Chair of Drilling and Completion Engineering

Master Thesis

Mechanical Stuck Pipe Events - Development of Digital Best Practices for NPT Prevention

Julian Klaus Pflügl, BSc

Mai 2019

It gives me great pride to dedicate this thesis to my family which constantly supported me during the past six years of my studies.

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.

Ich erkläre, dass ich die Richtlinien des Senats der Montanuniversität Leoben zu „Gute wissenschaftliche Praxis" gelesen, verstanden und befolgt habe.

Weiters erkläre ich, dass die elektronische und gedruckte Version der eingereichten wissenschaftlichen Abschlussarbeit formal und inhaltlich identisch sind.

<div align="center">

_____

Unterschrift Verfasser

Julian Klaus Pflügl

Matrikelnummer: 01335392

13 May 2019

</div>

Abstract

Mechanical stuck pipe events are one of the key technical challenges of the drilling industry and incidents usually lead to an enormous amount of non-productive time and costs. Most of these events show that they could have been avoided by an early reaction to possible sticking indications. To improve the reaction time and thereby the efficiency of future drilling operations and to reduce the mechanical stuck pipe related NPT, a state-of-the-art drilling simulator in combination with knowledge of industry experienced personnel gets used to design an interactive and realistic training case.

To provide a detailed analysation of different types of mechanical stuck pipe issues, relevant technical literature and the available best practice solutions within the industry get summarised. Each type is reviewed for its symptoms, preventive actions and freeing procedures. Using eDrilling AS′s state-of-the-art drilling simulator software, a tailor-made mechanical stuck pipe training scenario based on theoretical data is designed and implemented into the simulator. A case-based best practice solution to complete the scenario successfully gets provided. Additionally, a handbook for the simulator and its user interfaces and a step-by-step guide for the development and implementation of a general training case is created.

After the evaluation of the drilling simulator opportunities, mechanical sticking due to a dogleg was established as a training case. It was found out that this type will provide the most realistic and reliable result. Not any parameter can be varied within the simulator software due to certain limitations from the provider. Besides the development from a theoretical idea to the actual simulation code and the analysation of important drilling parameters related to mechanical sticking, essential parts of the coding files get explained. Moreover, it was found that mostly the freeing procedures and not the recommended reactions to prior indications get described in the literature. Therefore, a best practice summary for the training scenario including preventive measures and freeing procedures is created. The detailed description of the simulator software including a guide for every application and user interface is originated. What remains unsolved is the actual impact on the reaction improvement during a real stuck pipe scenario of people who practised this training case on the simulator compared to others. Therefore, carrying out a study about the impact of the training on the simulator is highly recommended as future work.

This is the first interactive training case on the state-of-the-art eDrilling AS simulator which confronts engineers and operational personnel with mechanical stuck pipe issues. It allows the unique possibility to practise identifying crucial warning signs as wells as reacting appropriately to such events. It supports the transformation of trainees into highly qualified personnel. With the provided workflow description and software code, users are able to understand how to work with such a simulator and to design and implement new digital drilling-related training cases.

Zusammenfassung

Mechanisch festgewordene Böhrstränge sind eine der wichtigsten technischen Herausforderungen der Bohrindustrie. Zwischenfälle führen zu einer enormen Summe an unproduktiver Zeit und Kosten. Die meisten dieser Ereignisse zeigen, dass sie durch eine frühzeitige Reaktion auf mögliche Warnzeichen hätten vermieden werden können. Um die Reaktionszeit und damit die Effizienz zukünftiger Bohrvorgänge zu verbessern, und die mit mechanisch festgewordenen Bohrsträngen verbundene unproduktive Zeit zu verringern, wird ein moderner Bohrsimulator verwendet, um ein interaktives und realistisches Trainingsszenario zu entwerfen.

Um eine detaillierte Analyse der verschiedenen Arten von mechanisch festgewordenen Bohrsträngen zu ermöglichen, werden relevante Fachliteratur und die verfügbaren Best-Practice-Lösungen in der Branche zusammengefasst. Unter Verwendung der hochmodernen Bohrsimulatorsoftware von eDrilling AS wird ein maßgeschneidertes Trainingsszenario für mechanisch festsitzende Bohrstränge basierend auf theoretischen Daten entworfen, und in den Simulator implementiert. Eine fallbasierte Best-Practice-Lösung zum erfolgreichen Abschließen des Szenarios wird bereitgestellt. Zusätzlich wird ein Handbuch für den Simulator und seine Benutzeroberflächen sowie eine schrittweise Anleitung für die Entwicklung und Implementierung eines allgemeinen Trainingsfalls erstellt.

Es wurde ein mechanisches Festwerden durch einen Bohrlochknick (dogleg) als Übungsfall etabliert. Dieser Fehlertyp liefert das realistischste und zuverlässigste Ergebnis. Aufgrund bestimmter Einschränkungen des Anbieters kann in der Simulatorsoftware nicht jeder beliebige Parameter geändert werden. Neben der Analyse wichtiger Bohrparameter im Zusammenhang mit mechanischem Festwerden, werden wesentliche Teile der Codierungsdateien erläutert. Es wird eine Best-Practice-Zusammenfassung für das Trainingsszenario einschließlich vorbeugender Maßnahmen und Lösungsverfahren erstellt. Die detaillierte Beschreibung der Simulatorsoftware einschließlich einer Anleitung für jede Anwendung und Benutzeroberfläche wurde erstellt. Was ungelöst bleibt, ist die tatsächliche Auswirkung auf die Reaktionsverbesserung während eines realen Szenarios von Personen, die diesen Trainingsfall auf dem Simulator geübt haben, im Vergleich zu anderen. Es wird daher dringend empfohlen, in Zukunft eine Studie über die Auswirkungen des Trainings auf dem Simulator durchzuführen.

Dies ist der erste interaktive Trainingsfall für den hochmodernen eDrilling AS Simulator, der Ingenieure und Betriebspersonal mit Problemen in Bezug auf festgewordene Bohrstränge konfrontiert. Dies bietet die einzigartige Möglichkeit, wichtige Warnzeichen zu erkennen und auf solche Ereignisse angemessen zu reagieren. Die Transformation von Auszubildenden in hochqualifiziertes Personal wird unterstützt. Mit der bereitgestellten Beschreibung des Simulators und dem Softwarecode können Benutzer neue digitale Trainingsszenarien für das Bohren entwerfen und implementieren.

Acknowledgements

x

Contents

Chapter 1 Introduction

Prevention of stuck pipe is far more economic than even the best of freeing procedures. But when prevention fails, the operator must move swiftly to select the most effective treatment (Adelung et al. 1991).

Due to the current oil price and energy market situation, it is obligatory to reduce the non-productive time (NPT) and achieve the maximum levels of operational excellence. This is vital for oil and gas companies to overcome the upcoming economic challenges like budget and spending reductions or enhancement of the return on investment (Aljubran, Al-Yami, and Madan 2017).

Mechanical stuck pipe events are one of the most common NPT causes in the drilling industry and therefore, cost a lot of money. As an example, the study of the Menengai (Kenya) field showed that, on average, stuck pipe events lasted for six days and made up 12% of the total drilling time of a well in this field. Back in 1991, stuck pipe events summed up to USD 200-500 million spent annually in the drilling industry and arose in 15% of all wells drilled. In 2009, about 69,5% of all stuck pipe events were due to mechanical sticking. Moreover, in 2010, 25% of the total NPT were due to stuck pipe events (Muqeem, Weekse, and Al-Hajji 2012). Saudi Aramco reported over 24% of the overall drilling and workover time as NPT due to stuck pipe and borehole problems in their shaly offshore formations (Aljubran, Al-Yami, and Madan 2017).

Mechanical stuck pipe events do not only lead to NPT and loss of money but can also create severe safety issues in context with the wellbore and the people working on a drilling rig. A sudden release of high stresses in the drill string can generate huge forces which then can damage wellbore, downhole installations and human (Oketch 2014).

Currently, there cannot be set any threshold limits in the drillers cabin that alert the driller about the occurrence of a stuck pipe event on most of the drilling rigs. The responsible person needs to interpret them and make operational decisions. Often it is already too late when possible signs for mechanical stuck pipe get realized, and the problem becomes severe (Cullen and Kucs 2018).

The objective of the thesis is to improve the efficiency of drilling operations and reduce mechanical stuck pipe related NPT using state-of-the-art drilling simulators in combination with knowledge of industry experienced people. Tailor-made training scenarios, if available based on real data from OMV AG, will be established for engineers and operational personnel. In case no suitable data is available, imaginative scenarios will be created. Best practice for mechanical stuck pipe events will be summarized based on these cases.

There exist several best practices for avoiding stuck pipe and for counteracting the event once it occurred, but often the right solution is not performed. A best practice handbook for designed scenarios will be generated to solve this problem. The manual will allow students and professionals to train their skills on stuck pipe detection and mitigation.

Chapter 2 Types of Mechanical Stuck Pipe

The term "stuck pipe" describes the drill string to be suspended and cannot be recovered or pulled back out of hole, regardless of the ability or inability to rotate it, lower it or to circulate (Aljubran, Al-Yami, and Madan 2017). Equation (1) describes the condition of a stuck pipe. If the frictional force (FF) created by side forces on the walls in the wellbore plus the sticking force (SF) exerted on the string exceeds the maximum overpull (MO) capacity of the rig or pipe, the string is considered to be stuck (Schlumberger and Anadrill 1997).

$$FF + SF > MO \qquad (1)$$

As already stated, in 2009 more than two-thirds of all reported stuck pipe scenarios were due to mechanical sticking. In the following chapter, the most common types of mechanical sticking are described. Besides general information about each type, prevention, symptoms, and recommendations for freeing the pipe will be discussed. Figure 1 indicates how crucial it is to identify the mechanism and type of stuck pipe in time to be able to free the pipe. There is only a chance of about 10% to free the pipe after four hours of sticking (Alshaikh et al. 2018).
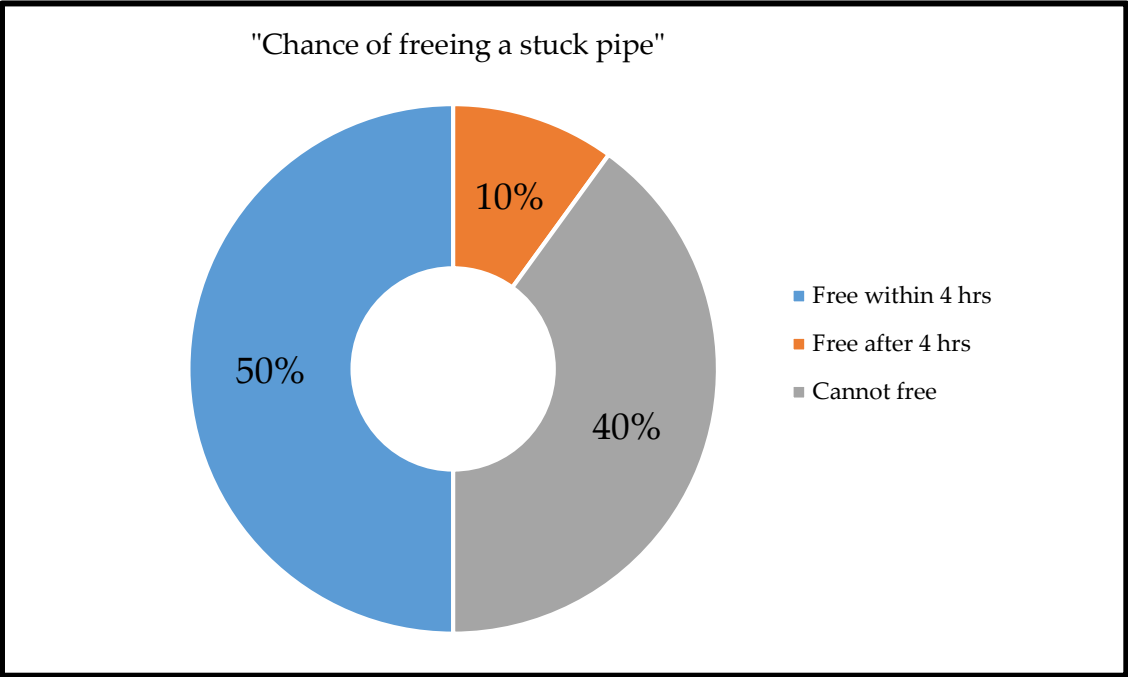

Figure 1: Chance of freeing a stuck pipe (Alshaikh et al. 2018)

2.1 Key Seating

Around 50 % of all stuck pipe cases in deep wells occur due to, or are related to, key seating (Brydon et al. 2015).

It occurs due to rotation of the drill pipe against the open hole or cased wellbore wall, generally in a deviated well. This wear can result in a furrow or so-called "key seat". If the key seat is deep enough and the drill string gets tripped out of the hole, the bottom hole assembly (BHA) or tool joints can get stuck. In case of a cased hole, the key seating typically happens at a worn casing shoe (Bowes and Procter 1997). The four fundamental factors influencing the formation of a key seat are dogleg severity[1] (DLS), pipe tension, pipe rotation and rotating time. Not all of them need to be present at the same time to allow a key seat to form. A standard output of any current torque and drag analysis software is the calculated side force at all points of the trajectory for the depth and conditions under which it is calculated – this side force is a function of the tension in the string and the dogleg. The more tension in the string and the larger the dogleg, the higher is the side force and therefore, the quicker a key seat can develop. Hence, doglegs up high in the wellbore are more severe in deep vertical wells because there is more pipe below the dogleg to create bigger tensional loads (Mitchell 2014).

This type of stuck pipe usually arises:

- Due to sudden changes in the drilling direction
- In deep vertical wells with doglegs up high in the wellbore
- In hard formations with long rotating periods on the same spot
- In case of high side wall forces combined with the rotation of the drill string
- While tripping out of the hole
- When no wiper trip or reaming job is performed in the deviated hole section during an extensive drilling period

Figure 2 shows an example of a key seat where soft formation layers get enlarged more than the harder layers.

---

[1] Dogleg severity field unit is degree per 100 feet [°/100ft], metric unit degree per 30 meter [°/30m]
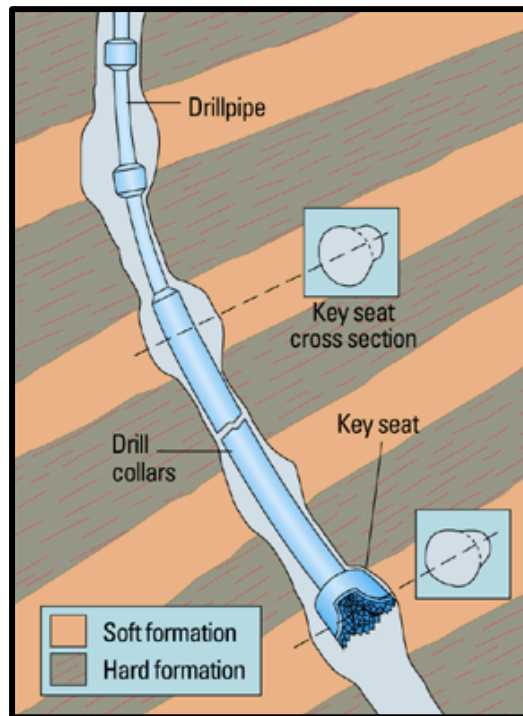
Figure 2: Example of key seating (Schlumberger 2019)

2.1.1 Prevention

As key seating usually happens in highly deviated wells, minimizing dogleg severity and side forces is very important. Avoiding sharp doglegs high in the wellbore in vertical wells, where the tension is at its maximum, is essential. It is proposed that the rotating time off bottom should be minimized because the side loads against a possible dogleg increase with increasing hook load (HL). The string should be rotated slowly while pulling the BHA through a problematic section. This action helps the drill collars and stabilizers to roll out of the key seat (Mitchell 2014). Furthermore, performing a reaming or wiper trip regularly in dogleg sections can be beneficial. A lot of tripping and tool handling time is saved by installing a reamer or key seat wiper in the drill string (Bowes and Procter 1997). Additionally, setting the casing right after the build or drop section to prevent key seating while drilling deeper should be considered (Zende 2018).

2.1.2 Symptoms

The most common symptoms for key seating on the rig site are (Bowes and Procter 1997) (Adelung et al. 1991):

- Abrupt overpull as the BHA or tool joint gets pulled into key seat
- Normal circulation behavior because the cross-sectional area of the annulus does not decrease
- This type can only occur when POOH
- Recurring overpulls while the tool joints pass the key seat
- Periodic torque and drag (T&D) peaks
- A sudden increase of drag is often an indicator for the BHA reaching highly inclined sections representing a potential key seat

- The stuck point matches to the dogleg
- The stuck point occurs at the same depth for recurring BHA trips

2.1.3 Actions to Free Stuck Pipe Due to Key Seating

It is recommended not to pull the string with maximum load and to move the pipe downwards while applying torque. Otherwise, jarring downwards and pumping a highly lubricating pill is advised. A jar is an impact tool installed in the drill string that delivers a sudden impact force upwards or downwards to free the pipe in case of a stuck pipe situation. A general rule for jarring operations in mechanical stuck pipe scenarios is to jar in the opposite direction of the last pipe movement before it became stuck. Furthermore, applying maximum torque while moving downward, but no torque while moving upward is proposed (Mitchell 2014). The jar can be actuated either mechanically or hydraulically. A spotting fluid or also called "pill", is a volume of fluid containing special treatment additives (lubricants, surfactants…) which gets pumped downhole to the critical section of stuck pipe. Lubricants are added to reduce friction and to help the pipe sliding around edges and through tight spots. If these methods are unsuccessful, back off[2] as near to the stuck point as possible, POOH the string and then use a fishing jarring string. Use a jarring string with an outside diameter (OD) small enough to pass through the key seat and try to work the string free. If the string can be moved down and rotated but cannot be pulled over the key seat, slowly rotate the string under minimum tension. Another option is the usage of a wireline resonance tool to lower the effective friction between the pipe and the key seat. The last option before drilling a side track is to back off again, holding the fish with a spear to prevent it from falling, and using a washover (Adelung et al. 1991) (Mitchell 2014).

---

[2] Back off = unscrew parts of drill string downhole (Schlumberger 2019)

2.2 Ledges / Doglegs

A ledge can form at the interface of different rock types with varying hardness when drilling through a formation. Soft/unconsolidated formation gets worn away, while the hard formation keeps its structure (Devereux 1998). Therefore, the interface is not as smooth as within one type of rock but can have sharp edges. A typical example of a formation prone to form ledges is the Wasia formation[3] with its interbedded sand-shale-carbonate layers. Ledges can easily form due to the brittle carbonates, and shale interbeds (Aljubran, Al-Yami, and Madan 2017).

The dogleg represents a section in the trajectory of a wellbore where rapid changes in the bit direction happen. These changes can either be created by the directional driller on purpose or due to formation related reasons. To quantify the dogleg, the term "dogleg severity" is used. A high dogleg severity usually brings along harmful side effects. While tripping in or out, parts of the BHA and tool joints can get stuck quite easily (Bowes and Procter 1997). This can happen if a stiff BHA such as casing or coring BHA, as shown in Figure 3, is run directly after a flexible one which was used to build up the angle. High lateral forces are required to bend the assembly to fit into the hole. Therefore, high frictional forces arise between the string and the wellbore wall (Zende 2018).
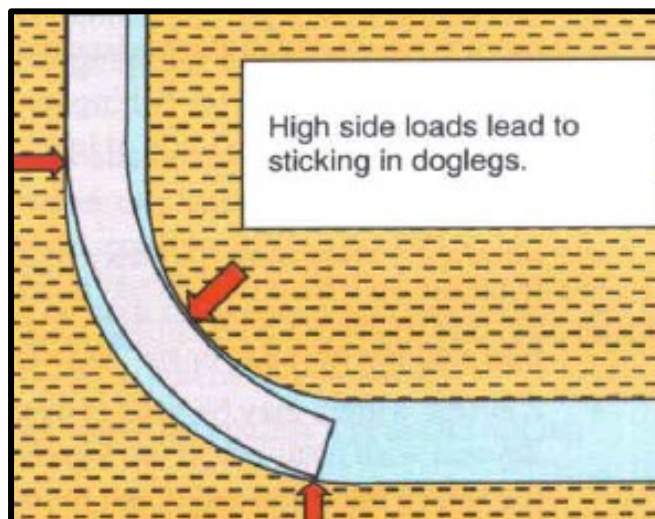
Figure 3: Dogleg (Mitchell 2014)

This mechanism of stuck pipe can occur (Aldred et al. 1999) (Mitchell 2014):

- When using an unsuitable BHA
- After forcing a stiff BHA into a dogleg
- When using a stiff BHA right after a flexible one
- During a casing running operation, if there is little clearance between the casing and the wellbore wall
- While running in hole (RIH) and pull out of hole (POOH)
- After drilling through interbedded formations with various types of rock
- When fractures or faults are encountered

---

[3] Arabian Gulf oil fields

- After quick changes in the drilling direction
- Due to "bit walk", if the bit is not stabilized adequately in vertical wells with interbedded formations
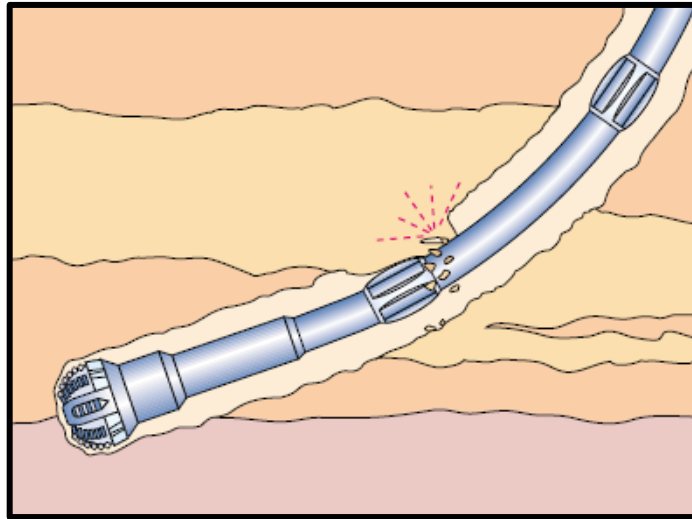
Figure 4: Ledge (Aldred et al. 1999)

Figure 4 shows a stuck pipe issue due to a ledge.

2.2.1 Prevention

If the formation is known to form ledges, minimizing the number of BHA changes and avoiding changes in the bit direction is extremely important. However, if the BHA configuration needs to be changed during the drilling process in such a formation, it must be ensured to perform a reaming job. Additionally, it is recommended to slow down the tripping speed before approaching a suspected ledge zone or dog leg (Bowes and Procter 1997). Consider using inhibitive mud to minimize the different ledge forming behavior of interbedded formations (Mitchell 2014). Avoid excessive circulation across soft interbedded sections and perform sufficient survey logs to identify problematic zones (Cullen and Kucs 2018). Running a stiff BHA right after a more flexible one should be avoided. The tubular stiffness is a function of diameter raised to the fourth power. Therefore, if the diameter of certain parts of the BHA is doubled, the stiffness gets increased times 16. It can be helpful to run a bit-sized hole opener trip with an undergauged bullnose by using low weight on bit (WOB) and higher revolutions per minute (RPM) before installing the stiff BHA. The maximum allowable set-down weight should be limited to pull back in time in case of too much resistance. (Mitchell 2014) (Devereux 1998).

2.2.2 Symptoms

The following indications are typical for stuck pipe due to ledges or dog legs (Bowes and Procter 1997) (Aljubran, Al-Yami, and Madan 2017):

- Unexpected overpull or set-down weight depending on the tripping direction
- Unrestricted circulation but an increase in pump pressure
- The problem commonly occurs at fixed depths

- Wear on the BHA and out of gauge downhole equipment pulled from the well could indicate severe doglegs
- Irregular T&D spikes
- Fluctuation in the rate of penetration (ROP) due to interbedding
- Large rock pieces on shaker system
- Deviation of modelled predicted and actual hook load while tripping in hole

2.2.3 Actions to Free Stuck Pipe Due to Ledges/Doglegs

If the movement of the string is still possible at least a bit, back reaming or reaming past the problem slowly while rotating the string should be considered (Bowes and Procter 1997). In case of a stuck pipe while moving upward, apply torque and jar downwards with maximum load. If the sticking occurred while tripping in, jarring upward without applying torque must be performed. Otherwise, the string might part due to too high forces. If a stiff BHA was forced into a dogleg, try to jar downward while applying torque though because the drill collars and stabilizers are more flexible under compression. If no jarring actions are working, use a spotting fluid to reduce the friction across the dogleg or to dissolve parts of the rock around the stuck point. Another option is to liquefy the rock surrounding the stuck point by using wireline resonance tools. As fishing jobs and drilling sidetracks are expensive, they should always be the last solution considered (Mitchell 2014).

2.3 Undergauged Hole

Drilling in a harsh environment like abrasive sands reduces the gauge of stabilizers and bit which leads to a smaller hole than planned. After the BHA is changed and running an in-gauge bit into the hole, it will experience resistance in the undergauged section from the previously worn bit. Figure 5 illustrates a typical undergauged hole scenario. If the driller runs the drill string in too fast, the bit could shove into this section and become stuck (Bowes and Procter 1997).
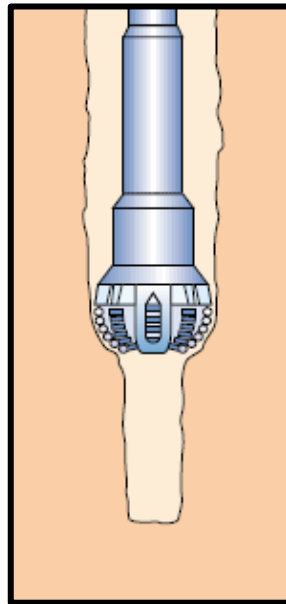
Figure 5: Undergauged hole (Aldred et al. 1999)

This kind of stuck pipe happens typically(Aldred et al. 1999):

- When a new in-gauge bit gets run
- After a coring job because coring bits are usually smaller than drill bits for the same section
- When using a polycrystalline diamond compact (PDC) bit after a roller cone bit
- When drilling in abrasive formations since the bit and stabilizers get worn

2.3.1 Prevention

One pro-active option to prevent undergauged hole stuck pipe is to use gauge-protected bits and stabilizers. Moreover, the installation of roller reamers in the drill string and the inspection of all problem-related tools of the BHA before and after running them is crucial. If a critical hole section gets identified, reaming before continuing the drilling operation is proposed. The tripping speed should be reduced when approaching an undergauged hole section to prohibit stuck pipe and prevent pinching of the bit due to high point-loading (Bowes and Procter 1997).

2.3.2 Symptoms

Typical symptoms for an undergauged hole stuck pipe problem are (Bowes and Procter 1997):

- Undergauged BHA components get identified during an inspection at the surface after tripping out
- This type can only occur when RIH
- A sudden increase of WOB
- The decrease of ROP due to a worn bit
- Normal or slightly restricted circulation behavior
- Stuck pipe happened while bit was near the bottom or the top of a cored interval

2.3.3 Actions to Free Stuck Pipe Due to Undergauged Hole

At first, it is recommended to jar up with the maximum load without using torque and pumping a lubricant pill to reduce the friction. Secondly, the usage of an acid pill for carbonates and a freshwater pill for salt to dissolve the rock around the stuck point is advised. Torqueing should be the last option to be selected before performing a remedial action like back-off, fishing or sidetrack. (Bowes and Procter 1997).

2.4 Junk

This type of sticking happens if objects from the surface get dropped accidentally into the hole due to human errors or parts of downhole equipment break off. This debris/junk, shown in Figure 6, settle somewhere in the wellbore where the drill string can quickly get stuck. Reasons for junk in the hole are (Bowes and Procter 1997):

- The rig crew did not install the hole cover on the rig floor while no string was in place
- An unexpected failure of a downhole tool
- Inadequate housekeeping on the rig floor
- Cement blocks of previous cementing operations
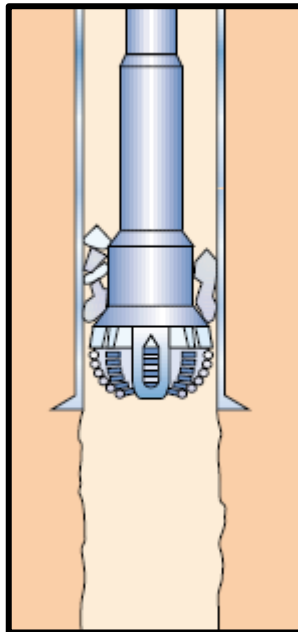
Figure 6: Junk in the hole (Aldred et al. 1999)

2.4.1 Prevention

Regularly clean-up of the rig floor is advised and as a supervisor encourage the drilling crew to adequate housekeeping. The downhole equipment should be inspected after every BHA configuration change before it is run back in the hole. Another preventive tool which should get installed as quickly as possible while tripping is the drill string wiper rubber. It avoids, similar to the hole cover, unwanted objects from falling into the wellbore (Bowes and Procter 1997). Also close the blind rams of the blow out preventer (BOP) when the BHA is out of the hole if there are no problems with hole. Do not forget to open them again prior to running the next BHA.

2.4.2 Symptoms

Stuck pipe due to junk can be recognised by (Bowes and Procter 1997):

- Metal splinters from junk observed on the shale shakers
- Abrupt torque changes
- Slightly restricted circulation
- Missing tools from the rig floor

Table 1 shows the symptoms of a stuck pipe event due to junk in the Menengai field (Oketch 2014).

Well	Depth stuck [m]	Torque	WOB	(ROP)	Returns	Other
MW09	1984	Sudden drop	Unchanged	Unchanged	Partial	RPM suddenly to 0

Table 1: Example of a stuck pipe due to junk (Oketch 2014)

2.4.3 Actions to Free Stuck Pipe Due to Junk

Several general suggestions to mitigate this problem do exist (Bowes and Procter 1997):

- Make sure that an unrestricted circulation is provided
- In case of a stuck pipe while tripping out, apply torque and fire the jar downward
- In case of a stuck pipe while tripping in, jar upward without applying torque
- To perform the jarring job, start with small loads and then systematically increase the load to the maximum over one hour
- Pump lubricant pill to support freeing operation (Adelung et al. 1991)

Figure 7 shows an example of a retrieved piece of junk after a stuck pipe scenario(Cullen and Kucs 2018).

Figure 7: Junk retrieved after stuck pipe event (Cullen and Kucs 2018)

2.5 Collapsed Casing

Casing collapse, as illustrated in Figure 8, occurs if the casing cannot withstand the pressure from outside anymore. Either the existing pressure conditions exceed the initially proposed collapse pressure rating, or the collapse pressure rating is not valid anymore due to wear and corrosion problems. Already slight reductions of the casing wall thickness due to friction and corrosion can have a significant impact on the collapse pressure resistance. After the casing collapses, it pushes against the drill string which gets stuck. Causes for a collapsed casing can be (Bowes and Procter 1997):

- The casing gets damaged/buckled due to incautious running procedures
- Casing pressure rating is not valid anymore or gets exceeded due to wear and corrosion
- If too light displacement fluid gets used while cementing a casing
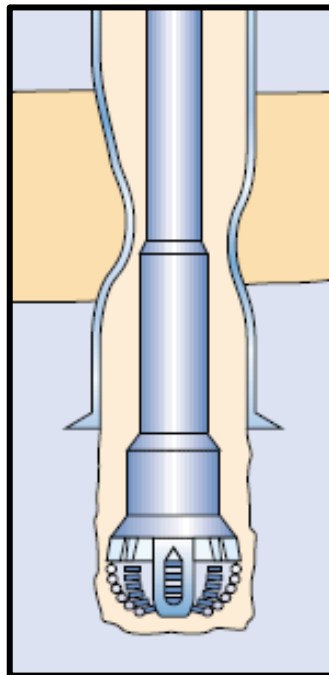
Figure 8: Collapsed casing (Aldred et al. 1999)

2.5.1 Prevention

To protect the casing, wear must be avoided, and corrosion inhibitors added to the drilling fluid. The drilling crew should precisely inspect the casing before installing it. Moreover, a well-performed cement job is essential. An accurate casing design process is crucial to prevent collapse due to arithmetic mistakes. (Bowes and Procter 1997).

2.5.2 Symptoms

Typical symptoms for a collapsed casing are (Bowes and Procter 1997):

- While RIH, a sudden decrease in HL occurs
- Using a caliper log should indicate a collapsed casing
- Restricted circulation
- Abrupt torque change
- Sudden overpull while POOH

2.5.3 Actions to Free Stuck Pipe Due to Collapsed Casing

At first, it is recommended to jar up with the maximum load without applying torque. Secondly, the usage of a lubricant pill to reduce the friction around the stuck point is a freeing possibility. Applying torque should be the last option to be selected before performing a remedial action like backing-off, fishing or drilling a sidetrack. (Bowes and Procter 1997).

2.6 Green Cement

Green cement is defined as cement which has not fully hardened. Stuck pipe happens if the cement was not prepared adequately and the time for waiting on cement (WOC) to harden differs from the expected one. After a cement job, the drill string gets run into the hole to drill away the remaining cement. While the driller thinks the BHA has not reached the top of the cement, it passes already through green cement. At some point, the bit reaches hardened cement and the drilling operation starts. After the green cement on top flash sets[4], the string gets stuck. The problem is illustrated in Figure 9. (Bowes and Procter 1997).

Figure 9: Green cement (Bowes and Procter 1997)

2.6.1 Prevention

It is critical to know the exact cement setting time, and not relying on assumptions when tripping back in the hole. Moreover, the correct volume and the height of the remaining cement column in the casing should be known. Always suppose it to be higher than expected. Never rely only on the weight indicators in the drillers cabin to approach the top of the cement (Bowes and Procter 1997).

2.6.2 Symptoms

Typical symptoms recognised on the rig site are (Bowes and Procter 1997):

- A constant increase in pump pressure, ending up with an inability to circulate
- Reduction of the HL
- Abrupt decrease of torque
- After a particular lag time, green cement appears in the mud returns, and discoloration of the mud takes place

---

[4] Cement hardens rapidly

2.6.3 Actions to Free Stuck Pipe Due to Green Cement

Before jarring up with maximum load, releasing any trapped pump pressure and working the string up and down is recommended. If jarring up does not work, the application of torque and jarring downwards while trying to re-establish circulation is essential. Another option is to pump an acid pill to dissolve the cement around the stuck pipe. If the cement fully hardens and none of the proposed freeing actions is successful, remedial measures need to be considered (Bowes and Procter 1997).

2.7 General Recommendations for Stuck Pipe Scenarios

An ounce of prevention is worth a pound of cure (Aljubran, Al-Yami, and Madan 2017). This statement should indicate how critical in terms of time and money freeing operations of stuck pipe are.

Two key factors for stuck pipe scenarios are training and crew consciousness. Investing the time to effectively brief the driller and other crew members on the practices expected to be followed, including when the supervisor/tool pusher should be called if there are issues, is highly recommended. An appropriate handover between drillers is crucial since many stuck pipe events happen within two hours of a shift change. The supervisor/tool pusher should be on the rig floor during the handover time as well. The maximum allowable pull on the pipe should always be known for enabling fast actions in case of a stuck pipe issue (Devereux 1998).

A five-step problem-solving process has been defined (Mitchell 2014):

The first step is to state the problem accurately. In this case, the problem would be identified as pipe sticking due to mechanical reasons. Often people panic to find the correct solution to the problem, although they did not define the problem upfront. This reaction can result in focusing on a cure for one of the causes, rather than a treatment for the initial problem itself.

After a stuck pipe event occurred, the second step is to identify the causes of the problem. There is rarely only one cause, but a whole chain of events leading to any of these problems. If the problem turns out to be caused by mechanical sticking, one of the reasons could be junk in the hole. Junk, in turn, could have been caused due to a failure of downhole equipment. The rig crew might have missed recording a crack within the last inspection of the equipment. Often removing only one cause in the event chain would prevent the problem from occurring.

The next step is to formulate a procedure on how to free the stuck pipe. Freeing stuck pipe is a challenge of combining gentle persuasion and physical force. Gentle persuasion consists of changing the downhole circumstances to reduce sticking, so the pipe comes free on its own or is more receptive to force. The force can be applied by pulling, slacking off, holding torque or by using a jar. Another option is to open and close the head valve in the mud pump several times to create pressure surges in the mud flow which might create vibrations of the string. The important tools of persuasion are spotting fluids and hole conditioning. In the past, diesel oil products were used and proved successful half the time, but due to environmental regulations, these products are often not allowed anymore. In carbonate formations, for example, acid is added to the pill to dissolve parts of the structure where the pipe got stuck. If the pipe gets stuck in salt formations, fresh water is used as a spotting fluid to dissolve the salt (Cullen and Kucs 2018). Hole conditioning means changing specific properties like flow rate or mud properties (e.g. pH, density). In the Gulf of Mexico, different companies had a fishing success rate between 33% and 50%. Whether fishing is worth it or not depends on the proposed costs of the fishing job. There exist many cases where drilling a side track was chosen over a fishing job. (Adelung et al. 1991). Before starting with the freeing operation of a stuck pipe, it is highly recommended to determine the overpull limits of the rig to avoid any

damage (Aljubran, Al-Yami, and Madan 2017). Overpull is described as the difference between the actual and expected force while POOH, while the difference during RIH is called set-down force. They exist due to frictional forces. If these two forces are zero, the wellbore is supposed to be "free". In case of an increase, the wellbore is said to be getting "tight" (Zende 2018).

The fourth step requires to choose between all the possible solutions and to apply the most suitable one. Some procedures could cause more harm than good. Jarring upward in the case of key seating, for instance, would make the problem even worse.

The final step is the evaluation of the solution likely process. It gets, unfortunately, like the first step, often skipped and the crew misses the chance of learning from it. The information gained in such post-problem meetings can improve future operations and prevent mistakes to be made.

Four basic questions need to be answered to determine what caused the string to be stuck (Mitchell 2014):

- What was the movement direction just before the string got stuck?
- Is downward movement possible after sticking?
- Is it possible to rotate the string?
- What is the circulation behavior after sticking?

Table 2 illustrates exactly these four questions and the three main types of sticking – solid induced sticking (here as pack-off bridge), differential sticking and mechanical sticking (here as wellbore geometry). The numbers represent probabilities from 0 (low) to 2 (high). As an example, the likelihood of mechanical sticking while the string is static equals zero. Circle the number in the row of each question and then sum all circled numbers up in each column. The column with the highest number, in this case wellbore geometry with 19 points, indicates the responsible most likely type of sticking (Mitchell 2014).

Questions	Pack-off Bridge	Differential	Wellbore Geometry
Pipe motion before sticking?			
Moving up	2	0	2
Rotating up	1	0	2
Moving down	1	0	2
Rotating down	0	0	2
Static	2	2	0
Pipe motion after sticking?			
Down free	1	0	2
Down restricted	1	0	2
Down impossible	0	2	1
Pipe rotation after sticking			
Rotate free	0	0	2
Rotation restricted	2	0	2
Rotation impossible	0	2	0
Circulation pressure after sticking?			
Circulation free	0	2	2
Circulation restricted	2	0	0
Circulation impossible	2	0	0
Total ∑	**14**	**8**	**19**

Table 2: Type of sticking worksheet (Mitchell 2014)

If none of the recommended freeing actions is successful, a fishing operation is necessary to remove/retrieve the stuck pipe from the wellbore. In general, fishing operations are extremely expensive and can last for several days. Many different fishing tools are available on the market, and highly experienced people are required to perform a successful job. Knowing the exact inside diameter (ID) and the OD of all downhole equipment parts is key. Other important factors are the depth of the top of the fish and the exact dimensions of the BHA. As the string approaches the fish, the flow rate needs

to be reduced to avert vibrations. After engaging the fish, high pull forces must be applied (Adelung et al. 1991).

Figure 10 shows the general recommendations for freeing a mechanically stuck pipe depending on the sticking issue (Draskovic 2017).
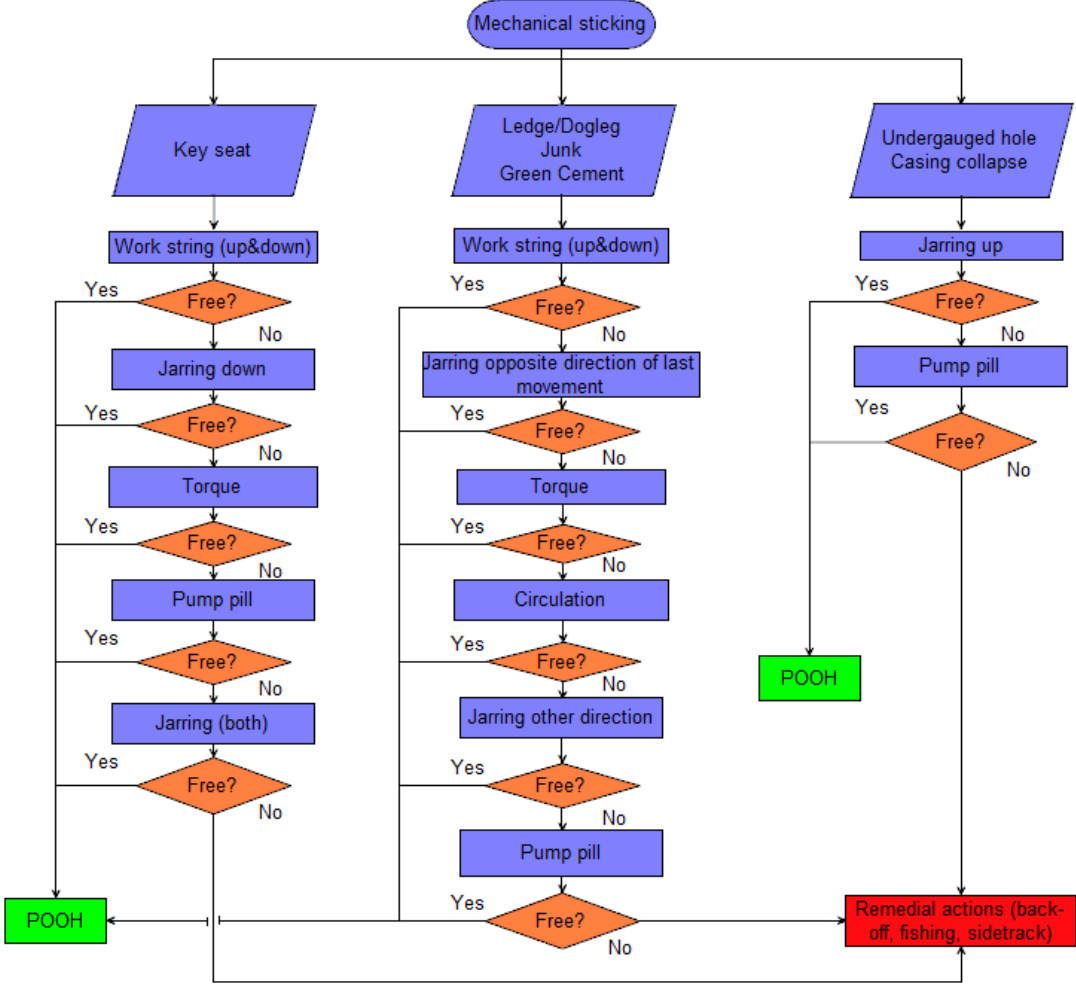

Figure 10: Freeing a mechanically stuck pipe (Draskovic 2017)

Chapter 3 State-of-the-Art Drilling Simulator

In the following chapter the state-of-the-art drilling simulator, developed by the company eDrilling AS and provided by the Department of Petroleum Engineering (DPE) Leoben, gets described. Besides an introduction with some historical and general information about eDrilling AS and drilling simulators, a simulator handbook with detailed explanations of the user interfaces and a workflow for the implementation of a general training case will be provided.

3.1 Introduction

The topic "drilling simulators" may sound to be a fairly new one, but it is not. People from the petroleum industry developed computer-controlled drilling simulators already in the early 1980s. According to (Child and Ward 1988), a drill string simulator was designed and used to assist drilling engineers during the planning, drilling and analysing phase. Written in the programming language "Pascal", the simulator was able to predict loads and stresses in drill strings and casing strings during most normal drilling operations. In addition, the graphical representation of data streams was possible, and the accuracy of the models was sufficient to be used with confidence for real drilling operations. During that time the computers have proven to be of limited capacity and costly. Most hurdles were on the electrical and mechanical side of the whole system. The soft- and hardware capabilities were limited as well (Brett and Warren 1990).

As nowadays the soft- and hardware is comparatively cheap and advanced, and the petroleum industry is moving towards automation and digitalization in big steps, it is extremely important to make use of such simulators. Next generation drilling rigs are using integrated real time digital technologies and smart designs. Drilling operations and maintenance get optimized to a maximum extent to increase the asset value (Gholami Mayani et al. 2018).

How does the usage of a drilling simulator add value to companies and their operations: (Mirhaj, Oteri, and Saelevik 2013) (Yarim et al. 2007):

- The training of drilling operations is a key contributor to minimising the severity and number of stuck pipe incidents and other wellbore issues. Therefore, it directly affects the revenue of the company
- Enables the possibility of simulating scenarios that are not possible to experience on a real drilling rig at a certain time
- Increases the efficiency of the rig personnel with training provided in a 1:1 drilling environment
- Real operational concepts get applied
- Improve the well design understanding of the personnel
- Reduce the operational risk by simulating it in advance
- Wells can be drilled safer and faster

- The utilization of a real-time simulator for a well in the South China Sea contributed significantly to the NPT reduction by 60%. A safer and cheaper drilling process was promoted as well (Nadhan, Mayani, and Rommetveit 2018)
- New drilling technologies can be easily implemented and updated for the simulator
- Forecasting, so-called "look ahead simulations", of wellbore problems and safety issues during certain operations
- Compare simulation results with real parameters to automatically detect differences

eDrilling AS, a company based in Norway, is nowadays one of the world leading supplier of drilling and well performance solutions. They collaborate closely with E&P (exploration & production) companies, operators and several service companies all over the world. The main target is to help them save costs, enhance the efficiency of drilling operations, and to increase safety by using a solution with mathematical models established, tested and verified over 20 years. All their available products are parts of their "Life Cycle Drilling Simulation" concept – complex dynamic drilling models combined with analysis and 3D (three-dimensional) visualization technology. The final solution links a transient integrated drilling simulator with a decision assistance system which supports planning, drilling, analysing and training (eDrilling AS 2019).

3.2 eDrilling AS Simulator Applications

Only the relevant and essential features for fulfilling the aim of this thesis get explained in this chapter. There would be certainly more details and applications to describe, but a common user should be able to work on the simulator with this information.

The main objective of the simulator is to represent a real reaction of the well to the actions of the drilling rig components, drill string, and the drilling mud. Moreover, drilling incidents and wellbore problems should get provided realistically. All required variables, affecting for example rig, wellbore, drill string, fluids, and geological data, will be the inputs for the simulation. The input feeds the simulator models like the fluid or T&D model. The output of these models are the required parameters, like torque or ROP, for the driller which are also provided in a real drillers cabin (Mirhaj, Oteri, and Saelevik 2013).

The simulator software/products which get used for this thesis can be divided into three main applications shown in Figure 11:

- eDrillingHub
- OpenView2D
- wellViz3D
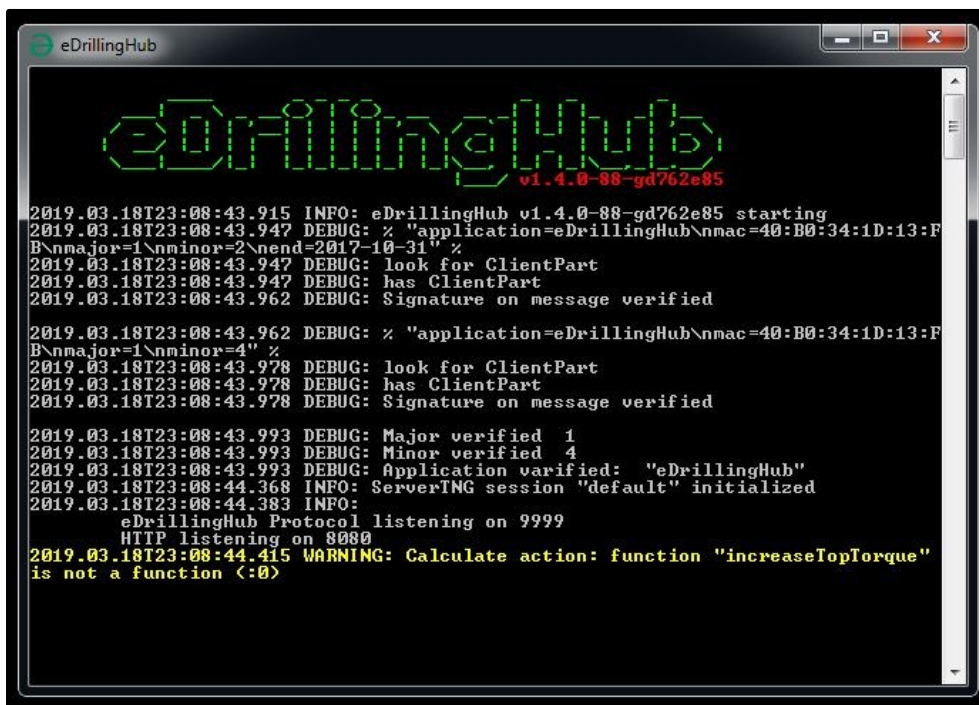
Figure 11: Icons of the three main eDrilling applications

3.2.1 eDrillingHub

The first step to start the simulator is to open the "eDrillingHub" client. As the term "hub" already explains, it is the central connection point for all other eDrilling applications. An internet connection is mandatory to be able to connect to the online server of eDrilling. Nothing can be edited or written in that console window, but important messages regarding the coding of the config directory files, see Chapter 3.2.4, get displayed.

If a message pops up in yellow, it does not mean that a file was coded wrong, but the system gives a warning that a function or variable might be unspecified. In the example of Figure 12, the function "increaseTopTorque" gets called by the sequence.json file but does not exist as a function in the calculate.js file. The system still can be run normally.

Figure 12: eDrillingHub console warning

On the other hand, if a critical message in red pops up, the simulation cannot be run. Important hints about the error type and location get displayed. For the example in Figure 13:

- "Error parsing config/sequence.json" means that the error can be found in the sequence.json file in the config directory where the "eDrillingHub" was installed
- "object is missing after a comma, at offset 4761" means that something is missing after a comma at offset 4761. The offset marks the absolute character position from the beginning of the file. It can be searched for by pressing "Ctrl + G" when using the Notepad++ editor[5].

---

[5] Free source coding editor for several programming languages

Figure 13: eDrillingHub console error

Each time a change is made in one of the coding files, the "eDrillingHub" must be restarted to reload the files and to check them again for mistakes.

After the "eDrillingHub" was run successfully, the following uniform resource locator (URL) can be opened in any web browser:

http://localhost:8080/admin

A window with the web interface of a list of all existing "tags" with their name, type of value, value, and timestamp will open. Such a tag might be called "Calculate.HklTd", which, in this case, represents a hook load value. This interface gets illustrated in Figure 14.
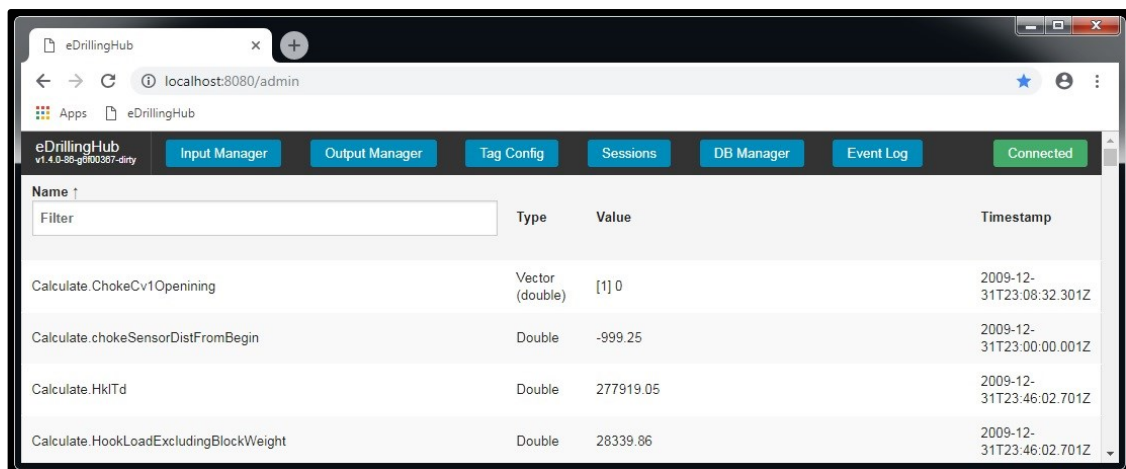
Figure 14: eDrillingHub web interface

It is mandatory that the button "Connected" is green and visible meaning the web interface is correctly connected to the server.

A tag can be right clicked to open a menu with further options to inspect the tag or to write/define the value of the tag. This can be done as well while the simulation is running.

The value "-999.25" is the internal null value of the system. It indicates that there is a lack of data/value for this parameter, but it does not mean that the parameter has the value zero.

3.2.2 OpenView2D

The next step, after running the "eDrillingHub" and the webpage, is to start the "OpenView2D"application.

It represents the real-time simulator for training of several well engineering disciplines. In its basic form, it characterises the ultimate engineering system to provide reliable feedback from the well, reacting in real time on interactive user input. This ,in addition, provides students and engineers insight and understanding of drilling operations, which else would take years of operational experience (eDrilling AS 2019).

Several other benefits are:

- Reduction of NPT by improved planning and risk management
- Move real operational problems to the simulator to explore alternatives
- Possibility of sharing the simulations within several disciplines and teams
- Understanding of well behaviour
- Downhole optimizations
- Provides solutions for drilling problems
- Interactive user input/control ("Drilling Control")
- Possibility of creating malfunctions, kicks, and losses in scenarios

After the application gets started, the following screen displayed in Figure 15, will appear.
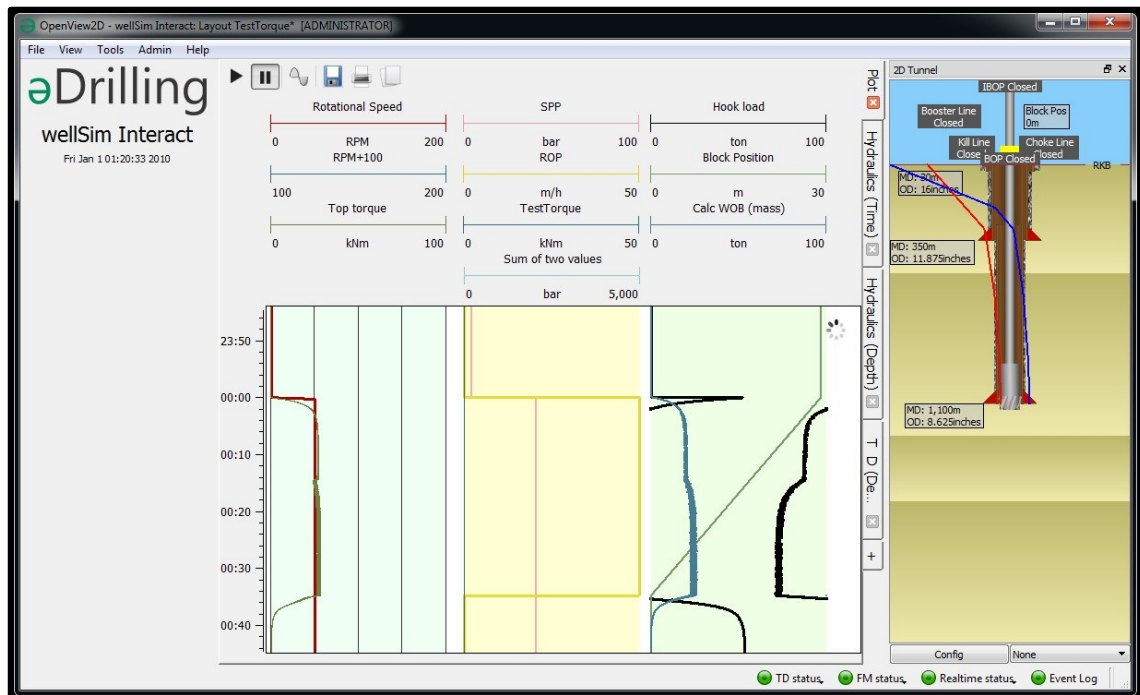
Figure 15: OpenView2D

By selecting "Tag List" within "Admin", all the available tags and their status, as in the web interface, get listed.

The four light-emitting diodes (LED) in the bottom right corner are important indicators regarding the status of models, real-time stream and events:

- TD status: Turns green if the torque and drag model is running, red if not and grey if server signals are missing. A clock gets displayed if the model is taking too long to calculate. Starting, restarting, pausing and stopping of the model is possible by clicking on it.
- FM status: Behaves like the TD status but for the fluid model.
- Realtime status: Turns green if real-time data can be received, red if not.
- Event Log: Turns green if the log is empty, red if there is at least one message. The messages get listed by clicking on the icon.

Moreover, an individual layout, depending on the parameters the driller wants to monitor, with depth- or time-based plots can get displayed in the center of the pane. By pressing the "Plot Configurator" button, the layout and all plots can be modified. This is shown in Figure 16. By clicking on the green "plus" symbol in the top left corner, new plots can be added to the layout.
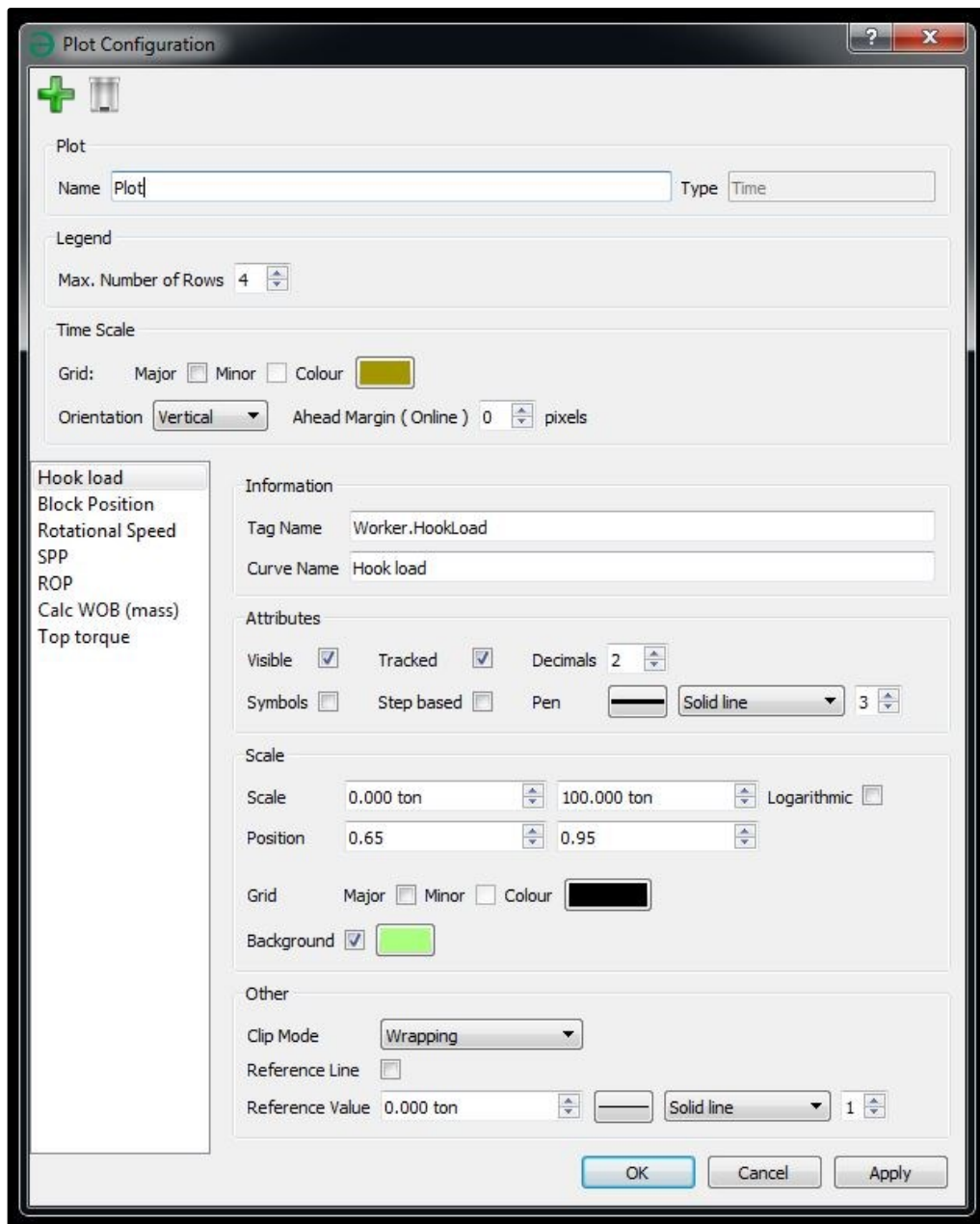
Figure 16: Plot Configurator

A two-dimensional (2D) view of the wellbore configuration can get displayed on the right-hand side of the "OpenView2D" window by clicking the "View" button and selecting "Tunnel".

Another important tool is the "Real Time Short List", shown in Figure 17, which can be selected under the tab "Tools". Important parameters which are desired to monitor during the simulation can be displayed and edited. Boundary values can be chosen and with colours highlighted, illustrated in Figure 18. The required parameters must be, as in the "Plot Configurator", specified in the "taglibrary.xml" and "taglist.xml" files.

Figure 17: Real Time Short List
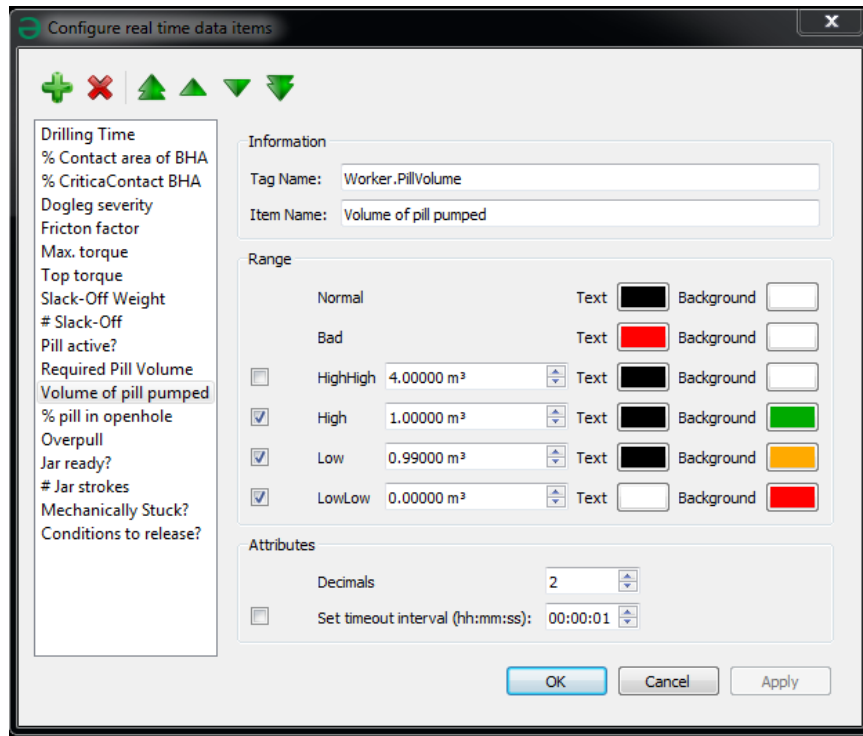
Figure 18: Real Time Short List Configurator

The next important step is to open the "Well Configuration Editor" under the tab "Tools". Different parameters, shown in Figure 19, need to be specified here. This must be done very cautiously to get the optimum output of the simulation. It is possible to simply copy and paste the data from spreadsheet applications. As this is the most time consuming and sensitive part of setting up a new case to simulate, a detailed step-by-step guide for each sub-tab of the editor can be found in Chapter 3.3.
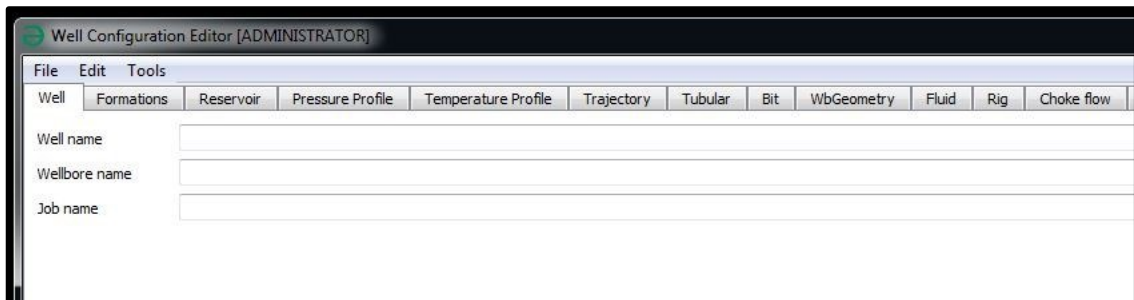
Figure 19: Well Configuration Editor

After filling in all the required data, the button "Check data quality" in "Tools" verifies the data and indicates errors. An example is illustrated in Figure 20. The "Tubular" data, more specific the weight per length of component two, marked in red, contains an error. The system also recommends a range for the expected value. Changes must be made until all the data is valid. Some logical quality checks in advance are:

- The bit OD needs to be equal to the outside diameter maximum (ODMx)
- The weight per length should be comparable to the diameter if the material has a density of steel/iron
- The ID of the tubular components needs to be smaller than the OD

- The measured depth (MD) of trajectory, pressure and temperature profile and formations needs to increase continually
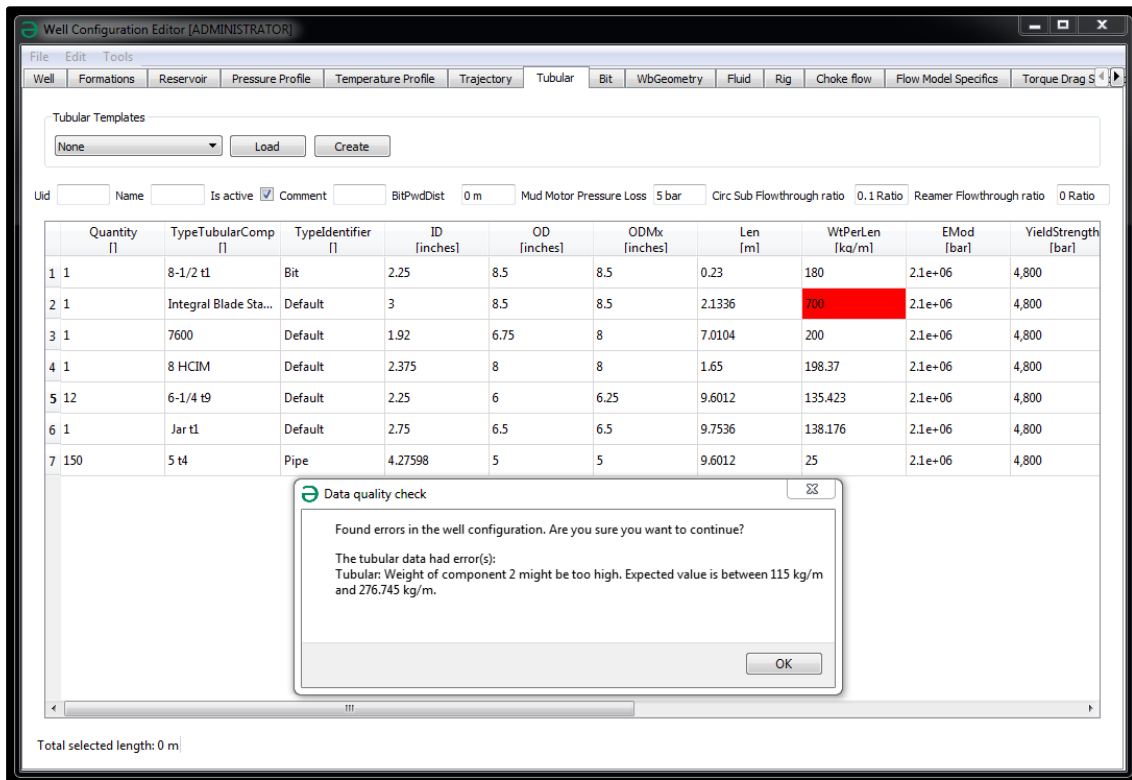
Figure 20: Data quality check

If the data has passed the quality check, it is always obligatory to press "Send to server" under "Edit" to send the current well configuration to the server and set it as active.

Another important tool, illustrated in Figure 21, is the "Drilling Control". The whole drilling process gets simulated and manipulated with it by simply varying the three main parameters – rotational speed, block speed, and pump rate. This is done manually by entering a numerical value. The other displayed parameters, like WOB, ROP or standpipe pressure (SPP), are then generated as a result.
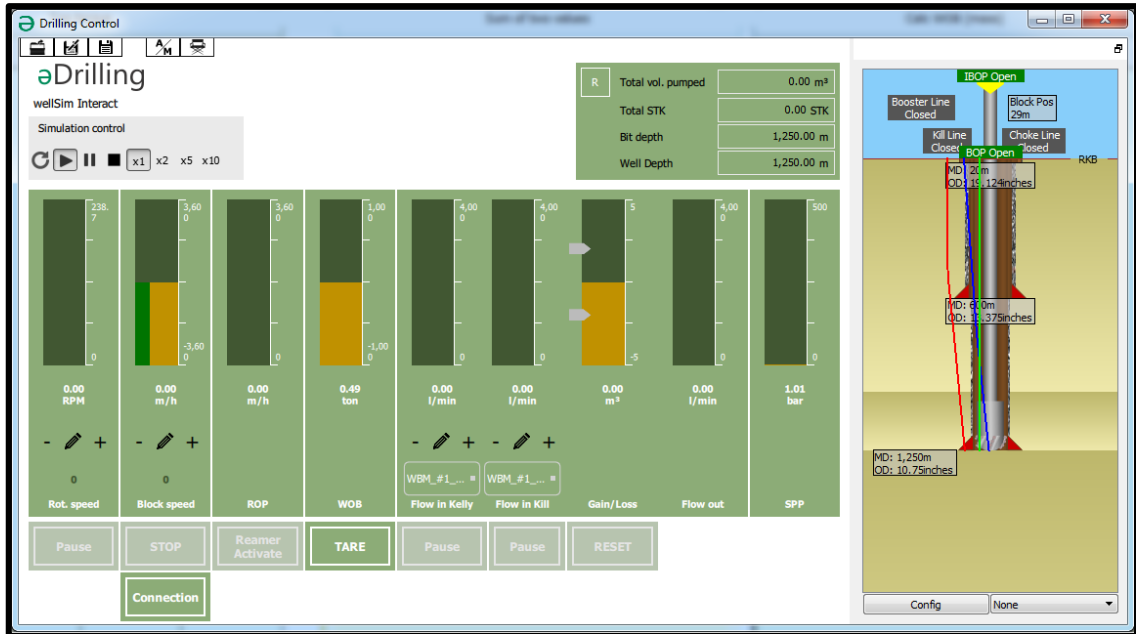
Figure 21: Drilling Control

The simulation gets operated by using the different buttons of the "Simulation control". It can be either run in real time or in time-lapse stages. By pressing the quadratic stop icon, the whole simulation gets reset to its initial conditions.

If it is necessary to add or remove a stand of drill pipes, the "Connection" button must be pressed. The whole process of adding/removing a stand will be performed automatically by the simulator itself. To add a stand, the block position needs to be lower than the full length of one stand before pressing the button. To remove one, it needs to be higher.

A detailed description of all relevant parameters in the "Drilling Control" application is listed in the following Table 3.

Parameter	Description
Rot. speed	Rotational speed measured in RPM at the rotary kelly bushing (RKB)
Block speed	Speed of the travelling block measured in m/h. A negative value represents moving upward, a positive value moving downward
ROP	Rate of penetration measured in m/h
WOB	Weight on bit measured in t
Flow in Kelly	Flow rate in, measured at the kelly in l/min
Flow in Kill	Flow rate in, measured in the kill line in l/min
Gain/Loss	Volume gained or lost measured in m^3
Flow out	Mud flow rate out measured in l/min
SPP	Standpipe pressure measured in bar
Total vol. pumped	Total volume of mud pumped measured in m^3
Total STK	Total strokes (STK) made by the pump
Bit depth	Position of the bit, with the RKB as a reference measured in m
Well Depth	Depth of the well, with the RKB as a reference measured in m
Block Pos.	Distance between the RK and the travelling block measured in m

Table 3: Description of parameters

The 2D tunnel view of the wellbore is again, similar to the "OpenView2D" window, on the right-hand side. The internal blowout preventer (IBOP), BOP, booster line, choke line and kill line can be opened and closed by simply clicking on the icon. As shown in Figure 22, they are marked green when open, grey if closed. Furthermore, the red and blue lines represent the pore and fracture pressure, the green line the dynamic equivalent circulation density (ECD).
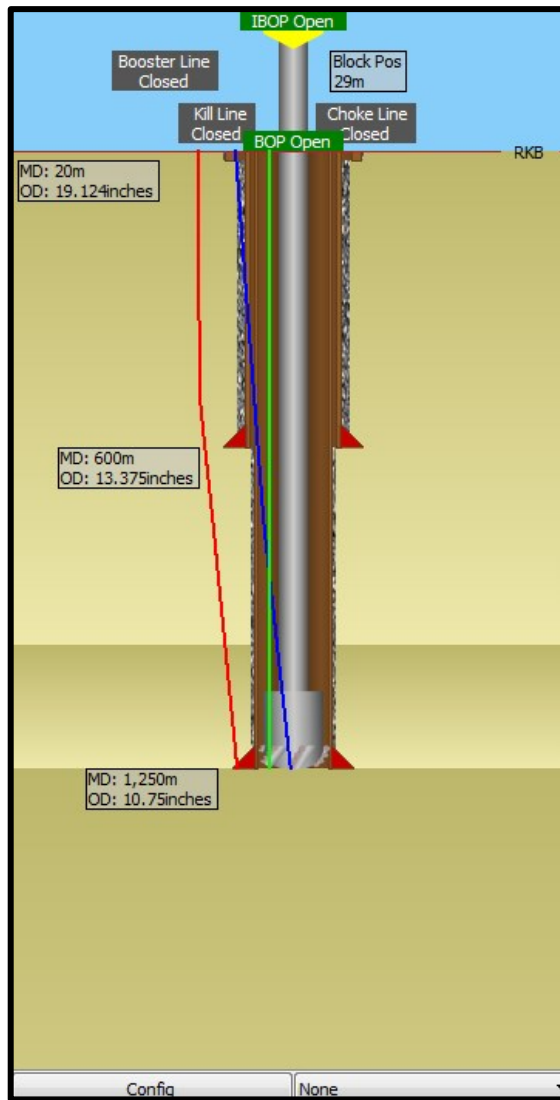
Figure 22: Tunnel view 2D

More conditions regarding the responsiveness of the simulation process can be defined by clicking on the "Scenario editor", shown in Figure 23. Besides some starting conditions or rig specifications, several pre-defined malfunctions can be actuated before a simulation.

Figure 23: Drilling Control - Scenario editor

3.2.3 wellViz3D

The last main application of the eDrilling software is the "wellViz3D". This is a 3D tool to create and display a virtual wellbore with all its components like the formations, trajectory, seismic, and tools. It is extremely useful to provide a better understanding of what is going on during the drilling operation. It reacts, like all other tools, in real time and is connected to a range of data sources of the simulation (eDrilling AS 2019).

Other features are:

- All information displayed in its context
- Three flying modes
- Free flying possibility
- Follow any BHA part
- Follow the trajectory
- Display information along the BHA or wellbore
- Visualize simulated data
- Display fluid and cuttings flow
- WOB and drilling speed indications
- Capability to create plots

The general interface after starting the application gets illustrated in Figure 24. The 2D tunnel view of the wellbore is again, as in the previous sections, on the right-hand side. The left side is the "Objects" area. Various objects and information can be selected to be displayed. It is necessary to change the "User Level" in "File" to "System Administrator" to be able to select a 3D model for each tubular component. The password for this user level stage was selected as "admin".
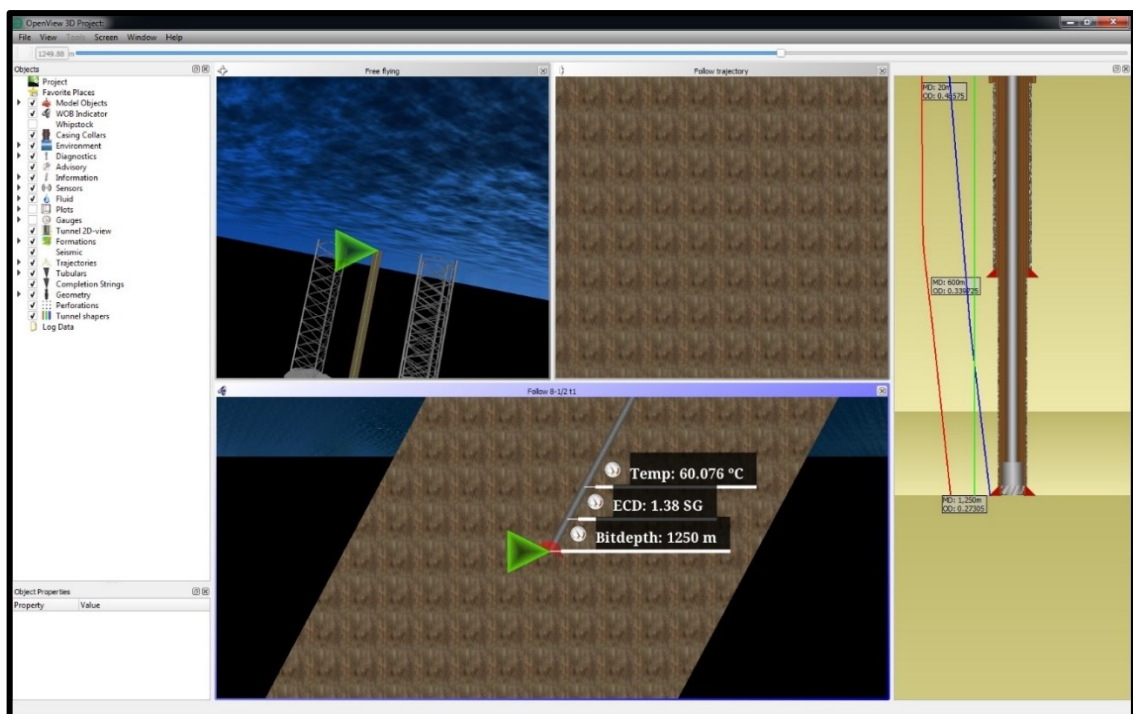
Figure 24: wellViz3D

The depiction area in the middle can be designed individually. By clicking "Window", the area can be split vertically/horizontally in up to four separate windows. It is possible to follow single objects like different parts of the BHA, shown in Figure 25, by using the "Flying mode". The red conical shape in this example represents the last casing shoe. By switching to the "Selection mode", objects can be selected individually with the mouse pointer.
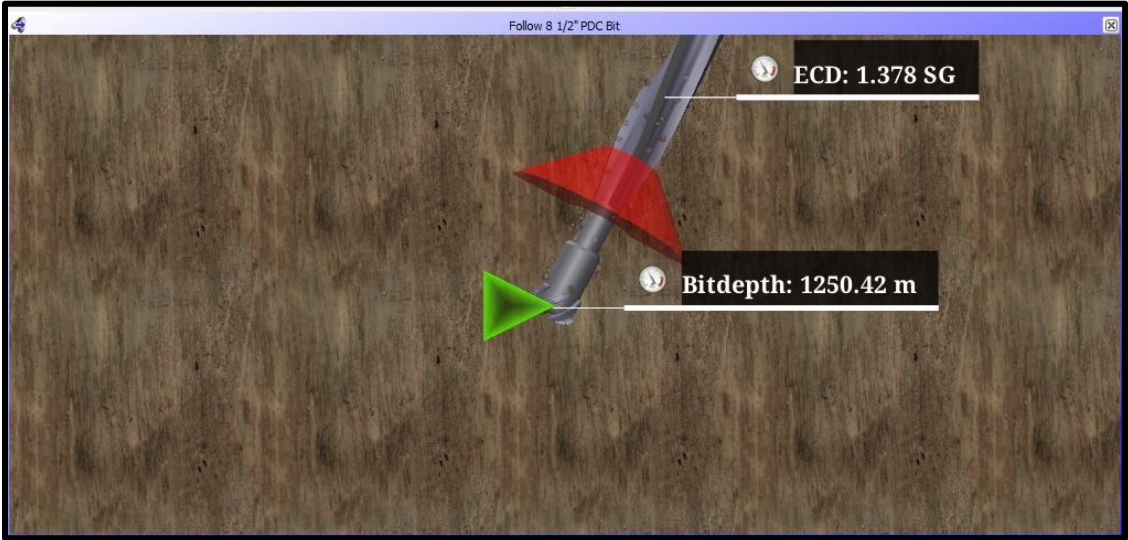
Figure 25: wellViz3D - BHA illustration

Moreover, it is possible to create depth-based plots of important drilling parameters along the wellbore.

The following keyboard shortcuts, shown in Table 4, can be useful while using the application:

Action	Description
Space	Toggle between selection and navigation mode
Ctrl+1	Flight mode
Ctrl+2	Trajectory mode
Ctrl+3	Follow drill bit mode
Ctrl+Shift+F	Fly to the selected object
Alt+Enter	Properties of a selected object
Ctrl+H	Split current 3D window horizontally
Ctrl+V	Split current 3D window vertically
Ctrl+N	Go to next well bottom hole camera position
Ctrl+M	Go to the previous well bottom hole camera position

Table 4: Keyboard shortcuts for wellViz3D

3.2.4 Files

It is crucial to know about the different files and file types used by the simulator, before even using the simulator itself. Three different file types are important to know about for a user of this handbook:

- JavaScript (JS)

JavaScript is a type of a programming language which is often used in web development. It is a client-side coding language, which means the source code gets processed by the client´s web browser instead of the web server. This means JavaScript functions are able to run after a webpage has loaded without connecting to the server (Sharpened Productions 2019).
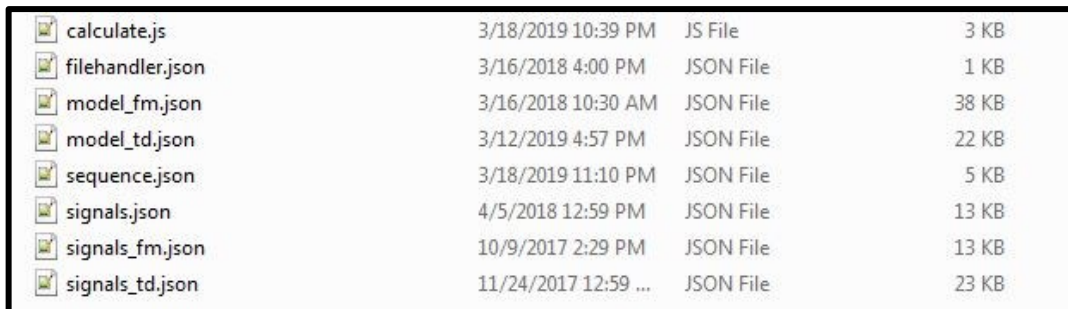
- JavaScript object notation (JSON):

This is a text-based data structure designed for processing structured data. It gets normally used to transfer data amongst web applications and web servers. JSON is often more compact than other data interchange formats (Sharpened Productions 2019).

- Extensible markup language (XML):

XML is a meta-language rather than a markup language. It can be used to create markup languages for applications. As an example, it can define items that may be accessed when loading a web page. It is possible to create a database of information without even having a real database (Sharpened Productions 2019).

The files, illustrated in Figure 26, are the main coding files in the config directory "C:\eDrilling\eDrillingHub-wellSimInteract\config" of the "eDrillingHub". A more specified guide named "eDrillingHub User Manual" from eDrilling AS is recommended to read to be able to fully understand the coding of this simulator.

calculate.js	3/18/2019 10:39 PM	JS File	3 KB
filehandler.json	3/16/2018 4:00 PM	JSON File	1 KB
model_fm.json	3/16/2018 10:30 AM	JSON File	38 KB
model_td.json	3/12/2019 4:57 PM	JSON File	22 KB
sequence.json	3/18/2019 11:10 PM	JSON File	5 KB
signals.json	4/5/2018 12:59 PM	JSON File	13 KB
signals_fm.json	10/9/2017 2:29 PM	JSON File	13 KB
signals_td.json	11/24/2017 12:59 ...	JSON File	23 KB

Figure 26: config directory files

To show how these files interact, an example code was created:

The first stage is to create a new function, demonstrated in Figure 27, in the "calculate.js" file.

```
26    // START OF TEST
27    function stuck (testTorque, maxTorque) {
28        if (testTorque>25000) {
29            maxTorque=5000;
30            return maxTorque;
31        }
32        else {
33            return maxTorque;
34        }
35    }
36    // END OF TEST
37
38    /*
39    This is an example for a
40    multi-line comment in a
41    JS file.
42    */
```

Figure 27: Test function in "calculate.js" file

A new function always gets created by typing "function". The name of the function "stuck", as well as the two input arguments in brackets "testTorque and "maxTorque", can be selected randomly. Afterwards, the main part, the if/else-loop follows and a value gets returned in the end. A single line comment in JS files is made by using "//", a multiline comment by using "/* … */".

The "sequence.json" file guides the flow, defines the input sources and chooses the files to be run. It calls the test function from the "calculate.js" file, shown in Figure 28.

```
1    {
2        "input": {
3            "id": "DC_input",
4            "type": "TAG",
5            "logging": false,
6            "authoritative": true,
7            "trigger": "SystemTime",
8            "post": [
9                {"action": "calculate", "arguments": ["Worker.testTorque", "Worker.Torque"],
10               "fn": "stuck", "resulttag": "Worker.Torque"},
```

Figure 28: Test function in "sequence.json" file

Lines one to eight already pre-exist in the file. Below, in line nine and ten, the new coding part is written. The function "stuck" from the "calculate.js" file gets called by using the term "fn". The two input arguments in brackets now need to be written in the same order as in the original function, but with their real names from the simulator. These names can easily be found by searching for them in the web interface of the "eDrillingHub", mentioned in Chapter 3.2.2. The result of the calculation gets at the end assigned to a "resulttag". No comments should be made in JSON files because the simulator marks them as an error.

Furthermore, the "sequence.json" file also configures the action to perform on the "signals.json" file. If an input argument of the function does not already pre-exist in the system, it must be created in the "signals.json" file. This gets Illustrated in Figure 29.

```
1    {
2        "Worker": {
3            "double":{
4
5
6            "testTorque":{"initial":0.0, "logging":true},
```

Figure 29: Test function in "signals.json" file

Lines one to three again are pre-existing. The missing input argument "testTorque" gets configured in line six with its initial value and logging ability. If logging is set to "true", the value will be calculated and stored for every time and depth step. This enables the user of the simulator to plot this argument over time and depth.

The "filehandler.json" file is essential to add the capability of handling files within the "eDrillingHub", but no coding is done in it.

The "signals_fm.json" and "signals_td.json" files interact with the "model_fm.json" and "model_td.json" files, respectively, but no coding was necessary in them.

After saving the changes in all these files, the "eDrillingHub" always needs to be restarted.

Changes or new implementations of plots can be made by editing the taglibrary.xml and taglist.xml files, as shown in Figure 30 and Figure 31, in both directories:

- "C:\eDrilling\eDrillingHub-wellSimInteract\config\webapp"
- "C:\eDrilling\OpenView2D-wellSimInteract\config"

A new taggroup named "MechanicalStuckPipeParameters" and the tag "TestTorque" were implemented as an example.

Single and multi-line comments in XML files are made using by "<!--…. -->".

```
1    <?xml version="1.0" encoding="UTF-8"?>
2    <taglibrary>
3    <!--
4    Start of taglist
5    -->
6    <taggroup NAME="MechanicalStuckPipeParameters">
7        <tag NAME="TestTorque" TAG="Worker.testTorque">
8            <plottypes>
9                <plottype TYPE="all" />
10           </plottypes>
11       </tag>
12   </taggroup>
13   <!--
14   End of taglist
15   -->
```

Figure 30: taglibrary.xml file

Figure 31: taglist.xml file

The unit of the value always needs to be defined in SI, and the unit system will handle the unit conversion automatically.

3.2.5 Troubleshooting / Error Messages

Common error messages which occur during the normal usage of the drilling simulator are listed in Table 5.

Error	Occurrence	Solution
Sudden SPP increase to a value > 10^{25};	While drilling, visible in "Drilling Control"; irreproducible;	Restart of the drilling simulator;
Bit depth does not change anymore, but block position does;	After pressing "Connection"; irreproducible;	Restart of the drilling simulator;
Block position sometimes changes rapidly, sometimes slowly;	During the procedure of "Connection"; irreproducible;	Restart of the drilling simulator;
Visualization error of the 2D tunnel view, often not the same in "Drilling Control" and "wellViz3D";	After starting "Drilling Control"; irreproducible;	Restart of "Drilling Control";
Crash of the simulator software;	After opening "Drilling Control"; irreproducible; When loading a well configuration while the "Drilling Control" is opened; reproducible; After making changes in a JS, JSON or XML file while the simulator is running; reproducible;	Restart of the drilling simulator;
Simulation crashes due to the fluid model;	After pressing the "play" button in the "Simulation control", often due to a bad well configuration; irreproducible;	Change the fluid model settings in the "Well Configuration Editor" and restart of the "Drilling Control";

"Drilling control" crashes;	After stuck pipe malfunction gets activated, reproducible; After the torque reached the torque limit; reproducible;	Restart of the drilling simulator;
Pre-coded malfunctions do not work as they should;	After activating a malfunction; reproducible;	-
Loading the well configuration in the "Well Configuration Editor" does not load all data;	Irreproducible;	Restart of the drilling simulator multiple times; Reloading the well configuration;
Gains/Losses & Outflow in the "Drilling Control" have random fluctuations;	While drilling; irreproducible;	-

Table 5: Error messages

3.3 Development & Implementation of a General Training Case

The following chapter will provide a workflow on how to develop and implement a general training case starting from a theoretical idea to the final simulation process. Figure 32 illustrates the main structure of the workflow.
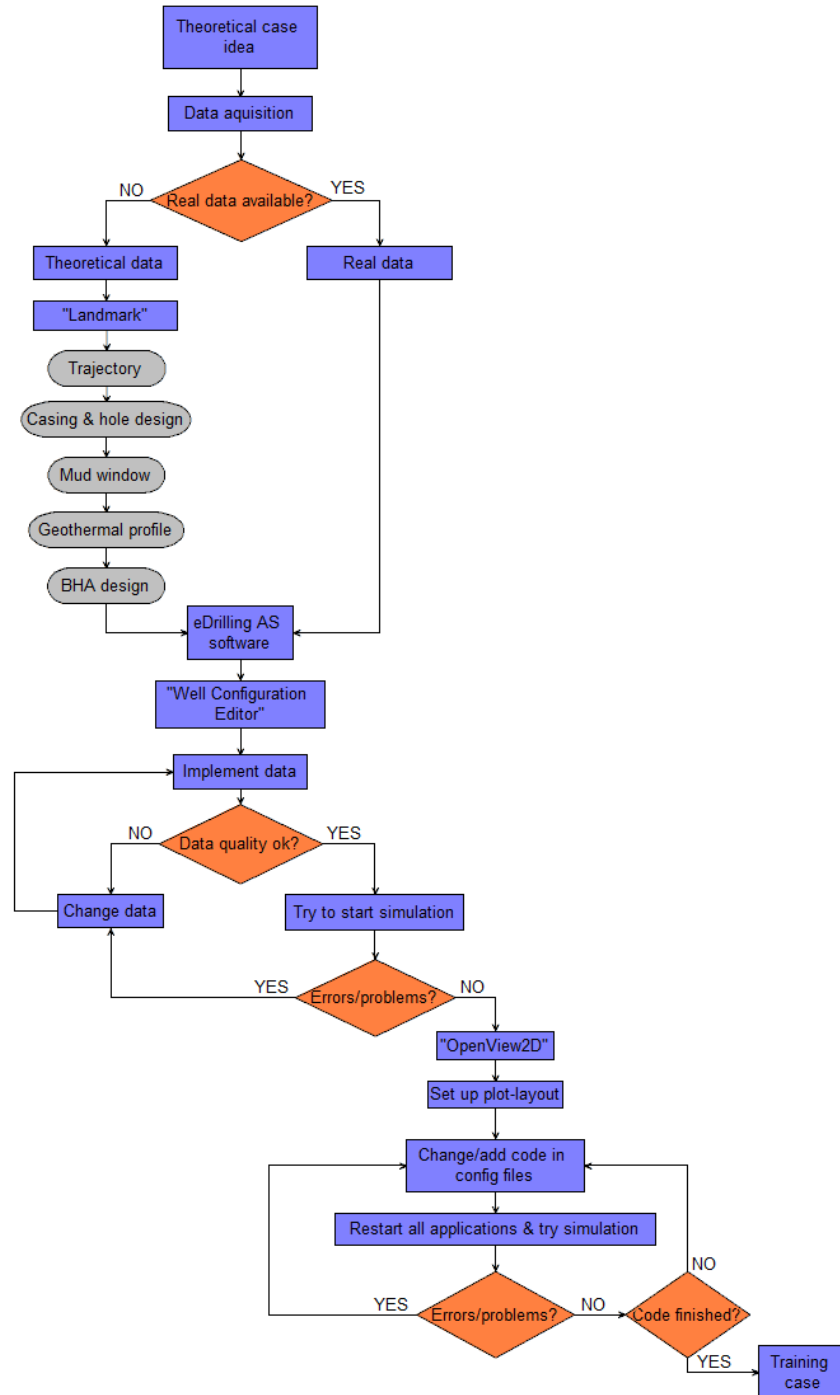
Figure 32: Workflow to implement training case

The whole development starts with a theoretical idea of a drilling-related case scenario which should be available for training purpose. Cementing, casing running, tripping, or mechanical pipe sticking are just a few examples of such scenarios. After a particular case was chosen, in this case, a drilling operation, a reliable set of data needs to be implemented into the simulator to be able to train the scenario realistically. This data can either be already available from previous wells drilled from a company, or it must be generated imaginatively in a software like "Landmark[6]".

The next step is to implement the data into the "Well Configuration Editor" of the simulator. Several parameters are already assumed in the default settings of the simulator. This editor is divided into 14 sub-categories:

Data which is not available or unknown should be set to -999.25 before the quality check.

- **Well**

The names of the well, wellbore and job need to be specified. This tab does not directly affect the simulation.

- **Formations**

All different formations in the well and their MD and true vertical depth (TVD) need to be selected and defined in this tab. It will be utilized by the models of the software and for visualization purpose. By selecting the "Type" of the formation from the drop-down menu, all other parameters like yield strength and tensile strength get filled in automatically. These default properties are changeable and can be set to the value wished by the user.

- **Reservoir**

Can be left blank in case no reservoir was identified.

- **Pressure Profile**

It is crucial to insert the pressure data in the unit SG, and the column "CollapsePres" can be set equal to the column "PorePres".

- **Temperature Profile**

If no geothermal gradient is available, it is possible to calculate it based on the formation temperature by right-clicking a cell of the "FormationTemp" column and selecting "Calculate dynamic profile". The "StringTemp" and "AnnulusTemp" can be set equal to the "FormationTemp" to satisfy the simulation.

- **Trajectory**

The trajectory data should reach all the way to the final target depth and should contain the actual and the planned trajectory. After inserting the data, ticking the box "Is active" is mandatory.

- **Tubular**

The whole string including the BHA gets selected. It is important that the tubular elements are arranged bottom-up starting with the bit. By clicking on the tab "Tools"

---

[6] Data management software of Halliburton

and "Tubular Library", some pre-defined components can be chosen. The cumulative length of all components should be long enough to reach the target depth for the simulation. The OD and ODMx of the bit must be equal.

- **Bit**

Either the total flow area of the bit or the diameter for each nozzle gets classified.

- **WbGeometry**

Every casing section, but the one which should be drilled during the simulation, must be characterized. The software assumes the open hole section to be from the last entry in the table column "MDBottom" to the current well depth. The first element needs to start from zero depth, and the last casing ID must be bigger than the bit OD.

- **Fluid**

The used mud/fluid for the desired simulation gets defined. Besides the density and the name of the new fluid, all other input parameters like the fluid components or thermal data get pre-defined by the system and can be adopted. It is vital that the density is within the borders of the pore and fracture pressure of the proposed mud window.

- **Rig**

A rig from the tab "Rig Templates" should be selected to define the important rig specification like "WightOnBlock", which describes the weight of the travelling block including the top drive, and the kelly flowline parameters.

- **Choke flow**

Nothing needs to be changed from the default settings.

- **Flow Model Specifics**

Nothing needs to be changed from the default settings.

- **Torque Drag Specifics**

Nothing needs to be changed from the default settings.

A detailed sequence of the data implementation into the eDrilling AS software gets described in Figure 33.
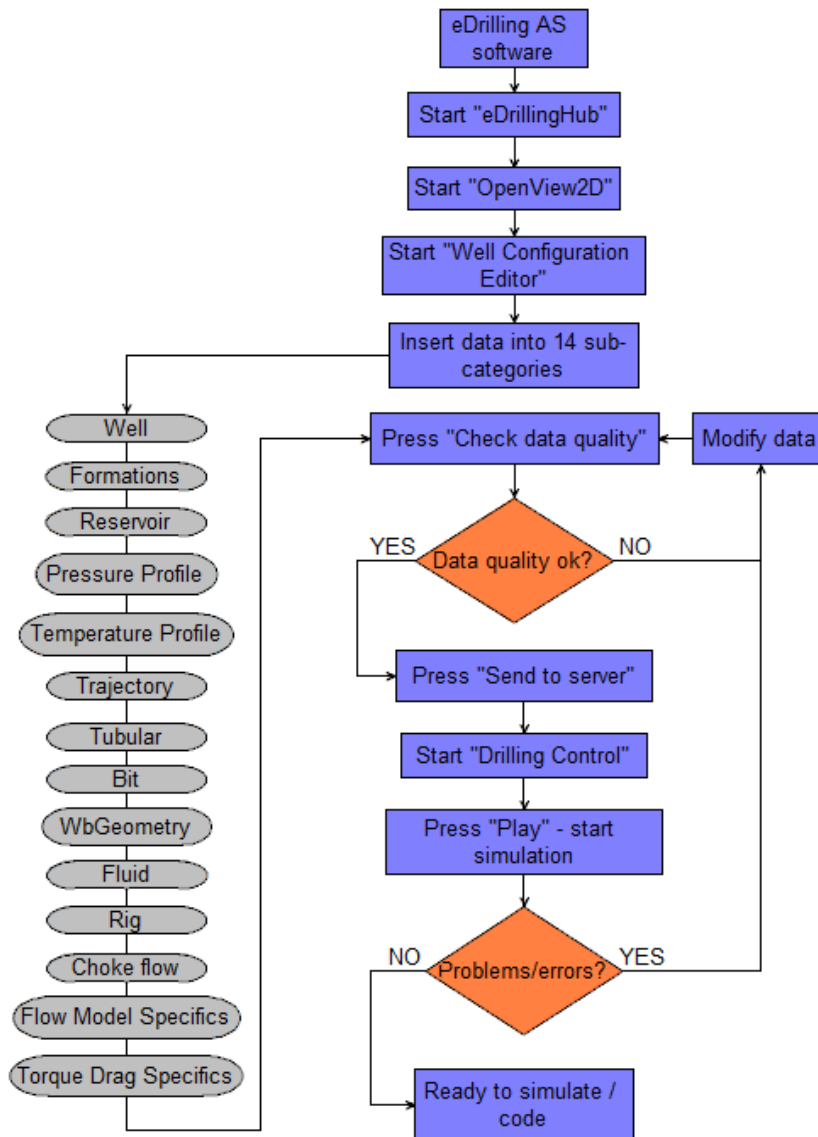
Figure 33: Sequence of data implementation

After the data has passed the quality check, the plot layout in the "OpenView2D" application can be customized, and the simulation can be tested. If the simulation launches without errors, the individual programming part of the config directory files starts. It is a trial and error process in which every single change of the code needs to be tested by restarting the whole simulator software. This step-by-step procedure gets repeated until the training case simulation performs as desired.

Chapter 4 Developed case

In the following chapter, the implemented mechanical stuck pipe training case for the eDrilling AS simulator gets described. Besides the development from a theoretical idea to the actual simulation code and the analysation of important drilling parameters related to mechanical sticking, essential parts of the coding files get explained. Furthermore, the best practice solution for this specific scenario will be presented. As the full and final simulation code was handed over in digital form the the supervisors of this thesis, more information regarding the full simulator code can be found at the Chair of Drilling Engineering of the DPE Leoben.

4.1 Case Description

After a detailed breakdown of all common types of mechanical stuck pipe events and evaluation of the drilling simulator opportunities, mechanical sticking due to a dogleg was chosen to be developed as a training case. It was found out that this type will provide the most realistic and reliable result. As it is described in Chapter 2.2, the drill string gets pushed into a dogleg which gets quantified by the dogleg severity (DLS) parameter and the contact area of the BHA with the wellbore wall increases. Consequently, the friction between the string and the wellbore wall builds up. With increasing friction, the calculated torque values reach the maximum allowable torque from the drilling rig, and as this value gets exceeded, the BHA becomes stuck, and no further rotation of the string is possible. Unrestricted circulation is still possible.

As not any parameter can be varied within the simulator software due to certain limitations from the provider, it was necessary to analyse which drilling-related tags actually influence the drilling operation and can be used to design the mechanical stuck pipe scenario. Table 6 shows the main used influenceable and working parameters including their tag name from the simulator system. Additionally, newly implemented parameters which are vital for the case are listed.

Parameter system	Tag name
RPM	DC_input.RotationalSpeed
Torque	TD.topTrqCalc
Maximum torque	Worker.Torque
Friction	TD.Static_friction
Pump rate	FM.flowIn_kellyline_rate
Block speed	DC_input.BlockSpeed
ROP	TD.tRopCalc
Bit depth	Worker.BitDepth

Initial well depth	Worker.initialWellDepth
Current well depth	Worker.WellDepth
Hook load	Worker.HookLoad
Change active fluid	FM.FluidChange
Parameter new	
DLS	Worker.DLS
Contact area of the BHA with the wellbore wall	Worker.ContactArea
Total drilling time	Worker.TotalDrillTime
Actual torque	Worker.ActualTorque
Actual ROP	Worker.ActualROP
Actual hook load	Worker.ActualHL
Reaming time	Worker.ReamTime
Reaming successful?	Worker.ReamingOK
Pumping pill successful?	Worker.PillOK
Jarring successful?	Worker.JarringOK
Slacking-off successful?	Worker.SlackingOK
Stuck?	Worker.StuckPipe
Free?	Worker.ReleaseString
Jarring action	Worker.JarStroke
Count jar strokes	Worker.NumberOfJars
Jar charged?	Worker.JarReady
Slack-off action	Worker.Slack_Move
Pill currently active?	Worker.PillCurrentlyPumped
Pill volume pumped	Worker.PillVolume
Percentage of pill in open hole	Worker.PercPillVol

Table 6: Influenceable and newly designed parameters

To design the dogleg stuck pipe situation, the following parameter tags were introduced:

- **Worker.TotalDrillTime**

The implementation of the function is shown in Figure 34. If the RPM and actual ROP exceed their limits, the drilling time starts to count. Otherwise, it stops to count and returns the last calculated value. As drilling gets continued, the time counter resumes again from where it had stopped before.

```javascript
var TimeArray =[];
var DrillTime = 0;
var ActualROP_limit = 0.0014; //corresponds to 5m/h
var RPM_limit=0.52; //corresponds to 5rpm
var TotalDrillTime=0;
function DrillingTime(ActualROP, RPM){
if (RPM>RPM_limit && ActualROP>ActualROP_limit) {
    TimeArray.push(Date.now());
    var TimeArrayLength=TimeArray.length-1;
    DeltaTime = (TimeArray[TimeArrayLength]-TimeArray[0])/1000;
    DrillTime = (DrillTime/1000) + DeltaTime;
    return DrillTime;
  }
    else {
    TimeArray=[];
    return 0.0;
        }
}
function TotalDrill(DrillTime, Corrtime){
   if(DrillTime>0) {
    TotalDrillTime=(TotalDrillTime)+(DrillTime-Corrtime)/10;
    return TotalDrillTime;
   }
   else {
    return TotalDrillTime;
   }
}
function CorrectionTime(DrillTime) {
    if(DrillTime>0) {
        var CorrTime=DrillTime-1;
        return CorrTime;
    }
    return 0.0;
}
```

Figure 34: Function DrillingTime, TotalDrill & CorrectionTime

- **Worker.DLS**

The minimum and maximum dogleg severity were chosen as, depicted in Figure 35, 0°/30m and 15°/30m. As the drilling time starts to run and the actual dogleg severity is below its maximum, the DLS increases over the next 210 seconds to 15°/30m. If the DLS has reached its maximum, it will stay at it.

```javascript
var minDLS = 0;
var maxDLS = 15;
var DLS_factor = 1;
var TimeToMaxDLS=210;
var SimulationStartTime = Date.now();
function IncreaseDLS (DLS, Time){
    if(Time>0 && DLS<maxDLS){
        DLS = minDLS+((maxDLS-minDLS)/TimeToMaxDLS*Time*DLS_factor);
        return DLS;
    }
    else if(Time>0 && DLS >= maxDLS){
        return maxDLS;
    }
    else{
        return minDLS;
    }
}
```

Figure 35: Function IncreaseDLS

- **Worker.ContactArea**

At the beginning of the simulation, the contact area is not critical and, as shown in Figure 36 and Figure 37, the area of the BHA which touches the wellbore wall can vary between zero to 45%. While the drilling time is running and the DLS increases, the contact area increases gradually to 45% where it reaches the critical limit. Only if the driller has recognised the symptoms for a possible sticking issue and performed a reaming job, or if all freeing actions after a stuck pipe situation were executed, the contact area will decrease to 17% and will not turn critical.

```
var ContactAreaCritical = false;
var CriticalArea=45;
var ContactArea=0;
function IncreaseContactArea (DLS, DrillTime){
  if (DrillTime>0 && PillOK==false && JarringOK==false && SlackingOK==true && ReamingOK==false){
    ContactArea = DLS*3;
    return ContactArea;
  }
  else if (DrillTime>0 && PillOK==false && JarringOK==false && SlackingOK==true && ReamingOK==true) {
    ContactArea = 17;
    return ContactArea;
  }
  else if (DrillTime>0 && PillOK==true && JarringOK==false && SlackingOK==true && ReamingOK==false) {
    ContactArea = DLS*3;
    return ContactArea;
  }
  else if (DrillTime>0 && PillOK==false && JarringOK==true && SlackingOK==true && ReamingOK==false) {
    ContactArea = DLS*3;
    return ContactArea;
  }
  else if (DrillTime>0 && PillOK==true && JarringOK==true && SlackingOK==true && ReamingOK==false) {
    ContactArea = 17;
    return ContactArea;
  }
  else {
    return ContactArea;
  }
}
```

Figure 36: Function IncreaseContactArea

```
function Critical_Area(Crit_Area){
  return CriticalArea;
}

function ContactIsCritical(ContactArea){
  if(ContactArea>=CriticalArea){
    ContactAreaCritical = true;
  }
  else {
    ContactAreaCritical = false;
  }
  return ContactAreaCritical;
}
```

Figure 37: Function Critical_Area & ContactIsCritical

The two main influencing parameters which must be used to get the simulator stuck, are the "TD.Static_friction" and "Worker.Torque" tags. By introducing the following functions, displayed in Figure 38 and Figure 39, the friction and torque values get modified leading to a stuck drill string. While the drilling time is running, the friction

gets increased. As the friction reaches its maximum value of ten, the maximum allowable torque value of the rig gets set to 1000, and therefore the calculated torque exceeds it. This triggers the variable "HoldStuck" to turn one. If all four requirements for the "GetStuck" function are fulfilled, the string becomes stuck.

The four requirements are:

- The friction exceeds the maximum friction value of ten
- The "HoldStuck" tag turns one
- The contact area of the BHA is critical
- The "Free" tag is stated as false, meaning no pipe freeing procedures were performed

```
function increaseSticking (DLS, DrillTime){
    if (DrillTime>0){
        var Friction = DLS * DrillTime/300;
        return Friction;
    }
    else {return 0.3;}
}
var MaxTorque = 90000;
var MinTorque=1000;
var MechanicalStuck = false;
var Free = false;
var HoldStuck = 0;

function IncreaseTorque(MechanicalStuck, Free, Friction, Torque){
        if(MechanicalStuck == true && Free == false){
            Torque = MaxTorque-Friction*50000;
        }
        else if(MechanicalStuck == true && Free == true){
            Torque = Torque + Friction*600;
        }
        if(Torque<=MinTorque){
            Torque = MinTorque;
        }
        else if(Torque > MaxTorque){
            Torque=MaxTorque;
        }
        return Torque;
}
```

Figure 38: Function increaseSticking & IncreaseTorque

```
var MaxFriction = 10;

function GetStuck (Friction){
    if(Friction>=MaxFriction && HoldStuck == 0 && ContactAreaCritical==true){
        if(Free == false){
            MechanicalStuck = true;
            HoldStuck = 1;
        }
        else {MechanicalStuck = false;}
    }
    return MechanicalStuck;
}
```

Figure 39: Function GetStuck

4.2 Best Practice

The best practice solution for the training scenario gets presented in this chapter. To assure a flawless simulation process without a software crash, reference values for the drilling operation are recommended. These values are 50rpm, 50m/h and 1000l/min for the rotational speed, block speed and pump rate respectively. One full simulation including drilling, getting stuck, performing freeing actions and starting to drill again, takes a person who is familiar with the simulator about fifteen minutes. Obviously, this duration is shortened if the correct preventive measures for the mechanical stuck pipe issue get carried out in time by the driller.

During the driller simulates the operation, he will recognise the following symptoms during drilling after a certain time on the time-based plots:

- Abnormal torque increase
- Sudden ROP decrease
- Abnormal WOB increase

These indications get displayed in Figure 40 below the horizontal red line.
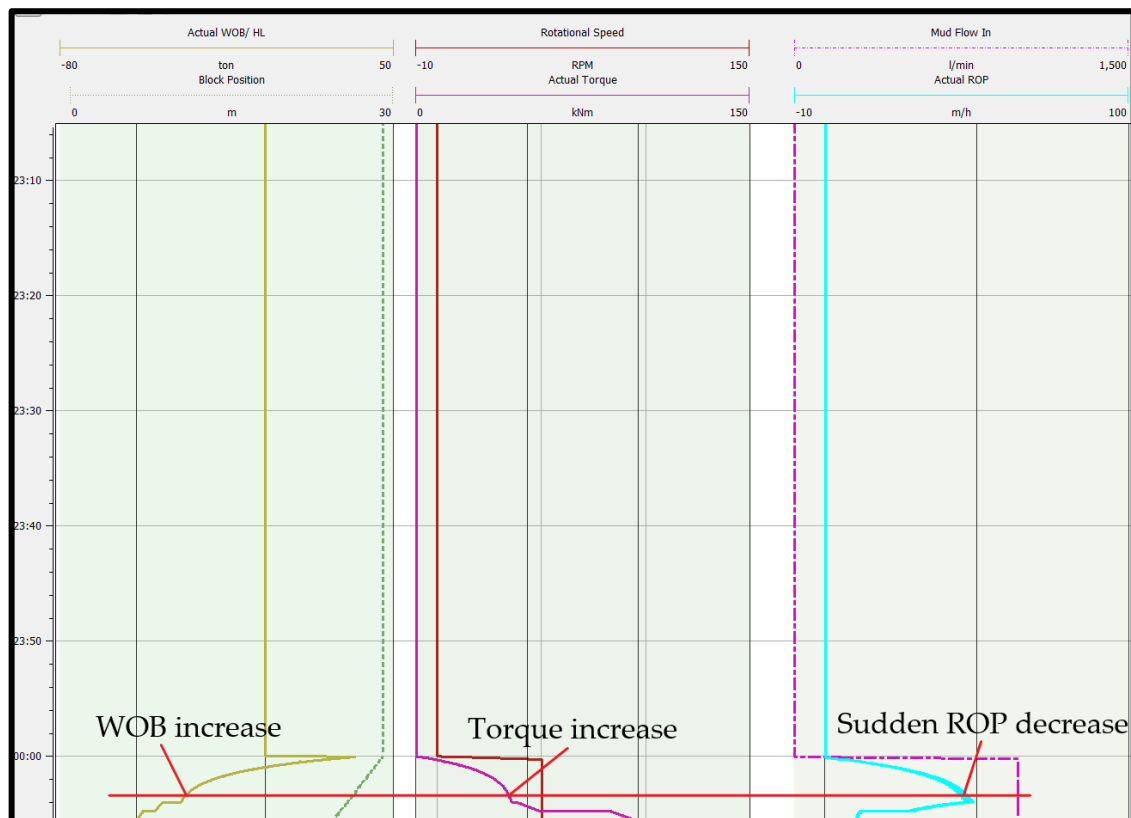
Figure 40: Symptoms on time-based plots

To prevent the drill string from sticking, the driller needs to recognise these indications quickly and perform a reaming job properly. Figure 41 shows how the reaming operation was implemented into the simulator. If the rotational speed gets increased to 120rpm or higher, the driller pulls the string upwards with 50m/h block speed and the current bit depth is smaller than the well depth, the reaming time starts to count. For a

successful reaming job, this must be continued for 30 seconds, and then the tag "ReamingOK" turns "true". Therefore, the stuck pipe situation gets prevented, and the friction and contact area get reduced. Afterwards, drilling can be continued without any issues.

```javascript
var HighRPM_limit=12.48; //corresponds to 120rpm
var BlockSpeed_limit=-0.012; //corresponds to negative Blockspeed of 50m/h
var ReamingOK=false;
var ReamTimeArray = [];
var ReamWaitTime = 30;
var ReamTime = 0;
function Ream(RPM, BlockSpeed, Bitdepth, Welldepth) {
if (RPM>=HighRPM_limit && BlockSpeed<=BlockSpeed_limit && ReamingOK == false && Bitdepth<Welldepth){
        ReamTimeArray.push(Date.now());
        var ReamTimeArrayLength=ReamTimeArray.length-1;
        ReamDeltaTime = (ReamTimeArray[ReamTimeArrayLength]-ReamTimeArray[0])/1000;
        ReamTime = (ReamTime/1000) + ReamDeltaTime;
        return ReamTime;
        }
        else {
    return ReamTime;
        }
}
function Reaming (ReamTime) {
if(ReamTime>=ReamWaitTime){
        ReamingOK = true;
        return ReamingOK;
    }
    return ReamingOK;
}
```

Figure 41: Function Ream & Reaming

If the driller could not manage to react to the symptoms in time, the drill string becomes stuck, and the RPM, torque and ROP values get zero. Immediately after the string becomes stuck, all operational parameters which can be varied within the "Drilling Control" application must be set to zero.

To get free again, the following freeing procedures must be executed:

- **Spotting a pill**

By changing the fluid and selecting the spotting pill for the primary pump, the tag "PillCurrentlyPumped" turns "true". Before ramping-up the pump, the "Total volume pumped" must be set to zero by pressing the reset button. Afterwards, the pump can be started, and the required pill volume of one m^3 be pumped downhole. To ensure that the pill has a positive impact on the stuck pipe situation, 85% of the pill volume must reach the open hole section. This demands about 14m^3 of drilling mud to be pumped after the pill. Subsequently, the tag "PillOK" turns true, and the pill got spotted successfully. The simulator code gets presented in Figure 42 & Figure 43.

```
var PillOK = false;
var PillCurrentlyPumped = false;
var VolumeOfPillPumped = 0;
var RequiredPillVolume=1; //1m3 of pill must be pumped
var RequiredPillPercentage = 85; //85% of the pill have to land in the open hole
var StringIdVolume_total =[];
var ResetVolume = false;
function PillCurPumped(FluidID){
  if (FluidID.item(0) == 1) { //The pill needs to be the second fluid in the list to have the ID 1
    PillCurrentlyPumped = true;
  }
  else {PillCurrentlyPumped = false;}
  return PillCurrentlyPumped;
}

function ReqPillVolume(ReqPillVolume){
   var x = RequiredPillVolume;
  return x;
}
function PillVolume(PillCurrentlyPumped, PumpedVolume, PillVolume){
  if(PumpedVolume<=0.2){
    ResetVolume=true;
  }
  if (PillCurrentlyPumped==true && ResetVolume==true) {
    var VolumeOfPillPumped = PumpedVolume;
    return VolumeOfPillPumped;
  }
  else {
    ResetVolume = false;
    return PillVolume;
    }
}
```

Figure 42: Function PillCurPumped, ReqPillVolume & PillVolume

```
function PercPillVol(ComponentID, ComponentLength, PumpedVolume, PillVolume){
 var StringIdVolume =[];
  for (var i=0; i<ComponentLength.length-1; i++) {
    var x=((ComponentID.item(i)*ComponentID.item(i))*3.141593/4)*ComponentLength.item(i);
    StringIdVolume.push(x);
  }
  var StringIdVolume_sum = 0;
  for (var m=0; m<StringIdVolume.length-1; m++) {
    StringIdVolume_sum=StringIdVolume_sum + StringIdVolume[m];
  }
    if(PillVolume>=0.1){
  var PercentagePill = 0;
  var VolumeInOpenHole=PumpedVolume-StringIdVolume_sum;
    if(VolumeInOpenHole>0){
      PercentagePill=VolumeInOpenHole/RequiredPillVolume*100;
     if(PercentagePill>100){
       return 100;
     }
     return PercentagePill;
    }
    else {return 0;}
  }
  else {return 0;}
}
function Pill_Sufficient(PercentagePill){
   if(PercentagePill>=RequiredPillPercentage){
      PillOK=true;
   }
 return PillOK;
}
```

Figure 43: Function PercPillVol & Pill_Sufficient

- **Jarring**

As in this case scenario, the sticking happens during the downward motion of the drill string, jarring upward is a key freeing action. Before the jarring operation can be performed, turning off the rotation, block speed and mud pumps is mandatory. Then the driller needs to pull on the string to reach the required overpull force of 4500kg. Afterwards, the force needs to be reduced below 300kg, and after 25 seconds the jar is re-charged and ready to fire again. This process must be repeated four times to result in a successful jarring job. The implementation into the simulator gets illustrated in Figure 44 & Figure 45.

```javascript
var JarringOK = false;
var JarReady = true;
var ReqJarForce = 4500;
var RequiredJarrings = 4;
var JarFired = false;
var JarCounter = 0;
var JarTimeArray = [];
var JarWaitTime = 25;
function OverPull (CurrentHookLoad, StaticHookLoad){
  var Op = 0;
    if(CurrentHookLoad>=StaticHookLoad){
      Op = CurrentHookLoad-StaticHookLoad;
    }
    return Op;
}
function FireJar (Overpull, RPM, FlowRate, MechanicalStuck){
      if (MechanicalStuck == true && RPM==0 && FlowRate.item(0)==0 && JarReady == true){
      if(Overpull>=ReqJarForce){
              JarFired = true;
              JarReady = false;
          }
      }
  return JarFired;
}
```

Figure 44: Function OverPull & FireJar

```javascript
function Jar_Counter (MechanicalStuck, Stroke, Overpull){
    var JarDeltaTime= 0;
    if(MechanicalStuck == true && Stroke == true){
        JarCounter = JarCounter + 1;
        JarFired = false;
    }
    if (Overpull<300 && Stroke == false && JarReady == false){
            JarTimeArray.push(Date.now());
            x=JarTimeArray.length-1;
            JarDeltaTime = (JarTimeArray[x]-JarTimeArray[0])/1000;
            if(JarDeltaTime>=JarWaitTime){
                JarReady = true;
            }
    }
return JarCounter;
}
function Jar_Ready (Overpull){
    var JarReadyTag = Overpull*0;
    if (JarReady==true){
        JarReadyTag = 1;
    }
    else {JarReadyTag = 0;}
    return JarReadyTag;
}
function Jarring(MechanicalStuck, NumberOfJars){
    if(MechanicalStuck == true){
        if(NumberOfJars>=RequiredJarrings){
                JarringOK=true;
        }
    }
return JarringOK;
}
```

Figure 45: Function Jar_Counter & Jar_Ready & Jarring

- **Slacking-off**

As additional downward force would push the BHA even further into the dogleg and make the stuck pipe situation worse, slacking-off is not acceptable. The limit for the maximum allowable downward force was set to 7000kg. If the driller accidentally exceeds this value once, the tag "JarringOK" turns "false" and no freeing is possible anymore. The programming code is shown in Figure 46.

```
var SlackingOK = true;
var SlackOffCounter = 0;
var ReqSlackWeight = -7000;
var RequiredSlackOff = 2;
var SlackCounter =0;
var h= true;
function ApplyWeight (CurrentHookLoad, StaticHookLoad){
  var Aw = 0;
    if(CurrentHookLoad<=StaticHookLoad){
       Aw = CurrentHookLoad-StaticHookLoad;
    }
    return Aw;
}
function SlackOffMove (RPM, AppliedWeight){
  var Int = 0;
    if (RPM<RPM_limit){
        if(AppliedWeight<ReqSlackWeight && h == true){
            SlackCounter=SlackCounter+1;
            h=false;
        }
        if(AppliedWeight >= 0){
            h=true;
        }

    }
    return SlackCounter;
}
function SlackOffSufficient(SlackOffCount){
    if(SlackOffCount>=RequiredSlackOff){
        SlackingOK=false;
    };
    return SlackingOK;
}
```

Figure 46: Function ApplyWeight, SlackOffMove & SlackOffSufficient

If the driller has made no mistakes during the simulation and successfully performed all freeing procedures, the drill string gets free again, and drilling can be continued. One full simulation process displayed on the time-based plots of the "OpenView2D" application gets shown in Figure 47.
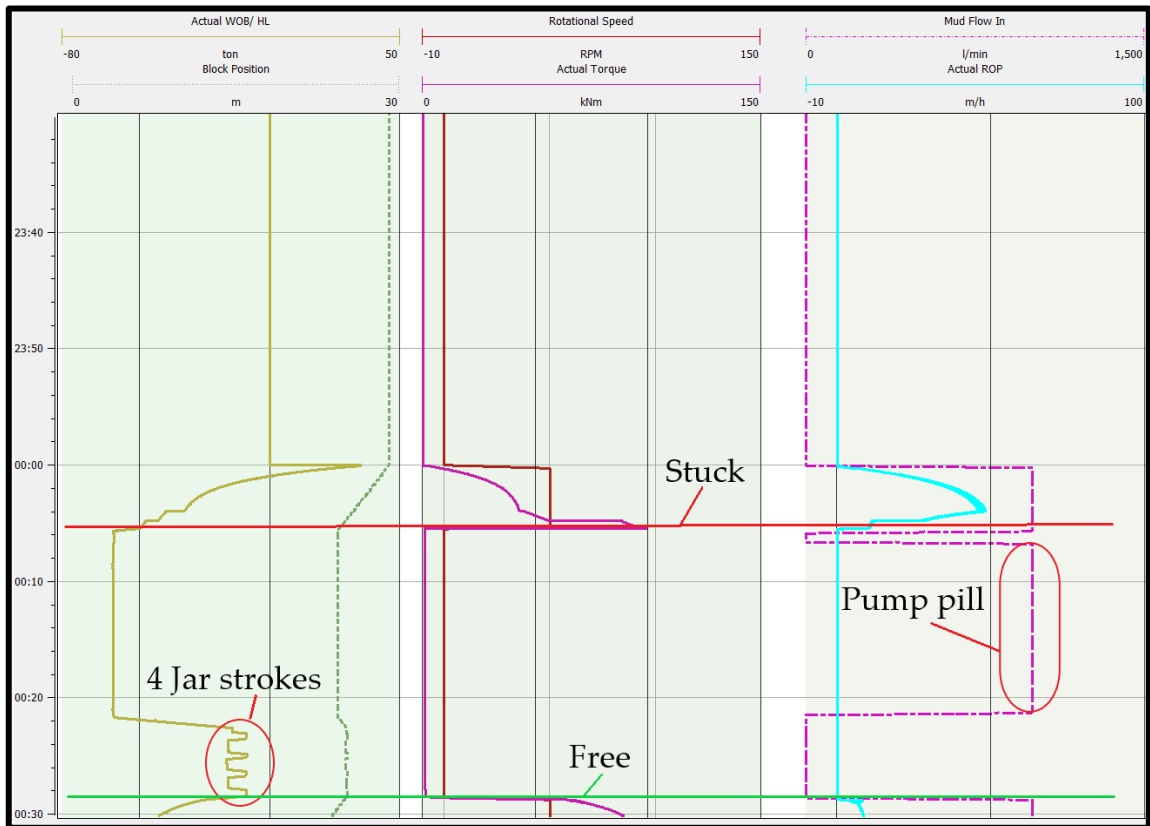
Figure 47: Time-based plots after freeing procedures

To demonstrate the best practice solution for the mechanical stuck pipe case scenario on the simulator, a detailed step-by-step workflow, shown in Figure 48, was created.
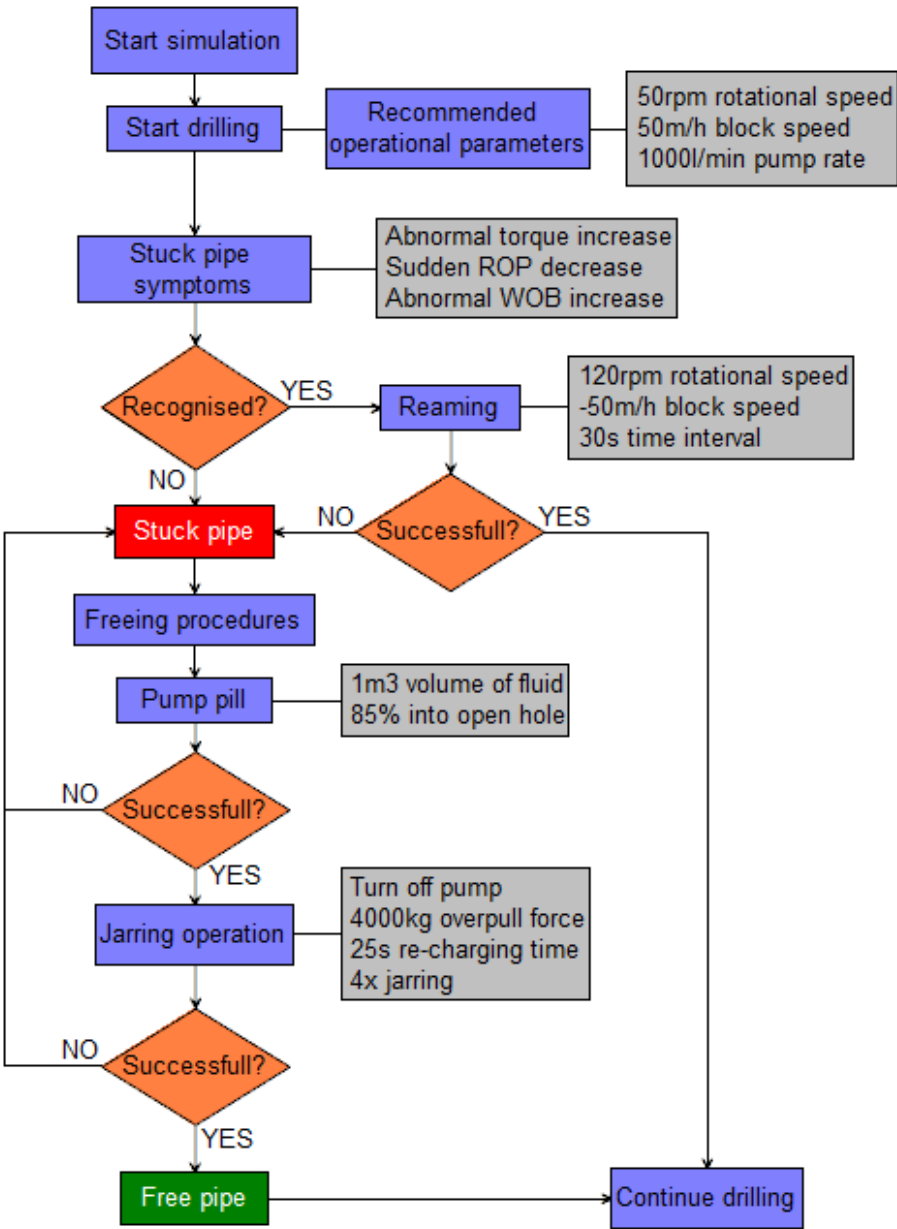
Figure 48: Workflow for best practice solution

Chapter 5 Conclusion & Future Work

The developed mechanical stuck pipe training case can effectively improve the efficiency of drilling operations by preparing operational personnel and engineers for real scenarios. By using the state-of-the-art drilling simulator in combination with knowledge of industry experienced people, the stuck pipe related NPT can be reduced. Identifying crucial warning signs in real time as well as reacting properly to stuck pipe events can be practised. Additionally, the simulator software provides a practical utilisation for university staff and students to lift their knowledge regarding digitalisation of real drilling operations.

Reliable input data, such as trajectory, BHA design and formation design is essential for a working training case scenario. The preparation of this data must be done adequately and can be assisted by the supportive application of a well design software. Furthermore, basic knowledge of coding languages such as JavaScript or XML facilitates the process of designing new training cases.

The simulator handbook is a detailed analysis of the simulator software package and its user interfaces. By applying the step-by-step guide for the development and implementation of a new training case, a user is able to create and code any desired drilling-related scenario.

There exist several best practices for avoiding stuck pipe and for counteracting the event once it occurred, but often the right solution is not performed. A best practice handbook for the designed scenario was created to solve this problem. The manual provides the necessary preventive actions and pipe freeing procedures to successfully run the simulation.

The provided training case acts as a pilot project and should be the foundation for further and more complex development of realistic drilling-related operations on the drilling simulator. An even deeper analysation of the internal simulator software models is necessary to create more extensive scenarios. What remains unsolved is the actual impact on the reaction improvement during a real stuck pipe scenario of people who practised this training case on the simulator compared to others. Therefore, carrying out a study about the impact of the training on the simulator is highly recommended as future work.

Bibliography

Adelung, Denny, Askew Warren, Jaime Bernardini, and A.T Campbell. 1991. 'Techniques for Breaking Free'.

Aldred, Walt, Dick Plumb, Ian Bradford, John Cook, and Vidhya Gholkar. 1999. 'Managing Drilling Risk'.

Aljubran, M. J., A. S. Al-Yami, and M. A. Madan. 2017. 'Best Practices to Prevent Stuck Pipe Incidents in Offshore Shaly Formations'. In *SPE Middle East Oil & Gas Show and Conference*. Manama, Kingdom of Bahrain: Society of Petroleum Engineers. https://doi.org/10.2118/183717-MS.

Alshaikh, Abrar A, Mohammed K Albassam, Salem H Al Gharbi, and Abdullah S Al-Yami. 2018. 'Detection of Stuck Pipe Early Signs and the Way Toward Automation', 15.

Bowes, Colin, and Ray Procter. 1997. 'Guidelines & Drillers Handbook Credits'.

Brett, J F, and T M Warren. 1990. 'Field Experiences With Computer-Controlled Drilling', 16.

Brydon, Kerry, Leigh Davies, Virginia Hilliard, and Peter Kank. 2015. *THE DRILLING MANUAL*. Fifth Edition.

Child, A J, and A L Ward. 1988. 'The Refinement of a Drillstring Simulator: Its Validation and Applications', 8.

Cullen, Matthew, and Richard Kucs. 2018. Project Description.

Devereux, Steve. 1998. *Practical Well Planning and Drilling Manual*. Tulsa, Okla: PennWell.

Draskovic, Ivan. 2017. 'Analytical Solution for Stuck Pipe Problems Based on Worldwide Company Practices'. Montanuniversitaet Leoben.

eDrilling AS. 2019. 'EDrilling AS Homepage'. www.edrilling.no.

Gholami Mayani, Maryam, Rolv Rommetveit, Sven Inge Oedegaard, and Morten Svendsen. 2018. 'Drilling Automated Realtime Monitoring Using Digital Twin'. In *Abu Dhabi International Petroleum Exhibition & Conference*. Abu Dhabi, UAE: Society of Petroleum Engineers. https://doi.org/10.2118/192807-MS.

Mirhaj, Seyed Ahmad, Vincent Akpojevwe Oteri, and Gunnstein Pia Saelevik. 2013. 'Tight Hole Spotting in 3D Virtual Drilling Simulator'. In *SPE Offshore Europe Oil and Gas Conference and Exhibition*. Aberdeen, UK: Society of Petroleum Engineers. https://doi.org/10.2118/166656-MS.

Mitchell, John. 2014. *Trouble-Free Drilling*. Drilbert Engineering Inc.

Muqeem, Muhammad A., Alexander E. Weekse, and Ali A. Al-Hajji. 2012. 'Stuck Pipe Best Practices - A Challenging Approach to Reducing Stuck Pipe Costs'. In *SPE Saudi Arabia Section Technical Symposium and Exhibition*. Al-Khobar, Saudi Arabia: Society of Petroleum Engineers. https://doi.org/10.2118/160845-MS.

Nadhan, Derek, Maryam Gholami Mayani, and Rolv Rommetveit. 2018. 'Drilling with Digital Twins'. In *IADC/SPE Asia Pacific Drilling Technology Conference and Exhibition*. Bangkok, Thailand: Society of Petroleum Engineers. https://doi.org/10.2118/191388-MS.

Oketch, Billy Awili. 2014. 'ANALYSIS OF STUCK PIPE INCIDENTS IN MENENGAI'.

Bibliography

Schlumberger. 2019a. 'Back Off'. *Oilfield Glossary* (blog). 2019. https://www.glossary.oilfield.slb.com/en/Terms/b/backoff.aspx.

———. 2019b. 'Keyseat'. *Oilfield Glossary* (blog). 2019. https://www.glossary.oilfield.slb.com/en/Terms/k/keyseat.aspx.

Schlumberger, and Anadrill. 1997. 'STUCK-PIPE PREVENTION SELF LEARNING COURSE'.

Sharpened Productions. 2019a. 'JavaScript Definition'. https://techterms.com/definition/javascript.

———. 2019b. 'JSON Definition'. https://techterms.com/definition/json.

———. 2019c. 'XML Definition'. https://techterms.com/definition/xml.

Yarim, G, R Uchytil, R May, A Trejo, and P Church. 2007. 'Stuck Pipe Prevention—A Proactive Solution to an Old Problem', 12.

Zende, Prasad. 2018. 'Real-Time Stuck Pipe Detection Based on Historic Data Analysis'. Montanuniversitaet Leoben.

Acronyms

BHA	Bottom hole assembly
BOP	Blowout preventer
DLS	Dogleg severity
DPE	Department of Petroleum Engineering
ECD	Equivalent circulation density
E&P	Exploration & production
FF	Frictional force
HL	Hook load
IBOP	Internal blowout preventer
ID	Inside diameter
JS	JavaScript
JSON	JavaScript object notation
LED	Light-emitting diode
MD	Measured depth
MO	Maximum overpull
NPT	Non-productive time
OD	Outside diameter
ODMx	Outside diameter maximum
PDC	Polycrystalline diamond compact
POOH	Pull out of hole
RIH	Run in hole
RKB	Rotary kelly bushing
ROP	Rate of penetration
RPM	Revolutions per minute
SF	Sticking force
SPP	Standpipe pressure
T&D	Torque and drag
TVD	True vertical depth
URL	Uniform resource locator
WOB	Weight on bit

Acronyms

WOC	Waiting on cement
XML	Extensible markup language
2D	Two-dimensional
3D	Three-dimensional

Symbols

ft	feet	[ft]
m	meter	[m]
cm	centimeter	[cm]
°	degree	[°]
min	minutes	[min]
h	hour	[h]
t	ton	[t]
kg	kilogram	[kg]
l	liters	[l]
m^3	cubic meters	[m^3]
bar	bar	[bar]
SG	specific gravity	[SG]
rpm	revolutions per minute	[rpm]

List of Figures

Figure 1: Chance of freeing a stuck pipe (Alshaikh et al. 2018) 2

Figure 2: Example of key seating (Schlumberger 2019) 4

Figure 3: Dogleg (Mitchell 2014) 6

Figure 4: Ledge (Aldred et al. 1999) 7

Figure 5: Undergauged hole (Aldred et al. 1999) 9

Figure 6: Junk in the hole (Aldred et al. 1999) 11

Figure 7: Junk retrieved after stuck pipe event (Cullen and Kucs 2018) 12

Figure 8: Collapsed casing (Aldred et al. 1999) 13

Figure 9: Green cement (Bowes and Procter 1997) 15

Figure 10: Freeing a mechanically stuck pipe (Draskovic 2017) 20

Figure 11: Icons of the three main eDrilling applications 23

Figure 12: eDrillingHub console warning 24

Figure 13: eDrillingHub console error 25

Figure 14: eDrillingHub web interface 25

Figure 15: OpenView2D 27

Figure 16: Plot Configurator 29

Figure 17: Real Time Short List 30

Figure 18: Real Time Short List Configurator 31

Figure 19: Well Configuration Editor 31

Figure 20: Data quality check 32

Figure 21: Drilling Control 33

Figure 22: Tunnel view 2D 35

Figure 23: Drilling Control - Scenario editor 36

Figure 24: wellViz3D 37

Figure 25: wellViz3D - BHA illustration 38

Figure 26: config directory files 39

Figure 27: Test function in "calculate.js" file 40

Figure 28: Test function in "sequence.json" file 40

Figure 29: Test function in "signals.json" file 41

Figure 30: taglibrary.xml file 41

Figure 31: taglist.xml file 42

Figure 32: Workflow to implement training case 45

Figure 33: Sequence of data implementation 48

Figure 34: Function DrillingTime, TotalDrill & CorrectionTime 51

Figure 35: Function IncreaseDLS 52

Figure 36: Function IncreaseContactArea 53

Figure 37: Function Critical_Area & ContactIsCritical 53

Figure 38: Function increaseSticking & IncreaseTorque 54

Figure 39: Function GetStuck 54

Figure 40: Symptoms on time-based plots 55

Figure 41: Function Ream & Reaming 56

Figure 42: Function PillCurPumped, ReqPillVolume & PillVolume 57

Figure 43: Function PercPillVol & Pill_Sufficient 57

Figure 44: Function OverPull & FireJar 58

Figure 45: Function Jar_Counter & Jar_Ready & Jarring 59

Figure 46: Function ApplyWeight, SlackOffMove & SlackOffSufficient 60

Figure 47: Time-based plots after freeing procedures 61

List of Tables