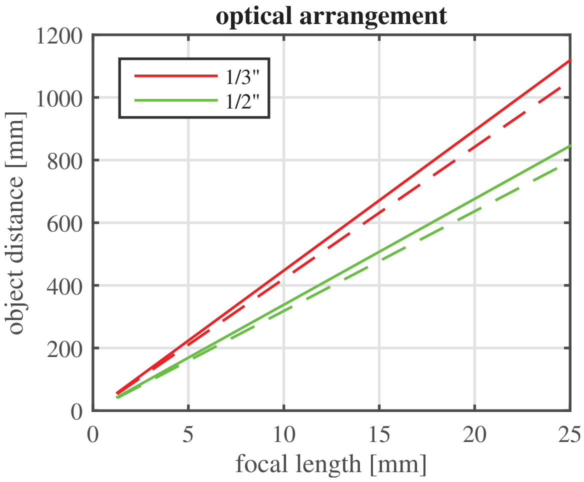


Fig. 4.9 The plot shows the linear relation between the focal length f and the object distance g (g_W solid line, g_H dashed line) for two common sensor sizes (1/3" red, 1/2" green). The target size G has been arbitrarily chosen to be equal to a metric A5 paper size ($G_\mathrm{W} = 210\,[\mathrm{mm}], G_\mathrm{H} = 148\,[\mathrm{mm}]$). Note, that this simplified model becomes more inaccurate for small values of f; nevertheless, it allows a good estimation for the magnitude of g.

Lenses are interfaced to the camera's body via standardized lens mounts, which are basically screw threads. The flange focal distance is the distance from rear of the lens to the image plane. Cheap lenses often require a shorter flange focal distance (CS-mount) than standard industrial lenses (C-mount). The S-mount appears on imaging systems which are attached directly to a printed circuit board (PCB). An overview is given in Table 4.1.

type	thread	flange focal distance
S-mount	M12×0.5	unspecified
C-mount	1" × 32 TPI	17.526 [mm]
CS-mount	1" × 32 TPI	12.526 [mm]

Tab. 4.1 Overview of lens mounts used in industrial and scientific applications [197].

(4) Imaging sensor. There are two dominant technologies for imaging sensors available: CCD (charged coupled device) and CMOS (complementary metal oxide semiconductor); both types are based on the photoelectric effect. CCD sensors have their sensitivity maximum at 550 [nm] (green, same as humans) and CMOS sensors have their sensitivity maximum between 650 . . . 700 [nm] (red). CCD sensors have one controller for all pixels, whereas CMOS sensor have an individual controller for each one of their pixels. This enables the acquisition of only a certain region of interest (ROI) via windowing or the dynamic reduction of the resolution via binning. CCD sensors acquire images always as a whole data set, but possess a better responsiveness; hence, they have lower dark noise than CMOS sensors. On the other hand, CMOS sensors have a higher full-well capacity. If CCD sensors are over-saturated, blooming and smearing effects occur, which is not possible for CMOS sensors. Furthermore, CMOS sensors are faster, because all functions are placed on the imaging sensor. Consequently, CCD sensors are best used in scenarios with bad illumination because of their better noise behavior. CMOS sensors are best used for fast image acquisition with appropriate lighting.

Sensors with higher resolutions require higher quality lenses and better lighting. High-resolution images incorporate greater amounts of data with a higher noise level, lower the possible framerate

size	1/4”	1/3”	1/2”	1/1.8”	2/3”	1”	4/3”	full
B_W [mm]	3.65	4.80	6.40	7.18	8.80	12.70	17.30	36.00
B_H [mm]	2.74	3.60	4.80	5.32	6.60	9.50	13.00	24.00

Tab. 4.2 Selection of available imaging sensor formats [197].

and image processing becomes more computationally expensive. CMOS sensors feature windowing and binning, which can effectively reduce the amount of data to be processed [208]. Thanks to sub-pixel accuracy, low-resolution sensors can be employed in most industrial applications and still deliver results with acceptable accuracy. The size of an individual pixel depends on the size of the sensor, Tab. 4.2 gives an overview of available sensor sizes. Note, that the inch-values do not describe the actual physical size of the sensor.

Depending on the application, monochrome or color sensors are employed. Color sensors only provide a quarter of the available resolution, because of the RGB Bayer pattern. A further demosaicing step is required to ensure the data quality, this is described in [111, 185]. Depending on the application, a HSV (hue, saturation, value) encoding is to be preferred over the conventional RGB (red, green, blue) color space during image processing [111].

Shutters expose the sensor to the light; there are two types available: the global shutter, which exposes the complete sensor at once, and the cheaper rolling (or interleaved) shutter, which exposes only separate lines of the sensor. As a consequence, the last-mentioned is not suitable for capturing fast moving objects [125, 197].

(5) Interface. The camera’s interface defines the overall architecture of the vision system [56], see Fig. 4.10. An overview has been given in [185, 199]. Common modern interfaces[5] are:

1. **USB Vision** has a data rate of 350 [MB/s] (USB 3.0), supports a cable length of 8 [m], has low CPU load, low latency and jitter, enables plug and play and has an energy management/suspend mode, but multi-camera setups are hard to implement;

2. **FireWire** has data rates of 32 [MB/s] (IEEE 1394a) or 64 [MB/s] (IEEE 1394b) respectively, supports a cable length of 4.5 [m], has a low CPU load, low latency and jitter, but suffers from decreasing manufacturer support;

3. **Camera Link** has data rates of 255 [MB/s] (base), 510 [MB/s] (medium) or 850 [MB/s] (full), supports a cable length of 10 [m], provides PoCL (Power over Camera Link), but requires a dedicated framegrabber hardware and inflicts high costs;

4. **GigE** has a data rate of 100 [MB/s], supports a cable length of 100 [m] (copper) or > 1 [km] (fiber) respectively, provides PoE (Power over Ethernet) when using copper wires [12], is based on standard UDP/IP and supports multicast with a flexible architecture, but inflicts higher CPU load [11].

GenICam is a standard established by the EMVA[6] in order to provide a generic programming interface for controlling arbitrary imaging hardware.

[5] USB2.0 and CoaXPress are considered as outdated.

[6] European Machine Vision Association, `www.emva.org` and `www.genicam.org`

(6) System Architecture. As shown in Fig.4.1, a generic machine vision system basically consists of two main parts: the optical (hardware) and the image processing (software) components. Conventional systems have a camera connected to a computer with an interface described in the previous paragraph. In a CPS environment, there are more flexible implementations available, see Fig. 4.10.

1. The **embedded vision system**[7] has a camera connected to an embedded system, which is either a generic MCU based system or a more specific (industrial) FPGA/ASIC system. This implementation follows more the CPS understanding of monitoring and control via distributed intelligent motes within a sensor network. However, it must be considered that image processing is a computationally intensive task and, therefore, it is not suitable for low-end embedded systems;

2. A **smart camera** is attached to a network interface, e.g., GigE. The camera is an imaging sensor in the physical space. Either the raw data is transmitted to the PC or basic preprocessing tasks are already carried out on the camera. The GigE interface enables a dynamic network of cameras, e.g., for monitoring an object from different point of views.

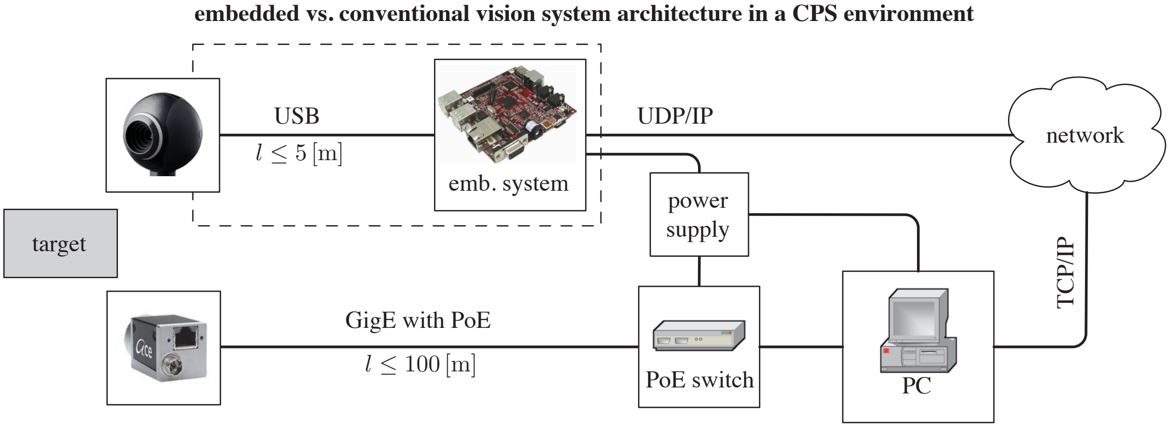


Fig. 4.10 Vision systems are either implemented as decentralized, on-site embedded vision systems, performing only basic image processing tasks, or as centralized remote vision systems, performing complex image processing tasks with data acquired from multiple sensors/cameras. The system architecture is defined by the used interface, which also restricts the maximal cable length l.

4.2 Image Processing

Image processing is an adaptive procedure, the process illustrated in Fig. 4.11 is a possible implementation. State-of-the-art approaches for the image processing chain are presented in literature [66, 93, 192]. The sequence of the individual steps is subject to change and depends on the particular application. This is especially true for the process parameters, which are mostly based on experience.

Simplifications and assumptions are essential for image processing, strengthening the importance of adequate verification techniques such as cross-validation. The measurement error propagates

[7] Embedded Vision Alliance, `www.embedded-vision.com`

non-linearly through the whole measurement chain. The non-deterministic behavior causes interactive effects where errors annihilate each other, may leading to results with higher accuracy.

Many projects at the Chair of Automation utilize image processing as the tool of choice for the analysis of measurement data, especially for automatic inspection in quality assurance. Detailed descriptions for the light-sectioning procedure with structured light and associated metric methods have been given in [181, 188, 203]. This includes the measurement and calibration procedures.

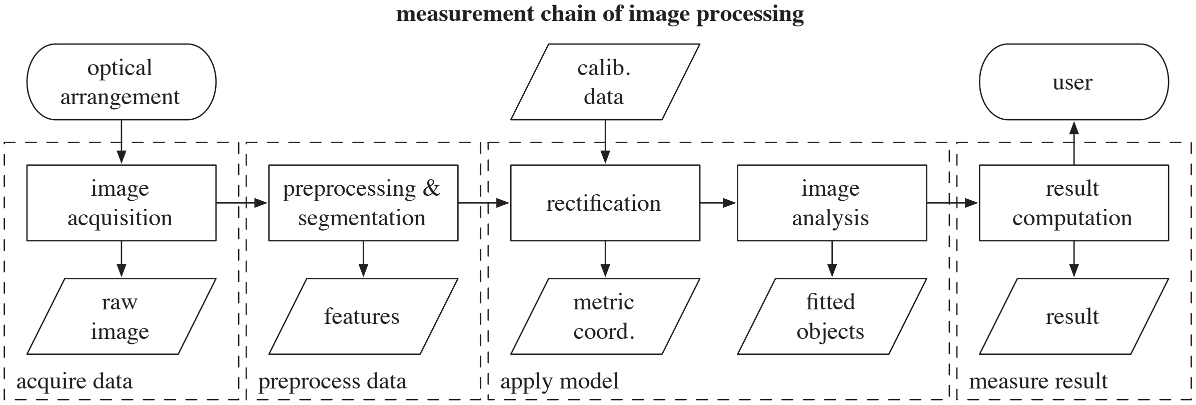
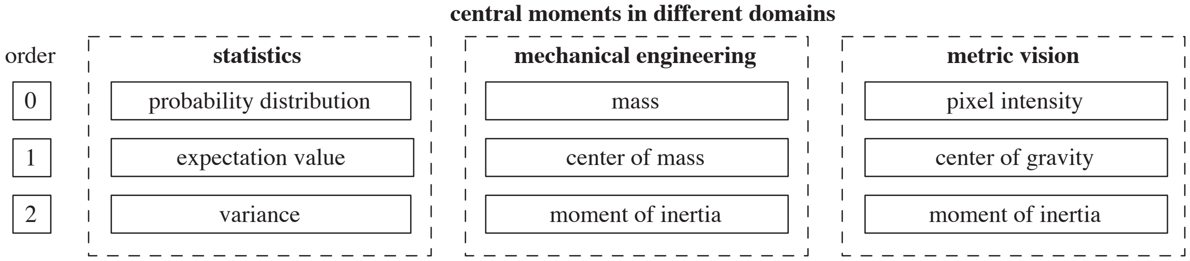


Fig. 4.11 The quality of the raw data from image acquisition strongly depends on the setup of the optical arrangement and the environmental conditions. Preprocessing and segmentation yield features, which are rectified with parameters acquired from an initial calibration step. Metric objects are fitted to these rectified features; the measurement results are based on these fitted objects. Note, that each step implicitly applies simplifications and assumptions.

(1) Image acquisition. The acquisition of single images or sequences of images delivers the raw data and depends on the chosen interface. GenICam greatly simplifies acquisition because of the common interface for all technologies. The primary measurement errors originate in this process. When using a CMOS chip, there's the possibility to read only a certain region of the sensor via windowing or artificially reduce the resolution by merging adjacent pixels via binning. Windowing and binning are performed on hardware level, whereas the related cropping and resizing operations are performed on software level. MATLAB's Image Acquisition Toolbox [137] supports GenICam.

(2) Preprocessing and segmentation. The acquired images are conditioned for further processing. This includes image enhancement operations such as resizing, filtering, contrast normalization, thresholding [70] and morphological operations [111]. During segmentation, data points are grouped to sets of points, which correspond to individual features within a region of interest (ROI). Features are extracted via edge or contour detection and image moment computation [199]. Fig. 4.12 reveals the interesting relations between moments applied in different scientific domains. Moment invariants are of special importance for pattern recognition. MATLAB's Image Processing Toolbox [138] and Computer Vision System Toolbox [135] implement various image processing techniques [66], whereby the last-mentioned also supports automatic code generation.



central moments in different domains

order	statistics	mechanical engineering	metric vision
0	probability distribution	mass	pixel intensity
1	expectation value	center of mass	center of gravity
2	variance	moment of inertia	moment of inertia

Fig. 4.12 Central moments have counterparts in different domains. The probability in statistics corresponds to mass in mechanical engineering and illumination in image processing.

(3) Rectification. Considering the systematic error caused by optical distortions (Fig. 4.8), images must be rectified to achieve accurate results. This is especially the case when deploying wide-angle or fish-eye lenses [185]. Utilizing the reference calibration information, measurement data from pixel coordinates is mapped to metric coordinates, as visualized in Fig. 4.13.

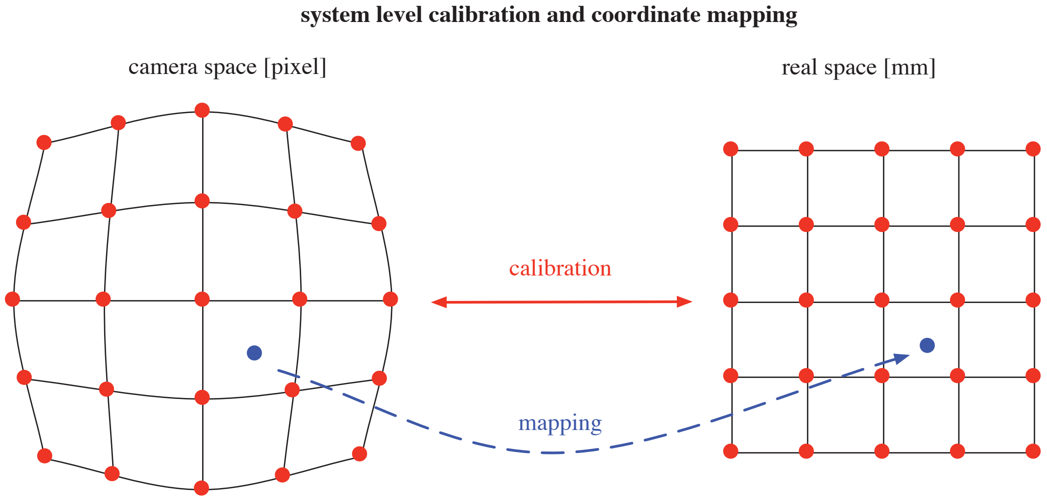


Fig. 4.13 Rectification in image processing is a plane-to-plane mapping process between the camera space in [pixel] and the metric real space in [mm]. The calibration established a logical relationship between both spaces via reference points (red), which are known in both spaces, via a suitable mathematical model. System level calibration effectively cancels out the distortion associated with the optical components and the inexactness of the mechanical construction. This knowledge allows the coordinate mapping of any point from one plane to the other plane.

Possible coordinate mapping methods for 2D-planes are [70]:

1. **look-up-tables** (LUT) via nearest neighbor search or bilinear interpolation;
2. plane-to-plane **homography** via direct linear transformation (DLT) [53, 70, 87, 93, 157];
3. **polynomial control points** [67, 70]; and
4. **bivariate tensor interpolation** [70, 71, 157].

A divide and conquer approach via hierarchical subdivision in the plane, i.e., quad-tree decomposition, is presented in [70, 71] in order to decrease the computational cost. In case of an auto-calibrating measurement system, the calibration procedure is integrated in the measurement chain. The physical structure that serves as the calibration target is visible in the measurement scene and the calibration is performed with the acquired measurement image. Nevertheless, the dimension of the calibration target must be known a-priori.

(4) Image analysis. Abstraction is performed by analyzing the extracted features and assigning geometric objects to these features. The approximation of objects to the sets of extracted points is performed via minimization of an error function [93]. The combination of numerous discrete values leads to sub-pixel accuracy, i.e., the achievable accuracy is higher than treating pixels individually. Elaborate explanations of metric object fitting via Grassmanian manifolds have been given in [53, 87, 111, 188, 203], this includes fitting of (parallel and/or othogonal) lines, higher-order polynomial curves, (concentric) conics and splines. Surface description and abstraction is presented in [69, 199].

(5) Result computation. The final step prepares the identified measurement data and its associated uncertainty for the superior control system. This includes the computation of interdependencies and relations between fitted objects in spatial or temporal domain as well as data preparation for machine learning in pattern recognition applications [69, 199]. Spectral analysis and regularization can be utilized to compress the data for transmission to a supervisory system.

Like any measurement system, two types of errors are associated with machine vision systems: the random measurement error and the systematic calibration error [188, 203]. The random error is mainly caused by noise and cannot be avoided completely; nevertheless, providing a better hardware components and an adequate environmental setup also reduces the noise. When using a suitable model, the systematic error can effectively be removed. However, it must be considered, that simpler, i.e., computationally cheap, mapping methods may still fulfill the system requirements w.r.t. uncertainty. In other words, the application's accuracy demands and relation between the method's achievable accuracy and its complexity must be reasonably proportional to each other. This is extraordinary important when deploying methods on embedded systems.

Part II

Algebraic Models

5 | Instrumentation

Instrumentation plays a key role for CPS[1]. Measurement theory is a ubiquitous topic in engineering literature and many algorithms in numerical mathematics are somehow related to processing measured and discretized data [65, 176]. This chapter gives an introduction to classical measurement theory and explains the problems associated to CPS applications in the condition monitoring and control domain. The mathematical theory in the next chapters is a generalized approach to handle discretized data which has been sampled from a continuous signal. The feedback loop of a control system is effectively a measurement system (Fig. 3.2). The measurement process acquires data from a physical system in order to determine the system's current state. The sensor is the interface between the continuous physical space and the discrete cyber space, leading to real-time dynamic hybrid systems.

Established literature on digital signal processing (DSP) [22, 23, 170] implicitly assumes a set of limitations. Only streaming data is handled, i.e., there is no *beginning* and *end* of the data. As a consequence, the conventional DSP models insufficiently describe the signals at the borders of the support and the Runge phenomenon is ignored. Furthermore, DSP theory is limited to periodic signals; any non-zero linear portion within a periodic signal would inevitably lead to an unbounded signal. Any DSP processing chain presumes a digital-analog conversion (DAC) as the final step. In contrast to the established theory, the new framework is designed to handle

1. **finite sets of data** with correct formulations at the borders of the support;
2. **non-periodic signals** with suitable basis functions to reduce the Gibbs error; and
3. a processing chain without the need of **digital-analog conversion** as the final step.

Considering this, it is important to note that every measurement inflicts a discretization error when using digital instruments: measurement values are quantified from the continuous physical space with a finite sample rate and analog-digital converters (ADC) possesses only a finite resolution. Measurements are subject to noise, most commonly modeled as Gaussian (normal) distribution in engineering applications. The random error within sensory data is a physical property and it is impossible to completely avoid noise; this introduces the concept of measurement uncertainty [105]. Systematic error is effectively compensated through a suitable model. The observed system is influenced by the measurement. This is known as Heisenberg's uncertainty principle in quantum mechanics [92]; however, this is also true in the macroscopic scale of classical physics. Objectivity implies that measurement results convey information only related to the system under measurement and not its environment. Considering all these effects, the *true* value of a measurement is never known [105].

[1] A list of common terms in metrology along with their definitions can be found in Appendix A.

Within this thesis, the forward (or *direct*) problem is defined as $x = f(y)$, where y is the input data (measurand), x is the output data (intermediate instrument representation) and $f(\cdot)$ is the forward operation. The corresponding inverse problem is defined as $\hat{y} = f^{-1}(x)$, where $\hat{y}$ is the estimation of the true value of y and $f^{-1}(\cdot) \triangleq l(\cdot)$ is the inverse operation incorporating contextual knowledge about the associated forward process. Classical measurement literature [55] does not clearly differentiate between forward and inverse problems.

In contrast to conventional measurement methods, the presented algebraic framework is designed to process multiple measurement values simultaneously. In a statistical sense, discretizing a single measurement variable from a continuous signal is equivalent to processing a univariate random variable. The scalar variables $\{x, y\} \in \mathbb{R}^{(1\times 1)}$ are univariate random variables, whereas the vector variables $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathbb{R}^{(m\times 1)}$ are multivariate random variables in a generalized model. The functions are expanded to $\boldsymbol{x} \triangleq f(\boldsymbol{y})$ for the forward problem and $\hat{\boldsymbol{y}} = f^{-1}(\boldsymbol{x}) \triangleq l(\boldsymbol{x})$ for the inverse problem.

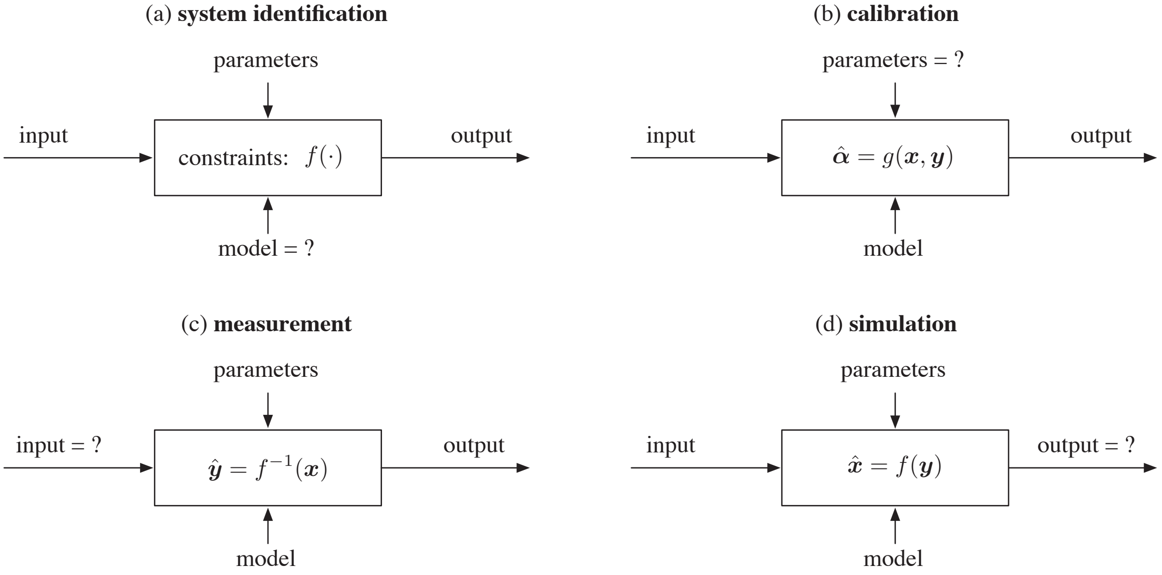


Fig. 5.1 An instrumentation system is defined by a data tuple consisting of the input data $\boldsymbol{y}$, the output data $\boldsymbol{x}$, the model formulated as the forward operation $f(\cdot)$ and the associated inverse operation $f^{-1}(\cdot) \triangleq l(\cdot)$ as well as the parameters $\hat{\boldsymbol{\alpha}}$. Every measurement system victim to noise.

Most generally, the design and execution of a measurement system consists of following steps, see also Fig. 5.1 for a schematic visual representation,

1. The goal of **system identification** is to find a mathematical model to describe the functional relation $f(\cdot)$ between the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ under application specific constraints, see Fig. 5.1 (a). Principally, system identification yields a hypothesis about the system model, which must then be verified during system validation. Systematic error effects are only removed when selecting a suitable model;

2. Having the model $f(\cdot)$ available, the **calibration** is the act of applying a range of known input values $\boldsymbol{y}$ for the purpose of observing the system's output $\boldsymbol{x}$ in order to determine the system parameters, see Fig. 5.1 (b). Depending on the problem's dimension, the parameters $\hat{\boldsymbol{\alpha}}$ are found via the approximation of a curve, a plane or a hyper-plane

respectively. In this thesis, the calibration procedure is designed to deliver a perturbed overdetermined system of equations, i.e., $m > n$, when m is the number of calibration points and n is the number of variables in the equation;

3. **Measurement** is the act of quantifying a specific value $\hat{\boldsymbol{y}}$ to a physical variable $\boldsymbol{x}$ in order to acquire the current state of the system [55], i.e., the cause of a perturbed observation is sought, see Fig. 5.1 (c). The measurement function $l(\cdot)$ is applied during cross validation in order to verify the correctness of the model;

4. During **simulation**, a value for the output $\hat{\boldsymbol{x}}$ is predicted for a specific input $\boldsymbol{y}$, whereby the model $f(\cdot)$ and the system parameters $\hat{\boldsymbol{\alpha}}$ are known, see Fig. 5.1 (d). An anomaly is a significant discrepancy between a predicted value $\hat{\boldsymbol{x}}$ and the true value $\boldsymbol{x}$. The concept of embedded simulation is especially useful for real-time temporal data mining, see also Section 15.4.

A general approach to acquire a maximum likelihood solution for $\hat{\boldsymbol{y}}$ is presented in the next chapters. According to Hadamard [5, 113], a well-posed mathematical model of a physical phenomenon requires the existence, uniqueness and stability of the solution. Problems that are not well-posed in the sense of Hadamard are ill-posed. In this vein, a forward problem is well-posed and an inverse problem is ill-posed, see Fig. 5.2.

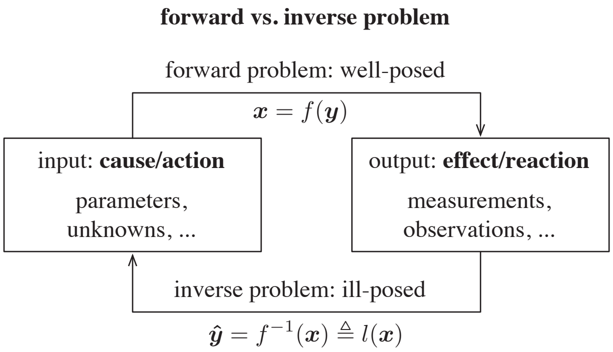

forward vs. inverse problem
forward problem: well-posed
$\boldsymbol{x} = f(\boldsymbol{y})$
input: cause/action
parameters, unknowns, ...
output: effect/reaction
measurements, observations, ...
inverse problem: ill-posed
$\hat{\boldsymbol{y}} = f^{-1}(\boldsymbol{x}) \triangleq l(\boldsymbol{x})$

Fig. 5.2 A well-posed problem always delivers the very same outputs for the same inputs, i.e., one and the same cause/action always inflicts the same effect/reaction. Contrarily, inverse problems are concerned with determining causes for an observed effect [45]. This process is ill-posed, because one effect can be generated by different causes. Regularization adds additional a-prior knowledge to the problem. The generalized solution $\hat{\boldsymbol{y}}$ is then a maximum likelihood estimation for the true value $\boldsymbol{y}$.

5.1 Calibration and Measurement Model

Contextual knowledge about the system under measurement and the involved physical process is required for effective formulation of a suitable mathematical model. This includes the a-priori knowledge about the data's nature, which is either spatial and/or temporal, the geometry of the measured system, the regular or irregular placement of sensors, the type of sensed data and the causality between variables. The problem classes within the thesis' scope are:

(1) Measurements from an overdetermined set of calibration points. The aim is to find a mapping between the measured value $\hat{\boldsymbol{y}}$ and the true value $\boldsymbol{y}$ via a intermediate instrument representation, see Fig. 5.3 (a). This is especially the case when the sensor employs non-linear behavior, see Fig. 5.3 (b) and (c). No unique solution exists, which makes the problems ill-posed by definition; however, a stable solution is possible when adequate constraints are placed on the solution space, i.e., such problems are well-conditioned. A preferred solution is derived using a maximum likelihood estimation via least squares approximation. The residual vector $\boldsymbol{r} = \boldsymbol{y} - \hat{\boldsymbol{y}}$

is the difference between the physical value $\boldsymbol{y}$ and its estimation $\hat{\boldsymbol{y}} = \mathsf{B}\,\hat{\boldsymbol{\alpha}}$, whereby B is a set of basis functions and $\hat{\boldsymbol{\alpha}}$ are the estimators for the coefficients. The cost function K which must be minimized is then

$$K = \|\boldsymbol{y} - \mathsf{B}\,\hat{\boldsymbol{\alpha}}\|_2^2. \tag{5.1}$$

Using the theory from Section 6.4 and Section 9.1 yields the estimation $\hat{\boldsymbol{\alpha}}$ for the coefficients

$$\hat{\boldsymbol{\alpha}} = \mathsf{B}^+\,\boldsymbol{y}, \tag{5.2}$$

where B^+ is the Moore-Penrose pseudoinverse of B. The basis B depends on the values of $\boldsymbol{x}$, i.e., $\mathsf{B} = \mathsf{B}(\boldsymbol{x})$. Consequently, the estimation for the coefficients $\hat{\boldsymbol{\alpha}}$ can be computed from a set of $\{\boldsymbol{y}, \boldsymbol{x}\}$ calibration points, i.e., the calibration function is defined as $\hat{\boldsymbol{\alpha}} = g(\boldsymbol{x}, \boldsymbol{y})$. The mapping between $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ is then given as a linear operation $\mathsf{L} \triangleq \mathsf{B}\,\mathsf{B}^+$, thus,

$$\hat{\boldsymbol{y}} = \mathsf{B}\,\mathsf{B}^+\,\boldsymbol{y}. \tag{5.3}$$

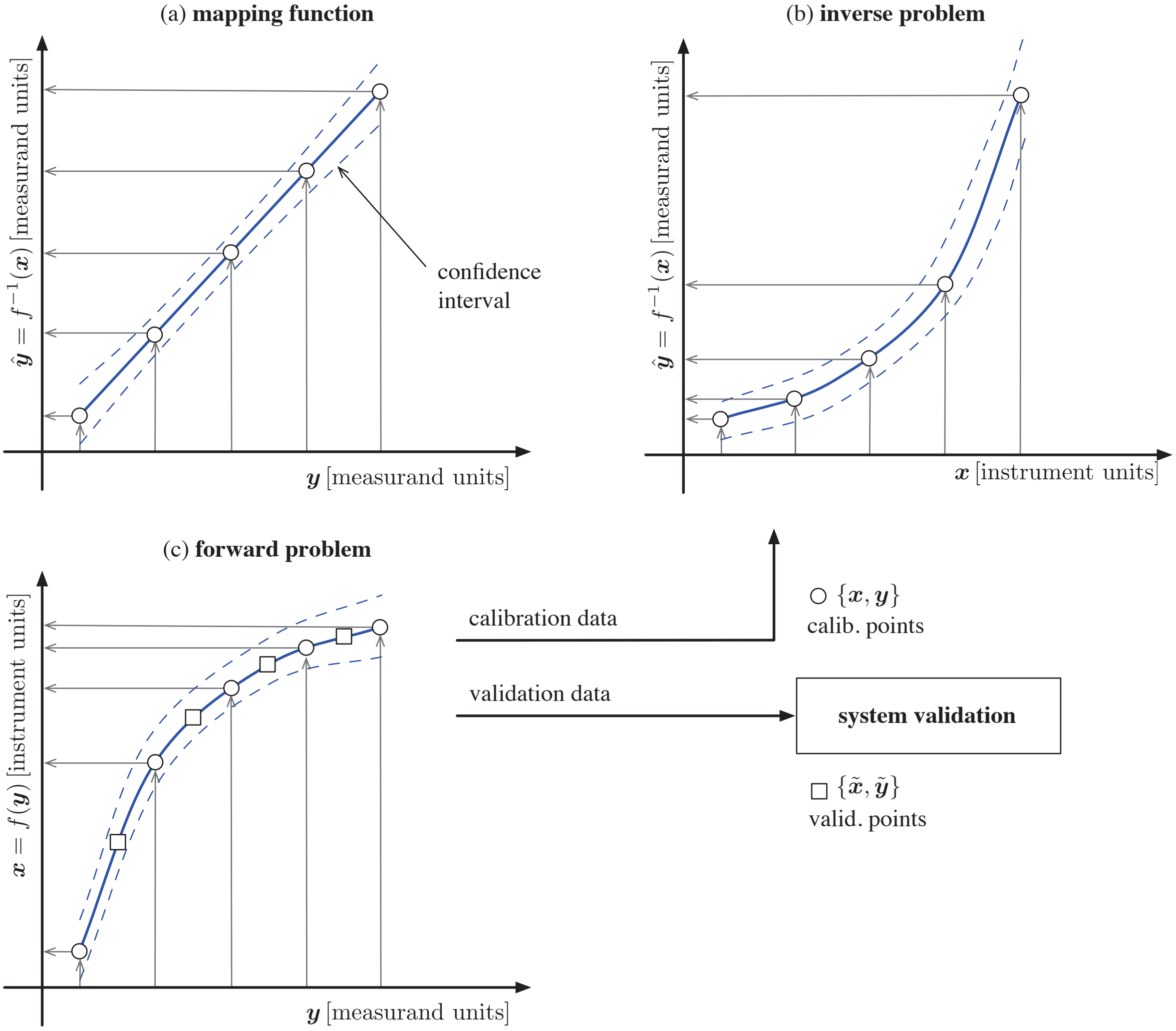

(a) mapping function
$\hat{\boldsymbol{y}} = f^{-1}(\boldsymbol{x})$ [measurand units]
confidence interval
$\boldsymbol{y}$ [measurand units]
(b) inverse problem
$\hat{\boldsymbol{y}} = f^{-1}(\boldsymbol{x})$ [measurand units]
$\boldsymbol{x}$ [instrument units]
(c) forward problem
$\boldsymbol{x} = f(\boldsymbol{y})$ [instrument units]
$\boldsymbol{y}$ [measurand units]
calibration data
validation data
○ $\{\boldsymbol{x}, \boldsymbol{y}\}$
calib. points
system validation
□ $\{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}\}$
valid. points

Fig. 5.3 The mapping function is a linear transformation between the measured value $\hat{\boldsymbol{y}}$ and the true value $\boldsymbol{y}$. Depending on the application, the problem can be formulated either as a forward or inverse problem. However, system validation is required to ensure the validity of the models.

(2) Temporal measurements of a dynamic system are formulated as initial value problems (IVPs). The continuous forcing function $z(x)$ of an ODE corresponds to the measurement, whereby at least $p \geq n$ independent constraints are required to find a unique solution for an ODE of n^{th} degree. As explained in Section 10.2.1, the continuous measurement model is

$$\sum_{i=1}^{n} a_i(x)\, D^{(i)}\, y(x) = z(x). \tag{5.4}$$

(3) Multi-sensor spatial measurements are formulated as boundary value problems (BVPs). The discrete samples $\boldsymbol{z}$ of the ODE's forcing function $z(x)$ correspond to the measurement. As described Section 10.2.2, the cost function

$$K = \|\mathsf{L}\,\boldsymbol{y} - \boldsymbol{z}\|_2^2 \tag{5.5}$$

must be minimized to acquire the least squares solution, whereby L is a linear differential operator. A unique solution is ensured by placing independent constraints onto the system, which are written as

$$\mathsf{C}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{d} \quad \text{and} \quad \boldsymbol{y} = \mathsf{B}\,\boldsymbol{\alpha}, \tag{5.6}$$

whereby C^{T} is the matrix of constraints and $\boldsymbol{d}$ holds the values of these constraints. This is an analytic a-posteriori approach, which is preferable in comparison to a-priori approaches such as Runge-Kutta, which only provide an estimated solution. Acquiring a time series of multi-sensor spatial measurements yields a PDE. Such problems occur in gradient measurements from inertial measurement units, e.g., inclinometers or accelerometers.

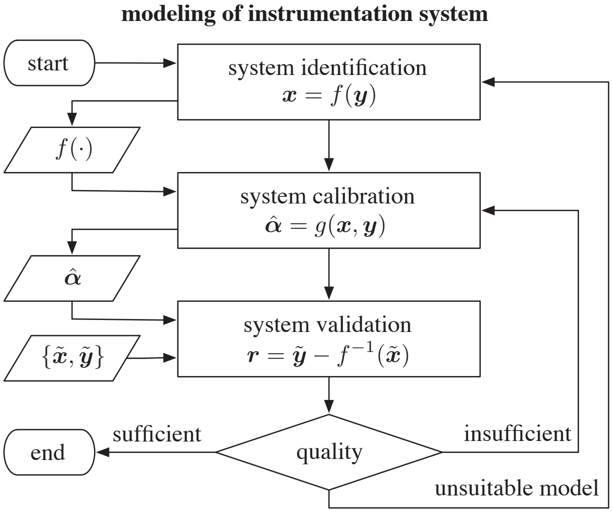

modeling of instrumentation system
start
system identification
$\boldsymbol{x} = f(\boldsymbol{y})$
$f(\cdot)$
system calibration
$\hat{\boldsymbol{\alpha}} = g(\boldsymbol{x}, \boldsymbol{y})$
$\hat{\boldsymbol{\alpha}}$
system validation
$\boldsymbol{r} = \tilde{\boldsymbol{y}} - f^{-1}(\tilde{\boldsymbol{x}})$
$\{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}\}$
end
sufficient
quality
insufficient
unsuitable model

Fig. 5.4 The system model $f(\cdot)$ is determined during system identification and the estimation of the coefficients $\hat{\boldsymbol{\alpha}}$ are found during calibration. The model is verified using cross validation, the cycle is repeated until a suitable setup is found.

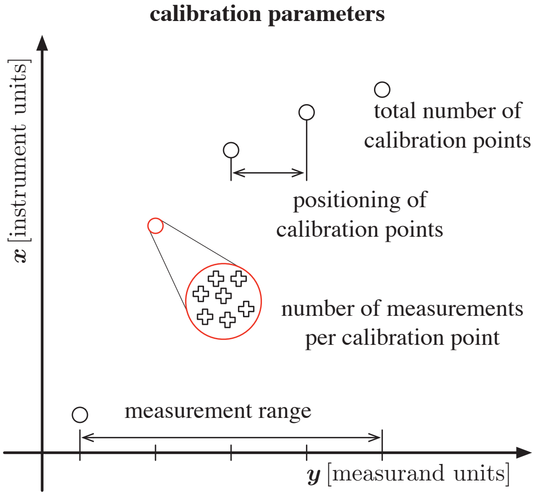

calibration parameters
$\boldsymbol{x}$ [instrument units]
total number of calibration points
positioning of calibration points
number of measurements per calibration point
measurement range
$\boldsymbol{y}$ [measurand units]

Fig. 5.5 The calibration points itself are victim to noise; therefore, their number and positioning within the measurement range has a significant effect on the quality of the calibration curve together with the chosen polynomial degree d.

The thesis focuses on the use of discrete orthogonal polynomial basis functions. Following measures influence the overall calibration's quality when using polynomials (Fig. 5.5):

1. The **polynomial degree** d has been either identified from the underlying physical phenomenon or estimated incrementally with the process in Fig. 5.4. Selecting an appropriate degree is an effective regularization technique in the form of a low pass filter;
2. The **measurement range** defines the input span expressed as the difference between the range limits. Polynomials tend to be incorrect outside of their support when extrapolated because of Runge's phenomenon;
3. The **number of calibration points** m determine the confidence of the fit. There are a minimum of $m = d + 1$ calibration points required for an exact fit, each further point increases the d.f. ν, i.e., $\nu = m - d - 1$. Most commonly, $m \gg (d + 1)$. Underestimated systems with $m \leq d$ and parameterized solutions are not covered in this thesis;
4. The **number of measurements** by calibration point influences the calibration point's confidence; measuring a calibration point multiple times reduces the associated standard error;
5. The **positioning of the calibration points** effectively improves the quality of the fit. While equidistant (regular) nodes are easier to produce in a calibration setup, irregular Chebyshev nodes can produce better results at the borders of the support.

5.2 System Validation

During data acquisition, the set of m data points $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathbb{R}^{(m\times 1)}$ is obtained for calibration purposes. In order to perform system validation, it is required to gather an additional independent set of n data points $\{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}\} \in \mathbb{R}^{(n\times 1)}$. The model's estimator for the coefficients $\hat{\boldsymbol{\alpha}}$ facilitates the formulation of the measurement function $f^{-1}(\cdot)$, see Chapter 8. The cross-validation procedure evaluates the functions at the positions of the known values $\tilde{\boldsymbol{x}}$, yielding the measured values $\hat{\boldsymbol{y}}$ (see Fig. 5.6),

$$\hat{\boldsymbol{y}} = l(\tilde{\boldsymbol{x}}), \tag{5.7}$$

whereby $l(\tilde{\boldsymbol{x}}) \triangleq l(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{\alpha}})$ actually takes the estimate for the coefficients $\hat{\boldsymbol{\alpha}}$ as a second parameter. The residual $\boldsymbol{r}$ is the difference between the measured points $\hat{\boldsymbol{y}}$ and the known points $\tilde{\boldsymbol{y}}$,

$$\boldsymbol{r} = \tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}} \quad \text{with} \quad \hat{\boldsymbol{y}} = f^{-1}(\tilde{\boldsymbol{x}}). \tag{5.8}$$

It is important, that the validation points $\{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}\}$ are not equal to the points originally used in the calibration $\{\boldsymbol{x}, \boldsymbol{y}\}$, because overfitting effects are undetectable this way (Fig. 5.7). The noise level δ,

$$\delta = \|\boldsymbol{r}\|_2, \tag{5.9}$$

is an estimate for the perturbation inside the data. If the noise level converges to 0, the measured value $\hat{\boldsymbol{y}}$ should become the true validation value $\tilde{\boldsymbol{y}}$ when using a suitable model,

$$\lim_{\delta\to 0} \hat{\boldsymbol{y}} = \tilde{\boldsymbol{y}}, \tag{5.10}$$

i.e., all random error effects are removed theoretically. Dividing δ by the number of cross validation samples n,

$$\tilde{\delta} = \frac{\|\boldsymbol{r}\|_2}{n}, \tag{5.11}$$

is an indicator $\tilde{\delta}$ of the model's quality w.r.t. the application, i.e., the standardized error of each sampled values. Fig. 5.4 visualizes the complete calibration process, which is repeated until a suitable model and/or parameters are found, i.e., the value of $\tilde{\delta}$ falls below a predefined threshold. Utilization of another measurement principle for the acquisition of $\{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}\}$ than the one used in the actual application may also independently verifies the correctness of the model.

Extensive cross validation is required when the analytic nature of the underlying process is unknown. Bootstrapping and/or jackknifing techniques are alternative approaches for system validation; unfortunately, generating validation points from artificially shuffled or perturbed calibration points delivers unreliable results in terms of system identification, because overfitting effects are hardly detectable, see Fig. 5.7. Furthermore, it is required to test the remaining residual vector $\boldsymbol{r}$ for normal distribution by applying a generic Kolmogorov-Smirnov test and/or a specific Anderson-Darling test [17]. Uncorrected systematic error effects are detectable using this statistical methodology.

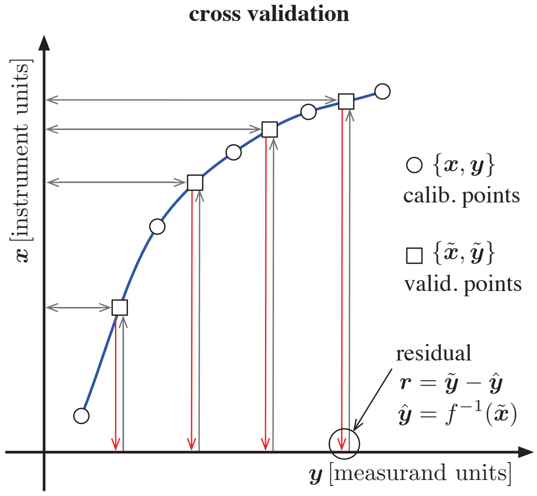

cross validation
$\boldsymbol{x}$ [instrument units]
○ $\{\boldsymbol{x}, \boldsymbol{y}\}$
calib. points
□ $\{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}\}$
valid. points
residual
$\boldsymbol{r} = \tilde{\boldsymbol{y}} - \hat{\boldsymbol{y}}$
$\hat{\boldsymbol{y}} = f^{-1}(\tilde{\boldsymbol{x}})$
$\boldsymbol{y}$ [measurand units]

Fig. 5.6 The calibration curve and its coefficients $\hat{\boldsymbol{\alpha}}$ respectively are approximated using the calibration points $\{\boldsymbol{x}, \boldsymbol{y}\}$. Known validation points $\{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}\}$ are measured and the residual vector $\boldsymbol{r}$ is determined in order to verify the system model.

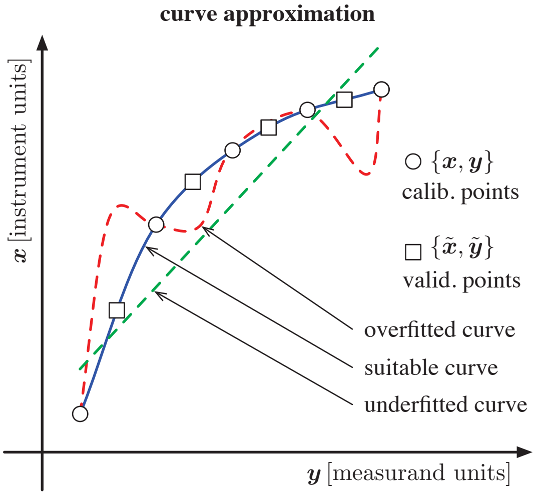

curve approximation
$\boldsymbol{x}$ [instrument units]
○ $\{\boldsymbol{x}, \boldsymbol{y}\}$
calib. points
□ $\{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}\}$
valid. points
overfitted curve
suitable curve
underfitted curve
$\boldsymbol{y}$ [measurand units]

Fig. 5.7 Contextual knowledge about the modeled physical process is required to achieve a suitable curve from the calibration points $\{\boldsymbol{x}, \boldsymbol{y}\}$. Known validation points $\{\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}\}$ enable the detection of under- and overfitting effects.

5.3 Uncertainty in Measurements

The methods of statistical uncertainty analysis described in this thesis are of general validity in metrology. The deviation from the true measurand's value is composed of random and systematic errors. These two kinds of errors are assumed to be distinguishable during system identification. As already argued, significant systematic errors can be widely eliminated with the employment of a suitable model, while random errors are intrinsic to the process of quantifying physical properties. Other sources of failures such as errors made by human operators, aging effects of the equipment, etc. are not dealt within this thesis.

Prior to further discussions, the terms of *accuracy* and *precision* are defined to avoid misconceptions in this context. According to [20, 106], accuracy is the closeness of agreement between a measured quantity value and a true quantity value of a measurand while precision is the closeness of agreement between indications (or measured quantity values) obtained by replicate measurements on the same or similar objects under specified conditions, see Fig. 5.8 for a visual comparison. In this vein, noise is a random variation of the value of the measured signal even under nominal control of the operating conditions, while the bias is a systematic deviation. Consequently, the goal when setting up a measurement is to remove the bias while simultaneously minimizing the noise. Unfortunately, the true quantity value is considered unknowable in practice, which justifies the need of thorough uncertainty analysis.

Most commonly, i.i.d. Gaussian noise is assumed for measurement applications, because:

1. The underlying **statistical population** of most measurement applications is Gaussian;

2. The **central limit theorem** states that, given certain conditions, the arithmetic mean of a sufficiently large number of iterations of independent random variables, i.e., measurements, each with a well-defined expected value and well-defined variance, will be approximately normally distributed, regardless of the underlying distribution [17, 210]. Care must be taken since this condition is not always fulfilled;

3. **Least squares approximation is a maximum likelihood predictor** if the noise is Gaussian. A least squares approximation will have a systematic bias if the above condition is not fulfilled. Least squares regression can be formulated efficiently using linear algebra;

4. The **mathematical formulation of uncertainty** in measurement applications is simplified when assuming a Gaussian distribution. The calculation of confidence and prediction intervals is based on the assumption of normally distributed noise. As shown in Section 5.5, there is a sound formulation available for deriving the associated Student-t, Fisher-$\mathcal{F}$ and χ^2 distributions using only a limited number of samples.

Fig. 5.9 shows the Gaussian probability density function (PDF) significance level α for the standard deviations $\{\sigma, 2\sigma, 3\sigma\}$ and Fig. 5.10 shows the Gaussian cumulative distribution function (CDF) for significance level $\alpha = \{90\%, 95\%, 99\%\}$. These levels of significance are common in most engineering applications.

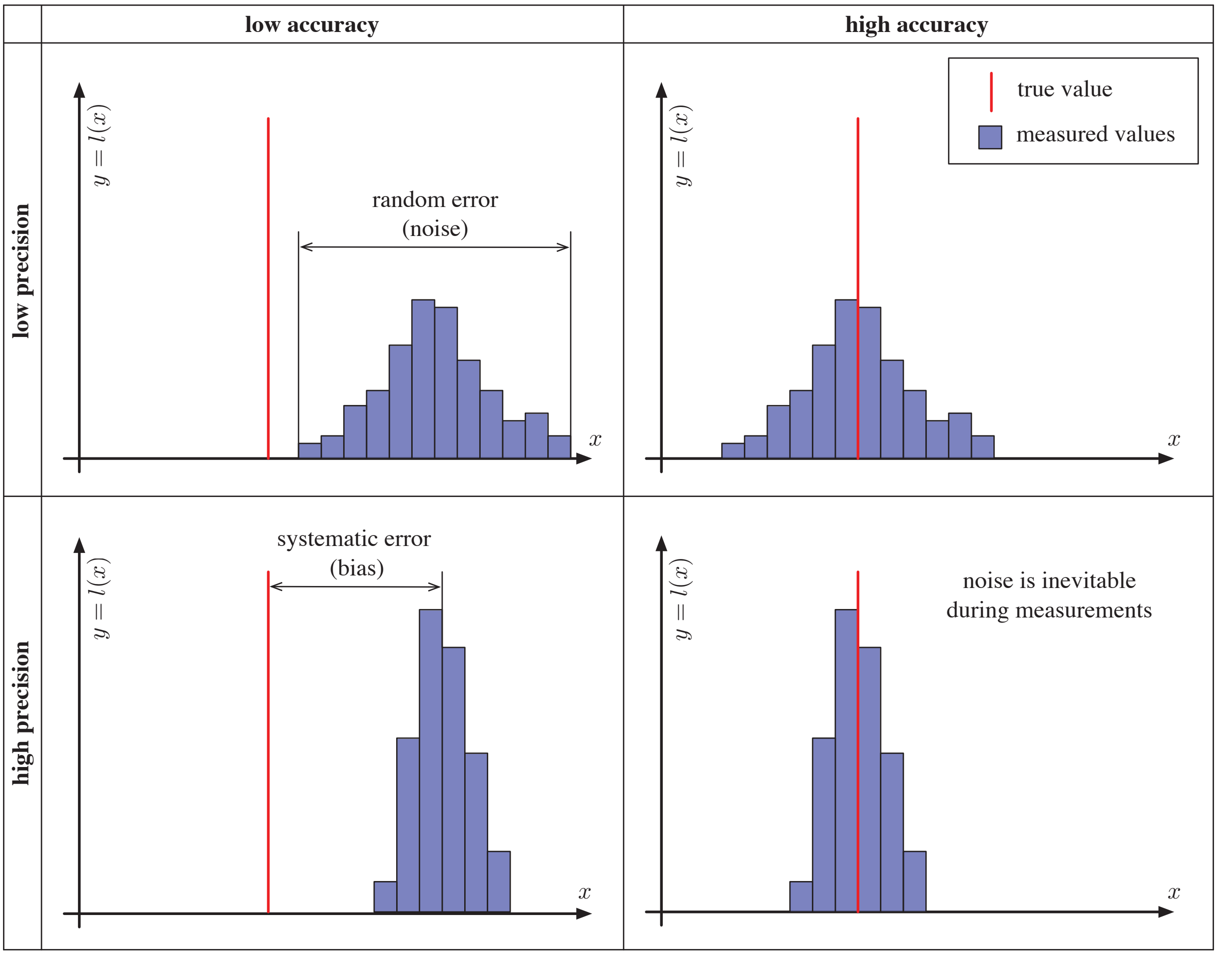


Fig. 5.8 The random scatter of measurements w.r.t. the samples' mean is called the precision error; it is caused by noise, determined by repeated measurements under stable conditions and described statistically. Any deviation of the samples' mean from the true value must be considered to be systematic; the bias is an estimator for this systematic error. Systematic influences on a measurement should be avoided in principle and are corrected by a suitable system model in order to achieve high accuracy.

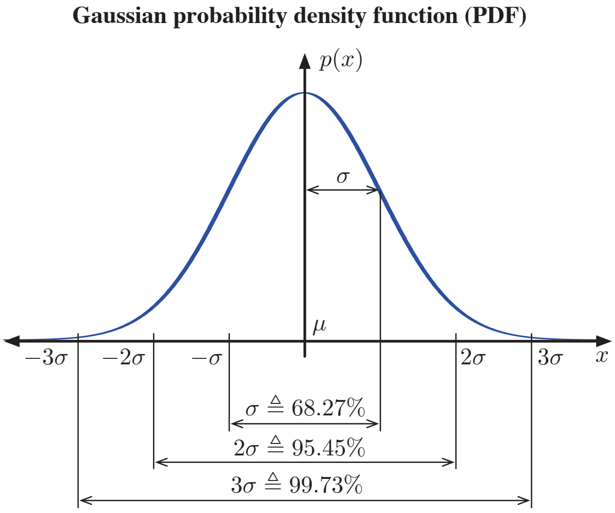


Fig. 5.9 Gaussian PDF

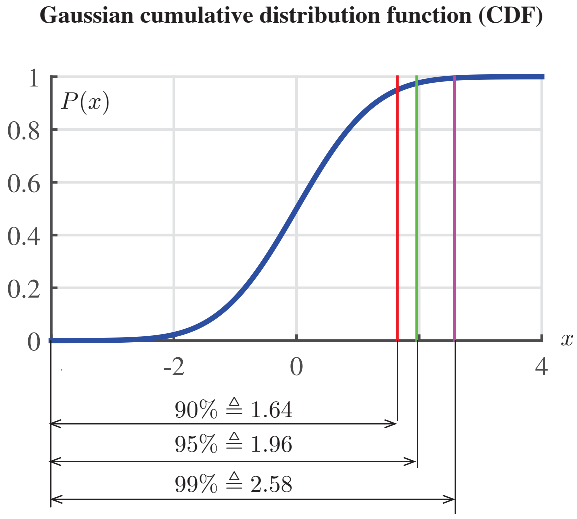


Fig. 5.10 Gaussian CDF

Guide to Uncertainty in Measurement

The *guide to the expression of uncertainty in measurement* (GUM) [105] is published by the JCGM and establishes general rules for evaluating and describing uncertainty in measurements. The reference guide is in agreement with the German legacy standard DIN1319 - fundamentals of metrology [37–40] as well as its European successor ENV13005 [46]. The ENV document is still a pre-standard, but covers all ideas mentioned in the GUM.

Besides the need of expressing the uncertainty for engineering applications [30, 192], there is also a legislative need of doing so. According to the European Union Council directives 2006/42/EC on machinery [51] and 2004/22/EC on measurement instruments [49], the uncertainty of measurements must be stated when determining critical threshold values for safety reasons and/or when the measurement device is classified by its accuracy. The error of measurement shall not exceed the *maximum permissible error* (MPE) value under rated operating conditions and in the absence of disturbances. Furthermore, measurement instruments must comply with the directive 2004/108/EC on electromagnetic compatibility (EMC) to avoid interference errors caused by electromagnetic disturbances.

The quality of the measurement's result is expressed as an estimate of the measurand along with an associated measurement uncertainty. The uncertainty reflects the incomplete knowledge about the measurand. The GUM differentiates two types of error analysis: type A, which is a stochastic approach, and type B, which is an empiric approach. The type A approach is summarized in this section.

(1) Univariate Case

The statistical method is based on evaluating data of repeated i.i.d. measurements of one and the same measurand under constant environmental conditions with Gaussian $\mathcal{N}(x, \mu_x, \sigma_x)$ noise. This approach also facilitates the implementation of Monte Carlo simulation [30] for verification purposes. The simple explicit model is

$$y = l(x) = x \tag{5.12}$$

with the measurand x and the uncorrected result y, whereby both x and y are random univariate variables. It is assumed, that the systematic error has been compensated. A number of m samples are drawn from x, yielding the vector $\boldsymbol{x} = [x_1 \dots x_m]^{\mathrm{T}}$. The sample mean $\bar{x}$ of $\boldsymbol{x}$ is an estimator of the first moment μ_x,

$$\bar{x} = \frac{1}{m} \sum_{i=1}^{m} x_i, \tag{5.13}$$

and the sample variance $s_{\boldsymbol{x}}^2$ is an estimator for the variance (second moment) σ_x^2,

$$s_{\boldsymbol{x}}^2 = \frac{1}{m-1} \sum_{i=1}^{m} (x_i - \bar{x})^2 \quad \text{with} \quad s_{\boldsymbol{x}} = +\sqrt{s_{\boldsymbol{x}}^2}, \tag{5.14}$$

whereby the sample standard deviation is $s_{\boldsymbol{x}}$. The uncertainty associated to the random error is equivalent to the standard error of the mean $s_{\bar{x}}$, i.e.,

$$s_{\bar{x}} = \frac{s_{\boldsymbol{x}}}{\sqrt{m}}. \tag{5.15}$$

The result y with a symmetric $(1-\alpha)$ confidence interval is written as

$$y = \bar{x} \pm s_{\bar{x}}\, t_{(\nu)}, \tag{5.16}$$

where $t_{(\nu)}$ is the value of the Student-t inverse cumulative distribution function with $\nu = m-1$ d.f. and significance level $\alpha = 95\%$. This methodology only covers the simplest case for uncertainty analysis with Gaussian noise without bias, a detailed description is found in [105].

(2) Multivariate Case

An estimation for the uncertainty in the multivariate case with $\mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu_x}, \mathsf{\Lambda}_{\boldsymbol{x}})$ Gaussian noise is given using a first-order Taylor series expansion of the function

$$\boldsymbol{y} = l(\boldsymbol{x}), \tag{5.17}$$

whose vector of input data $\boldsymbol{x}$ is represented as a multivariate random variable with mean $\boldsymbol{\mu_x}$ and covariance matrix $\mathsf{\Lambda}_{\boldsymbol{x}}$. The derivation is described with $\mathsf{J}_{l(\boldsymbol{\mu_x})}$, which denotes the Jacobian matrix of the measurement function $l(\cdot)$ w.r.t. the input data $\boldsymbol{x}$ and evaluated at the expectation value $\boldsymbol{\mu_x}$. For easier reading, it is defined that

$$\mathsf{J} \triangleq \mathsf{J}_{l(\boldsymbol{\mu_x})}. \tag{5.18}$$

This approach has been analytically described in [30] and applied in [53, 188]. This section is presented in concise matrix form, although the GUM does not use this efficient notation. Given the explicit function $\boldsymbol{y} = l(\boldsymbol{x})$, the Taylor series expansion is

$$\boldsymbol{y} = l(\boldsymbol{\mu_x}) + \mathsf{J}(\boldsymbol{x} - \boldsymbol{\mu_x}) + \mathcal{O}(\|\boldsymbol{x} - \boldsymbol{\mu_x}\|^2) \tag{5.19}$$

and, according to [30, 53], the first-order approximation for the covariance $\mathsf{\Lambda}_{\boldsymbol{y}}$ is

$$\mathsf{\Lambda}_{\boldsymbol{y}} \approx \mathsf{J}\mathsf{\Lambda}_{\boldsymbol{x}}\mathsf{J}^{\mathrm{T}}. \tag{5.20}$$

The explicit knowledge of the functional relationship $l(\cdot)$ between $\boldsymbol{x}$ and $\boldsymbol{y}$ is not required for estimating the covariance $\mathsf{\Lambda}_{\boldsymbol{y}}$ of the output data $\boldsymbol{y}$, as it depends only on the covariance $\mathsf{\Lambda}_{\boldsymbol{x}}$ of the input data $\boldsymbol{x}$ and the Jacobian matrix J, which is in fact a linear operator yielding the first order polynomial approximation.

5.4 Covariance Matrix

The computation of the covariance matrix allows to determine the uncertainty's mapping from the input to the output from any linear operator. Consider the perturbed input vector $\boldsymbol{x} = [x_1 \ldots x_m]^{\mathrm{T}}$ with m measured values, whereby x_i is the i^{th} sample of $\boldsymbol{x}$. The output vector $\boldsymbol{y}$ is computed via

$$\boldsymbol{y} = \mathsf{L}\,\boldsymbol{x}, \tag{5.21}$$

where L is an arbitrary linear operator. The expected value $\mu_{\boldsymbol{x}}$ is computed with the linear expectation operator $\mathbb{E}[\cdot]$, i.e., $\mathbb{E}[\boldsymbol{x}] = \mu_{\boldsymbol{x}}$, thus,

$$\mathbb{E}[\boldsymbol{x}] = [\mathbb{E}[x_1] \ldots \mathbb{E}[x_i] \ldots \mathbb{E}[x_m]]^{\mathrm{T}}. \tag{5.22}$$

The covariance matrix $\mathsf{\Lambda}_{\boldsymbol{x}}$ of the vector $\boldsymbol{x}$ is defined [17] as

$$\begin{aligned} \mathsf{\Lambda}_{\boldsymbol{x}} &= \mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])\,(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])^{\mathrm{T}}\right], & (5.23)\\ &= \mathbb{E}\left[(\boldsymbol{x} - \mu_{\boldsymbol{x}})\,(\boldsymbol{x} - \mu_{\boldsymbol{x}})^{\mathrm{T}}\right]. & (5.24) \end{aligned}$$

The covariance matrix $\mathsf{\Lambda}_{\boldsymbol{y}}$ of the vector $\boldsymbol{y}$ is then, using Eqn. (5.23),

$$\begin{aligned} \mathsf{\Lambda}_{\boldsymbol{y}} &= \mathbb{E}\left[(\boldsymbol{y} - \mathbb{E}[\boldsymbol{y}])\,(\boldsymbol{y} - \mathbb{E}[\boldsymbol{y}])^{\mathrm{T}}\right],\\ &= \mathbb{E}\left[(\mathsf{L}\boldsymbol{x} - \mathbb{E}[\mathsf{L}\boldsymbol{x}])\,(\mathsf{L}\boldsymbol{x} - \mathbb{E}[\mathsf{L}\boldsymbol{x}])^{\mathrm{T}}\right],\\ &= \mathbb{E}\left[(\mathsf{L}\boldsymbol{x} - \mathsf{L}\mathbb{E}[\boldsymbol{x}])\,(\mathsf{L}\boldsymbol{x} - \mathsf{L}\mathbb{E}[\boldsymbol{x}])^{\mathrm{T}}\right],\\ &= \mathsf{L}\,\mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])\,(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])^{\mathrm{T}}\right]\mathsf{L}^{\mathrm{T}},\\ &= \mathsf{L}\,\mathbb{E}\left[(\boldsymbol{x} - \mu_{\boldsymbol{x}})\,(\boldsymbol{x} - \mu_{\boldsymbol{x}})^{\mathrm{T}}\right]\mathsf{L}^{\mathrm{T}},\\ &= \mathsf{L}\mathsf{\Lambda}_{\boldsymbol{x}}\mathsf{L}^{\mathrm{T}}. \end{aligned} \tag{5.25}$$

The equation

$$\mathsf{\Lambda}_{\boldsymbol{y}} = \mathsf{L}\mathsf{\Lambda}_{\boldsymbol{x}}\mathsf{L}^{\mathrm{T}} \quad \text{with} \quad \boldsymbol{y} = \mathsf{L}\,\boldsymbol{x} \tag{5.26}$$

is valid for any linear operation L; as a result, the input's uncertainty can directly be mapped to the output. The uncertainty can be estimated a-priori to the measurement during calibration. If the vector $\boldsymbol{x}$ was acquired from a homoscedastic measurement of i.i.d. samples, then the variance $\sigma_{\boldsymbol{x}}^2$ and standard deviation $\sigma_{\boldsymbol{x}}$ is equal for all values in $\boldsymbol{x}$. This leads to

$$\mathsf{\Lambda}_{\boldsymbol{x}} = \sigma_{\boldsymbol{x}}^2\mathsf{I}, \tag{5.27}$$

when I is the identity matrix. In this simplified case, the output covariance is then

$$\begin{aligned} \mathsf{\Lambda}_{\boldsymbol{y}} &= \mathsf{L}\mathsf{\Lambda}_{\boldsymbol{x}}\mathsf{L}^{\mathrm{T}},\\ &= \mathsf{L}\left(\sigma_{\boldsymbol{x}}^2\mathsf{I}\right)\mathsf{L}^{\mathrm{T}},\\ &= \sigma_{\boldsymbol{x}}^2\,\mathsf{L}\mathsf{L}^{\mathrm{T}}, \end{aligned} \tag{5.28}$$

where $\mathsf{L}\mathsf{L}^{\mathrm{T}}$ is a projection onto the column vectors of the linear operator L.

5.5 Goodness-of-Fit Statistics

Using χ^2 goodness-of-fit statistics is a formal approach do determine the confidence and prediction intervals of a regression and, therefore, uncertainty bounds. The goodness-of-fit test is the inverse approach, i.e., it enables to test a regression against a given significance level. A general linear operation with known input uncertainty is written as

$$\hat{\boldsymbol{y}} = \mathsf{L}\,\hat{\boldsymbol{x}} \quad \text{with} \quad \Lambda_{\hat{\boldsymbol{y}}} = \mathsf{L}\Lambda_{\hat{\boldsymbol{x}}}\mathsf{L}^{\mathrm{T}}, \tag{5.29}$$

where $\hat{\boldsymbol{y}}$ is the estimator of the true output variable $\boldsymbol{y}$, $\hat{\boldsymbol{x}}$ is the estimator for the true input variable $\boldsymbol{x}$ and L is a linear operator. According to [210], the χ^2 test with r d.f. is formulated as

$$\chi^2_{(r)} \leq (\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}])^{\mathrm{T}} {\Lambda_{\hat{\boldsymbol{y}}}}^{-1} (\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}]), \tag{5.30}$$

i.e., the scalar product $(\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}])^{\mathrm{T}}(\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}])$ is standardized with the covariance $\Lambda_{\hat{\boldsymbol{y}}}$,

$$\begin{aligned} \chi^2_{(r)} &\leq (\mathsf{L}\hat{\boldsymbol{x}} - \mathbb{E}[\mathsf{L}\hat{\boldsymbol{x}}])^{\mathrm{T}} \big(\mathsf{L}\Lambda_{\hat{\boldsymbol{x}}}\mathsf{L}^{\mathrm{T}}\big)^{-1} (\mathsf{L}\hat{\boldsymbol{x}} - \mathbb{E}[\mathsf{L}\hat{\boldsymbol{x}}]), \\ &\leq (\mathsf{L}\hat{\boldsymbol{x}} - \mathsf{L}\boldsymbol{x})^{\mathrm{T}} \big(\mathsf{L}\Lambda_{\hat{\boldsymbol{x}}}\mathsf{L}^{\mathrm{T}}\big)^{-1} (\mathsf{L}\hat{\boldsymbol{x}} - \mathsf{L}\boldsymbol{x}), \\ &\leq (\mathsf{L}\hat{\boldsymbol{x}} - \mathsf{L}\boldsymbol{x})^{\mathrm{T}} \big(\sigma^2_{\hat{\boldsymbol{x}}}\mathsf{L}\mathsf{L}^{\mathrm{T}}\big)^{-1} (\mathsf{L}\hat{\boldsymbol{x}} - \mathsf{L}\boldsymbol{x}), \\ &\leq \frac{1}{\sigma^2_{\hat{\boldsymbol{x}}}} (\mathsf{L}\hat{\boldsymbol{x}} - \mathsf{L}\boldsymbol{x})^{\mathrm{T}} \big(\mathsf{L}\mathsf{L}^{\mathrm{T}}\big)^{-1} (\mathsf{L}\hat{\boldsymbol{x}} - \mathsf{L}\boldsymbol{x}). \end{aligned} \tag{5.31}$$

When estimating the population's variance $\sigma^2_{\hat{\boldsymbol{x}}}$ with the sample variance $s^2_{\hat{\boldsymbol{x}}}$, then the $\chi^2_{(r)}$ distribution can be approximated with a Fisher distribution $r\,\mathcal{F}_{(r,\nu)}$,

$$\lim_{\nu\to\infty} r\,\mathcal{F}_{(r,\nu)} = \chi^2_{(r)}. \tag{5.32}$$

As a consequence, this leads to the approximation

$$\mathcal{F}_{(r,\nu)} \leq \frac{1}{r\,s^2_{\hat{\boldsymbol{x}}}} (\mathsf{L}\hat{\boldsymbol{x}} - \mathsf{L}\boldsymbol{x})^{\mathrm{T}} \big(\mathsf{L}\mathsf{L}^{\mathrm{T}}\big)^{-1} (\mathsf{L}\hat{\boldsymbol{x}} - \mathsf{L}\boldsymbol{x}). \tag{5.33}$$

If the number or regressors $r = 1$, then the Fisher-$\mathcal{F}$ distribution is equivalent to a squared Student-t distribution,

$$\mathcal{F}_{(1,\nu)} = t^2_{(\nu)}, \tag{5.34}$$

and, consequently,

$$\begin{aligned} t^2_{(\nu)} &\leq \frac{1}{s^2_{\hat{\boldsymbol{x}}}} (\mathsf{L}\hat{\boldsymbol{x}} - \mathsf{L}\boldsymbol{x})^{\mathrm{T}} \big(\mathsf{L}\mathsf{L}^{\mathrm{T}}\big)^{-1} (\mathsf{L}\hat{\boldsymbol{x}} - \mathsf{L}\boldsymbol{x}), \\ &\leq \frac{1}{s^2_{\hat{\boldsymbol{x}}}} (\hat{\boldsymbol{x}} - \boldsymbol{x})^{\mathrm{T}} \mathsf{L}^{\mathrm{T}} \big(\mathsf{L}\mathsf{L}^{\mathrm{T}}\big)^{-1} \mathsf{L} (\hat{\boldsymbol{x}} - \boldsymbol{x}). \end{aligned} \tag{5.35}$$

If L is not rank deficient, i.e., non-singular and invertible, the equation simplifies to

$$t^2_{(\nu)} \leq \frac{1}{s^2_{\hat{\boldsymbol{x}}}} (\hat{\boldsymbol{x}} - \boldsymbol{x})^{\mathrm{T}} (\hat{\boldsymbol{x}} - \boldsymbol{x}), \tag{5.36}$$

i.e., the goodness-of-fit is operation-independent. A Student-t distribution converges to a Gaussian distribution Φ with increasing d.f.,

$$\lim_{\nu \to \infty} t_{(\nu)} = \Phi. \tag{5.37}$$

The χ^2 goodness-of-fit test is a special case of the Mahalanobis distance δ,

$$\delta = \sqrt{(\boldsymbol{x} - \mu_{\boldsymbol{x}})^{\mathrm{T}} \Lambda_{\boldsymbol{x}}{}^{-1} (\boldsymbol{x} - \mu_{\boldsymbol{x}})}. \tag{5.38}$$

The knowledge of the χ^2 goodness-of-fit test is applied in Chapter 8 for determining the prediction and confidence intervals for the regression.

5.6 Multi-Source Data Fusion

According to [85, 193], *multi-source data fusion* (MSDF) is the process of combining observations from a number of different sources (sensors) to provide a robust and complete description of an environment or physical process of interest. The sensors can be of different type; however, the definition also encompasses multiple observations of a single sensor over time, which are then combined. This is of special importance in a CPS environment where the motes of a WSAN often cover overlapping sections within the physical domain. Meaningful combination of the measured data is the main goal of MSDF.

The terms *data fusion* and *information fusion* are not clearly differentiated in literature, e.g., sometimes they refer to merging data from data bases or fusing sensor channels during data mining. In this thesis, MSDF is the combination of sensory data such that the resulting information is in some sense better than it would be possible when these sources were used individually. This includes low-level raw sensor data and intermediate-level preprocessed sensor data in a single measurement device [76] as well as high-level MSDF used in WSAN applications. In this vein [146], possible motivations for MSDF are:

1. **Limited domain coverage.** Multiple diverse sensors may be required to acquire all properties of a physical phenomenon. The system's robustness against interference is increased;

2. **Limited spatial coverage.** An individual sensor only covers a restricted region and the measurement range needs to be expanded;

3. **Limited temporal coverage.** The sample rate of a single sensor may be too slow and a higher resolution is required;

4. **Imprecision** and **inaccuracy**. The uncertainty employed by a single sensor may be too high and redundant data is required, leading to an increased confidence level;

5. **Robustness** and **reliability**. Employing redundant sensor enables the system to provide information even in case of partial failure.

Considering all these points, synergetic effects may increase the overall quality of the measurement result. The sensor configuration can be complementary (independent sensory data), competitive (using the best sensor data) or cooperative (fusion of the sensor data) [43, 146]. These concepts also work in a WSAN environment.

State-of-the-art MSDF methods employ a probabilistic approach based on the Bayes theorem [5, 18, 113], i.e., conditional probability for states and their observations. The most popular implementations of the Bayes filtering techniques are probability grids (look-up-tables for the conditional probability), Kalman filters (maximum likelihood predictor/update model) and sequential Monte Carlo techniques (simulation using weighted samples) [193]. Probabilistic methods are non-deterministic and always require a large population of samples to work properly. Established alternative approaches are interval calculus (using error bounds), fuzzy logic (with membership functions) and Shafer-Dampster methods (theory of evidence).

There are several architectural frameworks for MSDF available [84]. The most popular one is the joint directors of the laboratories (JDL) model, which was originally intended for military applications [85, 193]. This approach views MSDF in terms of source, object, situation, threat and process analysis levels with human-machine interaction. Other models are the waterfall fusion process model and the Boyd model [146].

Several problems and misconceptions are related to MSDF. Employment of multiple sensors is never a substitute for using sensors with a suitable quality, i.e., errors emerging in the sensor data cannot be removed and the incorporated uncertainty can only be limited to a lower bound. Moreover, wrong application of MSDF can yield incorrect results, e.g., when contradictory or inconsistent data is fused.

The algebraic framework presented in the next chapters is capable of analytically fuse sensor data based on uncertainty and weighted regression, see Fig 5.11. This is advantageous in certain applications when being compared to probabilistic methods, because the data of each measurement can be fused immediately without the need of acquiring multiple samples. This methodology has been applied in [75, 76] for fusing the views of multiple cameras in order to extend the measurement range; the associated project is described in Chapter 13.

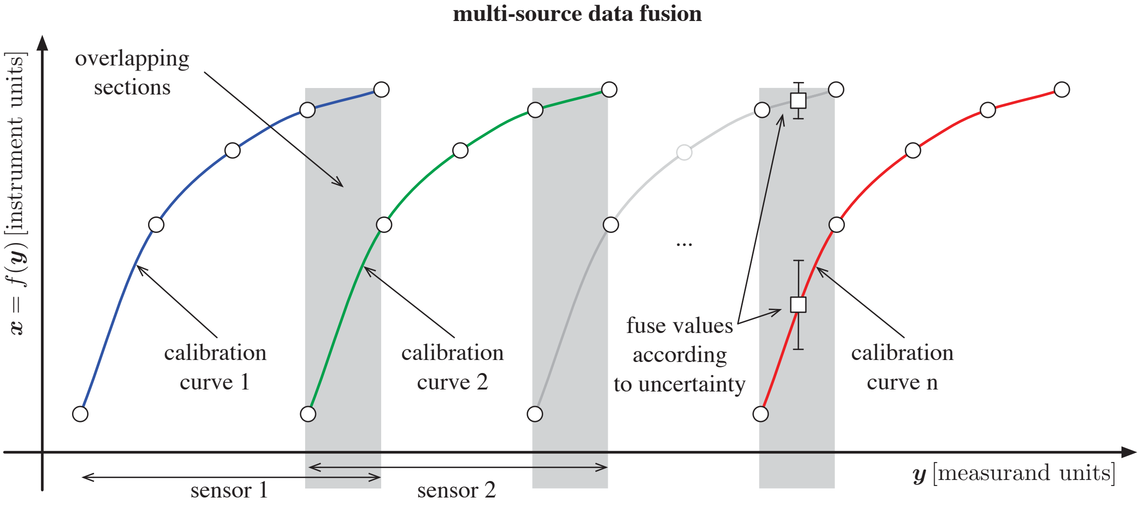


Fig. 5.11 Each sensor covers a limited spatial range, whereby the calibration curves overlap at their boundaries. The uncertainty, i.e., the confidence intervals, incorporated by each curve is analytically determined during calibration. The curves are fused w.r.t. the inverse of their associated uncertainty, i.e., the weighting corresponds to their confidence.

6 | Basis Functions

The theory of basis functions has been applied in numerous practical applications. Harker [87] gives a general overview of basis functions and their synthesis. Suesut [199] proposed a new concept for combining periodic with polynomial moments for modeling and analyzing 3D surface data. He implemented a system for the real-time inspection of metal surfaces based on light-sectioning. Tensor polynomial approximation on an invariant 2D Cartesian grid is applied to generate a smooth model for the surface. Gugg [69] compared different geometric and periodic bases for the modeling of surfaces and proposed spectral analysis for pattern recognition based on machine learning. Neumayr [157] and Gugg [70] implemented a transformation algorithm for mapping points between a distorted camera space and the associated metric real space using tensor interpolation based on discrete orthogonal polynomials. Badshah [9] improved the method of image registration for non-rigid distortion using hierarchical image processing with multiresolution pyramids. He compared the quality of different basis functions on artificial data and their behavior w.r.t. Gibbs error. An overview of moments has been given in [9, 163, 199]; according to these sources, moments are classified in three groups [168], which are shortly discussed in this section.

1. **Geometric moments.** It has been proven that there is one and only one complete unitary discrete polynomial basis [163]. Existing implementations are variations of one and the same solution. The numerically unstable Vandermonde basis is a popular choice in engineering applications despite its poor characteristics;

2. **Polar/radial moments.** These rotationally invariant moments consist of an orthogonal polynomial basis combined with a polar rotation. The Zernike moments and the orthogonal Fourier-Mellin moments belong to this group [199];

3. **Periodic/cyclic moments.** These moments are time shift invariant and well suited for analysis of periodic data, e.g., the Fourier basis and the discrete cosine basis.

The suitability of a basis depends on the nature of the data being analyzed and on the modeled features, i.e., the selection of the basis is application dependent. Using a suitable basis minimizes the amount of information redundancy and effectively eliminates some of the problems associated with Gibbs error and Runge's phenomenon.

6.1 Nomenclature

Preliminary definitions must be made before establishing the algebraic framework. The term *tensor* describes a multidimensional mathematical object in linear algebra, whereby a zero-order tensor is a *scalar*, a first-order tensor is a *vector* and a second order tensor is a *matrix*.

The term *discrete orthogonal polynomials* is used ambiguously in literature. Basically, *orthogonality* describes the idea of a mathematical relation thought of as describing non-overlapping, uncorrelated or independent objects of some kind. In linear algebra, a set $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathbb{R}^{(m\times 1)}$ of two vectors is orthogonal if and only if the inner/scalar product is zero, i.e., $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{x} = 0$ while $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = \xi$ and $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} = \psi$, where ξ and ψ are scalars.

So far, this does not refer to the magnitude of the vectors. In particular, a set $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathbb{R}^{(m\times 1)}$ is called *orthonormal* if it is an orthogonal set of unit vectors, i.e., $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = 1$ and $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} = 1$. If this set $\{\boldsymbol{x}, \boldsymbol{y}\} \in \mathbb{C}^{(m\times 1)}$ is complex, it is called a *unitary* set. The concept can be expanded to matrices, whereby the rows and columns of the matrix B fulfill the orthogonality condition. This leads to the nomenclature,

1. **orthogonal** basis $\mathsf{B}^{\mathrm{T}}\mathsf{B} = \mathsf{C}$, where $\mathsf{B} \in \mathbb{R}^{(m\times n)}$ and $\mathsf{C} \in \mathbb{R}^{(n\times n)}$ is diagonal;

2. **orthonormal** basis $\mathsf{B}^{\mathrm{T}}\mathsf{B} = \mathsf{I}$, where $\mathsf{B} \in \mathbb{R}^{(m\times n)}$ and I is the identity matrix;

3. **unitary** basis $\mathsf{B}^{\mathrm{T}}\mathsf{B} = \mathsf{I}$, where $\mathsf{B} \in \mathbb{C}^{(m\times n)}$ is complex;

4. **weighted orthogonal** basis $\mathsf{B}^{\mathrm{T}}\mathsf{W}\mathsf{B} = \mathsf{C}$, where $\mathsf{B} \in \mathbb{R}^{(m\times n)}$, $\mathsf{W} \in \mathbb{R}^{(m\times m)}$ is a weighting matrix and $\mathsf{C} \in \mathbb{R}^{(n\times n)}$ is diagonal.

Unfortunately, this definitions are often ignored in literature and the terms are used as synonyms for each other. The majority of methods presented in this thesis is meant to be applied on real measurement data within the domain of real valued numbers $\mathbb{R}$, although the methods are general enough to be applied in the complex plane $\mathbb{C}$. This is convenient when converting or combing geometric and periodic moments by synthesizing basis functions on the complex plane. The geometric discrete unitary polynomial basis is referred to as DOP (discrete orthogonal polynomial) basis within this thesis.

Many notations in linear algebra originate from MATLAB syntax. The i^{th} row or j^{th} column of a matrix A is selected via the element vector $\boldsymbol{e}_k = [0, \ldots, 1, \ldots, 0]^{\mathrm{T}}$, i.e., a vector of zeros except for a 1 on the k^{th} index. The operations are equivalent to the MATLAB syntax:

$$\mathsf{A}(i, :) \rightarrow \boldsymbol{x}_i^{\mathrm{T}} = \boldsymbol{e}_i^{\mathrm{T}}\mathsf{X}, \tag{6.1}$$

$$\mathsf{A}(:, j) \rightarrow \boldsymbol{y}_j = \mathsf{X}\,\boldsymbol{e}_j. \tag{6.2}$$

Selecting the specific elements $\boldsymbol{u} \in \mathbb{R}^{(k\times 1)}$ from a vector $\boldsymbol{x} \in \mathbb{R}^{(m\times 1)}$ with $k < m$ is denoted by

$$\boldsymbol{u} = \boldsymbol{x}(1 : k), \tag{6.3}$$

if $k = m$, then the operation is $\boldsymbol{u} = \boldsymbol{x}(1 : \mathrm{end}) = \boldsymbol{x}$. Selecting the diagonal elements of a matrix $\mathsf{A} \in \mathbb{R}^{(m\times m)}$ is denoted by

$$\mathrm{diag}\,(\mathsf{A}) = [a_{11}, a_{22}, \ldots, a_{ii}, \ldots, a_{mm}]^{\mathrm{T}}, \tag{6.4}$$

where $\mathrm{diag}\,(\mathsf{I}) = \boldsymbol{1}$, when I is the identity matrix and $\boldsymbol{1}$ is a vector of ones.

6.2 Overview

There is one and only one complete unitary discrete polynomial basis [163], which is abbreviated as G in the thesis. The exact Legendre, the Forsythe, the Chebyshev and the normalized Krawtchouk polynomials are fundamentally equivalent; the actual differences in performance are related to their specific numerical implementation. It was shown, that they only distinguish in the norms of the individual polynomials; the polynomials themselves are identical. Nevertheless, a short overview is given in order to enhance the conceptual understanding for basis functions. The symbols $\boldsymbol{p} = p(\boldsymbol{x})$ describe the polynomial basis functions of n^{th} degree synthesized on m points of $\boldsymbol{x} \in \mathbb{R}^{(m\times 1)}$ in the discrete and continuous domain respectively.

6.2.1 Geometric/Polynomial Moments

According to [163], the true unitary basis G consists of sums of monomials $\boldsymbol{g}$. Traditionally, polynomial bases are synthesized using a *three-term relationship*; the associated error propagation is the dominating limiting factor for the achievable accuracy. As a result, the degree of the polynomial to be generated is restricted as well as the size of the data to be analyzed. Improved algorithms were introduced in [151, 152]; however, they were limited to regular grids and introduced errors to low degree polynomials. The algorithm is evaluated in [162] using newly derived objective quality measures for bases.

In this thesis, the synthesis of a true unitary discrete basis using a generalized recurrence relationship is presented in Chapter 7. Near-perfect numerical behavior is achieved for the generation of polynomials computed from arbitrary nodes within the unit circle of the complex plane. With this approach, polynomial bases of almost arbitrarily degree can be implemented without noteworthy numerical error. The availability of a unitary basis also enables the development of understanding w.r.t. spectral propagation of Gaussian noise and noise bandwidth of polynomial filters. The numerically unstable Vandermonde (Taylor) polynomial basis V is still state-of-the-art for many engineering problems due to its clear and easy understandable structure. The Vandermonde basis is explained in Section 6.3 and is used for comparison/verification purposes throughout in this thesis.

(1) Legendre. The continuous Legendre polynomials are orthogonal over the range $x \in [-1, +1]$ in an integral sense, such that

$$\int_{-1}^{+1} P_p(x)\, P_q(x)\, \mathrm{d}x = \begin{cases} 0 & \text{if } p \neq q, \\ \frac{2}{2p+1} & \text{if } p = q. \end{cases} \tag{6.5}$$

The polynomials have a uniform weighting. Legendre polynomials do not form an orthogonal basis when computed at discrete points, which is a great cause of error propagation when implemented for higher degrees. The associated recursive relationship is

$$n\, \boldsymbol{p}_n(\boldsymbol{x}) - (2n-1)\, \boldsymbol{p}_1(\boldsymbol{x}) \circ \boldsymbol{p}_{n-1}(\boldsymbol{x}) + (n-1)\, \boldsymbol{p}_{n-2}\,(\boldsymbol{x}) = \boldsymbol{0} \tag{6.6}$$

for $n \geq 2$ with $\boldsymbol{p}_0 = \boldsymbol{1}$ and $\boldsymbol{p}_1 = \boldsymbol{x}$. The basis degenerates with increasing degree due to numerical errors caused by the three-term relationship. Nevertheless, the Legendre polynomials have become popular in machine vision applications because of their noise sensitivity and the image reconstruction properties.

(2) Forsythe. The Forsythe basis is defined in the range $x \in [0, m-1]$ and is generated by the recurrence relationship

$$\boldsymbol{p}_n(\boldsymbol{x}) = 2\,(\boldsymbol{x} - \alpha_n) \circ \boldsymbol{p}_{n-1}(\boldsymbol{x}) - \beta_n\, \boldsymbol{p}_{n-2}(\boldsymbol{x}), \tag{6.7}$$

where

$$\alpha_n = \frac{\|x \circ \boldsymbol{p}_{n-1}\|_2^2}{\|\boldsymbol{p}_{n-1}\|_2^2} \quad \text{and} \quad \beta_n = \frac{\|\boldsymbol{p}_{n-1}\|_2^2}{\|\boldsymbol{p}_{n-2}\|_2^2}, \tag{6.8}$$

given $\boldsymbol{p}_{-1} = 0$ and $\boldsymbol{p}_0 = 1$. The individual polynomials $\boldsymbol{p}_i$, i.e., the column vectors of the basis, have a non-uniform norm. The basis is orthogonal, but not orthonormal/unitary. The condition number of the matrix increases with increasing degree due to numerical errors [162]. The Forsythe polynomial basis is fundamentally equivalent to the Chebyshev basis.

(3) Chebyshev. The Chebyshev (also *Tchebichef* or *Tschebyscheff*) polynomials are generated in the range $x \in [0, m-1]$ from the recurrence relationship,

$$n\, \boldsymbol{t}_n(\boldsymbol{x}) - (2n-1)\, \boldsymbol{t}_1(\boldsymbol{x}) \circ \boldsymbol{t}_{n-1}(\boldsymbol{x}) + (n-1) \left\{ 1 - \frac{(n-1)^2}{m^2} \right\} \boldsymbol{t}_{n-2}(\boldsymbol{x}) = \boldsymbol{0}, \tag{6.9}$$

whereby $\boldsymbol{t}_0 = 1$ and $\boldsymbol{t}_1 = (2\,\boldsymbol{x} + 1 - m)/m$. The Chebyshev polynomials form a discrete and orthogonal basis, but they are not orthonormal/unitary. This indicates that the Chebyshev basis degenerates faster than the Legendre basis. The superiority of the Chebyshev moments to Legendre and Zernike moments has been proven in [151, 152]. The discrete Chebyshev polynomials do not have a uniform scaling, where as the Gram polynomials do.

(4) Krawtchouk. The Krawtchouk moments belong to the class of discrete orthogonal moments [199], whereas the orthogonality condition ensures minimal information redundancy. No coordinate space transformations are required and the implementation of these moments does not involve any numerical approximations. Krawtchouk polynomials are scaled to ensure numerical stability, thus, creating a set of weighted Krawtchouk polynomials. These properties made the Krawtchouk moments popular for finding pattern features in the analysis of two-dimensional images.

(5) Gram (-Schmidt). The Gram polynomials are continuous polynomials. They form a complete and orthonormal/unitary set of basis functions when synthesized from uniformly spaced nodes. The polynomials are generated via a three-term relationship in a process which is called a Gram-Schmidt orthogonalization. The recurrence relationship for the Gram polynomials is

$$p_n(x) = 2\,\alpha_n\, x\, p_{n-1}(x) - \frac{\alpha_{n-1}}{\alpha_{n-2}}\, p_{n-2}(x), \tag{6.10}$$

whereby

$$\alpha_{n-1} = \frac{m}{n} \left(\frac{n^2 - 1/2}{m^2 - n^2} \right)^{1/2} \tag{6.11}$$

and

$$p_0(x) = 1, \qquad p_{-1}(x) = 0 \quad \text{and} \quad \alpha_{-1} = 1. \tag{6.12}$$

The computation is performed for equidistant discrete nodes x_k on the real axis,

$$x = -1 + \frac{2k-1}{m} \quad \text{with} \quad 1 \leq k \leq m. \tag{6.13}$$

These points do not span the full range $[-1, +1]$. The bases functions are scaled by $\sqrt{m}$ yielding a unitary bases set. Theoretically, the Gram polynomial form an ideal basis; however, the synthesis via the three-term recurrence relationship introduces serious round-off errors, which accumulate and prevent the synthesis of high degree polynomials of acceptable numerical quality.

6.2.2 Polar/Radial Moments

(1) Zernike. The Zernike polynomials are a sequence of polynomials that are orthogonal on the unit disk [199]. Zernike moments can represent properties of an image with no redundancy or overlap of information between the moments. The moments are synthesized using polar coordinates; consequently, they are invariant to rotation. Thus, they can be utilized to extract features from images that describe the shape characteristics of an object. The computation using a direct method is very time consuming. Zernike polynomials are commonly used in optical applications in order to correct lens distortion or in adaptive optics where they are used to effectively cancel out atmospheric distortion.

(2) Fourier-Mellin. The orthogonal Fourier-Mellin moments are better suited for small data sets (small images) than Zernike moments w.r.t. reconstruction errors and signal to noise ratio [199].

6.2.3 Periodic/Cyclic Moments

(1) Fourier. This basis is complex and time shift invariant; consequently, it is suitable when there are periodic structures in the data. The Fourier basis is equivalent to a polynomial basis where the nodes $\boldsymbol{x}$ are evenly placed on the unit circle on the complex plane [167]. Many coefficients are required to model data which exhibits a simple linear gradient or sub-harmonics. This leads to the Gibbs error: the energy associated with the signal which cannot be modeled is spread over the whole spectrum; as a consequence, perfect reconstruction becomes impossible. On the other hand, the Fourier basis can model periodic signals with very few coefficients. The Fourier basis F is synthesized via

$$\mathsf{F} = \frac{1}{\sqrt{m}} \exp\left\{ -\mathrm{j} \frac{2\pi(k-1)(l-1)}{m} \right\}. \tag{6.14}$$

Given a vector of data $\boldsymbol{x}$ with m points, the spectrum $\boldsymbol{\alpha}$ (Fourier moments) is computed as $\boldsymbol{\alpha} = \mathsf{F}\,\boldsymbol{x}$. The derivatives of F are also periodic.

(2) Discrete cosine. In contrast to the Fourier basis, the discrete cosine transform (DCT) is a real basis and not time shift invariant. It is computationally less expensive than the Fourier basis. The DCT is equivalent to a polynomial basis where the nodes $\boldsymbol{x}$ are placed using Chebyshev points [167]. The DCT basis C is synthesized via

$$\mathsf{C} = \sqrt{\frac{2}{m}} \cos\left\{ \frac{(2k+1)\,l\,\pi}{2m} \right\}, \tag{6.15}$$

whereby the basis is orthonormal with $\mathsf{C}^{\mathrm{T}}\mathsf{C} = \mathsf{I}$ on m nodes. The most popular application employing the DCT is the JPEG image and MPEG video compression formats. Portions that cannot be modeled appear as visible artifacts inside the image or video respectively.

6.3 Vandermonde Polynomial Basis

The use of the geometric Vandermonde basis is a popular approach in literature and non-demanding engineering problems. Although theoretically sound, from a numerical point of view, the Vandermonde basis is poorly conditioned and quickly becomes degenerate as the degree of the polynomial increases [65, 176]. The maximum polynomial degree which can be used before the basis collapses is approximately $d = 8$, depending on the employed computing system. Consequently, computations with this matrix will have insufficient numerical characteristics. Furthermore, these moments are not orthogonal when computed at discrete points and they are not polynomial preserving; these features are important when performing polynomial filtering such as Savitzky-Golay smoothing [164]. No unique inverse for degenerated matrices exists; that being so, the Vandermonde basis is not suited for signal synthesis.

Consider a continuous polynomial $y = y(x)$ of degree d, which is defined as a sum of monomials,

$$y = \sum_{i=0}^{d} \alpha_i \, x^i = \alpha_0 + \alpha_1 \, x + \alpha_2 \, x^2 + \ldots + \alpha_{d-1} \, x^{d-1} + \alpha_d \, x^d. \tag{6.16}$$

Having a number of m points $[x_i, y_i]^{\mathrm{T}}$ available, the equation is discretized and rewritten in concise matrix form,

$$\boldsymbol{y} = \mathsf{V} \, \boldsymbol{\alpha}, \tag{6.17}$$

where $\boldsymbol{y} \in \mathbb{R}^{(m \times 1)}$ is the solution vector, $\mathsf{V} \in \mathbb{R}^{(m \times (d+1))}$ is the Vandermonde design matrix and $\boldsymbol{\alpha} \in \mathbb{R}^{((d+1) \times 1)}$ is the coefficient vector. The expanded form of the above equation is

$$\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} 1 & x_1 & \ldots & x_1^{(d-1)} & x_1^d \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_m & \ldots & x_m^{(d-1)} & x_m^d \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_d \end{bmatrix}. \tag{6.18}$$

The basis is complete, when $m = d + 1$. The common approach to estimate the regression's coefficient vector $\hat{\boldsymbol{\alpha}}$ is the use of the Moore-Penrose pseudoinverse $\mathsf{V}^+ \triangleq (\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}\mathsf{V}^{\mathrm{T}}$ as explained in Section 6.4,

$$\hat{\boldsymbol{\alpha}} = \mathsf{V}^+ \boldsymbol{y}, \tag{6.19}$$

which is a maximum likelihood estimator [95], thus, $\mathbb{E}[\hat{\boldsymbol{\alpha}}] = \boldsymbol{\alpha}$. Note, that $(\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1} = (\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-\mathrm{T}}$. The estimator $\hat{\boldsymbol{y}}$ for the solution vector $\boldsymbol{y}$ is then

$$\hat{\boldsymbol{y}} = \mathsf{V}\hat{\boldsymbol{\alpha}} = \mathsf{V}\mathsf{V}^+ \boldsymbol{y}, \tag{6.20}$$

where $\mathsf{V}\mathsf{V}^+$ is a projection onto the Vandermonde basis. According to Eqn. (5.26), the covariance $\Lambda_{\hat{\boldsymbol{y}}}$ of the solution vector $\hat{\boldsymbol{y}}$ is $\Lambda_{\hat{\boldsymbol{y}}} = \mathsf{L}\Lambda_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}}$ with the linear operator $\mathsf{L} \triangleq \mathsf{V}\mathsf{V}^+$. Hence,

$$\Lambda_{\hat{\boldsymbol{y}}} = \left(\mathsf{V}\mathsf{V}^+\right) \Lambda_{\boldsymbol{y}} \left(\mathsf{V}\mathsf{V}^+\right)^{\mathrm{T}}, \tag{6.21}$$

and the covariance $\Lambda_{\hat{\boldsymbol{\alpha}}}$ of the estimated coefficient vector $\hat{\boldsymbol{\alpha}}$ is $\Lambda_{\hat{\boldsymbol{\alpha}}} = \mathsf{L}\Lambda_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}}$ with $\mathsf{L} \triangleq \mathsf{V}^+$, thus,

$$\Lambda_{\hat{\boldsymbol{\alpha}}} = \mathsf{V}^+\Lambda_{\boldsymbol{y}}\mathsf{V}^{+\mathrm{T}}. \tag{6.22}$$

The polynomial nature of the Vandermonde basis facilitates the computation of its derivatives. According to [89], the analytic derivation y' of the polynomial $y = y(x)$ is

$$y' = \frac{\mathrm{d}y}{\mathrm{d}x} = \alpha_1 + 2\,\alpha_2\,x + \ldots + (d-1)\,\alpha_{d-1}\,x^{d-2} + d\,\alpha_d\,x^{d-1}. \tag{6.23}$$

The derivation $\boldsymbol{y}'$ of the solution vector $\boldsymbol{y}$ is simply computed with a weighting matrix W,

$$\mathsf{W} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & d \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix}. \tag{6.24}$$

However, this methodology does not account for any constraints such as initial or boundary values! The estimation for the derivative is then

$$\begin{aligned} \hat{\boldsymbol{y}}' &= \mathsf{V}\,\hat{\boldsymbol{\alpha}}', \\ &= \mathsf{V}\,\mathsf{W}\,\hat{\boldsymbol{\alpha}}, \\ &= \mathsf{V}\,\mathsf{W}\,\mathsf{V}^{+}\boldsymbol{y}. \end{aligned} \tag{6.25}$$

The covariance $\Lambda_{\hat{\boldsymbol{y}}'}$ of the estimation for the derivation $\hat{\boldsymbol{y}}'$ is $\Lambda_{\hat{\boldsymbol{y}}'} = \mathsf{L}\Lambda_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}}$ with $\mathsf{L} \triangleq (\mathsf{V}\,\mathsf{W}\,\mathsf{V}^{+})$,

$$\Lambda_{\hat{\boldsymbol{y}}'} = \big(\mathsf{V}\,\mathsf{W}\,\mathsf{V}^{+}\big)\,\Lambda_{\boldsymbol{y}}\big(\mathsf{V}\,\mathsf{W}\,\mathsf{V}^{+}\big)^{\mathrm{T}}. \tag{6.26}$$

Despite its bad numerical behavior, the Vandermonde matrix is still a popular choice to solve numerous engineering problems. For instance, MATLAB's built-in function for polynomial regression, `polyfit()`, as well as the methods provided by the more advanced Curve Fitting Toolbox [136], use the Vandermonde basis. The computation of the confidence and prediction intervals of polynomial regression using the Vandermonde basis is explained in Section 8.2.1 for comparison/verification purposes.

6.4 Moore-Penrose Pseudoinverse

The most general formulation of an overdetermined system of equations is written as

$$\boldsymbol{y} = \mathsf{B}\,\boldsymbol{\alpha}, \tag{6.27}$$

where $\boldsymbol{y} \in \mathbb{R}^{(m\times 1)}$ is the vector of measurement data, $\mathsf{B} \in \mathbb{R}^{(m\times(d+1))}$ is an incomplete set of basis functions and $\boldsymbol{\alpha} \in \mathbb{R}^{((d+1)\times 1)}$ are the true, yet unknowable, coefficients of the underlying curve. The matrix B is rectangular; unfortunately, there is no unique inverse to this matrix. Premultiplication of both sides with B^{T} yields the *normal equation*,

$$\mathsf{B}^{\mathrm{T}}\boldsymbol{y} = \mathsf{B}^{\mathrm{T}}\mathsf{B}\,\boldsymbol{\alpha}, \tag{6.28}$$

whereby the term $\{\mathsf{B}^{\mathrm{T}}\mathsf{B}\} \in \mathbb{R}^{((d+1)\times(d+1))}$ is now a square matrix. Assuming that this term is non-singular, inversion yields the minimum norm least squares estimate $\hat{\boldsymbol{\alpha}}$ for the coefficients,

$$\hat{\boldsymbol{\alpha}} = \left(\mathsf{B}^{\mathrm{T}}\mathsf{B}\right)^{-1}\mathsf{B}^{\mathrm{T}}\boldsymbol{y}. \tag{6.29}$$

The Moore-Penrose pseudoinverse B^+ of the matrix B is defined [5, 65] as

$$\mathsf{B}^+ \triangleq \left(\mathsf{B}^{\mathrm{T}}\mathsf{B}\right)^{-1}\mathsf{B}^{\mathrm{T}}. \tag{6.30}$$

A verbose derivation is documented in Section 9.1. The computation using the normal equations is computationally expensive and numerically unstable [65]. As an alternative approach, applying QR-decomposition to the linear operation $\mathsf{B} = \mathsf{Q}\,\mathsf{R}$, where Q is orthogonal with $\mathsf{Q}^{\mathrm{T}}\mathsf{Q} = \mathsf{I}$ and R is upper triangular,

$$\boldsymbol{y} = \mathsf{Q}\,\mathsf{R}\,\boldsymbol{\alpha}, \tag{6.31}$$

and premultiplication of Q^{T},

$$\begin{aligned}\mathsf{Q}^{\mathrm{T}}\boldsymbol{y} &= \mathsf{Q}^{\mathrm{T}}\mathsf{Q}\,\mathsf{R}\,\boldsymbol{\alpha},\\ &= \mathsf{R}\,\boldsymbol{\alpha}.\end{aligned} \tag{6.32}$$

The coefficient vector $\hat{\boldsymbol{\alpha}}$ is then estimated as

$$\hat{\boldsymbol{\alpha}} = \mathsf{R}^+\mathsf{Q}^{\mathrm{T}}\boldsymbol{y}, \tag{6.33}$$

where $\hat{\boldsymbol{\alpha}}$ has minimum norm. This is how MATLAB's `polyfit()` function implements the computation of the pseudoinverse. If the basis is complete, then $\mathsf{R}^{-1} = \mathsf{R}^+$, and the coefficients $\boldsymbol{\alpha}$ are

$$\boldsymbol{\alpha} = \mathsf{R}^{-1}\mathsf{Q}^{\mathrm{T}}\boldsymbol{y}. \tag{6.34}$$

The triangular form of R simplifies the computation of its inverse R^{-1}. Using Eqn. (6.33), the estimation for the output vector $\hat{\boldsymbol{y}} = \mathsf{B}\,\hat{\boldsymbol{\alpha}}$ is

$$\begin{aligned}\hat{\boldsymbol{y}} &= \mathsf{B}\,\mathsf{R}^+\mathsf{Q}^{\mathrm{T}}\boldsymbol{y},\\ &= \mathsf{QR}\,\mathsf{R}^+\mathsf{Q}^{\mathrm{T}}\boldsymbol{y},\\ &= \mathsf{QQ}^{\mathrm{T}}\boldsymbol{y},\end{aligned} \tag{6.35}$$

if R is full rank. In this case, $\mathsf{R}\,\mathsf{R}^+ = \mathsf{I}$ and QQ^{T} is a projection onto Q. Note, that for an incomplete basis, $\mathsf{QQ}^{\mathrm{T}} \neq \mathsf{I}$. Defining the linear operator $\mathsf{L} \triangleq \mathsf{QQ}^{\mathrm{T}}$ enables the computation of the estimation's covariance $\Lambda_{\hat{\boldsymbol{y}}} = \mathsf{L}\Lambda_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}}$ w.r.t. the original data $\boldsymbol{y}$ and its covariance $\Lambda_{\boldsymbol{y}}$,

$$\begin{aligned}\Lambda_{\hat{\boldsymbol{y}}} &= \left(\mathsf{QQ}^{\mathrm{T}}\right)\Lambda_{\boldsymbol{y}}\left(\mathsf{QQ}^{\mathrm{T}}\right)^{\mathrm{T}},\\ &= \mathsf{QQ}^{\mathrm{T}}\Lambda_{\boldsymbol{y}}\mathsf{QQ}^{\mathrm{T}}.\end{aligned} \tag{6.36}$$

Now, consider the projection matrix $\mathsf{P} \triangleq \mathsf{BB}^+$ of the basis $\mathsf{B} = \mathsf{QR}$, i.e.,

$$\begin{aligned}\mathsf{P} &= \mathsf{BB}^+,\\ &= \mathsf{B}\left(\mathsf{B}^{\mathrm{T}}\mathsf{B}\right)^{-1}\mathsf{B}^{\mathrm{T}},\\ &= \mathsf{QR}\left((\mathsf{QR})^{\mathrm{T}}\mathsf{QR}\right)^{-1}(\mathsf{QR})^{\mathrm{T}},\\ &= \mathsf{QR}\left(\mathsf{R}^{\mathrm{T}}\mathsf{Q}^{\mathrm{T}}\mathsf{QR}\right)^{-1}\mathsf{R}^{\mathrm{T}}\mathsf{Q}^{\mathrm{T}},\\ &= \mathsf{QR}\left(\mathsf{R}^{\mathrm{T}}\mathsf{R}\right)^{-1}\mathsf{R}^{\mathrm{T}}\mathsf{Q}^{\mathrm{T}},\\ &= \mathsf{QRR}^+\mathsf{Q}^{\mathrm{T}},\\ &= \mathsf{Q}(\mathsf{RR}^{-1})(\mathsf{R}^{-\mathrm{T}}\mathsf{R}^{\mathrm{T}})\mathsf{Q}^{\mathrm{T}},\\ &= \mathsf{QQ}^{\mathrm{T}}.\end{aligned} \tag{6.37}$$

Therefore, this derivation is valid for any basis B. The projections $\mathsf{P} = \mathsf{B}\mathsf{B}^{+} = \mathsf{Q}\mathsf{Q}^{\mathrm{T}}$, where $\mathsf{B}\mathsf{B}^{+}$ is the projection on B and $\mathsf{Q}\mathsf{Q}^{\mathrm{T}}$ is the projection on Q; therefore, both projections have the same range,

$$\text{range}\left(\mathsf{B}\mathsf{B}^{+}\right) = \text{range}\left(\mathsf{Q}\mathsf{Q}^{\mathrm{T}}\right). \tag{6.38}$$

As a matter of fact, there must exist a discrete orthogonal polynomial (DOP) basis set to perform least squares fitting, which leads to the concept of Gram polynomials G, i.e., the basis $\mathsf{B} \triangleq \mathsf{G}$,

$$\boldsymbol{y} = \mathsf{G}\,\boldsymbol{\alpha} \tag{6.39}$$

with the orthogonality properties $\mathsf{G}^{\mathrm{T}}\mathsf{G} = \mathsf{I}$ and its inverse $\mathsf{G}^{+} = \mathsf{G}^{\mathrm{T}}$. This directly leads to the estimation $\hat{\boldsymbol{\alpha}}$ of the coefficients,

$$\hat{\boldsymbol{\alpha}} = \mathsf{G}^{\mathrm{T}}\boldsymbol{y}. \tag{6.40}$$

The covariance $\Lambda_{\hat{\boldsymbol{\alpha}}} = \mathsf{L}\Lambda_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}}$ of the coefficients is directly computed from the measurement vector's i.i.d. noise $\sigma_{\boldsymbol{y}}^{2}$ with the linear operator $\mathsf{L} \triangleq \mathsf{G}^{\mathrm{T}}$,

$$\begin{aligned} \Lambda_{\hat{\boldsymbol{\alpha}}} &= \mathsf{G}^{\mathrm{T}}\Lambda_{\boldsymbol{y}}\mathsf{G} \quad \text{and} \quad \Lambda_{\boldsymbol{y}} = \sigma_{\boldsymbol{y}}^{2}\mathsf{I}, \\ &= \sigma_{\boldsymbol{y}}^{2}\mathsf{G}^{\mathrm{T}}\mathsf{G}, \\ &= \sigma_{\boldsymbol{y}}^{2}\mathsf{I}, \end{aligned} \tag{6.41}$$

therefore, $\Lambda_{\hat{\boldsymbol{\alpha}}} = \Lambda_{\boldsymbol{y}}$. In other words, the input noise directly maps to the output noise. The noise is evenly spread over all spectral components, which significantly simplifies the noise estimation and enables signal reconstruction of high numerical quality.

7 | Discrete Orthogonal Polynomials

State-of-the-art methodologies for the generation of discrete orthogonal polynomials are based on the classical *three-term recurrence relationship* from functional analysis [58, 65],

$$\boldsymbol{g}_n = \alpha_n \left(\boldsymbol{g}_{n-1} \circ \boldsymbol{x} \right) + \beta_n \, \boldsymbol{g}_{n-2}, \tag{7.1}$$

with the monomial $\boldsymbol{g}_n$ of n^{th} degree, the vector $\boldsymbol{x}$ with m arbitrarily spaced nodes and the scalar recurrence coefficients $\{\alpha_n, \beta_n\}$ of the n^{th} iteration. Different values for $\{\alpha_n, \beta_n\}$ deliver different kinds of polynomials. The Gram-Schmidt orthogonalization is known to be unstable. The limiting factor for the three-term recurrence relationship in this formulation is the systematic propagation of round-off errors; as a consequence, it is virtually impossible to synthesize bases of higher degree. The currently synthesized n^{th} monomial $\boldsymbol{g}_n$ only depends on its last two predecessors $\boldsymbol{g}_{n-1}$ and $\boldsymbol{g}_{n-2}$, the errors propagate from monomials of lower degree to monomials of higher degree. The central limit theorem cannot be applied and the errors do not behave according to Gaussian distribution.

7.1 Existence of the Basis

The presented generalized recurrence relationship is usable to synthesize a high quality polynomial basis for arbitrary nodes of virtually unlimited degree [162, 163, 165, 167]. Continuous polynomials are considered which form an orthogonal basis when evaluated over a discrete measure. The concept of the conventional three-term recurrence relationship is extended so that the n^{th} monomial $\boldsymbol{g}_n$ is a linear combination of $(\boldsymbol{g}_{n-1} \circ \boldsymbol{x})$ and of all previously generated basis functions G_{n-1}. The theoretical proof shows, that there is one and only one unitary discrete polynomial basis which can be generated from a sum of monomials [162]. The procedure is stable w.r.t. numerical errors; the maximum error corresponds to the numerical resolution of the data type being used during generation. Within this thesis, the matrix G refers to this discrete orthogonal polynomial (DOP) basis. The robust recurrence relationship is formulated as a vector-matrix algebraic operation,

$$\boldsymbol{g}_n = \alpha_n \left(\boldsymbol{g}_{n-1} \circ \boldsymbol{x} \right) + \mathsf{G}_{n-1} \, \boldsymbol{\beta}_n, \tag{7.2}$$

where $\mathsf{G}_{n-1} = [\boldsymbol{g}_0, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_{n-1}]$ is a matrix containing the complete polynomial basis, i.e., the monomials as column vectors, up to degree $n-1$, and $\boldsymbol{\beta}_n$ is the corresponding recurrence coefficient vector. In order to increase readability, it is defined $\alpha \triangleq \alpha_n$ and $\boldsymbol{\beta} \triangleq \boldsymbol{\beta}_n$ for the currently synthesized degree n. Two constraints are placed onto the basis functions,

$$C_1 : \quad \mathsf{G}_{n-1}^{\mathrm{T}} \, \boldsymbol{g}_n = \boldsymbol{0} \quad \text{and} \quad C_2 : \quad \boldsymbol{g}_n^{\mathrm{T}} \, \boldsymbol{g}_n = 1, \tag{7.3}$$

where C_1 is the orthogonality condition and C_2 is the unit norm condition. Using Eqn. (7.2) with the first constraint $C_1 : \mathsf{G}_{n-1}^{\mathrm{T}}\, \boldsymbol{g}_n = \boldsymbol{0}$, yields

$$\begin{aligned} \boldsymbol{0} &= \mathsf{G}_{n-1}^{\mathrm{T}} \{\alpha\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + \mathsf{G}_{n-1}\, \boldsymbol{\beta}\}, \\ &= \alpha\, \mathsf{G}_{n-1}^{\mathrm{T}}\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + \mathsf{G}_{n-1}^{\mathrm{T}}\, \mathsf{G}_{n-1}\, \boldsymbol{\beta}, \end{aligned} \tag{7.4}$$

where $\mathsf{G}_{n-1}^{\mathrm{T}}\, \mathsf{G}_{n-1} = \mathsf{I}$ is unitary. Solving the above equation for $\boldsymbol{\beta}$ delivers

$$\boldsymbol{\beta} = -\alpha\, \mathsf{G}_{n-1}^{\mathrm{T}}\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}). \tag{7.5}$$

Using Eqn. (7.2) with the second constraint $C_2 : \boldsymbol{g}_n^{\mathrm{T}}\, \boldsymbol{g}_n = 1$, yields

$$\begin{aligned} 1 &= \{\alpha\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + \mathsf{G}_{n-1}\, \boldsymbol{\beta}\}^{\mathrm{T}} \{\alpha\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + \mathsf{G}_{n-1}\, \boldsymbol{\beta}\}, \\ &= \alpha^2\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}} (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + 2\, \alpha\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}}\, \mathsf{G}_{n-1}\, \boldsymbol{\beta} + \boldsymbol{\beta}^{\mathrm{T}}\, \mathsf{G}_{n-1}^{\mathrm{T}}\, \mathsf{G}_{n-1}\, \boldsymbol{\beta}, \end{aligned} \tag{7.6}$$

where $\mathsf{G}_{n-1}^{\mathrm{T}}\, \mathsf{G}_{n-1} = \mathsf{I}$ is unitary,

$$1 = \alpha^2\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}} (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + 2\, \alpha\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}}\, \mathsf{G}_{n-1}\, \boldsymbol{\beta} + \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\beta}. \tag{7.7}$$

Substitution of $\boldsymbol{\beta}$ from Eqn. (7.5),

$$\begin{aligned} 1 = {} & \alpha^2\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}} (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) \\ & + 2\, \alpha\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}}\, \mathsf{G}_{n-1}\, \{-\alpha\, \mathsf{G}_{n-1}^{\mathrm{T}}\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})\} \\ & + \{-\alpha\, \mathsf{G}_{n-1}^{\mathrm{T}}\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})\}^{\mathrm{T}} \{-\alpha\, \mathsf{G}_{n-1}^{\mathrm{T}}\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})\}, \end{aligned} \tag{7.8}$$

and expansion of the terms yields

$$\begin{aligned} 1 = {} & \alpha^2\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}} (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) \\ & - 2\, \alpha^2\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}}\, \mathsf{G}_{n-1} \mathsf{G}_{n-1}^{\mathrm{T}}\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) \\ & + \alpha^2\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}}\, \mathsf{G}_{n-1} \mathsf{G}_{n-1}^{\mathrm{T}}\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}). \end{aligned} \tag{7.9}$$

Merging of the terms,

$$1 = \alpha^2\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}} (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) - \alpha^2\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}}\, \mathsf{G}_{n-1} \mathsf{G}_{n-1}^{\mathrm{T}}\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}), \tag{7.10}$$

and factorization delivers

$$1 = \alpha^2\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}}\, \{\mathsf{I} - \mathsf{G}_{n-1} \mathsf{G}_{n-1}^{\mathrm{T}}\}\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}). \tag{7.11}$$

The projection $\{\mathsf{I} - \mathsf{G}_{n-1} \mathsf{G}_{n-1}^{\mathrm{T}}\}$ is idempotent; therefore, it can be expanded to

$$1 = \alpha^2\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}}\, \{\mathsf{I} - \mathsf{G}_{n-1} \mathsf{G}_{n-1}^{\mathrm{T}}\}^{\mathrm{T}}\, \{\mathsf{I} - \mathsf{G}_{n-1} \mathsf{G}_{n-1}^{\mathrm{T}}\} (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) \tag{7.12}$$

in ordert to solve the above equation for α,

$$\alpha = \pm \frac{1}{\sqrt{(\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}}\, \{\mathsf{I} - \mathsf{G}_{n-1} \mathsf{G}_{n-1}^{\mathrm{T}}\}^{\mathrm{T}}\, \{\mathsf{I} - \mathsf{G}_{n-1} \mathsf{G}_{n-1}^{\mathrm{T}}\} (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})}}. \tag{7.13}$$

The ± is irrelevant since it only changes the sign of the moment. The above formula is then conveniently written as

$$\alpha = \frac{1}{\|\{\mathsf{I} - \mathsf{G}_{n-1}\mathsf{G}_{n-1}^{\mathrm{T}}\}\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x})\|_2}. \tag{7.14}$$

Complete reorthogonalization is performed by substituting $\boldsymbol{\beta}$ from Eqn. (7.5) into the recurrence relationship from Eqn. (7.2), thus,

$$\begin{aligned}
\boldsymbol{g}_n^{\perp} &= \alpha\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + \mathsf{G}_{n-1}\,\boldsymbol{\beta},\\
&= \alpha\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + \mathsf{G}_{n-1}\,\{-\alpha\,\mathsf{G}_{n-1}^{\mathrm{T}}\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x})\},\\
&= \alpha\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) - \alpha\,\mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x}),\\
&= \alpha\,\{\mathsf{I} - \mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\}\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x}).
\end{aligned} \tag{7.15}$$

The term $\mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}$ is a projection onto the basis function G_{n-1} and $\{\mathsf{I} - \mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\}$ is the projection onto the orthogonal complement of G_{n-1}. All correlations in $(\boldsymbol{g}_{n-1} \circ \boldsymbol{x})$ with G_{n-1} are removed; this effectively eliminates all quantization (rounding) errors occurring during the computation. Eqn. (7.15) is a generalized recurrence relationship, whereby the selection of the nodes $\boldsymbol{x}$ determines the function being synthesized. This truly arbitrary nodes may lie in the complex plane within the unit circle. The polynomial basis synthesized from a set of evenly spaced nodes on the real axis in the interval $[-1, +1]$ are called discrete unitary Gram polynomials and are denoted by G throughout in this thesis. The theorem from Eqn. (7.15) is proven by applying the projection onto the orthogonal complement onto $\boldsymbol{g}_n$, hence,

$$\begin{aligned}
\boldsymbol{g}_n^{\perp} &= \{\mathsf{I} - \mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\}\,\boldsymbol{g}_n,\\
&= \{\mathsf{I} - \mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\}\,\{\alpha\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + \mathsf{G}_{n-1}\,\boldsymbol{\beta}\},\\
&= \alpha\,\{\mathsf{I} - \mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\}\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + \{\mathsf{I} - \mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\}\,\mathsf{G}_{n-1}\,\boldsymbol{\beta}.
\end{aligned} \tag{7.16}$$

The projection of G_{n-1} on its own orthogonal complement is a matrix of zeros,

$$\{\mathsf{I} - \mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\}\,\mathsf{G}_{n-1} = \mathsf{0}. \tag{7.17}$$

As a consequence, Eqn. (7.16) is equivalent to Eqn. (7.15). The perfect conditioning of the unitary basis is achieved via normalization,

$$\hat{\boldsymbol{g}}_n^{\perp} = \frac{\boldsymbol{g}_n^{\perp}}{\|\boldsymbol{g}_n^{\perp}\|_2} = \frac{\boldsymbol{g}_n^{\perp}}{\sqrt{\boldsymbol{g}_n^{\perp\,\mathrm{T}}\boldsymbol{g}_n^{\perp}}}. \tag{7.18}$$

Substitution of Eqn. (7.15), and knowing that the projection is idempotent, yields

$$\hat{\boldsymbol{g}}_n^{\perp} = \frac{\alpha\,\{\mathsf{I} - \mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\}\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x})}{\sqrt{\alpha^2\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}}\,\{\mathsf{I} - \mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\}\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x})}}. \tag{7.19}$$

The scalar α is cancelled out since $\alpha/\sqrt{\alpha^2} = \operatorname{sgn}(\alpha)$,

$$\hat{\boldsymbol{g}}_n^{\perp} = \frac{\{\mathsf{I} - \mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\}\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x})}{\sqrt{(\boldsymbol{g}_{n-1} \circ \boldsymbol{x})^{\mathrm{T}}\,\{\mathsf{I} - \mathsf{G}_{n-1}\,\mathsf{G}_{n-1}^{\mathrm{T}}\}\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x})}}. \tag{7.20}$$

The normalization is independent from α. Comparison with Eqn. (7.15) reveals that

$$\hat{\boldsymbol{g}}_n^{\perp} = \boldsymbol{g}_n^{\perp} = \alpha \left\{ \mathsf{I} - \mathsf{G}_{n-1}\, \mathsf{G}_{n-1}^{\mathrm{T}} \right\} (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}). \tag{7.21}$$

This is an incremental orthogonalization process which eliminates all correlations with previously synthesized basis functions. The basis G_n of degree n is acquired by concatenating the existing basis G_{n-1} with the newly generated monomial $\hat{\boldsymbol{g}}_n^{\perp}$, thus,

$$\mathsf{G}_n = \left[\mathsf{G}_{n-1}, \hat{\boldsymbol{g}}_n^{\perp} \right] \quad \text{and} \quad \mathsf{G}_n^{\mathrm{T}} \mathsf{G}_n = \mathsf{I}. \tag{7.22}$$

The basis fulfills the orthogonality condition for almost arbitrary degree, which is demonstrated in Section 7.4. As a matter of fact, the condition number should be unity independent of the synthesized degree n and the number of points m in $\boldsymbol{x}$, i.e., $\mathrm{cond}\,(\mathsf{G}_n) = 1$.

7.2 Synthesis Algorithm

The theoretical proof from the previous section leads directly to the synthesis algorithm using a modified Lanczos process with complete reorthogonalization [65]. This is the first algorithm which synthesizes a unitary polynomial basis for a set of truly arbitrary nodes [89, 162]. This property is of special importance in measurement applications, where the positioning of sensors may be defined by the system requirements and/or environmental constraints.

- **Initial step.** Produce the monomials of degree 0 (constant) and 1 (slope). Generate $\boldsymbol{g}_0$ as a column vector of m ones $\boldsymbol{1}$ and normalize the vector to have a 2-norm of 1,

  $$\boldsymbol{g}_0 = \frac{\boldsymbol{1}}{\sqrt{m}} \quad \text{and} \quad \boldsymbol{g}_1 = \frac{\boldsymbol{x} - \bar{x}}{\|\boldsymbol{x} - \bar{x}\|_2}. \tag{7.23}$$

  The vector $\boldsymbol{x}$ has m elements containing the values of the nodes at which the basis functions are to be generated, $\bar{x}$ is the arithmetic mean of these values. The polynomial basis G_1 of degree 1 is then concatenated via

  $$\mathsf{G}_1 = \left[\boldsymbol{g}_0, \boldsymbol{g}_1 \right]. \tag{7.24}$$

- **Incremental process.** Synthesize monomials of degree n until the target degree d is reached. Within this thesis, it is defined $\mathsf{G}_d \triangleq \mathsf{G}$. Combining step two and three equals the operation formulated in Eqn. (7.15).

  1. Computation of the monomial $\boldsymbol{g}_n$ of degree n,

     $$\boldsymbol{g}_n = \boldsymbol{g}_1 \circ \boldsymbol{g}_{n-1}; \tag{7.25}$$

  2. Complete reorthogonalization of $\boldsymbol{g}_n$ w.r.t. G_{n-1} via

     $$\boldsymbol{g}_n^{\perp} = \left\{ \mathsf{I} - \mathsf{G}_{n-1}\, \mathsf{G}_{n-1}^{\mathrm{T}} \right\} \boldsymbol{g}_n, \tag{7.26}$$

     whereby G_{n-1} is the existing polynomial basis of degree $n - 1$. It is important to note, that the reorthogonalization is performed on the complete existing basis, not just the previous monomial;

3. Normalization of the monomial $\boldsymbol{g}_n^{\perp}$ via

$$\hat{\boldsymbol{g}}_n^{\perp} = \frac{\boldsymbol{g}_n^{\perp}}{\|\boldsymbol{g}_n^{\perp}\|_2}; \tag{7.27}$$

4. Augmentation of the existing polynomials G_{n-1} with the new monomial $\hat{\boldsymbol{g}}_n^{\perp}$,

$$\mathsf{G}_n = \left[\mathsf{G}_{n-1}, \hat{\boldsymbol{g}}_n^{\perp}\right]. \tag{7.28}$$

   The polynomial basis G_n of degree n now contains a linear combination of all lower degree monomials.

The synthesis algorithm uses only fundamental linear algebra. This property enables the automatic generation of `C` code which is functionally equivalent to MATLAB's model code. Consequently, this enables the generation of suitable basis functions on embedded systems in the field as the needs arises, e.g., within a WSAN environment.

7.3 Node Placement

The solution of inverse problems always incorporates the necessity of a-priori knowledge about the physical processes. In a mathematical sense, this includes the selection of a suitable basis function for the description of the system's behavior. Furthermore, when solving ODE/PDE boundary value problems, constraints are applied at predefined locations. The placement of the nodes $\boldsymbol{x}$ determines the form of the basis function synthesized; hence, the need for arbitrary node placement is facilitated. The most important issues are:

1. The design of **application specific, customized basis functions** for the optimal inverse solution of physical systems. In this manner, the computation can be improved w.r.t. noise propagation and numerical efficiency. This is of particular interest when signals need to be analyzed which are insufficiently described by conventional basis functions;

2. The **optimization of covariance propagation** and **reduction of Runge's phenomenon**. The variance of the solution depends on the placement of the sensors, which directly corresponds to the placement of the nodes. This means in fact, that with the very same sensors and with identical noise level, a measurement with a better confidence interval can be achieved by placing the sensors at the optimal locations and synthesizing the matching basis functions.

Fig. 7.1 visualizes the synthesized basis functions on the real axis for uniform as well as irregular node placement. Applications for such basis functions can be found in metric vision for the description of geometric systems or in the solution of differential equations. Fig. 7.3 shows nodes on the complex plane and the corresponding basis functions in Fig. 7.2, such a setup is required for processing of periodic signals. Geometric measurements commonly reside in the real domain $\mathbb{R}$; therefore, the focus is laid onto real-valued basis functions within this thesis. Nevertheless, the examples demonstrate that the DOP basis is valid for $\boldsymbol{x} \in \mathbb{R}$ as well as $\boldsymbol{x} \in \mathbb{C}$.

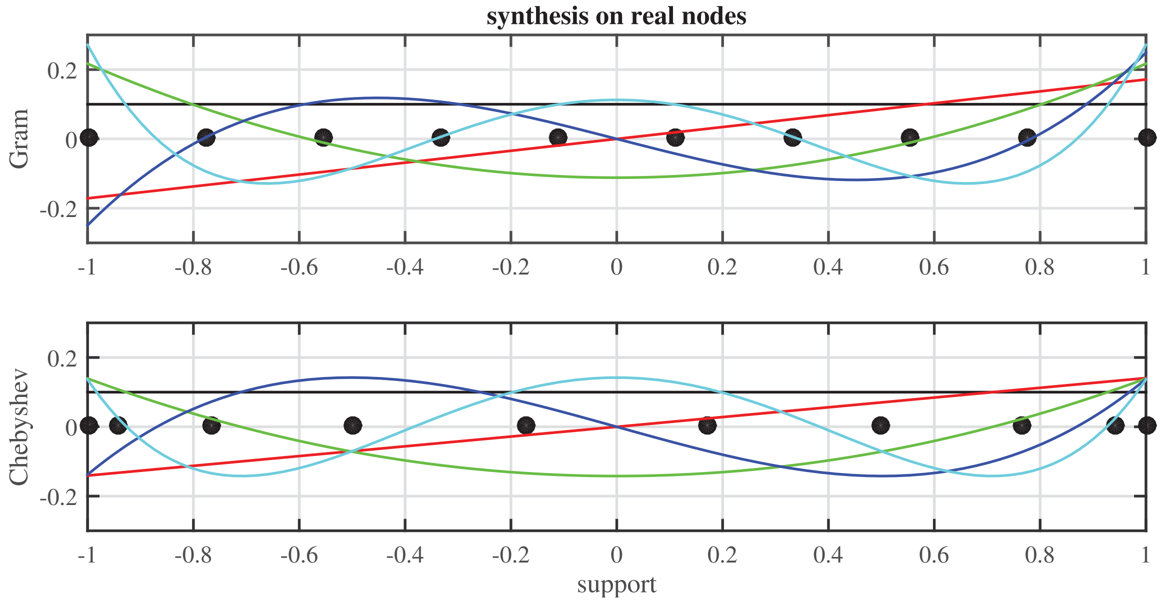


Fig. 7.1 The nodes $\boldsymbol{x}$ are located on the real axis $\mathbb{R}$ with either equidistant (Gram) or irregular (Chebyshev) spacing. The monomials $\boldsymbol{g}_0$ for the constant term (black), $\boldsymbol{g}_1$ for the slope (red), $\boldsymbol{g}_2$ for the parabola (green), $\boldsymbol{g}_3$ for the cubic (b) and $\boldsymbol{g}_4$ fourth order (cyan) term are plotted for both setups. It is clearly visible, that a difference in node placement produces different basis functions.

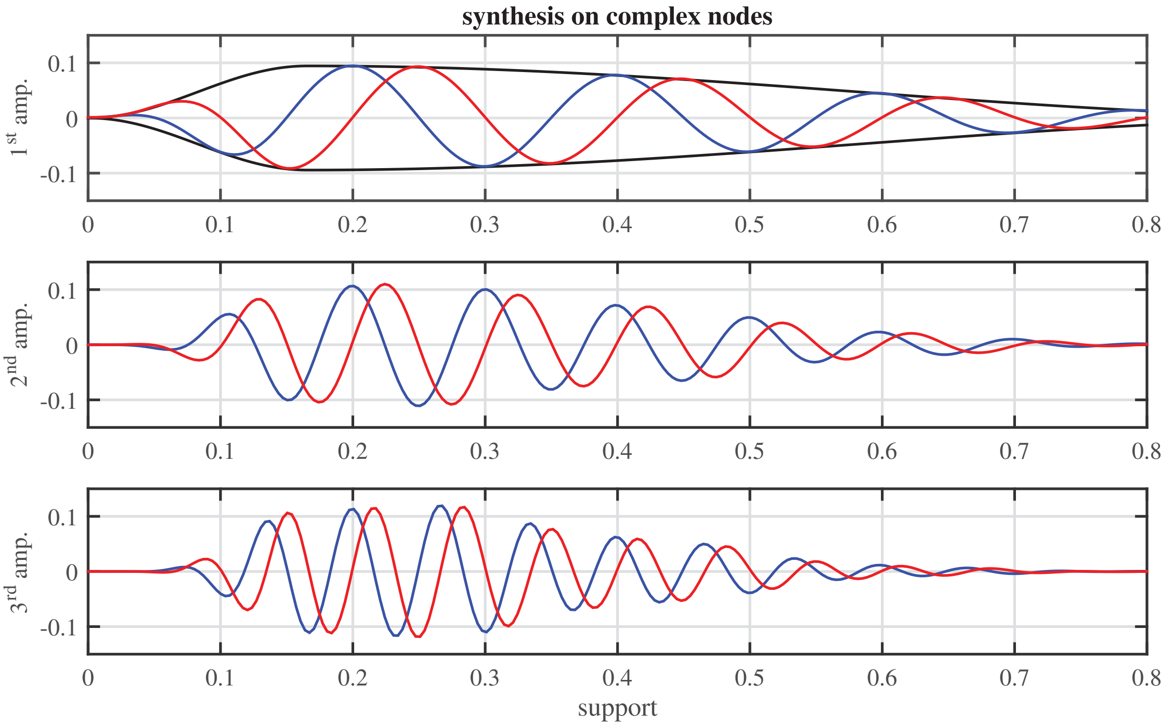


Fig. 7.2 The real (blue) and imaginary (red) amplitude (amp.) components of the first three basis functions. The real and imaginary components are in perfect quadrature to each other and bounded by one and the same envelope (black); as a result, the function is locally shift invariant and effectively eliminates the need for all-pass filtering in signal detection.

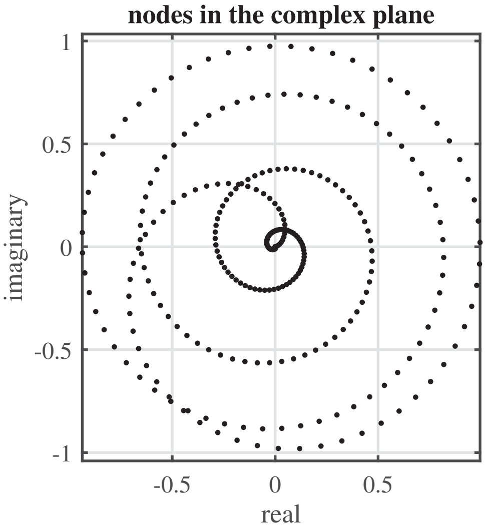


Fig. 7.3 The nodes $\boldsymbol{x}$ are located on the complex plane $\mathbb{C}$ within the unit circle for the generation of the basis functions shown in Fig. 7.2. The placement was performed using polar coordinates, i.e., $\boldsymbol{x} = \Re(\boldsymbol{x}) + \Im(\boldsymbol{x}) = \boldsymbol{r}\ (\cos(\boldsymbol{\phi}) + \mathrm{i}\sin(\boldsymbol{\phi})) = \boldsymbol{r}\, \mathrm{e}^{\mathrm{i}\boldsymbol{\phi}}$. The radius $\boldsymbol{r}$ determines the magnitude envelope for the set of complex basis functions and the angular coordinate $\boldsymbol{\phi}$ determines the instantaneous phase. The basis functions model the characteristic envelope of the oscillations and the characteristic frequencies of a given machine.

7.4 Numerical Quality

There exists numerous objective measures for the quality of basis functions in order to determine the sources of numerical error. These are a-posteriori measurements, i.e., the quality is evaluated after the synthesis. According to [89], this approach is to be preferred over a-priori prediction of the error bound, because a-priori predictions yield unreliable results. The most important measures for the numerical quality of the bases are:

1. **Rank.** A matrix $\mathsf{B} \in \mathbb{R}^{(m \times m)}$ is rank deficient when there are linear dependencies within the matrix, i.e., $\operatorname{rank}(\mathsf{B}) < m$; unfortunately, rounding errors degenerate a matrix. A numerically stable orthogonal basis set must remain full-rank at any given degree;

2. **Condition number.** A unitary basis B is required to have a condition number of 1, i.e., $\operatorname{cond}(\mathsf{B}) = 1$. This ensures a minimization of numerical error propagation;

3. **Matrix R from QR decomposition.** The matrix R reveals the linear correlations within a basis functions set, i.e., the numerical errors occurring during the synthesis with the recurrence relationship;

4. **Projection onto the orthogonal complement.** Perfect reconstruction of a signal is only possible if the projection onto the orthogonal complement $\mathsf{R} = \{\mathsf{I} - \mathsf{B}\,\mathsf{B}^{\mathrm{T}}\} = 0$ is a perfect matrix of zeros, see Fig. 7.4. This requires the basis to be complete, i.e., $\mathsf{B} \in \mathbb{R}^{(m \times m)}$ with degree $d = m - 1$;

5. **Orthogonality of the complete basis set.** The orthogonality condition must be fulfilled for the complete basis $\mathsf{B} \in \mathbb{R}^{(m \times m)}$ with degree $d = m-1$, i.e., $\mathsf{R} = \{\mathsf{I} - \mathsf{B}^{\mathrm{T}}\mathsf{B}\} = 0$, see Fig. 7.5;

6. **Orthogonality of the incomplete basis set.** Furthermore, the orthogonality conditions must also hold for an incomplete basis $\hat{\mathsf{B}} \in \mathbb{R}^{(m \times (d+1))}$ with degree $d < m - 1$, i.e., $\mathsf{R} = \{\mathsf{I} - \hat{\mathsf{B}}^{\mathrm{T}}\hat{\mathsf{B}}\} = 0$.

The last three quality measures are the most significant tests for the quality evaluation of orthogonal bases functions, because those operations actually appear in practical applications.

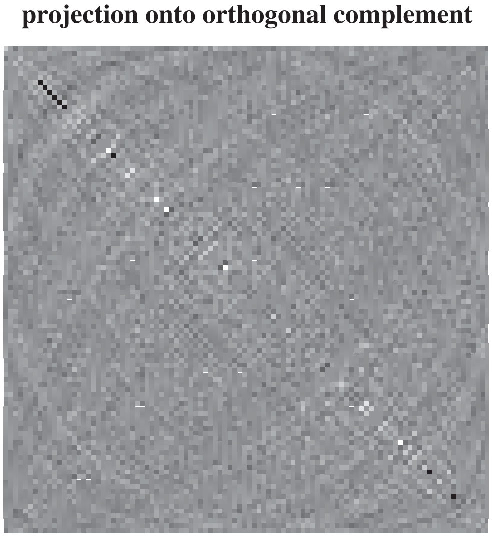


Fig. 7.4 The structure of the projection onto the orthogonal complement's residual $\mathsf{R} = \{\mathsf{I} - \mathsf{G}\mathsf{G}^{\mathrm{T}}\}$ with a complete DOP basis G of degree $d = 100$.

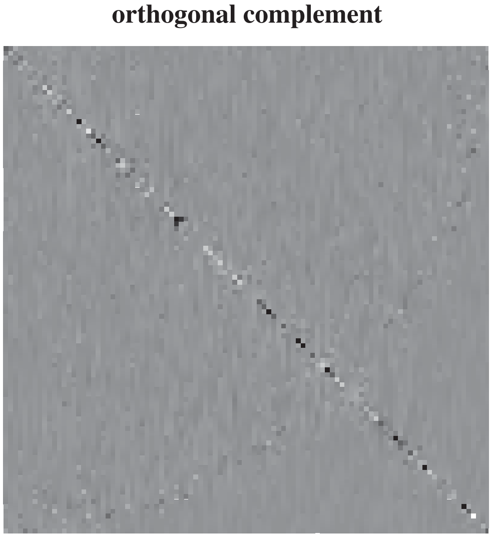


Fig. 7.5 The structure of the orthogonal complement's residual $\mathsf{R} = \{\mathsf{I} - \mathsf{G}^{\mathrm{T}}\mathsf{G}\}$. The values have a magnitude of 10^{-16} in both illustrations.

The following operations are suitable to summarize the numerical quality of the matrices into a single-number performance indicator, which is necessary to objectively compare different kinds of basis functions with varying parameters. The behavior of the three presented performance indicators is visualized in Fig. 7.6.

1. **Determinant.** The determinant of a matrix is only a theoretical quality measure as its computation is numerically unstable and incorporates high computation costs [65]. The theoretical estimate for the quality measure is calculated as $\epsilon_{\mathrm{det}} = \{1 - \det \mathsf{R}\} = 0$;

2. **Maximum norm.** The max norm is an elementwise norm where the single largest element within the matrix is selected, i.e., $\epsilon_{\mathrm{max}} = \|\mathsf{R}\|_{\mathrm{max}} = \max\{|r_{ij}|\}$. Nevertheless, analyzing only the largest occurring value may leads to an invalid conclusion of the actual quality of the basis;

3. **Frobenius norm.** The Frobenius norm is the sum of the squares of all errors in R w.r.t. the orthogonality of the basis functions, i.e., $\epsilon_{\mathrm{F}} = \|\mathsf{R}\|_{\mathrm{F}} = \sqrt{\mathrm{trace}\,(\mathsf{R}\mathsf{R}^{\mathrm{T}})}$. Consequently, the Frobenius norm yields the best estimation for the total error and will be used in the numerical tests.

Model- and Software-in-the-loop testing is performed using MATLAB model code and automatically generated `C` target code[1]. The quality of four different kinds of basis function sets is compared: the Gram polynomials using the classical Gram-Schmidt orthogonalization; a set of Chebyshev polynomials using the three-term recurrence relationship; a Vandermonde matrix and the new discrete orthogonal polynomials (DOP) synthesized by the generalized recurrence relationship.

[1] A 64-bit Unix based host system was used for the numerical software-in-the-loop experiments. MATLAB's numerical accuracy for floating point operations is $2.2204\,10^{-16}$.

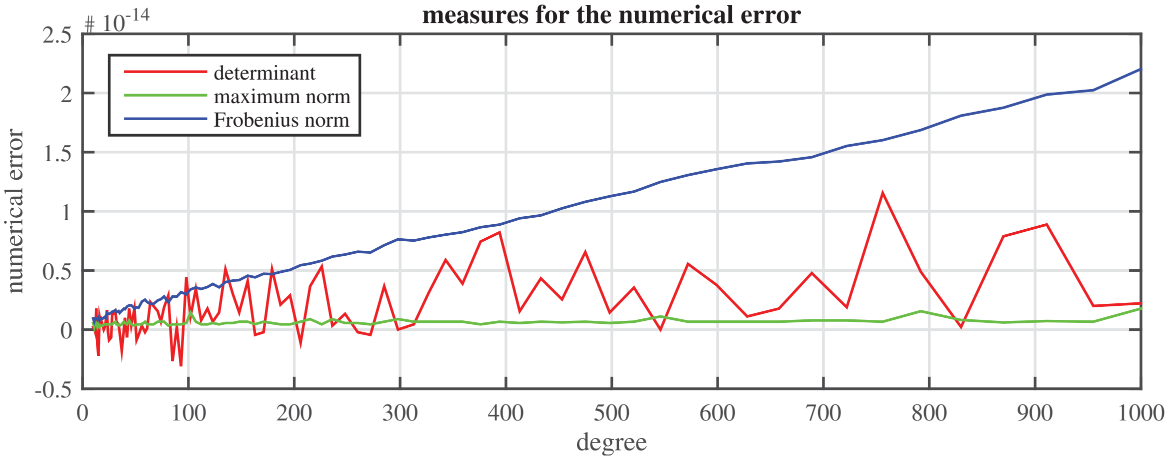


Fig. 7.6 The graphic demonstrates the behavior of the three proposed measures for the estimation of the numerical quality of $\mathsf{R} = \{\mathsf{I} - \mathsf{G}^{\mathrm{T}}\mathsf{G}\}$. The error indicators ϵ_{det}, ϵ_{max} and ϵ_{F} are plotted up to degree $d = 1000$. Obviously, the determinant is numerically unstable and delivers unreliable results, while the maximum norm is not significant and gives only information about the single largest error within the residual matrix. The Frobenius norm appropriately scales according to the expected numerical quality.

The number of significant digits η is estimated via

$$\eta = -\log_{10}(\epsilon). \tag{7.29}$$

Direct comparison of Fig. 7.4 and 7.5 reveals, that the errors for the orthogonal complement $\{\mathsf{I} - \mathsf{G}^{\mathrm{T}}\mathsf{G}\}$ are concentrated at the diagonal of the residual matrix, while the projection onto the orthogonal complement $\{\mathsf{I} - \mathsf{G}\mathsf{G}^{\mathrm{T}}\}$ provides a less clear structure. The numerical differences are minimal; therefore, only the orthogonal complement is computed for the four sets during the following numerical experiments. Fig. 7.7 shows the results for the complete sets. Fig. 7.8 compares the results between MATLAB code and the functionally identical C code for the DOP basis. Fig. 7.9 is for a fixed number of $m = 1000$ nodes and the degree of the polynomial d is constantly increasing. The Vandermonde and Gram-Schmidt bases are collapsing at relatively low degrees, while the Chebyshev basis progressively loses significant digits with increasing degree.

The results indicate, that the DOP algorithm presented in this thesis is numerically stable even for extremely high degrees. Furthermore, the automatically generated C can be considered numerically equivalent, the approximate number of significant digits is $< \pm 0.2$ up to degree $d = 500$.

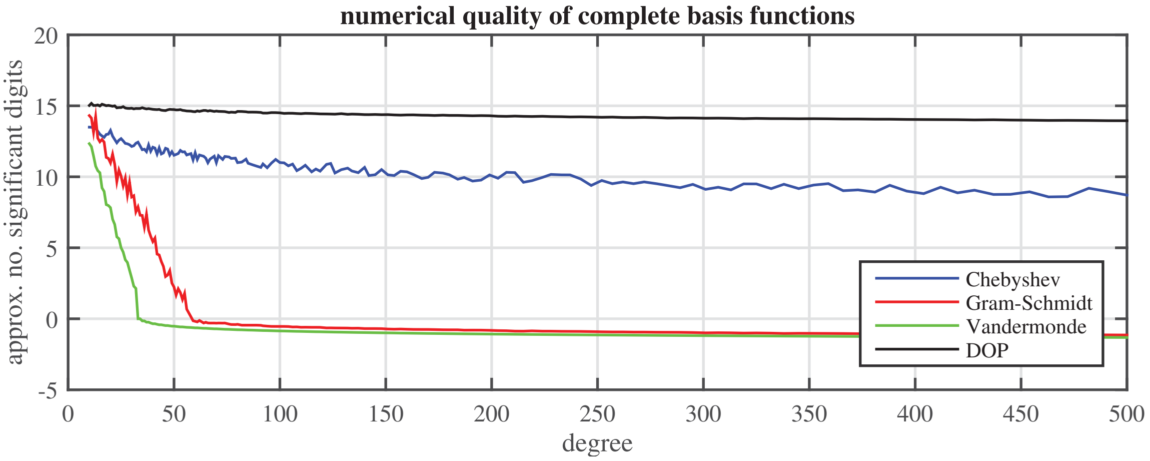


Fig. 7.7 Quality of the complete basis computed as $\epsilon_{\mathrm{F}} = \|\mathsf{I} - \mathsf{B}^{\mathrm{T}}\mathsf{B}\|_{\mathrm{F}}$ on $m = d + 1$ nodes.

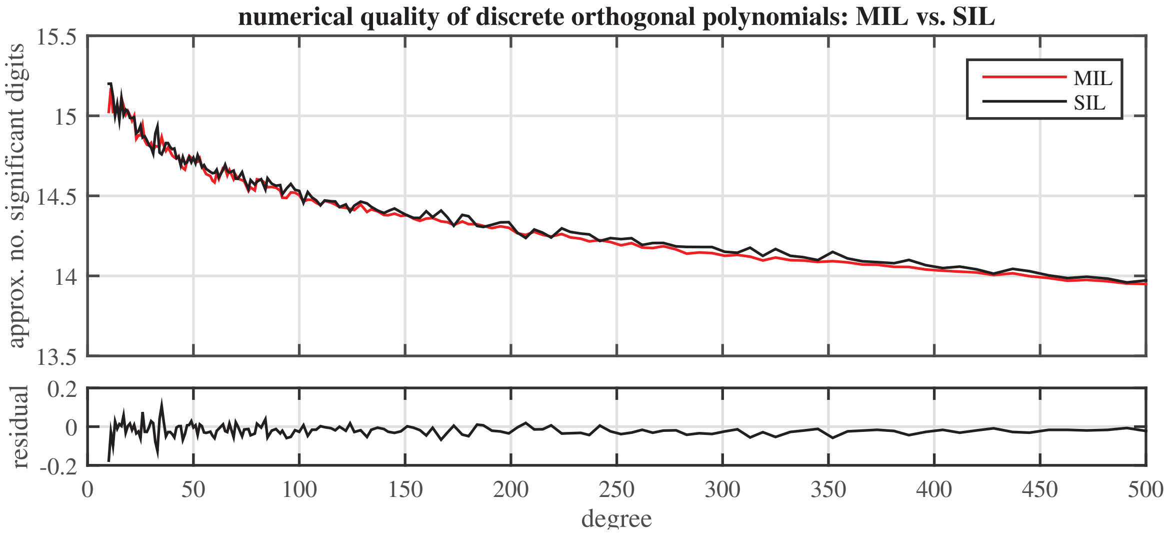


Fig. 7.8 Quality of the complete DOP basis in MATLAB model code (MIL) and C target code (SIL).

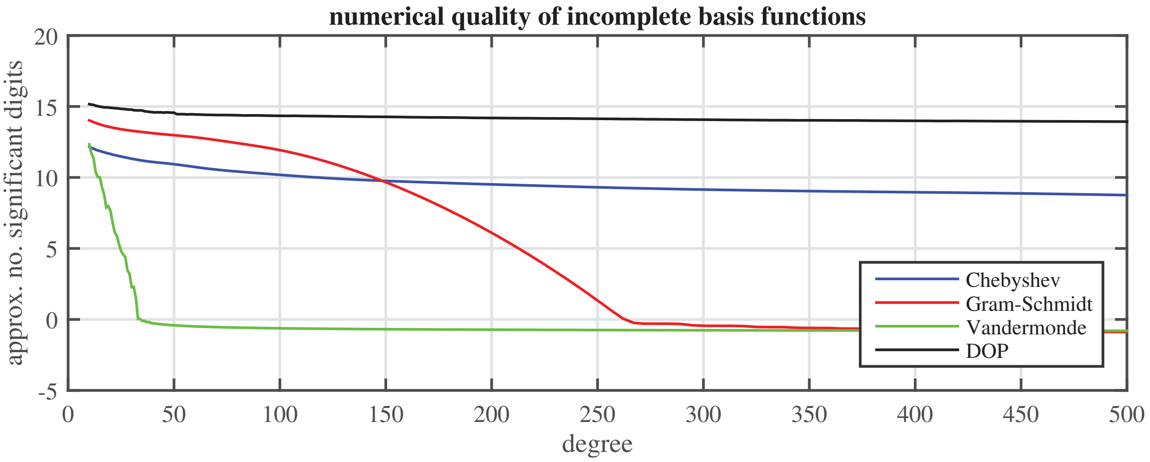


Fig. 7.9 Quality of the incomplete basis computed as $\epsilon_{\mathrm{F}} = \|\mathsf{I} - \hat{\mathsf{B}}^{\mathrm{T}}\hat{\mathsf{B}}\|_{\mathrm{F}}$ on $m = 1000$ nodes.

7.5 Noise Behavior

Throughout this thesis, it is implicitly assumed that $\boldsymbol{y} \triangleq \tilde{\boldsymbol{y}}$, whereby the observations $\tilde{\boldsymbol{y}}$ are perturbed by Gaussian noise $\boldsymbol{\epsilon}$. The perturbations are intrinsic to measurement data $\boldsymbol{y}$ and explicitly stating this property delivers no additional useful information. However, this section deals with the analysis of the noise behavior of these observation. Every measurement is subject to noise; consequently,

$$\tilde{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{\epsilon} = \boldsymbol{y} + \delta\, \boldsymbol{u}, \tag{7.30}$$

where here $\boldsymbol{y}$ is the true, yet unknown, ideal value. The additive white noise is $\boldsymbol{\epsilon} = \delta\, \boldsymbol{u}$, where $\boldsymbol{u}$ is a vector of normalized Gaussian noise and δ is the standard deviation of the noise. The spectrum $\tilde{\boldsymbol{\alpha}}$ of $\tilde{\boldsymbol{y}}$ is computed as

$$\tilde{\boldsymbol{\alpha}} = \mathsf{B}^{+}\, \tilde{\boldsymbol{y}} = \mathsf{B}^{+}\, (\boldsymbol{y} + \delta\, \boldsymbol{u}), \tag{7.31}$$

where B^{+} is the pseudoinverse of a general basis function B. The incorporated noise is $\boldsymbol{\epsilon} = \tilde{\boldsymbol{y}} - \boldsymbol{y} = \delta\, \boldsymbol{u}$. Therefore, the difference in the spectrum is

$$\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}} = \mathsf{B}^{+}\, (\boldsymbol{y} - \tilde{\boldsymbol{y}}) = \mathsf{B}^{+}\, \delta\, \boldsymbol{u} \tag{7.32}$$

and computation of any vector norm yields

$$\|\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}}\| = \|\mathsf{B}^{+}\, (\delta\, \boldsymbol{u})\|. \tag{7.33}$$

If the basis being used is unitary, i.e., $\mathsf{B} \triangleq \mathsf{G}$ with $\mathsf{G}^{\mathrm{T}}\mathsf{G} = \mathsf{I}$, $\mathsf{G}^{+} = \mathsf{G}^{\mathrm{T}}$ and $\|\mathsf{G}\| = 1$, then

$$\|\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}}\| = \|\mathsf{G}^{\mathrm{T}}\, (\delta\, \boldsymbol{u})\| = |\delta|\, \|\mathsf{G}^{\mathrm{T}}\|\, \|\boldsymbol{u}\| = \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|. \tag{7.34}$$

The result shows, that the noise has the same norm in the spectral domain as in the spatial/temporal domain, i.e., the transformation is energy preserving. The noise is evenly spread over all spectral components, since the singular values of G are all 1. The Gaussian noise has a flat power spectral density for all unitary basis function sets independent of their nature. As a consequence, Parseval's theorem is true for all unitary bases. More general, Plancherel's theorem states,

$$\begin{aligned} \boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} &= (\mathsf{G}\, \boldsymbol{\alpha}_{\boldsymbol{x}})^{\mathrm{T}}(\mathsf{G}\, \boldsymbol{\alpha}_{\boldsymbol{y}}), \\ &= \boldsymbol{\alpha}_{\boldsymbol{x}}^{\mathrm{T}}\, \mathsf{G}^{\mathrm{T}}\, \mathsf{G}\boldsymbol{\alpha}_{\boldsymbol{y}}, \\ &= \boldsymbol{\alpha}_{\boldsymbol{x}}^{\mathrm{T}}\, \boldsymbol{\alpha}_{\boldsymbol{y}}. \end{aligned} \tag{7.35}$$

Parseval's theorem is a special case where $\boldsymbol{x} = \boldsymbol{y}$. This, together with the covariance propagation, allows the evaluation of a basis function's performance w.r.t. Gaussian noise. Consider the noise propagation from the spatial/temporal domain to the spectrum. The computation of the signal's spectrum $\tilde{\boldsymbol{\alpha}}$ with the unitary basis G^{T} is

$$\tilde{\boldsymbol{\alpha}} = \mathsf{G}^{\mathrm{T}}\tilde{\boldsymbol{y}}. \tag{7.36}$$

As shown in Section 5.4, given a linear transformation $\mathsf{L} \triangleq \mathsf{G}^{\mathrm{T}}$ and the input covariance matrix $\Lambda_{\tilde{\boldsymbol{y}}}$, the output covariance $\Lambda_{\tilde{\boldsymbol{\alpha}}} = \mathsf{L}\, \Lambda_{\tilde{\boldsymbol{y}}}\, \mathsf{L}^{\mathrm{T}}$ is given by

$$\Lambda_{\tilde{\boldsymbol{\alpha}}} = \mathsf{G}^{\mathrm{T}}\Lambda_{\tilde{\boldsymbol{y}}}\, \mathsf{G}. \tag{7.37}$$

If the signal $\tilde{\boldsymbol{y}}$ is perturbed by i.i.d. Gaussian noise, then the covariance $\Lambda_{\tilde{\boldsymbol{y}}}$ of $\tilde{\boldsymbol{y}}$ is

$$\Lambda_{\tilde{\boldsymbol{y}}} = \operatorname{diag}\left(\sigma_1^2 \ldots \sigma_m^2\right) = \sigma_{\tilde{\boldsymbol{y}}}^2 \mathsf{I} \tag{7.38}$$

and, therefore, the output covariance is simplified to

$$\Lambda_{\tilde{\boldsymbol{\alpha}}} = \sigma_{\tilde{\boldsymbol{y}}}^2 \, \mathsf{G}^{\mathrm{T}} \mathsf{G} = \sigma_{\tilde{\boldsymbol{y}}}^2 \, \mathsf{I}. \tag{7.39}$$

Consider the propagation from the perturbed input $\tilde{\boldsymbol{y}}$ to the output $\hat{\boldsymbol{y}}$ of a linear filter F,

$$\hat{\boldsymbol{y}} = \mathsf{F} \, \tilde{\boldsymbol{y}}, \tag{7.40}$$

where $\hat{\boldsymbol{y}}$ is the filtered signal. The output covariance $\Lambda_{\hat{\boldsymbol{y}}} = \mathsf{L} \, \Lambda_{\tilde{\boldsymbol{y}}} \, \mathsf{L}^{\mathrm{T}}$ with $\mathsf{L} \triangleq \mathsf{F}$ is given by

$$\Lambda_{\hat{\boldsymbol{y}}} = \mathsf{F} \, \Lambda_{\tilde{\boldsymbol{y}}} \, \mathsf{F}^{\mathrm{T}} \tag{7.41}$$

and, for i.i.d. noise $\Lambda_{\tilde{\boldsymbol{\alpha}}} = \sigma_{\tilde{\boldsymbol{y}}}^2 \, \mathsf{I}$, the equation becomes

$$\Lambda_{\hat{\boldsymbol{y}}} = \sigma_{\tilde{\boldsymbol{y}}}^2 \, \mathsf{F} \, \mathsf{F}^{\mathrm{T}}. \tag{7.42}$$

In the special case, where the filtering operation is performed with a unitary basis G, this transformation fulfills the characteristics of a generalized FIR filter,

$$\hat{\boldsymbol{y}} = \mathsf{G} \, \mathsf{G}^{\mathrm{T}} \, \tilde{\boldsymbol{y}} \quad \text{with} \quad \mathsf{F} = \mathsf{G} \, \mathsf{G}^{\mathrm{T}}, \tag{7.43}$$

which directly leads to

$$\begin{aligned} \Lambda_{\hat{\boldsymbol{y}}} &= \sigma_{\tilde{\boldsymbol{y}}}^2 \, \{\mathsf{G} \, \mathsf{G}^{\mathrm{T}}\} \, \{\mathsf{G} \, \mathsf{G}^{\mathrm{T}}\}, \\ &= \sigma_{\tilde{\boldsymbol{y}}}^2 \, \mathsf{G} \, \{\mathsf{G}^{\mathrm{T}} \mathsf{G}\} \mathsf{G}^{\mathrm{T}}, \\ &= \sigma_{\tilde{\boldsymbol{y}}}^2 \, \mathsf{G} \, \mathsf{G}^{\mathrm{T}}, \end{aligned} \tag{7.44}$$

where $\mathsf{G} \, \mathsf{G}^{\mathrm{T}}$ is a projection onto the orthogonal basis G. The energy preservation theorems are also valid for bivariate transformations,

$$\Omega = \mathsf{Y} \, \Sigma \, \mathsf{X}^{\mathrm{T}}, \tag{7.45}$$

where $\mathsf{X} \triangleq \mathsf{G}_{\mathsf{X}}$ and $\mathsf{Y} \triangleq \mathsf{G}_{\mathsf{Y}}$ are the DOP basis functions in x- and y-dimension respectively, Σ is the associated spectrum and Ω is the 2D-field data, e.g., an image. Computing the spectrum yields

$$\Sigma = \mathsf{Y}^{\mathrm{T}} \, \Omega \, \mathsf{X}, \tag{7.46}$$

since $\mathsf{X}^{+\mathrm{T}} = \mathsf{X}$ and $\mathsf{Y}^{+} = \mathsf{Y}^{\mathrm{T}}$ for unitary bases. The Frobenius norm is unitarily invariant, hence,

$$\|\Sigma\|_{\mathrm{F}} = \|\mathsf{Y}^{\mathrm{T}} \, \Omega \, \mathsf{X}\|_{\mathrm{F}} \; = \|\Omega\|_{\mathrm{F}}. \tag{7.47}$$

The energy in the spectrum Σ and the data Ω is equivalent.

7.6 Upper Error Bound

It is possible to determine an upper bound for the reconstruction error as a functions of the condition number $\kappa\left(\mathsf{B}\right) = \|\mathsf{B}\|\,\|\mathsf{B}^{+}\|$ of the basis function B [163]. The residual vector $\boldsymbol{r}$ is

$$\begin{aligned}\boldsymbol{r} &= \boldsymbol{y} - \hat{\boldsymbol{y}},\\ &= \boldsymbol{y} - \mathsf{B}\,\mathsf{B}^{+}\boldsymbol{y},\\ &= \left(\mathsf{I} - \mathsf{B}\,\mathsf{B}^{+}\right)\boldsymbol{y}.\end{aligned} \tag{7.48}$$

The matrix B is perturbed by the matrix of perturbations E,

$$\mathsf{B}(\epsilon) = \mathsf{B} + \epsilon\,\mathsf{E}, \tag{7.49}$$

where error ϵ is small. A Taylor (Maclaurin) series expansion of $\boldsymbol{r}(\epsilon)$ yields,

$$\boldsymbol{r}(\epsilon) = \boldsymbol{r} + \epsilon\,\boldsymbol{r}'(0) + \mathcal{O}(\epsilon^2). \tag{7.50}$$

The first order estimation is

$$\boldsymbol{r}(\epsilon) - \boldsymbol{r} \approx \epsilon\,\boldsymbol{r}'(0) \tag{7.51}$$

and the relative error in the residual is

$$\frac{\|\boldsymbol{r}(\epsilon) - \boldsymbol{r}\|}{\|\boldsymbol{r}\|} \approx \frac{\|\epsilon\,\boldsymbol{r}'(0)\|}{\|\boldsymbol{r}\|}. \tag{7.52}$$

According to [163], the upper bound for the error in B is then

$$\frac{\|\boldsymbol{r}(\epsilon) - \boldsymbol{r}\|}{\|\boldsymbol{r}\|} \leq 2\,|\epsilon|\,\frac{\|\mathsf{E}\|}{\|\mathsf{B}\|}\,\kappa\left(\mathsf{B}\right). \tag{7.53}$$

If a unitary basis $\mathsf{G} \triangleq \mathsf{B}$ is used with $\operatorname{cond}\left(\mathsf{G}\right) = 1$ and $\|\mathsf{G}\| = 1$, then the term simplifies to

$$\frac{\|\boldsymbol{r}(\epsilon) - \boldsymbol{r}\|}{\|\boldsymbol{r}\|} \leq 2\,|\epsilon|\,\|\mathsf{E}\|. \tag{7.54}$$

The upper error bound depends exclusively on the perturbations of E and not on the basis G.

8 | Regression Analysis

In general, regression analysis is a statistical process for the estimation of the relationships among a dependent and one or more independent variables. As stated in Chapter 5, this is of special importance during the calibration of measurement instruments, where the functional relation between the input and the output variable is sought, i.e., the forward problem. Regularization refers to the process of introducing contextual knowledge about the problem in order to solve an ill-posed inverse problem and/or to prevent under-/overfitting. The advantage of having an equation to describe the data is that analytic techniques can be applied to determine properties of the measurement data and in turn to deduce behavior of the physical process being observed.

Measurement applications deal with overdetermined and perturbed systems of equations. The system of linear equations is described as

$$\boldsymbol{y} = \mathsf{B}\,\boldsymbol{\alpha}, \tag{8.1}$$

where $\boldsymbol{y} \in \mathbb{R}^{(m \times 1)}$ is the data vector with m values, B is the basis function and $\boldsymbol{\alpha}$ is the coefficient vector. Most commonly, more measurements are available than theoretically necessary to perform curve fitting. The estimation $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{((d+1) \times 1)}$ of the coefficients $\boldsymbol{\alpha}$ is then computed as

$$\hat{\boldsymbol{\alpha}} = \mathsf{B}^{-}\boldsymbol{y}, \tag{8.2}$$

where $\mathsf{B}^{-} \in \mathbb{R}^{((d+1) \times m)}$ is a generalized inverse of the basis function B. Furthermore, the task is complicated by the fact, that the measurements of the data $\tilde{\boldsymbol{y}}$ are perturbed by noise $\boldsymbol{\epsilon}$, thus,

$$\tilde{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{\epsilon}. \tag{8.3}$$

However, as every measurement victim to noise, the nomenclature in this thesis is simplified by defining $\boldsymbol{y} \triangleq \tilde{\boldsymbol{y}}$ as already stated in Section 7.5. Basically, two types of regression are distinguishable:

1. **Global approximation** applies the basis functions to the full length of the support;
2. **Local approximation** applies the basis functions to a limited portion of the data.

Modified basis functions for functionally extending this concept are introduced in Chapter 9. Regardless which approach is chosen to model the data, regularization must be performed to avoid overfitting. Within this chapter, spectral regularization techniques are facilitated, because the geometric nature of the basis functions support this efficient and understandable approach. Furthermore, this approach is equivalent to the structure of a FIR filter.

8.1 Global Approximation

Given a set of m noisy data points $\boldsymbol{y}$ and the corresponding points $\boldsymbol{x}$, global approximation is performed by generating a set of basis functions B of the desired degree d at the nodes defined by $\boldsymbol{x}$, whereby polynomials of high degree are necessary to model the signal's details. Approximation is achieved by computing the projection onto the basis functions,

$$\hat{\boldsymbol{y}} = \mathsf{B}\,\mathsf{B}^{+}\boldsymbol{y}, \tag{8.4}$$

whereby B^{+} is the least squares minimum norm pseudoinverse. This approach is valid for the case where the Vandermonde design matrix $\mathsf{B} \triangleq \mathsf{V}$ is used as a basis,

$$\hat{\boldsymbol{y}} = \mathsf{V}\,\mathsf{V}^{+}\boldsymbol{y} \quad \text{with coefficients} \quad \hat{\boldsymbol{\beta}} = \mathsf{V}^{+}\boldsymbol{y}. \tag{8.5}$$

Using the discrete orthogonal basis $\mathsf{B} \triangleq \mathsf{G}$, this operation simplifies to

$$\hat{\boldsymbol{y}} = \mathsf{G}\,\mathsf{G}^{\mathrm{T}}\boldsymbol{y} \quad \text{with coefficients} \quad \hat{\boldsymbol{\alpha}} = \mathsf{G}^{\mathrm{T}}\boldsymbol{y}, \tag{8.6}$$

because the orthogonality condition $\mathsf{G}^{\mathrm{T}}\mathsf{G} = \mathsf{I}$ implies $\mathsf{G}^{\mathrm{T}} = \mathsf{G}^{+}$. Clearly, there is a direct relationship between the coefficients $\hat{\boldsymbol{\alpha}}$ of the Gram polynomials and the corresponding coefficients $\hat{\boldsymbol{\beta}}$ of the Vandermonde polynomials. The conversion is computed as

$$\hat{\boldsymbol{\beta}} = \mathsf{V}^{+}\mathsf{G}\,\boldsymbol{\alpha}. \tag{8.7}$$

However, if the Vandermonde matrix V has become degenerate it will not be possible to compute the corresponding geometric coefficients $\hat{\boldsymbol{\beta}}$ due to numerical errors in V^{+}. Regardless of which basis is chosen, the Runge phenomenon is problematic for global approximation with polynomials of high degree, i.e., regularization is required to solve this problem. The Runge phenomenon causes serious oscillations of the curve at the borders of the support and, therefore, extrapolation is precluded. If the measurement range is well defined, using Chebyshev nodes for $\boldsymbol{x}$ significantly reduces the Runge phenomenon, but suffers the same limitations as the conventional approach w.r.t. extrapolation. Depending on the application, local approximation with a limited support length l_s as presented in Section 8.3 is a more robust approach.

8.1.1 Uncertainty in Regression

The output vector $\boldsymbol{y} \in \mathbb{R}^{(m\times 1)}$ with m values and its coefficient vector $\boldsymbol{\alpha} \in \mathbb{R}^{((d+1)\times 1)}$ are related with the basis $\mathsf{B} \in \mathbb{R}^{(m\times(d+1))}$ of degree d,

$$\boldsymbol{y} = \mathsf{B}\,\boldsymbol{\alpha}. \tag{8.8}$$

If the basis $\mathsf{B} \in \mathbb{R}^{(m\times m)}$ is complete, i.e., $m = d + 1$, then a unique matrix inverse B^{-1} exists. In practical measurement applications, the number of measurements $m \gg d$, i.e., an overdetermined system of equations has to be solved. The basis $\mathsf{B} \in \mathbb{R}^{(m\times(d+1))}$ is an incomplete design matrix in this case; a suitable instance for the generalized inverse B^{-} must be sought, e.g., via QR or singular value decomposition. The most common approach is the use of the Moore-Penrose pseudoinverse B^{+} with

$$\mathsf{B}^{+} \triangleq \left(\mathsf{B}^{\mathrm{T}}\mathsf{B}\right)^{-1}\mathsf{B}^{\mathrm{T}}, \tag{8.9}$$

which is a maximum likelihood least squares estimator with minimum norm, i.e., minimal variance. The estimated coefficient vector $\hat{\boldsymbol{\alpha}}$ is then

$$\hat{\boldsymbol{\alpha}} = \mathsf{B}^{+}\boldsymbol{y} \tag{8.10}$$

with $\mathbb{E}[\hat{\boldsymbol{\alpha}}] = \boldsymbol{\alpha}$. Substituting this relation into Eqn. (8.8) yields the fitted vector $\hat{\boldsymbol{y}}$,

$$\hat{\boldsymbol{y}} = \mathsf{B}\,\hat{\boldsymbol{\alpha}} = \mathsf{B}\mathsf{B}^{+}\boldsymbol{y} \tag{8.11}$$

with $\mathbb{E}[\hat{\boldsymbol{y}}] = \boldsymbol{y}$. The residual vector $\boldsymbol{r}$ is the difference between the measured values $\boldsymbol{y}$ and the estimation $\hat{\boldsymbol{y}}$, i.e.,

$$\begin{aligned}\boldsymbol{r} &= \boldsymbol{y} - \hat{\boldsymbol{y}},\\ &= \boldsymbol{y} - \mathsf{B}\mathsf{B}^{+}\boldsymbol{y},\\ &= \left(\mathsf{I} - \mathsf{B}\mathsf{B}^{+}\right)\boldsymbol{y},\end{aligned} \tag{8.12}$$

where $(\mathsf{I} - \mathsf{B}\mathsf{B}^{+})$ is the projection onto the orthogonal complement. In order to detect and avoid systematic errors, the residual vector $\boldsymbol{r}$ must be tested against Gaussian distribution utilizing a Kolmogorov-Smirnov test and/or an Anderson-Darling test given a significance level α, see Fig. 8.1. The 2-norm of the residual vector $\|\boldsymbol{r}\|_2$ is a quality measure for the regression. The covariance $\Lambda_{\boldsymbol{y}}$ is the expectation value of the residual's projection,

$$\begin{aligned}\Lambda_{\boldsymbol{y}} &= \mathbb{E}\left[(\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}])\,(\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}])^{\mathrm{T}}\right],\\ &= \mathbb{E}\left[(\hat{\boldsymbol{y}} - \boldsymbol{y})\,(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}}\right],\\ &= \mathbb{E}\left[(\boldsymbol{y} - \hat{\boldsymbol{y}})\,(\boldsymbol{y} - \hat{\boldsymbol{y}})^{\mathrm{T}}\right],\\ &= \mathbb{E}\left[\boldsymbol{r}\boldsymbol{r}^{\mathrm{T}}\right].\end{aligned} \tag{8.13}$$

The projection matrix $\mathsf{P} \triangleq \mathsf{B}\mathsf{B}^{+}$ has the properties

1. P is a **linear operator**, i.e., $\mathsf{L} \triangleq \mathsf{P}$;

2. P is **symmetric**, i.e., $\mathsf{P} = \mathsf{P}^{\mathrm{T}}$;

3. P is **idempotent**, i.e., $\mathsf{P} = \mathsf{P}^{n}$ when n is an integer exponent;

4. if B is **unitary**, then $\mathsf{B} \triangleq \mathsf{G}$ with $\mathsf{G}^{+} = \mathsf{G}^{\mathrm{T}}$ and $\mathsf{P}^{\perp} = \mathsf{G}\mathsf{G}^{\mathrm{T}}$.

An idempotent function $y = f(x)$ does not change the output's value y if it is executed multiple times, i.e., $y = f(x) = f(f(y))$. This is also true for linear operations such as projections. The covariance of the estimation $\Lambda_{\hat{\boldsymbol{y}}} = \mathsf{L}\Lambda_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}}$ with $\mathsf{L} \triangleq \mathsf{P}$ is computed using the projection's linear behavior,

$$\begin{aligned}\Lambda_{\hat{\boldsymbol{y}}} &= \mathsf{P}\Lambda_{\boldsymbol{y}}\mathsf{P}^{\mathrm{T}} \quad \text{with} \quad \mathsf{P} \triangleq \mathsf{B}\mathsf{B}^{+},\\ &= \left(\mathsf{B}\mathsf{B}^{+}\right)\Lambda_{\boldsymbol{y}}\left(\mathsf{B}\mathsf{B}^{+}\right)^{\mathrm{T}}.\end{aligned} \tag{8.14}$$

In the case of i.i.d. noise, i.e., $\mathsf{\Lambda}_{\boldsymbol{y}} = \sigma_{\boldsymbol{y}}^2 \mathsf{I}$, it is derived

$$\begin{aligned}\mathsf{\Lambda}_{\hat{\boldsymbol{y}}} &= \sigma_{\boldsymbol{y}}^2 \left(\mathsf{B}\mathsf{B}^+\right)\left(\mathsf{B}\mathsf{B}^+\right)^{\mathrm{T}}, \\ &= \sigma_{\boldsymbol{y}}^2\, \mathsf{P}\mathsf{P}^{\mathrm{T}} \quad \text{with} \quad \mathsf{P} = \mathsf{P}^{\mathrm{T}}, \\ &= \sigma_{\boldsymbol{y}}^2\, \mathsf{P}^2 \quad\ \ \text{and} \quad \mathsf{P}^n = \mathsf{P}, \\ &= \sigma_{\boldsymbol{y}}^2\, \mathsf{P}, \\ &= \sigma_{\boldsymbol{y}}^2\, \mathsf{B}\mathsf{B}^+. \end{aligned} \tag{8.15}$$

The covariance $\mathsf{\Lambda}_{\hat{\boldsymbol{\alpha}}} = \mathsf{L}\mathsf{\Lambda}_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}}$ with $\mathsf{L} \triangleq \mathsf{B}^+$ for the coefficients $\hat{\boldsymbol{\alpha}}$ is computed in the same way,

$$\mathsf{\Lambda}_{\hat{\boldsymbol{\alpha}}} = \mathsf{B}^+\mathsf{\Lambda}_{\boldsymbol{y}}\mathsf{B}^{+\mathrm{T}}. \tag{8.16}$$

8.1.2 Filtering Operation

Global polynomial approximation can be interpreted as a filtering operation for the noisy data $\boldsymbol{y} \in \mathbb{R}^{(m\times 1)}$ with m points,

$$\hat{\boldsymbol{\alpha}} = \mathsf{F}\,\mathsf{G}^{\mathrm{T}}\boldsymbol{y}, \tag{8.17}$$

where $\mathsf{F} \in \mathbb{R}^{((d+1)\times(d+1))}$ is the filtering matrix and $\mathsf{G}^{\mathrm{T}} \in \mathbb{R}^{((d+1)\times m)}$ is the DOP basis function. The filtered signal $\hat{\boldsymbol{y}} \in \mathbb{R}^{(m\times 1)}$ is then formulated as

$$\hat{\boldsymbol{y}} = \mathsf{G}\,\mathsf{F}\,\mathsf{G}^{\mathrm{T}}\boldsymbol{y}. \tag{8.18}$$

If the filter is factorizable, it can be defined $\mathsf{F} \triangleq \mathsf{K}\,\mathsf{K}^{\mathrm{T}}$ with $\mathsf{L} \triangleq \mathsf{G}\,\mathsf{K}$, therefore,

$$\hat{\boldsymbol{y}} = \mathsf{B}\,\mathsf{K}\,\mathsf{K}^{\mathrm{T}}\,\mathsf{B}^{\mathrm{T}}\boldsymbol{y} = \mathsf{L}\,\mathsf{L}^{\mathrm{T}}\boldsymbol{y} \tag{8.19}$$

and $\mathsf{P} \triangleq \mathsf{L}\,\mathsf{L}^{\mathrm{T}}$ is a projection onto the filtering basis function. Given the projection $\mathsf{P} \in \mathbb{R}^{(m\times m)}$, each filtered point $\hat{y}_i$ in $\hat{\boldsymbol{y}}$ is a linear combination of all input values, i.e., the i^{th} row of the projection matrix $\mathsf{P}(i,:)$ can be regarded as the coefficients of a FIR filter for $\hat{y}_i$. This enables the direct computation of the frequency response. Knowing that the Gaussian input noise is evenly spread over the complete spectrum yields the noise gain $g_n = \|\mathsf{F}\|_{\mathrm{F}}^2$. This allows the time analytic computation of a polynomial filter's noise behavior. Using a unitary discrete basis G yields a polynomial preserving filter. In this manner, both low and bandpass filters can be implemented.

8.2 Confidence and Prediction Intervals

The fitted curve is an approximation to the true curve; as a result, the confidence interval of a fitted curve is a measure for the goodness-of-fit; an example is plotted in Fig. 8.2. The concept has been used for a-priori estimation of uncertainty in [72]. The prediction interval takes the perturbations of new measurement values into account and describes an upper bound of uncertainty. The discrete multiple regression model with m measurements and degree d is

$$\text{true:} \quad \boldsymbol{y} = \mathsf{B}\,\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{8.20}$$

$$\text{estimate:} \quad \hat{\boldsymbol{y}} = \mathsf{B}\,\hat{\boldsymbol{\alpha}}, \tag{8.21}$$

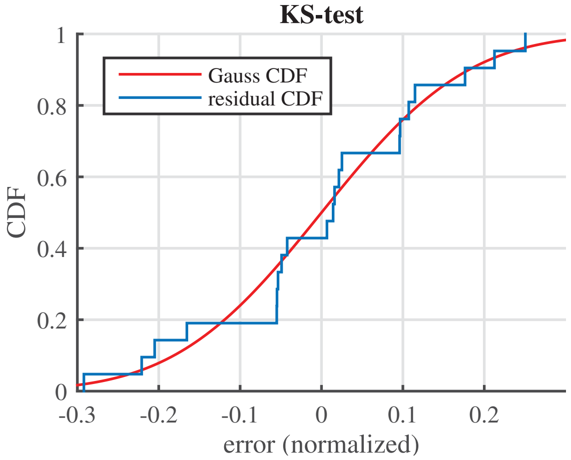


Fig. 8.1 The residual vector $\boldsymbol{r} = (\mathsf{I} - \mathsf{BB}^{+})\,\boldsymbol{y}$ must be tested for its Gaussian distribution using a Kolmogorov-Smirnov (KS) test using a given significance level $\alpha = 95\%$ [17].

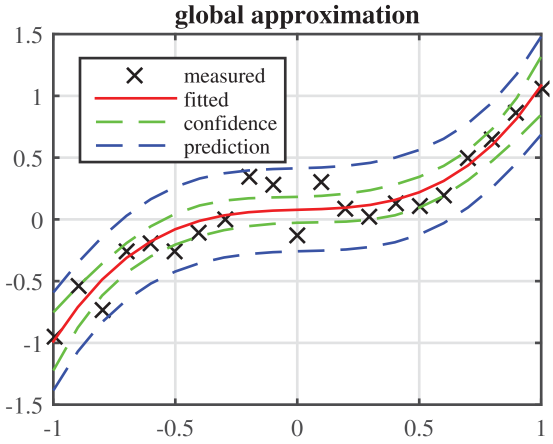


Fig. 8.2 The global approximation curve $\hat{\boldsymbol{y}}$ (red) is computed using the measured values $\boldsymbol{y}$ (black ×). The confidence (green) and prediction (blue) intervals depend on the standard deviation $s_{\boldsymbol{y}}$.

with the true coefficient vector $\boldsymbol{\alpha} \in \mathbb{R}^{((d+1)\times 1)}$ and the corresponding estimation $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{((d+1)\times 1)}$, the measured solution vector $\boldsymbol{y} \in \mathbb{R}^{(m\times 1)}$ and its estimation $\hat{\boldsymbol{y}} \in \mathbb{R}^{(m\times 1)}$, the error vector $\boldsymbol{\epsilon} \in \mathbb{R}^{(m\times 1)}$, and the polynomial basis function set $\mathsf{B} \in \mathbb{R}^{(m\times (d+1))}$.

It is assumed, that the error $\boldsymbol{\epsilon}$ is i.i.d.; hence, the homoscedasticity condition is fulfilled. The expectation $\mathbb{E}[\boldsymbol{\epsilon}]$ and covariance $\Lambda_{\boldsymbol{\epsilon}}$ of the error ϵ are

$$\mathbb{E}[\boldsymbol{\epsilon}] = \boldsymbol{0} \quad \text{and} \quad \Lambda_{\boldsymbol{\epsilon}} = \sigma_{\boldsymbol{\epsilon}}^2 \mathsf{I}, \tag{8.22}$$

because

$$\Lambda_{\boldsymbol{\epsilon}} = \begin{bmatrix} \mathrm{cov}[\epsilon_1, \epsilon_1] & \mathrm{cov}[\epsilon_1, \epsilon_2] & \dots & \mathrm{cov}[\epsilon_1, \epsilon_m] \\ \mathrm{cov}[\epsilon_2, \epsilon_1] & \mathrm{cov}[\epsilon_2, \epsilon_2] & \dots & \mathrm{cov}[\epsilon_2, \epsilon_m] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{cov}[\epsilon_m, \epsilon_1] & \mathrm{cov}[\epsilon_m, \epsilon_2] & \dots & \mathrm{cov}[\epsilon_m, \epsilon_m] \end{bmatrix} = \begin{bmatrix} \sigma_{\epsilon_1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{\epsilon_2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{\epsilon_m}^2 \end{bmatrix}, \tag{8.23}$$

whereby $\mathrm{cov}[\epsilon_i, \epsilon_j] = 0$ and $\mathrm{cov}[\epsilon_i, \epsilon_i] = \sigma_{\epsilon_i}^2$. The covariance $\Lambda_{\boldsymbol{\epsilon}}$ of $\boldsymbol{\epsilon}$ is equivalent to the covariance $\Lambda_{\boldsymbol{y}}$ of $\boldsymbol{y}$, i.e.,

$$\Lambda_{\boldsymbol{y}} = \Lambda_{\boldsymbol{\epsilon}} = \sigma_{\boldsymbol{\epsilon}}^2 \mathsf{I} \quad \text{and, therefore,} \quad \sigma_{\boldsymbol{y}}^2 = \sigma_{\boldsymbol{\epsilon}}^2. \tag{8.24}$$

The unbiased multivariate estimator for the populations variance $\sigma_{\boldsymbol{y}}^2$ with $\nu = m - d - 1$ d.f. is

$$s_{\boldsymbol{y}}^2 = \frac{1}{\nu}(\boldsymbol{y} - \hat{\boldsymbol{y}})^{\mathrm{T}}(\boldsymbol{y} - \hat{\boldsymbol{y}}), \tag{8.25}$$

which is equivalent to the MSE goodness-fit-parameter. Besides the MSE, there are several other relevant regression parameters, see Table 8.1. These parameters describe the overall quality of the global approximation. In the following subsections, the confidence and prediction intervals using the Vandermonde basis V as well as the DOP basis G are derived and compared in terms of numerical stability.

	parameter name	equation	comment
SSE	sum of squared residuals	$(\boldsymbol{y}-\hat{\boldsymbol{y}})^{\mathrm{T}}(\boldsymbol{y}-\hat{\boldsymbol{y}})$	
SSR	regression sum of squares	$(\hat{\boldsymbol{y}}-\bar{y})^{\mathrm{T}}(\hat{\boldsymbol{y}}-\bar{y})$	
SST	total sum of squares	$(\boldsymbol{y}-\bar{y})^{\mathrm{T}}(\boldsymbol{y}-\bar{y})$	SSE + SSR
R^2	coefficient of determination	SSR / SST	1 - SSE / SST
DF	degrees of freedom	$m-(d+1)$	also d.f. or ν
$\bar{\mathrm{R}}^2$	adjusted R^2	1 - (SSE·(m-1)) / (SST·DF)	quality of regression
MSE	mean square error	SSE / DF	$s_{\boldsymbol{y}}^2 \triangleq$ MSE
RMSE	root mean square error	sqrt(MSE)	$s_{\boldsymbol{y}} \triangleq$ RMSE

Tab. 8.1 Statistical goodness-of-fit parameters according to literature [18, 210]; which are also implemented in MATLAB's Curve Fitting Toolbox [136]. Most commonly, these acronyms are used instead of shorter variables names.

8.2.1 Vandermonde Basis

The common basis set used for polynomial regression in literature and applications is the Vandermonde design matrix. The system of linear equations is

$$\boldsymbol{y} = \mathsf{V}\,\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{8.26}$$

where $\boldsymbol{y} \in \mathbb{R}^{(m\times 1)}$ is the solution vector, $\mathsf{V} \in \mathbb{R}^{(m\times(d+1))}$ is the Vandermonde design matrix, $\boldsymbol{\alpha} \in \mathbb{R}^{((d+1)\times 1)}$ is the coefficient vector and $\boldsymbol{\epsilon} \in \mathbb{R}^{(m\times 1)}$ is the error vector with $\mathbb{E}[\boldsymbol{\epsilon}] = \boldsymbol{0}$. The basis is complete when $m = d+1$. This approach is analytically fine, but numerically unstable if polynomial degrees of $d > 7$ are used [65, 176]. The state-of-the art approach to estimate the coefficient vector $\hat{\boldsymbol{\alpha}}$ of the regression is the use of the Moore-Penrose pseudoinverse $\mathsf{V}^+ \triangleq (\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}\mathsf{V}^{\mathrm{T}}$,

$$\hat{\boldsymbol{\alpha}} = (\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}\mathsf{V}^{\mathrm{T}}\,\boldsymbol{y} = \mathsf{V}^+\boldsymbol{y} \quad \text{with} \quad \mathbb{E}[\hat{\boldsymbol{\alpha}}] = \boldsymbol{\alpha}, \tag{8.27}$$

which implements a maximum likelihood estimator [95] for $\boldsymbol{\alpha}$. It is defined, that $\mathsf{U} \triangleq (\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}$ while $(\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1} = (\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-\mathrm{T}}$. The estimator $\hat{\boldsymbol{y}}$ for the solution vector $\boldsymbol{y}$ is then

$$\hat{\boldsymbol{y}} = \mathsf{V}\hat{\boldsymbol{\alpha}} = \mathsf{V}\mathsf{V}^+\boldsymbol{y}. \tag{8.28}$$

As a result, the covariance $\Lambda_{\hat{\boldsymbol{\alpha}}} = \mathsf{L}\Lambda_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}}$ with $\mathsf{L} \triangleq (\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}\mathsf{V}^{\mathrm{T}}$ of the coefficients estimator $\hat{\boldsymbol{\alpha}}$ is

$$\begin{aligned}
\Lambda_{\hat{\boldsymbol{\alpha}}} &= ((\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}\mathsf{V}^{\mathrm{T}})\,\Lambda_{\boldsymbol{y}}\,((\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}\mathsf{V}^{\mathrm{T}})^{\mathrm{T}} \quad \text{and} \quad \Lambda_{\boldsymbol{y}} = \sigma_{\boldsymbol{y}}^2\mathsf{I},\\
&= \sigma_{\boldsymbol{y}}^2\,(\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}(\mathsf{V}^{\mathrm{T}}\mathsf{V})(\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1},\\
&= \sigma_{\boldsymbol{y}}^2\,(\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1} \quad \text{and} \quad \mathsf{U} \triangleq (\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1},\\
&= \sigma_{\boldsymbol{y}}^2\,\mathsf{U},
\end{aligned} \tag{8.29}$$

while the covariance matrix $\mathsf{\Lambda}_{\hat{\boldsymbol{y}}} = \mathsf{L}^{\mathrm{T}}\mathsf{\Lambda}_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}}$ with $\mathsf{L} \triangleq \mathsf{V}\mathsf{V}^{+} = \mathsf{V}(\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}\mathsf{V}^{\mathrm{T}}$ for the estimated solution $\hat{\boldsymbol{y}}$ is then formulated as

$$\begin{aligned}\mathsf{\Lambda}_{\hat{\boldsymbol{y}}} &= (\mathsf{V}(\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}\mathsf{V}^{\mathrm{T}})\,\mathsf{\Lambda}_{\boldsymbol{y}}\,(\mathsf{V}(\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}\mathsf{V}^{\mathrm{T}})^{\mathrm{T}} \quad \text{and} \quad \mathsf{\Lambda}_{\boldsymbol{y}} = \sigma_{\boldsymbol{y}}^{2}\mathsf{I},\\ &= \sigma_{\boldsymbol{y}}^{2}\,\mathsf{V}(\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}(\mathsf{V}^{\mathrm{T}}\mathsf{V})(\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}\mathsf{V}^{\mathrm{T}},\\ &= \sigma_{\boldsymbol{y}}^{2}\,\mathsf{V}(\mathsf{V}^{\mathrm{T}}\mathsf{V})^{-1}\mathsf{V}^{\mathrm{T}},\\ &= \sigma_{\boldsymbol{y}}^{2}\,\mathsf{V}\mathsf{V}^{+}, \end{aligned} \tag{8.30}$$

which is equivalent to $\mathsf{\Lambda}_{\hat{\boldsymbol{y}}} = \mathsf{L}\mathsf{\Lambda}_{\hat{\boldsymbol{\alpha}}}\mathsf{L}^{\mathrm{T}}$ with $\mathsf{L} \triangleq \mathsf{V}$, thus,

$$\mathsf{\Lambda}_{\hat{\boldsymbol{y}}} = \mathsf{V}\mathsf{\Lambda}_{\hat{\boldsymbol{\alpha}}}\mathsf{V}^{\mathrm{T}}. \tag{8.31}$$

(1) Simultaneous Confidence Intervals for the Coefficients

The relation between all $(d+1)$ coefficients is considered when computing the simultaneous confidence intervals. Therefore, a $\chi^2_{(d+1)}$ goodness-of-fit test is carried out,

$$\begin{aligned}\chi^2_{(d+1)} &\le (\hat{\boldsymbol{\alpha}} - \mathbb{E}[\hat{\boldsymbol{\alpha}}])^{\mathrm{T}}\mathsf{\Lambda}_{\hat{\boldsymbol{\alpha}}}{}^{-1}(\hat{\boldsymbol{\alpha}} - \mathbb{E}[\hat{\boldsymbol{\alpha}}]),\\ &\le (\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha})^{\mathrm{T}}(\sigma_{\boldsymbol{y}}^{2}\,\mathsf{U})^{-1}(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}).\end{aligned} \tag{8.32}$$

Estimating $\sigma_{\boldsymbol{y}}^2$ with $s_{\boldsymbol{y}}^2$ and approximating $\chi^2_{(d+1)}$ with $(d+1)\,\mathcal{F}_{(d+1,\nu)}$ yields

$$s_{\boldsymbol{y}}^2\,(d+1)\,\mathcal{F}_{(d+1,\nu)} \le (\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha})^{\mathrm{T}}\mathsf{U}^{-1}(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}), \tag{8.33}$$

which described a hyper-ellipsoid with $(d+1)$ dimensions.

(2) Non-Simultaneous Confidence Intervals for the Coefficients

When computing the non-simultaneous intervals, each coefficient's confidence is computed individually; hence, only 1 d.f. is required for the $\chi^2_{(1)}$ goodness-of-fit test. Therefore, $\chi^2_{(1)}$ can be approximated with $1\,\mathcal{F}_{(1,\nu)}$. Selecting one diagonal element of U via $\boldsymbol{u} = \operatorname{diag}(\mathsf{U})$ and $u_i = \boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{u}$ enables the computation of a single coefficient α_i from the coefficient vector $\boldsymbol{\alpha}$,

$$\begin{aligned}(\alpha_i - \hat{\alpha}_i)^2 &\le \mathcal{F}_{(1,\nu)}\,s_{\boldsymbol{y}}^2\,u_i,\\ (\alpha_i - \hat{\alpha}_i)^2 &\le t_{(\nu)}^2\,s_{\boldsymbol{y}}^2\,u_i,\\ |\alpha_i - \hat{\alpha}_i| &\le t_{(\nu)}\,s_{\boldsymbol{y}}\,\sqrt{u_i},\\ \alpha_i &\le \hat{\alpha}_i \pm t_{(\nu)}\,s_{\boldsymbol{y}}\,\sqrt{u_i}.\end{aligned} \tag{8.34}$$

Parallel computing yields the non-simultaneous confidence intervals for all coefficients,

$$\boldsymbol{\alpha} \le \hat{\boldsymbol{\alpha}} \pm t_{(\nu)}\,s_{\boldsymbol{y}}\,\sqrt{\operatorname{diag}(\mathsf{U})}. \tag{8.35}$$

(3) Non-Simultaneous Confidence Intervals for the Regression

The whole regressions is computed via $\hat{\boldsymbol{y}} = \mathsf{V}\hat{\boldsymbol{\alpha}}$. However, each coefficient's confidence is still computed separately; therefore, the χ^2-test has only 1 d.f.,

$$\begin{aligned}\chi^2_{(1)} &\leq (\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}])^{\mathrm{T}} \mathsf{\Lambda}_{\hat{\boldsymbol{y}}}{}^{-1} (\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}]),\\ &\leq (\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}} \left(\mathsf{V}\mathsf{\Lambda}_{\hat{\boldsymbol{\alpha}}}\mathsf{V}^{\mathrm{T}}\right)^{-1} (\hat{\boldsymbol{y}} - \boldsymbol{y}),\\ &\leq (\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}} \left(\mathsf{V}(\sigma^2_{\boldsymbol{y}}\mathsf{U})\mathsf{V}^{\mathrm{T}}\right)^{-1} (\hat{\boldsymbol{y}} - \boldsymbol{y}).\end{aligned} \tag{8.36}$$

Computing only the i^{th} point by selecting the row vector $\boldsymbol{v}_i^{\mathrm{T}} = \boldsymbol{e}_i^{\mathrm{T}}\mathsf{V}$ yields

$$\chi^2_{(1)} \leq (\hat{y}_i - y_i)^2 \left(\sigma^2_{\boldsymbol{y}}\, \boldsymbol{v}_i^{\mathrm{T}}\mathsf{U}\boldsymbol{v}_i\right)^{-1}. \tag{8.37}$$

Approximation of the χ^2 distribution with the Student-t distribution simplifies the terms to

$$\begin{aligned}(\hat{y}_i - y_i)^2 &\leq \chi^2_{(1)}(\sigma^2_{\boldsymbol{y}}\, \boldsymbol{v}_i^{\mathrm{T}}\mathsf{U}\boldsymbol{v}_i),\\ &\leq \mathcal{F}_{(1,\nu)}(s^2_{\boldsymbol{y}}\, \boldsymbol{v}_i^{\mathrm{T}}\mathsf{U}\boldsymbol{v}_i),\\ &\leq t^2_{(\nu)}(s^2_{\boldsymbol{y}}\, \boldsymbol{v}_i^{\mathrm{T}}\mathsf{U}\boldsymbol{v}_i).\end{aligned} \tag{8.38}$$

Computation of the absolute value delivers

$$|\hat{y}_i - y_i| \leq \pm\sqrt{t^2_{(\nu)}(s^2_{\boldsymbol{y}}\, \boldsymbol{v}_i^{\mathrm{T}}\mathsf{U}\boldsymbol{v}_i)}. \tag{8.39}$$

Consequently, the confidence of one regression point is

$$y_i \leq \hat{y}_i \pm t_{(\nu)}\, s_{\boldsymbol{y}} \sqrt{\boldsymbol{v}_i^{\mathrm{T}}\mathsf{U}\boldsymbol{v}_i}. \tag{8.40}$$

Parallel computing yields the non-simultaneous confidence intervals for all points of the regression,

$$\boldsymbol{y} \leq \hat{\boldsymbol{y}} \pm t_{(\nu)}\, s_{\boldsymbol{y}} \sqrt{\operatorname{diag}\left(\mathsf{V}\mathsf{U}\mathsf{V}^{\mathrm{T}}\right)}. \tag{8.41}$$

(4) Non-Simultaneous Prediction Intervals for the Regression

For the estimation of the predicted value $\tilde{y}_i$ of an individual function value y_i, the error's variance for one observation must also be considered, i.e., the variance $\sigma^2_{\boldsymbol{y}}$ is added to the term,

$$\chi^2_{(1)} \leq (\hat{y}_i - \tilde{y}_i)^2 \left(\sigma^2_{\boldsymbol{y}}\, \boldsymbol{v}_i^{\mathrm{T}}\mathsf{U}\boldsymbol{v}_i + \sigma^2_{\boldsymbol{y}}\right)^{-1}. \tag{8.42}$$

Performing the same steps as in the previous section delivers

$$(\hat{y}_i - \tilde{y}_i)^2 \leq t^2_{(\nu)}(s^2_{\boldsymbol{y}}\, \boldsymbol{v}_i^{\mathrm{T}}\mathsf{U}\boldsymbol{v}_i + s^2_{\boldsymbol{y}}). \tag{8.43}$$

Consequently, the prediction of one regression point is

$$\tilde{y}_i = \hat{y}_i \pm t_{(\nu)}\, s_{\boldsymbol{y}} \sqrt{\boldsymbol{v}_i^{\mathrm{T}}\mathsf{U}\boldsymbol{v}_i + 1}. \tag{8.44}$$

Parallel computing yields the non-simultaneous prediction intervals for all points of the regression,

$$\tilde{\boldsymbol{y}} = \hat{\boldsymbol{y}} \pm t_{(\nu)}\, s_{\boldsymbol{y}} \sqrt{\operatorname{diag}\left(\mathsf{V}\mathsf{U}\mathsf{V}^{\mathrm{T}}\right) + \boldsymbol{1}}. \tag{8.45}$$

8.2.2 Discrete Orthogonal Basis

A DOP basis $\mathsf{G} \in \mathbb{R}^{(m\times(d+1))}$ of degree d is composed of a sum of $(d+1)$ uncorrelated monomials. Orthogonality ensures minimal information redundancy and ensures numerical stability. By definition [5, 113],

$$\mathsf{G}^{\mathrm{T}}\mathsf{G} = \mathsf{I} \quad \text{and} \quad \mathsf{G}^{+} = \mathsf{G}^{\mathrm{T}}. \tag{8.46}$$

For applications utilizing regression, the degree d of the basis is $d < m - 1$, when m is the number of sampled points. The d.f. are then $\nu = m - d - 1$. Hence, the basis G is incomplete and, as a consequence, $\mathsf{G}\mathsf{G}^{\mathrm{T}} \neq \mathsf{I}$. The solution vector $\boldsymbol{y}$ is

$$\boldsymbol{y} = \mathsf{G}\,\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{8.47}$$

where $\boldsymbol{\alpha}$ is the coefficient vector and $\boldsymbol{\epsilon}$ is the error vector with $\mathbb{E}[\boldsymbol{\epsilon}] = \boldsymbol{0}$ and $\Lambda_{\boldsymbol{\epsilon}} = \sigma_{\boldsymbol{\epsilon}}^2\mathsf{I}$. Thanks to the orthogonality condition, the coefficients $\boldsymbol{\alpha}$ can effectively estimated via

$$\hat{\boldsymbol{\alpha}} = \mathsf{G}^{\mathrm{T}}\boldsymbol{y} \quad \text{with} \quad \mathbb{E}[\hat{\boldsymbol{\alpha}}] = \boldsymbol{\alpha}. \tag{8.48}$$

The estimator $\hat{\boldsymbol{y}}$ for the solution vector $\boldsymbol{y}$ is then

$$\hat{\boldsymbol{y}} = \mathsf{G}\,\hat{\boldsymbol{\alpha}} = \mathsf{G}\mathsf{G}^{\mathrm{T}}\boldsymbol{y}. \tag{8.49}$$

The covariance matrix $\Lambda_{\hat{\boldsymbol{\alpha}}} = \mathsf{L}\Lambda_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}}$ with $\mathsf{L}^{\mathrm{T}} \triangleq \mathsf{G}^{\mathrm{T}}$ for the estimated coefficients $\hat{\boldsymbol{\alpha}}$ is

$$\begin{aligned} \Lambda_{\hat{\boldsymbol{\alpha}}} &= \mathsf{G}^{\mathrm{T}}\Lambda_{\boldsymbol{y}}\mathsf{G} \quad \text{and} \quad \Lambda_{\boldsymbol{y}} = \sigma_{\boldsymbol{y}}^2\mathsf{I}, \\ &= \mathsf{G}^{\mathrm{T}}(\sigma_{\boldsymbol{y}}^2\mathsf{I})\,\mathsf{G}, \\ &= \sigma_{\boldsymbol{y}}^2\,\mathsf{G}^{\mathrm{T}}\mathsf{G} \quad \text{and} \quad \mathsf{G}^{\mathrm{T}}\mathsf{G} = \mathsf{I}, \\ &= \sigma_{\boldsymbol{y}}^2\mathsf{I}. \end{aligned} \tag{8.50}$$

The covariance matrix $\Lambda_{\hat{\boldsymbol{y}}} = \mathsf{L}\Lambda_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}}$ with $\mathsf{L} \triangleq \mathsf{G}\mathsf{G}^{\mathrm{T}}$ for the estimated solution $\hat{\boldsymbol{y}}$ is

$$\begin{aligned} \Lambda_{\hat{\boldsymbol{y}}} &= (\mathsf{G}\mathsf{G}^{\mathrm{T}})\Lambda_{\boldsymbol{y}}(\mathsf{G}\mathsf{G}^{\mathrm{T}})^{\mathrm{T}}, \\ &= \mathsf{G}\mathsf{G}^{\mathrm{T}}\,\Lambda_{\boldsymbol{y}}\,\mathsf{G}^{\mathrm{T}}\mathsf{G} \quad \text{and} \quad \mathsf{G}^{\mathrm{T}}\mathsf{G} = \mathsf{I}, \\ &= \mathsf{G}\mathsf{G}^{\mathrm{T}}\,\Lambda_{\boldsymbol{y}} \quad \text{and} \quad \Lambda_{\boldsymbol{y}} = \sigma_{\boldsymbol{y}}^2\mathsf{I}, \\ &= \sigma_{\boldsymbol{y}}^2\mathsf{G}\mathsf{G}^{\mathrm{T}}, \end{aligned} \tag{8.51}$$

which is equivalent to $\Lambda_{\hat{\boldsymbol{y}}} = \mathsf{L}\Lambda_{\hat{\boldsymbol{\alpha}}}\mathsf{L}^{\mathrm{T}}$ with $\mathsf{L} \triangleq \mathsf{G}$,

$$\Lambda_{\hat{\boldsymbol{y}}} = \mathsf{G}\Lambda_{\hat{\boldsymbol{\alpha}}}\mathsf{G}^{\mathrm{T}}. \tag{8.52}$$

(1) Simultaneous Confidence Intervals for the Coefficients

The relation between all $(d+1)$ coefficients is considered when computing the simultaneous confidence intervals. Therefore, a $\chi^2_{(d+1)}$ goodness-of-fit test is carried out,

$$\begin{aligned} \chi^2_{(d+1)} &\leq (\hat{\boldsymbol{\alpha}} - \mathbb{E}[\hat{\boldsymbol{\alpha}}])^{\mathrm{T}}\Lambda_{\hat{\boldsymbol{\alpha}}}{}^{-1}(\hat{\boldsymbol{\alpha}} - \mathbb{E}[\hat{\boldsymbol{\alpha}}]), \\ &\leq (\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha})^{\mathrm{T}}(\sigma_{\boldsymbol{y}}^2\mathsf{I})^{-1}(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}). \end{aligned} \tag{8.53}$$

Estimating $\sigma_{\boldsymbol{y}}^2$ with $s_{\boldsymbol{y}}^2$ and approximating $\chi^2_{(d+1)}$ with $(d+1)\,\mathcal{F}_{(d+1,\nu)}$ yields

$$s_{\boldsymbol{y}}^2\,(d+1)\,\mathcal{F}_{(d+1,\nu)} \leq (\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha})^{\mathrm{T}}(\hat{\boldsymbol{\alpha}}-\boldsymbol{\alpha}), \tag{8.54}$$

which described a hyper-ellipsoid with $(d+1)$ dimensions. The solution of this equation is an inverse problem; however, thanks to the orthogonality condition, the covariance is $\mathsf{\Lambda}_{\hat{\boldsymbol{\alpha}}} = \sigma_{\boldsymbol{y}}^2\mathsf{I}$. This reduces the result of the equation to a simple scalar product.

(2) Non-Simultaneous Confidence Intervals for the Coefficients

When computing the non-simultaneous intervals, each coefficient's confidence is computed individually; thus, only 1 d.f. is required for the $\chi^2_{(1)}$ goodness-of-fit test. Therefore, $\chi^2_{(1)}$ can be approximated with $1\,\mathcal{F}_{(1,\nu)}$,

$$\begin{aligned}(\hat{\alpha}_i-\alpha_i)^2 &\leq \mathcal{F}_{(1,\nu)}\,s_{\boldsymbol{y}}^2,\\ (\hat{\alpha}_i-\alpha_i)^2 &\leq t^2_{(\nu)}\,s_{\boldsymbol{y}}^2,\\ |\hat{\alpha}_i-\alpha_i| &\leq t_{(\nu)}\,s_{\boldsymbol{y}},\\ \alpha_i &\leq \hat{\alpha}_i \pm t_{(\nu)}\,s_{\boldsymbol{y}}.\end{aligned} \tag{8.55}$$

Parallel computing yield the non-simultaneous confidence for all coefficients,

$$\boldsymbol{\alpha} \leq \hat{\boldsymbol{\alpha}} \pm t_{(\nu)}\,s_{\boldsymbol{y}}\,\boldsymbol{1}. \tag{8.56}$$

(3) Non-Simultaneous Confidence Intervals for the Regression

The whole regression is computed via $\hat{\boldsymbol{y}} = \mathsf{G}\,\hat{\boldsymbol{\alpha}}$. However, each coefficient's confidence is still computed separately; therefore, the χ^2-test has only 1 d.f.,

$$\begin{aligned}\chi^2_{(1)} &\leq (\hat{\boldsymbol{y}}-\mathbb{E}[\hat{\boldsymbol{y}}])^{\mathrm{T}}\mathsf{\Lambda}_{\hat{\boldsymbol{y}}}{}^{-1}(\hat{\boldsymbol{y}}-\mathbb{E}[\hat{\boldsymbol{y}}]),\\ &\leq (\hat{\boldsymbol{y}}-\boldsymbol{y})^{\mathrm{T}}(\mathsf{G}\mathsf{\Lambda}_{\hat{\boldsymbol{\alpha}}}\mathsf{G}^{\mathrm{T}})^{-1}(\hat{\boldsymbol{y}}-\boldsymbol{y}),\\ &\leq (\hat{\boldsymbol{y}}-\boldsymbol{y})^{\mathrm{T}}(\mathsf{G}(\sigma_{\boldsymbol{y}}^2\mathsf{I})\mathsf{G}^{\mathrm{T}})^{-1}(\hat{\boldsymbol{y}}-\boldsymbol{y}).\end{aligned} \tag{8.57}$$

Computing only the i^{th} point by selecting the row vector $\boldsymbol{g}_i^{\mathrm{T}} = \boldsymbol{e}_i^{\mathrm{T}}\mathsf{G}$ yields

$$\chi^2_{(1)} \leq (\hat{y}_i-y_i)^2(\sigma_{\boldsymbol{y}}^2\,\boldsymbol{g}_i^{\mathrm{T}}\boldsymbol{g}_i).^{-1} \tag{8.58}$$

Note, that $\boldsymbol{g}_i^{\mathrm{T}}\boldsymbol{g}_i \neq 1$, because G is an incomplete basis. Approximation of the χ^2 distribution with the Student-t distribution simplifies the terms to

$$\begin{aligned}(\hat{y}_i-y_i)^2 &\leq \chi^2_{(1)}(\sigma_{\boldsymbol{y}}^2\,\boldsymbol{g}_i^{\mathrm{T}}\boldsymbol{g}_i),\\ &\leq \mathcal{F}_{(1,\nu)}(s_{\boldsymbol{y}}^2\,\boldsymbol{g}_i^{\mathrm{T}}\boldsymbol{g}_i),\\ &\leq t^2_{(\nu)}(s_{\boldsymbol{y}}^2\,\boldsymbol{g}_i^{\mathrm{T}}\boldsymbol{g}_i).\end{aligned} \tag{8.59}$$

Computing the absolute value delivers

$$|\hat{y}_i-y_i| \leq \pm\sqrt{t^2_{(\nu)}(s_{\boldsymbol{y}}^2\,\boldsymbol{g}_i^{\mathrm{T}}\boldsymbol{g}_i)}. \tag{8.60}$$

Consequently, the confidence of one regression point is

$$y_i \leq \hat{y}_i \pm t_{(\nu)}\, s_{\boldsymbol{y}} \sqrt{\boldsymbol{g}_i^{\mathrm{T}} \boldsymbol{g}_i}. \tag{8.61}$$

Parallel computing yields the non-simultaneous confidence intervals for all points of the regression,

$$\boldsymbol{y} \leq \hat{\boldsymbol{y}} \pm t_{(\nu)}\, s_{\boldsymbol{y}} \sqrt{\operatorname{diag}(\mathsf{G}\mathsf{G}^{\mathrm{T}})}. \tag{8.62}$$

(4) Non-Simultaneous Prediction Intervals for the Regression

For the estimation of the predicted value $\tilde{y}_i$ of an individual function value y_i, the error's variance for one observation must also be considered, i.e., the variance $\sigma_{\boldsymbol{y}}^2$ is added to the term,

$$\chi_{(1)}^2 \leq (\hat{y}_i - \tilde{y}_i)^2 (\sigma_{\boldsymbol{y}}^2\, \boldsymbol{g}_i^{\mathrm{T}} \boldsymbol{g}_i + \sigma_{\boldsymbol{y}}^2)^{-1}. \tag{8.63}$$

Performing the same steps as in the previous section delivers

$$(\hat{y}_i - \tilde{y}_i)^2 \leq t_{(\nu)}^2 (s_{\boldsymbol{y}}^2\, \boldsymbol{g}_i^{\mathrm{T}} \boldsymbol{g}_i + s_{\boldsymbol{y}}^2). \tag{8.64}$$

Consequently, the prediction of one regression point is

$$\tilde{y}_i = \hat{y}_i \pm t_{(\nu)}\, s_{\boldsymbol{y}} \sqrt{\boldsymbol{g}_i^{\mathrm{T}} \boldsymbol{g}_i + 1}. \tag{8.65}$$

Parallel computing yields the non-simultaneous prediction intervals for all points of the regression,

$$\tilde{\boldsymbol{y}} = \hat{\boldsymbol{y}} \pm t_{(\nu)}\, s_{\boldsymbol{y}} \sqrt{\operatorname{diag}(\mathsf{G}\mathsf{G}^{\mathrm{T}}) + \boldsymbol{1}}. \tag{8.66}$$

8.2.3 Comparison of Basis Functions

The computation of the confidence and prediction intervals demonstrate the superiority of the proposed DOP basis over the state-of-the-art Vandermonde basis. The DOP basis is numerically stable for degrees $d < 1000$, see also Chapter 7, while the Vandermonde basis collapses when a degree of $d > 7$ is used [65]. The MATLAB code in Lst. 8.1 shows how the non-simultaneous confidence and prediction intervals are computed for both bases. The code for the Vandermonde basis is equivalent to the implementation of MATLAB's Curve Fitting Toolbox [136].

```
%% (1) Vandermonde basis
% -------------------------------------------------------------
yv = V * (V\y); % = V * pinv(V) * y;
sv = sqrt( 1/df * (y-yv)^T * (y-yv) );
%
%% Vandermonde confidence interval
yv_conf_pos = yv + t * sv * sqrt(diag(V * inv(V^T*V) * V^T));
yv_conf_neg = yv - t * sv * sqrt(diag(V * inv(V^T*V) * V^T));
%
%% Vandermonde prediction interval
yv_pred_pos = yv + t * sv * sqrt(diag(V * inv(V^T*V) * V^T) + 1);
yv_pred_neg = yv - t * sv * sqrt(diag(V * inv(V^T*V) * V^T) + 1);
%
%
%% (2) discrete orthogonal polynomials (DOP) basis
% -------------------------------------------------------------
yg = G*G^T * y;
sg = sqrt( 1/df * (y-yg)^T * (y-yg) );
%
%% DOP confidence interval
yg_conf_pos = yg + t * sg * sqrt(diag(G*G^T));
yg_conf_neg = yg - t * sg * sqrt(diag(G*G^T));
%
%% DOP prediction interval
yg_pred_pos = yg + t * sg * sqrt(diag(G*G^T) + 1);
yg_pred_neg = yg - t * sg * sqrt(diag(G*G^T) + 1);
%
```

Lst. 8.1 MATLAB code for the computation of confidence and prediction intervals using Vandermonde and DOP basis functions respectively.

8.3 Local Approximation

Consider the local approximation on a set of m data points, a support length l_s and degree d. Note, that $l_s = d$ is a special case and corresponds to global approximation. The original approach has been presented in [186]. Local polynomial approximation was introduced to smooth and evaluate the derivatives of noisy spectrometer data. The smoothing was based on fitting a geometric polynomial (Vandermonde basis) of low degree to a data set of limited length; as a consequence, it is only applicable to low degree approximations due to the numeric instability. The algorithm has been improved in [164] and extended for cyclic data in [167].

The concept of local approximation is vital for the generation of a regularizing differentiating linear operator as presented in Chapter 10. The process for generating the local approximation (smoothing) operator is visualized in Fig. 8.3.

1. Synthesize a unitary basis $\mathsf{G} \in \mathbb{R}^{(l_s \times (d+1))}$ for $m = l_s$ of degree d. This is a unitary, but incomplete basis since $d < l_s - 1$;

2. Compute the local projection matrix $\mathsf{P} = \mathsf{G}\mathsf{G}^{\mathrm{T}}$ with $\mathsf{P} \in \mathbb{R}^{(l_s \times l_s)}$. The center row P corresponds to computing the projection at the center of the support $\boldsymbol{p}_{x=0}$. These coefficients are symmetric, implying that the frequency response of the FIR filter is strictly linear phase. The rows above and below the center correspond to the projection onto the basis functions at the start and the end of the data, i.e., the end of the support. The coefficients in this region are asymmetric, the corresponding non-linear phase is responsible for the Gibbs error, i.e., they tend to oscillate at the end of their support because the signal cannot be modeled adequately;

3. Generate the global complete band matrix $\bar{\mathsf{P}} \in \mathbb{R}^{(m \times m)}$, which is a linear transformation, but not a projection. The top and the bottom of the projection matrix P is placed at the start and the end of $\bar{\mathsf{P}}$ respectively; note, that $\bar{\mathsf{P}}$ is non-symmetric. The *core* of the matrix $\bar{\mathsf{P}}$ is filled diagonally with $\boldsymbol{p}_{x=0}$. Consequently, the core of the approximation is strictly linear phase and produces no oscillatory behavior. The linear transformation matrix $\bar{\mathsf{P}}$ spans the complete region of m points.

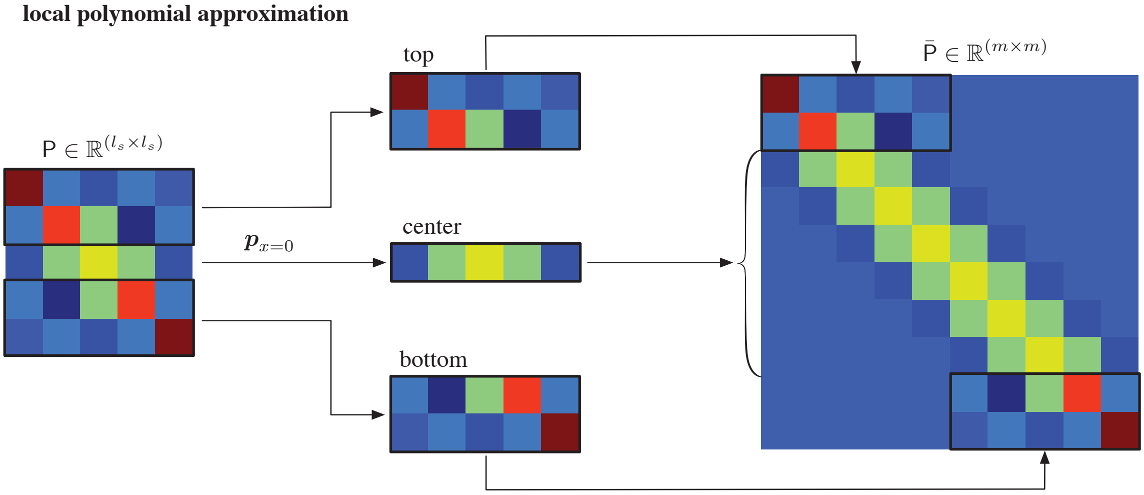

local polynomial approximation
$\mathsf{P} \in \mathbb{R}^{(l_s \times l_s)}$
$\boldsymbol{p}_{x=0}$
top
center
bottom
$\bar{\mathsf{P}} \in \mathbb{R}^{(m \times m)}$

Fig. 8.3 This example illustrates the synthesis of the linear transformation matrix $\bar{\mathsf{P}}$ on $m = 10$ nodes, with support length $l_s = 5$ and a degree $d = 5$. The local projection matrix $\mathsf{P} = \mathsf{G}\,\mathsf{G}^{\mathrm{T}}$ is of dimension $(l_s \times l_s)$. The asymmetric top and bottom of P are selected and placed at the start and end of the support of the band matrix $\bar{\mathsf{P}}$, which is of dimension $(m \times m)$. The diagonal elements of $\bar{\mathsf{P}}$ are filled up with the symmetric center row $\boldsymbol{p}_{x=0}$ of P.

The process of local polynomial approximation becomes computationally intensive for large matrices. The process can be adapted by using the coefficient vector $\boldsymbol{p}_{x=0}$ as a convolution kernel, which significantly reduces the numerical work. Furthermore, this is especially useful in time series data mining, because streaming data has no *beginning* and *end*; continuous convolution of the streamed data enables efficient real-time approximation of derivatives and smoothing of the data within a given support length l_s.

8.4 Interpolation of Incomplete Grids

Interpolation is performed by analyzing basis functions evaluated at incomplete sets of nodes and by subsequent synthesis of the basis functions at the corresponding complete sets of nodes. Spectra on regular grids can still be computed even when data points are missing; this is true for all complete basis functions. Consider a complete set of n nodes from the complete data vector $\boldsymbol{y} \in \mathbb{R}^{(n\times 1)}$, whereby m nodes are known in the incomplete data vector $\hat{\boldsymbol{y}} \in \mathbb{R}^{(m\times 1)}$, see Fig. 8.4. The number of missing elements is $(n-m)$ with $n > m$. Both vectors are related via the permutation matrix $\mathsf{P} \in \mathbb{R}^{(m\times n)}$,

$$\hat{\boldsymbol{y}} = \mathsf{P}\,\boldsymbol{y}. \tag{8.67}$$

The index of the k^{th} known value of $\boldsymbol{y}$ is $\boldsymbol{\rho}(k)$. P has following structure,

$$\mathsf{P}(i,j) = p_{ij} = \begin{cases} 1 & \text{if } i = k \text{ and } j = \boldsymbol{\rho}(k), \\ 0 & \text{if otherwise.} \end{cases} \tag{8.68}$$

The matrix P eliminates the values from $\boldsymbol{y}$ which are generated by interpolation.

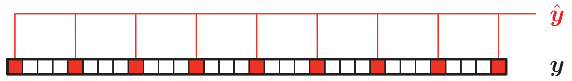


Fig. 8.4 The known vector $\hat{\boldsymbol{y}}$ is a subset of the complete vector $\boldsymbol{y}$, i.e., $\hat{\boldsymbol{y}} \in \boldsymbol{y}$. The values are selected with the permutation matrix P.

The spectra of the known and interpolated data must be identical except for the zero entries, since interpolation does not generate any new information. The complete coefficient vector $\boldsymbol{\alpha} \in \mathbb{R}^{(n\times 1)}$ and the incomplete coefficient vector $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{(m\times 1)}$ are related with the permutation matrix $\mathsf{Q} \in \mathbb{R}^{(n\times m)}$,

$$\boldsymbol{\alpha} = \mathsf{Q}\,\hat{\boldsymbol{\alpha}}, \tag{8.69}$$

whereby Q has the structure,

$$\mathsf{Q}(i,j) = q_{ij} = \begin{cases} 1 & \text{if } i = \boldsymbol{b}_k \text{ and } j = k, \\ 0 & \text{if otherwise,} \end{cases} \tag{8.70}$$

where $\boldsymbol{b}_k$ stores the k^{th} basis functions representing of the known values in $\hat{\boldsymbol{y}}$. This is the most general case of zero padding, where Q inserts zeros at the appropriate places in the spectrum $\boldsymbol{\alpha}$. The values for $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are synthesized from their respective spectra $\boldsymbol{\alpha}$ and $\hat{\boldsymbol{\alpha}}$, i.e.,

$$\boldsymbol{y} = \mathsf{B}\,\boldsymbol{\alpha} \quad \text{and} \quad \hat{\boldsymbol{y}} = \mathsf{B}\,\hat{\boldsymbol{\alpha}}. \tag{8.71}$$

Using Eqn. (8.67),

$$\begin{aligned}\hat{\boldsymbol{y}} &= \mathsf{P}\,\boldsymbol{y},\\ &= \mathsf{P}\,\mathsf{B}\,\boldsymbol{\alpha},\\ &= \mathsf{P}\,\mathsf{B}\,\mathsf{Q}\,\hat{\boldsymbol{\alpha}},\end{aligned} \tag{8.72}$$

and solving for $\hat{\boldsymbol{\alpha}}$ in a least squares sense,

$$\hat{\boldsymbol{\alpha}} = \{\mathsf{P}\,\mathsf{B}\,\mathsf{Q}\}^{+}\,\hat{\boldsymbol{y}}. \tag{8.73}$$

Substitution of Eqn. (8.69),

$$\boldsymbol{\alpha} = \mathsf{Q}\,\{\mathsf{P}\,\mathsf{B}\,\mathsf{Q}\}^{+}\,\hat{\boldsymbol{y}}, \tag{8.74}$$

yields the interpolated data vector $\boldsymbol{y}$,

$$\boldsymbol{y} = \mathsf{B}\,\mathsf{Q}\,\{\mathsf{P}\,\mathsf{B}\,\mathsf{Q}\}^{+}\,\hat{\boldsymbol{y}}. \tag{8.75}$$

The derivation is completely general for any complete basis function B and/or unitary basis function G. The theory can be extended to a 2D Cartesian grid, i.e., a uniform lattice. This enables the implementation of separable (anisotropic) basis functions in x- and y-dimension. Consider the data matrix $\hat{\Omega} \in \mathbb{R}^{(m\times n)}$ to be interpolated,

$$\hat{\Omega} = \hat{\mathsf{Y}}\,\hat{\Sigma}\,\hat{\mathsf{X}}^{\mathrm{T}}, \tag{8.76}$$

where the incomplete bases functions $\hat{\mathsf{X}} \in \mathbb{R}^{(n\times v)}$ and $\hat{\mathsf{Y}} \in \mathbb{R}^{(m\times u)}$ are zero patched with the permutation matrices P and Q,

$$\hat{\mathsf{X}} = \mathsf{P}\,\mathsf{X}\,\mathsf{Q} \quad \text{and} \quad \hat{\mathsf{Y}} = \mathsf{P}\,\mathsf{Y}\,\mathsf{Q}. \tag{8.77}$$

The $\hat{\Sigma} \in \mathbb{R}^{(m\times n)}$ spectrum of the known data is

$$\hat{\Sigma} = \hat{\mathsf{Y}}^{+}\,\hat{\Omega}\,\hat{\mathsf{X}}^{+\mathrm{T}}. \tag{8.78}$$

The interpolated data $\Omega \in \mathbb{R}^{(u\times v)}$, where $u > m$ and $v > n$, uses the complete basis functions $\mathsf{X} \in \mathbb{R}^{(v\times v)}$ and $\mathsf{Y} \in \mathbb{R}^{(u\times u)}$,

$$\begin{aligned}\Omega &= \mathsf{Y}\,\hat{\Sigma}\,\mathsf{X}^{\mathrm{T}},\\ &= \mathsf{Y}\,\{\hat{\mathsf{Y}}^{+}\,\hat{\Omega}\,\hat{\mathsf{X}}^{+\mathrm{T}}\}\,\mathsf{X}^{\mathrm{T}},\end{aligned} \tag{8.79}$$

which is in means of dimensions,

$$(u\times v) = (u\times u)(u\times m)(m\times n)(n\times v)(v\times v).$$

The complete 2D tensor interpolation process is then written as

$$\Omega = \mathsf{Y}\,\hat{\mathsf{Y}}^{+}\,\hat{\Omega}\,(\hat{\mathsf{X}}\,\mathsf{X}^{+})^{\mathrm{T}}. \tag{8.80}$$

This methodology has be used for the mapping of 2D coordinates from a perturbed camera space to metric space in [70, 71, 157] and is explained in Section 9.3 as well as in the project in Chapter 11.

9 | Modified Basis Functions

The concept of basis function and their formulation in linear algebra opens the possibility to extend their functionality with linear operations. In this chapter, weighted and constrained basis functions are introduced as well as bivariate extension. Weighting is of special importance when the nodes on which the basis function is synthesized have qualitative differences, e.g., their uncertainty is known. This is useful for analytic multi-source data fusion as published in [75, 76]. Constrained basis functions incorporate a-priori knowledge about the solution space and, therefore, they are more accurate while simultaneously being more numerically efficient [69]. Furthermore, constrained basis functions are the theoretical foundation for solving ODE and PDE using linear differential operators, see Chapter 10. The extension of basis functions to the 2D space is used in the processing of images and 3D scan data [71, 158] as well as in the solution of PDE [161]. Bivariate basis functions are independent in x and y dimension, i.e., anisotropic bases are usable.

9.1 Weighted Approximation

The weighted cost function $K(\boldsymbol{\alpha})$ with the coefficient vector $\boldsymbol{\alpha}$ is written as

$$K(\boldsymbol{\alpha}) = \|\mathsf{W}^{\frac{1}{2}}\,\boldsymbol{r}\|_2, \tag{9.1}$$

where $\boldsymbol{r}$ is the residual vector and $\mathsf{W}^{\frac{1}{2}}$ is the symmetric square root of the full-rank weighting matrix $\mathsf{W} = \operatorname{diag}(w_1 \dots w_m)$, i.e., the weightings evaluated at the nodes $\boldsymbol{x}$. W is a real positive definite matrix, i.e., all eigenvalues of W are real positive. From a measurement application's point of view, it makes sense to choose $\mathsf{W}^{1/2} \triangleq \Lambda_{\boldsymbol{y}}^{-1/2}$. The weighting then corresponds to the inverse of the input signal's standard deviation, which is effectively usable for uncertainty weighted multi-source data fusion. Expansion of Eqn. (9.1) yields

$$\begin{aligned}
K(\boldsymbol{\alpha}) &= \left(\mathsf{W}^{\frac{1}{2}}\,\boldsymbol{r}\right)^{\mathrm{T}}\left(\mathsf{W}^{\frac{1}{2}}\,\boldsymbol{r}\right), \\
&= \boldsymbol{r}^{\mathrm{T}}\left(\mathsf{W}^{\frac{1}{2}}\right)^{\mathrm{T}}\left(\mathsf{W}^{\frac{1}{2}}\right)\boldsymbol{r}, \\
&= \boldsymbol{r}^{\mathrm{T}}\mathsf{W}\boldsymbol{r}.
\end{aligned} \tag{9.2}$$

In the special case, where $\mathsf{W} = \mathsf{I}$, the scalar product $\boldsymbol{r}^{\mathrm{T}}\boldsymbol{r}$ is computed as the weighting function, leading to unweighted regression with i.i.d. noise. The concept of weighted regression arises when the measurements are corrupted by anisotropic noise. In this case, the residual is defined as $\boldsymbol{r} = \boldsymbol{y} - \mathsf{B}\,\boldsymbol{\alpha}$. Consequently,

$$K(\boldsymbol{\alpha}) = (\boldsymbol{y} - \mathsf{B}\,\boldsymbol{\alpha})^{\mathrm{T}}\mathsf{W}\,(\boldsymbol{y} - \mathsf{B}\,\boldsymbol{\alpha})\,. \tag{9.3}$$

The cost function $K(\boldsymbol{\alpha})$ represents a Mahalanobis distance which must be minimized in order to acquire a least squares solution. In statistics, this is a direct analogy to a χ^2-goodness-of-fit test, see Section 5.5. Hence,

$$\chi^2 \leq (\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}])^{\mathrm{T}} {\Lambda_{\hat{\boldsymbol{x}}}}^{-1} (\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}]) \tag{9.4}$$

for a variable $\hat{\boldsymbol{y}} \in \mathbb{R}^{(m\times 1)}$, where the weighting $\mathsf{W} = {\Lambda_{\hat{\boldsymbol{y}}}}^{-1}$ is the inverse of the covariance. The minimization is performed by differentiation,

$$\frac{\partial K(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = \frac{\partial}{\partial \boldsymbol{\alpha}} \Big[(\boldsymbol{y} - \mathsf{B}\,\boldsymbol{\alpha})^{\mathrm{T}} \mathsf{W}\,(\boldsymbol{y} - \mathsf{B}\,\boldsymbol{\alpha}) \Big] = \boldsymbol{0}. \tag{9.5}$$

Deriving $K(\boldsymbol{\alpha})$ w.r.t. $\boldsymbol{\alpha}$ yields the unit vector $\boldsymbol{e}_i \in \mathbb{R}^{(m\times 1)}$, i.e., a vector of $m-1$ zeros with a single one at the i^{th} position,

$$\frac{\partial \boldsymbol{\alpha}}{\partial \alpha_i} = \boldsymbol{e}_i = [0,\ldots,0,1,0,\ldots,0]^{\mathrm{T}}. \tag{9.6}$$

This mathematical formalism is necessary to conserve the correct matrix dimensions. Using the product rule, the differentiation for a single value y_i of the vector $\boldsymbol{y}$ yields

$$0 = -\,(\mathsf{B}\boldsymbol{e}_i)^{\mathrm{T}} \mathsf{W}(\boldsymbol{y} - \mathsf{B}\boldsymbol{\alpha}) - (\boldsymbol{y} - \mathsf{B}\boldsymbol{\alpha})^{\mathrm{T}} \mathsf{W}\mathsf{B}\boldsymbol{e}_i. \tag{9.7}$$

Expansion of both terms,

$$0 = -\,(\mathsf{B}\boldsymbol{e}_i)^{\mathrm{T}} \mathsf{W}\boldsymbol{y} + (\mathsf{B}\boldsymbol{e}_i)^{\mathrm{T}} \mathsf{W}\mathsf{B}\boldsymbol{\alpha} - \boldsymbol{y}^{\mathrm{T}} \mathsf{W}\mathsf{B}\boldsymbol{e}_i + (\mathsf{B}\boldsymbol{\alpha})^{\mathrm{T}} \mathsf{W}\mathsf{B}\boldsymbol{e}_i, \tag{9.8}$$

and factorization yields

$$0 = -\,\boldsymbol{e}_i^{\mathrm{T}} \mathsf{B}^{\mathrm{T}} \mathsf{W}\boldsymbol{y} + \boldsymbol{e}_i^{\mathrm{T}} \mathsf{B}^{\mathrm{T}} \mathsf{W}\mathsf{B}\boldsymbol{\alpha} - \boldsymbol{y}^{\mathrm{T}} \mathsf{W}\mathsf{B}\boldsymbol{e}_i + \boldsymbol{\alpha}^{\mathrm{T}} \mathsf{B}^{\mathrm{T}} \mathsf{W}\mathsf{B}\boldsymbol{e}_i. \tag{9.9}$$

Each term has a dimension of $\mathbb{R}^{(1\times 1)}$, i.e., the individual terms describe scalars. Therefore, the terms can be transposed in order to simplify the equation,

$$0 = \;2\,\boldsymbol{e}_i^{\mathrm{T}} \mathsf{B}^{\mathrm{T}} \mathsf{W}\boldsymbol{y} - 2\,\boldsymbol{e}_i^{\mathrm{T}} \mathsf{B}^{\mathrm{T}} \mathsf{W}\mathsf{B}\boldsymbol{\alpha}. \tag{9.10}$$

The scalar 2 cancels out,

$$0 = \;\boldsymbol{e}_i^{\mathrm{T}} \mathsf{B}^{\mathrm{T}} \mathsf{W}(\boldsymbol{y} - \mathsf{B}\boldsymbol{\alpha}). \tag{9.11}$$

As a consequence, the equation in matrix form is

$$\boldsymbol{e}_i^{\mathrm{T}} \mathsf{B}^{\mathrm{T}} \mathsf{W}\boldsymbol{y} = \boldsymbol{e}_i^{\mathrm{T}} \mathsf{B}^{\mathrm{T}} \mathsf{W}\mathsf{B}\boldsymbol{\alpha}. \tag{9.12}$$

Stacking the results for each y_i delivers

$$\begin{bmatrix} \boldsymbol{e}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}_i^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}_m^{\mathrm{T}} \end{bmatrix} (\mathsf{B}^{\mathrm{T}} \mathsf{W}\boldsymbol{y}) = \begin{bmatrix} \boldsymbol{e}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}_i^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}_m^{\mathrm{T}} \end{bmatrix} (\mathsf{B}^{\mathrm{T}} \mathsf{W}\mathsf{B}\boldsymbol{\alpha}), \tag{9.13}$$

whereby each stack of element vectors equals the identity matrix I. This leads to the normal equations of weighted regression,

$$\mathsf{B}^{\mathrm{T}} \mathsf{W}\,\boldsymbol{y} = \mathsf{B}^{\mathrm{T}} \mathsf{W}\mathsf{B}\,\boldsymbol{\alpha}. \tag{9.14}$$

(1) Unweighted Regression

In the special case where $\mathsf{W} \triangleq \mathsf{I}$, i.e., unweighted regression, the equation simplifies to the classical normal equation [65],

$$\mathsf{B}^{\mathrm{T}}\boldsymbol{y} = \mathsf{B}^{\mathrm{T}}\mathsf{B}\,\boldsymbol{\alpha}, \tag{9.15}$$

which is also shown in Section 6.4. The term $(\mathsf{B}^{\mathrm{T}}\mathsf{B})$ is an invertible square matrix, even when B is rectangular. Therefore, the estimation $\hat{\boldsymbol{\alpha}}$ of the true coefficients $\boldsymbol{\alpha}$ is

$$\hat{\boldsymbol{\alpha}} = (\mathsf{B}^{\mathrm{T}}\mathsf{B})^{-1}\mathsf{B}^{\mathrm{T}}\boldsymbol{y} = \mathsf{B}^{+}\,\boldsymbol{y}. \tag{9.16}$$

The Moore-Penrose pseudoinverse of the basis B is defined as $\mathsf{B}^{+} \triangleq (\mathsf{B}^{\mathrm{T}}\mathsf{B})^{-1}\mathsf{B}^{\mathrm{T}}$. The unweighted approximation $\hat{\boldsymbol{y}} = \mathsf{B}\,\hat{\boldsymbol{\alpha}}$ of the true curve $\boldsymbol{y}$ is then

$$\hat{\boldsymbol{y}} = \mathsf{B}\mathsf{B}^{+}\,\boldsymbol{y}. \tag{9.17}$$

When using a unitary basis $\mathsf{G} \triangleq \mathsf{B}$ with $\mathsf{G}^{+} = \mathsf{G}^{\mathrm{T}}$, then the equation simplifies to

$$\hat{\boldsymbol{y}} = \mathsf{G}\,\mathsf{G}^{\mathrm{T}}\,\boldsymbol{y}. \tag{9.18}$$

(2) Weighted Regression with General Basis B

Solving for the estimation $\hat{\boldsymbol{\alpha}}$ of the coefficient vector $\boldsymbol{\alpha}$ for the general (weighted) case yields

$$\begin{aligned}\hat{\boldsymbol{\alpha}} &= \left(\mathsf{B}^{\mathrm{T}}\mathsf{W}\mathsf{B}\right)^{-1}\mathsf{B}^{\mathrm{T}}\mathsf{W}\,\boldsymbol{y},\\ &= \left(\mathsf{B}^{\mathrm{T}}\mathsf{W}^{\frac{1}{2}}\mathsf{W}^{\frac{1}{2}}\mathsf{B}\right)^{-1}\mathsf{B}^{\mathrm{T}}\mathsf{W}^{\frac{1}{2}}\mathsf{W}^{\frac{1}{2}}\,\boldsymbol{y},\\ &= \left(\mathsf{W}^{\frac{1}{2}}\mathsf{B}\right)^{+}\mathsf{W}^{\frac{1}{2}}\,\boldsymbol{y}.\end{aligned} \tag{9.19}$$

The computation of the pseudoinverse $(\mathsf{W}^{\frac{1}{2}}\mathsf{B})^{+}$ is numerically more stable than the inversion of $\left(\mathsf{B}^{\mathrm{T}}\mathsf{W}\mathsf{B}\right)^{-1}$. Utilizing this result for the polynomial approximation $\hat{\boldsymbol{y}} = \mathsf{B}\,\hat{\boldsymbol{\alpha}}$,

$$\hat{\boldsymbol{y}} = \mathsf{B}\left(\mathsf{W}^{\frac{1}{2}}\mathsf{B}\right)^{+}\mathsf{W}^{\frac{1}{2}}\boldsymbol{y}, \tag{9.20}$$

generates the weighted least squares transformation matrix L, which is defined as

$$\mathsf{L} \triangleq \mathsf{B}\left(\mathsf{W}^{\frac{1}{2}}\mathsf{B}\right)^{+}\mathsf{W}^{\frac{1}{2}}. \tag{9.21}$$

The linear operation is derived by substituting L from Eqn. (9.21) into Eqn. (9.20),

$$\hat{\boldsymbol{y}} \triangleq \mathsf{L}\,\boldsymbol{y}. \tag{9.22}$$

(3) Weighted Regression with DOP Basis G

If a unitary basis $\mathsf{B} \triangleq \mathsf{G}$ is used, the weighting is generated in the form

$$\bar{\mathsf{G}}^{\mathrm{T}}\mathsf{W}\bar{\mathsf{G}} = \mathsf{I}, \tag{9.23}$$

where $\bar{\mathsf{G}}$ is the weighted unitary basis, such that $\bar{\mathsf{G}} = \mathsf{G}\,\mathsf{U}$ and U is a full rank upper triangular matrix since G is sorted according to increasing degree. Substitution into the above equation yields

$$(\mathsf{G}\,\mathsf{U})^{\mathrm{T}}\mathsf{W}\,\mathsf{G}\,\mathsf{U} = \mathsf{I}, \quad \text{thus,} \quad \mathsf{U}^{\mathrm{T}}\,\mathsf{G}^{\mathrm{T}}\mathsf{W}\,\mathsf{G}\,\mathsf{U} = \mathsf{I}. \tag{9.24}$$

Since U is full rank, inversion is possible,

$$\mathsf{G}^{\mathrm{T}}\mathsf{W}\,\mathsf{G} = \mathsf{U}^{-\mathrm{T}}\mathsf{U}^{-1}. \tag{9.25}$$

A unique Cholesky decomposition exists for real positive definite matrices, i.e., the Cholesky decomposition of $\{\mathsf{G}^{\mathrm{T}}\mathsf{W}\,\mathsf{G}\}$ exists if W is real positive definite, since G is unitary. Decomposition yields

$$\mathsf{G}^{\mathrm{T}}\mathsf{W}\,\mathsf{G} = \mathsf{K}\,\mathsf{K}^{\mathrm{T}}, \tag{9.26}$$

where K is a full rank lower triangular matrix, consequently,

$$\mathsf{K}\,\mathsf{K}^{\mathrm{T}} \triangleq \mathsf{U}^{-\mathrm{T}}\mathsf{U}^{-1} \quad \text{with} \quad \mathsf{U} = \mathsf{K}^{-\mathrm{T}}. \tag{9.27}$$

The weighted basis function $\bar{\mathsf{G}} = \mathsf{G}\,\mathsf{U} = \mathsf{G}\,\mathsf{K}^{-\mathrm{T}}$ is fully defined by this computation. Note, that the transpose of a lower triangular matrix is an upper triangular matrix. The condition number of the transformation solely depends on the condition number of W, see also Section 7.6. Substitution of $\mathsf{B} \triangleq \bar{\mathsf{G}}$ into Eqn. (9.14) gives

$$\bar{\mathsf{G}}^{\mathrm{T}}\mathsf{W}\boldsymbol{y} = \left(\bar{\mathsf{G}}^{\mathrm{T}}\mathsf{W}\bar{\mathsf{G}}\right)\boldsymbol{\alpha} \tag{9.28}$$

and, by the definition of the weighted discrete basis functions $\bar{\mathsf{G}}^{\mathrm{T}}\mathsf{W}\bar{\mathsf{G}} = \mathsf{I}$, no matrix inversion is necessary to solve the linear system of equations w.r.t. the coefficient's estimation $\hat{\boldsymbol{\alpha}}$,

$$\hat{\boldsymbol{\alpha}} = \bar{\mathsf{G}}^{\mathrm{T}}\,\mathsf{W}\,\boldsymbol{y}. \tag{9.29}$$

The weighted approximation $\hat{\boldsymbol{y}} = \bar{\mathsf{G}}\,\hat{\boldsymbol{\alpha}}$ using a unitary basis is then

$$\hat{\boldsymbol{y}} = \bar{\mathsf{G}}\,\bar{\mathsf{G}}^{\mathrm{T}}\,\mathsf{W}\boldsymbol{y}. \tag{9.30}$$

Obviously, the weighting has no influence on the result when using complete bases,

$$\mathsf{W}^{-1} \triangleq \bar{\mathsf{G}}^{\mathrm{T}}\bar{\mathsf{G}} \quad \text{with} \quad \bar{\mathsf{G}}^{\mathrm{T}}\bar{\mathsf{G}} = \mathsf{I}. \tag{9.31}$$

(4) Alternative: Synthesis of a Weighted DOP Basis

The previous approach assumes that the real positive definite weighting matrix W is full-rank. In the case, where W is rank deficient, a weighted unitary basis function must be synthesized from the recurrence relationship. The generalized recurrence relationship from Eqn. (7.2) is

$$\boldsymbol{g}_n = \alpha\,(\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + \mathsf{G}_{n-1}\,\boldsymbol{\beta}, \tag{9.32}$$

whereby the constraints C_1 for the orthogonality condition and C_2 for unit norm condition are already weighted as following,

$$C_1: \quad \mathsf{G}_{n-1}^{\mathrm{T}}\,\mathsf{W}\,\boldsymbol{g}_n = \boldsymbol{0} \quad \text{and} \quad C_2: \quad \boldsymbol{g}_n^{\mathrm{T}}\,\mathsf{W}\,\boldsymbol{g}_n = c_n. \tag{9.33}$$

Substitution of the recurrence relationship into the constraint C_1 yields

$$\boldsymbol{0} = \alpha\, \mathsf{G}_{n-1}^{\mathrm{T}}\, \mathsf{W}\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x}) + \mathsf{G}_{n-1}^{\mathrm{T}}\, \mathsf{W}\, \mathsf{G}_{n-1}\, \boldsymbol{\beta}. \tag{9.34}$$

Since the generated basis functions are already orthogonal w.r.t. weighting function,

$$\mathsf{G}_{n-1}^{\mathrm{T}}\, \mathsf{W}\, \mathsf{G}_{n-1} = \mathsf{C}_n = \mathrm{diag}\,(c_1 \ldots c_n)\,. \tag{9.35}$$

The $\boldsymbol{\beta}$ recurrence coefficient vector is

$$\boldsymbol{\beta} = -\alpha\, \mathsf{C}_n^{-1}\, \mathsf{G}_{n-1}^{\mathrm{T}}\, \mathsf{W}(\boldsymbol{g}_{n-1} \circ \boldsymbol{x}). \tag{9.36}$$

Using $\boldsymbol{\beta}$ and the constraint C_2 yields the α recurrence coefficient,

$$\alpha = \frac{\sqrt{c_n}}{\|\mathsf{W}^{\frac{1}{2}}\, \{\mathsf{I} - \mathsf{G}_{n-1}\, \mathsf{C}_n^{-1}\, \mathsf{G}_{n-1}^{\mathrm{T}}\, \mathsf{W}\}\, (\boldsymbol{g}_{n-1} \circ \boldsymbol{x})\|_2}. \tag{9.37}$$

Using these weighted α and $\boldsymbol{\beta}$ recurrence coefficients during the synthesis algorithm delivers the weighted DOP basis functions.

9.2 Constrained Basis Functions

When considering physical structures, there are generally a number of constraints which need to be placed on the solution [165]. Initial value problems (IVPs) and boundary value problems (BVPs) are common examples and are discussed in detail in Chapter 10. The homogeneously constrained DOP basis function $\bar{\mathsf{G}}$ satisfies the linear equation

$$\mathsf{C}^{\mathrm{T}} \bar{\mathsf{G}} = \mathsf{0} \quad \text{where} \quad \bar{\mathsf{G}}^{\mathrm{T}} \bar{\mathsf{G}} = \mathsf{I}. \tag{9.38}$$

The matrix $\mathsf{C}^{\mathrm{T}} \in \mathbb{R}^{(p \times m)}$ contains p constraints and $\bar{\mathsf{G}} \in \mathbb{R}^{(m \times (d+1))}$ is a DOP basis of degree d synthesized at m points, i.e., for a full basis $m = d+1$. Each constraint removes a d.f. from the least squares solution space. Reducing the dimension of the solution space prior to the estimation's computation is advantageous both w.r.t. numerical efficiency and noise behavior. In this section, a procedure is proposed for the synthesis of discrete unitary basis functions, which can fulfill an arbitrary set of positional (e.g., Dirichlet for BVP) and derivative (e.g., Neumann for BVP) constraints. Infinite basis functions fulfill these constraints in Eqn. (9.38), i.e., the admissible functions are non-unique. The goal is to find a formulation that ensures a unique solution. The constrained basis function $\bar{\mathsf{G}}$ is related to the unconstrained basis function $\mathsf{G} \in \mathbb{R}^{(m \times (d+1))}$ via the upper triangular matrix $\bar{\mathsf{X}} \in \mathbb{R}^{((d+1) \times (d+1-p))}$,

$$\bar{\mathsf{G}} = \mathsf{G}\, \bar{\mathsf{X}}, \tag{9.39}$$

i.e., $\bar{\mathsf{G}}$ is a linear combination of G. Using Eqn.(9.38) reveals the orthogonality of $\bar{\mathsf{X}}$,

$$\begin{aligned} \bar{\mathsf{G}}^{\mathrm{T}} \bar{\mathsf{G}} &= \bar{\mathsf{X}}^{\mathrm{T}} \mathsf{G}^{\mathrm{T}} \mathsf{G}\, \bar{\mathsf{X}}, \\ &= \bar{\mathsf{X}}^{\mathrm{T}} \bar{\mathsf{X}}. \end{aligned} \tag{9.40}$$

Since $\bar{\mathsf{G}}^{\mathrm{T}} \bar{\mathsf{G}} = \mathsf{I}$, it follows that $\bar{\mathsf{X}}^{\mathrm{T}} \bar{\mathsf{X}} = \mathsf{I}$. The upper triangular structure of $\bar{\mathsf{X}}$ ensures the same ordering of $\bar{\mathsf{G}}$ as G w.r.t. the degree. Using Eqn. (9.39) in Eqn. (9.38) yields

$$\mathsf{C}^{\mathrm{T}} \mathsf{G}\, \bar{\mathsf{X}} = \mathsf{0}. \tag{9.41}$$

The number of independent constraints is $p = \text{rank}\left(\mathsf{C}^{\mathrm{T}}\mathsf{G}\right)$. Applying singular value decomposition on $\mathsf{C}^{\mathrm{T}}\mathsf{G}$,

$$\mathsf{C}^{\mathrm{T}}\mathsf{G} = \mathsf{U}\,\mathsf{S}\,\mathsf{V}^{\mathrm{T}}, \tag{9.42}$$

and selecting the last $(p+1)^{\text{th}}$ columns of V yields $\mathsf{N} \in \mathsf{V}$,

$$\mathsf{N} = \mathsf{V}(:, (p+1) : \text{end}). \tag{9.43}$$

RQ decomposition[1] of N delivers

$$\mathsf{N} = \mathsf{R}\,\mathsf{Q}, \tag{9.44}$$

where R is upper triangular and Q is orthogonal,

$$\mathsf{Q}^{\mathrm{T}}\mathsf{Q} = \mathsf{I}, \quad \text{and it follows that} \quad \mathsf{R}^{\mathrm{T}}\mathsf{R} = \mathsf{I}, \tag{9.45}$$

since $\mathsf{V}^{\mathrm{T}}\mathsf{V} = \mathsf{I}$ and $\mathsf{N}^{\mathrm{T}}\mathsf{N} = \mathsf{I}$, hence,

$$\text{range}\,(\mathsf{G}\mathsf{N}) = \text{range}\,(\mathsf{G}\mathsf{R}) \quad \text{and, therefore,} \quad \bar{\mathsf{X}} \triangleq \mathsf{R}. \tag{9.46}$$

Synthesizing the matrix R for $\bar{\mathsf{X}}$ with this algorithm ensures the fulfillment of all constraints. It is worthy to note, that this procedure is valid for all discrete unitary bases, not only polynomials.

In the proposed framework, a constraint is implemented by restricting a linear combination $\boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{y}$ of the solution vector $\boldsymbol{y}$ to have a scalar vector d_i, i.e.,

$$\boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{y} = d_i. \tag{9.47}$$

This is a general mechanism, since any constrained function can be implemented. Consider following example: $y(0) = y(1)$, $y'(0) = y'(1)$ and $y''(0) = y''(1)$. Given the differentiating matrices D and D^2, the three constraints are formulated as

$$[1, 0, \ldots, 0, -1]\,\boldsymbol{y} = \boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{y} = 0, \tag{9.48}$$

$$\{\mathsf{D}(1,:) - \mathsf{D}(\text{end},:)\}\,\boldsymbol{y} = \boldsymbol{c}_2^{\mathrm{T}}\boldsymbol{y} = 0, \tag{9.49}$$

$$\{\mathsf{D}^2(1,:) - \mathsf{D}^2(\text{end},:)\}\,\boldsymbol{y} = \boldsymbol{c}_3^{\mathrm{T}}\boldsymbol{y} = 0. \tag{9.50}$$

Given a set of p constraints, the constraining vectors $\boldsymbol{c}_i \in \mathbb{R}^{(m\times 1)}$ are concatenated to form the matrix of constraints $\mathsf{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_p]$ and the corresponding scalars d_i form the vector $\boldsymbol{d} = [d_1, \ldots, d_p]^{\mathrm{T}} \in \mathbb{R}^{(p\times 1)}$, such that

$$\mathsf{C}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{d}. \tag{9.51}$$

9.3 Bivariate Transformation

The methodology of extending basis functions to a 2D regular lattice has been shown in [162, 164, 168] as well as in the project described in Chapter 11. Furthermore, two worked examples are appended to this section. Bivariate basis functions are of special importance in machine vision and measurement of surfaces where the values within the data matrix $\Omega \in \mathbb{R}^{(m\times n)}$ correspond to illumination or elevation respectively. The data points of Ω lie on an invariant

[1] The RQ decomposition is not to be confused with the more common QR decomposition.

Cartesian grid, i.e., a uniform lattice, with m rows and n columns; this structure significantly simplifies the computation. The geometric surface $\mathsf{\Omega}$ is formulated as

$$\mathsf{\Omega} = \mathsf{Y}\,\mathsf{\Sigma}\,\mathsf{X}^{\mathrm{T}}, \tag{9.52}$$

where the spectrum $\mathsf{\Sigma}$ is evaluated at the Cartesian grid by the separable basis functions $\mathsf{X} \triangleq \mathsf{B}_{\mathsf{X}}$ in x-dimension and $\mathsf{Y} \triangleq \mathsf{B}_{\mathsf{Y}}$ in y-dimension respectively. The bivariate transformation in this formulation supports anisotropic bases, i.e., $\mathsf{X} \in \mathbb{R}^{(n\times(d_{\mathsf{X}}+1))}$ and $\mathsf{Y} \in \mathbb{R}^{(m\times(d_{\mathsf{Y}}+1))}$ can be implemented with different degrees d_{X} and d_{Y}, or even as completely different types of basis functions. This is especially useful when the data is better modeled by different basis functions along different axes. The following proof makes no assumptions about the nature of the basis functions and their evaluation at the nodes, except that the bases do not degenerate. The residual matrix R contains the geometric error,

$$\mathsf{R} = \mathsf{\Omega} - \mathsf{Y}\,\mathsf{\Sigma}\,\mathsf{X}^{\mathrm{T}}. \tag{9.53}$$

The values for the measurement $\mathsf{\Omega}$ are known from the analysis process. The basis functions X and Y are then synthesized with appropriate degrees d_{X} and d_{Y} respectively, whereby $d_{\mathsf{X}} = n-1$ and $d_{\mathsf{Y}} = m-1$ describe complete bases. The goal is to acquire the spectrum (moments) in a least squares sense as a maximum likelihood estimation. The cost function $K(\mathsf{\Sigma})$ of the 2D tensor polynomial regression is then

$$\begin{aligned} K(\mathsf{\Sigma}) &= \|\mathsf{R}\|_{\mathrm{F}}^2 = \operatorname{trace}\left(\mathsf{R}\,\mathsf{R}^{\mathrm{T}}\right), \\ &= \operatorname{trace}\left(\{\mathsf{\Omega} - \mathsf{Y}\,\mathsf{\Sigma}\,\mathsf{X}^{\mathrm{T}}\}\,\{\mathsf{\Omega} - \mathsf{Y}\,\mathsf{\Sigma}\,\mathsf{X}^{\mathrm{T}}\}^{\mathrm{T}}\right), \\ &= \operatorname{trace}\left(\mathsf{Y}\,\mathsf{\Sigma}\,\mathsf{X}^{\mathrm{T}}\,\mathsf{X}\,\mathsf{\Sigma}^{\mathrm{T}}\,\mathsf{Y}^{\mathrm{T}}\right) - \operatorname{trace}\left(\mathsf{Y}\,\mathsf{\Sigma}\,\mathsf{X}^{\mathrm{T}}\,\mathsf{\Omega}\right) \\ &\quad - \operatorname{trace}\left(\mathsf{\Omega}\,\mathsf{X}\,\mathsf{\Sigma}^{\mathrm{T}}\,\mathsf{Y}^{\mathrm{T}}\right) + \operatorname{trace}\left(\mathsf{\Omega}\,\mathsf{\Omega}^{\mathrm{T}}\right). \end{aligned} \tag{9.54}$$

The differentiation of the cost function $K(\mathsf{\Sigma})$ w.r.t. the spectrum $\mathsf{\Sigma}$ and setting it equal to zero transforms the scalar equation to a matrix equation, a detailed description is given in [90],

$$\frac{\partial K}{\partial \mathsf{\Sigma}} = 2\,\left(\mathsf{Y}^{\mathrm{T}}\mathsf{Y}\,\mathsf{\Sigma}\,\mathsf{X}^{\mathrm{T}}\mathsf{X} - \mathsf{Y}^{\mathrm{T}}\mathsf{\Omega}\,\mathsf{X}\right). \tag{9.55}$$

This operation directly leads to the approximation $\hat{\mathsf{\Sigma}}$ of the spectrum $\mathsf{\Sigma}$,

$$\begin{aligned} \hat{\mathsf{\Sigma}} &= (\mathsf{Y}^{\mathrm{T}}\mathsf{Y})^{-1}\mathsf{Y}^{\mathrm{T}}\mathsf{\Omega}\,\mathsf{X}\,(\mathsf{X}^{\mathrm{T}}\mathsf{X})^{-1}, \\ &= \mathsf{Y}^{+}\mathsf{\Omega}\,\mathsf{X}^{+\mathrm{T}}. \end{aligned} \tag{9.56}$$

Computation of the spectrum in this manner generates an 2D tensor approximation of the surface $\hat{\mathsf{\Omega}} = \mathsf{Y}\,\hat{\mathsf{\Sigma}}\,\mathsf{X}^{\mathrm{T}}$, where the sum of the squares of the vertical distances is minimized. Reconstruction of the approximated surface $\hat{\mathsf{\Omega}}$ at all points on the Cartesian grid is done by substitution of $\hat{\mathsf{\Sigma}}$ from the above equation,

$$\hat{\mathsf{\Omega}} = \mathsf{Y}\,\mathsf{Y}^{+}\mathsf{\Omega}\,(\mathsf{X}\,\mathsf{X}^{+})^{\mathrm{T}}. \tag{9.57}$$

This effectively implements a polynomial filter when using incomplete basis functions $\hat{\mathsf{X}}$ and $\hat{\mathsf{Y}}$ with $d_{\mathsf{X}} < n-1$ and $d_{\mathsf{Y}} < m-1$. In the case of unitary polynomial bases, $\mathsf{X} \triangleq \mathsf{G}_{\mathsf{X}}$ and $\mathsf{Y} \triangleq \mathsf{G}_{\mathsf{Y}}$, the analysis process simplifies to

$$\hat{\mathsf{\Sigma}} = \mathsf{Y}^{\mathrm{T}}\mathsf{\Omega}\,\mathsf{X}, \tag{9.58}$$

which is visualized in Fig. 9.1 for complete bases. The synthesis becomes

$$\hat{\mathsf{\Omega}} = \mathsf{Y}\,\mathsf{Y}^{\mathrm{T}}\mathsf{\Omega}\,\mathsf{X}\,\mathsf{X}^{\mathrm{T}}. \tag{9.59}$$

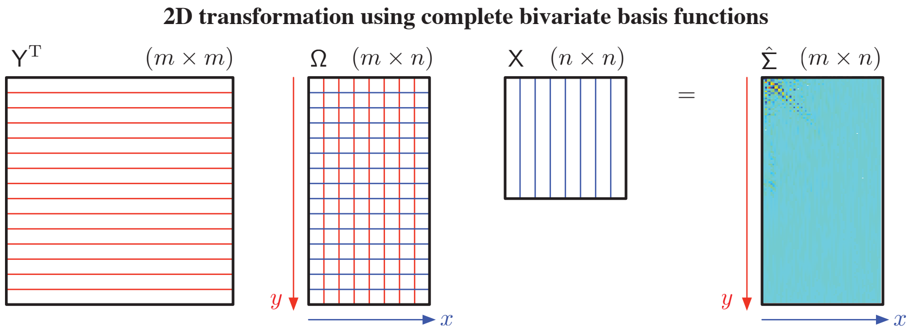

2D transformation using complete bivariate basis functions

Fig. 9.1 The bivariate basis functions X and Y enable 2D transformations along the x- and y-axis respectively in an anisotropic manner. The analysis process of the data matrix Ω yields the associated spectrum Σ. This visualization is valid for complete basis functions, whereby m are the number of points in y-dimension (rows) and n are the number of points in x-dimension (columns).

Example (1): Metal Surface Description

The practical relevance of this example has been presented in the bachelor thesis [69, 168]. Steel bars are marked permanently with robust 3D embossed digits to enable tracking and tracing during the complete production process. The embossed digits are placed on the front face of the steel bars, which is produced during the cutting process of the steel slabs. Unfortunately, the cut section's surface is very rough, cragged and rippled. The goal is to compute a surface approximation to effectively remove the surface flaws which negatively influence the digit classification while conserving the information about the embossed digits.

The data of the original 3D scan is stored in the matrix $\Omega \in \mathbb{R}^{(m \times n)}$. The metal's surface $\hat{\Omega}$ is a low-degree approximation of Ω of same dimensions. The reduced surface Φ is then computed by

$$\begin{aligned} \Phi &= \Omega - \hat{\Omega}, \\ &= \Omega - \hat{\mathsf{Y}}\,\hat{\Sigma}\,\hat{\mathsf{X}}^{+}, \\ &= \Omega - \hat{\mathsf{Y}}\hat{\mathsf{Y}}^{+}\,\Omega\,\hat{\mathsf{X}}\hat{\mathsf{X}}^{+}, \end{aligned} \tag{9.60}$$

where $\hat{\mathsf{X}}$ and $\hat{\mathsf{X}}$ are incomplete sets of generic basis functions. When using incomplete DOP basis functions $\hat{\mathsf{G}}_{\mathsf{X}} \triangleq \hat{\mathsf{X}}$ and $\hat{\mathsf{G}}_{\mathsf{Y}} \triangleq \hat{\mathsf{Y}}$ in x- and y-dimension, the reduced spectrum $\hat{\Sigma}$ is computed via $\hat{\Sigma} = \hat{\mathsf{Y}}^{\mathrm{T}}\,\Omega\,\hat{\mathsf{X}}$. It is worthy to note in this context, that $\hat{\mathsf{X}} \in \mathbb{R}^{(n \times (d_{\mathsf{X}}+1))}$ and $\hat{\mathsf{Y}} \in \mathbb{R}^{(m \times (d_{\mathsf{Y}}+1))}$ can be anisotropic, i.e., the computation is valid for different degrees $d_{\mathsf{X}} \neq d_{\mathsf{Y}}$ or even different kinds of basis functions. The reduced surface Φ is then acquired via

$$\Phi = \Omega - \hat{\mathsf{Y}}\hat{\mathsf{Y}}^{\mathrm{T}}\,\Omega\,\hat{\mathsf{X}}\hat{\mathsf{X}}^{\mathrm{T}} \tag{9.61}$$

when using DOP bases. Conventional image processing techniques are applied after this approximation step: Φ is binarized and prepared to be classified by an optical character recognition (OCR) algorithm, the process is visualized in Fig. 9.2. The next example shows an alternative approach to classical OCR software.

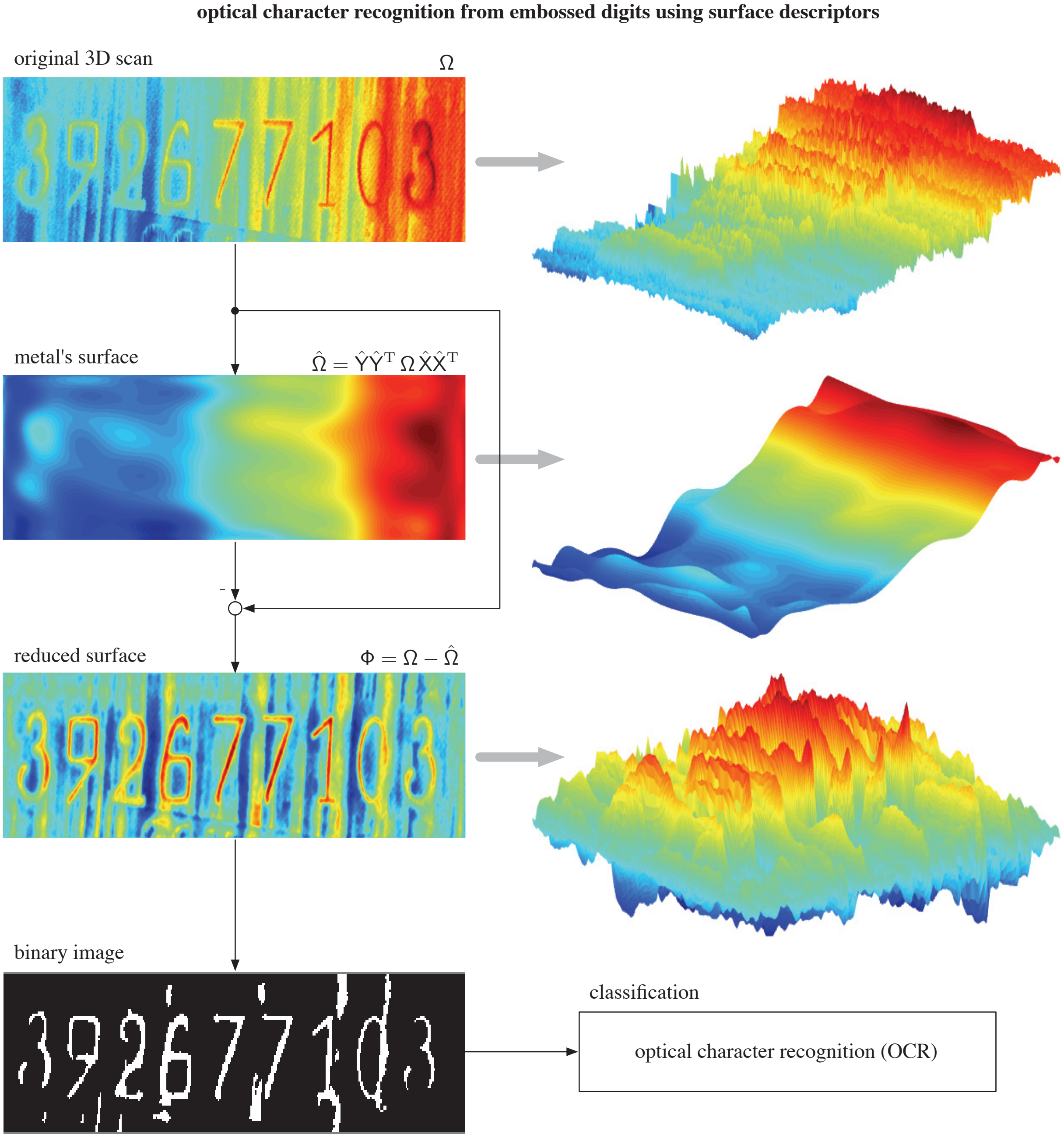

optical character recognition from embossed digits using surface descriptors
original 3D scan
Ω
metal's surface
Ω̂ = ŶŶ^T Ω X̂X̂^T
-
reduced surface
Φ = Ω − Ω̂
binary image
classification
optical character recognition (OCR)

Fig. 9.2 The left side of the illustration shows the data sets as 2D plots and the right side shows the corresponding surface plots for better understanding. The metal's surface is very rough; furthermore, the original 3D scan data Ω is heavily perturbed. The reduced surface Φ is acquired by subtracting a low-degree approximation $\hat{\Omega}$ of the metal's surface from the original surface scan Ω. The process effectively removes the surface flaws which negatively affect the digit recognition. The binarized image is then ready to be processed by a conventional OCR algorithm.

Example (2): Spectral Analysis of Embossed Digits

This example is directly related to the previous example and is described in detail in [69]. The work investigates the application of DOP basis functions to enable automatic recognition of the embossed digits. State-of-the-art machine learning algorithms for visual pattern recognition directly process the image data. In contrast to the conventional approach, the presented method uses the digits' spectra to construct a *support vector machine* (SVM) classifier. An SVM is described analytically, which makes it superior in comparison to other non-deterministic machine learning algorithms such as neural networks or meta-heuristics. According to Parseval's theorem [153], the energy within the spectrum is equal to the energy in the time/space domain, i.e., an image's spectrum contains the same information as the original image. When using unitary bases, the Gaussian noise is evenly spread over the complete spectrum. Cutting off higher degree polynomials effectively improves the signal-to-noise ratio via spectral regularization, which is advantageous in two aspects: the total data to be processed is decreased while the relevant information, i.e., the features, become more significant.

Fig. 9.3 illustrates the process of spectral analysis and synthesis. The complete image data Ω of the 3D scan has a resolution of $(m \times n) = (128\,[\text{pixel}] \times 64\,[\text{pixel}])$ in this worked example. The image is described by the discrete unitary Gram bases $\mathsf{G}_\mathsf{X} \triangleq \mathsf{X} \in \mathbb{R}^{(n\times n)}$ in x-dimension and $\mathsf{G}_\mathsf{Y} \triangleq \mathsf{Y} \in \mathbb{R}^{(m\times m)}$ in y-dimension; hence, the analysis is formulated as

$$\Sigma = \mathsf{Y}^\mathrm{T}\Omega\mathsf{X}. \tag{9.62}$$

The spectrum Σ contains the coefficients of the bivariate unitary basis functions. The bases are complete, i.e., $d_\mathsf{X} = n - 1$ and $d_\mathsf{Y} = m - 1$, and, hence, fully energy conservative. Reducing the degree $d = d_\mathsf{X} = d_\mathsf{Y}$ of the polynomial approximation is equivalent to low-pass filtering; this yields the reduced bases $\hat{\mathsf{X}} \in \mathbb{R}^{(n\times(d_\mathsf{X}+1))}$ and $\hat{\mathsf{Y}} \in \mathbb{R}^{(m\times(d_\mathsf{Y}+1))}$. As a result,

$$\hat{\Sigma} = \hat{\mathsf{Y}}^\mathrm{T}\Omega\hat{\mathsf{X}}, \tag{9.63}$$

since $\hat{\mathsf{X}}^+ = \hat{\mathsf{X}}^\mathrm{T}$ and $\hat{\mathsf{Y}}^+ = \hat{\mathsf{Y}}^\mathrm{T}$ for incomplete unitary bases. The reduced spectrum $\hat{\Sigma} \in \mathbb{R}^{((d+1)\times(d+1))}$ is fed into the SVM learning algorithm in order to construct a classifier. Numerical experiments in [69] showed, that a symmetric spectrum of $\hat{\Sigma} \in \mathbb{R}^{(30\times30)}$ holds enough information for robust digit classification. The spectral compactness C is then

$$C = \frac{(d+1)^2}{m\,n} = \frac{30^2}{128 \cdot 64} = 0.1099 \approx 11\%. \tag{9.64}$$

The result implies, that only approximately 11% of the original data is necessary to recognize an embossed digit. Moreover, approximately 89% of the Gaussian noise has been eliminated. This formidably demonstrates the advantages of this approach. The synthesis of the reduced image $\hat{\Omega}$ is not necessary for the actual machine learning process, nevertheless it demonstrates the principle of operation. As shown in Fig. 9.3, the synthesis is performed via

$$\hat{\Omega} = \hat{\mathsf{Y}}\hat{\Sigma}\hat{\mathsf{X}}^\mathrm{T}. \tag{9.65}$$

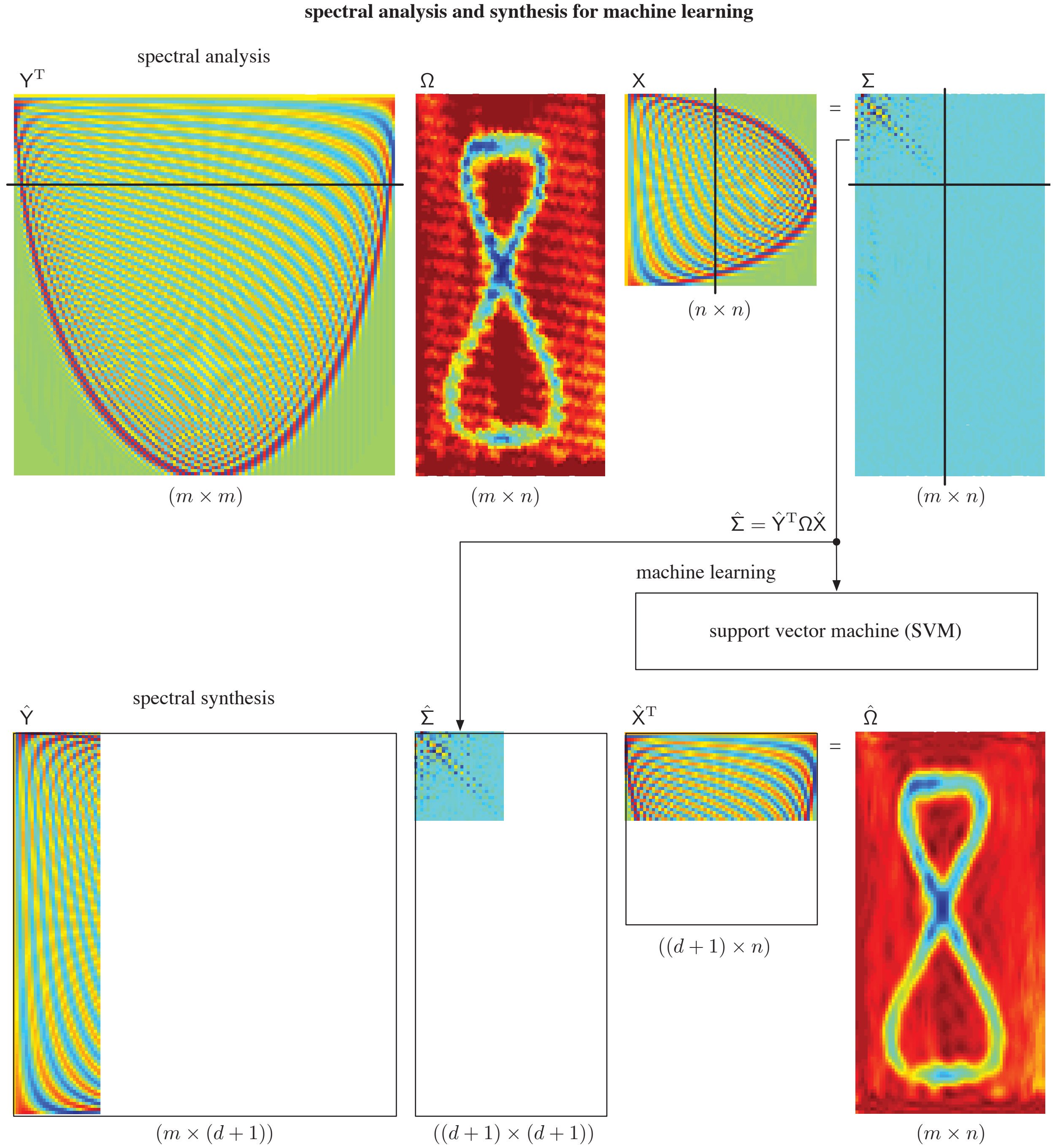


Fig. 9.3 Spectral analysis is performed via $\Sigma = \mathsf{Y}^{\mathrm{T}}\Omega\,\mathsf{X}$ using the unitary bases $\mathsf{X} \in \mathbb{R}^{(n\times n)}$ and $\mathsf{Y} \in \mathbb{R}^{(m\times m)}$. The complete spectrum $\Sigma \in \mathbb{R}^{(m\times n)}$ has the same dimensions as the original image $\Omega \in \mathbb{R}^{(m\times n)}$. Decreasing the degree d of the polynomial bases $\hat{\mathsf{X}} \in \mathbb{R}^{(n\times(d_{\mathsf{X}}+1))}$ and $\hat{\mathsf{Y}} \in \mathbb{R}^{(m\times(d_{\mathsf{Y}}+1))}$ yields the reduced spectrum $\hat{\Sigma} = \hat{\mathsf{Y}}^{\mathrm{T}}\Omega\,\hat{\mathsf{X}}$, which is then used to train the SVM. Synthesis of the reduced image is computed via $\hat{\Omega} = \hat{\mathsf{Y}}\,\hat{\Sigma}\,\hat{\mathsf{X}}^{\mathrm{T}}$. This approach increased the information density while simultaneously removing the majority of Gaussian noise.

10 Differential Equations

The concept of DOP basis functions G enables the efficient synthesis of their derivatives G'. Both, basis functions and their derivatives, are required in order to solve ordinary differential equations (ODE) and partial differential equations (PDE) as well as associated inverse problems. The derivative G' w.r.t. $\boldsymbol{x}$ can directly be computed from the generalized recurrence relationship as stated in Eqn. (7.2). The n^{th} monomial $\boldsymbol{g}_n$ is generated via

$$\boldsymbol{g}_n = \alpha \left(\boldsymbol{g}_{n-1} \circ \boldsymbol{x} \right) + \mathsf{G}_{n-1} \boldsymbol{\beta} \tag{10.1}$$

while its derivative $\boldsymbol{g}'_n$ is obtained as

$$\boldsymbol{g}'_n = \alpha \left(\boldsymbol{g}'_{n-1} \circ \boldsymbol{x} + \boldsymbol{g}_{n-1} \circ \boldsymbol{x}' \right) + \mathsf{G}'_{n-1} \boldsymbol{\beta}. \tag{10.2}$$

The lower degree derivatives of the monomials are concatenated to yield G'. Note, that the derivatives are not orthogonal. The values for α and $\boldsymbol{\beta}$ are already computed during the synthesis of the basis function as described in Chapter 7. Consequently, the derivatives G' are generated simultaneously with the corresponding basis function G and marginal additional numerical effort [165].

10.1 Linear Differential Operator

A *differentiating matrix* D computes the numerical derivatives $\boldsymbol{y}' \in \mathbb{R}^{(m \times 1)}$ of a vector $\boldsymbol{y} \in \mathbb{R}^{(m \times 1)}$ for each entry of the m nodes,

$$\boldsymbol{y}' = \mathsf{D}\, \boldsymbol{y}. \tag{10.3}$$

The matrix D is a linear operator, $\mathsf{D} \in \mathbb{R}^{(m \times m)}$ is required to formulate algebraic methods for the solution of ODE and PDE. The generation of a sufficiently accurate D is critical to the quality of the overall solution. Given a set of basis functions $\mathsf{G} \in \mathbb{R}^{(m \times (d+1))}$ of degree d and their derivatives $\mathsf{G}' \in \mathbb{R}^{(m \times (d+1))}$, they are related by the differentiating matrix D,

$$\mathsf{G}' = \mathsf{D}\, \mathsf{G}. \tag{10.4}$$

In general, differentiating matrices should be rank-1, and only rank-1, deficient; thus, it should incorporate a single null vector. However, differentiating matrices generally become degenerate if computed globally for a large number of points and a high polynomial degree; hence, the condition number $\kappa = \text{cond}\left(\mathsf{D}\right)$ increases. This is a fundamental property of such matrices and cannot be avoided even with methods of high numerical quality [165]. Inevitable numerical quantification errors introduce a higher-dimensional null-space when the differentiating matrix collapses and the rank-1 deficiency cannot be hold. The effect is shown in Fig. 10.1. The higher-dimensional null space is a problem when solving inverse problems with finite numerical resolution, because the solution becomes inexact and non-unique.

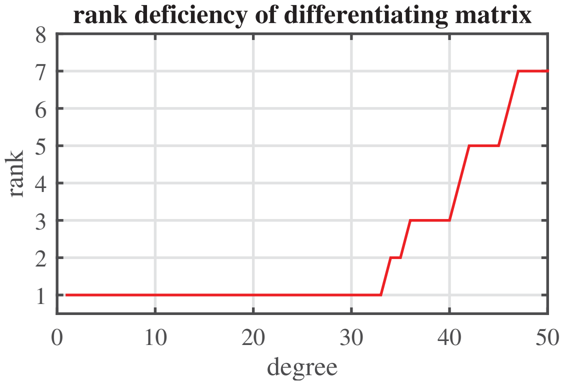


Fig. 10.1 The plot illustrates the rank deficiency of the differentiating matrix D when synthesized according to Eqn. (10.5) as a function of the degree; double numerical accuracy is used. D is rank-1 deficient until a degree of 33 is reached, i.e., the process is stable so far. When further increasing the degree, D degenerates and the dimensionality of the null space increases as well, which is disadvantageous for solving inverse problems.

As a consequence to the instability of a global D, a *regularizing linear differential operator* $\hat{\mathsf{D}}$ is introduced, which is defined as

$$\hat{\mathsf{D}} \triangleq \mathsf{G}'\mathsf{G}^{\mathrm{T}} = \mathsf{D}\mathsf{G}\mathsf{G}^{\mathrm{T}}, \tag{10.5}$$

where $\mathsf{G} \in \mathbb{R}^{(l_s \times (d+1))}$ is an incomplete DOP basis function of degree d and spans a support length $l_s < m$, $\mathsf{G}\mathsf{G}^{\mathrm{T}}$ is the projection onto the basis function G. If the basis is complete, then $\mathsf{G}\mathsf{G}^{\mathrm{T}} = \mathsf{I}$ and $\hat{\mathsf{D}} = \mathsf{D}$. The strength of the regularizing linear differential operator is best seen when estimating derivatives in the presence of noise. Regularization is advantageous in terms of reducing the numerical effort and simultaneously increasing the stability of the solution. The procedure is fundamentally the same as the method for local polynomial approximation presented in Section 8.3.

As already mentioned, the constant vector can be seen as the numerical equivalent to the constant of integration, i.e., $\mathsf{D}\,\boldsymbol{1}\gamma = \boldsymbol{0}$. Consider the cost function K,

$$K = \|\mathsf{D}\,\boldsymbol{y} - \boldsymbol{y}'\|_2^2, \tag{10.6}$$

solving the equation in a least squares manner w.r.t. $\boldsymbol{y}$ yields the estimate $\hat{\boldsymbol{y}}$,

$$\hat{\boldsymbol{y}} = \mathsf{D}^{+}\boldsymbol{y}' + \mathrm{null}\,(\mathsf{D})\ \boldsymbol{\gamma}, \tag{10.7}$$

where an ideal approximation should be

$$\hat{\boldsymbol{y}} = \mathsf{D}^{+}\boldsymbol{y}' + \gamma\,\boldsymbol{1}, \tag{10.8}$$

i.e., $\mathrm{null}\,(\mathsf{D})\ \boldsymbol{\gamma} = \gamma\,\boldsymbol{1}$ for a perfectly conditioned numerical variational integrator D^{+}. The constant of integration is represented by the scalar γ or the vector $\boldsymbol{\gamma}$ respectively. The presented approach is not limited to spectral regularization, e.g., Tikhonov regularization is implemented by adding a penalty term to the cost function,

$$T = \|\mathsf{D}\,\boldsymbol{y} - \boldsymbol{y}'\|_2^2 + \lambda\,\|\mathsf{S}\,\boldsymbol{y}\|_2^2. \tag{10.9}$$

Tikhonov regularization penalizes outliers via the additional cost function term $\lambda\,\|\mathsf{S}\,\boldsymbol{y}\|_2^2$, where the regularization parameter λ is a positive constant; its selection is commonly an iterative task. The combination of both approaches, Tikhonov regularization and spectral regularization, has been described in [165].

10.2 Ordinary Differential Equations

The class of inverse problems being considered in this chapter are derived from repetitive measurements of inclinations; associated examples are described in Chapter 12. The m measurements, forming the vector $\boldsymbol{z} \in \mathbb{R}^{(m\times 1)}$, correspond to discrete samples of the continuous forcing function $z(x)$. They may emanate from m sensors forming a spatial array or from a time sequence of m measurements from one single sensor. In the case of a chain of sensors, the physical position of the sensor corresponds to x. In temporal sequences, it is the time points of the individual measurements which define x. The positions, or nodes, $\boldsymbol{x}$, where the solutions are required, are determined by the measurement application's requirements. As a matter of fact, the positions of where to solve the ODE cannot be chosen freely. This precludes the use of variable step size algorithms such as Runge-Kutta methods. A further consequence is that a general framework for this type of inverse problem has to be capable of computing the solution for arbitrarily placed nodes.

10.2.1 Continuous Measurement Model

In this class of problems, the forcing function $z(x)$, the input, is considered to be perturbed, since it is formed from measurements which are subject to noise. Furthermore, only the values in the vector $\boldsymbol{z}$ change from one measurement to the next. The associated measurement model consist of an n^{th} order ODE of the form

$$a_n(x)\, y^{(n)} + a_{n-1}(x)\, y^{(n-1)} + \ldots + a_1(x)\, y' + a_0(x)\, y = z(x), \tag{10.10}$$

where $y = y(x)$ is a function of x, $y^{(i)}$ is the i^{th} derivative of y w.r.t. x, $a_i(x)$ are the coefficient functions and $z(x)$ is the forcing function. Additionally, a minimum of n independent initial-, inner- or boundary value constraints are required to ensure that there is a unique solution to the differential equation; moreover, the method is capable of dealing with overconstrained systems, i.e., there are $p \geq n$ independent constraints. The constraints correspond to the function value $y(x)$ or its derivatives $y^{(i)}(x)$ at specific x locations. Both, Dirichlet and Neumann boundary conditions, are special cases of such constraints. The nature of the constraints determines if the system is considered to be an initial value (IVP), boundary value (BVP) or inner value problem.

The numerical solution of an inverse problem requires the discrete approximation of a continuous system. Properties of the continuous operations are derived, which must be fulfilled by the corresponding discrete operators. The continuous domain differentiating operator D is defined such that $D^{(i)}\, y \equiv y^{(i)}$. The discrete equivalent of the linear differentiating operator is implemented using polynomial interpolation. The properties of D w.r.t. a polynomial are essential to the desired behavior of numerical differentiation. Defining a power series approximation for y with coefficients β_i,

$$y = \sum_{i=0}^{m} \beta_i\, x^i, \tag{10.11}$$

and applying the differentiating operator D yields

$$y' = D\, y = \sum_{i=1}^{m} i\, \beta_i\, x^{i-1}. \tag{10.12}$$

By definition of the derivative, the constant portion of the polynomial differentiates to zero; hence, the constant coefficient β_0 vanishes. It is assumed, that D is composed of formulae which are consistent, in the sense that in the limit they define a derivative. If this is the case, then the matrix D satisfies the following properties, such that D is a consistent discrete approximation to the continuous operator D:

1. The matrix D must be exactly rank-1 deficient; i.e, its null space is of dimension one;

2. The null space of D must be spanned by the constant vector $\gamma\, \boldsymbol{1}$; equivalently, the row-sums of D are all zero,
$$\mathsf{D}\, \boldsymbol{1} \gamma = \boldsymbol{0}. \tag{10.13}$$

These conditions are fundamentally important to linear differentiating operators as already discussed in Section 10.1. They ensure that the differentiating matrix D is consistent with the continuous domain definition of the derivative. The general notion of a linear differentiating operator for treating ODE is used according to [116]. Substitution of the continuous differentiating operator D for the differentials $y^{(i)}$ in Eqn. (10.10),

$$a_n(x)\, D^{(n)}\, y + a_{n-1}(x)\, D^{(n-1)}\, y + \ldots + a_1(x)\, D\, y + a_0(x)\, y = z(x), \tag{10.14}$$

and factoring y to the right,

$$\left\{a_n(x)\, D^{(n)} + a_{n-1}(x)\, D^{(n-1)} + \ldots + a_1(x)\, D + a_0(x)\right\}\, y = z(x), \tag{10.15}$$

yields the linear differential operator L for the continuous equation, which is now defined as

$$L \triangleq a_n(x)\, D^{(n)} + a_{n-1}(x)\, D^{(n-1)} + \ldots + a_1(x)\, D + a_0(x). \tag{10.16}$$

Consequently, Eqn. (10.10) is rewritten as

$$L\, y = z(x), \tag{10.17}$$

whereby a constraint at the point x is formulated as

$$D^{(i)}\, y(x) = d_i. \tag{10.18}$$

The continuous linear differential operator L in Eqn. (10.16) is discretized as the matrix L,

$$\mathsf{L} \triangleq \mathsf{A}_n\, \mathsf{D}_n + \mathsf{A}_{n-1}\, \mathsf{D}_{n-1} + \ldots + \mathsf{A}_1\, \mathsf{D} + \mathsf{A}_0, \tag{10.19}$$

where $\mathsf{A}_i = \mathrm{diag}(a_i(\boldsymbol{x}))$ and the matrix $\mathsf{D}_i \in \mathbb{R}^{(m \times m)}$ is a local discrete approximation with support length l_s to the continuous differentiating operator $D^{(i)}$. As a result, $\mathsf{L} \in \mathbb{R}^{(m \times m)}$ is a regularizing differential operator.

10.2.2 Efficient Numerical Solution

Consider the residual vector $\boldsymbol{r}$, which is computed via

$$\boldsymbol{r} = \mathsf{L}\, \boldsymbol{y} - \boldsymbol{z}, \tag{10.20}$$

where $\boldsymbol{y} \in \mathbb{R}^{(m\times 1)}$ is the vector of function values with m nodes, $\boldsymbol{z} \in \mathbb{R}^{(m\times 1)}$ is the vector of measured inclinations and $\mathsf{L} \in \mathbb{R}^{(m\times m)}$ is a linear differential operator as defined in Eqn. (10.19). The goal is to acquire the function values $\boldsymbol{y}$ from the perturbed measurements $\boldsymbol{z}$ via the relation $\boldsymbol{z} \approx \mathsf{L}\,\boldsymbol{y}$, which represents a perturbed ODE. Such a problem is ill-posed, because it incorporates an infinite number of solutions. Furthermore, the measurement's output is known, where actually the input which caused this output is sought. By definition, this is an inverse problem [202]. The problem to be solved is defined by the cost function $K = \|\boldsymbol{r}\|_2^2 = \|\boldsymbol{r}^{\mathrm{T}}\boldsymbol{r}\|_2$,

$$K = \|\mathsf{L}\,\boldsymbol{y} - \boldsymbol{z}\|_2^2, \tag{10.21}$$

which must be minimized to acquire the least squares solution. In order to ensure a unique solution, constraints must be placed onto the system, which are

$$C_1: \quad \mathsf{C}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{d} \quad \text{and} \quad C_2: \quad \boldsymbol{y} = \mathsf{G}\,\boldsymbol{\alpha}, \tag{10.22}$$

where C_1 defines the matrix of constraints $\mathsf{C}^{\mathrm{T}} \in \mathbb{R}^{(p\times m)}$ and the vector $\boldsymbol{d} \in \mathbb{R}^{(p\times 1)}$ contains the values of the p constraints, i.e., C_1 describes the physical structure of the observed system. Note, that an ODE of n^{th} degree requires at least $p \geq n$ constraints in order to acquire a unique solution; therefore, the condition

$$\operatorname{rank}\begin{bmatrix}\mathsf{L}\\ \mathsf{C}^{\mathrm{T}}\end{bmatrix} \geq n \tag{10.23}$$

must be fulfilled. C_2 defines the incomplete orthogonal basis function $\mathsf{G} \in \mathbb{R}^{(m\times(d+1))}$ of degree d and the vector of the associated coefficients $\boldsymbol{\alpha} \in \mathbb{R}^{((d+1)\times 1)}$. For instance, the basis functions may consist of the eigenfunctions of the structure being monitored. In this case, the method delivers a reconstructed curve as a linear combination of the system's modes. This leads naturally to the use of truncated basis functions as a means of spectral regularization. Solving the algebraic equation from Eqn. (10.21) with C_1 and C_2 provides the most general solution of the inverse problem. Substitution of $C_2 : \boldsymbol{y} = \mathsf{G}\,\boldsymbol{\alpha}$ in $C_1 : \mathsf{C}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{d}$ yields

$$\mathsf{C}^{\mathrm{T}}\mathsf{G}\,\boldsymbol{\alpha} = \boldsymbol{d} \tag{10.24}$$

and solving for the estimate $\hat{\boldsymbol{\alpha}}$ of the coefficients $\boldsymbol{\alpha}$,

$$\hat{\boldsymbol{\alpha}} = \{\mathsf{C}^{\mathrm{T}}\mathsf{G}\}^{+}\boldsymbol{d} + \mathsf{N}\,\boldsymbol{\gamma}, \tag{10.25}$$

where $\{\mathsf{C}^{\mathrm{T}}\mathsf{G}\}^{+} \in \mathbb{R}^{((d+1)\times p)}$ is the pseudoinverse of the least squares solution, $\mathsf{N} \in \mathbb{R}^{((d+1)\times p)}$ is an orthonormal basis set of the null space of $\{\mathsf{C}^{\mathrm{T}}\mathsf{G}\}$, i.e., $\mathsf{N}^{\mathrm{T}}\mathsf{N} = \mathsf{I}$ and $\operatorname{range}(\mathsf{N}) = \operatorname{range}(\operatorname{null}(\mathsf{C}^{\mathrm{T}}\mathsf{G}))$, the vector $\boldsymbol{\gamma} \in \mathbb{R}^{(p\times 1)}$ contains the coefficients of the basis set N. Substitution of the coefficient vector's estimate $\hat{\boldsymbol{\alpha}}$ into $C_2 : \hat{\boldsymbol{y}} = \mathsf{G}\,\hat{\boldsymbol{\alpha}}$ gives the curve's estimate $\hat{\boldsymbol{y}}$,

$$\hat{\boldsymbol{y}} = \mathsf{G}\,\{\mathsf{C}^{\mathrm{T}}\mathsf{G}\}^{+}\boldsymbol{d} + \mathsf{G}\,\mathsf{N}\,\boldsymbol{\gamma}, \tag{10.26}$$

and using $\hat{\boldsymbol{y}}$ in the cost function K from Eqn. (10.21) yields

$$K = \|\mathsf{L}\mathsf{G}\,\{\mathsf{C}^{\mathrm{T}}\mathsf{G}\}^{+}\boldsymbol{d} + \mathsf{L}\mathsf{G}\,\mathsf{N}\,\boldsymbol{\gamma} - \boldsymbol{z}\|_2^2. \tag{10.27}$$

Solving the equation for $\boldsymbol{\gamma}$ in a least squares sense,

$$\boldsymbol{\gamma} = \{\mathsf{L}\mathsf{G}\,\mathsf{N}\}^{+}\left(\boldsymbol{z} - \mathsf{L}\mathsf{G}\,\{\mathsf{C}^{\mathrm{T}}\mathsf{G}\}^{+}\boldsymbol{d}\right), \tag{10.28}$$

where $\{\mathsf{LG\,N}\}^+ \in \mathbb{R}^{(p\times m)}$. Substitution of $\boldsymbol{\gamma}$ into Eqn. (10.25),

$$\hat{\boldsymbol{\alpha}} = \{\mathsf{C}^{\mathrm{T}}\mathsf{G}\}^+\boldsymbol{d} + \mathsf{N}\,\{\mathsf{LG\,N}\}^+\,\left(\boldsymbol{z} - \mathsf{LG}\,\{\mathsf{C}^{\mathrm{T}}\mathsf{G}\}^+\boldsymbol{d}\right), \tag{10.29}$$

and combining this with $C_2 : \hat{\boldsymbol{y}} = \mathsf{G}\,\hat{\boldsymbol{\alpha}}$ yields the final result for the curve reconstruction $\hat{\boldsymbol{y}}$,

$$\hat{\boldsymbol{y}} = \mathsf{G}\,\{\mathsf{C}^{\mathrm{T}}\mathsf{G}\}^+\boldsymbol{d} + \mathsf{G\,N}\,\{\mathsf{LG\,N}\}^+\,\left(\boldsymbol{z} - \mathsf{LG}\,\{\mathsf{C}^{\mathrm{T}}\mathsf{G}\}^+\boldsymbol{d}\right). \tag{10.30}$$

For easier reading, it is defined,

$$\mathsf{K} \triangleq \mathsf{G}\,\{\mathsf{C}^{\mathrm{T}}\mathsf{G}\}^+ \quad \text{and} \quad \mathsf{P} \triangleq \mathsf{G\,N}\,\{\mathsf{LG\,N}\}^+ \tag{10.31}$$

with $\mathsf{K} \in \mathbb{R}^{(m\times p)}$ and $\mathsf{P} \in \mathbb{R}^{(m\times m)}$. Substitution of the terms,

$$\begin{aligned}\hat{\boldsymbol{y}} &= \mathsf{K}\,\boldsymbol{d} + \mathsf{P}\,\left(\boldsymbol{z} - \mathsf{L\,K}\,\boldsymbol{d}\right),\\ &= \mathsf{K}\,\boldsymbol{d} + \mathsf{P}\,\boldsymbol{z} + \mathsf{P\,L\,K}\,\boldsymbol{d},\end{aligned} \tag{10.32}$$

and collecting for $\boldsymbol{d}$ as well as $\boldsymbol{z}$ yields

$$\begin{aligned}\hat{\boldsymbol{y}} &= \left(\mathsf{K} - \mathsf{P\,L\,K}\right)\,\boldsymbol{d} + \mathsf{P}\,\boldsymbol{z},\\ &= \left(\mathsf{I} - \mathsf{P\,L}\right)\,\mathsf{K}\,\boldsymbol{d} + \mathsf{P}\,\boldsymbol{z},\end{aligned} \tag{10.33}$$

whereby it is defined that

$$\mathsf{H} \triangleq \left(\mathsf{I} - \mathsf{P\,L}\right)\,\mathsf{K} \tag{10.34}$$

with $\mathsf{H} \in \mathbb{R}^{(m\times p)}$. Therefore, the equation is rewritten in concise form,

$$\hat{\boldsymbol{y}} = \mathsf{H}\,\boldsymbol{d} + \mathsf{P}\,\boldsymbol{z}, \tag{10.35}$$

where $\boldsymbol{y}_{\mathrm{h}} \triangleq \mathsf{H}\,\boldsymbol{d}$ is the homogeneous solution and $\boldsymbol{y}_{\mathrm{p}} \triangleq \mathsf{P}\,\boldsymbol{z}$ is the particular solution. The final solution for the inverse problem, i.e., the curve reconstruction $\hat{\boldsymbol{y}}$, is then formulated as

$$\hat{\boldsymbol{y}} = \boldsymbol{y}_{\mathrm{h}} + \boldsymbol{y}_{\mathrm{p}}. \tag{10.36}$$

The matrices K, H and P are independent from the measurement data; they are determined by the kind of the sensors being used and their positioning on the structure. As a consequence, these matrices as well as the homogeneous solution $\boldsymbol{y}_{\mathrm{h}}$ can be computed a-priori during the calibration procedure. Only the particular solution $\boldsymbol{y}_{\mathrm{p}}$ must be computed during the measurement's run-time. As a matter of fact, the necessary operation to solve such a perturbed differential equation is reduced to a single matrix-vector multiplication and a vector addition, i.e.,

$$\hat{\boldsymbol{y}} = \boldsymbol{y}_{\mathrm{h}} + \mathsf{P}\,\boldsymbol{z}, \tag{10.37}$$

where only the vector of measurements $\boldsymbol{z}$ is subject to change with each new measurement. The exact number of floating point operations (FLOPS) is known a-priori, which makes this approach suitable for real-time applications by definition [214]. Given m measurement values, the resulting computational cost $W(m)$ is

$$W(m) = m + m^2, \tag{10.38}$$

which is of $\mathcal{O}(m^2)$ complexity. Knowing, that the vector of measurements $\boldsymbol{z}$ is perturbed, facilitates the computation of the reconstructed curve's covariance $\Lambda_{\hat{\boldsymbol{y}}} = \mathsf{L}\,\Lambda_{\boldsymbol{z}}\,\mathsf{L}^{\mathrm{T}}$ with the linear operation $\mathsf{L} \triangleq \mathsf{P}$, i.e.,

$$\Lambda_{\hat{\boldsymbol{y}}} = \mathsf{P}\,\Lambda_{\boldsymbol{z}}\,\mathsf{P}^{\mathrm{T}}. \tag{10.39}$$

As an alternative, the error vector $\boldsymbol{\epsilon}$ can be computed for each measurement as the difference between the forward and inverse problem,

$$\boldsymbol{\epsilon} = \boldsymbol{z} - \mathsf{L}\,(\boldsymbol{y}_{\mathrm{h}} + \mathsf{P}\,\boldsymbol{z})\,. \tag{10.40}$$

Applying a Kolmogorov-Smirnov test onto $\boldsymbol{\epsilon}$ reveals if the error is Gaussian and if all systematic errors have been removed. Unfortunately, the residual $\boldsymbol{r} = \boldsymbol{y} - \hat{\boldsymbol{y}}$ cannot be computed since the values of the true curve $\boldsymbol{y}$ remain unknown to the measurement system. Nevertheless, evaluating Eqn. (10.40) yields additional information on the suitability of the model for the specific measurement. The complete process is shown in Fig. 10.2.

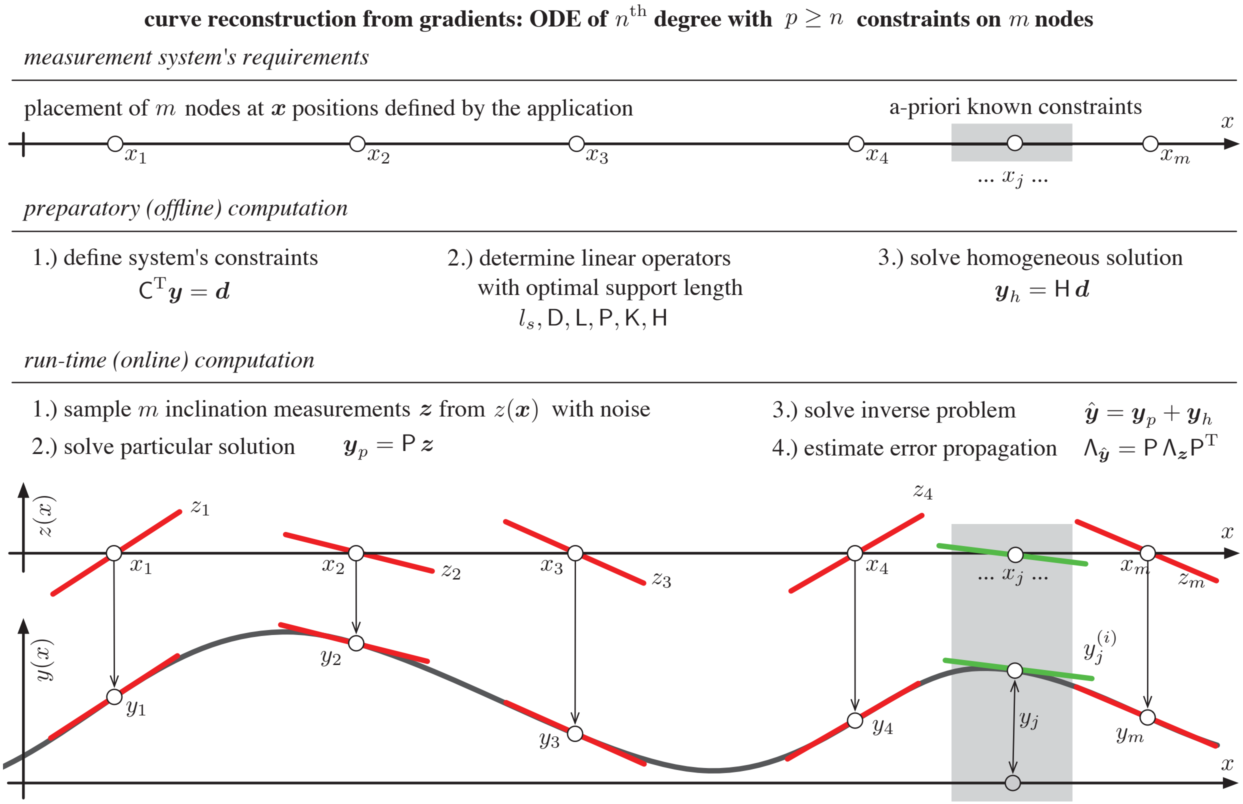

curve reconstruction from gradients: ODE of n^{th} degree with $p \geq n$ constraints on m nodes
measurement system's requirements
placement of m nodes at $\boldsymbol{x}$ positions defined by the application
a-priori known constraints
x
x_1
x_2
x_3
x_4
... x_j ...
x_m
preparatory (offline) computation
1.) define system's constraints
$\mathsf{C}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{d}$
2.) determine linear operators with optimal support length
$l_s, \mathsf{D}, \mathsf{L}, \mathsf{P}, \mathsf{K}, \mathsf{H}$
3.) solve homogeneous solution
$\boldsymbol{y}_h = \mathsf{H}\,\boldsymbol{d}$
run-time (online) computation
1.) sample m inclination measurements $\boldsymbol{z}$ from $z(\boldsymbol{x})$ with noise
2.) solve particular solution $\boldsymbol{y}_p = \mathsf{P}\,\boldsymbol{z}$
3.) solve inverse problem $\hat{\boldsymbol{y}} = \boldsymbol{y}_p + \boldsymbol{y}_h$
4.) estimate error propagation $\Lambda_{\hat{\boldsymbol{y}}} = \mathsf{P}\,\Lambda_{\boldsymbol{z}}\mathsf{P}^{\mathrm{T}}$
$z(x)$
z_1
z_2
z_3
z_4
z_m
$y(x)$
y_1
y_2
y_3
y_4
$y_j^{(i)}$
y_j
y_m

Fig. 10.2 There is an array of m sensors mounted on a large physical structure at the defined positions $\boldsymbol{x}$. A number of $p \geq n$ constraints are placed on the system which enable preparatory computations. During run-time computation, the curve reconstruction $\hat{\boldsymbol{y}}$ is computed as a maximum likelihood estimation for the true curve $\boldsymbol{y}$ in real-time from the perturbed inclinometer measurements $\boldsymbol{z}$. This is an inverse problem associated to an ODE of n^{th} degree.

10.2.3 Homogeneous Initial Value Problem

Homogeneously constrained initial value problems (IVPs) are a special subclass of the presented problem. Consider the cost function K from Eqn. (10.21),

$$K = \|\mathsf{L}\,\boldsymbol{y} - \boldsymbol{z}\|_2^2, \tag{10.41}$$

whereby the constraint C_1 is simplified and C_2 remains unchanged,

$$C_1 : \quad \mathsf{C}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{0} \quad \text{and} \quad C_2 : \quad \boldsymbol{y} = \mathsf{G}\,\boldsymbol{\alpha}. \tag{10.42}$$

In other words, the system is unconstrained. Substitution of C_2 into K yields

$$K = \|\mathsf{L}\,\mathsf{G}\,\boldsymbol{\alpha} - \boldsymbol{z}\|_2^2. \tag{10.43}$$

Solving for the estimate of the coefficients $\hat{\boldsymbol{\alpha}}$ gives

$$\hat{\boldsymbol{\alpha}} = \{\mathsf{L}\,\mathsf{G}\}^{+}\boldsymbol{z}. \tag{10.44}$$

A unique solution exists since null$(\mathsf{L}\,\mathsf{G}) = \boldsymbol{0}$. The simplified computation for the reconstruction $\hat{\boldsymbol{y}}$ of the curve $\boldsymbol{y}$ is, using $C_2 : \hat{\boldsymbol{y}} = \mathsf{G}\,\hat{\boldsymbol{\alpha}}$,

$$\hat{\boldsymbol{y}} = \mathsf{G}\,\{\mathsf{L}\,\mathsf{G}\}^{+}\boldsymbol{z} \tag{10.45}$$

for a homogeneously constrained IVP.

Part III

Projects and Publications

11 | Opto-Electronic Perpendicular

Structural health monitoring of large-scale underground constructions is a safety and maintenance issue [81, 172]. This project features a feasibility study of a long-range opto-electronic perpendicular (OEP) utilizing electro-active glass targets to be installed into shafts, tunnels and/or roadways as an alternative to common inclinometer based approaches. An exactly aligned laser beam serves as reference axis to enable the measurement of relative movements caused by geological activities.

This project covers the monitoring aspect of CPS. The systems components are shown in the block diagram in Fig. 11.1. The measurement stations are spatially distributed within the physical domain. The cameras are sensing the dynamic behavior of the physical system. The measurement principle is sketched in Fig. 11.2 and 11.3. The measurement process is carried out on the embedded mini PC within the cyber domain, the mapping process is performed using bivariate discrete unitary polynomials and hierarchical subdivision. The measurement results are sent to a server (supervisory system) for further processing. The system architecture can be described as a wired sensor chain, i.e., the simplest form of a sensor network with a-priori known topology.

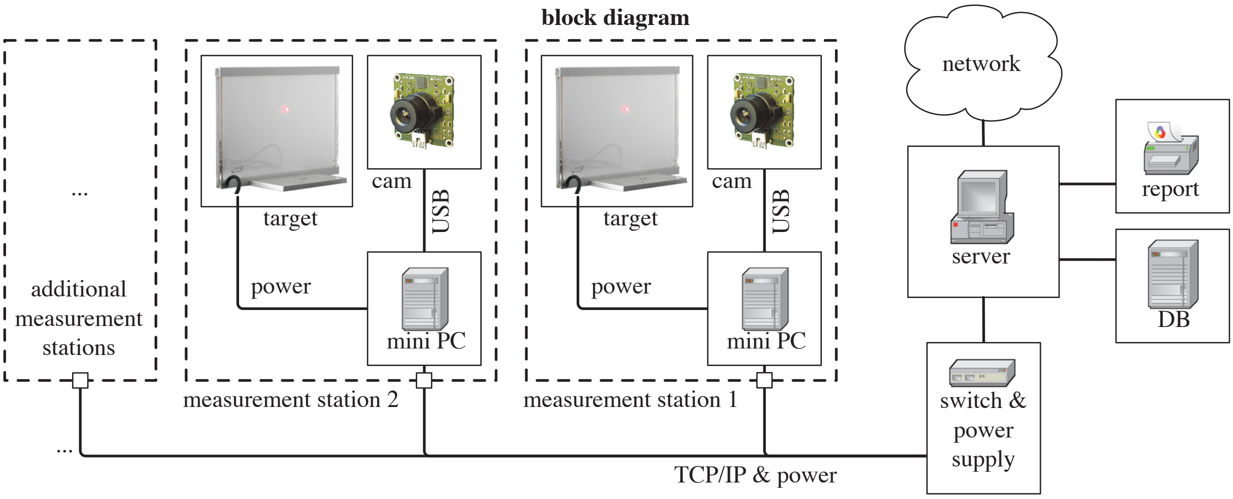


Fig. 11.1 The measurement system consists of n measurement stations (clients) in the field and a centralized supervisory system (server). Each measurement station contains an active-glass target, a USB camera and an embedded mini PC. The PC controls the target's transparency mode (actuator), acquires the images from the camera (sensor) and measures the location of the laser spot on the target. The processed results are forwarded to the server, which generates reports and stores the data in a data base (DB).

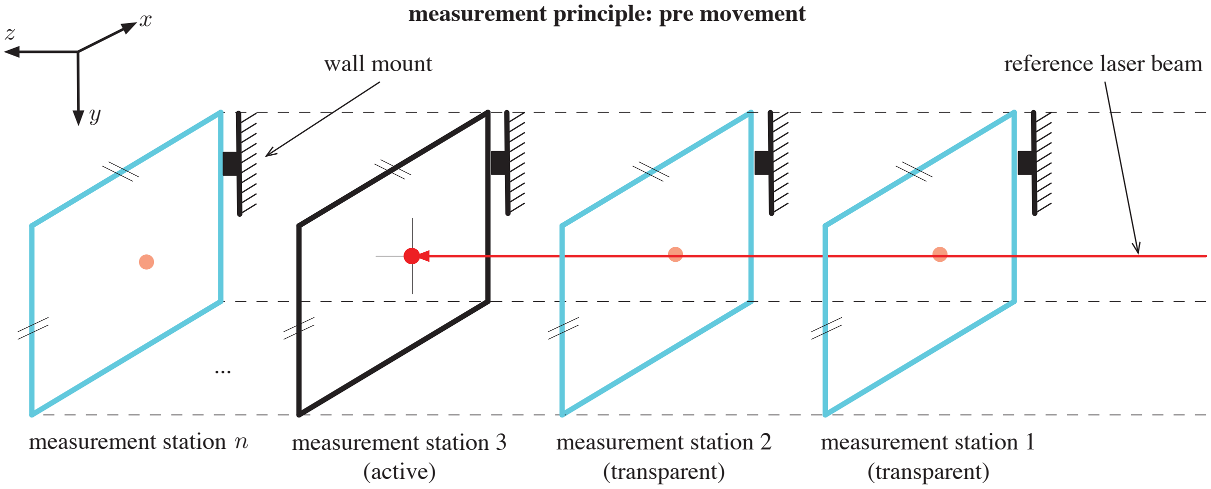


Fig. 11.2 A set of n consecutively aligned measurement stations is mounted onto the underground structure's wall. A reference laser beam impinges upon each target, whereby the active target is opaque and the other targets are transparent. A laser spot is projected onto the active target; its location is measured by a metric vision system. When initializing the measurement system, a reference measurement is carried out. The system covers a distance of several 100 [m]; however, the total achievable measurement range is determined by the environmental conditions. The required number n of measurement stations is application dependent.

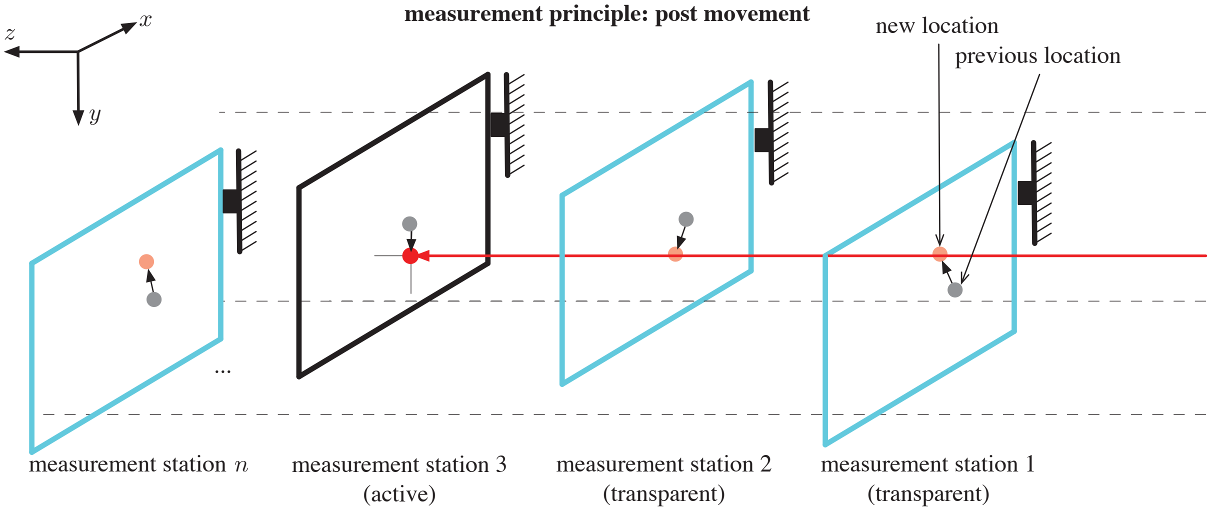


Fig. 11.3 Geological activities inflict movement of the underground structure's wall. The measurement stations move accordingly; the difference between the new and the previous location is an indicator of how much movement has appeared since the last reference measurement. If the measured value exceeds a given threshold, maintenance tasks are triggered to ensure the structural integrity of the monitored object.

11.1 IS&T/SPIE - Electronic Imaging 2013

February 3 - February 7, 2013 @ San Francisco, California, USA [71]

Citation[1]

```
@inproceedings{GuggHarkerOleary2013ei,
 author     = {Gugg, Christoph and Harker, Matthew and O'Leary, Paul},
 title      = {Structural deformation measurement via efficient tensor
               polynomial calibrated electro-active glass targets},
 booktitle  = {IS{\&}T/SPIE Electronic Imaging,
               Image Processing: Machine Vision Applications VI},
 volume     = {8661},
 pages      = {86610F},
 year       = {2013},
 doi        = {10.1117/12.2003679}}
```

Keywords

structural health monitoring; perpendicular; electro-active glass; coordinate mapping; homography; polynomial control points; tensor interpolation; hierarchical subdivision; quad tree decomposition;

[1] SPIE Digital Library, spiedigitallibrary.org

Structural deformation measurement via efficient tensor polynomial calibrated electro-active glass targets

Christoph Gugg, Matthew Harker and Paul O'Leary

University of Leoben, Chair of Automation, Peter Tunner-Strasse 27, 8700 Leoben, Austria

ABSTRACT

This paper describes the physical setup and mathematical modelling of a device for the measurement of structural deformations over large scales, e.g., a mining shaft. Image processing techniques are used to determine the deformation by measuring the position of a target relative to a reference laser beam. A particular novelty is the incorporation of electro-active glass; the polymer dispersion liquid crystal shutters enable the simultaneous calibration of any number of consecutive measurement units without manual intervention, i.e., the process is fully automatic. It is necessary to compensate for optical distortion if high accuracy is to be achieved in a compact hardware design where lenses with short focal lengths are used. Wide-angle lenses exhibit significant distortion, which are typically characterized using Zernike polynomials. Radial distortion models assume that the lens is rotationally symmetric; such models are insufficient in the application at hand. This paper presents a new coordinate mapping procedure based on a tensor product of discrete orthogonal polynomials. Both lens distortion and the projection are compensated by a single linear transformation. Once calibrated, to acquire the measurement data, it is necessary to localize a single laser spot in the image. For this purpose, complete interpolation and rectification of the image is not required; hence, we have developed a new hierarchical approach based on a quad-tree subdivision.

Cross-validation tests verify the validity, demonstrating that the proposed method accurately models both the optical distortion as well as the projection. The achievable accuracy is $e \leq \pm 0.01\,[\mathrm{mm}]$ in a field of view of $150\,[\mathrm{mm}] \times 150\,[\mathrm{mm}]$ at a distance of the laser source of $120\,[\mathrm{m}]$. Finally, a Kolmogorov Smirnov test shows that the error distribution in localizing a laser spot is Gaussian. Consequently, due to the linearity of the proposed method, this also applies for the algorithm's output. Therefore, first-order covariance propagation provides an accurate estimate of the measurement uncertainty, which is essential for any measurement device.

Keywords: structural deformation, electro-active glass, discrete orthogonal polynomials, tensor approximation, tensor interpolation, quad-tree search

1. INTRODUCTION

Structural deformations occur regularly in large scale technical installations. For instance, mines and tunnel systems are subject to deterioration through geological activities; cargo ships are bent by heavy swell and a bridge's structure is influenced by the weather conditions. Vertical structures, e.g., dams and shafts, are sometimes monitored using a *telependulum*[*], this device consists of a pendulum anchored at the top and having a large mass damped in an oil bath at the bottom. The steel wire of the pendulum forms the reference line and the position of anchors mounted in the wall are measured with respect to the wire. The device is, however, restricted to monitoring in the vertical direction. Additionally, the device is mechanical and therefore subject to damage, in particular the wire is subject to damage in the normal working conditions of a mine. *Theodolites*, e.g. from Leica [†], have a typical accuracy of 0.2 [mm] at a distance of 120 [m] in a field of view of 100 [mm] × 100 [mm] when used with a corner cube reflector.[1] The apparatus presented in this paper is an order of magnitude better. The main disadvantage of a theodolite is, however, that it requires a large angular space if it is to measure a

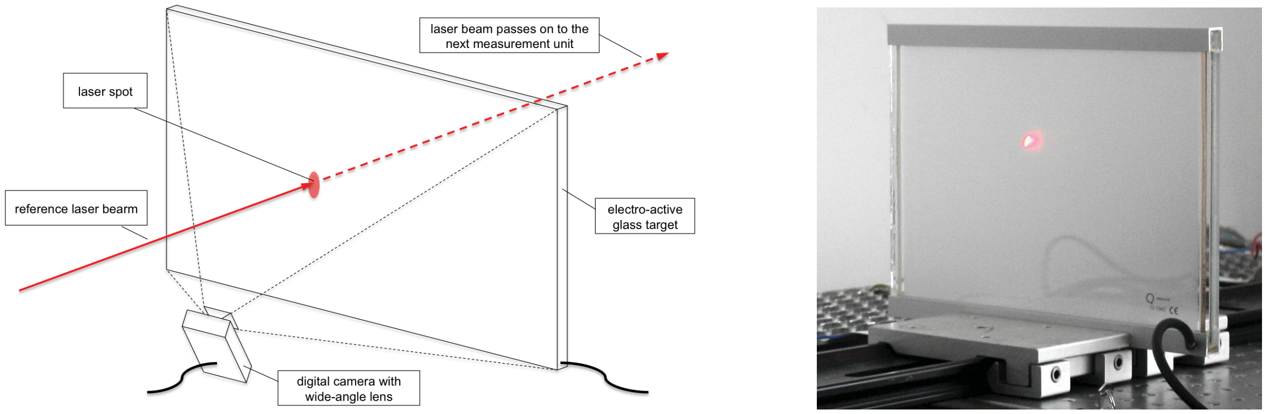
Further author information: http://automation.unileoben.ac.at

Christoph Gugg: E-mail: christoph.gugg@unileoben.ac.at, Telephone: +43 / (0)3842 / 402-5315

[*] ©Huggenberger AG, Horgen, Switzerland, www.huggenberger.com

[†] ©Leica Geosystems AG, Unterentfelden, Switzerland, www.leica-geosystems.com

series of reflectors correctly over a long range. This, for example, would mean that there may be no obstructions to the optical beam in a large portion of a shaft or tunnel.


laser beam passes on to the next measurement unit
laser spot
reference laser bearm
electro-active glass target
digital camera with wide-angle lens

(a) measurement unit's optical arrangement

(b) electro-active glass target

Figure 1: The optical arrangement (a) of the electro-active glass target (b) and the digital camera: the laser spot occurs when the beam impinges upon the target plate on its path. The camera detects this laser spot, ergo its relative location changes can be measured.

The optical arrangement of a single measurement unit is shown in Figure 1, the principle architecture of the device presented in this paper is shown in Figures 2 and 3. A reference laser beam is projected along the direction in which the monitoring is to be performed. Targets are mounted at desired intervals on the structure such that the *electro-active glass* portion of the target intersects the laser beam. In transparent mode, the *polymer dispersion liquid crystal* (PDLC) glass transmits the laser beam so that the laser can be seen by other targets along the structure. In the opaque mode the glass scatters the laser light and the position of where the laser beam penetrates the plane of glass can be measured. Clearly, if the target moves, the point of penetration of the laser will also move, this is the basis measurement mechanism. Each measurement unit consists of a PDLC glass plate and a camera, see Figure 1. The camera observes the surface of the glass plate to determine the point of penetration of the laser.

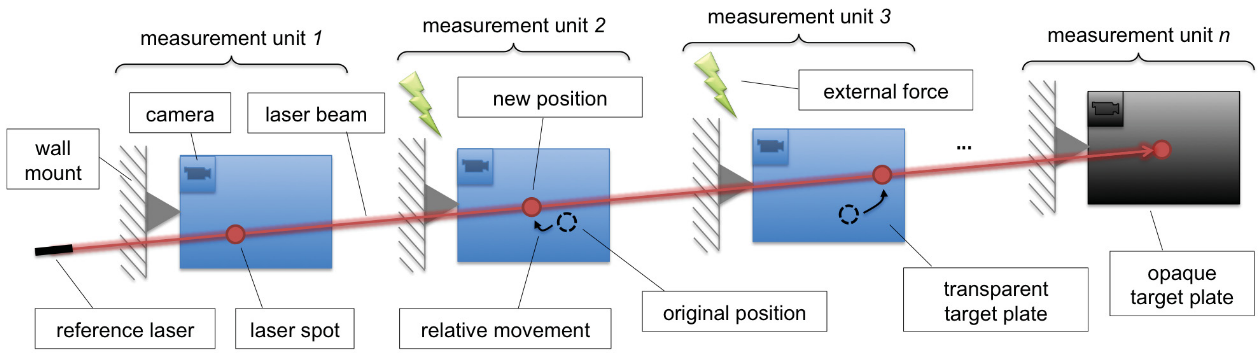
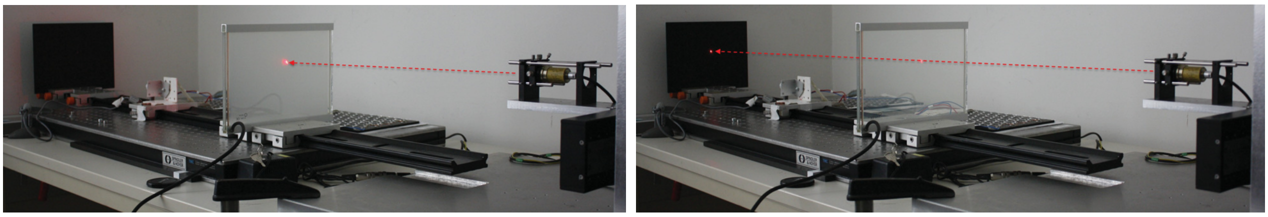


Figure 2: The instrument's alignment is composed of several sequentially arranged wall-mounted measurement units. The reference laser radiates a laser beam along a structure to enable the measurement of its movements.

The accuracy of the measurement device depends on the accuracy of the mapping from the camera pixel coordinates to the plane of the plate of glass. Both the projection, classically described by a *homography*,[2] and all the distortions associated with the optical arrangement have an influence on the obtained accuracy.

Consequently a large portion of this work is concerned with the modelling of lens distortions and the correction of projections. Additionally to enable a compact design, it is desirable to use lenses with a short optical length, which unfortunately exhibit large optical distortions, e.g., the fish eye effect. A *homography* alone is insufficient to obtain a mapping of satisfactory accuracy. Extensions of homographies to compensate for radial distortions would correct a portion of the optical distortion, however, they can not deal with non-radial distortions which are relevant at the accuracy which is to be obtained by the new device.



(a) glass in opaque mode

(b) glass in transparent mode

Figure 3: Electro-active glass allows the installation of a series of measurement units in a consecutive order. Measurements are carried out more accurately on opaque glass, whereas the transparent glass lets the laser beam pass.

Optical distortion of lenses is classically modelled using *Zernike moments*.[3] Zernike moments are a tensor product of a cyclic basis function set, e.g., the *Fourier* basis, and a radial polynomial which is subject to constraints. The constraints on the polynomial ensure that the model is C_n continuous at the origin. The Zernike moments are effectively moments around the optical axis which is perpendicular to the lens. Consequently, they can not be used to compensate for both the projection and optical distortion. The projection maps the optical distortion to an axis which does not correspond to the optical axis. *Polynomial control points* as proposed in the literature,[4, 5] are numerically unstable at higher degrees and/or at high accuracies, this fact is demonstrated in the testing section. In this paper we propose a new approach to a polynomial tensor mapping. *Gram polynomials*[6] are used since they exhibit high numerical accuracy even at high degrees and large numbers of nodes. Tensor polynomial approximation is used during calibration and an inverse tensor interpolation is used to determine the measurement result.[7–9]

The main contributions of this paper are:

1. A new optical arrangement for alignment measurement at multiple locations. The optical target uses an electro-active PDLC glass which can be switched between transparent and opaque. This arrangement enables the usage of multiple targets with respect to a single reference laser beam.

2. A calibration procedure which uses *tensor polynomial approximation* at selected nodes, whereby it implements the polynomial interpolation for interstitial points. This would enable a tensor polynomial interpolation from the real world coordinate to the camera coordinates. In this manner calibration is performed at a limited number of points, while it simultaneously enables a high precision mapping of the complete frame.

3. A quad-tree implementation of a search in the real world coordinates for the point corresponding to the observed laser point in the camera. This procedure implements an inverse tensor polynomial interpolation. This is necessary since tensor polynomial interpolation is by nature not bijective. The combination of approximation, interpolation and quad-tree search in this manner implements an efficient and high precision mapping from the camera coordinates to the real world coordinates. The mapping is capable of correcting for the projection and any type of optical distortion, as it is not restricted to radial distortion.

4. Extensive testing verifies that both the analytic and statistical properties of measurement device are as predicted. The comparative tests show that the new method is superior for this type of application.

2. CALIBRATION AND MEASUREMENT ARRANGEMENT

The mapping of a point's coordinates from camera space to real space and vice versa is a commonly encountered problem in image processing. The goal is the transformation of a camera point $\boldsymbol{p} = [x_c, y_c]^T$ in [pixel] to its corresponding real point $\boldsymbol{q} = [x_r, y_r]^T$ in [mm]. The required coordinate mapping function $\boldsymbol{q} = \boldsymbol{f}(\boldsymbol{p})$ is derived through a mathematical model, which is capable of describing the projection and, at the application at hand, the non-linear optical distortion.

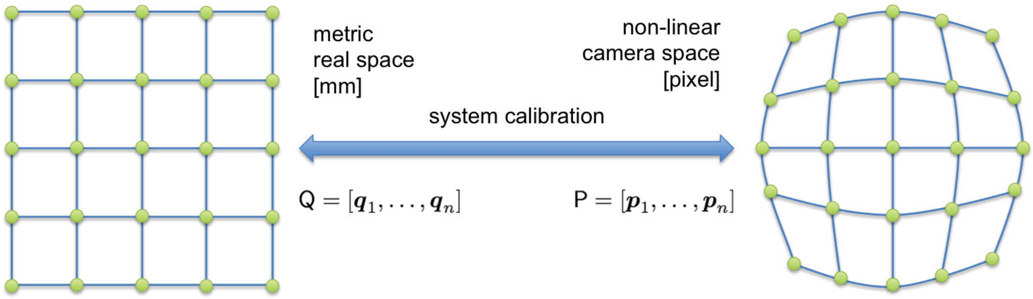

metric
real space
[mm]
non-linear
camera space
[pixel]
system calibration
$\mathsf{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n]$
$\mathsf{P} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n]$

Figure 4: The image's calibration points P in [pixel] are selected and matched to their corresponding real world calibration points Q in [mm] in order to derive the calibration data and approximate the mapping function.

The calibration arrangement consists of a theodolite mounted on a precision xy table, which is composed of two orthogonal linear drives in a vertical installation. The theodolite is used to generate the reference laser beam. It has been chosen because of its ability to easily correct the direction of the laser beam. The xy table moves the theodolite to multiple real points on a regular orthogonal grid $\mathsf{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n]$. Capturing the images of the laser spot's positions with the camera yields the corresponding calibration pattern, i.e. the matrix of camera points $\mathsf{P} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n]$. Obviously, the matrices Q and P do have the same dimensions. The center coordinates of the laser spots $\boldsymbol{p}_i$ are determined through conic fitting by approximating an ellipse based on contours. The generated grid must be sufficiently large so that the distortion of the optics and the projection associated with he optical arrangement can be characterized with the desired accuracy. All targets in the measurement system see the same grid of points during calibration, consequently, all targets are referenced to one and the same coordinate frame, effectively the coordinate frame of the xy table. In the calibration process, tensor polynomial approximation is used to determine the mapping from real world coordinates to camera coordinates.

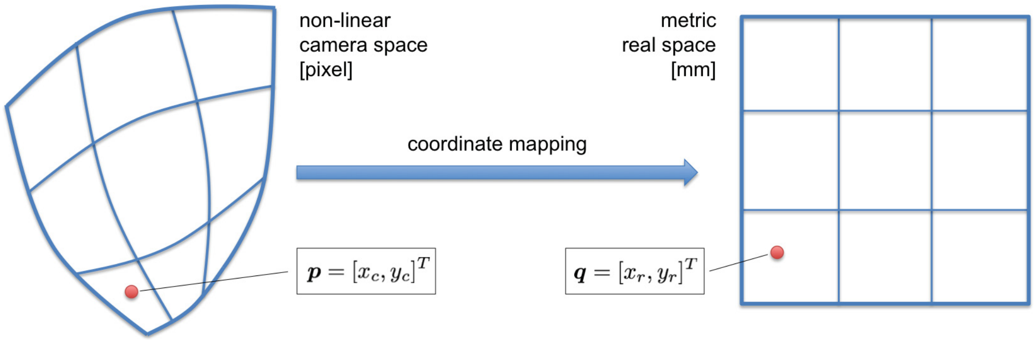


Figure 5: A non-linear camera space leads to a more complex mapping process during measurement, since the deployed model is also based on non-linear projections. By utilizing the calibration data, the coordinate mapping process delivers the real space coordinates of one laser spot $\boldsymbol{q}$ by assigning a metric value to the camera space point $\boldsymbol{p}$.

During measurement, a fixed laser projector generates the required beam of light. The targets are sequentially switched to opaque and during this period the position at the opaque target is determined. The location of the laser spot in the camera coordinate frame is determined first in pixel coordinates. The corresponding patch in the real world coordinate frame in which the laser spot lies is determined using the duality of the calibration points in both frames. This issue needs some thoughtful analysis for systems which have high accuracy demands. The presented system achieves an accuracy of $e \leq \pm 0.01\,[\mathrm{mm}]$ over a field of view of $150\,[\mathrm{mm}] \times 150\,[\mathrm{mm}]$ at distances of $120\,[\mathrm{m}]$. A quad-tree search is then performed to locate the point $\boldsymbol{q}$ in the real world which corresponds most closely to the point $\boldsymbol{p}$ in the pixel frame. Three main factors determine the reachable calibration quality: the real world's calibration points Q were produced with limited preciseness, the camera space's calibration points P can only be acquired with a certain accuracy and the used mathematical models are approximation methods.

3. CALIBRATION VIA TENSOR APPROXIMATION

The concept is to apply discrete unitary Gram polynomial basis functions to model the projection, i.e., a bivariate tensor product approximation is computed for image regularization. Such bases are suitable when the data, as in this case, is predominantly geometric in nature. However, the derivation provided in this section is completely generic and independent of the type of basis function used. Furthermore, due to the orthogonality, Gaussian noise is evenly spread over all spectral components. The polynomial decimation and tensor regularization serve to reduce the noise power significantly, resulting in registration that is not influenced by such disturbances. For the application at hand, we have used a rectangular calibration pattern; hence, the distorted images of the calibration points can be modelled by a set of double-indexed coordinates (x_{ij}, y_{ij}), which can be written concisely in matrix form as,

$$\mathsf{X}_r = [x_{ij}] \qquad \text{and} \qquad \mathsf{Y}_r = [y_{ij}]\,. \tag{1}$$

The grid points can therefore be modelled by bivariate tensor polynomials in the following manner,

$$\mathsf{X}_r = \mathsf{B}_v \mathsf{S}_x \mathsf{B}_u^T \qquad \text{and} \qquad \mathsf{Y}_r = \mathsf{B}_v \mathsf{S}_y \mathsf{B}_u^T, \tag{2}$$

where B_u and B_v represent discrete polynomials in the parameter space (u, v) and S_x and S_y represent the respective spectra of the transformations (i.e., the coefficient matrices). In this vein, the coordinates (x_{ij}, y_{ij}) are nodes in a system of curvilinear coordinates defined by bivariate polynomials. The nature of these curvilinear coordinates can be elucidated by as description of their coordinate curves. For a fixed u, say the k^{th} value of u, the (discrete) coordinate curve takes the form defined by the vectors,

$$\boldsymbol{x}_k = \mathsf{B}_v \mathsf{S}_x \mathsf{B}_u^T \boldsymbol{e}_k \qquad \text{and} \qquad \boldsymbol{y}_k = \mathsf{B}_v \mathsf{S}_y \mathsf{B}_u^T \boldsymbol{e}_k, \tag{3}$$

where the $\boldsymbol{e}_k$ are coordinate vectors. Thus if we define $\boldsymbol{u}_k$ as the k^{th} row of B_u then the coordinate curve takes the form,

$$\boldsymbol{x}_k = \mathsf{B}_v \mathsf{S}_x \boldsymbol{u}_k \qquad \text{and} \qquad \boldsymbol{y}_k = \mathsf{B}_v \mathsf{S}_y \boldsymbol{u}_k, \tag{4}$$

which is therefore polynomial in nature. Similarly, the l^{th} v-coordinate curve is given as,

$$\boldsymbol{x}_l^T = \boldsymbol{v}_l^T \mathsf{S}_x \mathsf{B}_u^T \qquad \text{and} \qquad \boldsymbol{y}_l^T = \boldsymbol{v}_l^T \mathsf{S}_y \mathsf{B}_u^T, \tag{5}$$

where $\boldsymbol{v}_l$ is the l^{th} row of B_v. For each target, we have a set of measured calibration points, represented in matrix form as $\hat{\mathsf{X}}_r$ and $\hat{\mathsf{Y}}_r$. In order to approximate the grid of points, we note that the sum of squared Euclidean distances from the measured points to the modelled points is given as,

$$\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij}^2 = \left\| \hat{\mathsf{X}}_r - \mathsf{X}_r \right\|_{\mathrm{F}}^2 + \left\| \hat{\mathsf{Y}}_r - \mathsf{Y}_r \right\|_{\mathrm{F}}^2. \tag{6}$$

Thus we define the cost function for the approximation as the least squares functional,

$$K\left(\mathsf{S}_x, \mathsf{S}_y\right) = \left\| \hat{\mathsf{X}}_r - \mathsf{B}_v \mathsf{S}_x \mathsf{B}_u^T \right\|_{\mathrm{F}}^2 + \left\| \hat{\mathsf{Y}}_r - \mathsf{B}_v \mathsf{S}_y \mathsf{B}_u^T \right\|_{\mathrm{F}}^2. \tag{7}$$

The partial derivatives with respect to the unknown spectra are given as,

$$\frac{\partial K}{\partial \mathsf{S}_x} = 2\left(\mathsf{B}_v^T\hat{\mathsf{X}}_r\mathsf{B}_u - \mathsf{B}_v^T\mathsf{B}_v\mathsf{S}_x\mathsf{B}_u^T\mathsf{B}_u\right), \tag{8}$$

$$\frac{\partial K}{\partial \mathsf{S}_y} = 2\left(\mathsf{B}_v^T\hat{\mathsf{Y}}_r\mathsf{B}_u - \mathsf{B}_v^T\mathsf{B}_v\mathsf{S}_y\mathsf{B}_u^T\mathsf{B}_u\right). \tag{9}$$

Equating the partial derivatives to zero and solving yields the expressions for the approximating spectra,

$$\mathsf{S}_x = (\mathsf{B}_v^T\mathsf{B}_v)^{-1}\mathsf{B}_v^T\hat{\mathsf{X}}_r\mathsf{B}_u(\mathsf{B}_u^T\mathsf{B}_u)^{-1} \quad \text{and} \quad \mathsf{S}_y = (\mathsf{B}_v^T\mathsf{B}_v)^{-1}\mathsf{B}_v^T\hat{\mathsf{Y}}_r\mathsf{B}_u(\mathsf{B}_u^T\mathsf{B}_u)^{-1}. \tag{10}$$

However, due to the orthonormality of B_u and B_v, these expressions simplify to,

$$\mathsf{S}_x = \mathsf{B}_v^T\hat{\mathsf{X}}_r\mathsf{B}_u \qquad \text{and} \qquad \mathsf{S}_y = \mathsf{B}_v^T\hat{\mathsf{Y}}_r\mathsf{B}_u. \tag{11}$$

Finally, substituting these expressions back into the model equations of the distorted grid, we obtain the least squares approximation,

$$\mathsf{X}_r = \mathsf{B}_v\mathsf{B}_v^T\hat{\mathsf{X}}_r\mathsf{B}_u\mathsf{B}_u^T \qquad \text{and} \qquad \mathsf{Y}_r = \mathsf{B}_v\mathsf{B}_v^T\hat{\mathsf{Y}}_r\mathsf{B}_u\mathsf{B}_u^T. \tag{12}$$

Clearly, if B_u and B_v are complete sets of basis functions, then $\mathsf{B}_u\mathsf{B}_u^T = \mathsf{I}$ and $\mathsf{B}_v\mathsf{B}_v^T = \mathsf{I}$, and the approximated grid is exactly the measured grid. However, using a truncated set of functions, we obtain a least squares approximation of the grid. For example, using only polynomials up to degree two, the grid is then modelled by biquadratic functions. Furthermore, the matrices $\mathsf{B}_u\mathsf{B}_u^T$ and $\mathsf{B}_v\mathsf{B}_v^T$ represent orthogonal projections onto the basis functions.

Once calibration is thus addressed, there remains the task of identifying a laser spot in the calibrated image. One possible approach would be the complete interpolation of the curvilinear coordinate system established above. This would entail interpolation of the $m \times n$ grid of points onto a $s \times t$ grid of points whereby s and t would necessarily be much larger than m and n. By the aforementioned derivation, this approximation would take the form,

$$\mathsf{X}_c = \tilde{\mathsf{B}}_v\bar{\mathsf{B}}_v^+\hat{\mathsf{X}}_r(\bar{\mathsf{B}}_u^+)^{\mathrm{T}}\tilde{\mathsf{B}}_u^T \qquad \text{and} \qquad \mathsf{Y}_c = \tilde{\mathsf{B}}_v\bar{\mathsf{B}}_v^+\hat{\mathsf{Y}}_r(\bar{\mathsf{B}}_u^+)^{\mathrm{T}}\tilde{\mathsf{B}}_u^T. \tag{13}$$

where, $\bar{\mathsf{B}}_u$ and $\bar{\mathsf{B}}_v$ are respectively submatrices of the matrices $\tilde{\mathsf{B}}_u$ and $\tilde{\mathsf{B}}_v$ which are discrete polynomials for the entire $s \times t$ grid, and the exponent $+$ denotes the Moore-Penrose pseudo-inverse.

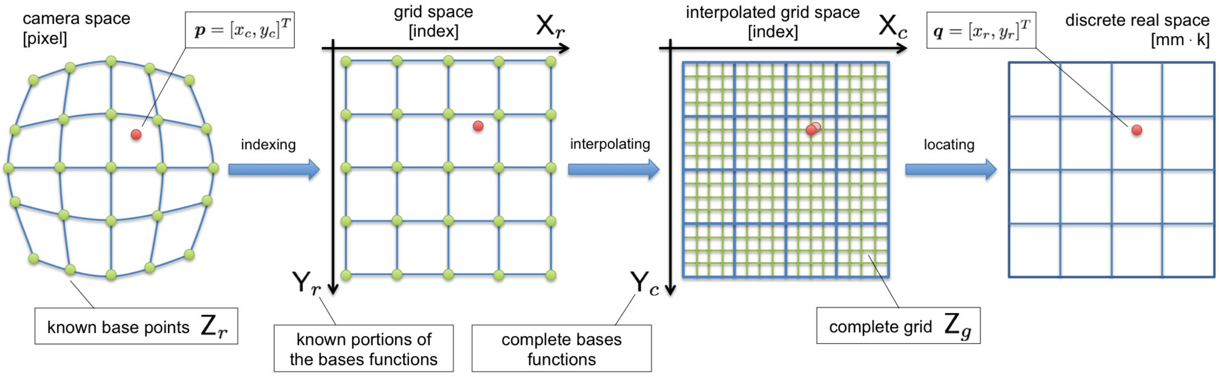


Figure 6: The goal is to map the point $\boldsymbol{p} = [x_c, y_c]^T$ from the camera space to its real space pendant $\boldsymbol{q} = [x_r, y_r]^T$. The known calibration points coordinates are stored in the tensor $\mathsf{Z}_r = \{\mathsf{X}_\mathsf{r}, \mathsf{Y}_\mathsf{r}\}$. The geometry is described by deploying a set of discrete orthogonal Gram polynomials in x and y direction. The reduced functions X_r and Y_r are the known portions of the complete bases X_c and Y_c. The performed interpolation yields the complete grid $\mathsf{Z}_g = \{\mathsf{X}_\mathsf{c}, \mathsf{Y}_\mathsf{c}\}$. The point $\boldsymbol{q}$ can now be located in the resulting discrete real space with the resolution k.

Since the task is to localize a single point in curvilinear coordinates, this is a particularly inefficient approach due to unacceptable computation and memory requirements. Further, since we can only localize the point up to the nearest interpolated grid point, its localization accuracy depends on the chosen resolution k. For instance, a field of view of $150\,[\mathrm{mm}] \times 150\,[\mathrm{mm}]$ would lead to a full grid of $150,000\,[\mu\mathrm{m}] \times 150,000\,[\mu\mathrm{m}]$ for a resolution of $k = 1000$, it's not reasonable to compute tensors of this size. Consequently, the classical tensor polynomial interpolation[7] can't be used in this application. The following section proposes a concept to handle the problem in a much more economic manner.

4. MEASUREMENT VIA QUAD-TREE INTERPOLATION

As was pointed out earlier, it is not practical to compute the complete grid which would be necessary to obtain the required accuracy. However, since it is only required to locate a single point, namely the laser point $\boldsymbol{p}$, it can be accomplished far more efficiently by introducing a hierarchical subdivision algorithm for interpolation. The approach has been employed in the past for non-rigid registration[10–12] and quad-tree decomposition[13–15] is used to partition an image into patches with similar levels of information content. As a matter of fact, this is the first application of a quad-tree search to improve the efficiency of tensor polynomial interpolation. In the following, we show that it is unnecessary to compose the full interpolation matrices X_c and Y_c. Firstly, it is noted that the spectra of the complete interpolation grid are computed according to Equation 13 as,

$$\bar{\mathsf{S}}_x = \bar{\mathsf{B}}_v^+ \hat{\mathsf{X}}_r (\bar{\mathsf{B}}_u^+)^{\mathrm{T}} \qquad \text{and} \qquad \bar{\mathsf{S}}_y = \bar{\mathsf{B}}_v^+ \hat{\mathsf{Y}}_r (\bar{\mathsf{B}}_u^+)^{\mathrm{T}}. \tag{14}$$

These matrices are, however, only of size equal or lesser than $m \times n$, i.e., the size of the calibration grid. As with extracting coordinate curves in the curvilinear coordinate system, we may also easily extract single points; namely, the point indexed (l, k) has the coordinates,

$$x_{lk} = \boldsymbol{e}_k^T \tilde{\mathsf{B}}_v \bar{\mathsf{S}}_x \tilde{\mathsf{B}}_u^T \boldsymbol{e}_l \qquad \text{and} \qquad y_{lk} = \boldsymbol{e}_k^T \tilde{\mathsf{B}}_v \bar{\mathsf{S}}_y \tilde{\mathsf{B}}_u^T \boldsymbol{e}_l. \tag{15}$$

Similarly by defining the k^{th} row of $\tilde{\mathsf{B}}_v$ as $\tilde{\boldsymbol{u}}_k$ and the l^{th} row of $\tilde{\mathsf{B}}_u$ as $\tilde{\boldsymbol{v}}_l$, this simplifies to,

$$x_{lk} = \tilde{\boldsymbol{v}}_l^T \bar{\mathsf{S}}_x \tilde{\boldsymbol{u}}_k \qquad \text{and} \qquad y_{lk} = \tilde{\boldsymbol{v}}_l^T \bar{\mathsf{S}}_y \tilde{\boldsymbol{u}}_k. \tag{16}$$

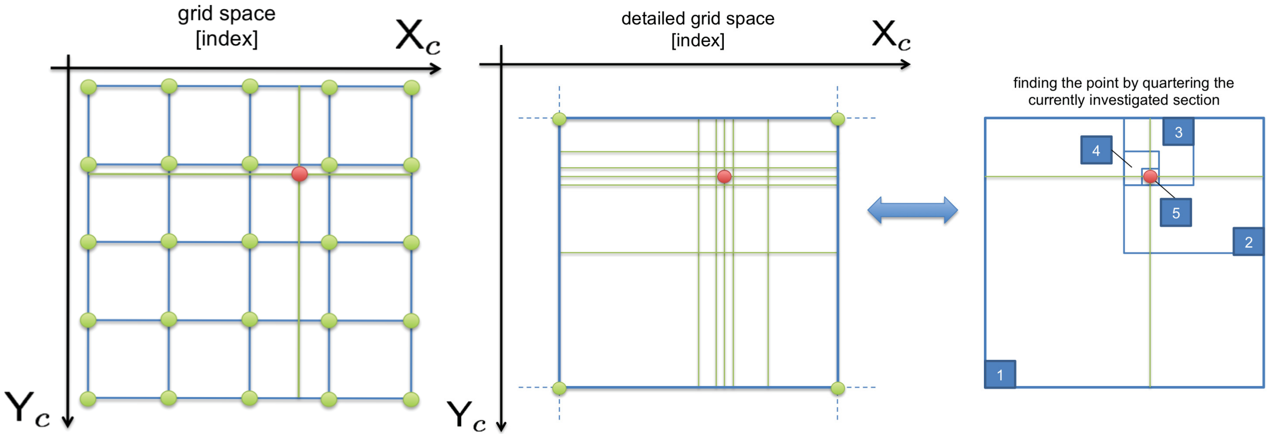
Since the vectors $\tilde{\boldsymbol{u}}_k$ and $\tilde{\boldsymbol{u}}_k$ are respectively m- and n-vectors, this computation is efficient; the computational overhead lies in the computation of $\bar{\mathsf{S}}_x$ and $\bar{\mathsf{S}}_y$. This method of coordinate extraction can subsequently be used to localize the laser point in the interpolated grid. The process starts with the a-priori knowledge of the location of the calibration points. The area enclosing the point is quartered and the quarter containing the sought point is selected. The four corners' locations are calculated utilizing the basis function vectors $\boldsymbol{u}_k$ and $\boldsymbol{v}_l$. Once again, the resulting rectangle is quartered and the procedure starts anew. The recursion stops when the desired accuracy is reached, namely at the iteration w when the distance,

$$\left(x_{kl}^{(w)} - x_c\right)^2 + \left(y_{kl}^{(w)} - y_c\right)^2, \tag{17}$$

is smaller than a given tolerance. Search trees are commonly employed to reduce the amount of data to be processed significantly. In this application, the *quad-tree search* (QTS) algorithm allows the efficient evaluation of the mapping function only at specific locations. This implementation is, however, different to common approaches, which use this idea to find certain elements in existing data structures. The resulting depth of the quaternary search tree τ is related to the number of possible locations inside the interpolated grid n, i.e.

$$\tau = \log_4 n. \tag{18}$$

An alternative approach is to locate the patch, bounded by four calibration points, in which the laser spot is observed. Then for this section of the coordinate frame, the bivariate tensor polynomial interpolation is performed, for brevity this method is called *basis function sectioning* (BFS). Experimental testing shows that this method yields comparable results to the QTS approach, however, it is significantly slower in computation.



(a) single basis function

(b) quad-tree search

Figure 7: (a) The point's position is described by the basis function vectors $\boldsymbol{x}_p$ and $\boldsymbol{y}_p$. (b) The section of Z_r containing the sought point is selected. The process incrementally approximates the point's location through quartering of the investigated area until the desired accuracy is reached.

5. EXPERIMENTAL VERIFICATION

In order to enable the evaluation of random error effects, the computation of the standard error is necessary.[16] This value describes the approximate 95% confidence interval for the mean of the sample values. Consequently, the underlying data must be normally distributed for the estimation. This condition is tested through the usage of the non-parametric *Kolmogorov Smirnov test* (KS-test), which checks the goodness of fit between one-dimensional empirical data and its expected reference continuous parent distribution. The test quantifies the distance between both cumulative distribution functions.[17] If the difference between the measured data and the expected distribution stays within a defined confidence interval, then the data is assumed to be normally distributed and the null-hypothesis H_0 is accepted. The tests showed, that this is always true for the camera coordinates $\boldsymbol{p}$. As a consequence, due to the proposed method's linear nature, this condition also holds for the mapped coordinates $\boldsymbol{q}$. This can be shown through the first order covariance propagation:

$$\Lambda_{\boldsymbol{q}} = \nabla \boldsymbol{f} \Lambda_{\boldsymbol{p}} \nabla \boldsymbol{f}^T. \tag{19}$$

This basically illustrates, that under the condition, that the mapping function $\boldsymbol{f}$ is a linear transformation, the output has the same type of distribution as the input, whereby Λ is the covariance.[18]

Table 1: Measurement results utilizing a lens with $f = 3.6$ [mm], a field of view of 150 [mm] × 150 [mm].

method:	degree	$\bar{e}_x/[\text{mm}]$	$\bar{e}_y/[\text{mm}]$	$\bar{e}_{xy}/[\text{mm}]$
homography	$d = 1$	3.4920	2.1580	4.5597
polyCtrlPts	$d = 2$	3.4600	2.1214	4.5473
	$d = 4$	0.3085	0.2423	0.4393
tensorBFS	$d = 2$	2.7969	1.4894	3.3621
	$d = 4$	0.1160	0.0374	0.1379
	$d = 6$	0.0063	0.0014	0.0074
tensorQTS	$d = 2$	2.7966	1.4888	3.3614
	$d = 4$	0.1154	0.0316	0.1316
	$d = 6$	0.0067	0.0020	0.0079

During the experiments, an interlaced cross validation matrix is produced to verify the quality of the interpolation through a set of independent test points. Only even degrees d are deployed, since the camera space has axis symmetry. A wide-angle lens with $f = 3.6\,[\mathrm{mm}]$ is used, thus a polynomial degree of up to $d = 6$ is chosen. The error $\bar{e}_x$ is the mean error in x dimension, the error $\bar{e}_y$ is the mean error in y dimension and the error $\bar{e}_{xy}$ is the mean geometric inaccuracy in x and y dimensions over all inspected points. This is documented in Table 1. Although the results delivered by the two tensor interpolation methods are of similar quality, the computational efficiency of the quad-tree search approach is far superior than the idea of basis function sectioning.

6. CONCLUSIONS

The proposed use of a polymer dispersion liquid crystal electro active glass has performed well. A positional measurement accuracy of $e \leq \pm 0.01\,[\mathrm{mm}]$ for a 95% confidence, verified using a Kolmogorov Smirnov test, was obtained. The architecture of the system permits the measurement to be performed for structures extending of several 100's of metres. The measurement accuracy has been achieved despite the use of lenses with a very short focal lengths and exhibiting significant optical distortions. This has been made possible by a new tensor polynomial approximation and interpolation method used to perform the mapping between the camera coordinate frame and the real world frame. The quad-tree search implements an efficient and accurate computation algorithm for the mapping. The possible use of short focal lengths has in turn enabled a very compact hardware design for the measurement units.

REFERENCES

[1] Bas, H. G., "The accuracy of using theodolite in close-range engineering measurements," *International Archives of Photogrammetry and Remote Sensing* **XXXIII** (2000).

[2] Hartley, R. and Zisserman, A., [*Multiple View Geometry in Computer Vision*], Cambridge University Press (2004).

[3] Malacara, D., [*Optical Shop Testing*], John Wiley and Sons, New York, Chichester, Brisbane, Toronto, Singapore (1992).

[4] Goshtasby, A., "Piecewise linear mapping functions for image registration," *Pattern Recognition* **19** (1986).

[5] Goshtasby, A., "Image registration by local approximation methods," *Image and Vision Computing* **6** (1988).

[6] Gram, J., "Ueber die entwicklung realer funktionen in reihen mittels der methode der kleinsten quadrate.," *Journal fuer die reine und angewandte Mathematik* (1883).

[7] O'Leary, P. and Harker, M., "An algebraic framework for discrete basis functions in computer vision," *IEEE Indian Conference on Computer Vision, Graphics and Image Processing* (2008).

[8] Badshah, A., O'Leary, P., and Harker, M., "Gram polynomial image decimation and its application to non-rigid registration," *Image Processing: Machine Vision Applications IV, SPIE* (2011).

[9] O'Leary, P. and Harker, M., "Discrete polynomial moments for real-time geometric surface inspection," *Journal of Electronic Imaging, SPIE* (2009).

[10] Zitov, B. and Flusser, J., "Image registration methods: a servey," *Image and Vision Computing* **21** (2003).

[11] Lester, H. and Arridge, S., "A survey of hierarchical non-linear medical image registration," *Image Vision Computation* **6** (1988).

[12] Badshah, A., O'Leary, P., Harker, M., and Sallinger, C., "Non-rigid registration for quality control of printed materials," *10th SPIE International Conference on Quality Control by Artificial Vision (QCAV)* (2011).

[13] Likar, B. and Pernu, F., "A hierarchical approach to elastic registration based on mutual information," *Image and Vision Computing* **19** (2001).

[14] Mellor, M. and Brady, M., "Non-rigid multimodal image registration using local phase," *MICCAI.LNCS* **3216** (2004).

[15] Andronache, A., Siebenthal, M., Szekely, G., and Cattin, P., "Non-rigid registration of multimodal images using both mutual information and cross correlation," *Medical Image Analysis* **12** (2008).

[16] Rönz, B. and Strohe, H., [*Lexikon Statistik*], Gabler Verlag (1994).

[17] Kreyszig, E., [*Statistische Methoden und ihr Anwendungen*], Vandenhoeck und Ruprecht, Göttingen (1979).

[18] Clarke, J. C., "Modelling uncertainty: A primer," tech. rep., Department of Engineering Science, Oxford University (1998).

12 | Embedded Geo-Sensing

Geo-mechanical effects are likely to cause structural deformations during underground constructions and mining operations [213]. Ground subsidence occurs in scenarios where subsurface voids (e.g., cavities) are collapsing. Incidents and accidents related to this topic are discussed in [112]. Geological models for manifestation and magnitude prediction of the subsidence are available in [15, 112]. In this section, the focus is laid on effects caused by mining and tunneling. Mine subsidence is the movement of the ground surface as a result of readjustments of the overburden. Sinkholes are caused by abrupt depression in the ground surface (*bump*), whereas troughs are progressive long-term effects (*squeeze*). Full extraction mining techniques which intentionally cause the extracted void to collapse such as *retreat room-and-pillar mining* [26] and *longwall mining* will most likely result in surface subsidence. Mining and tunneling-induced subsidence is localized to the surface above the mined area, which makes this area eligible for structural health monitoring (SHM). The goal is to monitor the deformation caused by the excavation and its impact on surrounding/overlying facilities such as building, bridges, rails and roads.

Instabilities are characterized by the assessment of pattern, depth and rate of movement/change. The use of inclinometers in tunneling projects as an alternative approach to conventional geodetical measurements for displacement monitoring was shown in [206]. On overview for applying inclinometers in structural and geomechanical monitoring systems is given in [129]. An inclinometer measures angles of slope/tilt and belongs to the class of inertial measurement units (IMU). IMUs are commonly used for movement detection including navigation, vibration analysis and subsurface deformation. Modern IMUs are MEMS-based sensors, which are highly sensitive and have a long term stability [62]. Stationary systems of in-place inclinometers employ a fixed chain of sensors; it is ideal for permanent and continuous SHM [42].

The sensor chain is situated in a protective casing. The inclinometer casing is installed either in a horizontal borehole/trench (Fig. 12.1 (a)) or a vertical borehole (Fig. 12.1 (b)) that passes through a suspected zone of movement [129]. It may also be attached directly to or embedded in a structure. An initial reference measurement is required to compensate for the inevitable inexactness such as twists and bends during installation. As a matter of fact, a horizontal alignment should be slightly inclined towards the open end to allow water drainage from the hole. The sensors measure the inclination of the casing. Movement of the ground deflect the casing, causing the sensors to undergo changes in inclination [42, 144]. Combination of all tilts yields the profile of lateral deflection, i.e., a gradient curve. This provides implicit information about the strain, i.e., tension or compression, of the material. Relative vertical and horizontal movements are determinable if a-priori knowledge about constraints are given.

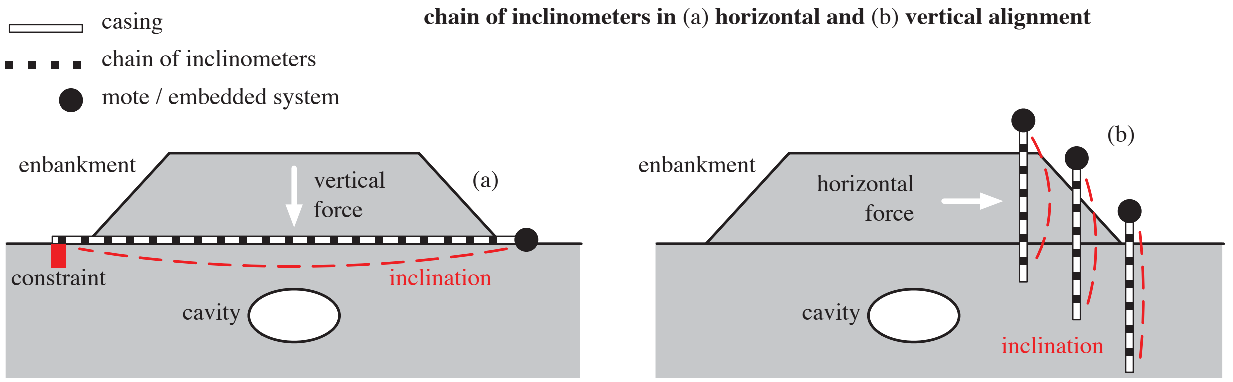


Fig. 12.1 A horizontal alignment of the sensor chain enables the detection of compression caused by a vertical force whereas a vertical alignment of the sensor chain enables the detection of displacement caused by a horizontal force. A combination of both alignments is required to monitor all occurring effects. Constraints ensure a unique solution for the curve reconstruction.

Networks of spatially distributed sensor nodes enable condition monitoring of large-scale structures in order to ensure safety for men and assets. Chains of inclinometers are connected to dataloggers in a conventional measurement system. Dataloggers acquire, consolidate and prepare the data for transmission, but they do not apply any abstract models [78]; the *raw* tilt data is forwarded to a supervisory system, either as a data stream or in a sequence of data packages consolidating multiple measurements. Near real-time processing is usually a system requirement in safety relevant SHM applications [81]; furthermore, the physical stability of network connections cannot be guaranteed in harsh subsurface environments. In contrast to a datalogger, an independent CPS mote must perform on-site computation in order to implement a reactive system which can operate without supervision. In this project, an individual mote is composed of an embedded system with a chain of inclinometers connected to it, see Fig. 12.1. The goal is to detect and localize damage in complex structures as well as to provide both long-term monitoring and rapid analysis in response to severe events. The application requires the real-time solution of an inverse problem on an embedded system; this increases the information density, lowers the bandwidth demands and reduces the post-processing effort.

Inclinometers measure the tilt, which is the first local derivative of the subsidence profile in a mathematical sense, i.e., the form of the curve directly corresponds to the subsidence of the ground. Reconstruction of this curve, i.e., the original profile, involves the solution of an ODE with perturbed values. This is, by definition, an inverse problem [202]. Contextual knowledge about the problem is required; in this vein, the implementation of constraints is application specific, e.g., a building's concrete base or a bridge's pillars. These fixed points in the structure correspond to boundary values. The curvature is the second derivative of the surface profile and is an indicator for the tilt's rate of change. In practical applications, the inclinometer measurements are often only interpreted qualitatively by a human operator without a suitable model, i.e., only simple threshold checks are performed. Without a model, it is virtually impossible to analyze the data from a spatially distributed CPS.

The theoretical framework for this particular project has been established in [162, 165]. The process is divided in an initial preparatory computation and a cyclic run-time computation, ensuring high efficiency for repeated measurements. The method is suitable for perturbed and unperturbed as well as overconstrained ODE. The numerical solver is implemented with model based design and uses only fundamental linear algebra; consequently, this approach supports automatic code generation for embedded targets. A complete in-the-loop verification process is presented in the following publications. The method is compared with state-of-the-art techniques via model-in-the-loop and the functional equivalence of the produced `C` code is tested with software-in-the-loop. A laboratory prototype verifies the model as well as it's deployment on an embedded system via processor-in-the-loop testing and real measurement data. The validity of the model was also verified independently in [177] using light sectioning and laser time-of-flight distance measurements on a laboratory setup.

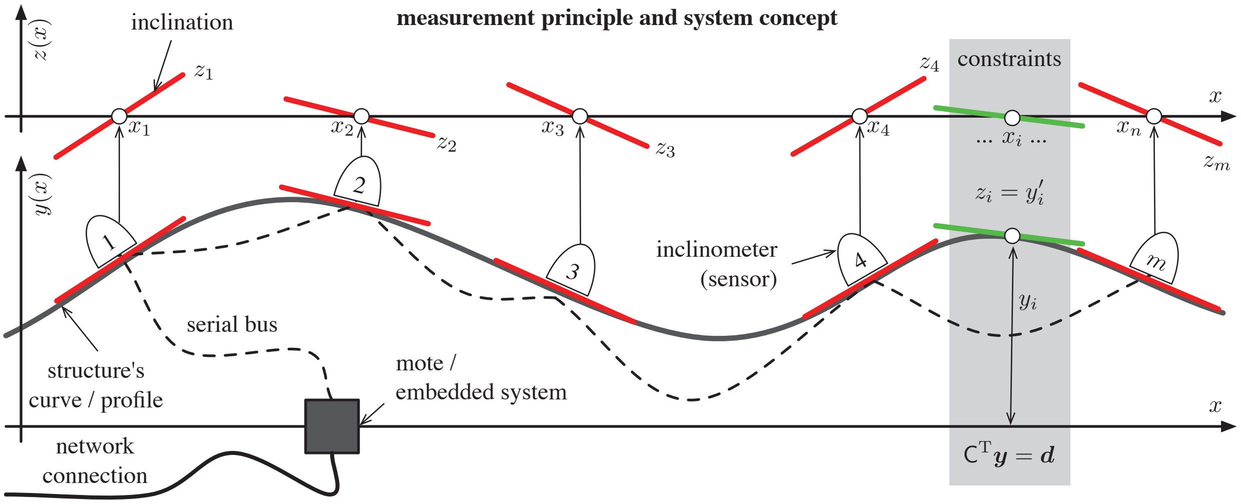


Fig. 12.2 The measurement principle involves a chain of m inclinometers which deliver the measured inclinations $\boldsymbol{z} = [z_1 \ldots z_m]$ at the locations of $\boldsymbol{x}$. The goal is the reconstruction of the curve $\boldsymbol{y} = y(\boldsymbol{x})$ from the perturbed $\boldsymbol{z}$ values, i.e., the first derivation. The constraints are formulated as $\mathsf{C}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{d}$ and ensure a unique solution. The sensors are connected to the embedded system via a serial bus, e.g., RS485, effectively forming a CPS mote. See Fig. 10.2 for the corresponding methodology.

12.1 ACM/IEEE - International Conference on Cyber-Physical Systems (ICCPS) 2014

April 15 - April 17, 2014 @ Berlin, Germany [88]

Citation[1]

```
@inproceedings{HarkerGuggOleary2014iccps,
 author     = {Harker, Matthew and Gugg, Christoph and O'Leary, Paul},
 title      = {Model Based Design for the Real-Time Solution
               of Inverse Problems in Cyber-Physical Systems},
 booktitle  = {IEEE/ACM Fifth International Conference on
               Cyber-Physical Systems (ICCPS)},
 year       = {2014},
 pages      = {217},
 doi        = {10.1109/ICCPS.2014.6843730}}
```

Keywords

ordinary differential equations; inverse problem; perturbed measurements; real-time computation; embedded system; model based design; automatic code generation; in-the-loop verification;

[1] IEEE Xplore Digital Library, ieeexplore.ieee.org

WiP Abstract: Model Based Design for the Real-Time Solution of Inverse Problems in Cyber-Physical Systems

Matthew Harker, Christoph Gugg and Paul O'Leary
Chair of Automation, Department Product Engineering, University of Leoben, Austria
automation@unileoben.ac.at

1. NUMERICAL ODE SOLVER

Measurement data is continuously processed by cyber-physical systems in condition monitoring applications, e.g. sensor networks with inclinometers. The presented numerical ordinary differential equation (ODE) solver runs on independent and decentralized embedded systems thanks to its real-time capability and computational efficiency. ODEs can be formulated as a least squares problem with linear constraints [1],

$$\mathsf{L}\,\boldsymbol{y} = \boldsymbol{g} \quad \text{given} \quad \mathsf{C}^{\mathrm{T}}\,\boldsymbol{y} = \boldsymbol{d}, \tag{1}$$

where $\boldsymbol{y}$ is the solution vector, $\boldsymbol{g}$ is the discrete perturbed forcing function vector, the matrix C and the vector $\boldsymbol{d}$ are the discretized constraints. The discrete linear differential operator L is defined as

$$\mathsf{L} \triangleq \mathsf{A}_n\,\mathsf{D}_n + \mathsf{A}_{n-1}\,\mathsf{D}_{n-1} + \ldots + \mathsf{A}_1\,\mathsf{D} + \mathsf{A}_0, \tag{2}$$

where $\mathsf{A}_n = \mathrm{diag}(a_n(\boldsymbol{x}))$ with $a_n(\boldsymbol{x})$ are the coefficient functions at the node locations $\boldsymbol{x}$ and the matrix D_n is a local approximation with support length l_s to a continuous differential operator of n^{th} degree; at least n initial-, inner- or boundary-values are required for the n nodes to ensure that there is a unique solution. A solution for $\boldsymbol{y}$ with $\boldsymbol{\beta}$ parameters is computed by

$$\boldsymbol{y} = \{\mathsf{C}^{\mathrm{T}}\}^{-}\,\boldsymbol{d} + \mathrm{null}\left\{\mathsf{C}^{\mathrm{T}}\right\}\,\boldsymbol{\beta}, \tag{3}$$

where $\{\mathsf{C}^{\mathrm{T}}\}^{-}$ is a generalized inverse of C^{T}. In general, $\boldsymbol{y}_c \triangleq \{\mathsf{C}^{\mathrm{T}}\}^{-}\,\boldsymbol{d}$ is a function which fulfills the constraints. The matrix $\mathsf{N}_c \triangleq \mathrm{null}\left\{\mathsf{C}^{\mathrm{T}}\right\}$ contains an orthonormal vector basis set for the null-space of C^{T}, i.e. $\mathsf{C}^{\mathrm{T}}\,\mathsf{N}_c\,\boldsymbol{\beta} = \boldsymbol{0}$. Furthermore, $\boldsymbol{y}_c$ and N_c form a complete orthogonal basis function set, hence $\boldsymbol{y} = \boldsymbol{y}_c + \mathsf{N}_c\,\boldsymbol{\beta}$. Substituting the term for $\boldsymbol{y}$ in $\mathsf{L}\,\boldsymbol{y} = \boldsymbol{g}$ yields,

$$\mathsf{L}\,\{\boldsymbol{y}_c + \mathsf{N}_c\,\boldsymbol{\beta}\} = \boldsymbol{g}. \tag{4}$$

The above equation is solved using a least squares approach to obtain the unique global minimum of

$$\min_{\boldsymbol{\beta}} \|\mathsf{L}\,\mathsf{N}_c\,\boldsymbol{\beta} + \mathsf{L}\,\boldsymbol{y}_c - \boldsymbol{g}\|_2^2. \tag{5}$$

The errors are globally evenly distributed. There is no implicit direction of integration, which may lead to an accumulation of errors as with typical initial value problem approaches such as Runge-Kutta. By defining $\mathsf{M} \triangleq \mathsf{N}_c\,\{\mathsf{L}\,\mathsf{N}_c\}^{+}$, where $\{\cdot\}^{+}$ denotes the pseudo inverse, the homogeneous solution $\boldsymbol{y}_h$ is computed as,

$$\boldsymbol{y}_h = \boldsymbol{y}_c - \mathsf{M}\,\mathsf{L}\,\boldsymbol{y}_c = (\mathsf{I} - \mathsf{M}\,\mathsf{L})\,\boldsymbol{y}_c. \tag{6}$$

The final solution for $\boldsymbol{y}$ is then,

$$\boxed{\boldsymbol{y} = \mathsf{M}\,\boldsymbol{g} + \boldsymbol{y}_h.} \tag{7}$$

Due to the model's linearity, the solution's covariance $\Lambda_{\boldsymbol{y}}$ is computed as $\Lambda_{\boldsymbol{y}} = \mathsf{M}\,\Lambda_{\boldsymbol{g}}\,\mathsf{M}^{\mathrm{T}}$, where $\Lambda_{\boldsymbol{g}}$ is the forcing function's covariance. Both, M and $\boldsymbol{y}_h$, can be computed a-priori during calibration, i.e. they need not to be computed during the measurement's run-time. Only the sensor data $\boldsymbol{g}$ changes with each measurement. The solution of the inverse ODE is reduced to a single matrix multiplication and a vector addition. The computational complexity is $\mathcal{O}(n^2)$; it is independent of the equation being solved and the sensor nodes' placement. Model based design and automatic code generation supports the straight-forward integration of the algorithm on an embedded system. Functionally equivalent C code is generated via MATLAB Coder Toolbox from the model and verified via software-in-the-loop (SIL) and processor-in-the-loop (PIL) testing for the Atmel 8-bit AVR RISC-based ATmega328 microcontroller.

2. REFERENCES

[1] P. O'Leary and M. Harker. A framework for the evaluation of inclinometer data in the measurement of structures. *IEEE T. Instrumentation and Measurement*, 61(5):1237–1251, 2012.

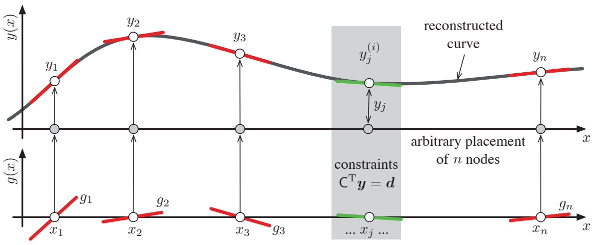
Model Based Design for the Real-Time Solution of Inverse Problems in Cyber-Physical Systems

Matthew Harker, Christoph Gugg and Paul O'Leary

Scope of this Work

Measurement data is continuously processed by cyber-physical systems in condition monitoring applications, e.g., sensor networks with inclinometers. We present an efficient numerical ordinary differential equation (ODE) solver that executes on embedded systems in real-time.

Structural Condition Monitoring via Inclinometers



Structural condition monitoring is required during subsurface constructions. Geo-mechanical effects such as ground subsidence are monitored using inclinometer sensors. Reconstruction of the ground displacement from the orientation of the inclinometers is an inverse problem. Additionally, there are points where constraints are placed on the construction, for example pillars, which in turn define initial-, inner- or boundary values.

The goal is to formulate an algebraic framework, which can be translated to functionally equivalent C code.

Numerical ODE Solver for Real-Time Applications

ODEs can be modeled as a least squares problem with linear constraints in the form,

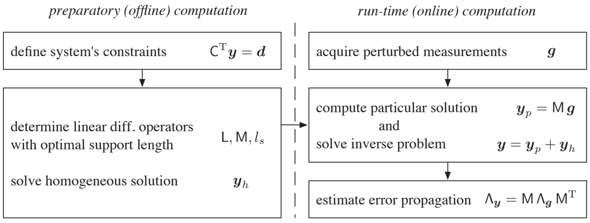
$$\mathsf{L}\,\boldsymbol{y} = \boldsymbol{g} \quad \text{subject to} \quad \mathsf{C}^{\mathrm{T}}\,\boldsymbol{y} = \boldsymbol{d}, \tag{1}$$

where $\boldsymbol{y}$ is the solution vector, $\boldsymbol{g}$ is the discrete perturbed forcing function vector, L is the discretized linear differential operator, the matrix C and the vector $\boldsymbol{d}$ are the discretized constraints. The ODE's solution for $\boldsymbol{y}$ is then efficiently computed as

$$\boldsymbol{y} = \mathsf{M}\,\boldsymbol{g} + \boldsymbol{y}_h. \tag{2}$$

The online solution of the inverse ODE is reduced to a single matrix multiplication and a vector addition.

Both, the constrained linear differential operator M and the ODE's homogeneous solution $\boldsymbol{y}_h$, can be computed offline with high-precision arithmetic as a preparatory step. Only the sensor data $\boldsymbol{g}$ changes with each measurement during run-time. The computational complexity is $\mathcal{O}(n^2)$; it is independent of the equation being solved and the placement of the sensor nodes.


preparatory (offline) computation

define system's constraints $\mathsf{C}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{d}$

determine linear diff. operators with optimal support length $\mathsf{L}, \mathsf{M}, l_s$

solve homogeneous solution $\boldsymbol{y}_h$

run-time (online) computation

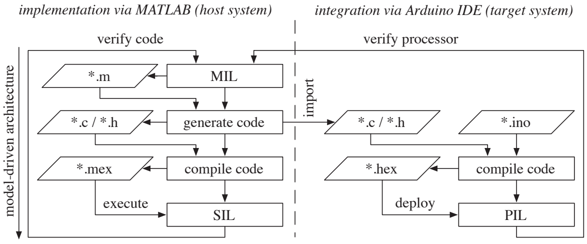
acquire perturbed measurements $\boldsymbol{g}$

compute particular solution $\boldsymbol{y}_p = \mathsf{M}\,\boldsymbol{g}$ and solve inverse problem $\boldsymbol{y} = \boldsymbol{y}_p + \boldsymbol{y}_h$

estimate error propagation $\Lambda_{\boldsymbol{y}} = \mathsf{M}\,\Lambda_{\boldsymbol{g}}\,\mathsf{M}^{\mathrm{T}}$

The system's uncertainty can be estimated from the solution's covariance $\Lambda_{\boldsymbol{y}}$, which is directly computable from the forcing function's covariance $\Lambda_{\boldsymbol{g}}$ via $\Lambda_{\boldsymbol{y}} = \mathsf{M}\,\Lambda_{\boldsymbol{g}}\,\mathsf{M}^{\mathrm{T}}$.

Code Generation and In-the-Loop Verification


implementation via MATLAB (host system)

integration via Arduino IDE (target system)

model-driven architecture

verify code

verify processor

import

*.m

MIL

*.c / *.h

generate code

*.mex

compile code

execute

SIL

*.c / *.h

*.ino

*.hex

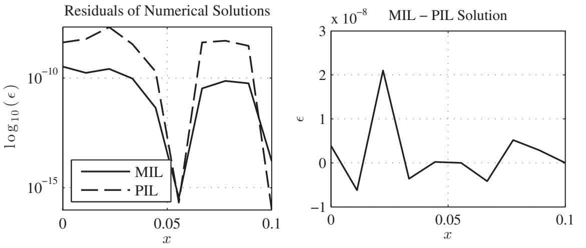
compile code

deploy

PIL

The C code is generated via the MATLAB Coder Toolbox from the model-in-the-loop (MIL) system and verified via software-in-the-loop (SIL) and processor-in-the-loop (PIL) testing for the Atmel 8-bit ATmega328 based Arduino Uno microcontroller.

Conclusions and Results



Numerical testing with an inverse boundary value problem of $n = 10$ nodes revealed a MIL error of $< 10^{-9}$ between the analytical and numerical solution, a SIL error of $< 10^{-13}$ of the C code and a PIL error of $< 10^{-7}$ between a 32-bit PC and the 8-bit microcontroller. A sample rate of > 550 [Hz] is reachable on the embedded system for the run-time computation. This proofs the real-time capability and numerical efficiency of the new numerical ODE solver, making it suitable for deployment even on a low-performance and power-saving embedded system directly in the field.

Chair of Automation, University of Leoben
Peter-Tunner Straße 27, 8700 Leoben, Austria
Email: christoph.gugg@unileoben.ac.at
URL: http://automation.unileoben.ac.at

12.2 IEEE - Sensors 2014

November 2 - November 5, 2014 @ Valencia, Spain [161]

Citation[2]

```
@inproceedings{OlearyGuggHarkerRath2014sensors,
 author     = {O’Leary, Paul and Gugg, Christoph and
               Harker, Matthew and Rath, Gerhard},
 title      = {Mathematical Model and Software Architecture for the
               Solution of Inverse Problems Involving Sensor Arrays},
 booktitle  = {IEEE Sensors Proceedings},
 year       = {2014},
 pages      = {586-589}}
```

Keywords

cyber-physical systems; ordinary differential equations; partial differential equations; perturbed measurements; inclinometers; geo-mechanical monitoring; structural health monitoring; processor-in-the-loop verification;

[2] IEEE Xplore Digital Library, `ieeexplore.ieee.org`

Mathematical Model and Software Architecture for the Solution of Inverse Problems Involving Sensor Arrays

Paul O'Leary, Christoph Gugg, Matthew Harker and Gerhard Rath
Chair of Automation, University of Leoben, A-8700 Leoben, Austria

***Abstract*—This paper presents a mathematical model and partitioning for a software system for the solution of inverse problems involving arrays of sensors. Handling the data from the array of sensors as vectors and matrices, while defining the inverse problem as a least squares computation with linear constraints, leads naturally to the use of matrix algebra for the solution of the system of equations. The matrix algebra is also well suited for the use of automatic code generation to support the rapid development of embedded code. The full functionality of the proposed methods was demonstrated with an inclinometer based system for the monitoring of structural deformation.**

I. Definition and Introduction

There are many different definitions for what constitutes a cyber physical system (CPS) [1]–[10]. The most succinct and pertinent to this paper is the definition given be the Institute of Electrical and Electronics Engineers (IEEE) [1] as well as the Association for Computing Machinery (ACM)[1]: *A CPS is a system with a coupling of the cyber aspects of computing and communications with the physical aspects of dynamics and engineering that must abide by the laws of physics. This includes sensor networks, real-time and hybrid systems.*

In this paper we present a matrix algebra formulation for the solution of inverse problems which utilize the ordinary differential equation (ODE) describing the physical behavior of the system, together with their naturally occurring constraints. This ensures that the solution computed from the sensor data abides by the laws of physics describing the system being monitored. Such inverse problems are of general interest when monitoring critical infrastructure [11]. An introduction to inverse problems in measurement can be found in [12].

The method is applied to the measurement of structural deformations [13] and geo-mechanical monitoring. A good overview of applications can be found in Machan and Bennett [14]. Further applications, in the monitoring of rigid structures, can be found in [15]–[17]. None of the cited literature deals with the solution of over constrained systems. This paper extends on the work previously published [18], [19] by developing a procedure which works for both homogeneous and non-homogeneous arbitrary constraints. Additionally a system partitioning and software architecture is presented for the implementation of a CPS for monitoring critical infrastructure.

II. Mathematical Model and Software Architecture.

A schematic diagram for the architecture proposed for the monitoring of large structures with an array of sensors is shown in Fig. 1. Starting from the left in this figure: there is an array of sensors mounted on a large physical structure. The sampled sensor data can be concatenated to form a vector of measurement values $\boldsymbol{m}$. These are spatially and temporally discrete observations of a continuous system. Obtaining the desired measurement result $\boldsymbol{y}$ from $\boldsymbol{m}$ is an inverse problem associated with an ODE. Now concatenating a time sequence of n measurements yields a matrix $\mathsf{M} = [\boldsymbol{m}_1 \ldots \boldsymbol{m}_n]$; whereby the vertical and horizontal directions correspond to the spatial and time domains respectively. Consequently, starting from M it is possible to perform both spatial and temporal processing simultaneously. This corresponds to inverse partial differential equations (PDE).

Handling the data in this manner and defining the inverse problem as a least squares computation with linear constraints, leads naturally to the use of matrix algebra for the solution of the system of equations.

A. Inverse problems for ODEs

The aim is to reconstruct the response of the system $\boldsymbol{y}$, given: the linear differential operator [20] L describing the behavior of the sensors; an algebraic description [19] $\mathsf{A}\,\boldsymbol{y} = \boldsymbol{d}$ for the constraints implicitly defined by the physical structure of the system; and a set of basis functions B suitable to model the modes of the system, i.e., a linear combination of the modes describes the deformation $\mathsf{B}\,\boldsymbol{z} = \boldsymbol{y}$. The problem to be

[1] ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS) (iccps.acm.org)

solved is,

$$\min_{\boldsymbol{y}} \| \mathsf{L}\, \boldsymbol{y} - \boldsymbol{m} \|_2^2, \tag{1}$$

$$\text{given} \quad \mathsf{A}\, \boldsymbol{y} = \boldsymbol{d}, \tag{2}$$

$$\text{and} \quad \boldsymbol{y} = \mathsf{B}\, \boldsymbol{z}. \tag{3}$$

The matrix of basis functions B is orthonormal, i.e., $\mathsf{B}^{\mathrm{T}}\mathsf{B} = \mathsf{I}$. There is a numerically very efficient direct algebraic solution to this set of equations: defining $\mathsf{H} \triangleq \mathsf{B}\,\{\mathsf{A}\,\mathsf{B}\}^{+}$, $\mathsf{P} \triangleq \mathsf{B}_c\,\{\mathsf{L}\,\mathsf{B}_c\}^{+}$ and $\mathsf{K} \triangleq (\mathsf{I} - \mathsf{P}\,\mathsf{L})\,\mathsf{H}$, yields,

$$\boxed{\boldsymbol{y} = \mathsf{K}\,\boldsymbol{d} + \mathsf{P}\,\boldsymbol{m}.} \tag{4}$$

The constrained basis functions B_c fulfill the constraints in a homogeneous manner, their derivation and generation are described in [19]. The matrices K and P can be computed a-priori, consequently solving the inverse problem only requires the computation of Eqn. 4. The exact number of computations and the memory requirements are known in advance, making the solution suitable for real-time computations. This portion of the computation is performed on the embedded system, enabling each node to independently determine if a measurement reveals a problem with the system being monitored. In this manner each node can notify the appropriate users via web-services of a potential problem independent of the availability of the cloud.

B. *Inverse problems for PDEs*

To enable long-term monitoring and trend analysis, each measurement $\boldsymbol{m}$ is sent via a TCP/IP link to the supervisory system, see Fig. 1. The individual results are now concatenated to form a matrix M. The most general formulation for the PDE describing Y in terms of M is,

$$\mathsf{M} = \mathsf{Q}_s\,\mathsf{Y}\,\mathsf{Q}_t^{\mathrm{T}} + \mathsf{G}_s\,\mathsf{Y} + \mathsf{Y}\,\mathsf{G}_t^{\mathrm{T}} + \mathsf{E}. \tag{5}$$

The subscripts s and t indicate that the matrix is operating in the spatial or time domains respectively. Solving this equation for Y given M is an inverse PDE problem. For many cases, e.g., surface reconstruction from gradient fields [21], there are very efficient numerical solutions to this equation. An example of surface reconstruction with regularization from a sequence of measurements is shown in Fig.4. this computation has the fundamental structure,

$$\mathsf{Y} = \mathsf{V}_s\,\mathsf{M}\,\mathsf{V}_t^{\mathrm{T}} - \mathsf{W}. \tag{6}$$

III. Experimental Testing

The experimental setup[2] used to test the methods proposed in this paper is shown in Fig. 2. It consists of a deformable beam of length $l = 1540\,mm$ with a series of 14 inclinometers which measure the local gradients. A digital serial bus connects the chain of sensors to an ARM based embedded computing platform. The system software has been partitioned according to the structure shown in Fig. 1. Automatic code generation has been used to generate the C-code required to solve the inverse problem involving the ODE. This portion of the system, i.e., solving Eqn. 4 in real-time, is embedded on the ARM processor. The results of the computation are made available via a TCP/IP connection and via a web-server.

The data from the embedded system were acquired and concatenated on a supervisory system. Here the matrix equation (Eqn. 6) is solved to implement simultaneous reconstruction and filtering. The result is a surface representing the deformation of the structure as a function of time.

A. *Inverse problems for ODEs*

For these measurements the left end $x = 0$ of the beam was fixed at $y_1 = 0\,mm$ and the right end $x = 1540\,mm$ at $y_{14} = 80\,mm$ to simulate homogeneous and non-homogeneous constraints respectively. Then the beam was subjected to three different deformations, the local gradients were measured and the curves were reconstructed from these gradients while fulfilling the constraints. Additionally, mechanical calipers were used to perform reference measurements, enabling a verification of the measurement method. The reconstructions from the three measurements, together with their reference mechanical measurements are shown in Fig. 3. There is an excellent agreement between the curve reconstruction from gradients and the reference data.

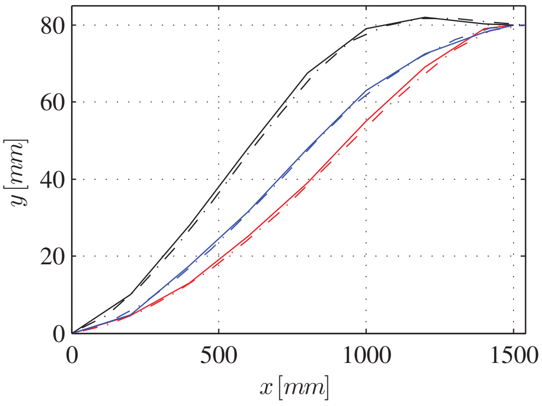


Fig. 3. Three independent measurements with different deformations of the beam. The continuous lines correspond to the reference mechanical measurements and the dashed lines are the respective reconstructions from the gradient data. The left end of the beam was fixed at $y_1 = 0\,mm$ and the right end at $y_{14} = 80\,mm$ to simulate homogeneous and non-homogeneous constraints respectively.

[2] The raw sensor data, the reference measurements and the code to perform the reconstruction are made available at http://www.mathworks.at/matlabcentral/fileexchange/authors/321598.

B. *Inverse problems for PDEs*

In an additional test $n = 100$ measurements were made and concatenated to form a measurement matrix M. The PDE corresponding to reconstruction in the spatial domain and low-pass filtering in the time domain was implemented with the following equation:

$$\mathsf{Y} = \{\mathsf{K}\,[\boldsymbol{d}\ldots\boldsymbol{d}] + \mathsf{P}\,\mathsf{M}\}\,\mathsf{F}. \tag{7}$$

The matrix $\mathsf{F} \triangleq \mathsf{B}(:, 1 : j)\,\mathsf{B}^{\mathrm{T}}(:, 1 : j)$ is the projection onto a set of discrete orthogonal polynomials truncated to have j basis functions. The results for the simultaneous reconstruction and filtering are shown in Fig. 4.

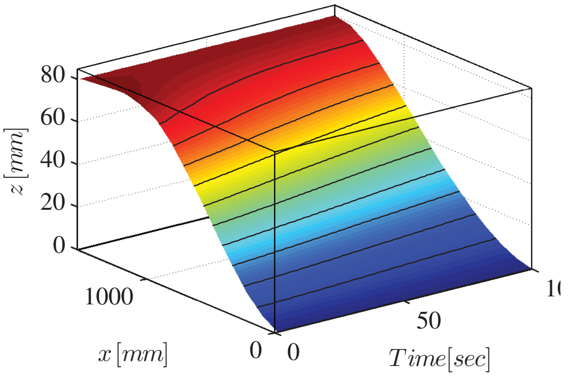


Fig. 4. Surface reconstruction from $n = 100$ sequential measurements of local gradients, 10 intermediate contours are shown on the surface.

IV. Conclusion

This paper has presented a mathematical model and partitioning for a software system for the solution of inverse problems involving arrays of sensors. Handling the data in the proposed manner and defining the inverse problem as a least squares problem with linear constraints, leads naturally to the use of matrix algebra for the solution of the system of equations. The full functionality of the proposed methods was demonstrated with an inclinometer based system for the monitoring of structural deformation.

Acknowledgment

We wish to thank the company GEODATA Messtechnik GmbH, Göss, Austria, who made the sensors available for the measurement setup.

References

[1] R. Baheti and H. Gill, “Cyber-physical systems,” *The Impact of Control Technology*, pp. 161–166, 2011.

[2] E. Geisberger and M. Broy, *agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems*. Springer, 2012, vol. 1.

[3] D. Spath, O. Ganschar, S. Gerlach, M. Hämmerle, T. Krause, and S. Schlund, *Produktionsarbeit der Zukunft-Industrie 4.0.* Fraunhofer IAO Stuttgart, 2013.

[4] S. C. for Foundations in Innovation for Cyber-Physical Systems, “Strategic r&d opportunities for 21st century cyber-physical systems,” National Institute of Standards and Technology (NIST), Tech. Rep., January 2013. [Online]. Available: www.nist.gov

[5] E. Incorporated, “Cyber-physical systems: Situation analysis of current trends, technologies, and challenges,” National Institute of Standards and Technology (NIST), Tech. Rep., March 2012. [Online]. Available: www.nist.gov

[6] K.-J. Park, R. Zheng, and X. Liu, “Cyber-physical systems: Milestones and research challenges,” *Computer Communications*, vol. 36, no. 1, pp. 1–7, 2012.

[7] F. IOSB, “Industry 4.0 information technology is the key element in the factory of the future,” Press Information, January 2013.

[8] D. Spath, S. Gerlach, M. Hämmerle, S. Schlund, and T. Strölin, “Cyber-physical system for self-organised and flexible labour utilisation,” *Personnel*, vol. 50, p. 22.

[9] E. A. Lee, “Cyber physical systems: Design challenges,” in *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*. IEEE, 2008, pp. 363–369.

[10] P. Tabuada, “Cyber-physical systems: Position paper,” in *NSF Workshop on Cyber-Physical Systems*, 2006.

[11] D. Necsulescu and G. Ganapathy, “Online solving of inverse problems in critical infrastructure monitoring,” in *Instrumentation and Measurement Technology Conference, 2005. IMTC 2005. Proceedings of the IEEE*, vol. 2, 2005, pp. 1585–1589.

[12] G. Solopchenko, “Inverse problems in measurement,” *Measurement*, vol. 5, no. 1, pp. 10 – 19, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0263224187900236

[13] *Standard Test Method for Monitoring Ground Movement Using Probe-Type Inclinometers*, American Society for Testing and Materials Norm D6230 - 13, 2013.

[14] G. Machan and V. G. Bennett, “Use of inclinometers for geotechnical instrumentation on transportation projects,” *Transportation Research E-Circular*, vol. E-C129, 2008. [Online]. Available: http://worldcat.org/issn/00978515

[15] X. Hou, X. Yang, and Q. Huang, “Using inclinometers to measure bridge deflection,” *Journal of Bridge Engineering*, vol. 10, no. 5, pp. 564–569, 2005. [Online]. Available: http://link.aip.org/link/?QBE/10/564/1

[16] J. Van Cranenbroeck, “Continuous beam deflection monitoring using precise inclinometers,” in *FIG Working Week 2007*, Hong Kong, SAR, 13..17 May, 2007.

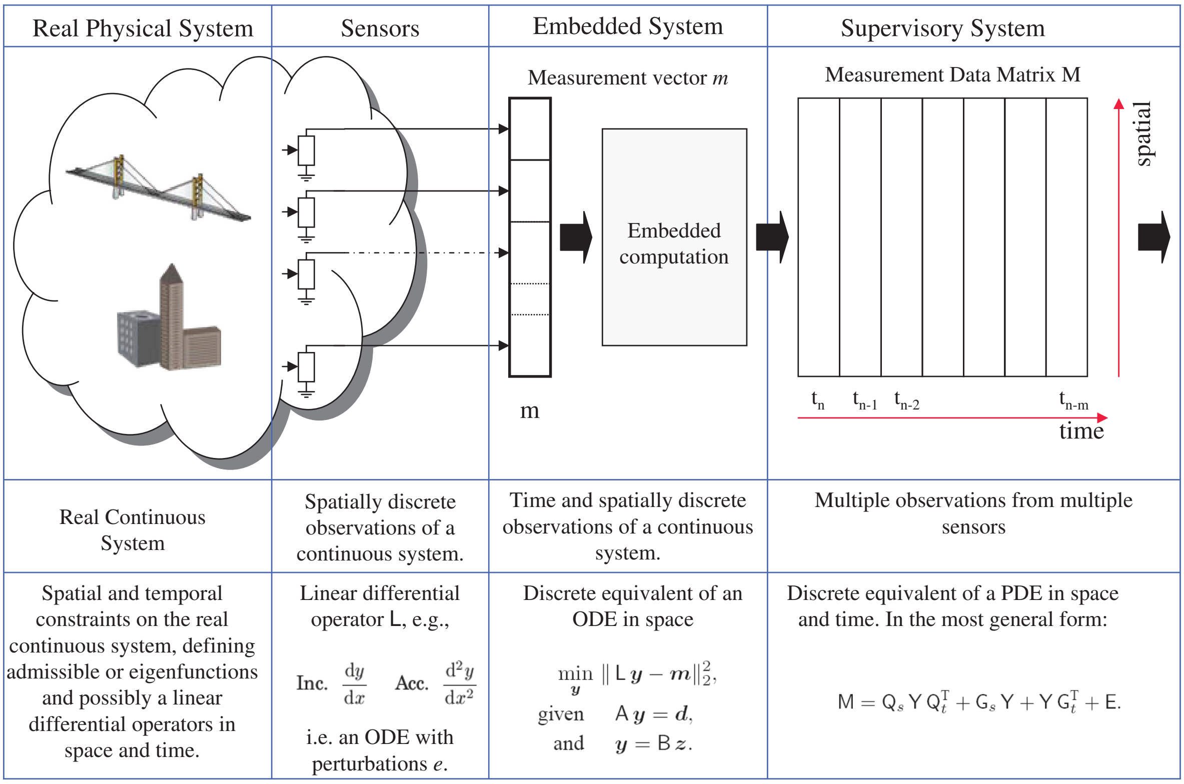
[17] J. Golser, “Fallbeispiel zur Bauwerksberwachung mittels online Neigungssensoren,” in *25. Cristian Veder Kolloquium*, 2010.

[18] P. O’Leary and M. Harker, “Direct discrete variational curve reconstruction from derivatives and its application to track subsidence measurements,” in *International Instrumentation and Measurement Technology Conference (I2MTC 2011)*, 5 2011.

[19] ——, “A framework for the evaluation of inclinometer data in the measurement of structures,” *IEEE T. Instrumentation and Measurement*, vol. 61, no. 5, pp. 1237–1251, 2012.

[20] C. Lanczos, *Linear Differential Operators*, ser. Dover books on mathematics. Dover Publications, 1997. [Online]. Available: http://books.google.at/books?id=WLEVEY-3gIAC

[21] M. Harker and P. OLeary, “Regularized reconstruction of a surface from its measured gradient field,” *Journal of Mathematical Imaging and Vision*, pp. 1–25, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10851-014-0505-4



Real Physical System	Sensors	Embedded System	Supervisory System
Real Continuous System	Spatially discrete observations of a continuous system.	Time and spatially discrete observations of a continuous system.	Multiple observations from multiple sensors
Spatial and temporal constraints on the real continuous system, defining admissible or eigenfunctions and possibly a linear differential operators in space and time.	Linear differential operator L, e.g., Inc. $\frac{\mathrm{d}y}{\mathrm{d}x}$ Acc. $\frac{\mathrm{d}^2 y}{\mathrm{d}x^2}$ i.e. an ODE with perturbations *e*.	Discrete equivalent of an ODE in space $\min_{\boldsymbol{y}} \|\mathsf{L}\,\boldsymbol{y} - \boldsymbol{m}\|_2^2,$ given $\mathsf{A}\,\boldsymbol{y} = \boldsymbol{d},$ and $\boldsymbol{y} = \mathsf{B}\,\boldsymbol{z}.$	Discrete equivalent of a PDE in space and time. In the most general form: $\mathsf{M} = \mathsf{Q}_s\,\mathsf{Y}\,\mathsf{Q}_t^{\mathrm{T}} + \mathsf{G}_s\,\mathsf{Y} + \mathsf{Y}\,\mathsf{G}_t^{\mathrm{T}} + \mathsf{E}.$

Fig. 1. Schematic of the system architecture, showing the emergence of the algebraic formulation for the solution of the computation associated with an array of sensors.

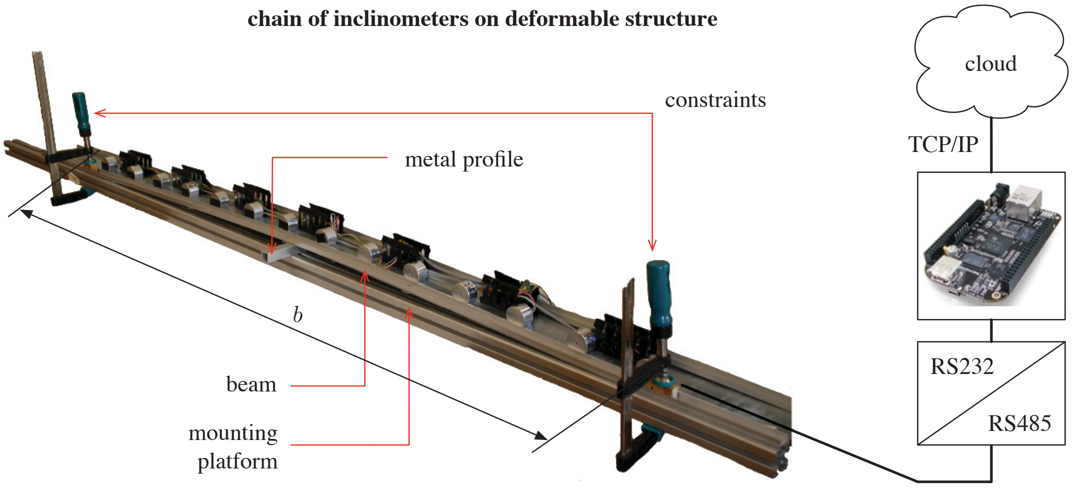


Fig. 2. Experimental setup for an array of sensors, connected via a digital bus to an embedded computation node. The inverse problem is solved locally and the result is made available via TCP/IP and web server.

12.3 arXiv - Article 2014

Open Access Journal Article [73]

Citation[3]

```
@article{GuggOlearyHarker2014arxiv,
 author     = {Gugg, Christoph and O'Leary, Paul and Harker, Matthew},
 title      = {An Algebraic Framework for the Real-Time Solution
               of Inverse Problems on Embedded Systems},
 journal    = {arXiv preprint, arXiv:1406.0380},
 year       = {2014}}
```

Keywords

cyber-physical systems; embedded systems; inclinometers; perturbed measurements; numerical solver; ordinary differential equations; inverse problems; constraints; model based design; automatic code generation; in-the-loop verification;

[3] Cornell University Library, `arxiv.org`

An Algebraic Framework for the Real-Time Solution of Inverse Problems on Embedded Systems

Christoph Gugg, Matthew Harker, Paul O'Leary and Gerhard Rath

This article presents a new approach to the real-time solution of inverse problems on embedded systems. The class of problems addressed corresponds to ordinary differential equations (ODEs) with generalized linear constraints, whereby the data from an array of sensors forms the forcing function. The algebraic discretization of the problem enables a one-to-one mapping of the ODE to its discrete equivalent linear differential operator, together with an additional matrix equation representing the constraints. The solution of the equation is formulated as a least squares (LS) problem with linear constraints. The LS approach makes the method suitable for the explicit solution of inverse problems where the forcing function is perturbed by noise. The algebraic computation is partitioned into a initial preparatory step, which precomputes the matrices required for the run-time computation; and the cyclic run-time computation, which is repeated with each acquisition of sensor data. The cyclic computation consists of a single matrix-vector multiplication, in this manner computation complexity is known a-priori, fulfilling the definition of a real-time computation. Numerical testing of the new method is presented on perturbed as well as unperturbed problems; the results are compared with known analytic solutions and solutions acquired from state-of-the-art implicit solvers. In all performed numerical tests the new method was both faster and more accurate for repeated solutions of the same ODE. The solution is implemented with model based design and uses only fundamental linear algebra; consequently, this approach supports automatic code generation for deployment on embedded systems. The targeting concept was tested via software- and processor-in-the-loop verification on two systems with different processor architectures. Finally, the method was tested on a laboratory prototype with real measurement data for the monitoring of flexible structures. The measurement arrangement consists of an embedded system with a chain of 14 inclinometer sensors connected to it, two additional nodes implement a total of four constraints. The problem solved is: the real-time overconstrained reconstruction of a curve from measured gradients. Such systems are commonly encountered in the monitoring of structures and/or ground subsidence.

Categories and Subject Descriptors: C.3 [**Special-Purpose and Application-Based Systems**]: Real-Time and Embedded Systems; G.1.7 [**Ordinary Differential Equations**]: Boundary value problems; I.2.2 [**Automatic Programming**]: Program transformation

General Terms: Design, Performance, Experimentation

Additional Key Words and Phrases: cyber-physical systems, embedded systems, inclinometers, measurement, numerical solver, ordinary differential equations, inverse problems, constraints, model based design, automatic code generation, in-the-loop verification

1. MOTIVATION AND PROBLEM STATEMENT

The original motivation for this work was the development of a large scale cyber-physical system (CPS) to monitor ground subsidence and possible deformation of structures during the construction of the new City-Circle Line subway in Copenhagen, Denmark. Very stringent geo-mechanical monitoring requirements have been established

Author's address: Chair of Automation, Department Product Engineering, University of Leoben, 8700 Leoben, Austria; URL: automation.unileoben.ac.at; Email: christoph.gugg@unileoben.ac.at;

for underground construction projects in urban areas following an accident on March 3, 2009 in Cologne, Germany: the building of the city's archive collapsed into a Stadtbahn tunnel under construction on the Severinstraße, killing two people[1]. The monitoring concept consists of a large number of vertical holes sunk along the planned path of the tunnel distributed over a distance of approximately $15\,[\mathrm{km}]$. Each of these holes is equipped with a series of rods, and each rod is equipped with a pair of inclinometers, effectively forming a chain of inclinometers. Chains of inclinometers are used in the monitoring of ground subsidence [Machan and Bennett 2008] and for measuring the deformation of structures [O'Leary and Harker 2012]. Determining the ground movement from the orientation of the rods is an inverse problem. Additionally, there are points where constraints are placed on the construction, for example pillars, which in turn define initial-, inner- or boundary values for the inverse problems. Reconstructing the deformation under these circumstances requires the solution of an inverse boundary value problem for each chain of rods. Consequently, it is necessary to solve a large number of inverse initial-, inner- or boundary value problems in real-time for different sets of measurement data. Each chain of inclinometers is equipped with an embedded system that acquires and processes the data from the sensors, forming an independent sensor node. The individual sensor nodes are part of a larger sensor network. Decentralized processing of measurement data introduces an implicit form of parallelism thanks to distributed computing. The network's bandwidth demands are lowered due to the higher information density.

Necsulescu also identified the necessity of solving inverse problems in critical infrastructure monitoring [Necsulescu and Ganapathy 2005]. Lee [Lee et al. 2012] identified that predictable real-time solutions of complex systems, with an understandable concurrency, are a key issue for future developments of CPS. He points out that this issue was inadequately dealt with in the past. There are numerous engineering and scientific applications which require the real-time solution of inverse problems, e.g. [Loh and Dickin 1996]. Therefore, this is clearly an area of research which is of significance.

2. SCOPE OF THE ARTICLE

This article develops a new method for the numerical solution of inverse problems based on a matrix algebraic approach. It provides global least squares solutions to inverse initial-, inner- or boundary value problems. The method has been developed specifically with the aim of solving inverse problems associated with measurement systems in an efficient manner, whereby multiple measurements are performed over time and repeated solutions of the same equation are required. The goal is to directly embed the solver onto the sensor node's hardware. The main contributions of the article are:

(1) A new algebraic approach to the numerical solution of inverse problems is derived. The method splits the calculations into two portions: a preparatory (offline) computation and a run-time (online) computation. The run-time computation is repeatedly performed with each new measurement. Solving the inverse problem at run-time is reduced to one matrix multiplication and one vector addition. In this manner, the exact number of floating point operations (FLOPs) is known a-priori, $W(n) = 2\,n^2$, where n is the number of measurement points. Additionally, the memory requirements are known in advance. Consequently, a strict upper-bound $\mathcal{O}(n^2)$ can be determined for the execution time on a given processor. This makes the method, by definition, suitable for real-time applications. Furthermore, the covariance propagation for perturbations of the sensor inputs to the solution is derived.

[1] An article relating to the incident can be found at http://www.ksta.de/html/artikel/1266930835566.shtml

This enables the computation of a confidence interval for the solution. The runtime computational complexity of estimating the confidence is $\mathcal{O}(n)$.
Extensive model-in-the-loop (MIL) testing of the method on a personal computer (PC) is presented to validate the method. The results of a classical Runge-Kutta type approach are compared with those obtained using the new approach. The results demonstrate the accuracy of the method and its numerical efficiency.

(2) A model based design (MBD) approach is presented which enables the system formulation at an abstract level. The presented model only utilizes fundamental linear algebra operations such as matrix multiplication and vector addition; consequently, automatic generation of C code becomes possible. Software-in-the-loop (SIL) verification is used to proof the functional equivalence of the model and the generated code. Embedded targeting enables the deployment of the code directly onto a microcontroller. The results computed by the embedded processor are compared to the results computed by the model running on a PC via processor-in-the-loop (PIL) verification. The viability of the model is demonstrated on a very limited, yet cheap and available, 8-bit microcontroller. Furthermore, a laboratory setup with a chain of inclinometers mounted on a flexible structure demonstrates the applicability of the model for real measurement data.

3. CONTINUOUS MEASUREMENT MODEL

The measurement model is central to this article: it defines the class of problem which is being solved. Furthermore, it defines the requirements for the MBD environment. The aim is to use MBD to automatically generate the functionally equivalent code which is capable of solving any example of this problem on an embedded system in real-time.

The class of inverse problems being considered in this article consist of an ordinary differential equation (ODE) of degree m of the form

$$a_m(x)\, y^{(m)} + a_{m-1}(x)\, y^{(m-1)} + \ldots + a_1(x)\, y' + a_0(x)\, y = g(x), \tag{1}$$

where y is a function of x, $y^{(i)}$ is the notation for the i^{th} derivative of y with respect to x, $a_i(x)$ are the coefficient functions and $g(x)$ is the forcing function. Additionally, a minimum of m independent initial-, inner- or boundary values are required to ensure that there is a unique solution to the equation. The n measurements, forming the vector $\boldsymbol{g}$, correspond to discrete samples of the forcing function $g(x)$. The n measurements may emanate from n sensors forming a spatial array or from a time sequence of n measurements from one single sensor. In this class of problems, the forcing function $g(x)$, the input, is considered to be perturbed, since it is formed from measurements which are subject to noise. Only the forcing function $g(x)$ changes from one measurement to the next. The task is to recompute $y(x)$ for each new measurement $g(x)$. This type of problem occurs, for example, in the monitoring of structures [Burdet and Zanella 2002; Golser 2010; Harker and O'Leary 2013a].

The new method can, however, deal with overconstrained systems, i.e., there are p independent constraints whereby $p > m$. The initial-, inner- or boundary values correspond to constraints on the function value $y(x)$ or its derivatives $y^{(i)}(x)$ at specific x locations. Both Dirichlet and Neumann boundary conditions are special cases of such constraints. The nature of the constraints determines if the system is considered to be an initial value (IVP) or boundary value (BVP) or inner value problem.

One peculiarity of this class of inverse problems is: that the abscissae, i.e., the positions where the solutions are required, is determined by the measurements; these positions are called the nodes. In the case of a chain of sensors, the physical position of the sensor corresponds to the abscissae x. In temporal sequences, it is the time points of the individual measurements which define the abscissae. Consequently, we are not free to select the positions of where the ODE is to be solved. This precludes the use of variable step size algorithms. A further consequence is that a general framework for this type of inverse problem has to be capable of computing the solution for arbitrary nodes.

4. THEORY OF ORDINARY DIFFERENTIAL EQUATIONS

Some preliminary theory is required if an objective evaluation of previous work is to be performed. The numerical solution of an inverse problem requires the discrete approximation of a continuous system. Consequently, we can derive properties of the continuous operations which must be fulfilled by the corresponding discrete operators. We first define the continuous domain differential operator D such that, $D^{(i)}\, y \equiv y^{(i)}$. Most commonly, the discrete implementation of the differentiating matrix is implemented using polynomial interpolation. The properties of D with respect to a polynomial are essential to the desired behavior of numerical differentiation. Defining a power series approximation for y with coefficients c_i,

$$y = \sum_{i=0}^{m} c_i\, x^i. \tag{2}$$

Applying the differential operator D yields,

$$y' = D\, y = \sum_{i=0}^{m} i\, c_i\, x^{i-1} = \sum_{i=1}^{m} i\, c_i\, x^{i-1}. \tag{3}$$

By definition of the derivative, the constant portion of the polynomial differentiates to zero, hence the constant coefficient c_0 vanishes. We assume that D is composed of formulae which are consistent, in the sense that in the limit they define a derivative. If this is the case then the matrix D should satisfy the following properties, such that D is a consistent discrete approximation to the continuous operator D:

(1) The matrix D must be rank-1 deficient; i.e, its null space is of dimension one.
(2) The null space of D must be spanned by the constant vector $\boldsymbol{1}\,\alpha$; equivalently, the row-sums of D are all zero,

$$\mathsf{D}\,\boldsymbol{1}\,\alpha = \boldsymbol{0}. \tag{4}$$

These conditions ensure that the differentiating matrix D is consistent with the continuous domain definition of the derivative. Given that, interpolating polynomials are unique, the formula for the derivative should be independent of the particular polynomials chosen for interpolation. However, differences do lie in the numerical behavior of different formulas; regardless, a given set of nodes, $\boldsymbol{x}$, should uniquely define the differentiating matrix of a given polynomial degree of accuracy.

For the purpose of treating ODEs, we use the general notion of a linear differential operator [Lanczos 1997]. Specifically, by substituting the continuous differential operator D for the differentials $y^{(i)}$ in Eqn. (1) yields,

$$a_m(x)\, D^{(m)}\, y + a_{m-1}(x)\, D^{(m-1)}\, y + \ldots + a_1(x)\, D\, y + a_0(x)\, y = g(x). \tag{5}$$

Factoring y to the right yields,

$$\left\{ a_m(x)\, D^{(m)} + a_{m-1}(x)\, D^{(m-1)} + \ldots + a_1(x)\, D + a_0(x) \right\}\, y = g(x). \tag{6}$$

The linear differential operator L for the continuous equation can now be defined as,

$$L \triangleq a_m(x)\, D^{(m)} + a_{m-1}(x)\, D^{(m-1)} + \ldots + a_1(x)\, D + a_0(x). \tag{7}$$

Consequently, Eqn. (1) is written as,

$$\boxed{L\,y = g(x).} \tag{8}$$

5. OVERVIEW OF NUMERICAL ODE SOLVERS

The *Taylor matrix* uses the known analytical relationship between the coefficients, s, of a Taylor polynomial and those of its derivatives, $\dot{s}$, to compute a differentiating matrix D for the solution of ODEs [Kurt and Cevik 2008]. The matrix D together with the matrix of basis functions arranged as the columns of the matrix B are used to compute numerical solutions to the differential equations. The method of the Taylor matrix was extended to the computation of fractional derivatives [Keskyn et al. 2011]. The most serious problem associated with the Taylor matrix approach is that it requires the inversion of the Vandermonde matrix, a process which is numerically unstable. The errors in the differentiating matrix are strongly dependent on the degree of the polynomial, i.e., the number of nodes and the node placement.

A *Chebyshev matrix* approach was presented by Sezer [Sezer and Kaynak 1996] and others [Welfert 1997; Weideman and Reddy 2000; Driscoll et al. 2008; Jewell 2013]. The approach is fundamentally the same as for the Taylor matrix, whereby the Chebyshev polynomials are used as an alternative to geometric polynomials. The advantage of defining polynomials on the Chebyshev points is that they deliver stable polynomials and differentials. The main disadvantage, however, is that the numerical solution to the differential equations is restricted to the locations of the Chebyshev points; this lacks the generality needed for inverse problems[2] being considered here.

Synthesizing differentiating matrices for arbitrary nodes is an issue one might assume has been sufficiently dealt with in literature. However, a closer examination of literature and textbooks shows that some clarification is still necessary. Most books on *spectral and pseudo-spectral techniques*, e.g., [Fornberg 1998], approach differentiation matrices from the view point of simulation and do not consider the connotations of inverse problems. In a simulation, it is in general possible to select the position of the nodes, so that they are well suited to the solution method, e.g., it is possible to use either the Chebyshev or Legendre collocation nodes. This luxury is not given with inverse problems; the placement of the sensors may be arbitrary and or the time points for which solutions are required are evenly spaced. Consequently, it is necessary to generate differentiating matrices for truly arbitrary nodes.

There are a number of papers [Welfert 1997; Weideman and Reddy 2000] which explicitly claim to compute *differentiating matrices using global methods for arbitrary nodes* and there are some toolboxes which suggest this is possible [Jewell 2013]. The published code for all these methods generate degenerate differentiating matrices with null spaces of dimensions higher than one. That is, they do not fulfill the prerequisites defined in Section 4. In contrast, the *local polynomial approximation to differentiation* [Savitzky and Golay 1964] with correct end-point formulas [Burden and Faires 2005] generates a consistent matrix. The poor behavior of high order polynomial interpolation and differentiation is due to Runge's phenomenon, which will be always be present due to the uniqueness of interpolating polynomials; hence, approximations of relatively low degree are preferable to global approaches. The published methods [Welfert 1997; Weideman and Reddy 2000] work reliably only for very small

[2] This is not dismissing the Chebyshev methods, it simply points out that they are limited in their applications. Furthermore, the methods in this article work for truly arbitrary node placements. Consequently, the Chebyshev polynomials are only a special case.

problems[3], $n \leq 10$; this is not sufficient to address most real inverse problems encountered in engineering applications. We conclude that global techniques for computing differentiating matrices are not applicable to large scale inverse problems.

Finite difference methods [Strikwerda 2004; Smith 1985] rarely deal with higher degree approximations, typically 3 or 5 point formulas are used. The issue of correct end point formulas is sacrificed for the advantage of band diagonal matrices. In general, these techniques deal with Dirichlet and possibly Neumann boundary conditions. However, they provide no method of implementing general boundary conditions of the form

$$D^{(i)}\, y(x_j) = d, \tag{9}$$

where $D^{(i)}$ represents the i^{th} derivation of y evaluated at the point $x = x_j$ with the value d. There may be $p \geq m$ such constraints.

A *new matrix approach for the solution of inverse problems*, associated with monitoring of structures using inclinometers, was presented [O'Leary and Harker 2012] and generalized in [Harker and O'Leary 2013a]. It was proven that ODEs can be formulated as a least squares problem with linear constraints, of the form:

$$\mathsf{L}\, \boldsymbol{y} = \boldsymbol{g} \quad \text{subject to} \quad \mathsf{C}^{\mathrm{T}}\, \boldsymbol{y} = \boldsymbol{d}, \tag{10}$$

whereby L is the discretized linear differential operator, $\boldsymbol{y}$ is the solution vector sought (function values), $\boldsymbol{g}$ is the discrete forcing function (measurement values), C defines the type of constraints and $\boldsymbol{d}$ are the values of the constraints. The least squares solution makes the method suitable for problems where the forcing function $g(x)$ is perturbed.

The continuous linear differential operator L in Eqn. (7) is discretized as the matrix L, such that

$$\mathsf{L} \triangleq \mathsf{A}_m\, \mathsf{D}_m + \mathsf{A}_{m-1}\, \mathsf{D}_{m-1} + \ldots + \mathsf{A}_1\, \mathsf{D} + \mathsf{A}_0, \tag{11}$$

where $\mathsf{A}_i = \operatorname{diag}(a_i(\boldsymbol{x}))$, the matrix D_i is a local discrete approximation with support length l_s to the continuous differential operator $D^{(i)}$. Care is taken to implement the correct end-point formulas, ensuring the degree of approximation is constant for the complete support. The details of generating these matrices can be found in [Harker and O'Leary 2013a], as can the explanation for the generation of the constraints $\mathsf{C}^{\mathrm{T}}\, \boldsymbol{y} = \boldsymbol{d}$. Furthermore, MATLAB toolboxes are available [Harker and O'Leary 2013b; 2013c] for all the functions required in this article.

6. SOLVING THE INVERSE PROBLEM

Previously the problem in Eqn. (10) was solved using an efficient and accurate solution which is found in [Golub and Van Loan 1996, Chapter 12]. In this paper we take a different approach to partitioning the numerical computations, which takes advantage of the fact that the inverse problem is to be solved repeatedly. Fundamentally, the new approach delivers exactly the same explicit solution; however, through the new partitioning of the computation it is possible to ensure that the numerical work $W(n)$ and the memory required are run-time are known exactly in advance. Consequently, an exact upper-bound for the execution time can be determined, this by definition makes the solution suitable for real-time applications[4].

The computation of the solution is separated into two portions:

[3] This can be verified by running the available code with $n = 20$. Testing the resulting D matrix or its singular values reveals that a null space of higher dimension is present. As a consequence, the matrix does not fulfill the necessary prerequisites.

[4] A *real-time system* is defined as any information processing activity or system which has to respond to externally generated input stimuli within a finite and specified period [Young 1982].

(1) The preparatory computations which can be performed offline. They are characteristic for the equation being solved and change neither with the acquisition of new measurement data, nor with new values for the boundary conditions. These computations need not be performed on the embedded system and may be computed with higher precision arithmetic on a host system if necessary.
(2) The online computation, which must be performed repeatedly with each new set of sensor data. This is the solution which is computed explicitly on the embedded system in real-time.

6.1. Preparatory Computations

The constraints on the solution are defined by,

$$\mathsf{C}^{\mathrm{T}} \boldsymbol{y} = \boldsymbol{d}. \tag{12}$$

Each column of C, together with the corresponding row of $\boldsymbol{d}$, defines a constraint. Consequently, $p = \operatorname{rank}\{\,\mathsf{C}\,\}$ is the number of linearly independent constraints. Additionally, the constraints must be consistent, i.e., $\boldsymbol{d} \in \operatorname{range}\left\{\,\mathsf{C}^{\mathrm{T}}\,\right\}$. A minimum of $p \geq m$ constraints are required to ensure a unique solution to an ODE of degree m. We now define the matrices: P, such that $\operatorname{range}\{\,\mathsf{P}\,\} = \operatorname{range}\{\,\mathsf{C}\,\}$, i.e., P is the Moore-Penrose pseudo inverse of C^{T}, hence $\mathsf{P} = \{\mathsf{C}^{\mathrm{T}}\}^{+}$; F, an orthonormal basis function set for the null-space of C^{T}, i.e., $\mathsf{F}^{\mathrm{T}}\,\mathsf{F} = \mathsf{I}$ and $\operatorname{range}\{\,\mathsf{F}\,\} = \operatorname{null}\left\{\mathsf{C}^{\mathrm{T}}\right\}$; and $\mathsf{H} \triangleq \mathsf{P} + \mathsf{F}\,\mathsf{R}$, where R is an arbitrary matrix. In this manner the solution for $\boldsymbol{y}$ can be parameterized as,

$$\boldsymbol{y} = \mathsf{H}\,\boldsymbol{d} + \mathsf{F}\,\boldsymbol{\beta}, \tag{13}$$

where $\boldsymbol{\beta}$ is the parameter vector. It is important to realize that neither H nor F are unique. Any function which fulfills the constraints is a valid selection for $\boldsymbol{y}_c$. A function $\boldsymbol{y}_c$ which fulfills the constraints can be defined as,

$$\begin{aligned} \boldsymbol{y}_c &\triangleq \mathsf{H}\,\boldsymbol{d}, && (14)\\ &= \{\mathsf{P} + \mathsf{F}\,\mathsf{R}\}\,\boldsymbol{d}. && (15) \end{aligned}$$

The matrix R is arbitrary, consequently the values can be selected so that $\boldsymbol{y}_c$ fulfills additional conditions without altering the solution for $\boldsymbol{y}$. In Fig. 1 three different solutions for the constraints $y(0) = 1$ and $y(1) = 0$ are shown, to demonstrate this fact. It may be advantageous for a specific problem to select a particular solution for $\boldsymbol{y}_c$ which has desirable properties; for example, when solving the ODE for a cantilever it may be appropriate to select a polynomial solution for $\boldsymbol{y}_c$, since the solution to the ODE is known to be a polynomial. More formally: the matrix $\mathsf{H} = \left\{\mathsf{C}^{\mathrm{T}}\right\}^{-}$ is a generalized inverse [Ben-Israel and Greville 2003] of C^{T}. A generalized inverse A^{-} of a matrix A fulfills the condition,

$$\mathsf{A}\,\mathsf{A}^{-}\,\mathsf{A} = \mathsf{A}. \tag{16}$$

The Moore-Penrose pseudo inverse is the particular generalized inverse, where $\mathsf{R} = 0$; it yields an inverse which minimizes the 2-norm of the solution vector; alternatively, a QR decomposition can be used to compute a generalized inverse which leads to a solution vector with a minimum number of nonzero entries. The selection of an appropriate solution for $\boldsymbol{y}_c$ is more important when solving inverse problems, since it has implications for the implementation of regularization.

The orthonormal basis functions F for the null-space of C^{T} are also not unique. They can be obtained directly from C^{T} by applying QR decomposition and partitioning Q according to the $\operatorname{rank}\{\,\mathsf{R}\,\}$. Alternatively, constrained basis functions, e.g. constrained polynomials, can be used to implement a set of orthogonal basis functions F, Fig. 2 shows an example of such admissible functions for the constraints $y(0) = 1$, and

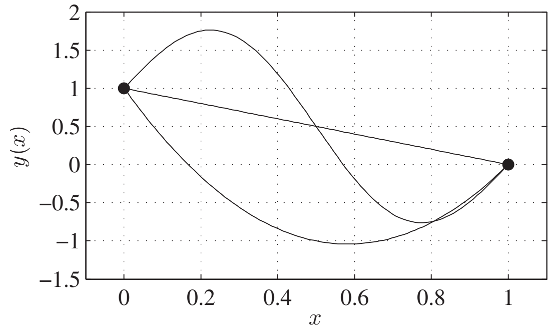


Fig. 1: Three different solutions to the constraints $y(0) = 1$ and $y(1) = 0$.

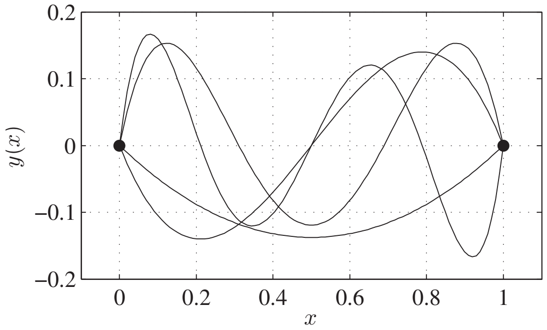


Fig. 2: The first four discrete orthogonal constrained polynomials for the constraints $y(0) = 1$ and $y(1) = 0$, i.e., the first 4 basis functions from F.

$y(1) = 0$. In the case of inverse problems constrained basis functions offer a method of implementing spectral regularization [O'Leary and Harker 2012]. The MATLAB library required to generate discrete orthogonal constrained polynomials is available at [Harker and O'Leary 2013b].

$$\mathsf{C}^{\mathrm{T}}\,\mathsf{F} = 0 \quad \text{and hence} \quad \mathsf{C}^{\mathrm{T}}\,\mathsf{F}\,\boldsymbol{\beta} = \boldsymbol{0}. \tag{17}$$

Substituting Eqn. (13) for $\boldsymbol{y}$ in $\mathsf{L}\,\boldsymbol{y} = \boldsymbol{g}$ now yields an unconstrained algebraic equation for the ODE,

$$\mathsf{L}\,\{\mathsf{H}\,\boldsymbol{d} + \mathsf{F}\,\boldsymbol{\beta}\} = \boldsymbol{g}. \tag{18}$$

In the class of inverse problems being considered in this article, the forcing function $\boldsymbol{g}$ is formed from the measurement values which are perturbed, i.e., is subject to noise. Consequently, the solution of Eqn. (18) is formulated as a least squares problem to obtain the unique global minimum of

$$\min_{\boldsymbol{\beta}} \|\mathsf{L}\,\mathsf{H}\,\boldsymbol{d} + \mathsf{L}\,\mathsf{F}\,\boldsymbol{\beta} - \boldsymbol{g}\|_2^2. \tag{19}$$

The least squares approach has been selected since it delivers a maximum likelihood solution in the case that $\boldsymbol{g}$ is perturbed by Gaussian noise. A further advantage of the global least squares formulation is that the solution has no implicit direction of integration. Avoiding a direction of integration eliminates the problem of accumulation of errors, as are typical with IVP approaches such as Runge-Kutta. Additionally, the least squares approach yields a solution which is globally minimum with respect to all errors in Eqn. (19), i.e., it is also minimizing the consequences of the errors in the numerical computations. Consequently, the method is suitable for solving both perturbed and unperturbed problems. Now solving the minimization problem defined by Eqn. (19) yields,

$$\boldsymbol{\beta} = \{\mathsf{L}\,\mathsf{F}\}^{+}\,\{\boldsymbol{g} - \mathsf{L}\,\mathsf{H}\,\boldsymbol{d}\} + \mathsf{K}\,\boldsymbol{\gamma}, \tag{20}$$

where K is an orthonormal vector basis set for the null-space of $\mathsf{L}\,\mathsf{F}$, i.e., $\mathsf{K}^{\mathrm{T}}\,\mathsf{K} = \mathsf{I}$ and $\mathrm{span}\,\{\mathsf{K}\} = \mathrm{null}\,\{\mathsf{L}\,\mathsf{F}\}$. This equation is now expanded into three relevant terms,

$$\boldsymbol{\beta} = \{\mathsf{L}\,\mathsf{F}\}^{+}\,\boldsymbol{g} - \{\mathsf{L}\,\mathsf{F}\}^{+}\,\mathsf{L}\,\mathsf{H}\,\boldsymbol{d} + \mathsf{K}\,\boldsymbol{\gamma}. \tag{21}$$

A non-empty vector basis set K indicates that the linear differential operator L is not sufficiently constrained to ensure a unique solution, i.e., there is no unique solution to the problem being posed. The requirement for a unique solution is

$$\operatorname{rank}\begin{bmatrix}\mathsf{L}\\ \mathsf{C}^{\mathrm{T}}\end{bmatrix} = n, \tag{22}$$

where n is the number of nodes. This is a method of determining if the problem is well defined. Alternately, the singular values of the matrix can be used to determine if the problem is numerically well posed. We will now assume that the problem is well posed: with this, the term involving γ vanishes. Now back-substituting for $\boldsymbol{\beta}$ in Eqn. (13), yields,

$$\boldsymbol{y} = \mathsf{F}\,\{\mathsf{L}\,\mathsf{F}\}^{+}\,\boldsymbol{g} + \mathsf{H}\,\boldsymbol{d} - \mathsf{F}\,\{\mathsf{L}\,\mathsf{F}\}^{+}\,\mathsf{L}\,\mathsf{H}\,\boldsymbol{d}. \tag{23}$$

Defining the following abbreviations:

$$\mathsf{M} \triangleq \mathsf{F}\,\{\mathsf{L}\,\mathsf{F}\}^{+} \qquad \text{and} \qquad \mathsf{N} \triangleq \{\mathsf{I} - \mathsf{M}\,\mathsf{L}\}\,\mathsf{H}, \tag{24}$$

yields,

$$\boldsymbol{y} = \mathsf{N}\,\boldsymbol{d} + \mathsf{M}\,\boldsymbol{g} \tag{25}$$

$$= \boldsymbol{y}_h + \boldsymbol{y}_p. \tag{26}$$

The homogeneous portion of the solution $\boldsymbol{y}_h = \mathsf{N}\,\boldsymbol{d}$ is only dependent of the constraint values and the particular solution $\boldsymbol{y}_p = \mathsf{M}\,\boldsymbol{g}$ is only dependent on the forcing function, i.e., the measurement values. In the problems considered in this paper the constraint values do not change from one measurement to the next. Consequently, $\boldsymbol{y}_h$ can be computed a-priori and made available as a vector of constraints for the run-time computation.

6.2. Run-Time Computation

Both M and $\boldsymbol{y}_h$ are computed a-priori. A standard PC with higher precision arithmetic can be used for these computations. In this manner, the final errors in M and $\boldsymbol{y}_h$ are dominated by the rounding effects of converting the double precision values to single precision for the embedded computation, should the embedded system not support double precision arithmetic. Substituting M and $\boldsymbol{y}_h$ into Eqn. (23) yields,

$$\boxed{\boldsymbol{y} = \mathsf{M}\,\boldsymbol{g} + \boldsymbol{y}_h.} \tag{27}$$

Only the vector of sensor data $\boldsymbol{g}$ changes with each measurement. Consequently, the run-time solution of the inverse problem is reduced to a single matrix multiplication and a vector addition. This makes the repeated computation of the solution very efficient. Given n measurement values, the computational cost $W(n)$ is,

$$W(n) = 2n^2. \tag{28}$$

For example, a sensor chain with $n = 21$ inclinometers would require $W(n) = 882$ FLOPs to solve the inverse problem. The computation effort reduces to

$$W(n) = n^2. \tag{29}$$

if the processor architecture being used supports a multiply-accumulate[5] operation. Both the exact number of FLOPs and memory required are known prior to the run-time computation. This enables the computation of a strict upper bound for the execution time of the equation. Consequently, the method is, by definition, suitable for real-time

[5] See for example the specifications for the ARM Cortex Microcontroller Software Interface Standard (CMSIS) at http://www.arm.com/products/processors/cortex-m/.

applications. The computational complexity, $\mathcal{O}(n^2)$, is independent of the placement of the nodes, the equation being solved and the support length selected.

6.3. Error Estimation and Confidence Interval

There is uncertainty associated with the solution of any inverse problem. Regularization is used to control this uncertainty. The aim now is to quantify the uncertainty associated with the solution presented here. The are two primary sources of possible errors involved in the computation of $\boldsymbol{y}$ in Eqn. (27):

(1) Errors in the values contained in M and $\boldsymbol{y}_h$. The numerical testing (see Section 7) demonstrates that these errors are negligible, when computed in double precision, in comparison to realistic perturbations of $\boldsymbol{g}$. For some embedded systems it is necessary to reduce the values from double to single precision. Software-in-the-loop (SIL) and processor-in-the-loop (PIL) testing are used to quantify these errors. In Section 9 it is experimentally verified that these errors can be ignored.

(2) The errors at run-time are dominated by the perturbations of $\boldsymbol{g}$, these errors are orders of magnitude larger than the residual numerical errors in M and $\boldsymbol{y}_h$. Consequently, only errors in $\boldsymbol{g}$ are considered for the covariance propagation. There is also an approximation error in M based on the choice of the interpolating functions. These may not be insignificant depending on the nature of the solution $\boldsymbol{y}$.

The following computation assumes that only the forcing function $\boldsymbol{g}$ is subject to Gaussian perturbation. The covariance $\Lambda_{\boldsymbol{y}}$ associated with the computation of $\boldsymbol{y}$ using Eqn. (27), can be explicitly [Brandt 1998] calculated as

$$\Lambda_{\boldsymbol{y}} = \mathsf{M}\,\Lambda_{\boldsymbol{g}}\,\mathsf{M}^{\mathrm{T}}, \tag{30}$$

where $\Lambda_{\boldsymbol{g}}$ is the covariance of the forcing function. In practical applications we determine the magnitude of the noise component for each sensor, using dedicated noise measurements. In this case, and assuming that the noise is independent identically distributed (i.i.d.) Gaussian noise with standard deviation $\sigma_{\boldsymbol{g}}$, then,

$$\Lambda_{\boldsymbol{g}} = \sigma_{\boldsymbol{g}}^2\,\mathsf{I}, \tag{31}$$

where $\sigma_{\boldsymbol{g}}$ is a measured value. Substituting this into Eqn. (30) yields,

$$\Lambda_{\boldsymbol{y}} = \sigma_{\boldsymbol{g}}^2\,\mathsf{M}\,\mathsf{M}^{\mathrm{T}}. \tag{32}$$

An upper-bound estimate within a given confidence interval for the vector of standard deviations for $\boldsymbol{y}$, is computed as,

$$\boxed{\boldsymbol{\sigma}_{\boldsymbol{y}} = \sigma_{\boldsymbol{g}}\,\boldsymbol{s},} \tag{33}$$

where $\mathsf{M} = (m_{ij})$ and the individual elements of the unscaled standard deviation $\boldsymbol{s}$ are $s_i = (\sum_{j=1}^{n} m_{ij}^2)^{1/2}$, i.e. the square root of the diagonal elements of $(\mathsf{M}\,\mathsf{M}^{\mathrm{T}})$. This term can be computed a-priori; consequently, the run-time computational complexity for determining the standard deviation of each solution point is $\mathcal{O}(n)$.

Alternatively, the error vector $\boldsymbol{\epsilon}$ can be computed for each measurement as the difference between the forward and inverse problem, i.e.,

$$\boldsymbol{\epsilon} = \boldsymbol{g} - \mathsf{L}\,(\mathsf{M}\,\boldsymbol{g} + \boldsymbol{y}_h)\,. \tag{34}$$

A Kolmogorov-Smirnov test can be applied to $\boldsymbol{\epsilon}$ to determine if it is Gaussian. This yields additional information on the suitability of the model for the specific measurement. Given the standard deviation, the confidence interval with a specific degree of certainty is computable via the inverse Student-t distribution [Brandt 1998].

7. MODEL-IN-THE-LOOP TESTING

The aim of this section is to verify the numerical accuracy and efficiency of the new method on a PC, embedded testing is presented later. We have chosen to solve unperturbed problems for the first tests, since these enable the comparison with analytical solutions and with standard engineering approaches such as Runge-Kutta methods[6]. The unperturbed tests enable the separation of the errors involved in computing M and $\boldsymbol{y}_h$ from those resulting from the perturbation of $\boldsymbol{g}$. It is difficult to define a truly objective method of comparing solution approaches which are fundamentally different[7]. Each approach has its own weaknesses and strengths. The tests have been devised to reflect, as close as possible, the conditions which are to be expected from the application specific method.

7.1. Test A: Initial Value Problem 1

The ODE (details can be found in [Adams 2006]) is a third order ($m = 3$) non-homogeneous ODE with constant coefficients a_i and $p = 3$ constraints. The equation is

$$\begin{aligned} & y^{(3)} + 3\,y'' + 3y' + y = 30\,\mathrm{e}^{-x} \quad \text{given} \\ & y(0) = 3, \quad y'(0) = -3, \quad y''(0) = -47 \end{aligned} \tag{35}$$

in the interval $0 \leq x \leq 8$. The analytical solution to this equation is

$$y(x) = (3 - 25\,x^2 + 5\,x^3)\,\mathrm{e}^{-x}. \tag{36}$$

In the case of the inverse problems being addressed, the position of the solution points is determined by the measurement. To simulate this condition, the `ode45` solver [Shampine and Reichelt 1997] in MATLAB has been used to solve this differential equation. This is a variable step size method which yields both a vector of abscissae $\boldsymbol{x}$ consisting of $n = 77$ points and the solution vector $\boldsymbol{y}$. Exactly those $n = 77$ points on the abscissae and a support length $l_s = 9$ was used for the test of the new method. Using the `ode45` solver in addition to the analytical solution enables the comparison of the new method with well established techniques. The results of the three computations are shown in Fig. 3. The residual errors, i.e., the difference between the analytical solution, the new method and the `ode45` solutions are shown in Fig. 4. The 2-norm of the residual errors $|\boldsymbol{\epsilon}|_2$ and the computation time for $k = 10000$ iterations[8] for the solution of the ODE are given in Table I for the `ode45` method as well as the new method. The first observation is that the residual numerical errors for the new method

Table I: The 2-norm of the residual errors $|\boldsymbol{\epsilon}|_2$ and the computation time for $k = 10000$ iterations for the solution of the ODE, for the `ode45` method and new method (New). These computations were performed with an Intel Core 2 Duo CPU P8600 at 2.4 [GHz] with 2.9 [GB] RAM.

method	$\lvert\boldsymbol{\epsilon}\rvert_2$	time ($k = 10000$)
`ode45`	$1.79\,10^{-3}$	29.823728 [s]
New	$1.14\,10^{-7}$	0.061681 [s]

[6] The MATLAB `ode45` implementation of a Runge-Kutta method was used for this purpose.

[7] To support independent verification of our results, we have made the MATLAB code available which we used to generate all the results presented in this section, see http://www.mathworks.com/matlabcentral/fileexchange/45947.

[8] It is not the absolute times which are important, since they will change from one platform to another. It is the relative speed which shows the potential performance of the new method.

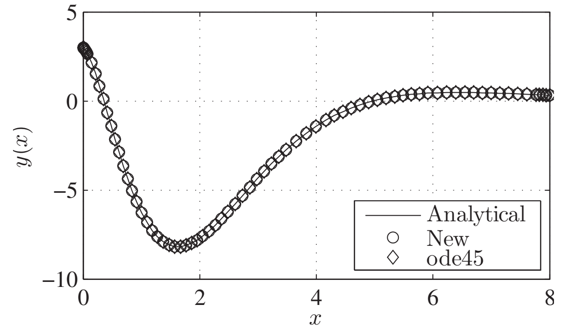


Fig. 3: The plot shows the analytic solution from Eqn. (36), the solution with the new method with $l_s = 9$ as well as the solution by a Runge-Kutta `ode45` method. All solutions are evaluated at exactly the $n = 77$ points provided by the `ode45` method.

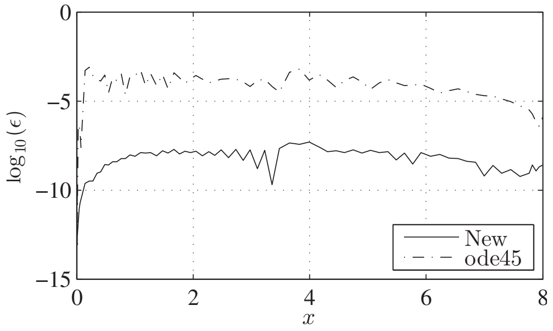


Fig. 4: Residual errors: difference of the new numerical solution vs. the analytical solution and the solution by a Runge-Kutta `ode45` method vs. the analytical solution. The new method is approximately 4-orders of magnitude more accurate than the `ode45` method, for exactly the same abscissae.

are approximately 4-orders of magnitude smaller than with the `ode45` method and the computation is almost 500 times faster. Reducing the error bound for the Runge-Kutta method will improve the numerical accuracy, but at the expense of computational effort. In order to reduce the error by 4-orders of magnitude in the Runge-Kutta solution, an even higher degree Runge-Kutta method must be formulated, which in turn would require unreasonably high computational effort. The very small errors are significant: since, when computing the confidence interval for the solution, they can be neglected when they are small in comparison with the perturbations of the forcing function.

The comparison of speed is somewhat subjective, since we have no insights into how much function-call-overhead is involved in the MATLAB implementation; nevertheless, it does show the potential speed of the new approach. This test demonstrates the ability of the new method to compute solutions to the ODE at arbitrary given nodes with a very high accuracy.

7.2. Test B: Alternative Node Placement for Initial Value Problem 1

In this test the same ODE is solved as in Test A, however, a reduced number of $n = 20$ evenly spaced nodes has been selected for the new method since systems sampling in time in general use even spacing. The results are shown in Fig. 5 and 6. The new method achieves the same solution quality, in terms of accuracy, as the `ode45` method, however with a significantly reduced number of nodes. This corresponds to an accurate solution of the inverse problem with a small number of sensors.

7.3. Test C: Initial Value Problem 2

The second example is a second order ($m = 2$) ODE with variable coefficients $a_i(x)$ and $p = 2$ constraints. This demonstrates the ability of the method to deal with variable coefficients and with solutions which are irrational functions. The equation is,

$$\begin{aligned} &2\,x^2\,y'' - x\,y' - 2y = 0 \quad \text{given} \\ &y(1) = 5, \qquad y'(1) = 0 \end{aligned} \tag{37}$$

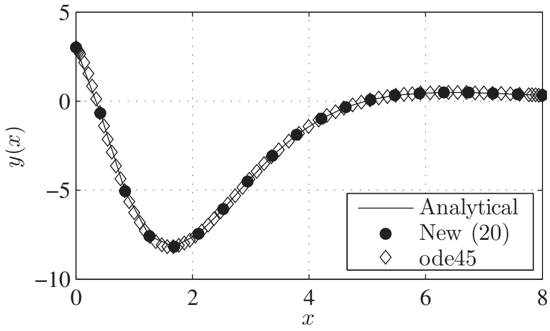


Fig. 5: The plot shows the analytic solution from Eqn. (36), the solution with the new method with $l_s = 9$ and $n = 20$ as well as the solution by a Runge-Kutta ode45 method with $n = 77$.

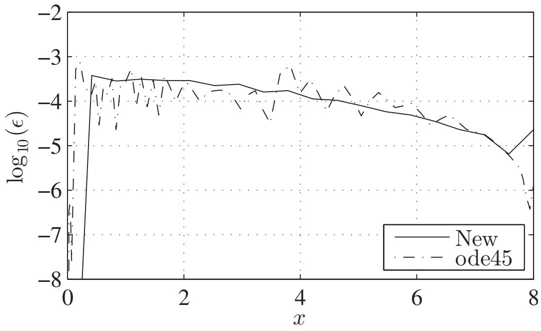


Fig. 6: Residual errors: difference of the new numerical solution with $n = 20$ vs. the analytical solution and the solution by a Runge-Kutta ode45 method with $n = 77$ vs. the analytical solution.

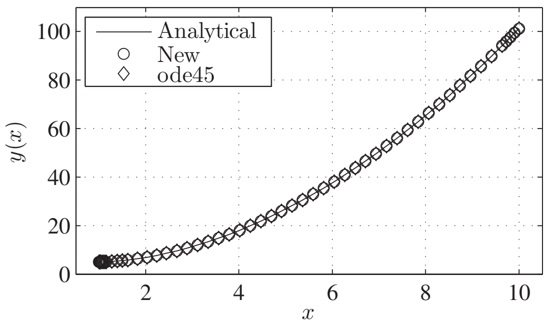


Fig. 7: The plot shows the analytic solution from Eqn. (38), the solution with the new method with $l_s = 15$ and $n = 69$ evenly placed nodes as well as the solution by a Runge-Kutta ode45 method with $n = 69$ variably placed nodes.

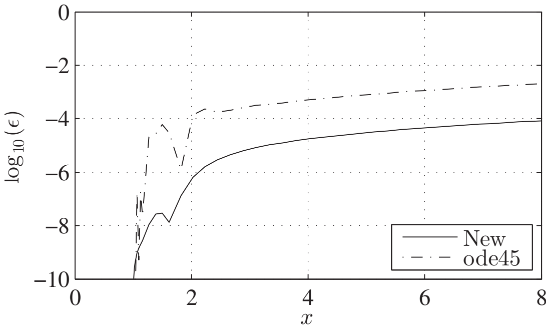


Fig. 8: Residual errors: difference of the new numerical solution vs. the analytical solution and the solution by a Runge-Kutta ode45 method vs. the analytical solution.

in the interval $1 \leq x \leq 10$. The analytical solution to this equations is

$$y(x) = x^2 + \frac{4}{\sqrt{x}}. \tag{38}$$

The solution's appearance would not suggest that this is a demanding problem. However, the analytical solution is the sum of a polynomial and an irrational function. Computing good estimates for the derivatives of such functions can require a high degree of polynomial approximation. The solution obtained using the new method, the analytical solution and the result of the ode45 solver are shown in Fig. 7. The high density of nodes at the start of the interval produced by the ode45 method indicates that the method required disproportionately many steps for finding a solution with sufficient accuracy. The new method is once again more accurate than the ode45 solver.

7.4. Test D: Selecting a Support Length for Initial Value Problem 2

In this test, the same ODE is solved as in Test C. As pointed out in Section 5, there is an issue in selecting the support length l_s (or degree) of the local approximation for the differentiating matrix D. This matrix and the linear differential operator L have been implemented for all odd support lengths in the range $3 \le l_s \le 25$ and the IVP was solved for each of these implementations. The relative error was computed for the corresponding solutions as

$$\epsilon(l_s) = \frac{|\boldsymbol{y}_a - \boldsymbol{y}|_2}{|\boldsymbol{y}_a|_2}, \tag{39}$$

where $\boldsymbol{y}_a$ is the sampled analytical solution and $\boldsymbol{y}$ is the solution computed with the new method. The $\log_{10}(\epsilon)$ vs. l_s is shown in Fig. 9. This result shows that there is a minimum in the relative error for $l_s = 15$, indicating that there is a justification for implementing local approximation to derivatives for specific problems with high numbers of nodes[9]. The dependence of ϵ on l_s is a function of the equation being solved. There will be no solution that is optimal for all cases. With the proposed method the necessary l_s is determined during the preparatory computations and not at run-time.

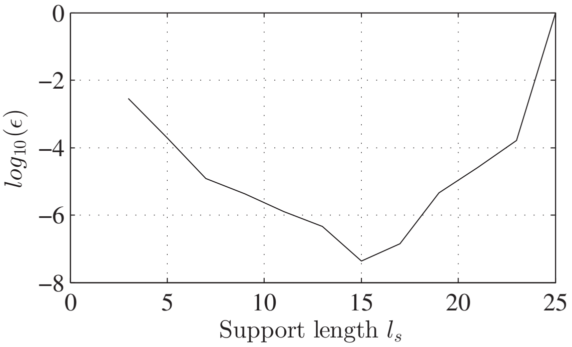


Fig. 9: The relative error ϵ has a minimum for $l_s = 15$.

7.5. Test E: Inverse 3-Point Boundary Value Problem

The following test is an over constrained first order ($m = 1$) 3-point inverse BVP[10]; however, with $p = 4$ constraints at 3 locations on the abscissae. It belongs to the class of inverse multi-point BVPs[11]. The constraints implemented are both: homogeneous and non-homogeneous; as well as Dirichlet and Neumann boundary conditions. This example has been chosen to demonstrate the numerical efficiency and behavior of the method with respect to a perturbed inverse BVP. Furthermore, it demonstrates the ability of the algebraic framework to deal with generalized constraints. Synthetic data is produced for a function and its analytic derivatives, in this manner the result of the reconstruction can be compared with the function from which the data was derived.

The problem being considered is to reconstruct a curve $\boldsymbol{y}$ from multiple local measurements of the curve's gradients $\boldsymbol{g}$ while fulfilling a set of constraints $\mathsf{C}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{d}$, which are not restricted to the ends of the support, i.e., inner constraints are also present.

[9] Many books [Burden and Faires 2005; Lapidus and Pinder 1999; Strikwerda 2004] discuss the possibility of implementing approximations of higher degree; however, they never actually show comparative numerical results for practical problems.

[10] At least one of the constraints is interior, since the constraints are not restricted to the boundaries.

[11] Although we have been able to find a number of publications on methods relating to multi-point BVPs [Welsh and Ojika 1980; Agarwal et al. 2003], there is very little literature available on inverse multi-point BVPs, e.g. [Kurylev 1993]. There are no general approaches available at the present time.

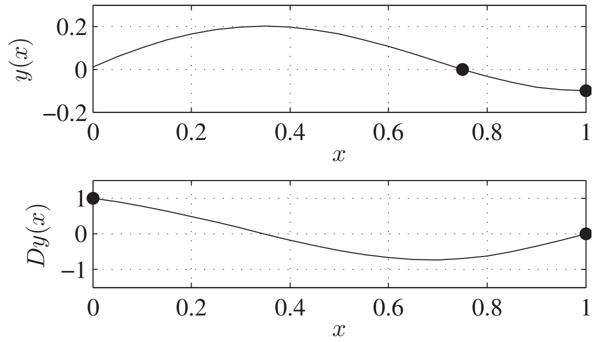


Fig. 10: First order ($m = 1$) constrained test function and its analytical derivatives: this function fulfills all $p = 4$ constraints specified in Eqn. (41) and (42). The homogeneous and non-homogeneous Dirichlet and Neumann constraints are marked at the three locations $[0, 0.7895, 1]$ in the top and bottom plots respectively.

Furthermore, the measurements are perturbed by white noise. The ODE is

$$y'(x) = g(x) + \epsilon, \tag{40}$$

where ϵ is the error caused by the forcing function's perturbation. This equation is subject to the homogeneous and non-homogeneous Dirichlet boundary conditions,

$$y(0.7895) = 0 \quad \text{and} \quad y(1) = -0.1 \tag{41}$$

as well as the non-homogeneous and homogeneous Neumann boundary conditions,

$$y'(0) = 1 \quad \text{and} \quad y'(1) = 0. \tag{42}$$

A synthetic test function which fulfills these conditions was generated by combining an arbitrary polynomial of 4^{th} degree,

$$y(x) = 1.1x^4 + 0.4x^3 + 0.5x^2 - 1.2x - 0.3, \tag{43}$$

with the four constraints, this yields an 8^{th} degree polynomial[12] which also fulfills the constraints:

$$\begin{aligned} y_c(x) = &- 0.46985x^8 + 0.41127x^7 + 0.34891x^6 + 0.03827x^5 \\ &+ 1.0323x^4 - 1.5886x^3 - 0.88426x^2 + x + 0.011895. \end{aligned} \tag{44}$$

The first derivative of Eqn. (44) can be computed analytically, making it a suitable test function for constrained curve reconstruction from gradients. The function $g(x)$, its analytical gradient $g'(x)$ and the constraints are shown in Fig. 10.

The analytical gradient is evaluated at $n = 21$ points[13]. Then in a Monte Carlo simulation, with $k = 10000$ iterations, the gradients are perturbed by artificial Gaussian noise with a standard deviation of 1% of the maximum value of $Dy(x)$. For each simulation, a reconstruction is performed with the appropriate M and $\boldsymbol{y}_h$ and the statistics are computed. The result of solving this problem using the proposed method is shown in Fig. 11, together with the error bars corresponding to the standard deviation of the reconstructed values observed in the Monte Carlo simulation. The error bars have been magnified by a factor of 10 to increase the visibility.

In Fig. 12, the bias of the reconstruction is shown, i.e., the difference between the analytical solution and the mean of the Monte Carlo simulations. Additionally, the

[12] The theory behind this computation can be found in [Harker and O'Leary 2013d].

[13] For example, these would correspond to the positions of the inclinometers on a structure being monitored.

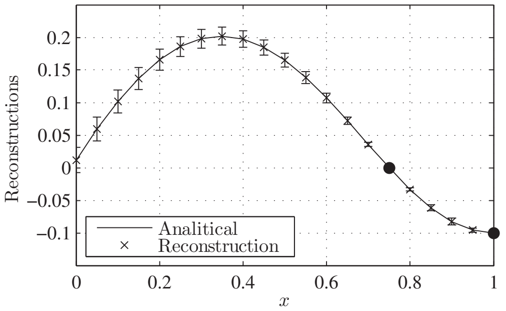


Fig. 11: Comparison of the analytical solution and the result of the reconstruction from the perturbed gradients. The two Dirichlet boundary conditions are marked on the reconstruction, the two Neumann conditions are not shown. The error bars correspond to the standard deviations, i.e., an estimate for the 68.3% confidence interval. They are obtained from a Monte Carlo simulation with $k = 10000$ iterations. The error bars have been magnified by a factor of 10 to increase the visibility.

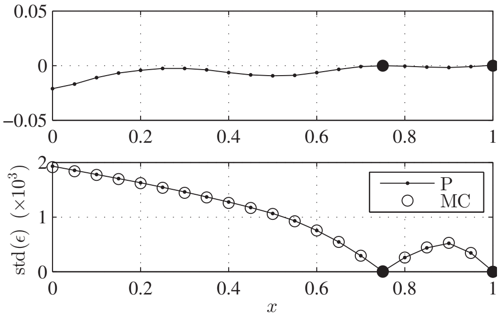


Fig. 12: Top: Bias of the reconstruction, i.e., the difference between the mean reconstruction from the $k = 10000$ Monte Carlo iterations and the analytical solution (the result is scaled by 10^5 to make the error visible.). Bottom: The standard deviation of the reconstruction as predicted (P) by the covariance propagation according to Eqn. (33) and the results from the Monte Carlo simulations (MC) (the result is scaled by 10^3).

standard deviation of the result as predicted by Eqn. (33) and the results of the Monte Carlo simulation are compared. These results verify that the solution is, for all intents and purposes, bias free and the predicted uncertainty is correct. The method has been successfully applied to an inverse three-point boundary value problem, with two Dirichlet and two Neumann boundary conditions. Not only is the problem solved, but in addition the uncertainty of the solution is delivered by the new method.

7.6. Summary of the Numerical Testing

The implications of the above test can be summarized as follows:

(1) The new method is capable of solving both perturbed and unperturbed inverse problems, including initial-, inner- and boundary value problems.
(2) The method enables the formulation and solution of problems with constraints on arbitrary derivatives of the solution function. Consequently, both Dirichlet and Neumann boundary conditions can be dealt with. The constraints can be both homogeneous ($d = 0$) and non-homogeneous ($d \neq 0$).
(3) The method exhibits significantly smaller numerical errors than the Runge-Kutta `ode45` approach, while being significantly faster. Reducing the error bound for the `ode45` will improve the numerical accuracy, but at the expense of computational effort. The numerical errors are so small that they can be neglected when solving inverse problems where the perturbation of the forcing function is significant.
(4) The separation of the solution into a preparatory (offline) and run-time (online) computation makes the method suitable for embedding in real-time systems.

8. AUTOMATIC CODE GENERATION

The aim of the code generation is, given the definition of specific inverse problem in terms of a measurement model (see Section 3), to automatically generate the code required to solve the problem on an embedded computing system. That is, the problem is defined as a symbolic definition of a differential equation together with a suitable set of constraints and a source of data. The system will then solve any ODE, regardless of its nature (IVP, BVP, etc.) from this specification. A similar concept of automatic code generation (ACG) for the embedding of convex optimization was explained in literature [Mattingley and Boyd 2012]. Furthermore, it was shown, that numerical ODE solvers can be deployed on FPGAs using VHDL in [Huang et al. 2013]. During MBD, the system is designed on an abstract model level based on the system's requirements while ensuring the consistency of the system's physical representation. ACG is the task of converting the models and their algorithms to usable code, effectively automating the time-consuming and error-prone process of low-level programming. Basically, there are three low-level target languages for embedded systems: multi-purpose *ANSI-C code*, which is the focus of this article; *hardware description language (HDL)* for field programmable gate arrays (FPGA) or application specific integrated circuits (ASIC); and IEC 61131-3 compliant languages such as *structured text (ST)* for programmable logic controllers (PLC).

Code deployment is the integration of code on the embedded systems. In most cases, the architecture of the development (host) system (x86 or x64) is largely different from the embedded (targeted) system (ARM, ATmel, etc.). There are two approaches for solving this issue *compilation of code on the target system* if an OS and an appropriate compiler is present; or *cross compilation* on the host system via processor virtualization, a popular tool to perform this task is the LLVM compiler infrastructure.

After MBD is complete, so called production code is generated. The process strips out all parameters needed during testing and optimizes the code for performance (low memory usage, high computational speed) or safety (data consistency, robust algorithms).

8.1. Embedded Target Hardware

The goal is to show that even highly abstract and complex mathematical models are deployable on the simplest embedded hardware, demonstrating the scalability of the method. The open-source *Raspberry Pi* or the proprietary *BeagleBone Black* are popular entry-level embedded systems for target programming. The *WAGO PFC-200* is an IEC 61131-3 compliant industrial PLC with open source software. These three systems are based on 32-bit ARM processors and they run an embedded Linux derivative as operating system (OS); in the case of the WAGO device it's a real-time OS. The automated resource management is the main advantage of embedded systems with an OS. The low end of the systems is represented by the fully open source *Arduino Uno* platform. The utilized Atmel 8-bit AVR RISC-based ATmega328 microcontroller has no dedicated OS, the program logic is directly stored on the chip's 32 [kB] flash memory as firmware. The Arduino Uno has been chosen for experimental PIL testing, see Section 9. A laboratory experiment featuring the BeagleBone Black is presented in Section 10.

8.2. MBD Software for Code Generation

Most engineering and scientific software for designing mathematical models has the functionality to automatically generate standard ANSI-C code from its application-specific syntax, e.g., LabVIEW, Maple or Mathematica. This is usually necessary, because most industrial controllers are only programmable with C. A short survey on tools for ACG has been given by Rafique et al. [Rafique et al. 2013]. This article is focused on the usage of MATLAB and its Coder toolbox, because it is the standard software for mathematical MBD. Code generation fully supports linear algebra. Nevertheless, the presented approach is so simple, that a C code parser could be implemented manually without much effort. Two test cases confirmed the correct functionality of algebraic functions generated by MATLAB Coder: C code was generated for QR decomposition and singular value decomposition. Compiling the code with Microsoft Visual Studio 2010 for the 32-bit host development system delivered an executable program, which successfully validated the code via SIL verification. Deploying the same code on the Arduino Uno confirmed the correct functionality via PIL verification.

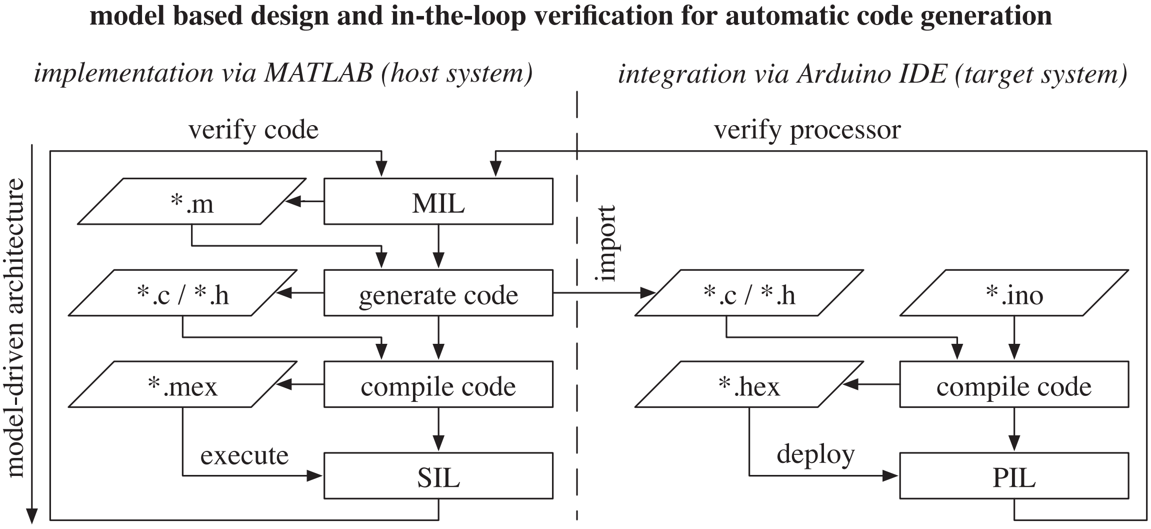

model based design and in-the-loop verification for automatic code generation
implementation via MATLAB (host system)
integration via Arduino IDE (target system)
verify code
verify processor
model-driven architecture
*.m
MIL
*.c / *.h
generate code
*.mex
compile code
execute
SIL
import
*.c / *.h
*.ino
*.hex
compile code
deploy
PIL

Fig. 13: MBD is an iterative approach. Each step requires verification to ensure that the (sub)system's requirements are met. In this article, model-in-the-loop (MIL) verifies the correctness of the algebraic framework on model level, software-in-the-loop (SIL) verifies the functional equivalence of the generated C code on code level and processor-in-the-loop (PIL) verifies the correct computation on the employed microcontroller on binary level. This graphic shows the process for the Arduino platform.

8.3. Targeting and Verification Process

The ACG process is completely general, the illustration in Fig. 13 shows the procedure for the Arduino platform. Following steps must be carried out:

Model-in-the-loop (MIL): The system is identified, designed and simulated on abstract model level in an artificial environment, producing MATLAB model code (*.m). This is an efficient way to estimate model parameters with varying configurations. This includes the determination of the optimal support length l_s, computation of the constrained linear differential operator M and the homogeneous solution $\boldsymbol{y}_h$. Furthermore, changes in the requirements can easily be implemented in this early design stage.

Code generation: The MATLAB Coder toolbox is a sophisticated parser engine. It converts the model code (*.m) into C code (*.c) and the associated header files (*.h).

Code compilation for SIL: MATLAB Coder features the ability to replace model function calls with calls for MATLAB executables (*.mex). Such functions are wrappers around compiled C code, which can be directly called from the MATLAB development environment.

Software-in-the-loop (SIL): The model and the generated C code must be functionally equivalent, i.e., a certain input must deliver the same output on all abstraction layers. This is especially relevant when the target language misses certain features of the model language, e.g. shorter bit-lenghts of variable types or no support for floating point operations. The consistency of the model must be ensured on all levels.

Code compilation for PIL: The C code (*.c/*.h) is imported into the Arduino IDE. The header (*.h) files must be included in the Arduino project's main (*.ino) file. The C code is cross-compiled for the Arduino platform delivering a (*.hex) file, which is stored directly on the ATmega328's flash memory as firmware.

Processor-in-the-loop (PIL): The code runs on the embedded real-time system. The outcome is not necessarily the same as during simulation, because the hardware platform used during MIL and SIL is different from the PIL target.

9. SOFTWARE- AND PROCESSOR-IN-THE-LOOP TESTING

In Section 7, the viability of the new method was shown during MIL. In this section, the test cases are directly executed on the Arduino Uno for PIL verification. The microcontroller features $2\,[\mathrm{kB}]$ SRAM and a processing power of 16 million instructions per second (MIPS). The 23 general purpose I/O lines, the 6-channel 10-bit A/D converter and the operating voltage of $1.8 - 5.5\,[\mathrm{V}]$ makes it a well suited setup for acquiring and processing sensor data. The problem size must be scaled down in order to fit the Arduino Uno's limited system resources. An Arduino Uno double variable requires $4\,[\mathrm{B}]$ of memory, so theoretically, the ATmega328 chip stores up to 512 double variables in its memory of $2\,[\mathrm{kB}]$. Obviously, operations and other variables also require memory space, therefore the problem size has been shrunk to 10 input signals. This corresponds in means of problem size to the Arduino Uno's 6 analog I/O ports, which are usable to connect sensors to the device. However, the problem classes are still the same.

The constrained linear differential operator M has then a size of (10×10) and the homogeneous solution vector $\boldsymbol{y}_h$ has a size of (10×1). Both, M and $\boldsymbol{y}_h$, are computed a-priori during offline calibration. Only the measurement vector $\boldsymbol{g}$ with size (10×1) changes its values from one measurement to the next. Consequently, the result of the online computation, i.e. the solution vector $\boldsymbol{y}$, is of size (10×1).

9.1. Initial Value Problem 1

The test case in Section 7.1 has been modified to have $n = 10$ evenly spaced nodes in the interval $0 \leq x \leq 0.1$ with a support length of $l_s = 5$. The computation time on the Arduino Uno is $t = 1.788\,[\mathrm{ms}]$, i.e., a sample rate of $> 500\,[\mathrm{Hz}]$ is possible. The error plots of the numerical computations are shown in Fig. 14 and 15.

9.2. Initial Value Problem 2

The test case in Section 7.3 has been modified to have $n = 10$ evenly spaced nodes in the interval $1 \leq x \leq 2$ with a support length of $l_s = 5$. The computation time on the Arduino Uno is $t = 1.228\,[\mathrm{ms}]$, i.e., a sample rate of $> 800\,[\mathrm{Hz}]$ is possible. The error plots of the numerical computations are shown in Fig. 16 and 17

9.3. Inverse 3-Point Boundary Value Problem

The test case in Section 7.5 has been modified to have $n = 10$ evenly spaced nodes in the interval $0 \leq x \leq 0.1$ with a support length of $l_s = 5$. New constraints have been defined to conserve the test case's characteristics:

$$y(0.0556) = 0, \qquad y(0.1) = -0.1, \tag{45}$$

$$D\,y(0) = 1, \qquad D\,y(0.1) = 0. \tag{46}$$

The computation time on the Arduino Uno is $1.796\,[\mathrm{ms}]$, i.e., a sample rate of $> 550\,[\mathrm{Hz}]$ is possible. The error plots of the numerical computations are shown in Fig. 18 and 19. In contrast to the previous test cases, here the SIL verification delivered an error

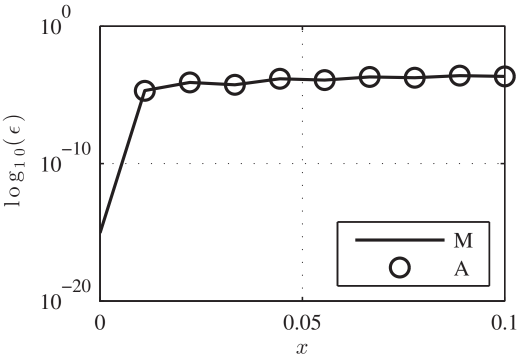


Fig. 14: The plot shows the error between the analytic solution and MATLAB's solution (M), the error norm is $|\epsilon|_2 = 4.6771\,10^{-4}$, as well as the error between the analytic solution and Arduino's solution (A), the error norm is $|\epsilon|_2 = 4.6749\,10^{-4}$.

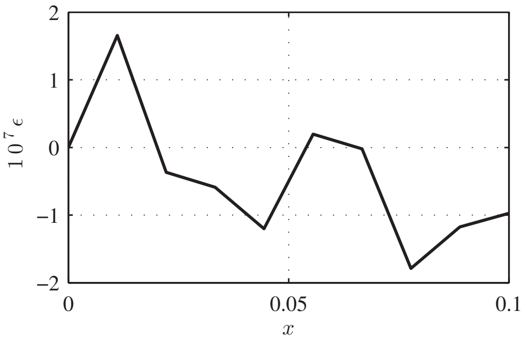


Fig. 15: PIL verification: the difference between MATLAB's solution and Arduino's solution is shown, the error norm is $|\epsilon|_2 = 3.1999\,10^{-7}$. The result is scaled by 10^7 to increase the visibility.

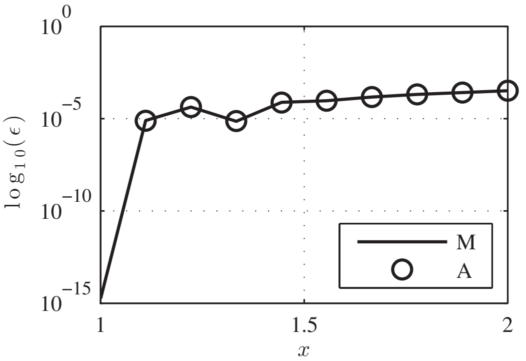


Fig. 16: The plot shows the error between the analytic solution and MATLAB's solution (M), the error norm is $|\epsilon|_2 = 4.9541\,10^{-4}$, as well as the error between the analytic solution and Arduino's solution (A), the error norm is $|\epsilon|_2 = 4.9568\,10^{-4}$.

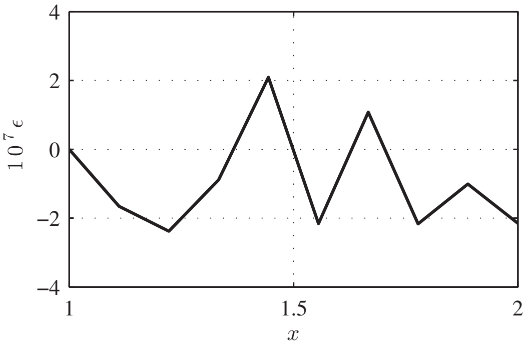


Fig. 17: PIL verification: the difference between MATLAB's solution and Arduino's solution is shown, the error norm is $|\epsilon|_2 = 5.4520\,10^{-7}$. The result is scaled by 10^7 to increase the visibility.

vector with norm of $|\epsilon|_2 = 3.3866\,10^{-14}$, i.e., the C code's result is slightly different than the MATLAB model code's result.

10. LABORATORY TESTING

The introduced algebraic model is tested on a laboratory setup, see Fig. 20. A chain of equally spaced one-dimensional inclinometers is mounted on a $b = 1.8\,[\mathrm{m}]$ long flexible structure. The arrangement consists of 14 sensors with an additional 2 screw clamps, effectively forcing 2 pairs, i.e. $p = 4$, of homogeneous Dirichlet and Neumann constraints at the structure's ends. These leads to a total of $n = 16$ points for the computation, i.e., the vector of measurement data $\boldsymbol{g}$ is of size (16×1). In order to vary the structure's bending, a square metal profile with feed size $h = 20\,[\mathrm{mm}]$ is placed

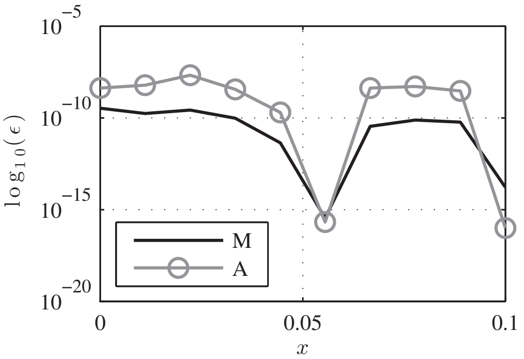


Fig. 18: The plot shows the error between the analytic solution and MATLAB's solution (M), the error norm is $|\boldsymbol{\epsilon}|_2 = 4.7958\,10^{-10}$, as well as the error between the analytic solution and Arduino's solution (A), the error norm is $|\boldsymbol{\epsilon}|_2 = 2.3902\,10^{-8}$.

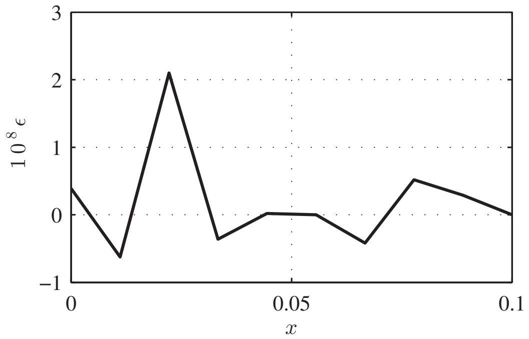


Fig. 19: PIL verification: the difference between MATLAB's solution and Arduino's solution is shown, the error norm is $|\boldsymbol{\epsilon}|_2 = 2.3673\,10^{-8}$. The result is scaled by 10^8 to increase the visibility.

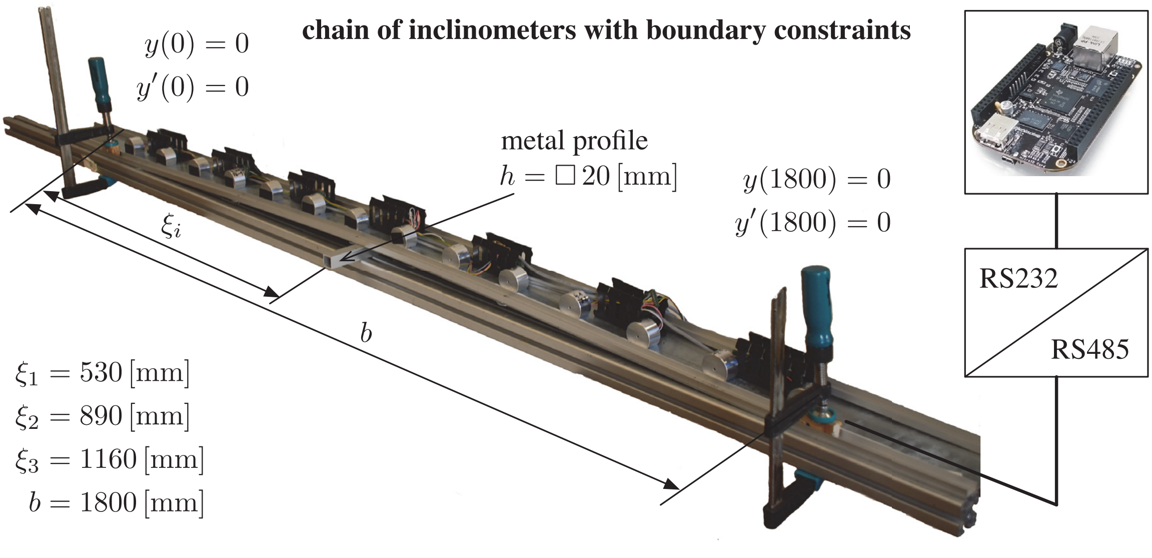


Fig. 20: The illustration shows the chain of inclinometers mounted on a flexible structure. Each of the 14 sensors is connected to an industrial RS-485 bus. This bus is converted to a RS-232 serial interface, which enables the connection of the BeagleBone Black. The 2 screw clamps force the homogeneous boundary values at the structure's ends.

between the structure and the supporting mounting platforms at the ξ_i positions. The results for these tests are shown in Fig. 21 to 26. Note, that the reference data has been acquired with calipers and hardly represents the true value; however, it is a good basis for comparisons. The constrained linear differential operator M is of size (16×16). M and the homogeneous solution y_h are computed in a preparatory step. The online computation is carried out by a BeagleBone Black. The hardware features a RISC processor based on the ARMv7 Cortex A8 platform with 1 [GHz] (2000 MIPS) and 512 [MB]

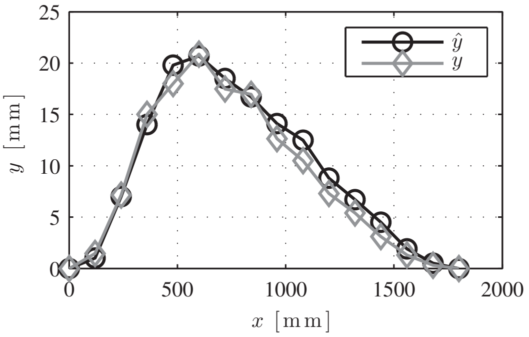


Fig. 21: The reference values $\hat{\boldsymbol{y}}$ and the computed curve $\boldsymbol{y}$ for the metal profile placed at $\xi_1 = 530$ [mm] are shown.

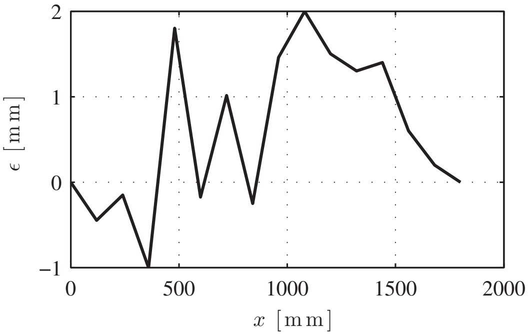


Fig. 22: The difference between both values $\boldsymbol{\epsilon} = \hat{\boldsymbol{y}} - \boldsymbol{y}$ is shown, the error norm is $|\boldsymbol{\epsilon}|_2 = 4.245$ [mm].

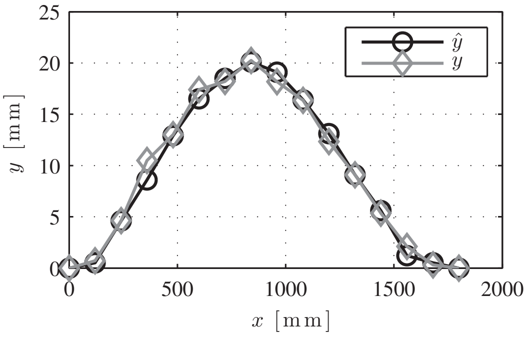


Fig. 23: The reference values $\hat{\boldsymbol{y}}$ and the computed curve $\boldsymbol{y}$ for the metal profile placed at $\xi_2 = 890$ [mm] are shown.

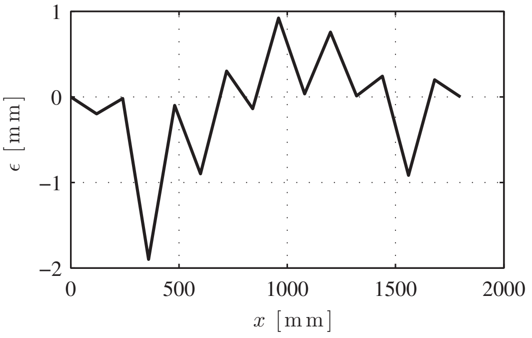


Fig. 24: The difference between both values $\boldsymbol{\epsilon} = \hat{\boldsymbol{y}} - \boldsymbol{y}$ is shown, the error norm is $|\boldsymbol{\epsilon}|_2 = 2.636$ [mm].

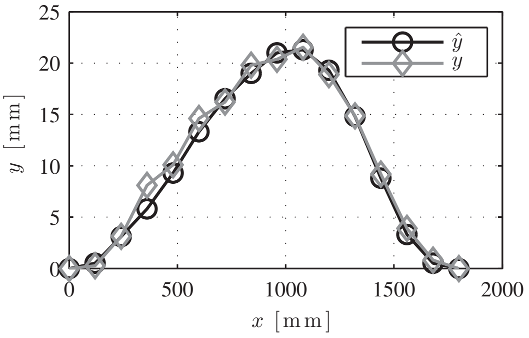


Fig. 25: The reference values $\hat{\boldsymbol{y}}$ and the computed curve $\boldsymbol{y}$ for the metal profile placed at $\xi_3 = 1160$ [mm] are shown.

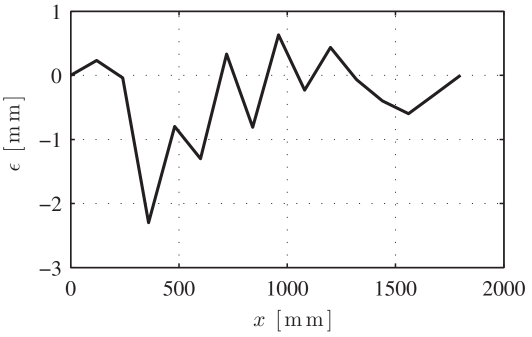


Fig. 26: The difference between both values $\boldsymbol{\epsilon} = \hat{\boldsymbol{y}} - \boldsymbol{y}$ is shown, the error norm is $|\boldsymbol{\epsilon}|_2 = 3.114$ [mm].

memory. The measurement data g is acquired via the built-in RS232 serial interface, the results y are transmitted to a centralized database. The WAGO PFC-200 would be an industrial-ready alternative hardware solution with similar architecture for this application. It features a variety of bus interfaces such as Modbus, Profibus and CAN bus.

11. CONCLUSION AND OUTLOOK

It can be concluded, from the numerical and experimental tests, that the newly proposed algebraic method outperforms previous solutions, both in accuracy and speed, for the class of problems being considered. The separation of the computation into an initial preparatory and a cyclic run-time portion yields a highly efficient numeric solution. The computation complexity of the explicit solution is only a function of the number of nodes (sensors) used. The automatic generation of C code, and the verification of its correct functionality on multiple embedded architectures has been demonstrated. The generation of C code also facilitates the use of the method in conjunction with commercial programmable logic controllers (PLCs), for the control of industrial plants and machinery. Here, the method was applied to a linear array of sensors. Presently, the tools are being extended to two-dimensional arrays and the resulting two-dimensional fields of data.

REFERENCES

R.A. Adams. 2006. *Calculus: Several Variables* (sixth ed.). Pearson Addison Wesley.

R.P Agarwal, H.B Thompson, and C.C Tisdell. 2003. Three-point boundary value problems for second-order discrete equations. *Computers and Mathematics with Applications* 45, 69 (2003), 1429 – 1435.

A. Ben-Israel and T.N.E. Greville. 2003. *Generalized Inverses: Theory and Applications*. Springer.

Siegmund Brandt. 1998. *Data Analysis* (third ed.). Springer, New York, NY, USA.

R.L. Burden and J.D. Faires. 2005. *Numerical Analysis*. Thomson Brooks/Cole.

O. Burdet and L. Zanella. 2002. Automatic Monitoring of the Riddes Bridges using Electronic Inclinometers. In *IABMAS, First International Conference on Bridge Maintenance, Safety and Management*.

T. A. Driscoll, F. Bornemann, and L. N. Trefethen. 2008. The chebop system for automatic solution of differential equations. *BIT* 48 (2008), 701–723.

Bengt Fornberg. 1998. *A practical guide to pseudospectral methods*. Vol. 1. Cambridge university press.

J. Golser. 2010. Fallbeispiel zur Bauwerksberwachung mittels online Neigungssensoren. In *25. Cristian Veder Kolloquium*.

G.H. Golub and C.F. Van Loan. 1996. *Matrix Computations* (third ed.). The Johns Hopkins University Press, Baltimore.

M. Harker and P. O'Leary. 06 May 2013 (Updated 11 Jul 2013)d. Polynomial Toolbox. (06 May 2013 (Updated 11 Jul 2013)). http://www.mathworks.com/matlabcentral/fileexchange/41658

M. Harker and P. O'Leary. 11 Apr 2013 (Updated 19 Aug 2013)b. Discrete Orthogonal Polynomial Toolbox. (11 Apr 2013 (Updated 19 Aug 2013)). http://www.mathworks.com/matlabcentral/fileexchange/41250

M. Harker and P. O'Leary. 18 Apr 2013 (Updated 17 Jun 2013)c. Ordinary Differential Equation Toolbox. (18 Apr 2013 (Updated 17 Jun 2013)). http://www.mathworks.com/matlabcentral/fileexchange/41354

M. Harker and P. O'Leary. 2013a. A Matrix Framework for the Solution of ODEs: Initial-, Boundary-, and Inner-Value Problems. *ArXiv e-prints* (April 2013).

Chen Huang, Frank Vahid, and Tony Givargis. 2013. Automatic Synthesis of Physical System Differential Equation Models to a Custom Network of General Processing Elements on FPGAs. *ACM Trans. Embed. Comput. Syst.* 13, 2, Article 23 (Sept. 2013), 27 pages. DOI: http://dx.doi.org/10.1145/2514641.2514650

Nathaniel Jewell. 29 Mar 2013 (Updated 01 Apr 2013). Collocation-based spectral-element toolbox. (29 Mar 2013 (Updated 01 Apr 2013)). http://www.mathworks.com/matlabcentral/fileexchange/41011

Yyldyray Keskyn, Onur Karaoglu, Sema Servy, and OturanÇ Galip. 2011. The Approximate Solution of High-Order Linear Fractional Differential Equations with Variable Coefficients in Terms of Generalized Taylor Polynomials. *Mathematical and Computational Applications* Vol. 16, No. 3 (2011), 617–629.

Nurcan Kurt and Mehmet Cevik. 2008. Polynomial solution of the single degree of freedom system by Taylor matrix method. *Mechanics Research Communications* 35, 8 (2008), 530 – 536.

Y.V. Kurylev. 1993. Multi-dimensional inverse boundary problems by BC-method: Groups of transformations and uniqueness results. *Mathematical and Computer Modelling* 18, 1 (1993), 33 – 45.

C. Lanczos. 1997. *Linear Differential Operators*. Dover Publications.

L. Lapidus and G.F. Pinder. 1999. *Numerical Solution of Partial Differential Equations in Science and Engineering*. Wiley.

Insup Lee, O. Sokolsky, Sanjian Chen, J. Hatcliff, Eunkyoung Jee, BaekGyu Kim, A. King, M. Mullen-Fortino, Soojin Park, A. Roederer, and K.K. Venkatasubramanian. 2012. Challenges and Research Directions in Medical Cyber Physical Systems. *Proc. IEEE* 100, 1 (jan 2012), 75 –90.

W. W. Loh and F. J. Dickin. 1996. A novel computer architecture for real-time solution of inverse problems [electric impedance tomography]. In *Advances in Electrical Tomography (Digest No: 1196/143), IEE Colloquium on*. 22/1–22/3. DOI: http://dx.doi.org/10.1049/ic:19960850

George Machan and Victoria Gene Bennett. 2008. Use of Inclinometers for Geotechnical Instrumentation on Transportation Projects. *Transportation Research E-Circular* E-C129 (2008).

Jacob Mattingley and Stephen Boyd. 2012. CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering* 13, 1 (1 March 2012), 1–27.

D. Necsulescu and G. Ganapathy. 2005. Online Solving of Inverse Problems in Critical Infrastructure Monitoring. In *Instrumentation and Measurement Technology Conference, 2005. IMTC 2005. Proceedings of the IEEE*, Vol. 2. 1585–1589.

Paul O'Leary and Matthew Harker. 2012. A Framework for the Evaluation of Inclinometer Data in the Measurement of Structures. *IEEE T. Instrumentation and Measurement* 61, 5 (2012), 1237–1251.

Omair Rafique, Manuel Gesell, and Klaus Schneider. 2013. Targeting Different Abstraction Layers by Model-Based Design Methods for Embedded Systems: A Case Study. *Real-Time Computing Systems and Applications (RTCSA), IEEE Computer Society* (2013).

Abraham Savitzky and Marcel JE Golay. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry* 36, 8 (1964), 1627–1639.

Mehmet Sezer and Mehmet Kaynak. 1996. Chebyshev polynomial solutions of linear differential equations. *International Journal of Mathematical Education in Science and Technology* 27, 4 (1996), 607–618.

Lawrence F. Shampine and Mark W. Reichelt. 1997. The MATLAB ODE Suite. *SIAM J. Sci. Comput.* 18, 1 (Jan. 1997), 1–22.

G.D. Smith. 1985. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Clarendon Press.

J.C. Strikwerda. 2004. *Finite Difference Schemes and Partial Differential Equations*. Society for Industrial and Applied Mathematics.

J. A. Weideman and S. C. Reddy. 2000. A MATLAB differentiation matrix suite. *ACM Trans. Math. Softw.* 26, 4 (Dec. 2000), 465–519.

B. Welfert. 1997. Generation of Pseudospectral Differentiation Matrices I. *SIAM J. Numer. Anal.* 34, 4 (1997), 1640–1657.

Wayne Welsh and Takeo Ojika. 1980. Multipoint boundary value problems with discontinuities I. Algorithms and applications. *J. Comput. Appl. Math.* 6, 2 (1980), 133 – 143.

S. Young. 1982. *Real Time Languages: Design and Development*. Ellis Horwood Publishers, Chichester.

13 | Laser Guided Miner

Continuous miners are used for underground mining tasks with high productivity, e.g., for mining coal or salt deposits, and/or as road headers. A continuous miner operates in a *room and pillar* system, where the mine is divided into a network of rooms (work areas) cut into the seam. The pillars ensure the structural stability of the mine, i.e., the pillars support the roof. Hence, the mining process must always consider the spatial requirements of these pillars. The presented navigation concept allows manual or autonomous control of the continuous miner in an accurate manner [29]. Autonomous control is required when the work safety cannot be guaranteed, e.g., during deep cover retreat mining [26] or in the presence of a hazardous atmosphere.

The optical instrument's principle of operation is sketched in Fig. 13.1 and 13.2. A reference laser plane is projected vertically onto the rear end of the continuous miner. The distance between the laser source and the miner is up to 300 [m], which is necessary because of the high advance during the cutting process. The location of the projected laser line is measured by two large-scale position sensitive detectors (PSD), which are mounted onto the machine with a given height difference. Knowing the location of the laser line on both measurement devices, the included yaw angle ψ between the reference laser beam and the machines principal axis can be determined.

Fig. 13.3 visualizes the measurement principle and Fig. 13.4 shows the system's components in a block diagram. The PSD consists of a housing with a glass window and integrated optical components. The four cameras are connected to a common controller, which has an USB interface to a measurement computer. The measurement data is processed locally and sent to a supervisory system.

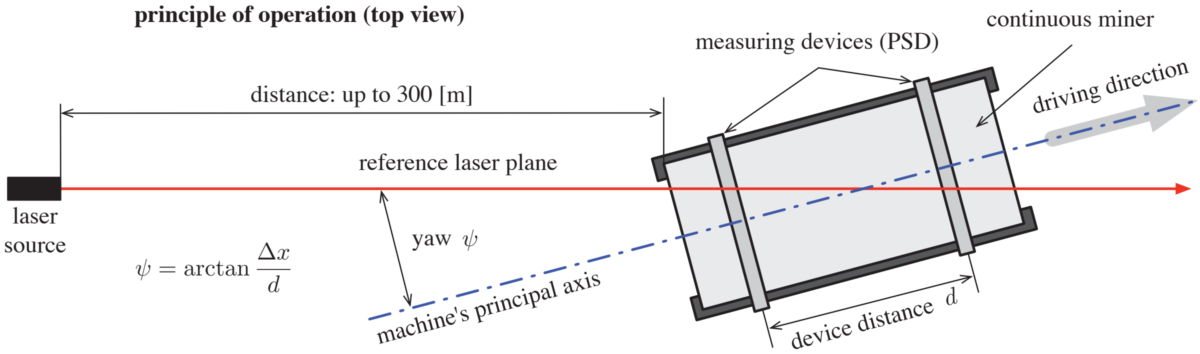


Fig. 13.1 The yaw angle ψ is included between the machine's principal axis and the reference laser plane. The continuous miner can be remotely controlled from distances up to 300 [m].

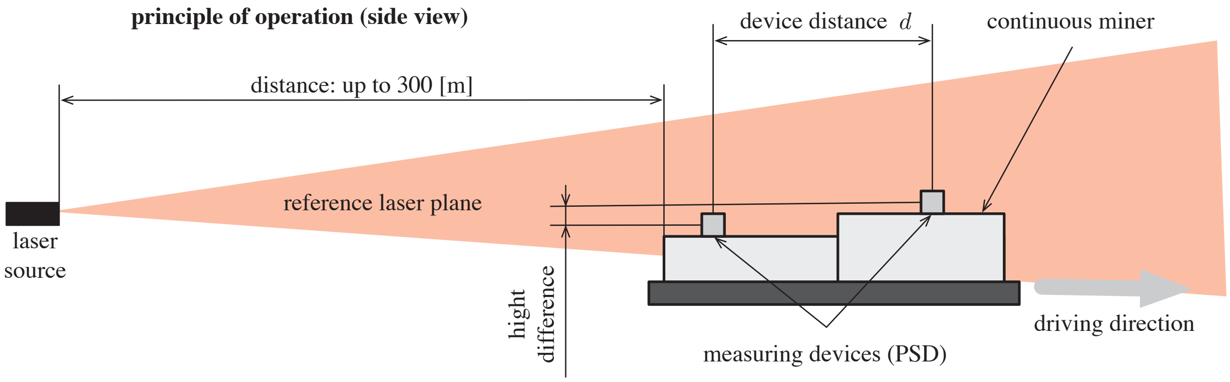


Fig. 13.2 The height difference between both measurement devices (PSD) allows to measure the displacement between both laser lines. The device distance d is known a-priori.

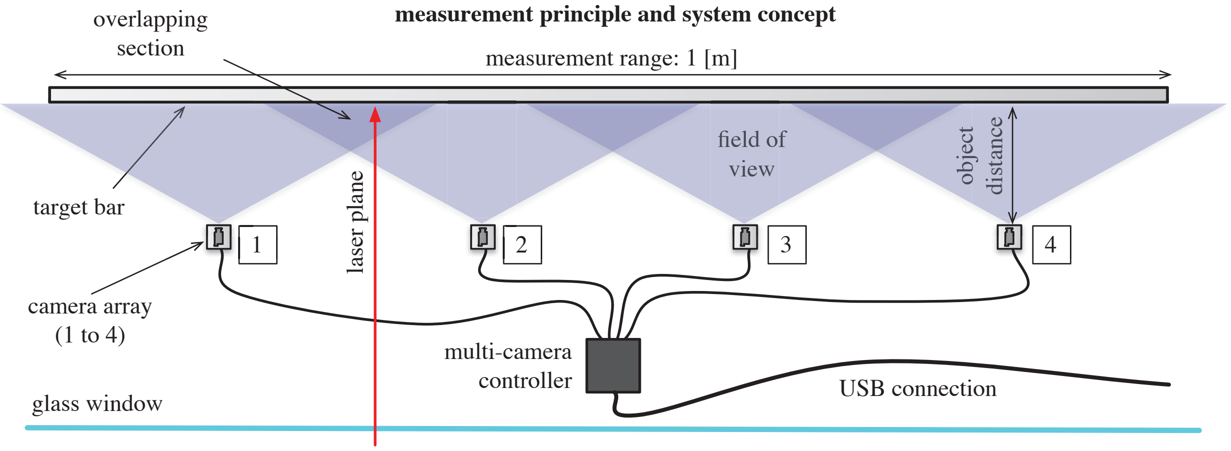


Fig. 13.3 An optical system utilizing four cameras with overlapping fields of view is used to implement the large-scale PSD. The measurement range is 1 [m], the laser plane impinges upon the target bar.

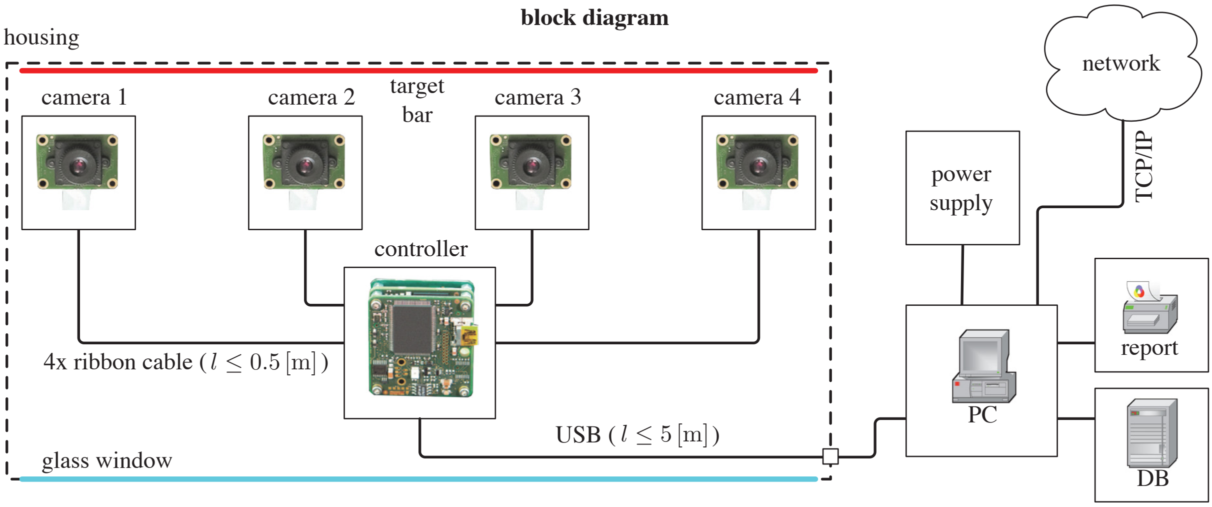


Fig. 13.4 The PC processes the measurement data acquired from the multi-camera controller, generates reports and forwards the information to a database (DB) or a remote supervisory system over network.

The system requirements demand the PSD to have a measurement range of 1 [m]; furthermore, a compact hardware design is required. As shown in Fig. 13.3, the measurement principle is based on four cameras, which significantly reduces the object distance in comparison to using a single camera in a conventional arrangement. Each camera observes a section of the target bar, the raw data for the sensor fusion emerges from the overlapping sections, see Fig. 13.5 and 13.6. The mathematical approach is based on uncertainty weighted multi-source data fusion. The idea is, that the measurements at the center of a camera's field of view are more accurate than at the borders of the support. The measurements are weighted by this uncertainty and fused accordingly. Two alternative approaches for implementing a large-scale PSD without data fusion are shortly presented in the next subsections.

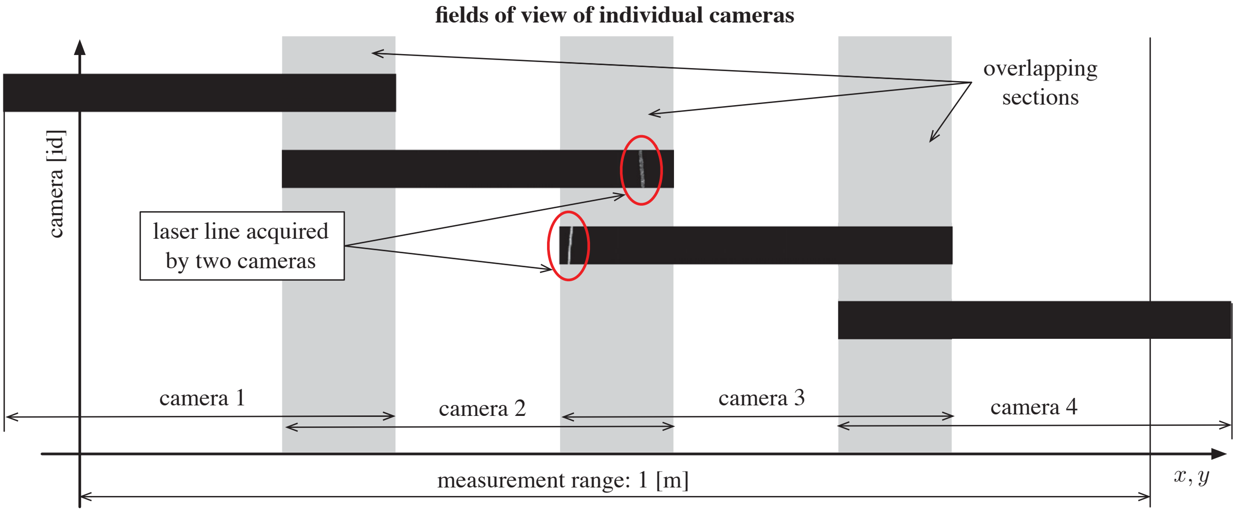


Fig. 13.5 Each camera observes a section of the 1 [m] long target bar. A single laser line is sensed by two adjacent cameras when the laser line is located in the overlapping sections, i.e., fields of view.

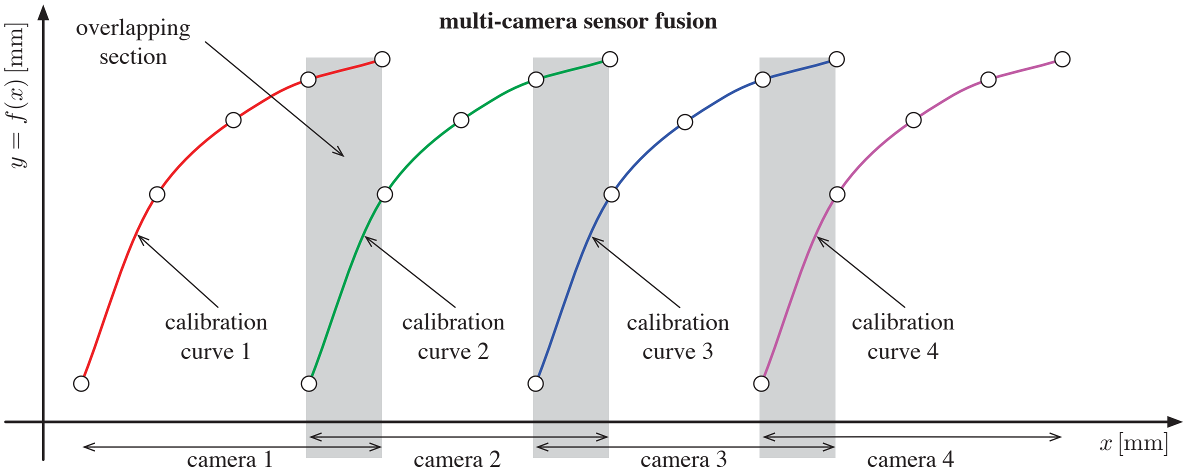


Fig. 13.6 Each of the four cameras yields a separate polynomial calibration curve. The goal is to analytically fuse the curves w.r.t. their uncertainty in order to achieve an extended measurement range while conserving a compact hardware design.

Alternative (1): Optical Fibers

The idea is to have a target bar with equidistant bore holes. Optical fibers are glued into these holes with the front ends facing the laser plane. The individual fibers are bundled and the rear ends face a single camera in a predefined pattern, see Fig.13.7. When the laser plane impinges upon an optical fiber's front end, the light is transported to the corresponding optical fiber's rear end. The measurement procedure matches the illuminated spot with the associated position on the target bar. The drawbacks of this design is the amount of optical fibers to be handled: when covering a 1 [m] measurement range in a 5 [mm] grid, a total number of 200 fibers are required; the cut, place and glue process is time consuming and error prone; and setup is fragile and almost impossible to repair in the field.

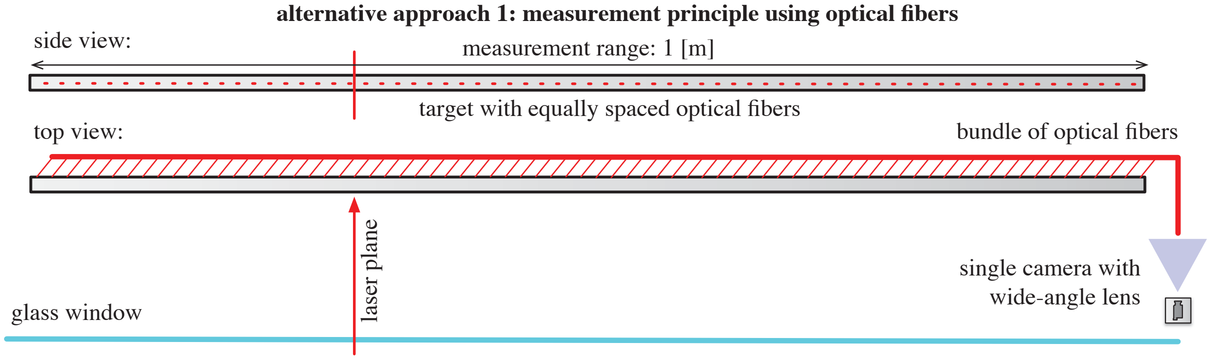


Fig. 13.7 PSD based on optical fibers.

Alternative (2): Scheimpflug Principle

Conventionally, a camera's principal axis is orthogonal to the image plane. An optical setup following the Scheimpflug principle (Chapter 4) enables a depth of field that is not orthogonal to the camera's principal axis, i.e., the subject plane is not parallel to the image plane. In this application, it would have been possible to reach a compact hardware design with a single camera. This setup incorporates several downsides: the lenses are application specific, i.e., they are non-standard, expensive and have a long lead time; and the camera's resolution must be high enough to cover the complete field of view adequately, see Fig. 13.8.

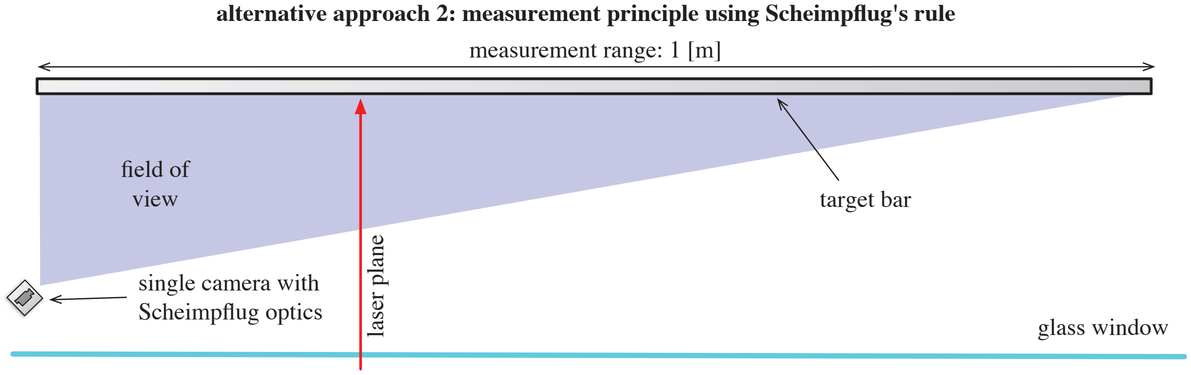


Fig. 13.8 Optical PSD with a single camera utilizing the Scheimpflug principle.

13.1 IEEE - International Instrumentation and Measurement Conference (I2MTC) 2013

May 6 - May 9, 2013 @ Minneapolis, Minnesota, USA [75]

Citation[1]

```
@inproceedings{GuggOlearyHarker2013i2mtc,
 author     = {Gugg, Christoph and O'Leary, Paul and Harker, Matthew},
 title      = {Large Scale Optical Position Sensitive Detector},
 booktitle  = {IEEE International Instrumentation and
               Measurement Technology Conference (I2MTC)},
 pages      = {1775-1780},
 year       = {2013},
 doi        = {10.1109/I2MTC.2013.6555720},
 ISSN       = {1091-5281}}
```

Keywords

continuous miner; machine guidance; multi-source data fusion; position sensitive detector (PSD); system level calibration; multi-camera image processing;

[1] IEEE Xplore Digital Library, ieeexplore.ieee.org

Large Scale Optical Position Sensitive Detector

Christoph Gugg, Paul O'Leary and Matthew Harker
Chair of Automation, Department Product Engineering
University of Leoben, Styria, Austria
Email: christoph.gugg@unileoben.ac.at

***Abstract*—This paper presents the development of a large scale optical position sensitive detector. The device is designed for the precise guidance of machines with respect to a reference laser plane in large working areas. The 1D detector has a measurement range of $1\,[\mathrm{m}]$ and, with the present implementation, a position measurement standard deviation of $s < \pm 0.6\,[\mathrm{mm}]$ in a 95% confidence interval. With this length it is orders of magnitude larger than all presently available position sensitive detectors. The instrument is based on a multi-camera image processing concept. An aluminum bar serves as the target for the laser. The target's surface is specially prepared to ensure optimal scattering of the laser light. Presently, four cameras with overlapping fields of view are deployed to observe the scattered light. Additional optical components reduce the susceptibility to extraneous light sources. Each camera is calibrated using Gram polynomials and the data from the four cameras is fused to give a consistent measurement over the complete measurement range. The linear nature of the computation's algebraic framework offers the advantage that the error propagation can be computed analytically. Weighted polynomial approximation determines the calibration coefficients and weighted polynomial interpolation is used to determine the measurement results. Complete testing of the instrument is presented, whereby cross validation ensures the correct determination of errors. A Kolmogorov-Smirnov test is performed to determine the statistical nature of the measurement errors.**

I. Introduction

The original motivation to develop a large scale position sensitive detector (PSD) was the automatic guidance of roadheaders in the extraction of alkaline salts in mining. The machines need to be positioned over distances of approx. $300-500\,[\mathrm{m}]$ during their operations. The alignment is solved most effectively with the assistance of a reference laser, see Fig. 1. However, significant lateral displacements may occur during facilitating roads and structuring pillars. Testing on retro-reflector-targets and PSDs in outdoor turbulent environments [1] has shown, that accuracies of approx. $10\,[\mathrm{mm}]$ can be achieved over distances of $300\,[\mathrm{m}]$. This indicates, that the instrument's required accuracy can be accomplished even in the presence of extraneous disturbances. Another constraint is that applications in mining commonly employ weak lasers for safety reasons: it must be assumed, that workers are crossing the laser beam's path. The laser light is further weakened through the dusty and moist environment created by the salt mining process. Optical PSDs and/or lateral-effect photo-diodes are utilized in many cases to support the alignment of stationary mechanical [2] components, for laser beam riding techniques [3] and in the positioning of tunnel boring machines [4]–[6]. The working range of the systems is limited by the physical size (i.e. length) of the PSD and/or the associated optics. Large area silicon PSDs from Hamamatsu [7] are limited in size to approx. $35\,[\mathrm{mm}]$ on a side, 1D thin film PSDs [8] are limited to approx. $80-100\,[\mathrm{mm}]$ and PSDs which use a detection board and image processing are limited in size to approx. $110\,[\mathrm{mm}]$ on a side [4]–[6]. This paper describes the development of a new large scale PSD which has a measurement range of $1\,[\mathrm{m}]$ and, in the present implementation, a standard deviation of the measured position error of $s < \pm 0.6\,[\mathrm{mm}]$ in a 95% confidence interval. This is one order of magnitude larger than presently available comparable solutions.

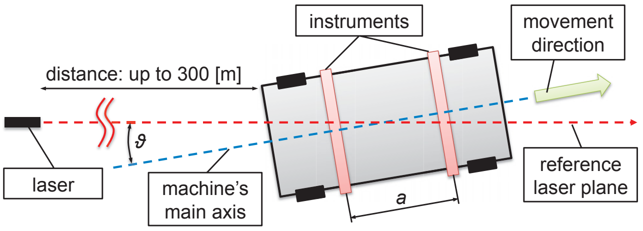


Fig. 1: Two measurement devices must be installed at different highs onto one mining machine. The distance a between them is known a-priori; therefore, it is possible to determine the angle ϑ between the machine's main axis and the reference laser plane. This information is used to guide the machine on the planned route.

The main contributions of this paper are:

1) An innovative optical design which deploys multiple cameras with overlapping fields of view to observe the laser light scattered from a specially prepared target. The test device is limited to four cameras, but the system is fundamentally arbitrary in the number of usable sensors;
2) A calibration and measurement procedure is developed, analyzed and tested which ensures the fusion of the individually acquired data from each of the four cameras into a single consistent displacement measurement;
3) The covariance propagation is calculated for the calibration and measurement computations, thereby enabling the application of weighted polynomial approximation and interpolation. This methodology ensures optimally conditioned basis functions, consequently minimizing the effects of Gaussian noise on the accuracy of the measurement device;

4) The system is calibrated and the accuracy is determined using a cross validation approach. A Kolmogorov-Smirnov test verifies the statistical nature of the measurement error.

II. Principle of Operation

The schematic for the principle of operation is shown in Fig. 2. The housing of the device has a glass window through which the laser light enters the instrument, this is illustrated in Fig. 3. The glass is coated with a special foil, which functions as a hot mirror and reflects up to 95% infrared light with a wavelength above 750 [nm]. This eliminates a large portion of the unwanted optical energy emerging from incandescent light sources. The laser beam impinges upon a 1 [m] long aluminum target bar, whose formerly glossy surface has been sandblasted to improve the Lambertian properties of the laser's scattering. The aluminum is anodized with a red color matched approximately to the wavelength of the laser. The anodized layer functions as a bandpass optical filter reducing the energy towards the blue end of the visible light's spectrum.

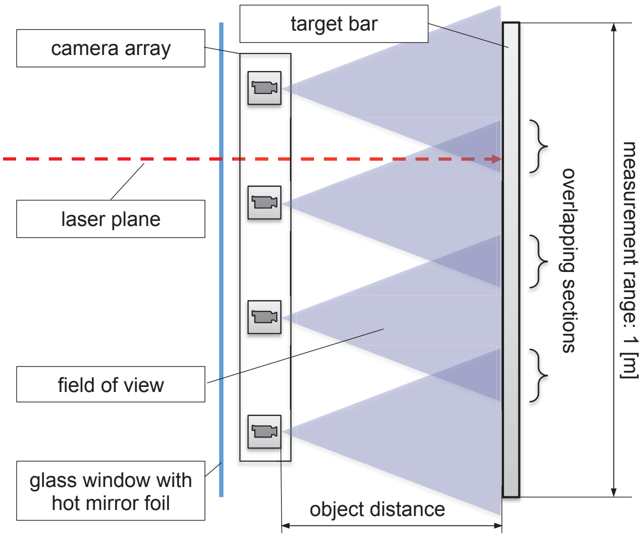


Fig. 2: Schematic of the optical arrangement: the four RGB cameras with the overlapping fields of view are shown together with the coated glass and the red anodized sandblasted aluminum target. The object distance is 125 [mm], which leads to a field of view of approx. 300 [mm] for each camera. Therefore, each overlapping section is approx. 50 [mm]. This hardware configuration delivers maximum red light sensitivity. The two end cameras have slightly different optical conditions; the effect of this can be seen in Fig. 5 and Fig. 6.

The combination of the hot mirror and anodized aluminum significantly improves the conditions within the device for the accurate position detection of the laser line on the target. The installation of multiple cameras with wide-angle lenses enables a compact design for the instrument. RGB sensors are employed; therefore, a monochromatic laser should deliver illumination in the sensor's red channel but not in either of the green or blue channels. This property further improves the spectral selectivity of the device. The images from all cameras are synchronously acquired through a common controller. The acquired laser line's position is then mapped from the individual pixel coordinates of each camera to a global real world coordinate frame via the polynomial calibration curves. Covariance weighted least squares approximation is used during the calibration process to determine the required Gram polynomial coefficients. The aim is to achieve a uniform region of uncertainty surrounding the complete range of the measurement. In the regions of two overlapping fields of view there are two results for the position of the laser line, a covariance weighted average fuses the two results into a combined result with an improved covariance.

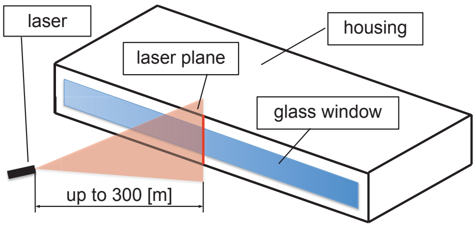


Fig. 3: The laser light enters the measurement instrument through the glass window. Other external environmental light sources are kept outside via the housing. The laser beam impinges upon the target bar, the occurring laser line can then be measured by a set of cameras. The device's dimensions are relatively compact considering the measuring range of 1 [m].

III. Covariance Weighted Polynomial Approximation of a Laser Line

The wide-angle lenses have significant fisheye distortion, which causes the formerly straight laser line to be observed as a curve. This curve can be approximated by an even-degree polynomial. Each camera delivers a region of interest C corresponding to the view of the target with m rows and n columns, whereby i is the row index and j is the column index, i.e. a pixel's illumination value is c_{ij}. This portion in each image is used to detect the position of the laser line, together with its covariance. The first moment x_i, i.e. the center of gravity, of each row is computed by:

$$x_i = \frac{\sum_{j=1}^{n} j c_{ij}}{\sum_{j=1}^{n} c_{ij}} \quad \text{thus} \quad \boldsymbol{x} = [x_1 \dots x_m]^{\mathrm{T}}. \tag{1}$$

The variance σ_i^2 of each row is approximated by the second moment around the measured values $\boldsymbol{x}$, i.e. the moment of inertia:

$$\sigma_i^2 = \sum_{j=1}^{n} c_{ij} (j - x_i)^2 \quad \text{thus} \quad \boldsymbol{\sigma}^2 = \left[\sigma_1^2 \dots \sigma_m^2\right]^{\mathrm{T}}. \tag{2}$$

Fig. 4 shows a typical example. The diagonal covariance matrix Σ is defined through the values of $\boldsymbol{\sigma}^2$:

$$\Sigma = \operatorname{diag}\left(\boldsymbol{\sigma}^2\right) = \operatorname{diag}\left(\sigma_1^2, \dots, \sigma_m^2\right). \tag{3}$$

The observed laser line is described by a weighted least squares polynomial approximation to the first moment $\boldsymbol{x}$; the corresponding cost function K is:

$$K = \|\mathsf{W}(\boldsymbol{x} - \boldsymbol{y})\|_2^2\,, \tag{4}$$

whereby the local weighting matrix is designed as $\mathsf{W} = \Sigma^{-\frac{1}{2}}$, the term $(\boldsymbol{x} - \boldsymbol{y})$ describes the difference between the data points $\boldsymbol{x}$ and the approximated polynomial $\boldsymbol{y} = \mathsf{G}\boldsymbol{\alpha}$. The set of discrete orthogonal Gram polynomial basis functions G has a degree $d = 2$ and $\boldsymbol{\alpha}$ is the coefficient vector [9]. Expanding Equation 4 yields

$$\begin{aligned}
K(\boldsymbol{\alpha}) &= \left\|\Sigma^{-\frac{1}{2}}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})\right\|_2^2, && (5)\\
&= \left[\Sigma^{-\frac{1}{2}}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})\right]^{\mathrm{T}}\left[\Sigma^{-\frac{1}{2}}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})\right], && (6)\\
&= (\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})^{\mathrm{T}}\left(\Sigma^{-\frac{1}{2}}\right)^{\mathrm{T}}\Sigma^{-\frac{1}{2}}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha}), && (7)\\
&= (\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha}). && (8)
\end{aligned}$$

The minimization is carried out through the differentiation of K with respect to the vector $\boldsymbol{\alpha}$ and setting the equation to 0.

$$\frac{\partial K(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = \frac{\partial}{\partial \boldsymbol{\alpha}}\left[(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})\right]. \tag{9}$$

Note, that differentiating $\boldsymbol{\alpha}$ delivers a vector of ones, which is denoted by $\boldsymbol{1}$. This formalism is necessary in order to conserve the correct matrix dimensions. Performing the differentiation in Equation 9 returns

$$\begin{aligned}
0 = &-(\mathsf{G}\boldsymbol{1})^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})\\
&-(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{1}. && (10)
\end{aligned}$$

Expanding leads to

$$\begin{aligned}
0 = &-(\mathsf{G}\boldsymbol{1})^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x} + (\mathsf{G}\boldsymbol{1})^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha}\\
&-\boldsymbol{x}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{1} + (\mathsf{G}\boldsymbol{\alpha})^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{1} && (11)
\end{aligned}$$

and furthermore

$$\begin{aligned}
0 = &-\boldsymbol{1}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x} + \boldsymbol{1}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha}\\
&-\boldsymbol{x}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{1} + \boldsymbol{\alpha}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{1}. && (12)
\end{aligned}$$

Each term now describes a scalar and can therefore be transposed in order to simplify the equation by merging equal expressions.

$$0 = -2(\boldsymbol{1}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x}) + 2(\boldsymbol{1}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha}), \tag{13}$$

which leads to

$$0 = -2\left(\boldsymbol{1}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})\right). \tag{14}$$

Elimination of the constant delivers

$$0 = \boldsymbol{1}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x} - \boldsymbol{1}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha}. \tag{15}$$

Consequently, the normal equation in matrix form is

$$\boldsymbol{1}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x} = \boldsymbol{1}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha}. \tag{16}$$

Premultiplication of $\boldsymbol{1}$ delivers

$$\boldsymbol{1}(\boldsymbol{1}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x}) = \boldsymbol{1}(\boldsymbol{1}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha}) \tag{17}$$

and leads to $\boldsymbol{1}\boldsymbol{1}^{\mathrm{T}} = \mathsf{I}$, i.e. the identity matrix, which can be ignored in this case. Solving the equation

$$\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x} = \mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha} \tag{18}$$

for the coefficients $\boldsymbol{\alpha}$ yields

$$\boldsymbol{\alpha} = \left(\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\right)^{+}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x}, \tag{19}$$

whereby $\mathsf{A}^{+} \triangleq (\mathsf{A}^{\mathrm{T}}\mathsf{A})^{-1}\mathsf{A}^{\mathrm{T}}$ and denotes the Moore-Penrose pseudo inverse. Utilizing this result for the polynomial approximation generates the weighted least squares transformation matrix L:

$$\boldsymbol{y} = \mathsf{G}\boldsymbol{\alpha} = \mathsf{G}\left(\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\right)^{+}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x} \triangleq \mathsf{L}\boldsymbol{x}. \tag{20}$$

The measurement's result is the value p of the polynomial's $\boldsymbol{y}$ midpoint, i.e. it is only necessary to compute this relevant point using the appropriate portion $\boldsymbol{l}^{\mathrm{T}}$ of the transformation matrix L, consequently

$$p = \boldsymbol{e}_t^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{e}_t^{\mathrm{T}}\mathsf{L}\boldsymbol{x} = \boldsymbol{l}^{\mathrm{T}}\boldsymbol{x}, \tag{21}$$

whereby $\boldsymbol{e}_t$ is a coordinate vector and $t = \frac{m-1}{2} + 1$ under the condition, that the number of rows m is odd. Equation 20 is a linear operation, the corresponding covariance propagation is

$$\hat{\Sigma} = \mathsf{L}\Sigma\mathsf{L}^{\mathrm{T}}, \tag{22}$$

see Fig. 4 for an example. Based on the matrix $\hat{\Sigma}$, the variances $\hat{\boldsymbol{\sigma}}^2$ of each fitted point $\boldsymbol{y}$ is determinable:

$$\hat{\boldsymbol{\sigma}}^2 = \operatorname{diag}(\hat{\Sigma}) = (\mathsf{L} \circ \mathsf{L})\boldsymbol{\sigma}^2, \tag{23}$$

where $\circ$ is the Hadamard product, off-diagonal elements of $\hat{\Sigma}$ do not have any meaning in this context. The variance of the point p is then $\lambda^2 = \boldsymbol{e}_t^{\mathrm{T}}\hat{\boldsymbol{\sigma}}^2$, this information is required for the calibration of the device.

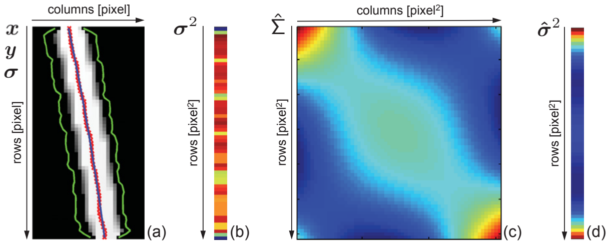


Fig. 4: A typical image of a laser line along with the first moment $\boldsymbol{x}$ (red), its standard deviation $\boldsymbol{\sigma}$ (green) and polynomial approximation $\boldsymbol{y}$ (blue) is shown in (a). The variance $\boldsymbol{\sigma}^2$ of the local position of the line is illustrated in (b). The covariance matrix $\hat{\Sigma}$ of the complete polynomial fit is presented in (c) along with the variance $\hat{\boldsymbol{\sigma}}^2$ of the polynomial estimate in (d).

IV. Calibration via Polynomial Approximation

The calibration process delivers the coefficients of four individual sets of calibration curves $\{\boldsymbol{z}_{(1)}, \boldsymbol{z}_{(2)}, \boldsymbol{z}_{(3)}, \boldsymbol{z}_{(4)}\}$.They are computed by combining the individually acquired camera space coordinates $\{\boldsymbol{p}_{(1)}, \boldsymbol{p}_{(2)}, \boldsymbol{p}_{(3)}, \boldsymbol{p}_{(4)}\}$ along with their variances $\{\boldsymbol{\lambda}^2_{(1)}, \boldsymbol{\lambda}^2_{(2)}, \boldsymbol{\lambda}^2_{(3)}, \boldsymbol{\lambda}^2_{(4)}\}$ with the global real space coordinates in $\boldsymbol{q}$. The complete device is calibrated using h equidistant calibration points, in this case $h = 99$ points were defined. The position of the points ensures that each camera has $> h/3$ points for calibration, i.e. here a minimum of approx. 33 were used for the application. The respective polynomial coefficients $\boldsymbol{z}_{(k)}$ are determined using a weighted least squares approximation by minimizing the according cost function $H_{(k)}$:

$$H_{(k)} = \left\| \mathsf{V}_{(k)} (\boldsymbol{p}_{(k)} - \boldsymbol{z}_{(k)}) \right\|_2^2 . \tag{24}$$

The global weighting matrix is designed as $\mathsf{V}_{(k)} = \Lambda_{(k)}^{-\frac{1}{2}}$ and the covariance matrix is $\Lambda_{(k)} = \operatorname{diag}(\boldsymbol{\lambda}^2_{(k)})$. Polynomial interpolation is utilized to determine the mapping $\boldsymbol{z}_{(k)}$ for intermediate points. Exact details of the least squares fitting an interpolation can be found in [9]. A single polynomial is defined as $\boldsymbol{z} = \mathsf{G}\boldsymbol{\beta}$, whereby the matrix G represents the Gram polynomial of degree $d = 6$. The submatrix $\bar{\mathsf{G}}$ is generated by extracting the rows of G corresponding to the matching calibration points, i.e. the known portions of G. The coefficient vector $\boldsymbol{\beta}$ is calculated analogously to Equation 4.

$$\boldsymbol{z} = \mathsf{G}\boldsymbol{\beta} = \mathsf{G} \left(\bar{\mathsf{G}}^{\mathrm{T}} \Lambda^{-1} \bar{\mathsf{G}} \right)^{+} \bar{\mathsf{G}}^{\mathrm{T}} \Lambda^{-1} \boldsymbol{p} \triangleq \mathsf{M}\boldsymbol{p}. \tag{25}$$

As in Equation 22, it is possible to compute the covariance matrix of the fitted polynomial $\boldsymbol{z}$ thanks to its linear nature:

$$\hat{\Lambda} = \mathsf{M}\Lambda\mathsf{M}^{\mathrm{T}} \quad \text{and} \quad \hat{\boldsymbol{\lambda}}^2 = \operatorname{diag}(\hat{\Lambda}) = (\mathsf{M} \circ \mathsf{M})\boldsymbol{\lambda}^2 \tag{26}$$

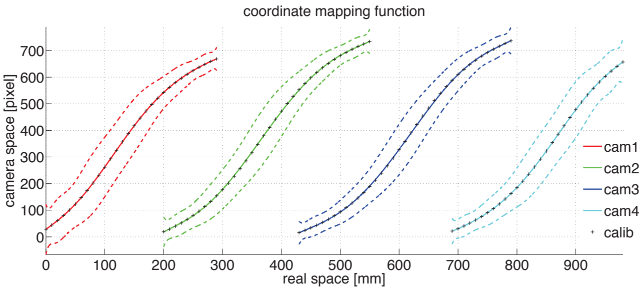


Fig. 5: From left to right are the four individual camera calibration curves $\{\boldsymbol{z}_{(1)}, \boldsymbol{z}_{(2)}, \boldsymbol{z}_{(3)}, \boldsymbol{z}_{(4)}\}$, which map the respective [pixel] coordinates to real world coordinates in [mm]. A 10 [mm] raster has been used, resulting in 99 equally spaced calibrations points. The confidence intervals are scaled by a factor of 30 to make them visible.

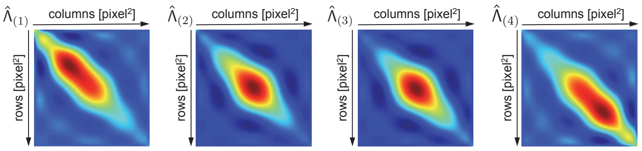


Fig. 6: The covariance matrices $\{\hat{\Lambda}_{(1)}, \hat{\Lambda}_{(2)}, \hat{\Lambda}_{(3)}, \hat{\Lambda}_{(4)}\}$ correspond to calibration curves in Fig. 5. Note the asymmetry of $\{\boldsymbol{z}_{(1)}, \boldsymbol{z}_{(4)}\}$ and $\{\hat{\Lambda}_{(1)}, \hat{\Lambda}_{(4)}\}$ caused by the optical conditions at the ends. The weighting at the borders of the support is amplified to acquire a better mapping over the whole measurement range.

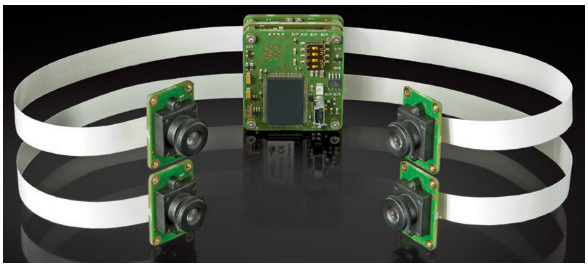


Fig. 7: The multi-camera controller *VRmagic MFC-12M-4* is capable of acquiring four images synchronously. Each sensor deploys its own calibration curve as shown in Fig. 5. The extended field of views are caused by deploying wide-angle lenses, however this is also the reason for serious non-linear distortions and results in the error propagation shown in Fig.6.

V. Measurement via Polynomial Interpolation

The individual calibration curves with their uncertainties for each camera can be seen in Fig. 5 and Fig. 6. The calibration curves are stored in the form of lookup tables with a real space resolution of 1 [mm]. The detected laser line position in [pixel] is mapped to the nearest neighbor element in the lookup table, the value and its variance for the corresponding real world coordinate in [mm] are retrieved. In areas with overlapping fields of view A and B, the mapping delivers two results q_A and q_B for the position, each with its variance λ_A^2 and λ_B^2. This effect is caused by the optical arrangement and is illustrated in Fig. 2. A variance weighted average is applied to fuse both results into a single position $\hat{q}$ with superior accuracy than the individual values:

$$\hat{q} = \frac{\lambda_A^{-1} q_A + \lambda_B^{-1} q_B}{\lambda_A^{-1} + \lambda_B^{-1}} . \tag{27}$$

VI. Experimental Verification

The complete prototype PSD is shown in Fig. 8, together with a prism on a linear drive's stage for system calibration and verification. The cross validation experiment was performed at 98 measurement points between the 99 equidistant calibration points, i.e. they are located where the largest error is to be expected. The measurement's standard deviation s is

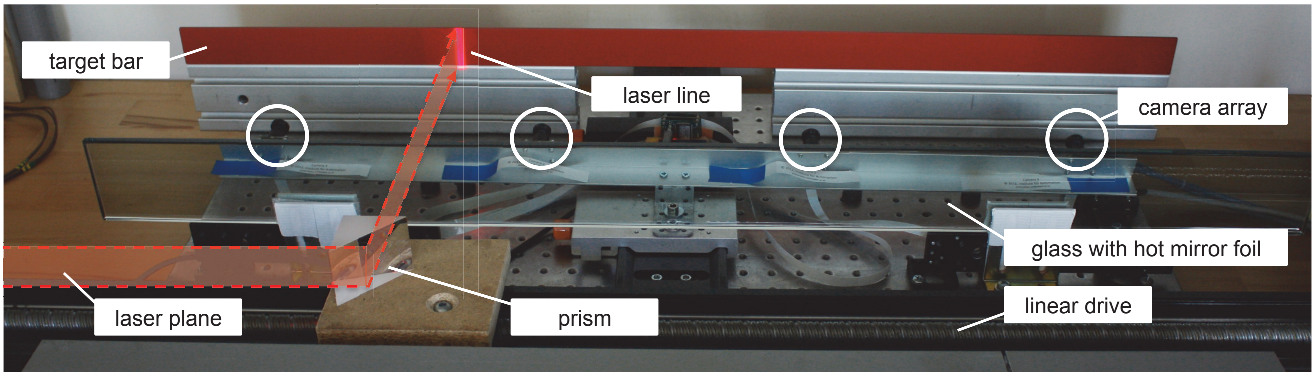


Fig. 8: The prototype PSD, showing the four cameras, the glass window with coated hot mirror foil and the red anodized aluminum target bar. A prism mounted on a linear drive's stage redirects the laser plane generated by the construction laser to cover the whole measurement range of $1\,[\mathrm{m}]$ for system calibration as well as cross validation experiments. The setup matches up with the draft in Fig. 2.

computed as a metric of quality, i.e measurement accuracy. Furthermore, the Gaussian distribution of the data was proven by applying a non-parametric Kolmogorov-Smirnov test with a 95% confidence interval on the samples [10]. The tests showed, that this is always true for the camera coordinates $\boldsymbol{p}$. As a consequence, due to the proposed method's linear nature, this condition also holds for the mapped coordinates $\boldsymbol{q}$ [11]. Following hardware components were deployed in the experimental setup and Fig. 8 shows their assembly:

1) The red sandblasted anodized aluminum target bar with a horizontal dimension of $1\,[\mathrm{m}]$;
2) The multi-camera controller *VRmagic MFC-12M-4* [12] with four 1/3" RGB CMOS sensors, a resolution of $752 \times 480\,[\mathrm{pixel}]$, a fixed aperture size of $f2$ and an adjustable exposure time t_e. The wide-angle lenses have a focal length of $2.1\,[\mathrm{mm}]$. The controller with the individually attached sensors is shown in Fig. 7;
3) A linear drive composed of two parts: the controller *iTK Pollux Drive Typ-3* [13] and the linear axis *THK KR5520A+1280LH0-0000* [14], featuring an usable length of $1.1\,[\mathrm{m}]$ an accuracy of $\pm 0.1\,[\mathrm{mm}]$;
4) The class $3A$ telecentric construction laser *AGL TCL2000* with a power of $4\,[\mathrm{mW}]$ and a wavelength of $635\,[\mathrm{nm}]$. The generated laser beam has a thickness of approx. $12\,[\mathrm{mm}]$. The laser oscillates on a frequency of $50\,[\mathrm{Hz}]$ and thereby generates the laser plane. The device is shown in Fig. 9;
5) A $10\,[\mathrm{mm}]$ thick window glass with coated hot mirror foil *3M Silver 20* [15] with 72% IR reduction, $t_e = 200\,[\mathrm{ms}]$ and *Bruxsafol LX-70* [16] with 95% IR reduction, $t_e = 50\,[\mathrm{ms}]$. For a comparison see Fig. 10 [17]. Although the exposure times are fixed in this setup, they need to be dynamically adapted in the real application when the machine is moving away from the laser source.

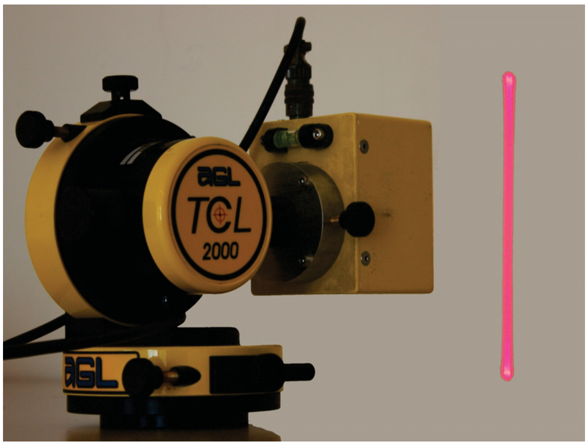


Fig. 9: The oscillating construction laser *AGL TCL2000* generates a laser line of approx. $12\,[\mathrm{mm}]$ thickness with a frequency of $50\,[\mathrm{Hz}]$, i.e. one phase has a duration of $20\,[\mathrm{ms}]$.

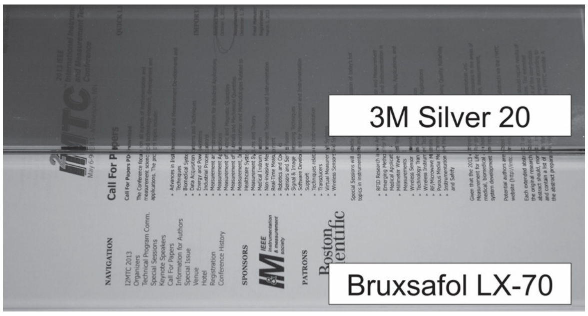


Fig. 10: The illustration shows clearly, that both types of hot mirror foils have non-identical light transmission rates. As a consequence, different exposure times t_e for the cameras must be set depending on the used foil.

The relationship between pixel dimensions, i.e. resolution, and the error in detecting the position of a sharp edge in an image was investigated in [18], [19]. It was determined that a subpixel accuracy of a factor $10 \dots 20$ can be achieved reliably. In this experimental setup $1\,[\text{pixel}]$ in the camera corresponds to approx. $0.4\,[\text{mm}]$ in the real space, i.e. with subpixeling an accuracy of at least $0.04\,[\text{mm}]$ is theoretically possible. The observed experimental results in Table I gave a 95% confidence interval of $< \pm 0.6\,[\text{mm}]$, therefore it is concluded, that the camera's resolution is not the limiting factor in this system. The primary uncertainty is the illumination associated with the laser, i.e. small variations in the width of the laser line and the distribution of intensities within the generated line.

In order to verify the effectiveness of the weighted polynomial approximation, the resulting measurement standard deviation s_w is compared to a conventional, i.e. unweighted, fit's standard deviation s_c. This approach utilizes the mean $\overline{\sigma^2}$ and $\overline{\lambda^2}$ of the variances $\boldsymbol{\sigma}^2$ and $\boldsymbol{\lambda}^2$. Multiplication with the identity matrix I delivers the values for the weighting matrices W and V.

	$s_w/\,[\text{mm}]$	$s_c/\,[\text{mm}]$
local weighting global weighting	$\mathsf{W} = \Sigma^{-\frac{1}{2}}$ $\mathsf{V} = \Lambda^{-\frac{1}{2}}$	$\mathsf{W} = \overline{\sigma^2}\mathsf{I}$ $\mathsf{V} = \overline{\lambda^2}\mathsf{I}$
no glass	± 0.507	± 0.594
Silver 20	± 0.552	± 0.728
LX-70	± 0.532	± 0.608

TABLE I: Cross validation results for the standard deviation in a 95% confidence interval under laboratory conditions. This table summarizes the obtained results of 98 equally spaced cross validation points.

Interpretation of the results in Table I leads to following insights: the weighted polynomial approximation is superior to its conventional pendant. The hot mirror foil *LX-70* establishes a better measurement environment with respect to extraneous light. In direct comparison, the hot mirror foil *Silver 20* absorbs too much light.

VII. Conclusion

This work has demonstrated the feasibility of deploying multi-sensor imaging systems in large scale measurement devices. The new position sensitive detector delivers a measurement standard deviation of $s < \pm 0.6\,[\text{mm}]$ in a 95% confidence interval over a measurement range of $1\,[\text{m}]$. The present system is limited to four cameras; however, the algebraic framework utilizing weighted polynomial approximation is capable of metrically fusing the data of an arbitrary number of sensors and has proved viable for real-time measurements. It also delivers the covariance propagation from the individual camera to the final single result. This will be of growing importance in future measurement systems due to the increasing availability of low cost hardware and cheap computation power.

Acknowledgment

We want to thank the company of Geodata[1] for the cooperative work and the supplied technical equipment.

References

[1] A. Makynen, J. Kostamovaara, and R. Myllyla, "Displacement sensing resolution of position sensitive detectors in atmospheric-turbulence using retroreflected beam," *IEEE Transactions on Instrumentation and Measurement*, vol. 46(5), pp. 1133–1136, 1997.

[2] J. Piotrowski, *Shaft Alignment Handbook*. M. Dekker, 1995.

[3] J. Dubois, "Very high angular resolution laser beam rider detector having a gated image intensifier and a video camera," United States patent no. 5.280.167, 1994.

[4] R. Neumayr, M. Harker, P. O'Leary, and J. Golser, "New approaches to machine vision based displacement analysis," *IEEE International Instrumentation and Measurement Technology Conference*, pp. 979–984, 2012.

[5] X. Shen, M. Lu, S. Fernando, and S. AbouRizk, "Tunnel boring machine positioning automation in tunnel construction," *Gerontechnology*, vol. 11(2), 2012.

[6] X. Shen and M. Lu, "Development of virtual laser target board to tunnel boring machine guidance control," *International Conference on Computing in Civil Engineering*, pp. 413–420, 2012.

[7] Hamamatsu Photonics K.K., "1d position sensitive detectors," technical specification, 2012. [Online]. Available: www.hamamatsu.com

[8] E. Fortunato, G. Lavareda, R. Martins, F. Soares, and L. Fernandes, "Large-area 1d thin-film position-sensitive detector with high detection resolution," *Sensors and Actuators A: Physical*, vol. 51(2-3), pp. 135–142, 1996.

[9] P. O'Leary and M. Harker, "An algebraic framework for discrete basis functions in computer vision," *Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 150–157, 2008.

[10] E. Kreyszig, *Statistische Methoden und ihre Anwendungen*. Göttingen: Vandenhoeck und Ruprecht, 1979.

[11] J. C. Clarke, "Modelling uncertainty: A primer," Department of Engineering Science, Oxford University, Tech. Rep., 1998.

[12] VRmagic GmbH, "MFC-12M-4," technical specification, 2012. [Online]. Available: www.vrmagic.com/imaging/

[13] ITK Dr. Kassen GmbH, "iTK Pollux Drive Typ-3," technical specification, 2012. [Online]. Available: www.itknet.de

[14] THK Co. Ltd., "THK KR5520A+1280LH0-0000," technical specification, 2012. [Online]. Available: www.thk.com

[15] 3M, "Silver 20," technical specification, 2012. [Online]. Available: www.3m.com

[16] Bruxsafol Folien GmbH, "LX-70," technical specification, 2012. [Online]. Available: www.bruxsafol.de

[17] W. Körner, "Transmissions- und Reflexionsgrad sowie g-Wert einer Folie auf Glas," Bayerisches Zentrum für angewandte Energieforschung E.V., Tech. Rep., 2008.

[18] P. Schalk, "Metric vision methods for material and product inspection," Ph.D. dissertation, University of Leoben, 2007.

[19] E. Fauster, "Statistical uncertainty analysis for image processing algorithms in metric vision systems," Ph.D. dissertation, University of Leoben, 2008.

[1] ©Geodata GmbH, Leoben, Austria, www.geodata.at

13.2 IEEE - Transactions on Instrumentation and Measurements (TIM) 2014

I2MTC 2013 Special Issue [76]

Citation[2]

```
@article{GuggOlearyHarker2013tim,
 author     = {Gugg, Christoph and O’Leary, Paul and Harker, Matthew},
 title      = {Analytic Multi-Source Data Fusion and its
               Application to a Large-Scale Optical PSD},
 journal    = {IEEE Transactions on Instrumentation and Measurement (TIM)},
 volume     = {63},
 number     = {5},
 pages      = {1116-1126},
 year       = {2014},
 doi        = {10.1109/TIM.2013.2286894},
 ISSN       = {0018-9456}}
```

Keywords

continuous miner; machine guidance; uncertainty-weighted data fusion; position sensitive detector (PSD); system level calibration; multi-camera image processing; analytic tensor interpolation; statistical testing;

[2] IEEE Xplore Digital Library, `ieeexplore.ieee.org`

Analytic Multi-Source Data Fusion and its Application to a Large-Scale Optical PSD

Christoph Gugg, Paul O'Leary and Matthew Harker

***Abstract*—This article introduces an analytic algebraic framework for multi-source data fusion (MSDF) utilizing covariance weighted discrete orthogonal polynomials (DOP). The approach is implemented and tested in a prototype for a large-scale optical position sensitive detector (PSD). The device is designed for the precise guidance of machines with respect to a reference laser plane in large working areas. The 1D detector has a measurement range of $1\,\mathrm{m}$ and, with the present implementation, a position measurement standard deviation of $s < \pm 0.6\,\mathrm{mm}$ in a $95\,\%$ confidence interval at a distance of $300\,\mathrm{m}$. With this length it is orders of magnitude larger than all presently available PSDs. The instrument's concept is based on a multi-camera image processing setup, enabling a relatively compact hardware design. An aluminum bar serves as the target for the laser. The target's surface is specially prepared to ensure optimal scattering of the laser light. Presently, four cameras with wide-angle lenses and overlapping fields of view monitor the scattered light; however, the theoretical framework supports the fusion of data from an arbitrary number of sensors. Additional optical components reduce the susceptibility to ambient light sources. Each camera is calibrated using Gram basis functions and the data from the four cameras is fused to give a consistent measurement over the complete measurement range. The linear nature of the computation offers the advantage that the error propagation can be derived analytically. Weighted polynomial approximation determines the calibration coefficients and weighted polynomial interpolation is used to obtain the measurement results. Complete testing of the instrument is presented, whereby cross validation ensures the correct quantification of errors. A Kolmogorov-Smirnov test is performed to prove the Gaussian nature of the measurement data and its error.**

***Index Terms*—multi-source data fusion, position sensitive detector, weighted discrete orthogonal polynomials, uncertainty estimation, subsurface working area, guidance system.**

I. Introduction

THIS article presents a new generic analytic approach to multi-source data fusion (MSDF) and calibration for consistent measurements. The method provides an analytic framework for the computation of the complete system's covariance propagation and, as a consequence, enables the determination of confidence envelopes through the complete calibration and measurement procedures. The approach is particularly well suited for merging data from multiple sensors within a single instrument and places no fundamental limit on the number of sensors used. A large-scale, yet compact, optical position sensitive detector (PSD) with an extended measurement range is designed based on this theory. The device employs an array of imaging sensors with partially overlapping fields of view, which are effectively used for the signal fusion.

A. Measurement Devices for Displacement Detection

PSDs and laser pointing devices are commonly employed for automatic positioning and alignment applications. Such instruments typically cover measurement ranges of several cm. Conventional optical PSDs or lateral-effect photo-diodes are utilized to support the alignment of stationary mechanical components [1], for laser beam riding techniques [2] and in the positioning of tunnel boring machines [3]–[5]. However, the working ranges of the named systems is limited by the physical size of the PSD and the associated optics. Large area silicon PSDs from Hamamatsu [6] are limited in size to approximately $35\,\mathrm{mm}$ on a side, thin film PSDs [7] are limited to approximately $80 - 100\,\mathrm{mm}$ and PSDs which use a detection board and image processing are limited in size to approximately $110\,\mathrm{mm}$ on a side [3]–[5].

B. Concepts for Multi-Source Data Fusion

With the emergence of cyber physical systems and its multi-modal sensor arrangements, much has been written on MSDF; overviews can be found in [8]–[10]. The hierarchical Joint Directors of Laboratories (JDL) fusion model [11] deals with data fusion in decision support systems. The model is predominantly feature and pattern based. It contains no concrete proposals for data fusion at the signal-processing level. The paper by Liu et al. [12] presents a survey of data fusion methods in non-destructive testing. Grouping is possible based on the techniques used: optimization methods, multiresolution analysis (MRA) approaches, heuristic methods and probabilistic methods (see [12, Table II]). None of the introduced techniques include analytic approaches for calibration and measurement, which is necessary for an estimation of uncertainty. Fuzzy methods [13] are not suitable for instrumentation due to the non-analytic nature of their membership functions; similarly, this is also the case for neural network data fusion [14] where the network has non-linear behavior. Dempster-Shafer theory [15] delivers a framework for merging evidence to obtain a level of belief in a result: it is a probabilistic approach applicable to determining the validity of a hypothesis; however, this is not the case presented in a measurement instrument. This criterion for exclusion also applies for Bayesian theory [10]. Covariance estimation [16] has been used to merge the results from collections of single measurements; however, no provision is made for the inclusion

The authors are with the Chair of Automation, Department Product Engineering, University of Leoben, 8700 Leoben, Austria (e-mail: christoph.gugg@unileoben.ac.at; paul.oleary@unileoben.ac.at; matthew.harker@unileoben.ac.at)

of calibration transfer curves. A confidence based approach has also been published, but used a fuzzy definition of confidence [17].

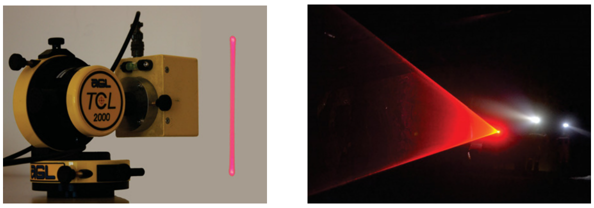
C. *Contributions of this Work*

This article describes the development of a new PSD, which covers fields of applications where large-scale measurements of up to 1m are required. Underground facilities such as mines, shafts and tunnels are investigated, because automatic machines often depend on accurate laser guidance systems [18]–[20]. The harsh environmental conditions provide a rich testing field. Some initial portions of this article were presented in [21]. The following main contributions are made:

1) A compact hardware concept for a large-scale optical PSD is present. An array of cameras with overlapping fields of view monitor a specially prepared target, thereby extending the instrument's measurement range.
2) A methodology for analytic MSDF for the calibration and measurement process of an instrument with multiple sensors is established. The fusion process is implemented by employing uncertainty weighted discrete orthogonal polynomials (DOP). It is possible to determine the expected uncertainty a-priori. The model is fundamentally arbitrary w.r.t. the number of sensors used.
3) The experimental prototype is calibrated and the accuracy is determined using a cross validation approach. In this implementation, the laboratory setup uses four individual cameras. The suitability of the mathematical model is verified and the statistical nature of the measurement error is validated.

II. Hardware Design of the Instrument

A. *Environmental Conditions and Constraints*



(a) construction laser

(b) reference laser plane

Fig. 1. The class 3A telecentric industrial construction laser *AGL TCL2000* in (a) with a power of 4 mW and a wavelength of 635 nm generates the reference laser plane in (b). The oscillating frequency of the rocking mirror is 50 Hz, thereby generating the laser line in Fig. 2. Measurement results for the environmental conditions are available for up to 200 m [22].

The presented application uses a reference laser plane for the guidance system, see Fig. 1. The PSD is installed on an automated mining machine and detects this plane. The machine needs to be positioned accurately over distances of up to 300 m. As shown in Fig. 2, the laser beam's diameter/width is a function of the distance between the laser source and the target. This is mainly caused by the dust concentration in the mine (see Fig. 3) and the resulting diffraction of the laser light. The dust concentration is homogeneous except for local peaks. Fig. 4 shows how the power and the distance of the laser depend on the distance. Only relatively weak lasers are permitted in subsurface working areas [23]. The presence of external white light sources negatively influences the measurement's quality. Hot mirror foils are therefore attached to the instrument's window to reflect up to 95 % of the IR/NIR portions of the visible spectrum. The system specification of the PSD requires a measurement range of 1 m and relatively compact hardware dimensions.

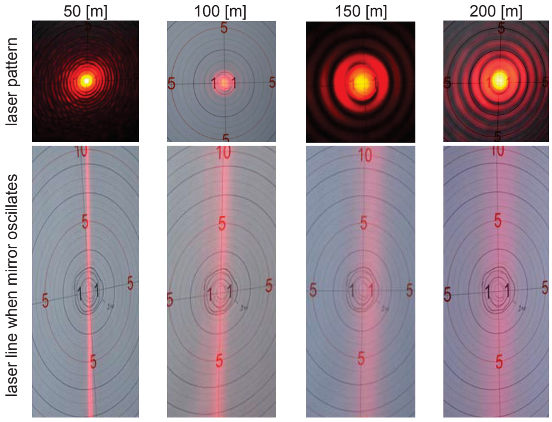


Fig. 2. A total of eight measurements are presented w.r.t. the laser pattern as a function of distance. These measurements were performed in a salt mine [22] which offers typical operating conditions. The top four images show the observed laser pattern at 50 m, 100 m, 150 m and 200 m respectively. The low four images show the corresponding laser lines when the oscillating mirror is active. The width of the laser line increases linearly from approximately 10 mm to approximately 40 mm.

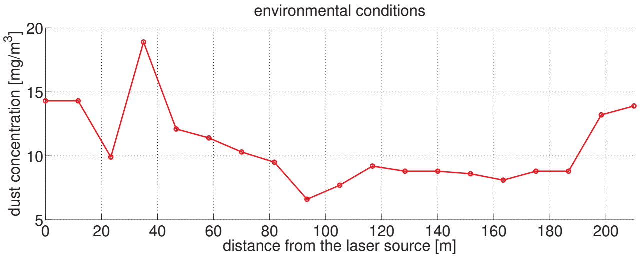


Fig. 3. The dust concentration has been measured in an underground environment over a distance of 200 m [22].

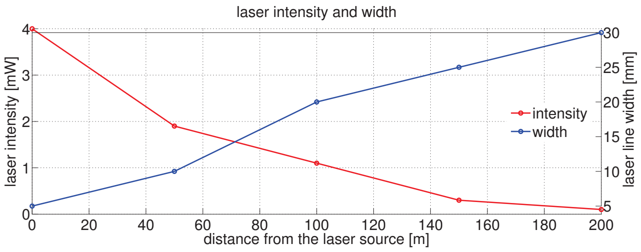


Fig. 4. The intensity of the laser light (red) and the width of the resulting laser line (blue) have been measured over a distance of 200 m [22].

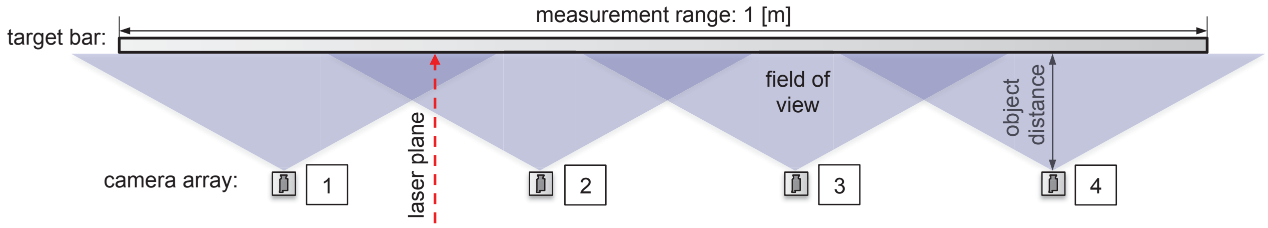


Fig. 5. Schematic of the optical arrangement: four cameras with wide-angle lenses provide an extended measurement range of 1 m. The object distance between the cameras and the target is 125 mm, which leads to a field of view of approximately 300 mm for each camera. The width of an overlapping section is approximately 50 mm. The two end cameras have different optical conditions.

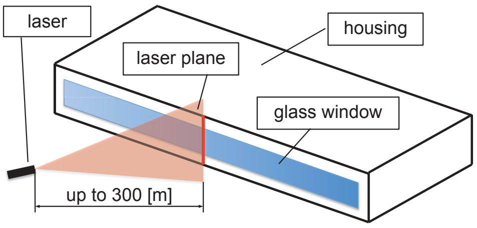


Fig. 6. The laser light enters the measurement instrument through the glass window. The laser beam impinges upon the target inside the housing, whereby the projected laser line is imaged by a set of cameras.

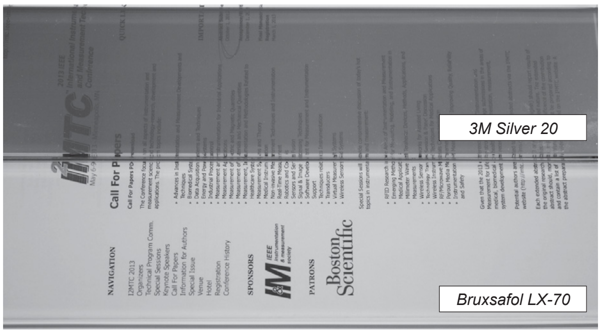


Fig. 7. Two samples of the 10 mm thick window glass have been coated with different kinds of hot mirror foils in order to remove the IR and NIR portions of visible light with a wavelength > 750 nm from the acquired measurement data: *3M Silver 20* [24] with 72 % IR/NIR reduction and *Bruxsafol LX-70* [25] with 95 % IR/NIR reduction.

B. Optical Arrangement of the Multi-Camera Setup

In order to achieve a compact hardware design, an array of cameras with overlapping fields of view are used; see the schematic in Fig. 5. The housing of the device is sketched in Fig. 6, the mentioned hot mirror foil (see Fig. 7) is attached to the window. The images from all cameras are synchronously acquired through a common controller as shown in Fig. 8. The wide-angle lenses cause non-linear optical distortion. RGB sensors are employed; therefore, a monochromatic laser with a wavelength of 635 nm should deliver illumination in the sensor's red channel but not in either of the green or blue channels. In the regions of two overlapping fields of view, there are two results for the position of the laser line, which are used for the MSDF.

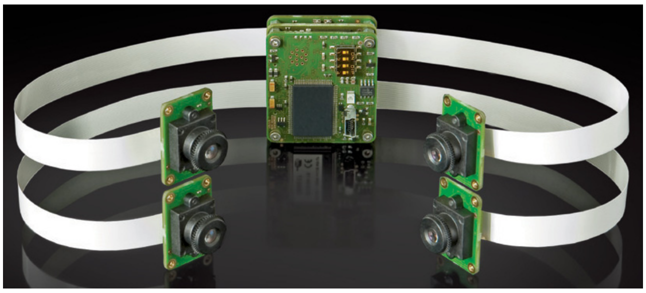


Fig. 8. The multi-camera controller *VRmagic MFC-12M-4* [26] is capable of acquiring four images synchronously with its 1/3” RGB CMOS sensors. The resolution of each sensor is 752 × 480 pixel. The focal length of the wide-angle lenses is $f = 2.1$ mm and the aperture size is $f/2$.

C. Laboratory Prototype

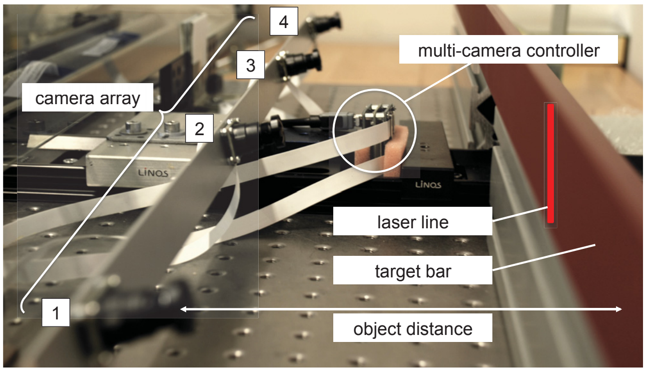


Fig. 9. The vertical section of the laboratory prototype shown in Fig. 10. The sensors are connected to the common controller shown in Fig. reffigCameraController. The laser line is projected onto the target bar.

The vertical section of the prototype PSD is shown in Fig. 9, the complete arrangement is pictured in Fig. 10. The laser beam impinges upon a 1 m long aluminum target bar, whose surface has been sand-blasted to improve the Lambertian properties of the laser light's scattering. The aluminum is anodized with a red color matched approximately to the wavelength of the laser light. The anodized layer functions as a bandpass optical filter reducing the energy towards the blue end of the visible spectrum. To summarize, the spectral selectivity of the device is improved by following measures:

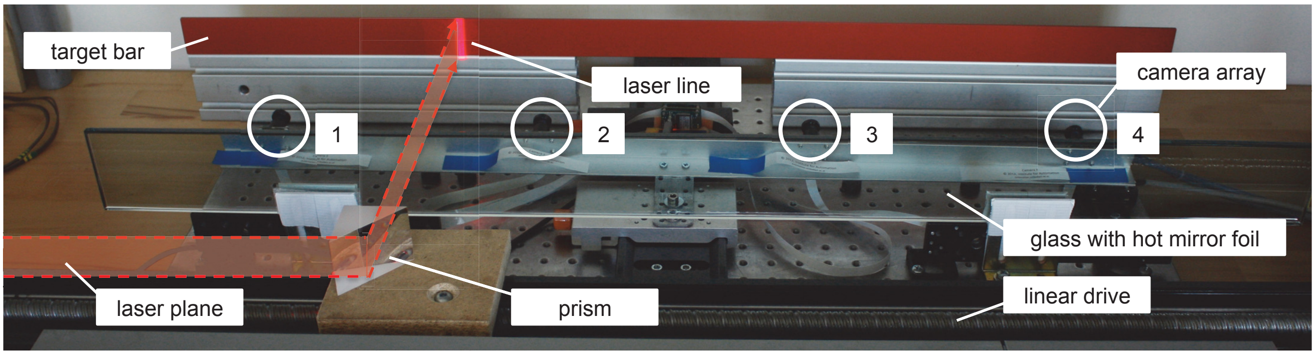


Fig. 10. The prototype PSD, showing the four cameras, the glass window with coated hot mirror foil and the sand-blasted red anodized aluminum target bar. A prism mounted on a linear drive's stage redirects the laser plane generated by the construction laser to cover the whole measurement range of 1 m for system calibration as well as cross validation experiments. The linear drive is composed of two parts: the controller *iTK Pollux Drive Typ-3* [27] and the linear axis *THK KR5520A+1280LH0-0000* [28], featuring an usable length of 1.1 m an accuracy of ±0.1 mm.

1) hot mirror foil has been attached to the glass window;
2) only the red channels of the RGB sensors are used;
3) the aluminum target bar is anodized with red color.

III. Analytic Algebraic Model

Following tasks need to be solved to enable consistent calibration and measurement processes with estimations for the confidence bounds:

- An algorithm for the synthesis of discrete orthogonal polynomials (DOP) is presented.
- In each acquired image, the laser line's position and its uncertainty is estimated, enabling the usage of uncertainty weighted DOP for the laser line's approximation.
- The system is calibrated using the information of the laser lines' positions and uncertainties in each camera's field of view with the weighted DOP approach. The computed coefficients are saved for analytic interpolation.
- A numeric and an analytic solution for the measurement procedure is presented. The measurement results of the individual cameras are merged as a linear operation, i.e. the confidence envelopes are determinable.

A. Synthesis of Discrete Orthogonal Polynomials

Gram [29] proposed what is now known as the Gram-Schmidt orthogonalization process to generate orthogonal polynomials [30]. The Gram-Schmidt process is, however, numerically unstable [31, Chapter 5] and will only deliver polynomials up to degree of $d \approx 20$ without significant error. Considerable research has been performed on discrete polynomials and their synthesis [32]–[39]. Nevertheless, none of these papers present a method which is capable of synthesizing discrete orthogonal polynomials of high quality for arbitrary nodes located within the unit circle on the complex plane. Here, a Lanczos process with complete reorthogonalization [31, Chapter 9], [40] is used to synthesize the polynomials. The procedure can be summarized as follows: Given a vector $\boldsymbol{x}$ of m nodes with mean $\bar{x}$: the two Gram basis functions $\boldsymbol{g}_0$, $\boldsymbol{g}_1$ are computed,

$$\boldsymbol{g}_0 = \boldsymbol{1}/\sqrt{m}, \qquad \boldsymbol{g}_1 = \frac{\boldsymbol{x} - \bar{x}}{\| \boldsymbol{x} - \bar{x} \|_2} \quad \text{and} \quad \mathsf{G} = [\boldsymbol{g}_0, \boldsymbol{g}_1] \,. \tag{1}$$

The remaining polynomials are synthesized by repeatedly performing the following computations:

1) Compute the polynomial of the next higher degree, where $\circ$ is the Hadamard product,

$$\boldsymbol{g}_m = \boldsymbol{g}_1 \circ \boldsymbol{g}_{m-1} \tag{2}$$

2) and perform a complete reorthogonalization,

$$\begin{aligned} \boldsymbol{g}_m &= \boldsymbol{g}_m - \mathsf{G}\,\mathsf{G}^{\mathrm{T}} \boldsymbol{g}_m, & (3)\\ &= \left\{ \mathsf{I} - \mathsf{G}\,\mathsf{G}^{\mathrm{T}} \right\} \boldsymbol{g}_m & (4) \end{aligned}$$

by projection onto the orthogonal complement of all previously synthesized polynomials[1].

3) Normalize the vector,

$$\boldsymbol{g}_m = \frac{\boldsymbol{g}_m}{\| \boldsymbol{g}_m \|_2} \tag{5}$$

4) and augment the matrix of basis functions G,

$$\mathsf{G} = [\mathsf{G}, \boldsymbol{g}_m] \,. \tag{6}$$

This procedure[2] yields a set of orthonormal polynomials up to degree $d = 1000$ without error, from a set of arbitrary nodes located within the unit circle on the complex plane. However, in this article only real nodes are used. Although in [41] the Lanczos process is used to compute discrete orthogonal polynomials, the authors seem to have overseen the possibility (necessity) of using complete reorthogonalization at each step of the polynomial synthesis.

B. Approximation of a Laser Line

Each camera delivers a region of interest C corresponding to the view of the target with m rows and n columns, whereby i is the row index and j is the column index, i.e. a pixel's illumination value is c_{ij}. The first moment x_i of each row i is computed by

$$x_i = \frac{\sum_{j=1}^{n} j c_{ij}}{\sum_{j=1}^{n} c_{ij}}, \tag{7}$$

[1] It is important to note that the reorthogonalization is w.r.t. the complete set of basis functions, not just the previous polynomial.

[2] A toolbox to implement this procedure is available online http://www.mathworks.com/matlabcentral/fileexchange/41250.

hence

$$\boldsymbol{x} = [x_1, \ldots, x_m]^{\mathrm{T}} \quad \text{in} \quad \text{pixel}. \tag{8}$$

The variance σ_i^2 of each row i is approximated by the second moment around the measured values $\boldsymbol{x}$

$$\sigma_i^2 = \sum_{j=1}^{n} c_{ij}(j - x_i)^2, \tag{9}$$

hence

$$\boldsymbol{\sigma}^2 = \left[\sigma_1^2, \ldots, \sigma_m^2\right]^{\mathrm{T}} \quad \text{in} \quad \text{pixel}^2. \tag{10}$$

Fig. 11 shows an example. The diagonal covariance matrix Σ is defined through the values of $\boldsymbol{\sigma}^2$:

$$\Sigma = \operatorname{diag}\left(\boldsymbol{\sigma}^2\right) = \operatorname{diag}\left(\sigma_1^2, \ldots, \sigma_m^2\right). \tag{11}$$

The cost function K of the weighted least squares polynomial approximation is

$$K = \|\mathsf{W}(\boldsymbol{x} - \boldsymbol{y})\|_2^2\,, \tag{12}$$

whereby the laser's weighting matrix W is designed as the inverse of standard deviations, i.e.

$$\mathsf{W} = \Sigma^{-\frac{1}{2}}. \tag{13}$$

The term $(\boldsymbol{x} - \boldsymbol{y})$ represents the residuum between the data points $\boldsymbol{x}$ and the approximated polynomial $\boldsymbol{y}$ and the term is minimized for a maximum likelihood estimation [42], whereby

$$\boldsymbol{y} = \mathsf{G}\boldsymbol{\alpha}. \tag{14}$$

The set of discrete orthogonal Gram polynomial basis functions G has a degree $d = 2$ and $\boldsymbol{\alpha}$ is the coefficient vector [43]. Expanding Eqn. 12 yields

$$\begin{aligned} K(\boldsymbol{\alpha}) &= \left\|\Sigma^{-\frac{1}{2}}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})\right\|_2^2 \\ &= \left[\Sigma^{-\frac{1}{2}}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})\right]^{\mathrm{T}}\left[\Sigma^{-\frac{1}{2}}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})\right] \\ &= (\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})^{\mathrm{T}}\left(\Sigma^{-\frac{1}{2}}\right)^{\mathrm{T}}\left(\Sigma^{-\frac{1}{2}}\right)(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha}) \\ &= (\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha}). \end{aligned} \tag{15}$$

Minimization is performed by

$$\frac{\partial K(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = \frac{\partial}{\partial \boldsymbol{\alpha}}\left[(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})\right] = \boldsymbol{0}. \tag{16}$$

Note, that taking the derivative $K(\boldsymbol{\alpha})$ w.r.t. $\boldsymbol{\alpha}$ delivers a coordinate vector $\boldsymbol{e}_i$ for each measurement value x_i, whereby

$$\frac{\partial \boldsymbol{\alpha}}{\partial \alpha_i} = \boldsymbol{e}_i = [0, \ldots, 0, 1, 0, \ldots, 0]^{\mathrm{T}}, \tag{17}$$

i.e. the i^{th} element is 1 and the other elements are 0. This formalism conserves the correct matrix dimensions. Performing the differentiation in Eqn. 16 for an individual x_i returns

$$\begin{aligned} 0 = &-(\mathsf{G}\boldsymbol{e}_i)^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha}) \\ &-(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha})^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{e}_i. \end{aligned} \tag{18}$$

Expansion of both terms yields

$$\begin{aligned} 0 = &-(\mathsf{G}\boldsymbol{e}_i)^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x} + (\mathsf{G}\boldsymbol{e}_i)^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha} \\ &-\boldsymbol{x}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{e}_i + (\mathsf{G}\boldsymbol{\alpha})^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{e}_i \end{aligned} \tag{19}$$

and furthermore

$$\begin{aligned} 0 = &-\boldsymbol{e}_i^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x} + \boldsymbol{e}_i^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha} \\ &-\boldsymbol{x}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{e}_i + \boldsymbol{\alpha}^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{e}_i. \end{aligned} \tag{20}$$

The terms have dimensions of (1×1), i.e. each one describes a scalar and can therefore be transposed in order to simplify the equation.

$$0 = \boldsymbol{e}_i^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x} - \mathsf{G}\boldsymbol{\alpha}). \tag{21}$$

Consequently, the equation in matrix form is

$$\boldsymbol{e}_i^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x} = \boldsymbol{e}_i^{\mathrm{T}}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha}. \tag{22}$$

Stacking the results for each x_i delivers

$$\begin{bmatrix} \boldsymbol{e}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}_i^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}_m^{\mathrm{T}} \end{bmatrix}(\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{e}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}_i^{\mathrm{T}} \\ \vdots \\ \boldsymbol{e}_m^{\mathrm{T}} \end{bmatrix}(\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha}), \tag{23}$$

whereby each stack of element vectors is the identity matrix I. This leads to the normal equations

$$\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x} = \mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\boldsymbol{\alpha}. \tag{24}$$

Solving for the coefficient vector $\boldsymbol{\alpha}$ yields

$$\boldsymbol{\alpha} = \left(\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\right)^{+}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x}, \tag{25}$$

whereby $+$ denotes the Moore-Penrose pseudo inverse. Utilizing this result for the polynomial approximation

$$\boldsymbol{y} = \mathsf{G}\boldsymbol{\alpha} = \mathsf{G}\left(\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\right)^{+}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\boldsymbol{x} \tag{26}$$

generates the weighted least squares transformation matrix L

$$\mathsf{L} \triangleq \mathsf{G}\left(\mathsf{G}^{\mathrm{T}}\Sigma^{-1}\mathsf{G}\right)^{+}\mathsf{G}^{\mathrm{T}}\Sigma^{-1}. \tag{27}$$

The linear operation is derived by substituting L from Eqn. 27 into Eqn. 26

$$\boldsymbol{y} = \mathsf{G}\boldsymbol{\alpha} \triangleq \mathsf{L}\boldsymbol{x}. \tag{28}$$

The measurement's result is the value p of the polynomial's $\boldsymbol{y}$ midpoint, i.e. it is only necessary to compute this relevant point using the appropriate portion $\boldsymbol{l}^{\mathrm{T}}$ of the transformation matrix L.

$$p = \boldsymbol{e}_t^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{e}_t^{\mathrm{T}}\mathsf{L}\boldsymbol{x} = \boldsymbol{l}^{\mathrm{T}}\boldsymbol{x}, \tag{29}$$

whereby $\boldsymbol{e}_t$ is the coordinate vector at $t = \frac{m-1}{2} + 1$ when m is odd. The point p now represents the laser line's camera coordinates in pixel. The covariance matrix $\hat{\Sigma}$ for the vector $\boldsymbol{y}$ is by definition

$$\hat{\Sigma} = \mathbb{E}\left[(\boldsymbol{y} - \mathbb{E}[\boldsymbol{y}])(\boldsymbol{y} - \mathbb{E}[\boldsymbol{y}])^{\mathrm{T}}\right], \tag{30}$$

where $\mathbb{E}[\boldsymbol{y}]$ is the expected value of $\boldsymbol{y}$ [44]. If $\boldsymbol{y} = \mathsf{L}\boldsymbol{x}$, then

$$\begin{aligned} \hat{\Sigma} &= \mathbb{E}[(\mathsf{L}\boldsymbol{x} - \mathbb{E}[\mathsf{L}\boldsymbol{x}])(\mathsf{L}\boldsymbol{x} - \mathbb{E}[\mathsf{L}\boldsymbol{x}])^{\mathrm{T}}] \\ &= \mathbb{E}[(\mathsf{L}\boldsymbol{x} - \mathsf{L}\,\mathbb{E}[\boldsymbol{x}])(\mathsf{L}\boldsymbol{x} - \mathsf{L}\,\mathbb{E}[\boldsymbol{x}])^{\mathrm{T}}] \\ &= \mathsf{L}\,\mathbb{E}[(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}])^{\mathrm{T}}]\,\mathsf{L}^{\mathrm{T}} \\ &= \mathsf{L}\Sigma\mathsf{L}^{\mathrm{T}}. \end{aligned} \tag{31}$$

Based on the matrix $\hat{\Sigma}$, the uncertainty, i.e. variance $\hat{\boldsymbol{\sigma}}^2$ of each fitted curve $\boldsymbol{y}$, is directly determinable:

$$\hat{\boldsymbol{\sigma}}^2 = \operatorname{diag}(\hat{\Sigma}) = (\mathsf{L} \circ \mathsf{L})\boldsymbol{\sigma}^2. \tag{32}$$

The variance λ^2 of a relevant point p is then

$$\lambda^2 = \boldsymbol{e}_t^{\mathrm{T}} \hat{\boldsymbol{\sigma}}^2. \tag{33}$$

Concatenating these single points for each laser line in an individual camera's field of view delivers the vectors

$$\boldsymbol{p} \quad \text{and} \quad \boldsymbol{\lambda}^2, \tag{34}$$

which are required for system calibration.

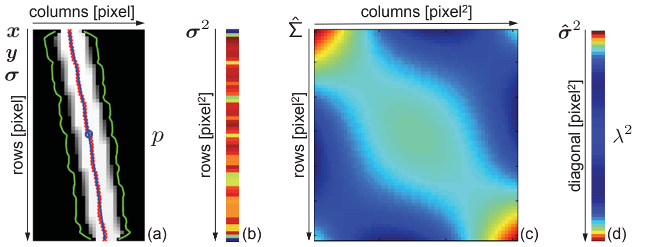


Fig. 11. An example image of a laser line with the first moment $\boldsymbol{x}$ (red), its standard deviation $\boldsymbol{\sigma}$ (green) and polynomial approximation $\boldsymbol{y}$ (blue) is shown in (a) along with the relevant point p. The shown experimental data delivers a standard deviation of approximately 10 mm. The variance $\boldsymbol{\sigma}^2$ of the local position of the line is illustrated in (b). The covariance matrix $\hat{\Sigma}$ of the complete polynomial fit has been analytically computed from the experimental data and it is presented in (c) along with the variance $\hat{\boldsymbol{\sigma}}^2$ and in λ^2 (d) of the polynomial estimate.

C. System Calibration via Polynomial Approximation

The overall system uses $k = 1, \dots, 4$ sensors in this specific implementation. The individually acquired camera space coordinates are then

$$\{\boldsymbol{p}_{(1)}, \boldsymbol{p}_{(2)}, \boldsymbol{p}_{(3)}, \boldsymbol{p}_{(4)}\} \quad \text{in} \quad \text{pixel} \tag{35}$$

along with their variances

$$\{\boldsymbol{\lambda}^2_{(1)}, \boldsymbol{\lambda}^2_{(2)}, \boldsymbol{\lambda}^2_{(3)}, \boldsymbol{\lambda}^2_{(4)}\} \quad \text{in} \quad \text{pixel}^2. \tag{36}$$

They are combined with the real space coordinates in $\boldsymbol{q}$, i.e. the resulting calibration points in Fig. 12. The device's complete measurement range of 1 m is calibrated using $n = 101$ equidistant calibration points, i.e.

$$\boldsymbol{q} = [q_1, \dots, q_n]^{\mathrm{T}} = [q_1, \dots, q_{101}]^{\mathrm{T}}, \tag{37}$$

which is equivalent to a calibration spacing of 10 mm. The main source of error and its corresponding uncertainty is related to the laser line's width. One standard deviation is approximately 10 mm at minimum measurement distance. Therefore, no further gain in accuracy can be achieved by reducing the step spacing. Increasing the spacing will lead to an increase of the uncertainty of the calibration and is considered undesirable. The position of the points ensures that each camera k has $m_{(k)} > n/3$ points for calibration, i.e. here a minimum of approximately $m_{(k)} \approx 33$ were used. The coefficients $\boldsymbol{\beta}_{(k)}$ for the polynomial $\boldsymbol{z}_{(k)}$ of the k^{th} camera are determined using a weighted least squares approximation by minimizing the according cost function $H_{(k)}$:

$$H_{(k)} = \left\| \mathsf{V}_{(k)} (\boldsymbol{p}_{(k)} - \boldsymbol{z}_{(k)}) \right\|_2^2, \tag{38}$$

which can be solved as shown for Eqn. 12. The system's weighting matrix $\mathsf{V}_{(k)}$ and the covariance matrix $\Lambda_{(k)}$ are

$$\mathsf{V}_{(k)} = \Lambda_{(k)}^{-\frac{1}{2}} \quad \text{and} \quad \Lambda_{(k)} = \operatorname{diag}\left(\boldsymbol{\lambda}^2_{(k)}\right). \tag{39}$$

A single polynomial is defined as

$$\boldsymbol{z}_{(k)} = \mathsf{G}\boldsymbol{\beta}_{(k)}. \tag{40}$$

Optical distortions are best modeled by even degree polynomials, therefore the matrix G represents a Gram basis set of degree $d = 6$. The k^{th} coefficient vector $\boldsymbol{\beta}_{(k)}$ is then

$$\boldsymbol{\beta}_{(k)} = \left(\mathsf{G}^{\mathrm{T}} \Lambda_{(k)}^{-1} \mathsf{G}\right)^{+} \mathsf{G}^{\mathrm{T}} \Lambda_{(k)}^{-1} \boldsymbol{p}_{(k)}. \tag{41}$$

The k^{th} transformation matrix $\mathsf{M}_{(k)}$ is

$$\mathsf{M}_{(k)} \triangleq \mathsf{G}\left(\mathsf{G}^{\mathrm{T}} \Lambda_{(k)}^{-1} \mathsf{G}\right)^{+} \mathsf{G}^{\mathrm{T}} \Lambda_{(k)}^{-1}, \tag{42}$$

consequently leading to

$$\boldsymbol{z}_{(k)} = \mathsf{G}\boldsymbol{\beta}_{(k)} \triangleq \mathsf{M}_{(k)} \boldsymbol{p}_{(k)}. \tag{43}$$

As shown in Eqn. 30, the covariance matrix $\hat{\Lambda}_{(k)}$ of the fitted polynomial $\boldsymbol{z}_{(k)}$ is computable by

$$\hat{\Lambda}_{(k)} = \mathsf{M}_{(k)} \Lambda_{(k)} \mathsf{M}_{(k)}^{\mathrm{T}} \tag{44}$$

and therefore each calibration curve's, $\boldsymbol{z}_{(k)}$, variance $\boldsymbol{\omega}^2_{(k)}$ is the diagonal of the covariance mattrix $\hat{\Lambda}_{(k)}$

$$\boldsymbol{\omega}^2_{(k)} = \operatorname{diag}(\hat{\Lambda}_{(k)}) = (\mathsf{M}_{(k)} \circ \mathsf{M}_{(k)}) \boldsymbol{\lambda}^2_{(k)}. \tag{45}$$

The calibration process yields four individual sets of calibration curves

$$\{\boldsymbol{z}_{(1)}, \boldsymbol{z}_{(2)}, \boldsymbol{z}_{(3)}, \boldsymbol{z}_{(4)}\} \quad \text{in} \quad \text{mm} \tag{46}$$

and they are visualized in Fig. 13 along with their variances

$$\{\boldsymbol{\omega}^2_{(1)}, \boldsymbol{\omega}^2_{(2)}, \boldsymbol{\omega}^2_{(3)}, \boldsymbol{\omega}^2_{(4)}\} \quad \text{in} \quad \text{mm}^2. \tag{47}$$

The confidence interval/standard deviation is

$$\boldsymbol{\omega}_{(k)} = \sqrt{\boldsymbol{\omega}^2_{(k)}} \quad \text{in} \quad \text{mm}, \tag{48}$$

the individual curves are shown as weighting functions in Fig. 14. The resulting covariance matrices are then

$$\{\hat{\Lambda}_{(1)}, \hat{\Lambda}_{(2)}, \hat{\Lambda}_{(3)}, \hat{\Lambda}_{(4)}\} \quad \text{in} \quad \text{mm}^2 \tag{49}$$

and they are visualized in Fig. 15. During measurement, either numeric or analytic polynomial interpolation is utilized to determine the mapping $\boldsymbol{z}_{(k)}$ for intermediate points $\tilde{\boldsymbol{z}}_{(k)}$. Exact details of the least squares fitting an interpolation can be found in [43]. Note, that this derivation is completely general for an arbitrary number of sensors.

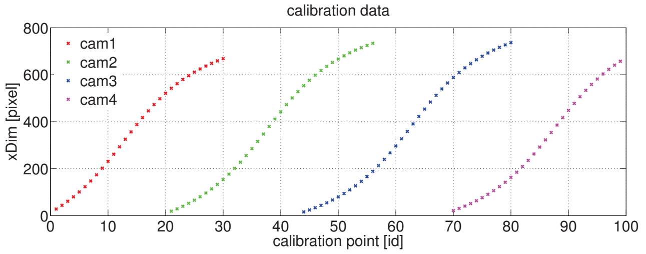


Fig. 12. The real space coordinate vector $\boldsymbol{q}$ features a 10 mm spacing, resulting in $n = 101$ equally spaced calibration points. The initial image acquisition and preprocessing yields $m_{(k)} > n/3 \approx 33$ individual calibration points for each of the $k = 1, \ldots, 4$ cameras.

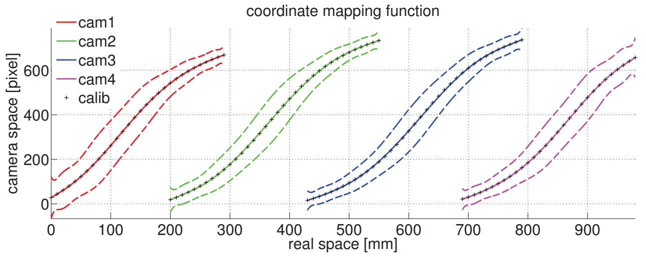


Fig. 13. From left to right are the four individual camera calibration curves $\{\boldsymbol{z}_{(1)}, \boldsymbol{z}_{(2)}, \boldsymbol{z}_{(3)}, \boldsymbol{z}_{(4)}\}$, which map the respective pixel coordinates to real world coordinates $\boldsymbol{q}$ in mm. The standard deviations $\{\boldsymbol{\omega}_{(1)}, \boldsymbol{\omega}_{(2)}, \boldsymbol{\omega}_{(3)}, \boldsymbol{\omega}_{(4)}\}$ have been scaled by a factor of 35 for better visibility.

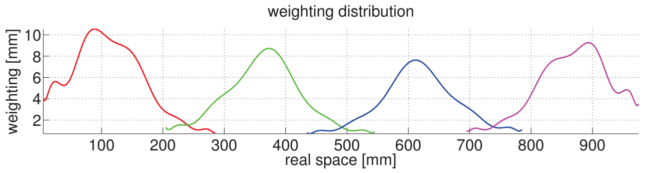


Fig. 14. The standard deviations $\{\boldsymbol{\omega}_{(1)}, \boldsymbol{\omega}_{(2)}, \boldsymbol{\omega}_{(3)}, \boldsymbol{\omega}_{(4)}\}$ can be represented as weighting functions. They share some common characteristics with membership functions, however these curves have been derived analytically without any subjective assumptions.

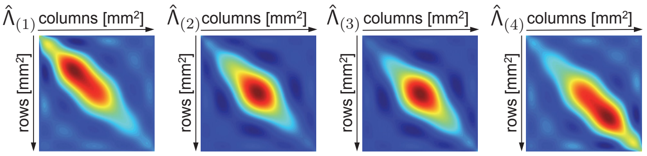


Fig. 15. The illustration shows the covariance matrices $\{\hat{\Lambda}_{(1)}, \hat{\Lambda}_{(2)}, \hat{\Lambda}_{(3)}, \hat{\Lambda}_{(4)}\}$. The asymmetry of $\{\hat{\Lambda}_{(1)}, \hat{\Lambda}_{(4)}\}$ is caused by the optical conditions at the borders of the measurement range. Note, that $\boldsymbol{\omega}^2_{(k)}$ is the diagonal of the covariance matrix $\operatorname{diag}(\hat{\Lambda}_{(k)})$, i.e. $\boldsymbol{\omega}^2_{(k)} = \operatorname{diag}(\hat{\Lambda}_{(k)})$.

D. Measurement via Polynomial Interpolation

The measurement is performed for each camera k, therefore the notation is simplified in this section. During calibration, the Gram polynomials are synthesized for $m = m_{(k)}$ known p values from the vector $\boldsymbol{p} = \boldsymbol{p}_{(k)}$ and the corresponding coefficients $\boldsymbol{\beta} = \boldsymbol{\beta}_{(k)}$ are determined for each camera's curve $\boldsymbol{z} = \boldsymbol{z}_{(k)}$.

1) Numerical Solution: The numerical solution for interpolation is based on the introduction of a resolution factor $\rho > 1$ and $\rho \in \mathbb{N}$ during the synthesis of the DOP basis G. The number of used nodes $\tilde{m}$ for the measurement's interpolation is higher than the number of known nodes m for the calibration's approximation, i.e. $\tilde{m} = \rho\, m$. The submatrix $\bar{\mathsf{G}}$ is generated by extracting the known portions of G, i.e. the matching calibration points at $\boldsymbol{q}$. In Matlab®, the syntax corresponding to the process would be

$$\bar{\mathsf{G}} = \mathsf{G}(\boldsymbol{q}, :). \tag{50}$$

The linear operator M from Eqn. 42 is adapted as following

$$\tilde{\mathsf{M}} \triangleq \mathsf{G}\left(\bar{\mathsf{G}}^{\mathrm{T}} \Lambda^{-1} \bar{\mathsf{G}}\right)^{+} \bar{\mathsf{G}}^{\mathrm{T}} \Lambda^{-1}, \tag{51}$$

whereby the dimensions of the matrices are $(\tilde{m} \times m)$ for $\tilde{\mathsf{M}}$, $(\tilde{m} \times \tilde{m})$ for G, $(m \times \tilde{m})$ for $\bar{\mathsf{G}}$ and $(m \times m)$ for Λ^{-1}. The resulting curves $\tilde{\boldsymbol{z}} = \tilde{\mathsf{M}}\boldsymbol{p}$ are stored in the form of lookup tables with a real space resolution of 1 mm, when $\rho = 10$ and the original calibration spacing was 10 mm. The detected laser line position in pixel is mapped to the nearest neighbor element $\tilde{z}$ in the lookup table $\tilde{\boldsymbol{z}}$. The approach has a runtime of $\mathcal{O}(\tilde{m}^2)$, implementing a binary search tree structure improves the computational efficiency and lowers the run-time to $2\,\mathcal{O}(\tilde{m}\log_2(\tilde{m}))$.

2) Analytic Solution: The synthesis process (see Section III-A) yields the coefficient vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ as required for the three term relationship [40],

$$g_m(p) = \gamma_{m-1}\, p\, g_{m-1}(p) - \delta_{m-1}\, g_{m-2}(p). \tag{52}$$

This recurrence is used during measurement to compute the Gram polynomials at a point $\tilde{p}$ between the nodes, i.e. between the calibration points $\boldsymbol{p}$. The vector of polynomial values at a point p is formed by

$$\boldsymbol{g} = \boldsymbol{g}(p) = [g_m(p) \ldots g_0]^{\mathrm{T}}. \tag{53}$$

Now given the weighted polynomial coefficient vector $\boldsymbol{\beta}$ from the calibration process, the corresponding $\tilde{z}$ value is computed by using $\tilde{p}$ as the function's parameter,

$$\tilde{z} = z(\tilde{p}) = \tilde{\boldsymbol{g}}^{\mathrm{T}}\boldsymbol{\beta} \quad \text{and} \quad \tilde{\boldsymbol{g}} = \boldsymbol{g}(\tilde{p}), \tag{54}$$

whereby $\tilde{z}$ is a single mapped point in mm. The recurrence relationship enables interpolation between the nodes up to degree of $d \approx 35$ without error. The process is linear; consequently, given the covariance Λ_β of the coefficient vector $\boldsymbol{\beta}$, the covariance Λ_z of the calibration coefficients $\boldsymbol{z}$ can be computed for the interpolation as

$$\Lambda_z = \boldsymbol{g}\, \Lambda_\beta\, \boldsymbol{g}^{\mathrm{T}}. \tag{55}$$

In comparison to the numerical approach, the computation's runtime is linear w.r.t. the degree d of the interpolated polynomial, i.e. $\mathcal{O}(d)$. Thus, the computation is faster and more accurate.

E. Algebraic Data Fusion

In neighboring cameras A and B with overlapping fields of view, the calibration delivers two results $\boldsymbol{z}_{(A)}$ and $\boldsymbol{z}_{(B)}$ for the same positions, each with its variance $\boldsymbol{\omega}^2_{(A)}$ and $\boldsymbol{\omega}^2_{(B)}$, see an example in Fig. 16. An uncertainty weighted average is applied to fuse both results into a single curve with superior accuracy than the individual values. A curve $\boldsymbol{z}_{(k)}$ is composed of three parts:

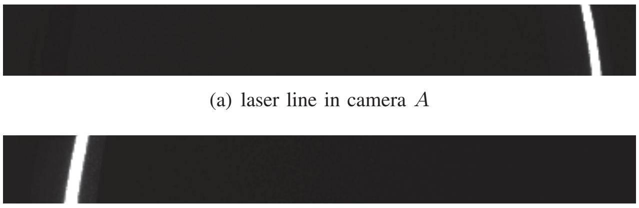
1) $\boldsymbol{z}_{(k,l)}$, the left overlapping section;
2) $\boldsymbol{z}_{(k,m)}$, the middle section without overlap and;
3) $\boldsymbol{z}_{(k,r)}$, the right overlapping section.

This means the vector can be concatenated as

$$\boldsymbol{z}_{(k)} = \left[\boldsymbol{z}_{(k,l)}, \boldsymbol{z}_{(k,m)}, \boldsymbol{z}_{(k,r)}\right]^{\mathrm{T}}, \tag{56}$$

and the same is true for the corresponding variance vectors

$$\boldsymbol{\omega}^2_{(k)} = \left[\boldsymbol{\omega}^2_{(k,l)}, \boldsymbol{\omega}^2_{(k,m)}, \boldsymbol{\omega}^2_{(k,r)}\right]^{\mathrm{T}}. \tag{57}$$



(a) laser line in camera A

(b) laser line in camera B

Fig. 16. The physical position of the laser line on the target bar is simultaneously imaged by two neighboring cameras A and B when it's located in the overlapping fields of view.

The number of elements is usually $m_{(A)} \neq m_{(B)}$, but the number or right nodes $m_{(A,r)}$ in the left section A is equal to the number of left nodes $m_{(B,l)}$ in the right section B, i.e. $m_{(A,r)} = m_{(B,l)}$. Note, that A is assumed to be the left boundary section and therefore has no left overlapping nodes $\boldsymbol{z}_{(A,l)}$. The vectors $\boldsymbol{z}_{(A,r)}$ and $\boldsymbol{z}_{(B,l)}$ are fused by a weighted average based on the inverse of their standard deviations $\boldsymbol{\omega}_{(A,r)}$ and $\boldsymbol{\omega}_{(B,l)}$.

$$\boldsymbol{z}_{(AB)} = \Omega^{-1}_{(AB)}\left(\Omega^{-1}_{(A,r)}\boldsymbol{z}_{(A,r)} + \Omega^{-1}_{(B,l)}\boldsymbol{z}_{(B,l)}\right), \tag{58}$$

whereby

$$\Omega_{(AB)} = \operatorname{diag}\left(\boldsymbol{\omega}_{(A,r)} + \boldsymbol{\omega}_{(B,l)}\right) \tag{59}$$

as well as

$$\Omega_{(A,r)} = \operatorname{diag}\left(\boldsymbol{\omega}_{(A,r)}\right), \tag{60}$$

$$\Omega_{(B,l)} = \operatorname{diag}\left(\boldsymbol{\omega}_{(B,l)}\right). \tag{61}$$

The combined weighting matrices, i.e. the fusion matrices, are

$$\mathsf{F}_{(AB)} \triangleq \Omega^{-1}_{(AB)}\Omega^{-1}_{(A,r)}, \tag{62}$$

$$\mathsf{F}_{(BA)} \triangleq \Omega^{-1}_{(AB)}\Omega^{-1}_{(B,l)}. \tag{63}$$

respectively. Eqn. 58 can then be rewritten as

$$\boldsymbol{z}_{(AB)} = \mathsf{F}_{(AB)}\boldsymbol{z}_{(A,r)} + \mathsf{F}_{(BA)}\boldsymbol{z}_{(B,l)}. \tag{64}$$

The fused calibration curve $\bar{\boldsymbol{z}}$ is then

$$\begin{bmatrix} \boldsymbol{z}_{(A,m)} \\ \boldsymbol{z}_{(AB)} \\ \boldsymbol{z}_{(B,m)} \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathsf{I} & 0 & 0 & 0 & \dots \\ 0 & \mathsf{F}_{(AB)} & \mathsf{F}_{(BA)} & 0 & \dots \\ 0 & 0 & 0 & \mathsf{I} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_{(A,m)} \\ \boldsymbol{z}_{(A,r)} \\ \boldsymbol{z}_{(B,l)} \\ \boldsymbol{z}_{(B,m)} \\ \vdots \end{bmatrix} \tag{65}$$

and in concise form

$$\bar{\boldsymbol{z}} = \mathsf{F}\left[\boldsymbol{z}_{(A)}, \boldsymbol{z}_{(B)}, \dots\right]^{\mathrm{T}}. \tag{66}$$

The approach forms a mathematical framework for the uncertainty weighted data fusion of an arbitrary number of sensors. The model delivers the result in Eqn. 68 for the specific solution utilizing $k = 1, \dots, 4$ cameras. The complete fused curve $\bar{\boldsymbol{z}}$ for this application is then

$$\bar{\boldsymbol{z}} = \mathsf{F}\left[\boldsymbol{z}_{(1)}, \boldsymbol{z}_{(2)}, \boldsymbol{z}_{(3)}, \boldsymbol{z}_{(4)}\right]^{\mathrm{T}}. \tag{67}$$

IV. Experimental Verification

A. Achievable Accuracy

The relationship between pixel dimensions, i.e. resolution, and the error in detecting the position of a sharp edge in an image was investigated in [45], [46]. It was determined, that a subpixel accuracy of a factor $10, \dots, 20$ can be achieved reliably. In this experimental setup, 1 pixel in the camera corresponds to approximately 0.4 mm in the real space, i.e. with subpixeling an accuracy of at least 0.04 mm is theoretically possible. The observed experimental results in Table I gave a 95 % confidence interval of $< \pm 0.6$ mm. It is concluded, that the camera's resolution is not the limiting factor in this system. The primary uncertainty is the illumination associated with the laser, i.e. small variations in the width of the laser

$$\begin{bmatrix} \boldsymbol{z}_{(1)} \\ \boldsymbol{z}_{(12)} \\ \boldsymbol{z}_{(2)} \\ \boldsymbol{z}_{(23)} \\ \boldsymbol{z}_{(3)} \\ \boldsymbol{z}_{(34)} \\ \boldsymbol{z}_{(4)} \end{bmatrix} = \begin{bmatrix} \mathsf{I} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathsf{F}_{(12)} & \mathsf{F}_{(21)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathsf{I} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathsf{F}_{(23)} & \mathsf{F}_{(32)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathsf{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathsf{F}_{(34)} & \mathsf{F}_{(43)} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathsf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_{(1,m)} \\ \boldsymbol{z}_{(1,r)} \\ \boldsymbol{z}_{(2,l)} \\ \boldsymbol{z}_{(2,m)} \\ \boldsymbol{z}_{(2,r)} \\ \boldsymbol{z}_{(3,l)} \\ \boldsymbol{z}_{(3,m)} \\ \boldsymbol{z}_{(3,r)} \\ \boldsymbol{z}_{(4,l)} \\ \boldsymbol{z}_{(4,m)} \end{bmatrix} \tag{68}$$

line and the distribution of intensities within the generated line. Furthermore, testing on retro-reflector-targets and PSDs in outdoor turbulent environments [47] has shown, that accuracies of approximately 10 mm can be achieved over distances of 300 m. This indicates that the instrument's required accuracy can be accomplished even in the presence of ambient disturbances.

B. Cross Validation of the Measurement

The cross validation experiments were performed under laboratory conditions with different glass window configurations. The 100 measurement points $\tilde{\boldsymbol{q}}$ are located between the 101 equidistant calibration points $\boldsymbol{q}$, i.e. where the largest error is to be expected. The norm of the measurement's standard deviation s is computed as a metric of quality. The resulting measurement standard deviation s_w is compared to a conventional, i.e. unweighted, fit's standard deviation s_c to verify the methodology. The conventional approach utilizes the mean $\overline{\sigma^2}$ and $\overline{\lambda^2}$ of the variances $\boldsymbol{\sigma}^2$ and $\boldsymbol{\lambda}^2$.

TABLE I
THE CROSS VALIDATION RESULTS IN A 95 % CONFIDENCE INTERVAL FOR THE STANDARD DEVIATION OF THE WEIGHTED (s_w) AND CONVENTIONAL (s_c) POLYNOMIAL APPROXIMATION ARE SHOWN AS ABSOLUTE AND RELATIVE (TO THE COMPLETE MEASUREMENT RANGE) VALUES.

approximation	weighted	conventional
laser's weighting system's weighting	$\mathsf{W} = \Sigma^{-\frac{1}{2}}$ $\mathsf{V} = \Lambda^{-\frac{1}{2}}$	$\mathsf{W} = \overline{\sigma^2}\mathsf{I}$ $\mathsf{V} = \overline{\lambda^2}\mathsf{I}$
standard deviation	s_w	s_c
absolut: no glass Silver 20 LX-70	 ±0.507 mm ±0.552 mm ±0.532 mm	 ±0.594 mm ±0.728 mm ±0.608 mm
relativ: no glass Silver 20 LX-70	 ±0.051 % ±0.055 % ±0.053 %	 ±0.059 % ±0.073 % ±0.061 %

The results in Table I prove, that the weighted polynomial approximation is up to 24 % better than its conventional pendant when using the hot mirror foil *LX-70*. It establishes a better measurement environment w.r.t. ambient light than the hot mirror foil *Silver 20*. The error plot of the *LX-70* setup is shown in Fig. 17; the norms of the resulting error vectors $\boldsymbol{\epsilon}_w$ (weighted) and $\boldsymbol{\epsilon}_c$ (conventional) are

$$\|\boldsymbol{\epsilon}_w\|_2 = 2.5\,\text{mm} \quad \text{and} \quad \|\boldsymbol{\epsilon}_c\|_2 = 3.0\,\text{mm}.$$

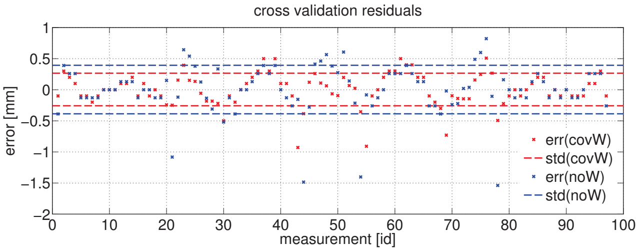


Fig. 17. The standard deviation of the uncertainty weighted regression (red) is smaller than the standard deviation of the conventional polynomial regression (blue). A glass window coated with *LX-70* was used for the measurement.

C. Remapping and Linearization Error

Merging the calibration curves of all cameras in Fig. 13 into a combined output yields the fused mapping function. Removing the linear portion delivers the linearization error in Fig. 18. The graph demonstrates the suitability of the mathematical model. The reduced standard deviation at the overlapping regions is an indicator of the superior measurement quality in this sections.

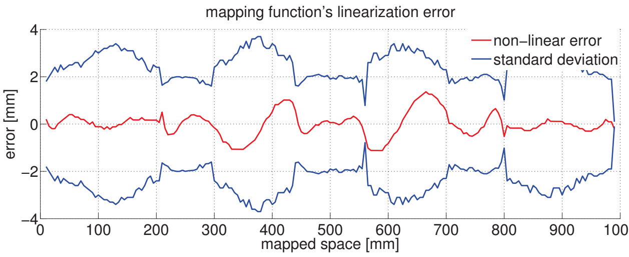


Fig. 18. The linearization error (red) of the combined curve is shown. The two central measurement ranges possess a systematic error. The measurement becomes more accurate in the overlapping fields of view, which is indicated by reduced standard deviations (blue).

D. Kolmogorov-Smirnov Test for Gaussian Distribution

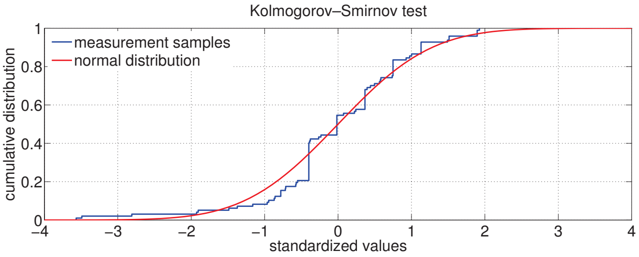


Fig. 19. The KS-test checks the goodness of fit between one-dimensional empirical data (blue) and its expected reference continuous parent distribution (red). The test quantifies the distance between both cumulative distribution functions.

The normal distribution of the data was proven by applying the non-parametric Kolmogorov-Smirnov test (KS-test) with a 95 % confidence interval on the samples [48]. All performed operations are linear, therefore the results possess

Gaussian distribution [49]. Normal distribution is required for mean/variance point estimators and the Student-*t* confidence interval estimator. The KS-test's results of the errors are visualized in Fig. 19.

V. Conclusion and Outlook

The growing usage of cyber physical systems increases the importance of adequate data fusion techniques for (partly) redundant data acquired from multi-modal sensors over the time and/or space domain. This article has demonstrated, that uncertainty weighted discrete orthogonal polynomials are suitable for fusing data originating from an arbitrary number of sources. The accuracy is $> 20\,\%$ better than the conventional, i.e. unweighted, approach. The established mathematical framework is computationally stable, efficient and universally applicable; consequently, it enables its deployment into highly integrated, low-cost embedded target hardware such as (virtual and/or wireless) sensor networks. The synergistic effect of combining the individual sensors leads to an increased quality of the overall measurement. For the experimental position sensitive detector, an uncertainty of $< \pm 0.6\,\mathrm{mm}$ or $< \pm 0.06\,\%$ in a $95\,\%$ confidence interval is achievable for a measurement range of $1\,\mathrm{m}$ over distances of up to $300\,\mathrm{m}$. The a-priori knowledge of the calibration's covariance allows the analytic estimation of the measurement's uncertainty within a given confidence interval - which is in fact essential for the system architecture of measurement devices.

Acknowledgment

We want to thank the company of Geodata[3] for the cooperative work, the supplied equipment and the provided data.

References

[1] J. Piotrowski, *Shaft Alignment Handbook*. M. Dekker, 1995.

[2] J. Dubois, "Very high angular resolution laser beam rider detector having a gated image intensifier and a video camera," United States patent no. 5.280.167, 1994.

[3] R. Neumayr, M. Harker, P. O'Leary, and J. Golser, "New approaches to machine vision based displacement analysis," in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*, 2012, pp. 979–984.

[4] X. Shen, M. Lu, S. Fernando, and S. AbouRizk, "Tunnel boring machine positioning automation in tunnel construction," *Gerontechnology*, vol. 11(2), 2012.

[5] X. Shen and M. Lu, "Development of virtual laser target board to tunnel boring machine guidance control," *International Conference on Computing in Civil Engineering*, pp. 413–420, 2012.

[6] Hamamatsu Photonics K.K., "1D position sensitive detectors," technical specification, 2012. [Online]. Available: www.hamamatsu.com

[7] E. Fortunato, G. Lavareda, R. Martins, F. Soares, and L. Fernandes, "Large-area 1D thin-film position-sensitive detector with high detection resolution," *Sensors and Actuators A: Physical*, vol. 51(2-3), pp. 135–142, 1996.

[8] D. Hall and J. Llinas, *Handbook of multisensor data fusion*. CRC Press LLC, 2001.

[9] H. Mitchell, *Data Fusion: Concepts and Ideas*, 2nd ed. Springer Verlag, February 2012.

[10] D. Hall and S. McMullen, *Mathematical Techniques in Multisensor Data Fusion*, 2nd ed. Artech House Information Warfare Library, February 2004.

[11] A. N. Steinberg, C. L. Bowman, and F. E. White, "Revisions to the JDL data fusion model," in *IRIS Unclassified National Sensor and Data Fusion Conference (NSSDF)*, May 1999.

[12] Z. Liu, D. Forsyth, J. Komorowski, K. Hanasaki, and T. Kirubarajan, "Survey: State of the art in NDE data fusion techniques," *Instrumentation and Measurement, IEEE Transactions on*, vol. 56, no. 6, pp. 2435–2451, December 2007.

[13] F. Russo and G. Ramponi, "Fuzzy methods for multisensor data fusion," *Instrumentation and Measurement, IEEE Transactions on*, vol. 43, no. 2, pp. 288–294, April 1994.

[14] B. Karlsson, J.-O. Jarrhed, and P. Wide, "A fusion toolbox for sensor data fusion in industrial recycling," *Instrumentation and Measurement, IEEE Transactions on*, vol. 51, no. 1, pp. 144–149, Feb 2002.

[15] Z. Liu, D. Forsyth, M.-S. Safizadeh, and A. Fahr, "A data-fusion scheme for quantitative image analysis by using locally weighted regression and Dempster-Shafer theory," *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, no. 11, pp. 2554–2560, Nov 2008.

[16] J. Sequeira, A. Tsourdos, and S. Lazarus, "Robust covariance estimation for data fusion from multiple sensors," *Instrumentation and Measurement, IEEE Transactions on*, vol. 60, no. 12, pp. 3833–3844, December 2011.

[17] J. Frolik, M. Abdelrahman, and P. Kandasamy, "A confidence-based approach to the self-validation, fusion and reconstruction of quasi-redundant sensor data," *Instrumentation and Measurement, IEEE Transactions on*, vol. 50, no. 6, pp. 1761–1769, December 2001.

[18] Geodata, "TAUROS TBM Guidance System," technical specification, September 2008. [Online]. Available: www.geodata.at

[19] ——, "TAUROS Roadheader Guidance System," technical specification, May 2010. [Online]. Available: www.geodata.at

[20] R. Hurteau, M. St-Amant, Y. Laperriere, and G. Chevrette, "Optical guidance system for underground mine vehicles," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, 1992, pp. 639–644 vol.1.

[21] C. Gugg, P. O'Leary, and M. Harker, "Large scale optical position sensitive detector," in *Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International*, 2013, pp. 1775–1780.

[22] Geodata, "Laser Range Test Mine Zielitz," Geodata GROUP, Tech. Rep., December 2011.

[23] IEC 60825-1:2001-08, "Safety of laser products," international standard, August 2001.

[24] 3M, "Silver 20," technical specification, 2012. [Online]. Available: www.3m.com

[25] Bruxsafol Folien GmbH, "LX-70," technical specification, 2012. [Online]. Available: www.bruxsafol.de

[26] VRmagic GmbH, "MFC-12M-4," technical specification, 2012. [Online]. Available: www.vrmagic.com/imaging/

[27] ITK Dr. Kassen GmbH, "iTK Pollux Drive Typ-3," technical specification, 2012. [Online]. Available: www.itknet.de

[28] THK Co. Ltd., "THK KR5520A+1280LH0-0000," technical specification, 2012. [Online]. Available: www.thk.com

[29] J. Gram, "Ueber die Entwickelung realer Funktionen in Reihen mittelst der Methode der kleinsten Quadrate," *Journal fuer die reine und angewandte Mathematik*, p. 150..157, 1883.

[30] R. Barnard, G. Dahlquist, K. Pearce, L. Reichel, and K. Richards, "Gram polynomials and the Kummer function," *Journal of Approximation Theory*, vol. 94, no. 1, pp. 128–143, 1998.

[31] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed. Baltimore: The Johns Hopkins University Press, 1996.

[32] R. Mukundan, S. Ong, and P. A. Lee, "Image analysis by Tchebichef moments," *Image Processing, IEEE Transactions on*, vol. 10, no. 9, pp. 1357–1364, 2001.

[33] P.-T. Yap, R. Paramesran, and S.-H. Ong, "Image analysis by Krawtchouk moments," *Image Processing, IEEE Transactions on*, vol. 12, no. 11, pp. 1367–1377, 2003.

[34] P.-T. Yap and R. Paramesran, "An efficient method for the computation of Legendre moments," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 12, pp. 1996–2002, 2005.

[35] G. Yang, H. Shu, G. C. Han, and L. Luo, "Efficient Legendre moment computation for grey level images," *Pattern Recognition*, vol. 39, pp. 74–80, 2006.

[36] K. M. Hosny, "Exact Legendre moment computation for gray level images," *Pattern Recognition*, vol. 40, no. 12, pp. 3597 – 3605, 2007.

[37] H. Zhu, H. Shu, J. Liang, L. Luo, and J.-L. Coatrieux, "Image analysis by discrete orthogonal Racah moments," *Signal Processing*, vol. 87, no. 4, pp. 687 – 708, 2007.

[38] H. Zhu, H. Shu, J. Zhou, L. Luo, and J. Coatrieux, "Image analysis by discrete orthogonal dual hahn moments," *Pattern Recognition Letters*, vol. 28, no. 13, pp. 1688 – 1704, 2007.

[3] ©Geodata GmbH, Leoben, Austria, www.geodata.at

[39] J. Baik, *Discrete orthogonal polynomials: asymptotics and applications*, ser. Annals of mathematics studies. Princeton University Press, 2007, no. Bd. 13.

[40] P. O'Leary and M. Harker, "An algebraic framework for discrete basis functions in computer vision," in *2008 6th ICVGIP*. Bhubaneswar, India: IEEE, 2008, pp. 150–157.

[41] G. Golub and G. Meurant, *Matrices, Moments and Quadrature with Applications*, ser. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

[42] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. New York, NY, USA: Cambridge University Press, 2007.

[43] P. O'Leary and M. Harker, "An algebraic framework for discrete basis functions in computer vision," *Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, pp. 150–157, 2008.

[44] S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*, 3rd ed. Springer Verlag, 1999.

[45] P. Schalk, "Metric vision methods for material and product inspection," Ph.D. dissertation, University of Leoben, 2007.

[46] E. Fauster, "Statistical uncertainty analysis for image processing algorithms in metric vision systems," Ph.D. dissertation, University of Leoben, 2008.

[47] A. Makynen, J. Kostamovaara, and R. Myllyla, "Displacement sensing resolution of position-sensitive detectors in atmospheric turbulence using retroreflected beam," *Instrumentation and Measurement, IEEE Transactions on*, vol. 46, no. 5, pp. 1133–1136, 1997.

[48] E. Kreyszig, *Statistische Methoden und ihre Anwendungen*. Göttingen: Vandenhoeck und Ruprecht, 1979.

[49] J. C. Clarke, "Modelling uncertainty: A primer," Department of Engineering Science, Oxford University, Tech. Rep., 1998.

14 Active Laser Target V2

Tunnel boring machines (TBM) excavate several [km] long tunnels with a circular cross section and diameters of up to 19.25 [m][1]. TBMs are used as an alternative to conventional drilling and blasting methods; the disturbance to the surrounding ground is limited; thus, making the approach suitable for rural as well as urbanized areas. TBMs have large overhead costs (i.e., construction and transportation); hence, this method is only appropriate for projects of a certain dimension. A detailed description of the boring process is given in [15, 112].

Highest accuracy and precision is required to control and navigate a TBM along the planned route. State-of-the-art control systems utilize a laser theodolite total station and photo-sensory targets with retro-reflectors/motor-prisms. The target units are mounted on defined reference points on the TBM. The theodolite continuously monitors their position and the control-system determines the TBM's position geometrically. Further sensors such as inclinometers or gyroscopes deliver additional information about horizontal and vertical deviation of the machine [142].

Navigation of machinery in underground construction projects is a challenge due to the harsh environmental conditions such as heat, dust and especially vibrations. The original *Active Laser Target* (ALT) presented in [157, 158] was a feasibility study to develop a fully integrated measurement device for optical displacement analysis of TBMs. A reference laser beam impinges upon two parallel aligned targets, the position of the laser spots enable the computation of the TBM's pitch and yaw angles. The measurement principle avoids the use of fragile mirror-like targets at exposed positions in the working area; it is visualized in Fig. 14.1 together with the pitch-yaw-roll angles of the device in Fig.14.2. The corresponding laboratory prototype is shown in Fig. 14.3 and the xy-table used for calibration is shown in Fig. 14.4. Following improvements are introduced in the second iteration (ALTv2) of the system design:

1. The aluminum target is **sandblasted** to increase the Lambertian reflectance and the target is **anodized** red to improve the absorption of non-red spectra of visible light. These measures increase the robustness of the measurement process w.r.t. the optical properties of the measurement instrument;

2. The use of **Gigabit Ethernet (GigE) cameras** with **Power over Ethernet (PoE)** support eliminates the need of a mini PC inside the housing controlling the originally employed USB2.0 cameras. The instrument can be remote-controlled from a supervisory system; GigE with PoE supports cable lengths of up to 100 [m], IP67 connectors are available. The used standard C-mount lenses are mechanically better suited for an industrial environment than their S-mount counterparts;

3. The cameras are controlled over a UDP/IP network with the **GenICam** standard, which

[1] Herrenknecht AG, www.herrenknecht.com

reduces the programming effort thanks to the common interface. Additional cameras can be flexibly added to the measurement system and/or defect devices can simply be replaced;

4. The software is implemented using **object-oriented programming (OOP)**, this facilitates functional decomposition, code reusability and event-driven control of the hardware components. The class diagrams of the calibration and measurement process are shown in Fig.14.5 and 14.6 respectively;

5. The **system level calibration process** incorporates a theodolite mounted on an xy-table, see Fig. 14.4. An **estimation for the achievable accuracy** can be given with the integrated cross validation procedure. The measurement process is designed to be robust w.r.t. to vibrations; plausibility checks are performed over consecutive iterations;

6. The optical components have been chosen with the goal to **minimize non-linear distortions**. The coordinate mapping between the camera space in [pixel] and the metric real space in [mm] is solved with a linear projective (homographic) transformation. Transformations involving higher-order tensor products tend to be computationally intensive and less robust in comparison to homography, especially when extrapolating the functions over the borders of their support. The remaining systematic error is within the tolerated error bounds.

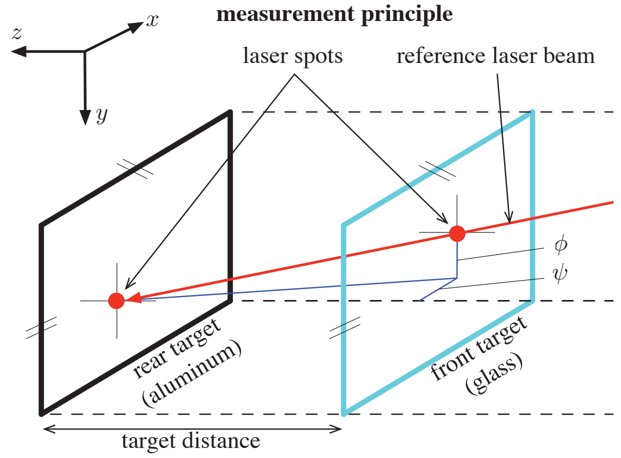


Fig. 14.1 The reference laser beam impinges upon the parallel aligned front (glass) target and the rear (aluminum) target. The pitch ϕ and yaw ψ angles are determined from the position of the projected laser spots.

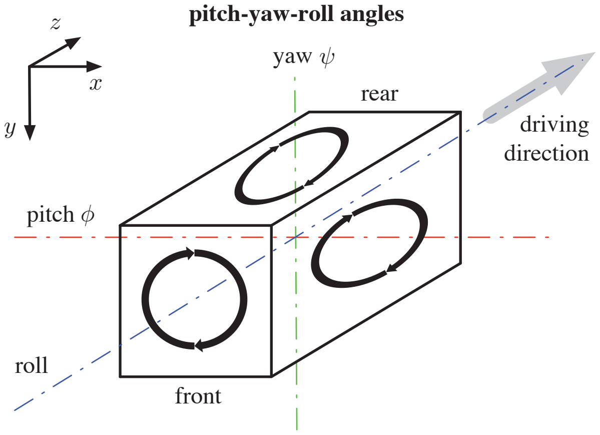


Fig. 14.2 The TBM's orientation is determined by the pitch ϕ and yaw ψ angles, the roll angle is measured separately with an inclinometer. The device's axis and the TBM's principal axis are coaxially aligned.

The software has been implemented in MATLAB using the Image Acquisition Toolbox as well as the Image Processing Toolbox. In an OOP sense, special focus was laid on functional decomposition and common interfaces. The software framework supports the interchange of certain support classes with alternative implementations thanks to defined interfaces, e.g., for projects with different system requirements. The class diagram of the calibration procedure is shown in Fig. 14.5. Although class diagrams only show the static behavior of a program, the four main classes are ordered sequentially:

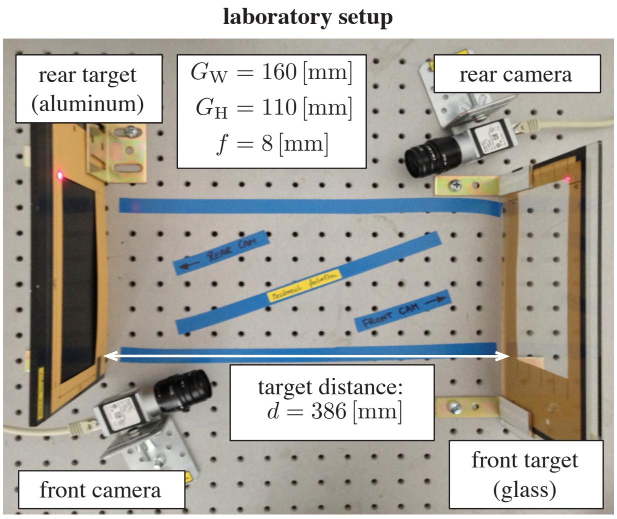


Fig. 14.3 The front and rear targets are observed by a pair of GigE cameras. The lenses have a focal length $f = 8\,[\mathrm{mm}]$, which leads to a target distance $d = 386\,[\mathrm{mm}]$ when considering the measurement range of $(G_{\mathrm{W}} \times G_{\mathrm{H}}) = (160\,[\mathrm{mm}] \times 110\,[\mathrm{mm}])$.

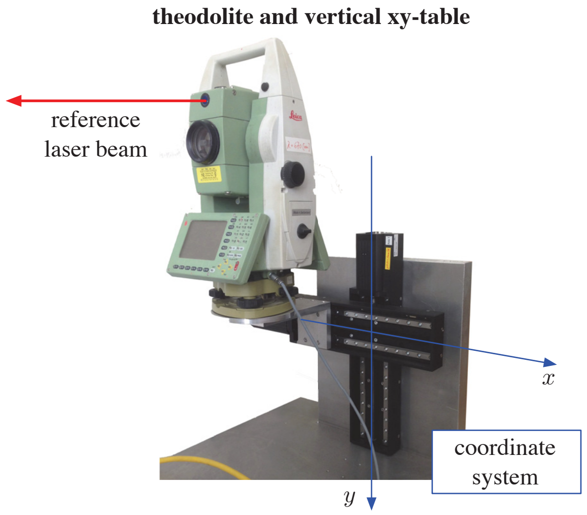


Fig. 14.4 The theodolite is mounted on a vertical xy-table; it follows a calibration pattern. The image acquisition of the cameras and the movement of the xy-table/theodolite are synchronized via event-based programming including callback functions and notifications.

1. **Acquisition.** The xy-table moves the theodolite according to a predefined calibration pattern. The cameras are synchronized to this movement and acquire images from each calibration node. The raw images are stored for documentation and/or for recomputing calibrating coefficients with a different set of parameters;

2. **Process.** Basic image improvement/filtering operations are performed, the laser spots are segmented and the center coordinates (features) are extracted. The most robust technique is the computation of the laser spot's center of gravity; an alternative method is conic fitting;

3. **Model.** The model's coefficients for the front (f) and rear (r) target $\{\mathsf{H}_{\mathrm{f}}, \mathsf{H}_{\mathrm{r}}\}$ are determined separately. The quality increases with increasing number of calibration nodes; however, there is a lower bound for the uncertainty;

4. **Evaluation.** The cross validation points are remapped using $\{\mathsf{H}_{\mathrm{f}}, \mathsf{H}_{\mathrm{r}}\}$, the residual is an indicator for the calibration's quality and the achievable accuracy. The calibration can be repeated with other parameters if the results are not satisfying. A calibration report is generated for documentation.

The calibration procedure is designed to be carried out under laboratory conditions with a MATLAB based development system. Having the model parameters available, the measurement process is relatively simple. The support classes from the calibration procedures are reused in the current implementation of the measurement process, see Fig. 14.6. Alternative application dependent implementations in other languages such as `C++/C#` or even `C` for an embedded vision system are also possible. Note, that in this case the process parameters must be kept consistent with the original calibration.

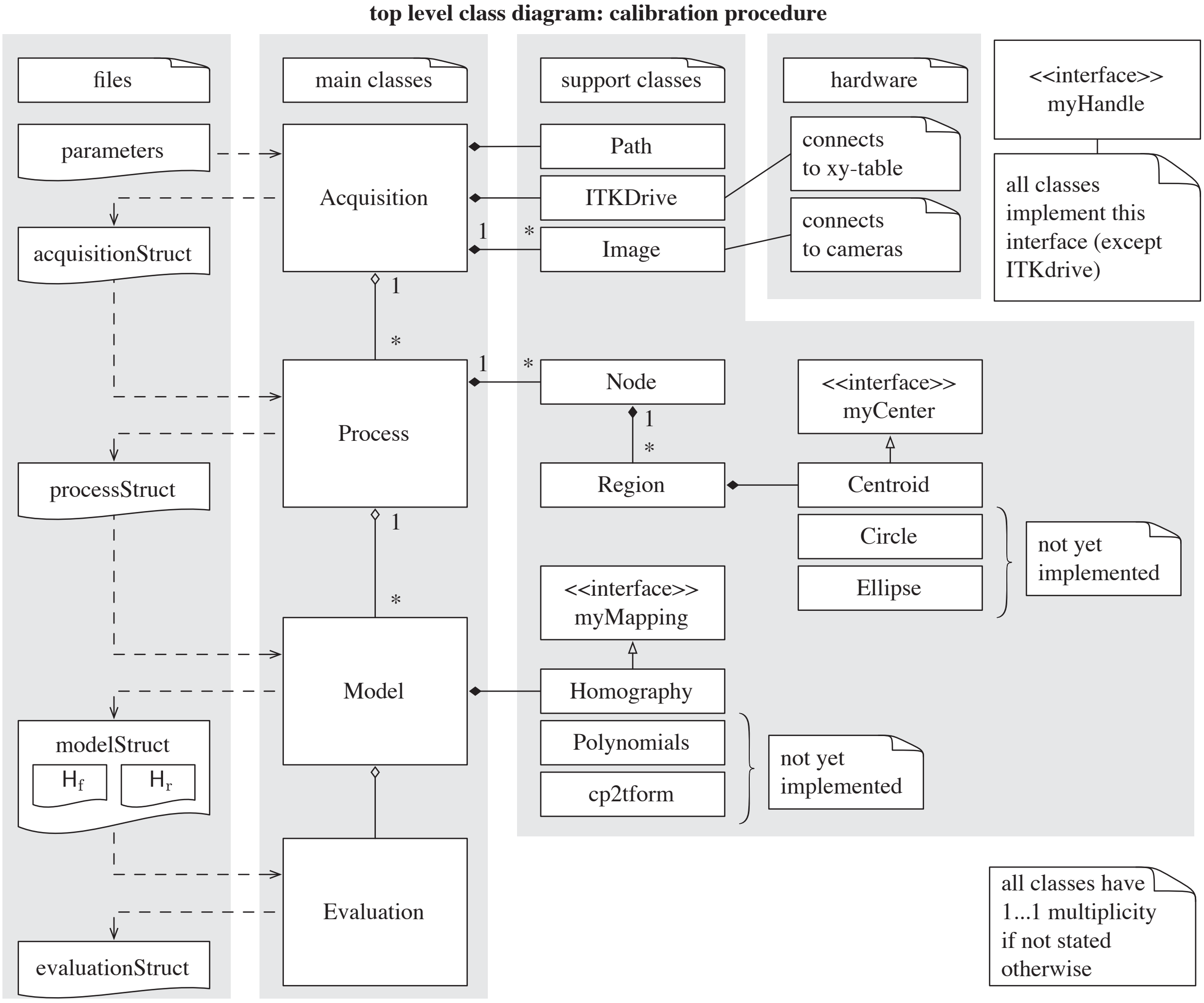


Fig. 14.5 Class diagram of the initial calibration procedure.

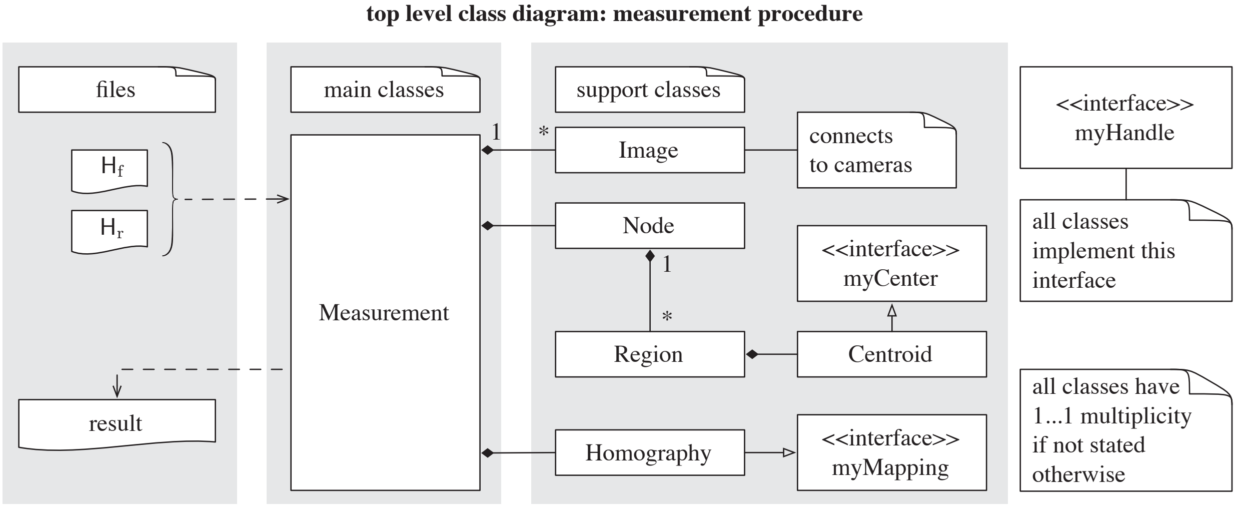


Fig. 14.6 Class diagram of the iterative measurement procedure.

14.1 IEEE - International Instrumentation and Measurement Conference (I2MTC) 2015

May 11 - May 14, 2015 @ Pisa, Italy [74]

Citation[2]

```
@inproceedings{GuggOleary2015i2mtc,
 author     = {Gugg, Christoph and O'Leary, Paul},
 title      = {Robust Machine Vision Based Displacement
              Analysis for Tunnel Boring Machines},
 booktitle  = {IEEE International Instrumentation and
              Measurement Technology Conference (I2MTC)},
 year       = {2015},
 pages      = {tba},
 doi        = {tba},
 ISSN       = {tba}}
```

Note: citation details are unknown at the release date of this thesis.

Keywords

tunnel boring machine (TBM); laser guidance; optical displacement analysis; projective transformation; measurement uncertainty; GigE interface; GenICam; image processing;

[2] IEEE Xplore Digital Library, `ieeexplore.ieee.org`

Robust Machine Vision Based Displacement Analysis for Tunnel Boring Machines

Christoph Gugg and Paul O'Leary
Chair of Automation, Department Product Engineering
University of Leoben, Styria, Austria
Email: christoph.gugg@unileoben.ac.at

***Abstract*—Underground construction projects require accurate techniques for machine guidance control. This paper presents an advanced design of a fully integrated active laser target (ALT), which is employed as an optical displacement and orientation sensor. The instrument is mounted on a tunnel boring machine (TBM) and delivers the TBM's pitch and yaw angles to a remote host system. The optical arrangement of the device includes a semitransparent glass target and an opaque aluminum target, whereby measures are taken to increase the sensitivity of red light. A red reference laser beam is projected onto both parallel targets; the positions of the laser spots are observed by two Power over Ethernet (PoE) enabled Gigabit Ethernet (GigE) cameras. A robust plane-to-plane mapping using projective transformation is presented together with a cross validation procedure, which evaluates the quality of the calibration. An a-priori estimation of measurement uncertainty can be given. The system level calibration process yields two sets of transformation coefficients, such that the distortion associated with the optical components and the inexactness of the mechanical construction are effectively canceled out. Multiple images are acquired for a single measurement and analyzed statistically to deliver a statement of measurement uncertainty in order to compensate for mechanical vibrations during the machine's operation.**

I. Introduction

The paper is a direct successor to the work presented in [1], which describes the design and manufacturing of a machine vision based measurement device for guiding tunnel boring machines (TBMs), called the 'Active Laser Target' (ALT). Besides this particular application, such measurement devices are used for shooting range simulation [2], positioning of machines [3] and/or large objects [4]. Measurement principles include the use of retroreflectors [5], position sensitive detectors [6], camera imaging systems [3] and laser beam rider[1] techniques [7]. Commonly, TBMs are controlled via remote theodolites in combination with two to six retroreflectors mounted at key positions on the machine [8]. The TBM's absolute position is then determined via triangulation [9] within an accuracy of $\pm 10\,[\mathrm{mm}]$, whereby a single measurement takes up to $45\,[\mathrm{s}]$ to complete [10]. Due to the TBM's slow movement, these measurement are performed only on demand. The harsh environmental conditions of underground constructions are not ideal for such complex measurement arrangements using mirrors and other fragile components; therefore, the first generation of the ALT was a feasibility study to develop a fully integrated measurement device without the need of additional exposed parts. The second, improved generation of the ALT, the 'ALTv2', is presented in this paper; new contributions are:
(1) The paper's main focus is the formulation of a calibration procedure which yields calibration data, i.e., the transformation matrices; furthermore, the integrated cross validation procedure systematically uses a set of *real* measurements to acquire an estimation of the expected measurement uncertainty;
(2) Multiple images are sampled during the measurement process for statistical analysis to give a confidence interval for the measurement. Coordinates are mapped via projective transformation utilizing the calibration data. The linear method is valid for automatic programming and, thus, suitable for deployment on an embedded vision system;
(3) The software architecture is implemented with object-oriented programming (OOP), enabling functional decomposition and reusability of code for future measurement applications. Furthermore, OOP enables an event-driven calibration procedure for hardware synchronization between the cameras and the xy-table;
(4) The system architecture is completely re-engineered. The original ALT utilized USB cameras and, as a consequence, required an integrated computer. The new concept uses cameras with a Gigabit Ethernet (GigE) interface and Power over Ethernet (PoE) energy supply; cable lengths of up to $100\,[\mathrm{m}]$ are supported. The remote host uses the GenICam[2] standard to acquire images;
(5) Underground constructions exhibit an atmosphere with dust, moisture and ambient sources of white light. The optical components are selected with the goal to compensate for these external influences and to increase the overall robustness of the machine vision system.

II. Principle of Operation

A remote 'Leica TPS1200' theodolite [9] is used as laser source, its 'PinPoint R300' laser unit has a wavelength of $\lambda = 670\,[\mathrm{nm}]$ within the red spectrum of light. The theodolite's accuracy is $\pm 0.1\,[\mathrm{mgon}]$ for angle measurements[3]. The reference laser beam is projected onto the device; in turn, the measurement system delivers the included horizontal yaw angle ψ and vertical pitch angle ϕ between the laser beam and the machine's principal axis, see Fig. 2. The roll angle is

[1] This technology is closely related to military applications.

[2] EMVA Generic Interface for Cameras (GenICam), www.genicam.org

[3] *Gon* is the common unit for angles in geodesy, $1^\circ \triangleq 0.9\,[\mathrm{gon}]$.

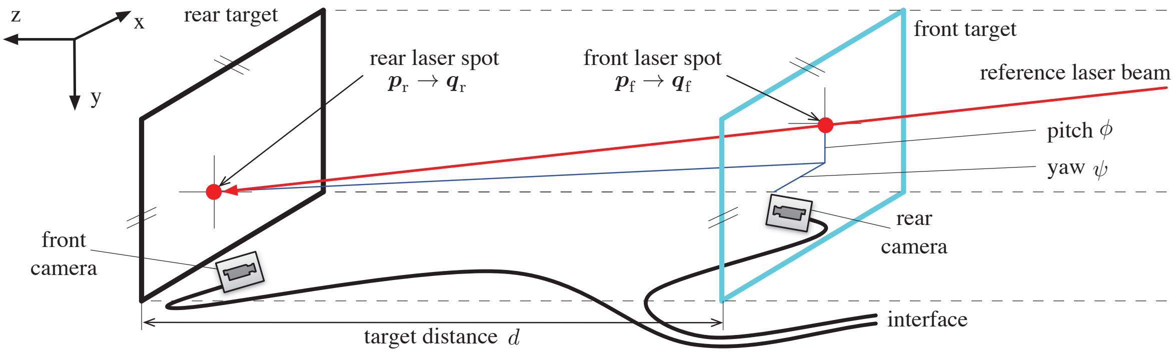


Fig. 1. The laser beam enters the measurement device via the semitransparent glass window. The front laser spot $\boldsymbol{p}_{\mathrm{f}}$ is projected onto the backside of this window, which appears as the front target. $\boldsymbol{p}_{\mathrm{f}}$ is detected by the front camera and mapped to $\boldsymbol{q}_{\mathrm{f}}$, whereas the rear laser spot $\boldsymbol{p}_{\mathrm{r}}$ is projected onto the rear target, detected by the rear camera and mapped to $\boldsymbol{q}_{\mathrm{r}}$ respectively. Both targets are aligned parallel, hence the differences Δx and Δy are simply determinable. The pitch ϕ and yaw ψ angles are calculated through the position of both laser spots and the known target distance d.

of no particular interest in this application. The front (f) laser spot $\boldsymbol{p}_{\mathrm{f}}$ on the front target and the rear (r) leaser spot $\boldsymbol{p}_{\mathrm{r}}$ on the rear target are imaged by two opposed cameras, see the illustrated measurement principle in Fig. 1. Fig. 3 shows the housing of the prototype and images of both laser points. The image processing yields the center coordinates within in the camera space in [pixel], these coordinates are then mapped to the corresponding metric real space coordinates $\boldsymbol{q}_{\mathrm{f}}$ and $\boldsymbol{q}_{\mathrm{r}}$ in [mm] via a projective transformation,

$$\{\boldsymbol{p}_{\mathrm{f}}, \boldsymbol{p}_{\mathrm{r}}\}\,[\mathrm{pixel}] \xrightarrow{\text{mapping}} \{\boldsymbol{q}_{\mathrm{f}}, \boldsymbol{q}_{\mathrm{r}}\}\,[\mathrm{mm}]. \qquad (1)$$

The targets are aligned parallel; hence, the horizontal difference Δx and the vertical difference Δy are calculated as

$$\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} = \boldsymbol{q}_{\mathrm{r}} - \boldsymbol{q}_{\mathrm{f}} = \begin{bmatrix}x_{\mathrm{r}}\\ y_{\mathrm{r}}\end{bmatrix} - \begin{bmatrix}x_{\mathrm{f}}\\ y_{\mathrm{f}}\end{bmatrix}. \qquad (2)$$

The front and rear target are separated by the known target distance d. The yaw ψ and pitch ϕ angles are computed via

$$\psi = \arctan\frac{\Delta x}{d} \quad \text{and} \quad \phi = \arctan\frac{\Delta y}{d}. \qquad (3)$$

Both angles are delivered to a supervisory system.

III. Optical Arrangement & System Architecture

Contractors in underground construction projects have little space to install their equipment; therefore, a compact hardware design is one of the design goals. The target distance d determines the outer dimensions of the device, whereby d directly depends on the object distance g and the space required by the camera l, i.e., $d \geq g+l$, see Fig 4. Furthermore, the width (W) and height (H) of the target ($G_{\mathrm{W}} \times G_{\mathrm{H}}$), i.e., the measurement range, the sensor size ($B_{\mathrm{W}} \times B_{\mathrm{H}}$) and the lens' focal length f give an estimate of the object distance g [11],

$$g_{(\mathrm{W,H})} = f\left(\frac{G_{(\mathrm{W,H})} + B_{(\mathrm{W,H})}}{B_{(\mathrm{W,H})}}\right). \qquad (4)$$

The object distance is separably approximated for both dimensions, where $g = \max(g_{\mathrm{H}}, g_{\mathrm{W}})$. A compact hardware design requires the value of g to be small; given the target size, g is shortened by using a lens with a small f and a relatively large sensor. Consequently, selecting appropriate components reduces non-linear optical distortions to a minimum:

1) the mono CCD camera 'Basler acA1300-30gm' [12] with a resolution of 1296 [pixel] × 966 [pixel] @ 30 [fps], a chip size of $(B_{\mathrm{W}} \times B_{\mathrm{H}}) = 4.86\,[\mathrm{mm}] \times 3.62\,[\mathrm{mm}]$ and a PoE enabled GigE interface with GenICam support;
2) the C-mount lens 'Kowa LM8JCM' [13] with $f = 8\,[\mathrm{mm}]$, a distortion of $-0.3\,[\%]$ and a minimal focus distance $g_{\mathrm{min}} = 100\,[\mathrm{mm}]$.

The architecture of the components is sketched in Fig. 5. Following measures are take to maximize the selectivity of red light and to suppresses ambient white light:

1) the rear aluminum target is sandblasted to improve the Lambertian reflectance; and
2) it is anodized red to provide the highest reflectance rate for the red laser relative to other wavelengths;

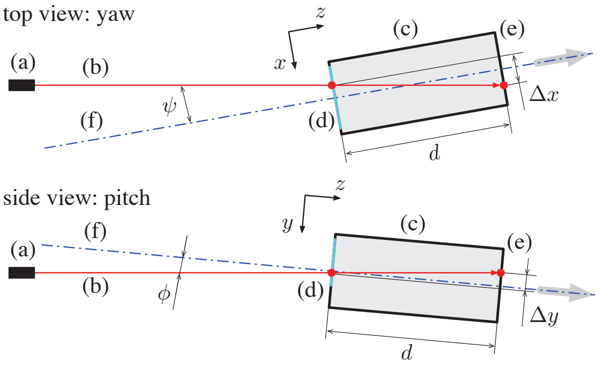


Fig. 2. The laser source (a), most commonly a theodolite, generates a reference laser beam (b). The laser beam impinges upon the device's (c) front semitransparent glass window (d) and the transmission portion illuminates a spot on the rear opaque aluminum target (e). Both targets are parallel to each other and orthogonal to the machines principal axis (f). Utilizing the target distance d and the horizontal difference Δx yields the yaw angle ψ, whereas utilizing d and the vertical difference Δy yields the pitch angle ϕ.

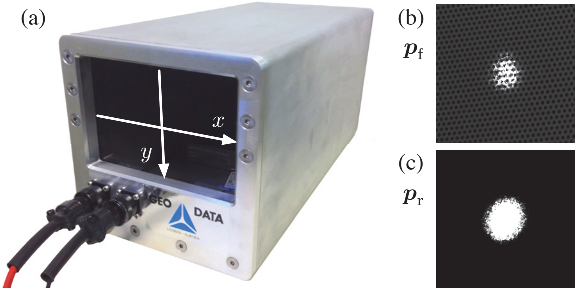


Fig. 3. The aluminum housing (a) of the device is sealed from dust and moisture. There are two IP65 connectors for power supply and network connection. The instrument's coordinate system is visualized at the glass window, which also acts as front target. The target's measurement range is $(G_\mathrm{W} \times G_\mathrm{H}) = (160\,[\mathrm{mm}] \times 110\,[\mathrm{mm}])$. The hexagonal pattern is screen printed on the inside of the window; it has a dot diameter of $0.5\,[\mathrm{mm}]$ and a dot distance of $1\,[\mathrm{mm}]$. This pattern determines the visibility of the laser spot on the front target $\boldsymbol{p}_\mathrm{f}$ (b) as well as the rear target $\boldsymbol{p}_\mathrm{r}$ (c).

3) the front semitransparent glass target has a hot mirror foil applied to it, which reflects the infrared portions of white light;
4) an optical interference filter, matched to the wavelength of the laser, is mounted between the camera's sensor and the lens.

Another option would be the use of color sensors and exclusively selecting the red channel. However, the Bayer pattern reduces the usable resolution.

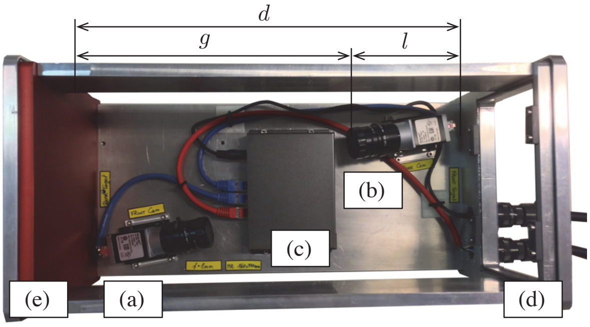


Fig. 4. The front camera (a) and rear camera (b) are connected to the PoE switch (c), which is in turn connected to the external power supply and remote host. The front glass target (d) is coated with hot mirror foil, the rear aluminum target (e) is sandblasted and red anodized. The object distance g and the space required by the camera l determine the lower bound of d. In this application, $d < 400\,[\mathrm{mm}]$.

IV. Calibration Procedure

The goal of the calibration process is to generate two independent matrices $\{\mathsf{H}_\mathrm{f}, \mathsf{H}_\mathrm{r}\}$ of calibration coefficients for the front (f) and rear (r) target to formulate the transformation function stated in Eqn. (1). According to Fig. 6, the set Q of m metric real space calibration nodes in [mm] is combined with the camera space calibration nodes $\{\mathsf{P}_\mathrm{f}, \mathsf{P}_\mathrm{r}\}$ in [pixel] to acquire a plane-to-plane mapping function. Possible methods are lookup tables, polynomial approximation [14], [15] or projective transformation [16]. The latter is chosen, because a homographic transformation is robust, supports extrapolation and the mapping is computationally cheap. During data acquisition, a set of camera space evaluation nodes $\{\tilde{\mathsf{P}}_\mathrm{f}, \tilde{\mathsf{P}}_\mathrm{r}\}$ corresponding to the n real space evaluation nodes stored in $\tilde{\mathsf{Q}}$ is also acquired for the cross validation in order to evaluate the calibration quality, i.e., the residuals between the measured points $\{\hat{\mathsf{Q}}_\mathrm{f}, \hat{\mathsf{Q}}_\mathrm{r}\}$ and the original points stored in $\tilde{\mathsf{Q}}$.

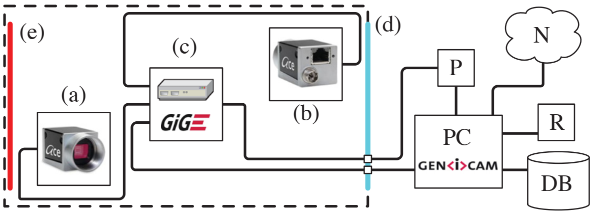


Fig. 5. The components described in Fig. 4 are connected to a remote host (PC) and power supply (P), whereby the GigE interface supports cable lengths of up to $100\,[\mathrm{m}]$. Images are acquired via the GenICam standard; the PC saves the data to a database (DB) and generates (hardcopy) reports (R). The measurement system is accessible from outside via network (N).

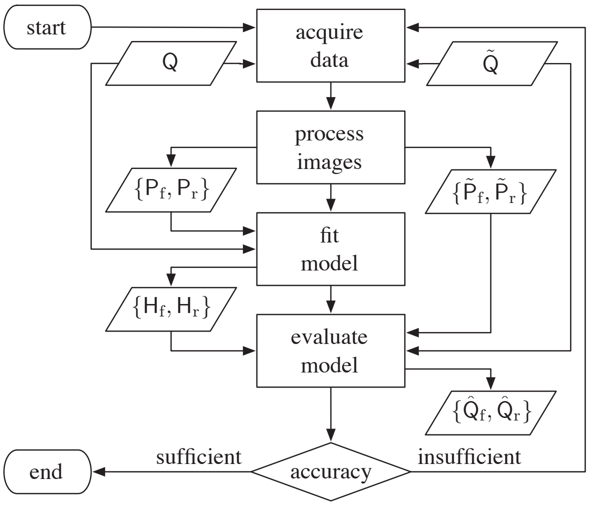

start
acquire data
Q
$\tilde{\mathsf{Q}}$
process images
$\{\mathsf{P}_\mathrm{f}, \mathsf{P}_\mathrm{r}\}$
$\{\tilde{\mathsf{P}}_\mathrm{f}, \tilde{\mathsf{P}}_\mathrm{r}\}$
fit model
$\{\mathsf{H}_\mathrm{f}, \mathsf{H}_\mathrm{r}\}$
evaluate model
$\{\hat{\mathsf{Q}}_\mathrm{f}, \hat{\mathsf{Q}}_\mathrm{r}\}$
end
sufficient
accuracy
insufficient

Fig. 6. The calibration process yields separate homography matrices $\{\mathsf{H}_\mathrm{f}, \mathsf{H}_\mathrm{r}\}$ for the front and rear target by combining Q with $\{\mathsf{P}_\mathrm{f}, \mathsf{P}_\mathrm{r}\}$. The integrated evaluation process enables an objective judgment of the accuracy to be expected by comparing $\tilde{\mathsf{Q}}$ with $\{\hat{\mathsf{Q}}_\mathrm{f}, \hat{\mathsf{Q}}_\mathrm{r}\}$. If the accuracy is insufficient, new data can be acquired with different parameters and/or more calibration points. A design of experiments test plan helps to find suitable parameters.

A. Data Acquisition and Image Processing

The non time-critical calibration procedure is implement as OOP in MATLAB. This includes the control of the two cameras via GenICam as well as the control of a vertical xy-table based on two orthogonal 'iTK ST9' linear drives [17], which move the theodolite and its laser unit. Fig. 7 illustrates the system level calibration and Fig. 8 shows the associated sequence diagram of the data acquisition process.

B. Model for Projective Transformation

The affine coordinates $\boldsymbol{p} = [a, b]^\mathrm{T}$ in [pixel] and $\boldsymbol{q} = [x, y]^\mathrm{T}$ in [mm] have homogeneous equivalents $\boldsymbol{\pi} = [\alpha, \beta, \gamma]^\mathrm{T}$ and

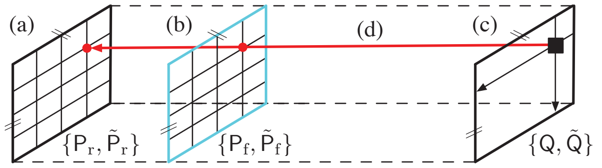

(a)
(b)
(d)
(c)
$\{\mathsf{P_r}, \tilde{\mathsf{P}}_\mathsf{r}\}$
$\{\mathsf{P_f}, \tilde{\mathsf{P}}_\mathsf{f}\}$
$\{\mathsf{Q}, \tilde{\mathsf{Q}}\}$

Fig. 7. During system level calibration, the rear target (a), the front target (b) and the vertical xy-table (c) must be aligned parallel to each other, i.e., the laser beam (d) is orthogonal to all planes. The xy-table moves a theodolite in a vertical plane according to a predefined pattern $\{\mathsf{Q}, \tilde{\mathsf{Q}}\}$, the movement is synchronized to the image acquisition, see Fig. 8. System level calibration corrects errors associated with the optical arrangement and the mechanical construction.

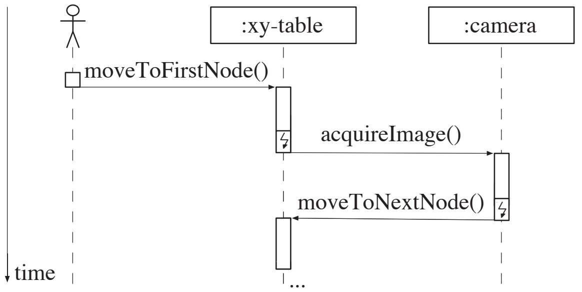


Fig. 8. The temporal behavior of data acquisition depends on the hardware components. The xy-table has only limited acceleration and velocity; therefore, reaching different points of the calibration pattern takes various amounts of time. The acquisition of images via the GigE interface has no distinct duration, e.g it is influenced by the network traffic. Synchronization of the tasks is achieved via OOP event-based (ϟ) control.

$\boldsymbol{\omega} = [\xi, \upsilon, \zeta]^\mathrm{T}$. The coordinate systems[4] are related via a non-bijective mapping of $\mathbb{R}^3 \to \mathbb{R}^2$, i.e.,

$$\boldsymbol{p} = \left[\frac{\alpha}{\gamma}, \frac{\beta}{\gamma}\right]^\mathrm{T} \quad \text{and} \quad \boldsymbol{q} = \left[\frac{\xi}{\zeta}, \frac{\upsilon}{\zeta}\right]^\mathrm{T}, \tag{5}$$

where $\{\gamma, \zeta\}$ are the homogeneous components. Singularities appear if these values are 0, i.e., points at infinity. The transformation is given by

$$\boldsymbol{\omega} = \mathsf{H}\boldsymbol{\pi}, \tag{6}$$

where H is the (3×3) homography matrix. Eqn. (6) is expressed in matrix form

$$\begin{bmatrix}\xi\\ \upsilon\\ \zeta\end{bmatrix} = \begin{bmatrix}h_{11} & h_{12} & h_{13}\\ h_{21} & h_{22} & h_{23}\\ h_{31} & h_{32} & h_{33}\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix}. \tag{7}$$

H may be divided by a non-zero real scalar, leading to eight unknown variables. Four non-collinear points in the real and camera space are required to solve this equation, i.e., each known point yields two equations. The common approach to compute H is the direct linear transformation (DLT) algorithm [16]. The DLT algorithm does not differentiate between statistically invariant constants as well as the linear and quadratic parts of the error. Residualization based on orthogonal matrix projections delivers a reduced error structure for the linear system of equations. This efficient non-iterative computation of H can be found in literature [18]. Using the information stored in the set Q of known real space coordinates (Fig. 9) and the sets $\{\mathsf{P_f}, \mathsf{P_r}\}$ of acquired camera space coordinates (Fig. 10), we compute $\{\mathsf{H_f}, \mathsf{H_r}\}$ separately for the front and rear target, i.e.,

$$\{\mathsf{Q}, \mathsf{P_f}\} \to \mathsf{H_f} \quad \text{and} \quad \{\mathsf{Q}, \mathsf{P_r}\} \to \mathsf{H_r}. \tag{8}$$

Overdetermined systems are solved in a least-squares sense using singular value decomposition (SVD).

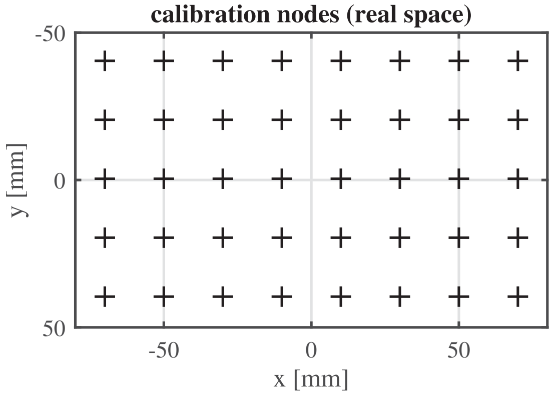


Fig. 9. The calibration pattern is stored in the set Q of real coordinates. The total number m of calibration points influences the achievable accuracy, whereby at least 4 independent points are required. In this example, $m = 8 \times 5 = 40$ calibration nodes are arranged in a regular 20 [mm] grid.

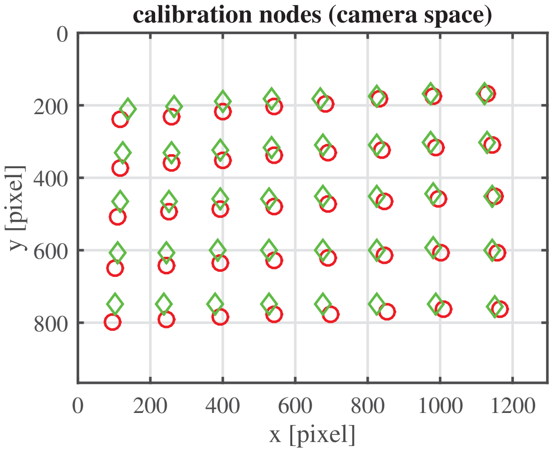


Fig. 10. The positions of the points stored in the set $\mathsf{P_r}$ of rear nodes (○) and the set $\mathsf{P_f}$ of front nodes (◊) correlating to the set of points stored in Q (Fig. 9) are not equal; hence, the computation of separate transformation matrices $\mathsf{H_f}$ and $\mathsf{H_r}$ is required.

C. Cross Validation

During data acquisition, the set of n independent *real* evaluation nodes $\tilde{\mathsf{Q}}$ delivers the sets $\{\tilde{\mathsf{P}}_\mathsf{f}, \tilde{\mathsf{P}}_\mathsf{r}\}$ of camera evaluation nodes. Mapping these coordinates back to the real space, yields

[4] This section is valid for the front and rear target.

the measured sets $\{\tilde{\mathsf{Q}}_\mathrm{f}, \tilde{\mathsf{Q}}_\mathrm{r}\}$. The residual matrices $\{\mathsf{R}_\mathrm{f}, \mathsf{R}_\mathrm{r}\}$ are separately determined for the front and rear target via

$$\mathsf{R}_\mathrm{f} = \tilde{\mathsf{Q}} - \hat{\mathsf{Q}}_\mathrm{f} \quad \text{and} \quad \mathsf{R}_\mathrm{r} = \tilde{\mathsf{Q}} - \hat{\mathsf{Q}}_\mathrm{r}. \tag{9}$$

A residual matrix $\mathsf{R} = [(\tilde{\boldsymbol{q}_1} - \hat{\boldsymbol{q}_1}) \ldots (\tilde{\boldsymbol{q}_n} - \hat{\boldsymbol{q}_n})]$ can be split up in x and y dimensions, i.e.,

$$\mathsf{R} = \begin{bmatrix} (\tilde{x}_1 - \hat{x}_1) \ldots (\tilde{x}_n - \hat{x}_n) \\ (\tilde{y}_1 - \hat{y}_1) \ldots (\tilde{y}_n - \hat{y}_n) \end{bmatrix}. \tag{10}$$

The Euclidean distance between a validation point $\tilde{\boldsymbol{q}}_j$ and a measured point $\hat{\boldsymbol{q}}_j$ is equivalent to the vector 2-norm,

$$r_j = ||\tilde{\boldsymbol{q}}_j - \hat{\boldsymbol{q}}_j)||_2 = \sqrt{(\tilde{x}_j - \hat{x}_j)^2 + (\tilde{y}_j - \hat{y}_j)^2}. \tag{11}$$

Concatenating the r_j values yields the residual vector $\boldsymbol{r} = [r_1 \ldots r_n]$. The standardized 2-norm $\|\,\boldsymbol{r}\,\|_2 / n$ is an indicator of the achievable accuracy. Note, that increasing the number m of calibration nodes in Q also increases the quality of the transformation; unfortunately, the measurement uncertainty can only be decreased to a lower bound. The results for this particular cross validation setup (Fig. 11 and 12) using $m = 40$ calibration points and $n = 95$ evaluation points are shown in Table I. Conventional triangulation has an accuracy of ±10 [mm]; however, the new approach is not directly comparable to state-of-the-art solutions because of the completely different measurement principle. Projective transformation does not compensate non-linear distortions, leading to a remaining systematic error $\bar{r}$. Taking a look at the standard deviations and the 2-norms reveals quality differences between the front and the rear target. This is mainly caused by the image processing, as the front laser spot is far more exposed to external influences than the rear laser spot, see Fig. 3. The Gaussian nature of the residuals can be verified statistically with a Kolmogorov-Smirnov and/or an Anderson-Darling test.

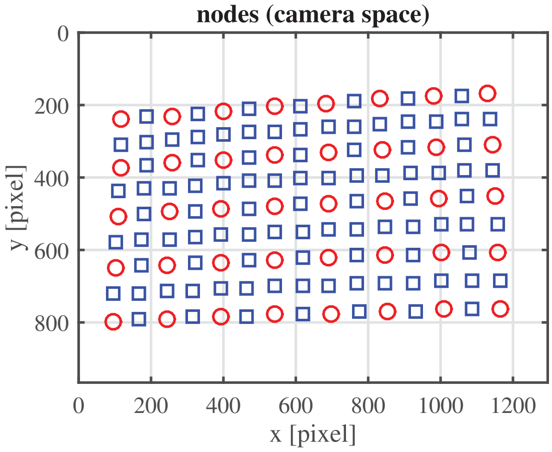


Fig. 11. According to Fig. 6, a set P of m calibration nodes (○) and and a set $\tilde{\mathsf{P}}$ of n evaluation nodes (□) are acquired. In this example, $n = 95$ evaluation nodes are placed between $m = 40$ calibration nodes, i.e., where the biggest error is to be expected. This is done separately for both targets.

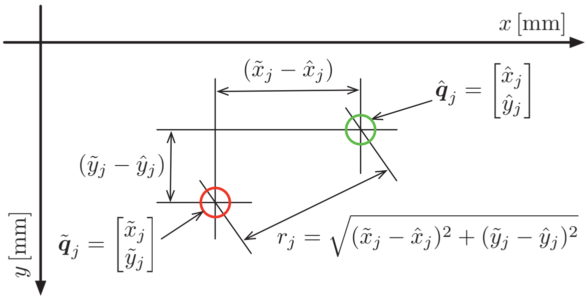

x [mm]
$(\tilde{x}_j - \hat{x}_j)$
$\hat{\boldsymbol{q}}_j = \begin{bmatrix} \hat{x}_j \\ \hat{y}_j \end{bmatrix}$
$(\tilde{y}_j - \hat{y}_j)$
y [mm]
$\tilde{\boldsymbol{q}}_j = \begin{bmatrix} \tilde{x}_j \\ \tilde{y}_j \end{bmatrix}$
$r_j = \sqrt{(\tilde{x}_j - \hat{x}_j)^2 + (\tilde{y}_j - \hat{y}_j)^2}$

Fig. 12. The residuals $\{\mathsf{R}_\mathrm{f}, \mathsf{R}_\mathrm{r}\}$ are the differences between the real evaluation points $\tilde{\mathsf{Q}}$ and the measured values $\{\hat{\mathsf{Q}}_\mathrm{f}, \hat{\mathsf{Q}}_\mathrm{r}\}$. The Euclidean distance r_j between two associated points $\tilde{\boldsymbol{q}}_j$ and $\hat{\boldsymbol{q}}_j$ is a measure of quality.

value	**front target**	**rear target**
systematic error $\bar{r}$	0.3511 [mm]	0.3175 [mm]
standard deviation $s_{\boldsymbol{r}}$	0.2152 [mm]	0.1977 [mm]
standard error $s_{\bar{r}}$	0.0221 [mm]	0.0203 [mm]
standardized 2-norm $\|\,\boldsymbol{r}\,\|_2$	0.0422 [mm]	0.0383 [mm]

TABLE I
CROSS VALIDATION RESIDUALS WITH $m = 40$ AND $n = 95$.

V. MEASUREMENT PROCEDURE

The measurement process shown in Fig. 13 utilizes the transformation matrices $\{\mathsf{H}_\mathrm{f}, \mathsf{H}_\mathrm{r}\}$ to map the two camera points $\{\boldsymbol{p}_\mathrm{f}, \boldsymbol{p}_\mathrm{r}\}$ to their corresponding real points $\{\boldsymbol{q}_\mathrm{f}, \boldsymbol{q}_\mathrm{r}\}$, see Eqn. (1). A measurement rate of 1 [Hz] has been specified for the measurement instrument, whereby the selected camera is capable of acquiring 30 [fps]. Depending on this desired measurement rate, a robust measurement result can be achieved by taking a number of $n > 1$ individual images into account per camera for one measurement result. This is of particular importance, as the machine is victim of vibrations, shock and other sources of mechanical stress negatively influencing the measurement. According to the JCGM Guide to the Expression of Uncertainty in Measurement (GUM) [19], we can determine a confidence interval as following using a set Q of n measured points $\boldsymbol{q}_j$,

$$\mathsf{Q} = [\boldsymbol{q}_1 \ldots \boldsymbol{q}_n] = \begin{bmatrix} x_1 \ldots x_n \\ y_1 \ldots y_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}^\mathrm{T} \\ \boldsymbol{y}^\mathrm{T} \end{bmatrix}. \tag{12}$$

The mean values in x and y dimension are computed via

$$\bar{x} = \frac{1}{n}(\boldsymbol{x}^\mathrm{T}\,\boldsymbol{1}) \quad \text{and} \quad \bar{y} = \frac{1}{n}(\boldsymbol{y}^\mathrm{T}\,\boldsymbol{1}), \tag{13}$$

where $\boldsymbol{1} = [1 \ldots 1]^\mathrm{T}$ is a vector of ones. For repeated measurements of single values, the degree of freedom is $\nu = n - 1$. The variances $s_{\boldsymbol{x}}^2$ and $s_{\boldsymbol{y}}^2$ as well as the standard deviations of the means (standard error) $s_{\bar{x}}$ and $s_{\bar{x}}$ are

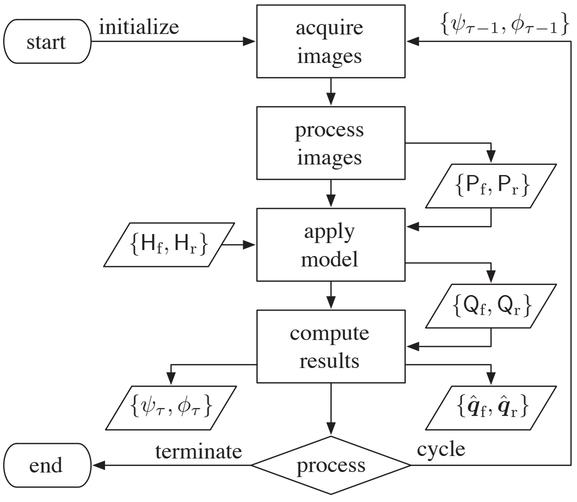
$$s_{\boldsymbol{x}}^2 = \frac{1}{\nu}(\boldsymbol{x} - \bar{x})^\mathrm{T}(\boldsymbol{x} - \bar{x}) \quad \text{and} \quad s_{\bar{x}} = \sqrt{\frac{s_{\boldsymbol{x}}^2}{n}}, \tag{14}$$

$$s_{\boldsymbol{y}}^2 = \frac{1}{\nu}(\boldsymbol{y} - \bar{y})^\mathrm{T}(\boldsymbol{y} - \bar{y}) \quad \text{and} \quad s_{\bar{y}} = \sqrt{\frac{s_{\boldsymbol{y}}^2}{n}}. \tag{15}$$

The complete measurement result $\hat{\boldsymbol{q}}$ including its confidence interval is then

$$\hat{\boldsymbol{q}} = \begin{bmatrix} \bar{x} \pm t_{(\nu,0.95)}\, s_{\bar{x}} \\ \bar{y} \pm t_{(\nu,0.95)}\, s_{\bar{y}} \end{bmatrix}. \qquad (16)$$

$t_{(\nu,0.95)}$ is the Student-t value of the inverse cumulative distribution function (ICDF) for a symmetric $95\,[\%]$ confidence interval, whereby $t_{(n-1,0.95)} = 2.05$ for $n = 30$. Consequently, an uncertainty of $< \pm 1\,[\text{mm}]$ is achievable during the measurement process. This is a valid approach, as the Gaussian nature of the measurement was confirmed during the calibration process. Having the results for $\{\hat{\boldsymbol{q}}_\text{f}, \hat{\boldsymbol{q}}_\text{r}\}$ available for the front and the rear target, the angles $\{\psi_\tau, \phi_\tau\}$ for the current iteration τ are calculated according to Eqn. (3). Plausibility checks are implemented with a feedback loop of previous measurement results. The mapping operation is a low-dimension matrix vector multiplication; consequently, the measurement process can be implemented in common programming languages such as C/C++, Java or C# for the final product. As long as the device remains in development, the MATLAB solution fulfills the requirements w.r.t. the sampling rate of $1\,[\text{Hz}]$.



start
initialize
acquire images
$\{\psi_{\tau-1}, \phi_{\tau-1}\}$
process images
$\{\mathsf{P}_\text{f}, \mathsf{P}_\text{r}\}$
$\{\mathsf{H}_\text{f}, \mathsf{H}_\text{r}\}$
apply model
$\{\mathsf{Q}_\text{f}, \mathsf{Q}_\text{r}\}$
compute results
$\{\psi_\tau, \phi_\tau\}$
$\{\hat{\boldsymbol{q}}_\text{f}, \hat{\boldsymbol{q}}_\text{r}\}$
end
terminate
process
cycle

Fig. 13. A number of n images are acquired for each target and measurement. The camera space coordinates $\{\mathsf{P}_\text{f}, \mathsf{P}_\text{r}\}$ are mapped to the real space coordinates $\{\mathsf{Q}_\text{f}, \mathsf{Q}_\text{r}\}$, which are statistically analyzed to compute the measurement results $\{\hat{\boldsymbol{q}}_\text{f}, \hat{\boldsymbol{q}}_\text{r}\}$ and the angles $\{\psi_\tau, \phi_\tau\}$ of the current iteration τ. They are compared with the angles of the previous iteration $\{\psi_{\tau-1}, \phi_{\tau-1}\}$ to perform a plausibility check and enable trend analysis.

VI. Conclusion

The fully integrated optical measurement device is robust in terms of machine vision and mechanic vibrations, making it suitable for underground construction projects. It was demonstrated, that the deployed projective transformation model is reasonably accurate with an uncertainty estimation of $< 0.6\,[\text{mm}]$ for a calibration setup using $m = 40$ calibration points and $n = 95$ cross validation points. The measurement procedure is computationally cheap. An uncertainty of $< \pm 1\,[\text{mm}]$ is expected within a $95\,[\%]$ confidence when using $n = 30$ individual images for a single measurement result while still ensuring a sampling rate of $1\,[\text{Hz}]$; in comparison, triangulation using retroreflectors has an associated uncertainty of $\pm 10\,[\text{mm}]$. The use of electro-active glass is an option for the front semitransparent glass target. The robustness of the system could be further increased by installing a second pair of redundant cameras for mutual monitoring of the laser spots. Besides the control of a TBM, the measurement device would be usable to control other large machinery working in harsh environments such as continuous miners or bolter miners.

Acknowledgment

We want to thank the company of Geodata[5] for the cooperative work and the supplied technical equipment.

References

[1] R. Neumayr, M. Harker, P. O'Leary, and J. Golser, "New approaches to machine vision based displacement analysis," in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*, May 2012, pp. 979–984.

[2] A. Soetedjo and E. Nurcahyo, "Developing of low cost vision-based shooting range simulator," *International Journal of Computer Science and Network Security*, vol. 11, no. 2, pp. 109–113, 2011.

[3] M. Herrenknecht, "Laser-controlled machines for microtunneling," in *Proceedings of the 8th ISARC, Germany*, 1991, pp. 789–800.

[4] S. Matsui and C. Zhang, "Alignment method for 50m distance using laser and ccd camera," in *Proceedings of 7th International Workshop on Accelerator Alignment. Japan: SPRING-8*, 2002, pp. 127–139.

[5] G. Retscher, "Multi-sensor systems for machine guidance and control," in *FIG XXII International Congress*, 2002, pp. 19–26.

[6] C. Gugg, P. O'Leary, and M. Harker, "Analytic multisource data fusion and its application to a large-scale optical PSD," *Instrumentation and Measurement, IEEE Transactions on*, vol. 63, no. 5, pp. 1116–1126, May 2014.

[7] J. B. Allen, "Laser beam rider guidance system," Sep. 5 1978, uS Patent 4,111,385.

[8] D. Kolymbas, *Geotechnik*. Springer, 1998.

[9] Leica Geosystems, "Theodolite TPS1200+," tech. spec., 2012. [Online]. Available: www.leica-geosystems.com

[10] Geodata GmbH, "TAUROS TBM-Steuerung," tech. spec., 2008. [Online]. Available: www.geodata.com

[11] S. Lipson, L. H., and T. D., *Optical Physics*. Cambridge University Press, Cambridge, 1995.

[12] Basler, "aca1300-30gm," tech. spec., 2014. [Online]. Available: www.baslerweb.com/products/ace.html

[13] Kowa, "LM8JCM," tech. spec., 2014. [Online]. Available: www.kowa.eu/fa/de/LM8JCM.php

[14] A. Goshtasby, "Piecewise linear mapping functions for image registration," *Pattern Recognition*, vol. 19, no. 6, pp. 459 – 466, 1986. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0031320386900440

[15] C. Gugg, M. Harker, and P. O'Leary, "Structural deformation measurement via efficient tensor polynomial calibrated electro-active glass targets," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2013, pp. 86 610F–86 610F.

[16] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[17] iTK Dr. Kassen GmbH, "ST9," tech. spec., 2014. [Online]. Available: www.itknet.de

[18] M. Harker and P. O'Leary, "Computation of homographies." in *BMVC*, 2005.

[19] JCGM 100:2008, "ISO/IEC Guide 98-3:2008: Evaluation of measurement data guide to the expression of uncertainty in measurement (GUM)," international standard.

[5] Geodata GmbH, Leoben, Austria, www.geodata.at

14.2 IEEE - Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA) 2014

May 5 - May 7, 2014 @ Ottawa, Canada [72]

Citation[3]

```
@inproceedings{GuggHarkerOleary2014civemsa,
 author     = {Gugg, Christoph and Harker, Matthew and O'Leary, Paul},
 title      = {Virtual Calibration Environment for A-Priori
              Estimation of Measurement Uncertainty},
 booktitle  = {IEEE International Conference on Computational
              Intelligence and Virtual Environments for
              Measurement Systems and Applications (CIVEMSA)},
 year       = {2014},
 pages      = {52-57},
 doi        = {10.1109/CIVEMSA.2014.6841438}}
```

Keywords

regression; discrete unitary polynomials; measurement uncertainty; uncertainty estimation; virtual calibration; confidence interval; prediction interval; cross validation;

[3] IEEE Xplore Digital Library, `ieeexplore.ieee.org`

Virtual Calibration Environment for A-Priori Estimation of Measurement Uncertainty

Christoph Gugg, Matthew Harker and Paul O'Leary
Chair of Automation, Department Product Engineering, University of Leoben, Austria
Email: christoph.gugg@unileoben.ac.at, Web: automation.unileoben.ac.at

***Abstract*—During product engineering of a measuring instrument, the question is which measures are necessary to achieve the highest possible measurement accuracy. In this context, a measuring instrument's target uncertainty is an essential part of its requirement specifications, because it is an indicator for the measurement's overall quality. This paper introduces an algebraic framework to determine the confidence and prediction intervals of a calibration curve; the matrix based framework greatly simplifies the associated proofs and implementation details. The regression analysis for discrete orthogonal polynomials is derived, and new formulae for the confidence and prediction intervals are presented for the first time. The orthogonal basis functions are numerically more stable and yield more accurate results than the traditional polynomial Vandermonde basis; the methods are thereby directly compared. The new virtual environment for measurement and calibration of cyber-physical systems is well suited for establishing the error propagation chain through an entire measurement system, including complicated tasks such as data fusion. As an example, an adaptable virtual lens model for an optical measurement system is established via a reference measurement. If the same hardware setup is used in different systems, the uncertainty can be estimated a-priori to an individual system's calibration, making it suitable for industrial applications. With this model it is possible to determine the number of required calibration nodes for system level calibration in order to achieve a predefined measurement uncertainty. Hence, with this approach, systematic errors can be greatly reduced and the remaining random error is described by a probabilistic model. Verification is performed via numerical experiments using a non-parametric Kolmogorov-Smirnov test and Monte Carlo simulation.**

***Keywords*—regression, discrete unitary polynomials, measurement uncertainty, uncertainty estimation, virtual calibration**

I. Introduction and Motivation

It is required to calibrate devices for measurements on system level when their behavior cannot be described analytically. A set of n a-priori known calibration nodes $\boldsymbol{x}$ is provided and the measuring instrument acquires the corresponding measured quantity values $\boldsymbol{y}$. The goal is to establish a mathematical relation $\boldsymbol{y} = f(\boldsymbol{x})$ between both data sets, i.e., a regression. The process yields the characteristic calibration curve of a measurement system. This approach is completely general for most types of sensors, no matter the physical phenomenon serving as a basis for a measurement. Each measurement induces an inevitable measurement error $\boldsymbol{\epsilon}$. According to the ISO/BIPM *Guide to the Expression of Uncertainty in Measurement* (GUM) [1], no measurement result is valid without a statement about its associated uncertainty. The instrument's uncertainty is a key indicator for the measurement's quality, i.e., the achievable accuracy. Although measurement uncertainty cannot be avoided, it can be limited to an acceptable amount within the laws of physics. The calibration process can be time-consuming and, therefore, be expensive. Consequently, estimating the expected uncertainty before the actual calibration procedure is performed is of great interest. The main parameter for the uncertainty, which can be influenced by the operator, is the number of calibration nodes n. It is possible to determine n for achieving a required target measurement uncertainty. Due to the growing importance of cyber-physical systems [2]–[4], connecting virtual and physical space through a suitable mathematical model becomes vital for efficient measurement applications. In this context, the main contributions of this paper are:

1) A general derivation of non-simultaneous confidence and prediction intervals for unitary polynomial Gram basis functions and a direct comparison to the common Vandermonde basis. The Gram polynomials yield more stable and accurate estimates than traditional methods. The matrix based approach to regression and error analysis facilitates a straightforward implementation of error propagation throughout an entire measuring system;
2) Numerical experiments with a Monte Carlo simulation and Kolmogorov-Smirnov tests to demonstrate how the number of calibration points affects the overall measurement uncertainty. The mathematical framework is verified with artificial and real data sets acquired from a machine vision application employing distorted optical lenses.

Note that this paper only covers a polynomial approach for curve fitting, other methods like piecewise, exponential or rational fitting techniques are not part of the scope.

II. Conventional Polynomial Regression and Associated Confidence and Prediction Intervals

The state-of-the art approach for determining a measuring system's calibration curve is regression with polynomials of degree d. For sensors with linear behavior, $d = 1$, and for sensors with non-linear behavior, $d > 1$. For $n > d + 1$, the approach delivers an overestimated system of linear equations of the form

$$\boxed{\boldsymbol{y} = \mathsf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},} \tag{1}$$

when $\boldsymbol{y} = [y_1, \ldots, y_i, \ldots, y_n]^{\mathrm{T}}$ is the solution vector of measured quantity values; X is the polynomial basis set of degree d; $\boldsymbol{\beta} = [\beta_d, \beta_{d-1}, \ldots, \beta_0]^{\mathrm{T}}$ is the coefficient vector; and $\boldsymbol{\epsilon} = [\epsilon_1, \ldots, \epsilon_i, \ldots, \epsilon_n]^{\mathrm{T}}$ is the error vector. It is assumed, that the error $\boldsymbol{\epsilon}$ is independent, identically distributed (i.i.d.) with

Gaussian noise and uncorrelated to $\boldsymbol{x}$. Hence, the homoscedasticity condition is fulfilled, i.e., the variance is constant within the data set. In most applications[1], a Vandermonde basis set with the design matrix

$$\mathsf{X} = \begin{bmatrix} x_1^d & x_1^{(d-1)} & \dots & x_1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_n^d & x_n^{(d-1)} & \dots & x_n & 1 \end{bmatrix} \tag{2}$$

is employed. The basis X is complete, when $d = n - 1$, but commonly $n \gg d$. This approach is analytically fine, but numerically unstable if polynomial degrees of $d > 7$ are used [5], [6]. The exponential growth of the condition number $\kappa(\mathsf{X})$ is visualized in Fig. 1. The common approach to estimate the coefficient vector $\hat{\boldsymbol{\beta}}$ of the regression is the usage of the Moore-Penrose pseudo inverse $\mathsf{X}^{+} \triangleq (\mathsf{X}^{\mathrm{T}}\mathsf{X})^{-1}\mathsf{X}^{\mathrm{T}}$,

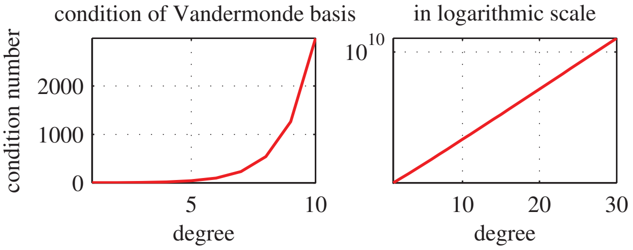


Fig. 1: The Vandermonde basis is numerically unstable for degrees of $d > 7$ and the system of equations becomes ill-conditioned. For the data set shown in Fig. 2, the condition number of X is approximately $\kappa(\mathsf{X}) = 3000$ for $d = 10$. The logarithmic plot illustrates the exponential growth of $\kappa(\mathsf{X})$. On the other hand, a unitary polynomial basis G ensure a condition number of $\kappa(\mathsf{G}) = 1$ at any given degree.

$$\boxed{\hat{\boldsymbol{\beta}} = \mathsf{X}^{+}\boldsymbol{y} \quad \text{with} \quad \mathbb{E}[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta},} \tag{3}$$

when X^{+} is a maximum likelihood estimator [7] for the generalized inverse. The operator $\mathbb{E}[\cdot]$ is the expected value. The estimator $\hat{\boldsymbol{y}}$ for the true solution vector $\boldsymbol{y}$ is then

$$\hat{\boldsymbol{y}} = \mathsf{X}\hat{\boldsymbol{\beta}} = \mathsf{X}\mathsf{X}^{+}\boldsymbol{y} \quad \text{with} \tag{4}$$

$$\mathbb{E}[\hat{\boldsymbol{y}}] = \mathbb{E}[\mathsf{X}\hat{\boldsymbol{\beta}}] = \mathsf{X}\,\mathbb{E}[\hat{\boldsymbol{\beta}}] = \mathsf{X}\boldsymbol{\beta} = \boldsymbol{y}. \tag{5}$$

In literature [8]–[10], the confidence and the prediction for one point of the regression utilizing a Vandermonde basis is computed via

1) non-simultaneous confidence interval:

$$y_i \leq \hat{y}_i \pm t_{(\nu)}\, s\, \sqrt{\boldsymbol{x}_i \mathsf{V} \boldsymbol{x}_i^{\mathrm{T}}}, \tag{6}$$

2) non-simultaneous prediction interval:

$$\tilde{y}_i \leq \hat{y}_i \pm t_{(\nu)}\, s\, \sqrt{\boldsymbol{x}_i \mathsf{V} \boldsymbol{x}_i^{\mathrm{T}} + 1}, \tag{7}$$

when using a maximum likelihood estimation (MLE) with ordinary least squares (OLS) for linear regression. This approach is justifiable via the *Gauss-Markov theorem* [8]: within the class of linear unbiased estimators, the least squares estimator has minimum variance[2]. The matrix $\mathsf{V} = (\mathsf{X}^{\mathrm{T}}\mathsf{X})^{-1}$. $t_{(\nu)}$ is the inverse cumulative distribution function (ICDF or quantile) of the Student-t distribution with ν degrees of freedom (d.f.). For each regressor (constraint) one d.f. is lost, consequently,

$$\boxed{\nu = n - d - 1,} \tag{8}$$

when d is equal to the number of fitted parameters. In the case of regression, d is the degree of the fitted polynomial, see also Fig. 4 for a visualization of the effect of increasing d.f. on the Student-t factor. In other words, the d.f. represent the number of additional measurements, i.e., there are no d.f. for a complete basis and there are infinite confidence and prediction intervals in this case. For easier readability, a symmetric $(1 - \alpha) = 95\%$ confidence interval with significance level $\alpha = 5\%$ is chosen for all ICDFs in this paper. The sample standard deviation s is an estimator for the population's true standard deviation σ, see Eqn. 16. Both, $t_{(\nu)}$ and s, act as scaling factors for the intervals, see Fig. 2 for an example.

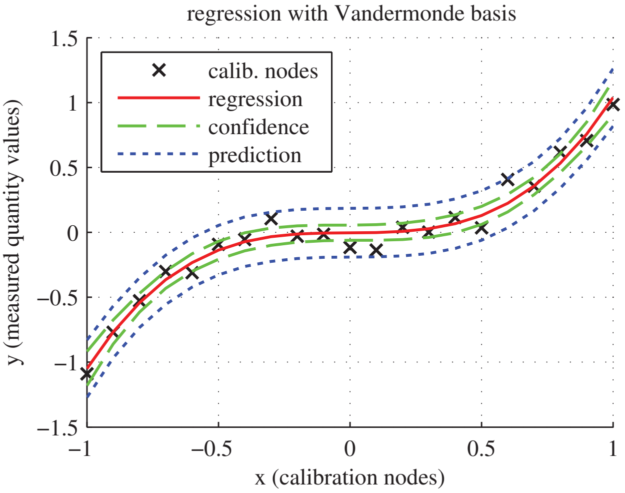


Fig. 2: The plot shows the function $y = x^3$ with artificial Gaussian error $\mathcal{N}(\boldsymbol{\epsilon}, 0, 0.01)$ in the interval $[-1, +1]$ with $n = 21$ calibration nodes (black), which has been generated with Monte Carlo simulation. The calibration curve (red solid) is computed via polynomial regression with degree $d = 3$. The 95% confidence (green dashed) and prediction (blue dotted) intervals are estimators for the expected measurement uncertainty. Both intervals form envelopes around the curve.

III. Regression Using Matrix Algebra

In most literature [8]–[10], the formulas for the confidence and prediction intervals in Eqn. 6 and 7 are derived for the univariate case with Vandermonde basis. In this section, the theoretical background for derivations in the multivariate case is established.

[1] MATLAB's Curve Fitting Toolbox also uses a Vandermonde basis.

[2] This concept is also known as *best linear unbiased estimator* (BLUE).

A. Multivariate Descriptive Statistics

In matrix notation, the expected value of a vector $\boldsymbol{u}$ is

$$\mathbb{E}[\boldsymbol{u}] = [\mathbb{E}[u_1], \ldots, \mathbb{E}[u_i], \ldots, \mathbb{E}[u_n]]^{\mathrm{T}} \tag{9}$$

and the covariance matrix $\Lambda_{\boldsymbol{v}}$ of a vector $\boldsymbol{v}$ is

$$\Lambda_{\boldsymbol{v}} = \mathbb{E}\left[(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])(\boldsymbol{v} - \mathbb{E}[\boldsymbol{v}])^{\mathrm{T}}\right]. \tag{10}$$

If $\boldsymbol{v} = \mathsf{L}\boldsymbol{u}$, when L is an arbitrary linear operator, then the output's covariance $\Lambda_{\boldsymbol{v}}$ is directly computable from the input's covariance $\Lambda_{\boldsymbol{u}}$, because $\mathbb{E}[\mathsf{L}] = \mathsf{L}$.

$$\begin{aligned}\Lambda_{\boldsymbol{v}} &= \mathbb{E}[(\mathsf{L}\boldsymbol{u} - \mathbb{E}[\mathsf{L}\boldsymbol{u}])(\mathsf{L}\boldsymbol{u} - \mathbb{E}[\mathsf{L}\boldsymbol{u}])^{\mathrm{T}}],\\ &= \mathbb{E}[(\mathsf{L}\boldsymbol{u} - \mathsf{L}\,\mathbb{E}[\boldsymbol{u}])(\mathsf{L}\boldsymbol{u} - \mathsf{L}\,\mathbb{E}[\boldsymbol{u}])^{\mathrm{T}}],\\ &= \mathsf{L}\,\mathbb{E}[(\boldsymbol{u} - \mathbb{E}[\boldsymbol{u}])(\boldsymbol{u} - \mathbb{E}[\boldsymbol{u}])^{\mathrm{T}}]\,\mathsf{L}^{\mathrm{T}},\\ &= \mathsf{L}\Lambda_{\boldsymbol{u}}\mathsf{L}^{\mathrm{T}}.\end{aligned} \tag{11}$$

It is assumed, that the error $\boldsymbol{\epsilon}$ is i.i.d and uncorrelated to $\boldsymbol{x}$. The multivariate Gaussian probability density function (PDF) is $\mathcal{N}(\boldsymbol{\epsilon}, \boldsymbol{0}, \Lambda_{\boldsymbol{\epsilon}})$, i.e., the expectation and covariance of the error are

$$\mathbb{E}[\boldsymbol{\epsilon}] = \boldsymbol{0} \quad \text{and} \quad \Lambda_{\boldsymbol{\epsilon}} = \sigma^2\mathsf{I}, \tag{12}$$

because $\mathrm{cov}[\epsilon_i, \epsilon_j] = 0$ when $i \neq j$. Hence,

$$\Lambda_{\boldsymbol{\epsilon}} = \begin{bmatrix}\mathrm{cov}[\epsilon_1,\epsilon_1] & \mathrm{cov}[\epsilon_1,\epsilon_2] & \ldots & \mathrm{cov}[\epsilon_1,\epsilon_n]\\ \mathrm{cov}[\epsilon_2,\epsilon_1] & \mathrm{cov}[\epsilon_2,\epsilon_2] & \ldots & \mathrm{cov}[\epsilon_2,\epsilon_n]\\ \vdots & \vdots & \ddots & \vdots\\ \mathrm{cov}[\epsilon_n,\epsilon_1] & \mathrm{cov}[\epsilon_n,\epsilon_2] & \ldots & \mathrm{cov}[\epsilon_n,\epsilon_n]\end{bmatrix} \tag{13}$$

$$= \begin{bmatrix}\sigma^2 & 0 & \ldots & 0\\ 0 & \sigma^2 & \ldots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \ldots & \sigma^2\end{bmatrix} = \sigma^2\mathsf{I}. \tag{14}$$

The covariance of $\boldsymbol{\epsilon}$ is equivalent to the covariance of $\boldsymbol{y}$, i.e.,

$$\Lambda_{\boldsymbol{y}} = \Lambda_{\boldsymbol{\epsilon}} = \sigma^2\mathsf{I}. \tag{15}$$

When working with artificial data generated by Monte Carlo simulation, it's recommended to verify this condition for both covariances via the Kolmogorov-Smirnov test, see Fig. 3. The unbiased multivariate estimator s^2 for the population's variance σ^2 with ν d.f. is

$$\boxed{s^2 = \frac{1}{\nu}(\boldsymbol{y} - \hat{\boldsymbol{y}})^{\mathrm{T}}(\boldsymbol{y} - \hat{\boldsymbol{y}})} \tag{16}$$

and the estimator for the standard deviation σ is $s = +\sqrt{s^2}$.

B. Multivariate Goodness-of-Fit Test

The χ^2 goodness-of-fit test with r d.f. for the multivariate variable $\hat{\boldsymbol{v}} = \mathsf{L}\hat{\boldsymbol{u}}$ with $\mathbb{E}[\hat{\boldsymbol{v}}] = \boldsymbol{v}$ and $\mathbb{E}[\hat{\boldsymbol{u}}] = \boldsymbol{u}$ is

$$\chi^2_{(r)} \leq (\hat{\boldsymbol{v}} - \mathbb{E}[\hat{\boldsymbol{v}}])^{\mathrm{T}}\Lambda_{\hat{v}}{}^{-1}(\hat{\boldsymbol{v}} - \mathbb{E}[\hat{\boldsymbol{v}}]), \tag{17}$$

i.e., the vector product $(\hat{\boldsymbol{v}} - \mathbb{E}[\hat{\boldsymbol{v}}])^{\mathrm{T}}(\hat{\boldsymbol{v}} - \mathbb{E}[\hat{\boldsymbol{v}}])$ is standardized with the covariance $\Lambda_{\hat{v}}$. Utilizing Eqn. 11 delivers

$$\begin{aligned}\chi^2_{(r)} &\leq (\mathsf{L}\hat{\boldsymbol{u}} - \mathbb{E}[\mathsf{L}\hat{\boldsymbol{u}}])^{\mathrm{T}}\left(\mathsf{L}\Lambda_{\hat{u}}\mathsf{L}^{\mathrm{T}}\right)^{-1}(\mathsf{L}\hat{\boldsymbol{u}} - \mathbb{E}[\mathsf{L}\hat{\boldsymbol{u}}]),\\ &\leq (\mathsf{L}\hat{\boldsymbol{u}} - \mathsf{L}\boldsymbol{u})^{\mathrm{T}}\left(\mathsf{L}\Lambda_{\hat{u}}\mathsf{L}^{\mathrm{T}}\right)^{-1}(\mathsf{L}\hat{\boldsymbol{u}} - \mathsf{L}\boldsymbol{u}).\end{aligned} \tag{18}$$

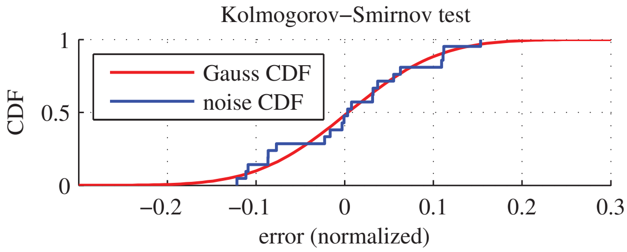


Fig. 3: The non-parametric Kolmogorov-Smirnov test (KS-test) [9] verifies the Gaussian nature of the noise $\boldsymbol{\epsilon}$ and the regression's residual $\boldsymbol{r} = |\boldsymbol{y} - \hat{\boldsymbol{y}}|$ on a given significance level α, typically $(1 - \alpha) = 95\%$. The test's principle is based on comparing the sample's (blue) cumulative distribution function (CDF) with the CDF of the standard normal distribution (red).

When estimating σ^2 with s^2, then the $\chi^2_{(r)}$ distribution can be approximated with the quasi-standardized Fisher-$\mathcal{F}$ distribution $r\,\mathcal{F}_{(r,\nu)}$, because

$$\boxed{\lim_{\nu\to\infty} r\,\mathcal{F}_{(r,\nu)} = \chi^2_{(r)}.} \tag{19}$$

If the number or regressors $r = 1$, then the Fisher-$\mathcal{F}$ distribution is equivalent to a squared Student-t distribution

$$\boxed{\mathcal{F}_{(1,\nu)} = t^2_{(\nu)}.} \tag{20}$$

Note that $\mathcal{F}_{(1,\nu)} \neq \mathcal{F}_{(\nu,1)}$. A Student-$t$ distribution converges to a Gaussian distribution Φ when $\nu \to \infty$, this is shown for the ICDF in Fig. 4,

$$\boxed{\lim_{\nu\to\infty} t_{(\nu)} = \Phi.} \tag{21}$$

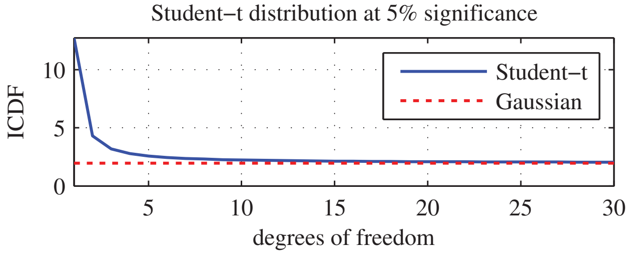


Fig. 4: During measurements, only a finite number n of sample values is acquired. The Student-t distribution takes care of this inexactness. With increasing d.f., the Student factor $t_{(\nu)}$ decreases. Theoretically, if $n \to \infty$, then the value of the Student's ICDF (blue) converges to the corresponding Gaussian ICDF (red dashed) at a given significance $\alpha = 5\%$.

C. Fundamental Matrix Operations

When computing the non-simultaneous bounds, it is necessary to select individual rows of the basis sets. The i^{th} row of a matrix U is selected via the coordinate vector

$e_i = [0, \dots, 1, \dots, 0]^{\mathrm{T}}$, i.e., a vector of zeros except for an 1 on the i^{th} position. The operation is $u_i = e_i^{\mathrm{T}}\mathsf{U}$, which is equivalent to the MATLAB syntax $u_i = \mathsf{U}(i,:)$. For the parallel computation of individual values it is required to select the diagonal elements of an arbitrary $n \times n$ matrix U, which is denoted by $\mathrm{diag}(\mathsf{U}) = [u_{11}, u_{22}, \dots, u_{ii}, \dots, u_{nn}]^{\mathrm{T}}$. For the identity matrix I, the operation delivers a vector of ones, i.e. $\mathrm{diag}(\mathsf{I}) = \mathbf{1}$.

IV. Discrete Unitary Polynomial Basis Functions

O'Leary et al. [11]–[13] introduced a numerically stable and computationally efficient discrete unitary basis G derived from Gram polynomials. It is proven, that there is one and only one unitary polynomial basis which can be generated from a sum of uncorrelated monomials by a recurrence relationship. The exact Legendre, Chebyshev and normalized Krawtchouk polynomial basis sets are equivalent and only the norms of the individual polynomials are different, i.e. the weighting of the inner product. The orthogonality condition ensures minimal information redundancy. For applications utilizing regression, the degree d of the basis is $d < n - 1$, when n is the number of sampled points, i.e., a filtering operation via spectral regularization is performed. Hence, the basis G is incomplete[3] and $\mathsf{G}\mathsf{G}^{\mathrm{T}} \neq \mathsf{I}$. By definition, an incomplete unitary discrete basis posses following traits,

$$\mathsf{G}^{\mathrm{T}}\mathsf{G} = \mathsf{I} \quad \text{and} \quad \mathsf{G}^{+} = \mathsf{G}^{\mathrm{T}}. \tag{22}$$

The orthogonality measure $P^{\perp} = \mathsf{G}^{\mathrm{T}}\mathsf{G} - \mathsf{I}$ should be exactly a matrix of zeros. However, due to numerical errors, this is not exactly the case, i.e. $P^{\perp} \approx 0$. However, $P^{\perp}$ is a measure for the quality of the generated basis function G. The number of significant digits can be estimated by $\eta = \log_{10}(\| P \|_{\mathrm{F}})$, when $\| \cdot \|_{\mathrm{F}}$ is the Frobenius norm. For a basis G with $d = 100$, there are still $\eta \approx 14.5$ significant digits. The solution vector $\boldsymbol{y}$ is formulated as

$$\boxed{\boldsymbol{y} = \mathsf{G}\boldsymbol{\alpha} + \boldsymbol{\epsilon},} \tag{23}$$

where $\boldsymbol{\alpha}$ is the coefficient vector and $\boldsymbol{\epsilon}$ is the i.i.d. error vector with $\mathbb{E}[\boldsymbol{\epsilon}] = \mathbf{0}$ and $\Lambda_{\boldsymbol{\epsilon}} = \sigma^2\mathsf{I}$. Thanks to the orthogonality condition, the coefficients $\boldsymbol{\alpha}$ can effectively estimated via

$$\boxed{\hat{\boldsymbol{\alpha}} = \mathsf{G}^{\mathrm{T}}\boldsymbol{y} \quad \text{with} \quad \mathbb{E}[\hat{\boldsymbol{\alpha}}] = \boldsymbol{\alpha}.} \tag{24}$$

The estimator $\hat{\boldsymbol{y}}$ for the solution vector $\boldsymbol{y}$ is then

$$\hat{\boldsymbol{y}} = \mathsf{G}\hat{\boldsymbol{\alpha}} = \mathsf{G}\mathsf{G}^{\mathrm{T}}\boldsymbol{y}. \tag{25}$$

V. Confidence and Prediction Intervals for a Discrete Orthogonal Basis

The covariance matrix $\Lambda_{\hat{\boldsymbol{\alpha}}}$ for the estimated coefficients $\hat{\boldsymbol{\alpha}}$ is computed from $\Lambda_{\boldsymbol{y}}$ by utilizing Eqn. 11 and Eqn. 24,

$$\begin{aligned} \Lambda_{\hat{\boldsymbol{\alpha}}} &= \mathsf{L}\Lambda_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}} \quad \text{with} \quad \mathsf{L} = \mathsf{G}^{\mathrm{T}}, \\ &= \mathsf{G}^{\mathrm{T}}\Lambda_{\boldsymbol{y}}\mathsf{G} \quad \text{and} \quad \Lambda_{\boldsymbol{y}} = \sigma^2\mathsf{I}, \\ &= \mathsf{G}^{\mathrm{T}}(\sigma^2\mathsf{I})\mathsf{G}, \\ &= \sigma^2\mathsf{G}^{\mathrm{T}}\mathsf{G} \quad \text{and} \quad \mathsf{G}^{\mathrm{T}}\mathsf{G} = \mathsf{I}, \\ &= \sigma^2\mathsf{I}. \end{aligned} \tag{26}$$

Using Eqn. 25, the covariance matrix $\Lambda_{\hat{\boldsymbol{y}}}$ for the estimated solution $\hat{\boldsymbol{y}}$ can be derived from $\Lambda_{\boldsymbol{y}}$ by

$$\begin{aligned} \Lambda_{\hat{\boldsymbol{y}}} &= \mathsf{L}\Lambda_{\boldsymbol{y}}\mathsf{L}^{\mathrm{T}} \quad \text{with} \quad \mathsf{L} = \mathsf{G}\mathsf{G}^{\mathrm{T}}, \\ &= (\mathsf{G}\mathsf{G}^{\mathrm{T}})\Lambda_{\boldsymbol{y}}(\mathsf{G}\mathsf{G}^{\mathrm{T}})^{\mathrm{T}}, \\ &= \mathsf{G}\mathsf{G}^{\mathrm{T}}\Lambda_{\boldsymbol{y}}\mathsf{G}^{\mathrm{T}}\mathsf{G} \quad \text{and} \quad \mathsf{G}^{\mathrm{T}}\mathsf{G} = \mathsf{I}, \\ &= \mathsf{G}\mathsf{G}^{\mathrm{T}}\Lambda_{\boldsymbol{y}} \quad \text{and} \quad \Lambda_{\boldsymbol{y}} = \sigma^2\mathsf{I}, \\ &= \sigma^2\mathsf{G}\mathsf{G}^{\mathrm{T}}. \end{aligned} \tag{27}$$

A. Simultaneous Confidence Intervals for the Coefficients

The interactive relation between all $(d+1)$ coefficients is considered when computing the simultaneous confidence intervals. Therefore, a $\chi^2_{(d+1)}$ goodness-of-fit test as presented in Eqn. 17 is carried out with $r = d + 1$ d.f.,

$$\begin{aligned} \chi^2_{(d+1)} &\leq (\hat{\boldsymbol{\alpha}} - \mathbb{E}[\hat{\boldsymbol{\alpha}}])^{\mathrm{T}}\Lambda_{\hat{\boldsymbol{\alpha}}}{}^{-1}(\hat{\boldsymbol{\alpha}} - \mathbb{E}[\hat{\boldsymbol{\alpha}}]), \\ \chi^2_{(d+1)} &\leq (\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha})^{\mathrm{T}}(\sigma^2\mathsf{I})^{-1}(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}). \end{aligned} \tag{28}$$

Estimating σ^2 with s^2 and approximating $\chi^2_{(d+1)}$ according to Eqn. 19 with $(d+1)\,\mathcal{F}_{(d+1,\nu)}$ yields

$$s^2\,(d+1)\,\mathcal{F}_{(d+1,\nu)} \geq (\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha})^{\mathrm{T}}(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}), \tag{29}$$

which described a hyper-ellipsoid with $(d+1)$ dimensions. The solution of this equation is an inverse problem. Thanks to the orthogonality condition, the covariance is $\Lambda_{\hat{\boldsymbol{\alpha}}} = \sigma^2\mathsf{I}$. This reduces the computation to a scalar product. The focus of this paper lies on the numerical solution of the non-simultaneous intervals.

B. Non-Simultaneous Confidence Intervals for the Coefficients

When computing the non-simultaneous intervals, each coefficient's confidence is computed individually; hence, only $r = 1$ d.f. is required for the $\chi^2_{(1)}$ goodness-of-fit test. Therefore, $\chi^2_{(1)}$ can be approximated with $1\,\mathcal{F}_{(1,\nu)}$. With this simplifications, Eqn. 29 yields

$$\begin{aligned} (\hat{\alpha}_i - \alpha_i)^2 &\leq \mathcal{F}_{(1,\nu)}\,s^2, \\ (\hat{\alpha}_i - \alpha_i)^2 &\leq t^2_{(\nu)}\,s^2, \\ |\hat{\alpha}_i - \alpha_i| &\leq t_{(\nu)}\,s, \\ \alpha_i &\leq \hat{\alpha}_i \pm t_{(\nu)}\,s. \end{aligned} \tag{30}$$

Parallel computing delivers the individual non-simultaneous confidence intervals for all coefficients

$$\boldsymbol{\alpha} \leq \hat{\boldsymbol{\alpha}} \pm t_{(\nu)}\,s\,\mathbf{1}. \tag{31}$$

C. Non-Simultaneous Confidence Intervals for the Regression

The complete curve's regression is computed via $\hat{\boldsymbol{y}} = \mathsf{G}\hat{\boldsymbol{\alpha}}$. However, each coefficient's confidence is still computed separately, therefore the $\chi^2_{(1)}$ test has $r = 1$ d.f.,

$$\begin{aligned} \chi^2_{(1)} &\leq (\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}])^{\mathrm{T}}\Lambda_{\hat{\boldsymbol{y}}}{}^{-1}(\hat{\boldsymbol{y}} - \mathbb{E}[\hat{\boldsymbol{y}}]), \\ &\leq (\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}}(\mathsf{G}\Lambda_{\hat{\boldsymbol{\alpha}}}\mathsf{G}^{\mathrm{T}})^{-1}(\hat{\boldsymbol{y}} - \boldsymbol{y}), \\ &\leq (\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}}(\mathsf{G}(\sigma^2\mathsf{I})\mathsf{G}^{\mathrm{T}})^{-1}(\hat{\boldsymbol{y}} - \boldsymbol{y}), \end{aligned}$$

[3] If G is complete with $d = n - 1$, then $\mathsf{G}^{-1} = \mathsf{G}^{+} = \mathsf{G}^{\mathrm{T}}$ and $\mathsf{G}\mathsf{G}^{\mathrm{T}} = \mathsf{I}$.

computing only the i^{th} point via $\boldsymbol{g}_i = \boldsymbol{e}_i^{\text{T}}\mathsf{G}$ yields

$$\chi^2_{(1)} \leq (\hat{y}_i - y_i)^2 (\sigma^2\, \boldsymbol{g}_i \boldsymbol{g}_i^{\text{T}})^{-1}, \tag{32}$$

approximation of the $\chi^2_{(1)}$ distribution with the $t_{(\nu)}$ distribution simplifies the terms to

$$\begin{aligned}(\hat{y}_i - y_i)^2 &\leq \chi^2_{(1)}(\sigma^2 \boldsymbol{g}_i \boldsymbol{g}_i^{\text{T}}),\\ &\leq \mathcal{F}_{(1,\nu)}(s^2 \boldsymbol{g}_i \boldsymbol{g}_i^{\text{T}}),\\ &\leq t^2_{(\nu)}(s^2 \boldsymbol{g}_i \boldsymbol{g}_i^{\text{T}}).\end{aligned} \tag{33}$$

Computing the absolute value delivers

$$|\hat{y}_i - y_i| \leq \pm\sqrt{t^2_{(\nu)}(s^2 \boldsymbol{g}_i \boldsymbol{g}_i^{\text{T}})} \tag{34}$$

and the confidence y_i of one regression point $\hat{y}_i$ is then

$$y_i \leq \hat{y}_i \pm t_{(\nu)}\, s\, \sqrt{\boldsymbol{g}_i \boldsymbol{g}_i^{\text{T}}}. \tag{35}$$

Parallel computing yields the non-simultaneous confidence intervals $\boldsymbol{y}$ for all points of the regression

$$\boldsymbol{y} \leq \hat{\boldsymbol{y}} \pm t_{(\nu)}\, s\, \sqrt{\operatorname{diag}(\mathsf{G}\mathsf{G}^{\text{T}})}. \tag{36}$$

D. Non-Simultaneous Prediction Intervals for the Regression

For the estimation of the predicted value $\tilde{y}_i$ of an individual function value y_i, the error's variance for one observation must also be considered, i.e., the variance σ^2 is added to the second term of Eqn. 32.

$$\chi^2_{(1)} \leq (\hat{y}_i - \tilde{y}_i)^2 (\sigma^2\, \boldsymbol{g}_i \boldsymbol{g}_i^{\text{T}} + \sigma^2)^{-1}, \tag{37}$$

performing the same steps as in the previous section delivers

$$(\hat{y}_i - \tilde{y}_i)^2 \leq t^2_{(\nu)}(s^2 \boldsymbol{g}_i \boldsymbol{g}_i^{\text{T}} + s^2) \tag{38}$$

and the prediction $\tilde{y}_i$ of one regression point $\hat{y}_i$ is then

$$\tilde{y}_i \leq \hat{y}_i \pm t_{(\nu)}\, s\, \sqrt{\boldsymbol{g}_i \boldsymbol{g}_i^{\text{T}} + 1}. \tag{39}$$

Parallel computing yields the non-simultaneous prediction intervals $\tilde{\boldsymbol{y}}$ for all points of the regression

$$\tilde{\boldsymbol{y}} \leq \hat{\boldsymbol{y}} \pm t_{(\nu)}\, s\, \sqrt{\operatorname{diag}(\mathsf{G}\mathsf{G}^{\text{T}}) + 1}. \tag{40}$$

The influence of the $t_{(\nu)}$ and s factors on the intervals has already been explained in Sec. II. The terms $\sqrt{\boldsymbol{g}_i \boldsymbol{g}_i^{\text{T}}}$ and $\sqrt{\boldsymbol{g}_i \boldsymbol{g}_i^{\text{T}} + 1}$ respectively further decrease the confidence and prediction intervals with increasing d.f., the effect is shown in Fig. 5.

VI. Virtual Optical Calibration Model

There are industrial implementations of optical measurement systems [14]–[16] using the provided basis functions [17], [18] in univariate as well as bivariate formulations. In this paper, a virtual lens model is established. Based on this model, the lower bounds of uncertainty for an optical measuring system can be estimated. The process steps are:

1) A physical lens and camera setup is utilized to acquire the initial reference data. Wide-angle lenses cause significant non-linear distortion, and as a consequence, rectification of the camera space is required. The center coordinates D_x and D_y of LEDs organized in a two-dimensional array act as calibration nodes, see Fig. 6.
2) The virtual lens model M_x and M_y is synthesized using the reference data via a bivariate tensor product,

$$\mathsf{S}_x = \mathsf{G}_y^{\text{T}} \mathsf{D}_x \mathsf{G}_x \quad \text{and} \quad \mathsf{M}_x = \mathsf{G}_y \mathsf{S}_x \mathsf{G}_x^{\text{T}}, \tag{41}$$

$$\mathsf{S}_y = \mathsf{G}_y^{\text{T}} \mathsf{D}_y \mathsf{G}_x \quad \text{and} \quad \mathsf{M}_y = \mathsf{G}_y \mathsf{S}_y \mathsf{G}_x^{\text{T}}, \tag{42}$$

where S_x and S_y are the spectra of coefficients, G_x and G_y are the polynomial basis sets, see Fig. 7 for a visualization. The model's variance s^2 can be estimated by Eqn. 16 using this data.
3) Having the reference model M_x and M_y available, the number of calibration points n can be varied for an uncertainty estimation. Therefore, if the same hardware setup is used in different systems, the uncertainty can be estimated a-priori to the system's initial calibration. Each calibration node must have noise s repeatedly added to it, which is accomplished via Monte Carlo simulation.
4) For each number of calibration nodes, the uncertainty can be estimated by the confidence and prediction intervals. The target measurement uncertainty defined in the requirements specification is the upper limit, see Fig. 8. Note that in this example using wide-angle lenses, the biggest distortion is to be expected at the acquired image's borders. Known cross-validation points are utilized to verify the model.

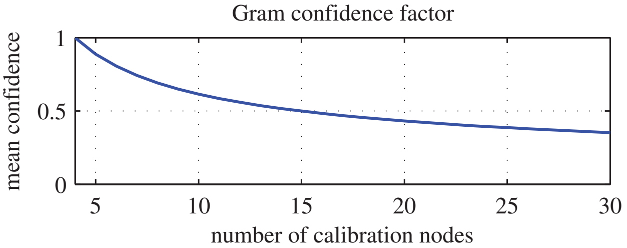
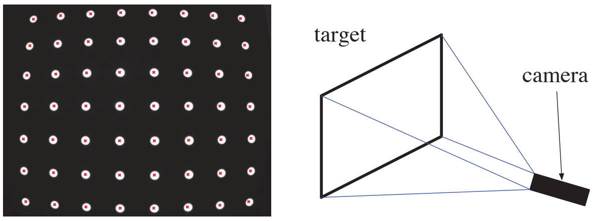


Fig. 5: The plot shows the mean of the matrix operation $\sqrt{\operatorname{diag}(\mathsf{G}\mathsf{G}^{\text{T}})}$, when G is a basis of degree $d = 3$. The value of the Gram confidence factor decreases with increasing number of calibration nodes or d.f. respectively.

VII. Conclusion

Possible fields of application for discrete orthogonal polynomials can be found in the solutions of initial value problems (IVPs), boundary value problems (BVPs), ordinary differential equations (ODEs) or partial differential equations (PDEs). Their numerical stability and computational efficiency is superior to conventional techniques. Furthermore, the high degree polynomials enable various applications such as geometric object modeling, surface description or the optimal fitting of highly non-linear calibration curves. The introduced algebraic framework enables the estimation of uncertainty for the mentioned applications. Especially in cyber-physical systems, multi-modal sensing is of growing importance and the proposed method enables uncertainty based multi-sensor data fusion.



(a) reference target with LEDs (b) camera setup with target

Fig. 6: For this example, the reference measurement is carried out on an orthogonal $25\,[\text{mm}]$ grid in metric real space with 7×8 LEDs, see (a). The LEDs' center coordinates act as calibration nodes for the model building. The distortion caused by the wide-angle lens is clearly visible. In this setup, an M-mount lens with a focal length of $f = 4.2\,[\text{mm}]$ and a $1/3$" monochrome CCD sensor with $1296 \times 966\,[\text{pixel}]$ resolution was deployed to capture the target, see (b).

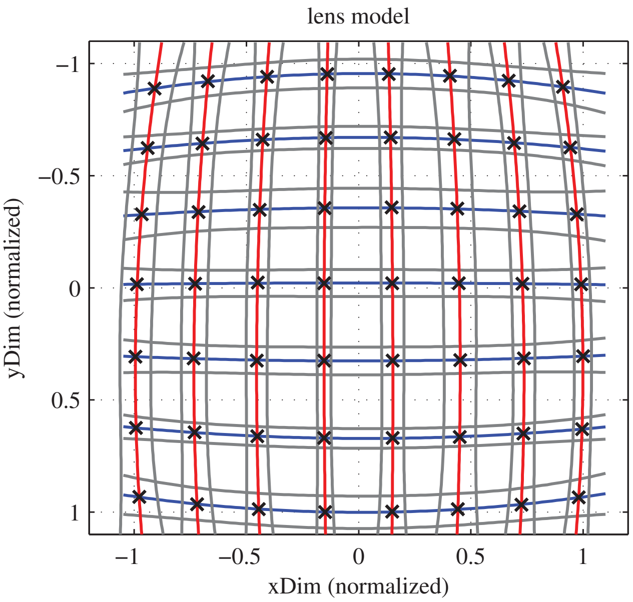


Fig. 7: The reference measurement enables the formulation of a virtual lens model M_x (blue) and M_y (red). Each row and column features a number of calibration nodes n_r and n_c. For better demonstration, only few calibration points have been produced for this visualization. The model's uncertainty (gray) has been scaled by a factor of $k = 40$ to make it visible.

References

[1] JCGM 100:2008, "Evaluation of measurement data guide to the expression of uncertainty in measurement (GUM)," international standard.

[2] E. Geisberger and M. Broy, *agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems*. Springer DE, 2012, vol. 1.

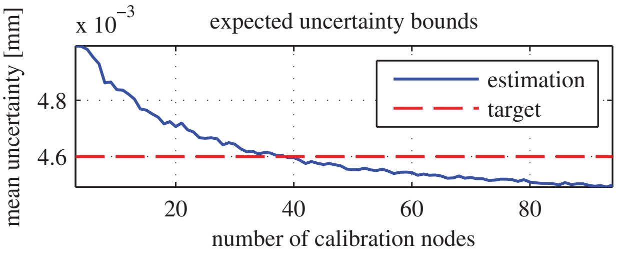


Fig. 8: This plot shows a Monte Carlo simulation of artificial calibration nodes with varying standard deviation s. The uncertainty converges to a lower limit with increasing n, 10^4 iterations have been performed. In this example, to reach a target uncertainty of $4.6 \times 10^{-3}\,[\text{mm}]$, $n \geq 40$ are required. Cross-validation confirms the result.

[3] E. A. Lee, "Cyber physical systems: Design challenges," in *International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC)*, May 2008, invited Paper.

[4] I. Lee, O. Sokolsky, S. Chen, J. Hatcliff, E. Jee, B. Kim, A. King, M. Mullen-Fortino, S. Park, A. Roederer, and K. Venkatasubramanian, "Challenges and research directions in medical cyber physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 75 –90, jan 2012.

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*. JHU Press, 2012, vol. 3.

[7] I. A. Ibragimov, R. Z. Has' minskii, and S. Kotz, *Statistical estimation: asymptotic theory*. Springer-Verlag, 1981, vol. 2.

[8] T. H. Wonnacott and R. J. Wonnacott, *Regression: a second course in statistics*. Wiley and Sons, 1981.

[9] E. Kreyszig, *Statistische Methoden und ihre Anwendungen*. Göttingen: Vandenhoeck und Ruprecht, 1979.

[10] S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*, 3rd ed. Springer Verlag, 1999.

[11] P. OLeary and M. Harker, "An algebraic framework for discrete basis functions in computer vision," in *2008 6th ICVGIP*. Bhubaneswar, India: IEEE, 2008, pp. 150–157.

[12] ——, "Discrete polynomial moments and savitzky-golay smoothing," in *Waset Special Journal*, vol. 72, 2010, pp. 439–443.

[13] ——, "A framework for the evaluation of inclinometer data in the measurement of structures," *IEEE T. Instrumentation and Measurement*, vol. 61, no. 5, pp. 1237–1251, 2012.

[14] R. Neumayr, M. Harker, P. O'Leary, and J. Golser, "New approaches to machine vision based displacement analysis," *IEEE International Instrumentation and Measurement Technology Conference*, pp. 979–984, 2012.

[15] C. Gugg, P. O'Leary, and M. Harker, "Large scale optical position sensitive detector," in *Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International*, 2013, pp. 1775–1780.

[16] C. Gugg, M. Harker, and P. O'Leary, "Structural deformation measurement via efficient tensor polynomial calibrated electro-active glass targets," in *IS&T/SPIE Electronic Imaging, Proceedings Vol. 8661, Image Processing: Machine Vision Applications VI*, 2013.

[17] M. Harker and P. O'Leary. (11 Apr 2013 (Updated 19 Aug 2013)) Discrete orthogonal polynomial toolbox. Matlab File Exchange. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/41250

[18] ——. (18 Apr 2013 (Updated 17 Jun 2013)) Ordinary differential equation toolbox. Matlab File Exchange. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/41354

Part IV

Discussion and Appendices

15 Outlook

The theory of DOP basis functions is completely general. It was shown in Part III, that the concept is eligible to describe various engineering problems and yields solutions of high numerical quality. The method is fully scalable and can be deployed on embedded system as well as on high-performance computing (HPC) systems. This chapter provides insights about continuative and related problems without the claim of completeness. The goal is to create an impression where future developments may take place. This includes abstract mathematical concepts such as the numerical solution of PDEs and Sturm-Liouville problems. Decomposition of large problems enables parallel computing, which is a requirement for utilizing modern HPC architectures as well as embedded FPGAs, e.g., for real-time time series/sequence data mining, which will be a strategic field of research for the Chair of Automation in the future.

15.1 Sturm-Liouville Problems

A Sturm-Liouville problem is a second order ODE. The structure is basically an eigenproblem,

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left[p(x)\,\frac{\mathrm{d}y}{\mathrm{d}x}\right] + q(x)\,y = \lambda\,w(x)\,y \tag{15.1}$$

in the finite interval $x_1 \leq x \leq x_n$, where $p(x)$, $q(x)$ and $w(x)$ are real-valued strictly positive. Additionally, there are two boundary conditions which are most commonly formulated as,

$$a_1\,y(x_1) + a_2\,y'(x_1) = 0, \tag{15.2}$$

$$b_1\,y(x_2) + b_2\,y'(x_2) = 0. \tag{15.3}$$

When implementing Sturm-Liouville problems in a discrete manner, following properties must be considered [89]:

1. All eigenvalues are real and there is no largest eigenvalue, i.e., there are an infinite number of eigenvalues and $\lambda_m \to \infty$ as $m \to \infty$. Given a set of n discrete points $\boldsymbol{x}$, there can be theoretically only n eigenvalues;

2. The m^{th} eigenfunction has m zeros on the interval $a < x < b$. However, given n points, only functions with a maximum of $n/2$ zeros are describable without aliasing;

3. The eigenfunctions are orthogonal w.r.t. the weighting function, $\int_a^b \Phi_i(x)\,w(x)\Phi_j(x) = \delta_{ij}$.

The general Sturm-Liouville problem in Eqn. (15.1) with its corresponding boundary value conditions can be directly discretized as

$$\{\mathsf{DPD} - \mathsf{Q}\}\,\boldsymbol{y} = -\lambda\mathsf{W}\boldsymbol{y} \quad \text{with} \quad \mathsf{C}^{\mathrm{T}}\,\boldsymbol{y} = \boldsymbol{0}, \tag{15.4}$$

whereby $\mathsf{P} = \mathrm{diag}(p(\boldsymbol{x}))$, $\mathsf{Q} = \mathrm{diag}(q(\boldsymbol{x}))$ and $\mathsf{W} = \mathrm{diag}(w(\boldsymbol{x}))$. Unfortunately, a direct solution of this equation is numerically unstable due to aliasing.

A weighted and constrained basis function $\bar{\mathsf{G}}$ is introduced. It fulfills the orthogonality condition $\bar{\mathsf{G}}^{\mathrm{T}}\mathsf{W}\bar{\mathsf{G}} = \mathsf{I}$ and the boundary condition $\mathsf{C}^{\mathrm{T}}\bar{\mathsf{G}} = \mathsf{0}$. These basis functions are admissible functions to the Sturm-Liouville problem. The number of zeros in the basis functions increase from left to right in the matrix. The number of zeros in the basis function is limited; in order to avoid aliasing, the basis function is truncated to the $k = n/2$ basis functions, i.e., $\hat{\mathsf{G}} = \bar{\mathsf{G}}(:, 1:k)$. The eigenfunctions are now found as linear combinations of these admissible functions, i.e., $\boldsymbol{y} = \hat{\mathsf{G}}\,\boldsymbol{\alpha}$. Substitution into Eqn. (15.4) yields

$$\{\mathsf{DPD} - \mathsf{Q}\}\;\hat{\mathsf{G}}\,\boldsymbol{\alpha} = -\lambda \mathsf{W}\hat{\mathsf{G}}\,\boldsymbol{\alpha}. \tag{15.5}$$

Premultiplication with $\hat{\mathsf{G}}^{\mathrm{T}}$ gives

$$\hat{\mathsf{G}}^{\mathrm{T}}\;\{\mathsf{DPD} - \mathsf{Q}\}\;\hat{\mathsf{G}}\,\boldsymbol{\alpha} = -\lambda\boldsymbol{\alpha}, \tag{15.6}$$

since $\hat{\mathsf{G}}^{\mathrm{T}}\mathsf{W}\hat{\mathsf{G}} = \mathsf{I}$. Defining $\mathsf{L} = \hat{\mathsf{G}}^{\mathrm{T}}\;\{\mathsf{DPD} + \mathsf{Q}\}\;\hat{\mathsf{G}}$ yields a standard eigenvector problem,

$$\{\mathsf{L} + \lambda\,\mathsf{I}\}\;\boldsymbol{\alpha} = 0. \tag{15.7}$$

Solving this equation for the eigenvalues λ_i and the associated eigenvectors $\boldsymbol{\alpha}_i$. Substitution of an eigenvector $\boldsymbol{\alpha}_i$ into the relation

$$\boldsymbol{y}_i = \hat{\mathsf{G}}\,\boldsymbol{\alpha}_i \tag{15.8}$$

yields the i^{th} eigenfunction $\boldsymbol{y}_i$ of the system.

15.2 Partial Differential Equations

It was shown in Chapter 10 how to reconstruct the curve $\hat{\boldsymbol{y}}$ from a set of m perturbed inclination measurements $\boldsymbol{z} \in \mathbb{R}^{(m\times 1)}$,

$$\hat{\boldsymbol{y}} = \boldsymbol{y}_{\mathrm{h}} + \mathsf{P}\,\boldsymbol{z}. \tag{15.9}$$

Concatenation of k individual measurement vectors $\boldsymbol{z}_i \in \mathbb{R}^{(m\times 1)}$ yields the measurement matrix $\mathsf{Z} = [\boldsymbol{z}_1 \ldots \boldsymbol{z}_k]$. The vertical dimension corresponds to the spatial domain and the horizontal dimension corresponds to the temporal domain, i.e., $\mathsf{Z} \in \mathbb{R}^{(m\times k)}$. This formalism enables simultaneous processing of measurement data in both dimensions; hence, this corresponds to an inverse PDE when reconstructing the plane in a least squares sense. The most general formulation for the PDE describing the function values over time Y in terms of Z is

$$\mathsf{Z} = \mathsf{U}_{\mathrm{s}}\,\mathsf{Y}\,\mathsf{U}_{\mathrm{t}}^{\mathrm{T}} + \mathsf{V}_{\mathrm{s}}\,\mathsf{Y} + \mathsf{Y}\,\mathsf{V}_{\mathrm{t}}^{\mathrm{T}} + \mathsf{E}. \tag{15.10}$$

The subscripts indicate the spatial (s) and time (t) domain respectively for the basis functions U and V. There are efficient methods available for the solution of such problems, e.g., surface reconstruction from gradient fields [90]. The approach is especially useful to analyze the dynamic behavior of objects measured by chains of inclinometers, e.g., structural health monitoring of regions subject to ground subsidence over time.

15.3 Model Based Production

As argued in Section 2.2.1, the concept of cyber-physical production systems (CPPS) will be essential for future complex industrial process control. The manufacturing of products is a physical process and, therefore, adequate process models are required. Ultimately, this enables the implementation of a reactive/operational control loop directly on the production machine as well as a strategic/predictive control loop as part of a supervisory system, which consolidates the production data of multiple machines. The idea is summarized with the term *model based production*, it is based on three fundamental pillars:

1. **Cyber-physical systems** provide the necessary system architecture as well as the infrastructure. This includes complexity management, interface definitions between the system's components as well as communication protocols for the agents within the network;
2. **Model based design** enables the formulation of physical processes on abstract model level. Continuous physical models are discretized and deployed on embedded systems without the need of detailed knowledge about the target platform;
3. **Temporal data mining** relies on the concept of cognition. The mathematical models are based on physical phenomena, i.e., cause and effect; this causality is the main difference to classical data mining approaches which are merely based on correlation.

It is important to note that the concept is not limited to the manufacturing of classical piece goods. The Chair of Automation's goal is to execute the concept on a variety of machines, e.g., on heavy machinery from the mining and tunneling industry, injection molding machines for the processing of polymers or on offshore drilling platforms from the petroleum industry.

15.4 Temporal Data Mining

The focus of this thesis was laid onto CPS from a bottom-up perspective, i.e., from sensor level (physical domain) to supervisory level (socio domain). The supervisory system's purpose is to acquire data from distributed sensor motes mounted on the facility to be monitored. The Chair of Automation's vision is to establish a new field of research regarding the real-time analytics of machine data emanating from a WSAN. From this point of view, this is a top-down approach. Systems for condition monitoring and machine diagnostics rely on *data mining* (DM) techniques, which is by definition [25, 64, 86] a methodology for knowledge discovery in large amounts of data. CRISP-DM provides a formal, yet generic framework for industrial DM projects [25], the phases and subtasks are visualized in Fig. 15.4. In literature as well as applications, DM is mostly related to *knowledge discovery in data bases* (KDD); the static data is free from temporal references. The data sets are analyzed statistically in order to find correlations [86, 115, 132]; however, this approach is not suitable for time-critical tasks based on physical processes.

In this vein, analysis of dynamic data, i.e., logically ordered sequences and/or chronologically ordered time series, is a DM problem of different nature [4, 48, 77, 118, 191], so called *temporal data mining*. The data stream emerges from a mechanical installation equipped with numerous sensors, most likely a mechatronic system, see Fig. 15.1 for an example. Hence, the underlying physical phenomena must be considered to derive suitable mathematical models with significance [82, 169]. In contrast to classical DM, these models enable the discovery of causality between data sets. Ultimately, predictive control loops can be implemented using this concept for preventive maintenance, whereby the goal is to find anomalies within the data stream [57]. An anomaly is an unexpected behavior, i.e., a discrepancy in comparison to frequent patterns. This requires the automatic recognition of the machine's operating states and reliable predictions using embedded simulation. Anomalies can occur on different time scales [28], which facilitates the need of hierarchical scale-space decomposition. The phenomenology approach is based on five cognition steps: awareness, feeling, perception, reasoning and judgment. The system architecture must implement these process steps for knowledge discovery in streaming data.

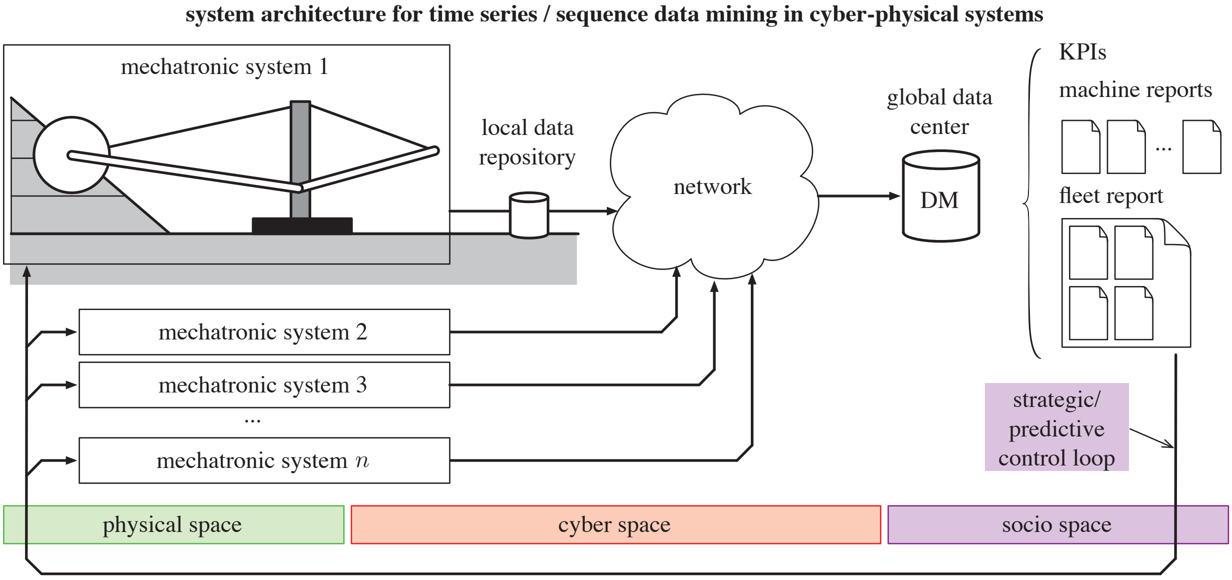


Fig. 15.1 A conventional reactive control loop controls the mechatronic system based on the machine's sensors. The sensor data as well as the control signals are transmitted to a local data repository. The global data center consolidates the data from numerous mechatronic systems via network. Key performance indicators (KPIs) as well as machine and fleet reports are generated based on the mined data for knowledge representation. Using this information, decisions are made which proactively influences the machine's operation. The illustration features a stylized bucket wheel reclaimer as an example.

The proposed solution is based on *lexical analysis*, i.e., the concept of natural language known from compiler theory. Symbolic approximation is used to embed semantic information about the system into the predictive model. The majority of the machine's operating states is extracted from the (cleansed) streaming data during the initial exploratory phase and stored in a lexical definition. However, this process requires thorough a-priori knowledge about the machine and its associated processes, which is formally described via a machine definition.

Each sensor requires at least one dedicated *single channel lexical analyzer* (SCLA), see Fig. 15.2. Multiple linear operations can be applied to data emerging from a single sensor, whereby each operation is implemented by an SCLA. Combining individual SCLAs yields a *multi channel lexical analyzer* (MCLA), which can be implemented as a systolic array for parallel processing, see Fig. 15.3. The initial phase is supervised in order to establish a dictionary with human-readable text.

The run-time process is composed of following steps:

1. **Linear differential operation.** The temporal data emerges from a physical system; thus, it is victim to perturbations. A regularizing linear differential operator (LDO) is required to process the raw data from the continuous data stream. Suitable LDOs can be synthesized using the theory in Chapter 10. In order to save network bandwidth, sampled values are commonly transmitted on value-change. This generates a non-equally spaced time basis;

2. **Symbolization.** Clustering groups of similar signals enables their classification utilizing lexical definitions; therefore, the machine's operating states can be identified. It is supposed, that a semi-positive definite distance measure exists between these clusters. Clustering can be achieved via statistical methods such as histogram analysis, principal component analysis and/or well-known machine learning methods such as k-means clustering, self-organizing maps, Bayesian networks or support vector machines [77]. Abstraction of these operation classes leads to their symbolic representation;

3. **Lexical compression.** Lexical compression yields *tokens* with predicates and introduces the requirement of *dynamic time warping* [180] for signal alignment as well as *symbolic regularization*. The predicates store information about the length and temporal position (index) of the symbol. The process allows complete backtracking, i.e., full reconstruction is supported despite the compression. This directly corresponds to the concept of LEX[1] and *symbolic aggregate approximation* (SAX) [114, 124];

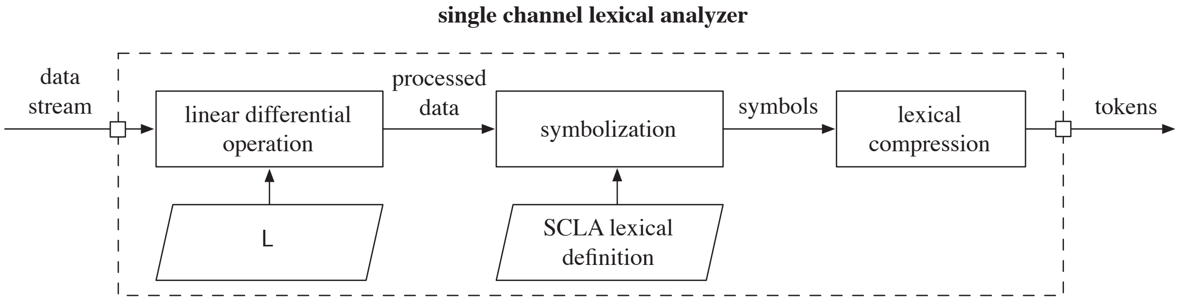


Fig. 15.2 Process model of the SCLA.

4. **Token merge.** The MCLA implements a setup of multiple, parallel SCLAs. Each SCLA generates tokens from streaming sensor data. Tokens of the same class are merged for each time slot, effectively forming chronologically ordered *merged tokens*, whereby their meaning is stored in the MCLA lexical definition;

5. **Textual mapping.** Each combination of tokens has a specific meaning within this context, i.e., human-readable text from the dictionary is mapped to a permutation of all sets of tokens. The process is implemented as a first-in-first-out (FIFO) buffer of limited length;

6. **Statistical evaluation.** All possible combinations of tokens are statistically evaluated as *events*. Multiple individual events may refer to a single operating state. Furthermore,

[1] The LEX and YACC Page, `dinosaur.compilertools.net`

statistically irrelevant events are referred to as *don't cares* while unknown, yet significant events which have net been identified during the exploratory phase, are referred to as *don't knows*. This methodology facilitates the generation of reports with high information density.

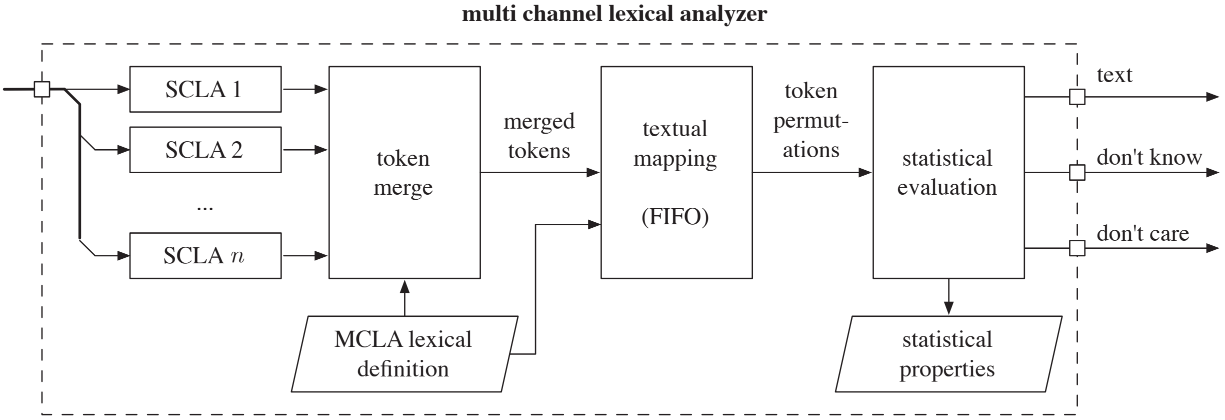


Fig. 15.3 Process model of the MCLA.

The SCLA architecture is suitable to implement multiple SCLAs in a parallel manner as a systolic array within an MCLA. Depending on the application's intended use, two target platforms for hardware acceleration are available:

1. An **FPGA** provides a low-level implementation on an embedded system using multiple DSP slices. The logic blocks are described using HDL; ideally, MBD supports the programming task by avoiding low-level coding. FPGAs enable highly efficient hardware implementations w.r.t. energy consumption and computational performance. SoC platforms such as Xilinx Zynq[2] allow to merge FPGA with MCU designs, which is required to perform real-time data analytics on the embedded system. This configuration is suitable to be integrated directly on the mechatronic system, e.g., as an extensions to an existing PLC system;

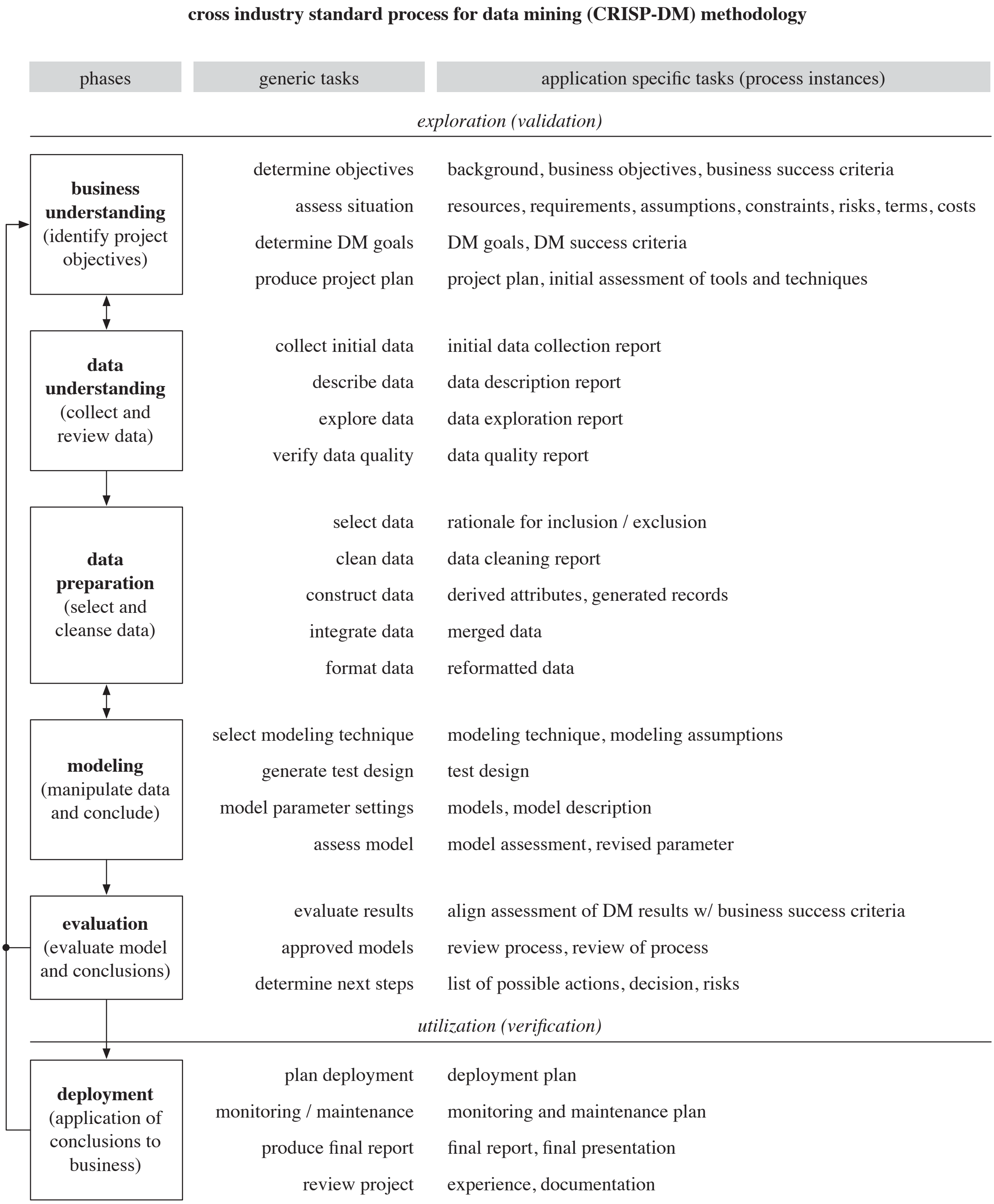
2. An **HPC** system is a high-level implementation using a PC platform with multi-core processors, hyper-threading technology and general purpose GPUs (GPGPUs). Parallel programs are written in high-level abstract languages such as MATLAB's Parallel Computing Toolbox [139] with integrated specialized application programming interfaces (APIs) such as nVidia's CUDA[3] or Khrono's OpenCL[4]. The separated CPU and GPU memory, the communication overhead between individual CPUs and GPUs and possible interlocks between co-processors must be considered during the implementation [47]. This configuration is meant to be integrated as part of the supervisory system.

The real-time analytics of machine data is of growing importance for new business models such as *smart services*. Constructors as well as operators are interested in the behavior of their equipment over the complete life cycle. Data mining of streaming data is an enabler technology to identify technical as well as economic key performance indicators (KPIs).

[2] Xilinx Zynq-7000 SoC, `www.xilinx.com/products/silicon-devices/soc/zynq-7000.html`

[3] Compute Unified Device Architecture, `developer.nvidia.com/cuda-zone`

[4] Open Computing Language, `www.khronos.org/opencl/`



cross industry standard process for data mining (CRISP-DM) methodology

phases	generic tasks	application specific tasks (process instances)
		exploration (validation)
business understanding (identify project objectives)	determine objectives	background, business objectives, business success criteria
	assess situation	resources, requirements, assumptions, constraints, risks, terms, costs
	determine DM goals	DM goals, DM success criteria
	produce project plan	project plan, initial assessment of tools and techniques
data understanding (collect and review data)	collect initial data	initial data collection report
	describe data	data description report
	explore data	data exploration report
	verify data quality	data quality report
data preparation (select and cleanse data)	select data	rationale for inclusion / exclusion
	clean data	data cleaning report
	construct data	derived attributes, generated records
	integrate data	merged data
	format data	reformatted data
modeling (manipulate data and conclude)	select modeling technique	modeling technique, modeling assumptions
	generate test design	test design
	model parameter settings	models, model description
	assess model	model assessment, revised parameter
evaluation (evaluate model and conclusions)	evaluate results	align assessment of DM results w/ business success criteria
	approved models	review process, review of process
	determine next steps	list of possible actions, decision, risks
		utilization (verification)
deployment (application of conclusions to business)	plan deployment	deployment plan
	monitoring / maintenance	monitoring and maintenance plan
	produce final report	final report, final presentation
	review project	experience, documentation

Fig. 15.4 CRISP-DM reference model [25].

16 | Conclusion

Cyber-physical systems (CPS) describe the abstract idea of independent, yet connected and collaborating devices. This concept of communicating mechatronic systems will definitely play a key role in future engineering applications across all branches of industry, no matter if the term *CPS* will remain as such. Wireless sensor actuator networks (WSAN) are the source of the machine data; although it is a promising technology, there are still no standardized protocols for network communication available. The CPS architecture relies on the existence of a supervisory system to consolidate the data of all motes; the growing computational performance of embedded systems may facilitates the implementation of swarm intelligence, which effectively abolishes the need of supervision and relies purely on self-organization of the agents. Ubiquitous computing and condition monitoring enables the complete life cycle analysis of assets together with total cost of ownership, i.e., total process transparency. However, establishing compatibility to legacy systems will remain a challenge because of missing standard interfaces.

The importance of physical phenomenology and adequate mathematical models was demonstrated throughout in this thesis. In order to understand the processes, it is required to fuse interdisciplinary knowledge. Structured design methodologies facilitate functional decomposition. Assisting technologies and new programming paradigms such as the presented model based design (MBD) approach help engineers to concentrate on the problem to be solved rather than the actual implementation; consequently, this also simplifies the way engineers from different fields are working together. Nevertheless, there is still unused potential, e.g., implementing a fully automatized embedded targeting procedure is a highly complex and time-consuming task. MBD eliminates the need of manually coding functions in low-level programming languages, but necessary organizational issues such as code management remain unchanged.

Model based production relies on machine to machine communication, whereby computational systems are directly embedded into the manufacturing process for data acquisition. At a given point, it will not make any sense to just increase the performance of the computing system to achieve improvements. Efficient mathematical models with high numerical quality and stability enable the forward and inverse solution of ODE and PDE, which effectively enables adaptive control. Inverse problems require regularization to acquire a maximum likelihood estimation; incorporation of contextual knowledge is the key for the formulation of regularizing linear operators. Inductive reasoning is the task of deriving information about a machine type from numerous exemplary machines in the field, which incorporate similar, yet slightly different behavior w.r.t. operation characteristics.

The correct functionality of the devices was demonstrated on laboratory prototypes together with in-the-loop verification of the algebraic framework on model, software and processor level. The deployment of a WSAN in a harsh environment and its combination with a supervisory system for temporal data mining is of special interest for applications in the heavy industry.

Appendices

A | Terms in Metrology

The following list represents a selection of common terms in metrology, the definitions are cited from ISO/IEC Guide 99-2007: International vocabulary of metrology (VIM) - Basic and general concepts and associated terms [20, 106] and the German terms were translated via DIN 1319 Grundbegriffe der Messtechnik [37–40]. The used notation is: **english term** [German translation]: description.

1. **metrology** [Metrologie]: science of measurement and its application;
2. **measurement** [Messung]: process of experimentally obtaining one or more quantity values that can reasonably be attributed to a quantity;
3. **measurand** [Messgröße]: quantity intended to be measured;
4. **measurement unit** [Maßeinheit]: real scalar quantity, defined and adopted by convention, with which any other quantity of the same kind can be compared to express the ratio of the two quantities as a number;
5. **measurement principle** [Messprinzip]: principle of measurement, physical phenomenon serving as a basis of a measurement;
6. **measurement method** [Messmethode]: generic description of a logical organization of operations used in a measurement;
7. **measurement procedure** [Messverfahren]: detailed description of a measurement according to one or more measurement principles and to a given measurement method, based on a measurement model and including any calculation to obtain a measurement result;
8. **measurement result** [Messergebnis]: set of quantity values being attributed to a measurand together with any other available relevant information. The complete measurement result is composed of the measured quantity value and the associated uncertainty;
9. **measured quantity value** [Messwert]: quantity value representing a measurement result;
10. **true quantity value** [wahrer Wert einer Größe]: true unknown value of a quantity value;
11. **reference quantitiy value** [Referenzwert]: quantity value used as a basis for comparison with values of quantities of the same kind;
12. **measurement accuracy** [Messgenauigkeit]: closeness of agreement between a measured quantity value and a true quantity value of a measurand;
13. **measurement trueness** [Messrichtigkeit]: closeness of agreement between the average of an infinite number of replicate measured quantity values and a reference quantity value;

14. **measurement precision** [Messpräzision]: closeness of agreement between indications or measured quantity values obtained by replicate measurements on the same or similar objects under specified conditions;

15. **measurement error** [Messabweichung]: measured quantity value minus a reference quantity value;

16. **systematic measurement error** [systematische Messabweichung]: component of measurement error that in replicate measurements remains constant or varies in a predictable manner;

17. **measurement bias** [Bias der Messung]: estimate of a systematic measurement error;

18. **correction** [Korrektion]: compensation for an estimated systematic effect;

19. **random measurement error** [zufällige Messabweichung]: component of measurement error that in replicate measurements varies in an unpredictable manner;

20. **measurement repeatability** [Wiederholpräzision]: measurement precision under a set of repeatability conditions of measurement;

21. **repeatability condition of measurement** [Wiederholbedingung von Messungen]: condition of measurement, out of a set of conditions that includes the same measurement procedure, same operators, same measuring system, same operating conditions and same location, and replicate measurements on the same or similar objects over a short period of time;

22. **measurement reproducibility** [Reproduzierbarkeit]: measurement precision under reproducibility conditions of measurement;

23. **reproducibility condition of measurement** [Messbedingungen für Reproduzierbarkeit]: condition of measurement, out of a set of conditions that includes different locations, operators, measuring systems, and replicate measurements on the same or similar objects;

24. **measurement uncertainty** [Messunsicherheit]: non-negative parameter characterizing the dispersion of the quantity values being attributed to a measurand, based on the information used;

25. **standard measurement uncertainty** [Standardmessunsicherheit]: measurement uncertainty expressed as a standard deviation;

26. **target measurement uncertainty** [Höchstwert der Messunsicherheit]: measurement uncertainty specified as an upper limit and decided on the basis of the intended use of measurement results, e.g., as part of the system requirements;

27. **expanded measurement uncertainty** [erweiterte Messunsicherheit]: product of a combined standard measurement uncertainty and a factor larger than the number one. Commonly, an inverse of a suitable cumulative distribution function is used;

28. **calibration** [Kalibrierung]: operation that, under specified conditions, in a first step, establishes a relation between the quantity values with measurement uncertainties provided by measurement standards and corresponding indications with associated measurement uncertainties and, in a second step, uses this information to establish a relation for obtaining a measurement result from an indication;

29. **metrological traceability** [metrologische Rückführbarkeit]: property of a measurement result whereby the result can be related to a reference through a documented unbroken chain of calibrations, each contributing to the measurement uncertainty;
30. **verification** [Verifizierung]: provision of objective evidence that a given item fulfills specified requirements;
31. **validation** [Validierung]: verification, where the specified requirements are adequate for an intended use;
32. **measurement model** [Modell der Messung]: mathematical relation among all quantities known to be involved in a measurement;
33. **measurement function** [Messfunktion]: function of quantities, the value of which, when calculated using known quantity values for the input quantities in a measurement model, is a measured quantity value of the output quantity in the measurement model;
34. **input quantity in a measurement model** [Eingangsgröße des Modells der Messung]: quantity that must be measured, or a quantity, the value of which can be otherwise obtained, in order to calculate a measured quantity value of a measurand;
35. **output quantity in a measurement model** [Ausgangsgröße des Modells der Messung]: quantity, the measured value of which is calculated using the values of input quantities in a measurement model;
36. **influence quantity** [Einflussgröße]: quantity that, in a direct measurement, does not affect the quantity that is actually measured, but affects the relation between the indication and the measurement result;
37. **measuring instrument** [Messgerät]: device used for making measurements, alone or in conjunction with one or more supplementary devices;
38. **measuring system** [Messsystem]: set of one or more measuring instruments and often other devices, including any reagent and supply, assembled and adapted to give information used to generate measured quantity values within specified intervals for quantities of specified kinds. A measuring system may consist of only one measuring instrument;
39. **measuring transducer** [Messumformer]: device, used in measurement, that provides an output quantity having a specified relation to the input quantity;
40. **sensor** [Messaufnehmer/Sensor]: element of a measuring system that is directly affected by a phenomenon, body, or substance carrying a quantity to be measured;
41. **detector** [Detektor]: device or substance that indicates the presence of a phenomenon, body, or substance when a threshold value of an associated quantity is exceeded;
42. **measuring chain** [Messkette]: series of elements of a measuring system constituting a single path of the signal from a sensor to an output element;
43. **adjustment of a measuring system** [Justierung eines Messsystems]: set of operations carried out on a measuring system so that it provides prescribed indications corresponding to given values of a quantity to be measured;
44. **measuring interval/range** [Messbereich]: set of values of quantities of the same kind that can be measured by a given measuring instrument or measuring system with specified instrumental measurement uncertainty, under defined conditions;

45. **sensitivity of a measuring system** [Empfindlichkeit]: quotient of the change in an indication of a measuring system and the corresponding change in a value of a quantity being measured;
46. **selectivity of a measuring system** [Selektivität]: property of a measuring system, used with a specified measurement procedure, whereby it provides measured quantity values for one or more measurands such that the values of each measurand are independent of other measurands or other quantities in the phenomenon, body, or substance being investigated;
47. **resolution** [Auflösung]: smallest change in a quantity being measured that causes a perceptible change in the corresponding indication;
48. **discrimination threshold** [Ansprechschwelle]: largest change in a value of a quantity being measured that causes no detectable change in the corresponding indication;
49. **stability of a measuring** [Messbeständigkeit]: property of a measuring instrument, whereby its metrological properties remain constant in time;
50. **instrumental drift** [Messgerätedrift]: continuous or incremental change over time in indication, due to changes in metrological properties of a measuring instrument;
51. **calibration diagram** [Kalibrierdiagramm]: graphical expression of the relation between indication and corresponding measurement result;
52. **calibration curve** [Kalibrierkurve]: expression of the relation between indication and corresponding measured quantity value.

B | List of Figures

C List of Publications

Conference Papers

1. C. Gugg, M. Harker, and P. O'Leary. Structural deformation measurement via efficient tensor polynomial calibrated electro-active glass targets. In *IS&T/SPIE Electronic Imaging, Image Processing: Machine Vision Applications VI*, volume 8661, pages 86610F–86610F–9, 2013
2. C. Gugg, P. O'Leary, and M. Harker. Large scale optical position sensitive detector. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1775–1780, May 2013
3. M. Harker, C. Gugg, and P. O'Leary. Model Based Design for the Real-Time Solution of Inverse Problems in Cyber-Physical Systems. In *Proceedings of the Fifth International Conference on Cyber-Physical Systems (ICCPS)*, page 217, 2014
4. C. Gugg, M. Harker, and P. O'Leary. Virtual Calibration Environment for A-Priori Estimation of Measurement Uncertainty. In *IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 52–57, 2014
5. P. O'Leary, C. Gugg, M. Harker, and G. Rath. Mathematical Model and Software Architecture for the Solution of Inverse Problems Involving Sensor Arrays. In *IEEE Sensors Proceedings*, pages 586–589, 2014
6. P. O'Leary, M. Harker, and C. Gugg. A Position Paper on: Sensor-Data Analytics in Cyber Physical Systems, from Husserl to Data Mining. In *INSTICC Sensornets Proceedings*, 2015
7. C. Gugg and P. O'Leary. Robust machine vision based displacement analysis for tunnel boring machines. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Pisa, Italy, May 2015
8. M. Pucher, P. O'Leary, C. Gugg, and C. Hoefer-Oellinger. An autonomous water sampling and monitoring device for deployment in harsh underground environment. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Pisa, Italy, May 2015
9. P. O'Leary, P. Lee, C. Gugg, and M. Harker. Tutorial: Mining sensor and measurement data in real-time. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Pisa, Italy, May 2015

Journal Articles

1. C. Gugg, P. O'Leary, and M. Harker. Analytic Multi-Source Data Fusion and its Application to a Large-Scale Optical PSD. *Instrumentation and Measurement, IEEE Transactions on*, 63(5):1116–1126, May 2014

2. M. Gram and C. Gugg. Einsatzsmöglichkeiten von Cyber-Physical Systems im Lebenszyklusmanagement von Anlagen. *Industrie Management*, 1:39–43, Feb. 2013

3. C. Gugg, M. Harker, P. O'Leary, and G. Rath. An algebraic framework for the real-time solution of inverse problems on embedded systems. *arXiv Open Access Journal Article, arXiv:1406.0380*, 2014

Theses

1. C. Gugg. Recognition of 3D Embossed Digits through Spectral Analysis and Machine Learning. Bachelor's thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2010

2. C. Gugg. Conceptual Development of an Opto-electronic Perpendicular Utilizing Electroactive Glass. Master's thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2011

D Bibliography

[1] M. Ahmadian, Z. J. Nazari, N. Nakhaee, and Z. Kostic. Model based design and SDR. In *The 2^{nd} IEEE/EURASIP Conference on DSP enabled radio*, page 8, Sept. 2005.

[2] G. B. Airy. On the diffraction of an object-glass with circular aperture. *Transactions of the Cambridge Philosophical Society*, 5:283, 1835.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications magazine, IEEE*, 40(8):102–114, 2002.

[4] C. M. Antunes and A. Oliveira. Temporal data mining: An overview. In *KDD Workshop on Temporal Data Mining*, pages 1–13, 2001.

[5] T. Arens, F. Hettlich, C. Karpfinger, U. Kockelkorn, K. Lichtenegger, and H. Stachel. *Mathematik*. Spektrum, Akad. Verlag, 2008.

[6] ARM. Compiler 6. Tech. spec., `ds.arm.com/ds-5/build/arm-compiler-6/`, 2014.

[7] H. Arno, W. Dieter, G. Rene, and J. Frederik. From IT4Metals to Industry 4.0. In L. Webb and R. Oxenford, editors, *Siemens Metals Magazine*. Siemens VAI, 2014.

[8] H. Atzrodt and A. Alizadeh. Simulating a Piezoelectric-Actuated Hydraulic Pump Design at Fraunhofer LBF and Ricardo. In L. Webb and R. Oxenford, editors, *MathWorks News&Notes*. The MathWorks, Inc., 2012.

[9] A. Badshah. *Polynomial Basis Functions and their Application to Hierarchical Image Processing*. PhD thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2012.

[10] R. Baheti and H. Gill. Cyber-physical systems. *The Impact of Control Technology*, pages 161–166, 2011.

[11] Basler. GigE Vision - CPU Load and Latency. White paper, `www.baslerweb.com`, Dec. 2007.

[12] Basler. Advantages of Power over Ethernet. White paper, `www.baslerweb.com`, Nov. 2011.

[13] Basler. Optics Recommendation. White paper, `www.baslerweb.com`, Sept. 2011.

[14] R. Beard, T. McLain, M. Goodrich, and E. Anderson. Coordinated target assignment and intercept for unmanned air vehicles. *Robotics and Automation, IEEE Transactions on*, 18(6):911–922, Dec. 2002.

[15] C. Boley. *Handbuch Geotechnik*. Springer, 2012.

[16] H. Bossel. *Modeling and simulation*. Informatica International, Inc., 1994.

[17] S. Brandt. *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*. Springer Verlag, 3rd edition, 1999.

[18] S. Brandt. *Datenanalyse für Naturwissenschaftler und Ingenieure*. Springer Verlag, 5th edition, 2013.

[19] H. Briehl. *Chemie der Werkstoffe*. Springer, 1995.

[20] B. Brinkmann. *Internationales Wörterbuch der Metrologie: Grundlegende und allgemeine Begriffe und zugeordnete Benennungen (VIM) Deutsch-englische Fassung ISO/IEC-Leitfaden 99: 2007*. Beuth Verlag, 2012.

[21] C. Buck. Selbst ist das Bauteil. In C. Heise, A. Heise, and M. Pietschmann, editors, *Technology Review*. Heise Zeitschriften Verlag, 2013.

[22] J. V. Candy. *Signal processing: model based approach*. McGraw-Hill, Inc., 1986.

[23] J. V. Candy. *Model-Based Signal Processing*, volume 36. John Wiley & Sons, 2005.

[24] B. Cazzolato. University of Adelaide Undergraduates Design, Build, and Control an Electric Diwheel Using Model-Based Design. In L. Webb and R. Oxenford, editors, *MathWorks News&Notes*. The MathWorks, Inc., 2012.

[25] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. CRISP-DM 1.0 Step-by-step data mining guide. Manual, 2000.

[26] F. E. Chase, C. Mark, and K. A. Heasley. Deep cover pillar extraction in the US coalfields. In *21st International conference on ground control in mining, Morgantown*, pages 6–8, 2002.

[27] R. Chen, M. Sgroi, L. Lavagno, G. Martin, A. Sangiovanni-Vincentelli, and J. Rabaey. UML and platform-based design. In *UML for Real*, pages 107–126. Springer, 2003.

[28] B. Chiu, E. Keogh, and S. Lonardi. Probabilistic discovery of time series motifs. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 493–498. ACM, 2003.

[29] Y. Chugh and H. Parkinson. Automation trends in room-and-pillar continuous-mining systems in the USA. *Journal of the South African Institute of Mining and Metallurgy*, 81(3):57–65, 1981.

[30] J. C. Clarke. Modelling uncertainty: A primer. Technical report, University of Oxford. Dept. Engineering science, Tech. Rep, 1998.

[31] M. Conrad and I. Fey. ISO 26262–Exemplary Tool Classification of Model-Based Design Tools. *Softwaretechnik-Trends*, 31(3), 2011.

[32] M. Conrad and G. Sandmann. A verification and validation workflow for IEC 61508 applications. Technical report, SAE Technical Paper, 2009.

[33] M. Conrad, G. Sandmann, and P. Munier. Software tool qualification according to ISO 26262. Technical report, SAE Technical Paper, 2011.

[34] A. Davare, D. Densmore, L. Guo, R. Passerone, A. L. Sangiovanni-Vincentelli, A. Simalatsar, and Q. Zhu. metro II: A design environment for cyber-physical systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 12(1s):49, 2013.

[35] A. Davare, D. Densmore, T. Meyerowitz, A. Pinto, A. Sangiovanni-Vincentelli, G. Yang, H. Zeng, and Q. Zhu. A next-generation design framework for platform-based design. In *Conference on using hardware design and verification languages (DVCon)*, volume 152, 2007.

[36] T. DeMarco. *Structured Analysis and System Specification.* Yourdon Press, 1979.

[37] DIN 1319-1:1985-06. Grundbegriffe der Messtechnik: Allgemeine Grundbegriffe. Deutsche Norm.

[38] DIN 1319-2:1980-01. Grundbegriffe der Messtechnik: Begriffe für die Anwendung von Messgeräten. Deutsche Norm.

[39] DIN 1319-3:1996-05. Grundlagen der Messtechnik: Auswertungen von Messungen einer einzelnen Messgröße, Messunsicherheit. Deutsche Norm.

[40] DIN 1319-4:1985-12. Grundbegriffe der Messtechnik: Behandlung von Unsicherheiten bei der Auswertung von Messungen. Deutsche Norm.

[41] H. Dubois, M. Peraldi-Frati, and F. Lakhal. A Model for Requirements Traceability in a Heterogeneous Model-Based Design Process: Application to Automotive Embedded Systems. In *Engineering of Complex Computer Systems (ICECCS), 15th IEEE International Conference on*, pages 233–242, Mar. 2010.

[42] Durham Geo Slope Indicator. *Guide to geotechnical instrumentation*, 2004.

[43] W. Elmenreich. An introduction to sensor fusion. *Vienna University Of Technology*, 2002.

[44] Energetics Incorporated. Cyber-Physical Systems: Situation Analysis of Current Trends, Technologies, and Challenges. Technical report, National Institute of Standards and Technology (NIST), Mar. 2012.

[45] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375. Springer, 1996.

[46] ENV-13005:1999-07. Leitfaden zur Angabe der Unsicherheit beim Messen. Europäische Vornorm.

[47] R. Ernst, J. Henkel, and T. Benner. Hardware-software cosynthesis for microcontrollers. *Readings in hardware/software co-design*, pages 18–29, 2002.

[48] P. Esling and C. Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12, 2012.

[49] EU Council 2004/22/EC. Measuring instruments. Directive.

[50] EU Council 2006/25/EC. Minimum health and safety requirements regarding the exposure of workers to risks arising from physical agents (artificial optical radiation). Directive.

[51] EU Council 2006/42/EC. Machinery. Directive.

[52] P. Fastermann. *3D-Druck/Rapid Prototyping.* Springer, 2012.

[53] E. Fauster. *Statistical Uncertainty Analysis for Image Processing Algorithms in Metric Vision Systems.* PhD thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2008.

[54] D. Fennibay, A. Yurdakul, and A. Sen. Introducing hardware-in-loop concept to the hardware/software co-design of real-time embedded systems. In *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pages 1902–1909. IEEE, 2010.

[55] R. S. Figliola and D. E. Beasley. *Theory and Design for Mechanical Measurements.* Wiley and Sons, 1995.

[56] R. Fintel. Vergleich der gängigsten digitalen Schnittstellentechnologien in der industriellen Bildverarbeitung. White paper, `www.baslerweb.com`, Nov. 2013.

[57] Fraunhofer IOSB. Industry 4.0: Information technology is the key element in the factory of the future. Press information, Jan. 2013.

[58] W. Gautschi. *Orthogonal polynomials: computation and approximation.* Oxford University Press, 2004.

[59] E. Geisberger and M. Broy. *agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems*, volume 1. Springer, 2012.

[60] Gesellschaft Mess- und Automatisierungstechnik. Automation 2020: Bedeutung und Entwicklung der Automation bis zum Jahr 2020. Technical report, VDI/VDE, Jan. 2013.

[61] Gesellschaft Mess- und Automatisierungstechnik. Cyber-Physical Systems: Chancen und Nutzen aus Sicht der Automation. Technical report, VDI/VDE, Apr. 2013.

[62] R. Ghodssi and P. Lin. *MEMS materials and processes handbook*, volume 1. Springer, 2011.

[63] J. E. Gibson, W. T. Scherer, and W. F. Gibson. *How to do systems analysis*, volume 47. John Wiley & Sons, 2007.

[64] M. Goebel and L. Gruenwald. A survey of data mining and knowledge discovery software tools. *ACM SIGKDD Explorations Newsletter*, 1(1):20–33, 1999.

[65] G. H. Golub and C. F. Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.

[66] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing using MATLAB*, volume 2. Gatesmark Publishing Knoxville, 2009.

[67] A. Goshtasby. Piecewise linear mapping functions for image registration. *Pattern Recognition*, 19(6):459–466, 1986.

[68] M. Gram and C. Gugg. Einsatzsmöglichkeiten von Cyber-Physical Systems im Lebenszyklusmanagement von Anlagen. *Industrie Management*, 1:39–43, Feb. 2013.

[69] C. Gugg. Recognition of 3D Embossed Digits through Spectral Analysis and Machine Learning. Bachelor's thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2010.

[70] C. Gugg. Conceptual Development of an Opto-electronic Perpendicular Utilizing Electro-active Glass. Master's thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2011.

[71] C. Gugg, M. Harker, and P. O'Leary. Structural deformation measurement via efficient tensor polynomial calibrated electro-active glass targets. In *IS&T/SPIE Electronic Imaging, Image Processing: Machine Vision Applications VI*, volume 8661, pages 86610F–86610F–9, 2013.

[72] C. Gugg, M. Harker, and P. O'Leary. Virtual Calibration Environment for A-Priori Estimation of Measurement Uncertainty. In *IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 52–57, 2014.

[73] C. Gugg, M. Harker, P. O'Leary, and G. Rath. An algebraic framework for the real-time solution of inverse problems on embedded systems. *arXiv Open Access Journal Article, arXiv:1406.0380*, 2014.

[74] C. Gugg and P. O'Leary. Robust machine vision based displacement analysis for tunnel boring machines. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Pisa, Italy, May 2015.

[75] C. Gugg, P. O'Leary, and M. Harker. Large scale optical position sensitive detector. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1775–1780, May 2013.

[76] C. Gugg, P. O'Leary, and M. Harker. Analytic Multi-Source Data Fusion and its Application to a Large-Scale Optical PSD. *Instrumentation and Measurement, IEEE Transactions on*, 63(5):1116–1126, May 2014.

[77] M. Gupta, J. Gao, C. Aggarwal, and J. Han. Outlier Detection for Temporal Data. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 5(1), 2014.

[78] D. W. Ha, H. S. Park, S. W. Choi, and Y. Kim. A Wireless MEMS-Based Inclinometer Sensor Node for Structural Health Monitoring. *Sensors*, 13(12):16090–16104, 2013.

[79] M. Habacher, P. O'Leary, M. Harker, and J. Golser. Machine vision system for the control of tunnel boring machines. In *IS&T/SPIE Electronic Imaging*, pages 86610G–86610G. International Society for Optics and Photonics, 2013.

[80] R. Haberfellner and W. F. Daenzer. *Systems Engineering: Methodik und Praxis.* Orell Füssli Verlag, Zürich, 1999.

[81] G. Hackmann, F. Sun, N. Castaneda, C. Lu, and S. Dyke. A holistic approach to decentralized structural damage localization using wireless sensor networks. In *Real-Time Systems Symposium, 2008*, pages 35–46. IEEE, 2008.

[82] E. Hagendorfer. Data Mining of Time Series and Sequences for Machine Diagnostics. Bachelor's thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2014.

[83] X. Haiping and S. Shatz. A framework for model-based design of agent-oriented software. *Software Engineering, IEEE Transactions on*, 29(1):15–30, Jan. 2003.

[84] D. Hall and S. McMullen. *Mathematical Techniques in Multisensor Data Fusion.* Artech House Information Warfare Library, 2nd edition, Feb. 2004.

[85] D. L. Hall and A. K. Garga. Pitfalls in data fusion (and how to avoid them). In *Proceedings of the Second International Conference on Information Fusion (Fusion 1999)*, volume 1, pages 429–436, 1999.

[86] J. Han, M. Kamber, and J. Pei. Data Mining: Concepts and Techniques, (The Morgan Kaufmann Series in Data Management Systems). 2006.

[87] M. Harker. *Algebraic and Geometric Techniques for Optimization in Metric Vision.* PhD thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2008.

[88] M. Harker, C. Gugg, and P. O'Leary. Model Based Design for the Real-Time Solution of Inverse Problems in Cyber-Physical Systems. In *Proceedings of the Fifth International Conference on Cyber-Physical Systems (ICCPS)*, page 217, 2014.

[89] M. Harker and P. O'Leary. A Matrix Framework for the Solution of ODEs: Initial-, Boundary-, and Inner-Value Problems. *arXiv:1304.3312*, 2013.

[90] M. Harker and P. O'Leary. Regularized reconstruction of a surface from its measured gradient field. *Journal of Mathematical Imaging and Vision*, pages 1–25, 2013.

[91] W. S. Harrison, D. M. Tilbury, and C. Yuan. From Hardware-in-the-Loop to Hybrid Process Simulation: An Ontology for the Implementation Phase of a Manufacturing System. *Automation Science and Engineering, IEEE Transactions on*, 9(1):96–109, 2012.

[92] U. Harten. *Physik: Einführung für Ingenieure und Naturwissenschaftler.* Springer, 2005.

[93] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2004.

[94] IBM. Rational DOORS. Tech. spec., `www.ibm.com/software/products/en/ratidoor`, 2014.

[95] I. A. Ibragimov, R. Z. Has'minskii, and S. Kotz. *Statistical estimation: asymptotic theory*, volume 2. Springer-Verlag, 1981.

[96] IEC 60825-1:2001-08. Safety of laser products. International standard.

[97] IEC 61131-3:2013. Programmable controllers - Part 3: Programming languages. International standard.

[98] IEC 61508-1:2010. Functional safety of electrical/electronic/programmable electronic safety-related systems. International standard.

[99] IEEE 1076-2008. Standard VHDL language reference manual. Manual.

[100] IEEE 1364-2001. Verilog Hardware Description Language (HDL). Manual.

[101] IEEE 1451.4. Transducer Electronic Data Sheets (TEDS). Tech. spec., `standards.ieee.org`, 2004.

[102] IEEE 1666:2011. Standard for Standard SystemC Language Reference Manual. Tech. spec., `standards.ieee.org`.

[103] Intel. Composer XE. Tech. spec., `software.intel.com/en-us/intel-composer-xe`, 2014.

[104] ISO 26262-4:2011. Road vehicles – Functional safety – Part 4: Product development at the system level. International standard.

[105] JCGM 100:2008. ISO/IEC guide 98-3:2008: Evaluation of measurement data: Guide to the expression of uncertainty in measurement (GUM). International standard.

[106] JCGM 200:2012. ISO/IEC guide 99:2007: International vocabulary of metrology: Basic and general concepts and associated terms (VIM). International standard.

[107] J. C. Jensen, D. H. Chang, and E. A. Lee. A model-based design methodology for cyber-physical systems. In *Wireless Communications and Mobile Computing Conference (IWCMC), 7th International*, pages 1666–1671. IEEE, 2011.

[108] H. Kagermann, W. Wahlster, and J. Helbig. Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0 - Abschlussbericht des Arbeitskreises Industrie 4.0. *Forschungsunion im Stifterverband für die Deutsche Wissenschaft. Berlin*, 2012.

[109] K. Keutzer, A. R. Newton, J. M. Rabaey, and A. Sangiovanni-Vincentelli. System-level design: orthogonalization of concerns and platform-based design. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 19(12):1523–1543, 2000.

[110] F. Kirschke-Biller, S. Fürst, S. Lupp, S. Bunzel, S. Schmerler, R. Rimkus, et al. AUTOSAR – A worldwide standard: Current developments, roll-out and outlook. In *15th International VDI Congress Electronic Systems for Vehicles, Baden-Baden, Germany*, 2011.

[111] N. Koller. *Fully Automated Repair of Surface Flaws using an Artificial Vision Guided Robotic Grinder.* PhD thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2007.

[112] D. Kolymbas. *Geotechnik.* Springer, 1998.

[113] E. Kreyszig. *Advanced Engineering Mathematics.* John Wiley and Sons, 8th edition, 1999.

[114] N. Kumar, V. N. Lolla, E. Keogh, S. Lonardi, and C. Ratanamahatana. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *SDM*, pages 531–535. SIAM, 2005.

[115] L. Kurgan and P. Musilek. A survey of knowledge discovery and data mining process models. *The Knowledge Engineering Review*, 21(01):1–24, 2006.

[116] C. Lanczos. *Linear Differential Operators*. Dover books on mathematics. Dover Publications, 1997.

[117] C. Lattner and V. Adve. LLVM: A compilation framework for lifelong program analysis & transformation. In *Code Generation and Optimization (CGO). International Symposium on*, pages 75–86. IEEE, 2004.

[118] S. Laxman and P. S. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 2006.

[119] E. A. Lee. Cyber Physical Systems: Design Challenges. In *Object Oriented Real-Time Distributed Computing (ISORC), 11th IEEE International Symposium on*, pages 363–369. IEEE, 2008.

[120] S. H. Lee, S. Lee, H. Song, and H.-S. Lee. Wireless sensor network design for tactical military applications: Remote large-scale environments. In *Military Communications Conference (MILCOM), IEEE*, pages 1–7, Oct. 2009.

[121] T. Lennon and N. Mass. Model-based design for mechatronic systems. *Electronics World*, 1865:23, 2008.

[122] R. Leupers. Code generation for embedded processors. In *System Synthesis, 2000. Proceedings. The 13th International Symposium on*, pages 173–178. IEEE, 2000.

[123] J. Lin. Measuring return on investment of model-based design. *EE Times Design*, 2011.

[124] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM, 2003.

[125] J. Linkemann and B. Weber. Global Shutter, Rolling Shutter - Funktionsweise und Merkmale zweier Belichtungsverfahren. White paper, `www.baslerweb.com`, Jan. 2014.

[126] S. Lipson, H. Lipson, and D. Tannhauser. *Optik*. Springer Verlag, Berlin, 1997.

[127] G. Litfin. *Technische Optik in der Praxis*. Springer, 1997.

[128] M. Löffler-Mang. *Optische Sensorik: Lasertechnik, Experimente, Light Barriers*. Vieweg Teubner Verlag/Springer Fachmedien Wiesbaden GmbH, 2012.

[129] G. Machan and V. G. Bennett. Use of inclinometers for geotechnical instrumentation on transportation projects: State of the practice. *Transportation Research E-Circular*, (E-C129), 2008.

[130] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, and A. Marrs. *Disruptive technologies: Advances that will transform life, business, and the global economy*. McKinsey Global Institute, 2013.

[131] Maplesoft. Maple CodeGeneration Package. Tech. spec., `www.maplesoft.com`, 2013.

[132] Ó. Marbán, G. Mariscal, and J. Segovia. A data mining & knowledge discovery process model. *Data Mining and Knowledge Discovery in Real Life Applications*, pages 1–17, 2009.

[133] Mathworks. MATLAB Coder TB. Tech. spec., `www.mathworks.com/products/matlab-coder/`, 2013.

[134] Mathworks. Simulink Embedded Coder. Tech. spec., `www.mathworks.com/products/embedded-coder/`, 2013.

[135] Mathworks. Computer Vision System TB. Tech. spec., `www.mathworks.de/products/computer-vision/`, 2014.

[136] Mathworks. Curve Fitting TB. Tech. spec., `www.mathworks.de/products/curvefitting/`, 2014.

[137] Mathworks. Image Acquisition TB. Tech. spec., `www.mathworks.de/products/imaq/`, 2014.

[138] Mathworks. Image Processing TB. Tech. spec., `www.mathworks.de/products/image/`, 2014.

[139] Mathworks. Parallel Computing TB. Tech. spec., `www.mathworks.com/products/parallel-computing/`, 2014.

[140] Mathworks. Polyspace Code Prover. Tech. spec., `www.mathworks.de/products/polyspace/`, 2014.

[141] J. Mattingley and S. Boyd. CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

[142] M. Messing. Steuerung der Tunnelbohrmaschine am Gotthard. *Geomatik Schweiz: Geoinformation und Landmanagement*, 108(12):571, 2010.

[143] Microsft. Visual Studio 2010. Tech. spec., `www.microsoft.com/visualstudio/eng`, 2010.

[144] Mine Subsidence Engineering Consultants. Introduction to Longwall Mining and Subsidence. Technical report, Mine Subsidence Engineering Consultants, 2007.

[145] MISRA-C:2004. Guidelines for the use of the C language in critical systems. Technical standard.

[146] H. Mitchell. *Data Fusion: Concepts and Ideas*. Springer Verlag, 2nd edition, Feb. 2012.

[147] P. J. Mosterman. Automatic code generation: Facilitating new teaching opportunities in engineering education. In *Frontiers in Education Conference, 36th Annual*, pages 1–6. IEEE, 2006.

[148] P. J. Mosterman, J. Ghidella, and J. Friedman. Model-based design for system integration. *Proceedings of the Canadian Engineering Education Association*, 2011.

[149] P. J. Mosterman, S. Prabhu, and T. Erkkinen. An industrial embedded control system design process. *Proceedings of the Canadian Engineering Education Association*, 2011.

[150] P. J. Mosterman and J. Zander. Advancing Model-Based Design by Modeling Approximations of Computational Semantics. In *Object-Oriented Modeling Languages and Tools (EOOLT)*, pages 3–7, 2011.

[151] R. Mukundan. Some computational aspects of discrete orthonormal moments. *Image Processing, IEEE Transactions on*, 13(8):1055–1059, 2004.

[152] R. Mukundan. Transform coding using discrete Tchebichef polynomials. 2006.

[153] B. Mulgrew, P. M. Grant, and J. Thompson. *Digital signal processing: concepts and applications*. Macmillan Press, 1999.

[154] B. Müller. Mensch im Mittelpunkt. In C. Heise, A. Heise, and M. Pietschmann, editors, *Technology Review*. Heise Zeitschriften Verlag, 2013.

[155] National Instruments. LabVIEW C Generator. Tech. spec., `sine.ni.com/nips/cds/view/p/lang/en/nid/209015`, 2013.

[156] Network Working Group. RFC2460: Internet Protocol, Version 6 (IPv6). Tech. spec., `www.ietf.org/rfc/rfc2460.txt`, 1998.

[157] R. Neumayr. *New Approaches to Machine Vision Based Displacement Analysis*. PhD thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2011.

[158] R. Neumayr, M. Harker, P. O'Leary, and J. Golser. New approaches to machine vision based displacement analysis. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 979–984, May 2012.

[159] Object Management Group. SysML specification. Tech. spec., `www.omgsysml.org`, 2007.

[160] Object Management Group. UML 2.4.1 Specification. Tech. spec., `www.omg.org/spec/UML/2.4.1/Infrastructure`, 2011.

[161] P. O'Leary, C. Gugg, M. Harker, and G. Rath. Mathematical Model and Software Architecture for the Solution of Inverse Problems Involving Sensor Arrays. In *IEEE Sensors Proceedings*, pages 586–589, 2014.

[162] P. O'Leary and M. Harker. An Algebraic Framework for Discrete Basis Functions in Computer Vision. *Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, pages 150–157, 2008.

[163] P. O'Leary and M. Harker. Discrete polynomial moments for real-time geometric surface inspection. *Journal of Electronic Imaging*, 18(1):013015–013015–13, 2009.

[164] P. O'Leary and M. Harker. Discrete polynomial moments and Savitzky-Golay smoothing. *Waset Special Journal*, 72:439–443, 2010.

[165] P. O'Leary and M. Harker. A Framework for the Evaluation of Inclinometer Data in the Measurement of Structures. *Instrumentation and Measurement, IEEE Transactions on*, 61(5):1237–1251, 2012.

[166] P. O'Leary, M. Harker, and C. Gugg. A Position Paper on: Sensor-Data Analytics in Cyber Physical Systems, from Husserl to Data Mining. In *INSTICC Sensornets Proceedings*, 2015.

[167] P. O'Leary, M. Harker, and R. Neumayr. Savitzky-Golay smoothing for multivariate cyclic measurement data. In *Instrumentation and Measurement Technology Conference (I2MTC), 2010 IEEE*, pages 1585–1590. IEEE, 2010.

[168] P. O'Leary, M. Harker, and T. Suesut. Combined polynomial and periodic moments for the analysis of measured 3D surfaces. In *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, pages 354–358. IEEE, 2008.

[169] P. O'Leary, P. Lee, C. Gugg, and M. Harker. Tutorial: Mining sensor and measurement data in real-time. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Pisa, Italy, May 2015.

[170] A. V. Oppenheim, R. W. Schafer, J. R. Buck, et al. *Discrete-time signal processing*, volume 2. Prentice-hall Englewood Cliffs, 1989.

[171] B. Oswald-Tranta. Automated Thermographic Non-destructive Testing. Habilitation thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2013.

[172] K.-J. Park, R. Zheng, and X. Liu. Cyber-physical Systems: Milestones and Research Challenges. *Computer Communications*, 36(1):1–7, 2012.

[173] H. Partsch. *Requirements-Engineering systematisch*. Springer DE, 1998.

[174] S. Perry. Model based design needs high level synthesis: a collection of high level synthesis techniques to improve productivity and quality of results for model based electronic design. In *Proceedings of the*

Conference on Design, Automation and Test in Europe, pages 1202–1207. European Design and Automation Association, 2009.

[175] G. Pintér and I. Majzik. Model based automatic code generation for embedded systems. In *Proceedings of the Regional Conference on Embedded and Ambient Systems (RCEAS 2007), Budapest, Hungary*, pages 97–106, 2007.

[176] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

[177] S. Pretzler. Discrete Basis Function Methods for the Solution of Inverse Problems in Mechanical Measurements. Master's thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2011.

[178] M. Pucher, P. O'Leary, C. Gugg, and C. Hoefer-Oellinger. An autonomous water sampling and monitoring device for deployment in harsh underground environment. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Pisa, Italy, May 2015.

[179] O. Rafique, M. Gesell, and K. Schneider. Targeting Different Abstraction Layers by Model-Based Design Methods for Embedded Systems: A Case Study. *Real-Time Computing Systems and Applications (RTCSA), IEEE Computer Society*, 2013.

[180] C. A. Ratanamahatana and E. Keogh. Making time-series classification more accurate using learned constraints. SIAM, 2004.

[181] I. Reindl. *Optical Configurations, Methods and Algorithms for the Automatic Inspection on Metallic Surfaces*. PhD thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2006.

[182] R. Ritt. Automatisierung und Optimierung des Inline-Tapelege-Prozesses. Bachelor's thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2014.

[183] L. Rongqing, Z. Rong, L. Guangxin, H. Weimin, Z. Xiaoqian, and T. Koo. A Prototype of Model-Based Design Tool and Its Application in the Development Process of Electronic Control Unit. In *Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*, pages 236–242, July 2011.

[184] RTCA DO-178C. Software considerations in airborne systems and equipment certification. Technical standard, www.rtca.org.

[185] C. Sallinger. *Inspection and Repair of Refractories Using a Telerobot Guided by Panoramic Optical Servoing*. PhD thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2004.

[186] A. Savitzky and M. J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[187] E. Schäfer. *Technische Optik: mathematische und physikalische Grundlagen; mit 39 Beispielen und 4 Tabellen*. Vieweg, 1997.

[188] P. Schalk. *Metric Vision Methods for Material and Product Inspection*. PhD thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2007.

[189] S. Schmerler, D. Aiello, A. Continental, R. Rimkus, G. Motors, A. Gilberg, P. P. Citroën, A. Titze, and A. Volkswagen. AUTOSAR–Shaping the Future of a Global Standard. In *5th VDI Congress Baden-Baden Spezial*, 2012.

[190] R. Schmidt. Automatisierte Thermographische Rissprüfung bei komplexer Bauteilgeometrie. Master's thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2014.

[191] M. Shahnawaz, A. Ranjan, and M. Danish. Temporal data mining: An overview. *International Journal of Engineering and Advanced Technology (IJEAT)*, 1(1), 2011.

[192] S. Shirmohammadi and A. Ferrero. Camera as the instrument: the rising trend of vision based measurement. *Instrumentation Measurement Magazine, IEEE*, 17(3):41–47, June 2014.

[193] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer, 2008.

[194] D. Spath, O. Ganschar, S. Gerlach, M. Hämmerle, T. Krause, and S. Schlund. *Produktionsarbeit der Zukunft-Industrie 4.0*. Fraunhofer IAO Stuttgart, 2013.

[195] D. Spath, S. Gerlach, M. Hämmerle, S. Schlund, and T. Strölin. Cyber-physical system for self-organised and flexible labour utilisation. *Personnel*, 50:22, 2013.

[196] Steering Committee for Foundations in Innovation for Cyber-Physical Systems. Strategic R&D Opportunities for 21st Century Cyber-Physical Systems. Technical report, National Institute of Standards and Technology (NIST), Jan. 2013.

[197] Stemmer. Das Handbuch der Bildverarbeitung, 2013.

[198] A. Stiller. Matrix reloaded. In C. Heise, A. Heise, and M. Pietschmann, editors, *c't Magazin für Computer Technik*. Heise Zeitschriften Verlag, 2014.

[199] T. Suesut. *Moments for Real-Time Geometric Surface Inspection.* PhD thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2008.

[200] P. Tabuada. Cyber-Physical Systems: Position paper. In *NSF Workshop on Cyber-Physical Systems*, 2006.

[201] S. Tamblyn, J. Henry, and J. Rapp. Accelerating NASA GN&C Flight Software Development. In L. Webb and R. Oxenford, editors, *MathWorks News&Notes.* The MathWorks, Inc., 2011.

[202] A. Tarantola. *Inverse problem theory and methods for model parameter estimation.* Siam, 2005.

[203] M. Tratnig. *Calibration and Registration Approaches for Light-Sectioning Setups Featuring Small Fields of View.* PhD thesis, University of Leoben, Department Product Engineering, Chair of Automation, 2005.

[204] VDI 2206. Design methodology for mechatronic systems. Technical standard.

[205] M. Violante. Teaching Model-Based Design at Politecnico di Torino. In L. Webb and R. Oxenford, editors, *MathWorks News&Notes.* The MathWorks, Inc., 2013.

[206] G. Volkmann and W. Schubert. The use of horizontal inclinometers for the optimization of the rock mass–support interaction. In *Underground space use: Analysis of the past and lessons for the future, World tunneling congress 2005*, pages 967–972, 2005.

[207] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister. Smart Dust: Communicating with a Cubic-Millimeter Computer. *Computer*, 34(1):44–51, Jan. 2001.

[208] B. Weber. Wie viele Pixel braucht man wirklich? White paper, www.basler-ipcam.com, Dec. 2002.

[209] Wolfram. Mathematica. Tech. spec., www.wolfram.com, 2013.

[210] T. H. Wonnacott and R. J. Wonnacott. *Regression: a second course in statistics.* Wiley and Sons, 1981.

[211] D. Wynn, P. J. Clarkson, and C. Eckert. A model-based approach to improve planning practice in collaborative aerospace design. In *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 537–548. American Society of Mechanical Engineers, 2005.

[212] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.

[213] L. E. Young and H. H. Stoek. Subsidence resulting from mining. Technical report, University of Illinois at Urbana Champaign, College of Engineering, 1916.

[214] S. Young. *Real Time Languages: Design and Development.* Ellis Horwood Publishers, Chichester, 1982.

[215] P. M. Ziegler. Losgröße 1. In C. Heise, A. Heise, and M. Pietschmann, editors, *c't Magazin für Computer Technik.* Heise Zeitschriften Verlag, 2013.