Universitetet i Stavanger

**FACULTY OF SCIENCE AND TECHNOLOGY**

# MASTER'S THESIS

| | |
|---|---|
| Study program/specialization:<br>Industrial economics/<br>Petroleum technology/Investment and finance | Spring semester, 2021 |
| Author:<br>Håvard Jordbrekk Blikra | ……………………………………………<br>(signature of author) |

Faculty supervisor: Mohammad Yahya, University of Stavanger

External supervisor: Wim Lekens, Geoprovider AS

Title of masters's thesis:

Detection of NPT and Drilling Events on the NCS from Depth-based Drilling Mechanics

Credits: 30

| | |
|---|---|
| Keywords:<br>Non-productive time (NPT)<br>Drilling events<br>Norwegian Continental Shelf (NCS)<br>Drilling mechanics<br>Petroleum technology | Number of pages: 125<br><br>+ supplemental material/other:<br>MATLAB codes p. 126-285 |

# ACKNOWLEDGEMENT

This thesis is carried out at the Faculty of Science and Technology at the University in Stavanger. The author thanks his supervisors Dr. Wim Lekens and Dr. Muhammad Yahya for their support throughout this project. The author thanks Geoprovider AS for proposing the topic for the thesis. The thesis is part of the Geoprovider RGM project, supported by Innovation Norway and industry partner Neptune Energy Norway.

The author would also like to thank his wife and family for their support throughout his studies at the University in Stavanger.

# ABSTRACT

Drilling events are subsurface related events that result in non-productive time (NPT) and increased costs. The aim of this thesis was to improve the understanding of drilling events to help mitigate non-productive time in drilling operations. NPT and drilling events related to drilling operations was studied and a consistent data analytics workflow in MATLAB was developed. An initial case study was conducted, and a prototype MATLAB workflow developed, testing different approaches for drilling events detection. Based on the result from the initial case study from the Northern North Sea, the workflow was updated and 20 wells from the North Sea and Norwegian Sea were analysed.

Mechanical specific energy and drilling strength models were calculated from depth-based drilling mechanics, and combined with the ratio between the two, the MSE/DS ratio, a workflow was created to detect drilling events. Based on cut-off values from the initial case study, the developed workflow enabled the detection of bit balling, vibrations, and bit wear. Unconfined compressive strength (UCS) was estimated from Sonic data and lithology, enabling the detection of hard rock.

Two methods were developed to estimate lost drilling time due to drilling events: The WOB-method and the Torque-method. Utilizing the linear relationship between ROP-WOB and Torque-WOB for efficient drilling, efficient bit-lines were created, enabling the estimation of lost drilling time due to inefficient drilling. By comparing bit changes to detected drilling events, lost time due to unplanned tripping was estimated.

The result from the initial case study showed a strong match between the reported events and the developed workflow. Among the 20 analysed wells, vibrations were the most common detected drilling event, with an average 41 m/km detected pr well. The largest contributor to NPT was bit balling, contributing to 44% of the lost time. On average, 69 hours were lost due to drilling events, equivalent to 5% of the total time for the drilling operations, with an estimated cost of 7 mill NOK.

The main advantage of the developed workflow is the ability to discover and map drilling events in offset wells from limited available data in an effectively manner, helping mitigate NPT in future wells.

# NOMENCLATURE

| | |
|---|---|
| $A_B$: | Bit area [$in^2$ or $m^2$] |
| BHA: | Bottom Hole Assembly |
| CCS: | Confined Compressive Strength [Psi or MPa] |
| DOC: | Depth of Cut [in or m] |
| DS: | Drilling Strength [Psi or MPa] |
| ECD: | Equivalent Circulating Density [$g/cm^3$] |
| $E_f$: | Efficiency factor [dimensionless] |
| FIT: | Formation Integrity Test |
| HSE: | Health, Safety and Environment |
| LOT: | Leak-Off Test |
| MSE: | Mechanical Specific Energy [Psi or MPa] |
| MSE/DS: | Ratio of Mechanical Specific Energy and Drilling Strength [dimensionless] |
| MW: | Mud Weight [$g/cm^3$] |
| NCS: | Norwegian Continental Shelf |
| NPT: | Non-Productive Time |
| NPV: | Net Present Value |
| OBM: | Oil-based Mud |
| R: | Bit radius [in or m] |
| ROP: | Rate of Penetration [m/hr or ft/hr] |
| RPM: | Revolution per minute [rpm] |
| SPPA: | Stand-Pipe Pressure Average [bar] |
| T: | Torque [ft-lbs or kNm] |
| TOB: | Torque on Bit [ft-lbs or kNm] |
| UCS: | Unconfined Compressive Strength [Psi or MPa] |
| WBM: | Water-based mud |
| WOB: | Weight on Bit [ton or lbs] |

# LIST OF CONTENTS

# 1 INTRODUCTION

Drilling operations constitutes a major part of the cost of exploring and producing oil and gas, especially offshore. The main cost is associated with the hire of the drilling rig which in most cases is time dependent and a function of rig-days. Drilling as efficiently as possible is therefore a high priority for any operator and partnership contracting a drilling rig.

*Non-productive time* (NPT) can be defined as unplanned breaks from drilling or very low penetration rates. Drilling events are subsurface related events that results in non-productive time. Increased NPT does not only result in delays and increased costs for the operator, but has also an impact on health and safety (HSE) performance.

NPT and drilling events are often hard to predict in advance as the challenges vary greatly from area to area. By studying offset wells and learning from drilling experience, preventative measures can be taken to limit NPT and avoid drilling events, which include bit, casing, drilling fluid, bottom hole assembly (BHA) design, among others (IADC, 2015). However, in many cases detailed information from offset wells is not readily available.

Every well on the NCS is listed on the NPD Factpages (NPD, 2021), along with information such as well depth, casing design, drilling fluid design, LOT/FITs, and a short summary of drilling objective and operation. Unfortunately, detailed descriptions of drilling events are rare.

In completion reports and final well reports, detailed descriptions of drilling events are more common, but reading and analysing these reports is time consuming, and often these reports are not available. Fortunately, drilling data and composite logs from offset wells are generally available in depth. Time-series data is another way to present well data, however these data are less commonly available and presented with less degree of standardization. Most depth data can be retrieved from the national database for the Norwegian Continental Shelf (NCS) called Diskos. According to the Petroleum Act of 1996, all raw data is to be published within two years after completion of the well and all processed and analyzed data is to be published within 20 years.

**Table 1** illustrates the most reported events on the NPD Factpages and how often they are mentioned. It is shown that major events such as losses, stuck pipe, swelling and shallow water flow are the most mentioned events. Bit balling, vibrations, low ROP and bit wear are however seldomly mentioned, making up only 14% of the mentioned events. It is also observed that from the 1807 exploration wells on the NCS, only 1277 drilling events are mentioned on the NPD Factpages. While not exact, these numbers do give an insight in the scale and diversity of drilling problems.

**Table 1. Drilling events mentioned on the NPD Factpages.**

| **Drilling Event** | **Times Mentioned** | |
|---|---|---|
| Losses | 558 | 44 % |
| Stuck pipe | 202 | 16 % |
| Swelling | 169 | 13 % |
| Shallow water flow | 163 | 13 % |
| Bit balling/Balling | 91 | 7 % |
| Vibration | 43 | 3 % |
| Low ROP | 42 | 3 % |
| Bit wear | 9 | 1 % |
| **Sum** | **1277** | **100%** |

### *1.1 Research Aims and Objectives*

Drilling events are subsurface related events that result in non-productive time and increased costs. The aim of this thesis is to improve the understanding of drilling events to help mitigate non-productive time in drilling operations. An assumption for this thesis is that by increasing the understanding of drilling events, where they occur and under what conditions, the events can be avoided in future wells. A consistent data analytics workflow in MATLAB is developed to detect drilling events from depth-based drilling data and estimate non-productive time. The drilling events in focus are the events related to drilling inefficiency: bit balling, vibrations, bit wear and hard rock. As part of Geoprovider's RGM (RealGeomechanics) project, the developed workflow lays a foundation for improved understanding of drilling data derived pore pressure predictions and real-time drilling optimization, but this is not an objective for this thesis.

The thesis has the following objectives:

1. Develop a prototype MATLAB workflow to test different approaches for drilling events detection and NPT estimations.
2. Conduct a case study from the Northern North Sea to test the prototype MATLAB workflow.
3. Update MATLAB workflow based on the results from the case study.
4. Analyse 20 wells from the North Sea and Norwegian Sea with the updated MATLAB Workflow.
5. Study the results and estimate the costs associated with drilling events and NPT.

# 2 THEORY

### *2.1 Drilling Events & NPT*

#### *2.1.1 Bit Balling*

Worldwide, 60% of all wells are drilled in claystone and shale formations. Claystone are neither hard nor abrasive and should therefore be easy to drill. However, when claystone cuttings react with water from the drilling fluid, the cuttings absorb the water, which can lead to swelling and sticking to the bit. The hydrated cuttings may stick to the spaces between the bit teeth or block the nozzles, consequently reducing the penetration depth and the flow of drilling fluid. Individual cones on a roller-cone bit may stop rotating, leading to excessive shear and bit-tooth wear (Roy & Cooper, 1993). Common issues related to bit balling include poor drilling rates and ineffective hole cleaning.

Indicators of bit balling include reduced ROP and torque. An increase in Standpipe Pressure without any changes in flow rates or drilling parameters may also indicate bit balling. Factors affecting bit balling include high WOB, poor bit design, poor drilling fluid design and low flow rate. Mitigations include increased RPM and reduced WOB to below founder (see *chaper 2.7 Drilling Mechanics and Performance*) (IADC, 2015). Other mitigations include using inhibitive water-based drilling fluids or oil-based drilling fluids, increased flow rates and optimizing the bit design. Bit design options include to select nozzles to increase hydraulic energy at the bit, increase junk slot area and utilize high-speed motors to increase ROP at WOB below founder point (IADC, 2015).

#### *2.1.2 Vibrations*

Vibrations are the most common cause of drilling inefficiency. Three different types of drill string vibrations can occur in the bottomhole assembly: axial (bit bounce), lateral (whirl) and torsional (stick slip) vibrations (IADC, 2015).

Axial vibration, or bit bounce, refers to vibrations parallel to the drill string. Axial vibrations are a common problem when drilling hard formations, and the phrase “bit bounce” refers to the bit repeatedly loosing contact with the formation. Common issues related to axial vibrations include reduced ROP and damage to bits and tools. Fluctuating WOB is an indicator of axial vibrations. Mitigations include changing the rotary speed and reduce WOB. Axial vibrations are a common problem for roller cone bits due to the working movement of the three cones. Axial vibrations for PDC bits are usually due to stick-slip (IADC, 2015).

Lateral vibrations, or whirl, refers to vibrations perpendicular to the drill strings axis. Lateral vibrations include repeated bending of the drill string components and is most common in soft to medium formations at low WOB and low RPM. Causes of lateral vibrations include high friction between the bottom hole assembly and the wellbore. Other causes include poor bit and bottom hole assembly design. Issues include over gauge hole, abnormal bit wear, low drill rates and high torque. Lateral vibrations are the most damaging vibrations for MWD tools. Mitigations include reducing RPM and increasing WOB (IADC, 2015). Lateral vibrations are a common problem for both roller cone and PDC bits.

Torsional vibrations, or stick slip, refers to vibrations around the drill string's central axis due to fluctuating torque. The phrase "stick-slip" refers to the bit slowing down, or stops rotating, and as the surface torque builds up, the bit eventually breaks free. Causes of torsional vibrations include high friction between the drill string and the wellbore, poor hole cleaning, complex wellbore trajectory and hard rock. High WOB or aggressive bit design, resulting in too high torque at the bit, is another common cause of torsional vibrations. Common issues include reduced ROP, increased bit wear and increased wear on drill string components. Torsional vibrations are a common problem for PDC bits. The shearing action of the PDC bit causes a torsional force 3-4 times higher than for a roller cone bit. Oscillations in surface torque is the main indicator of torsional vibrations. Mitigations include reduce WOB, increase RPM, attempt to reapply higher WOB (IADC, 2015).

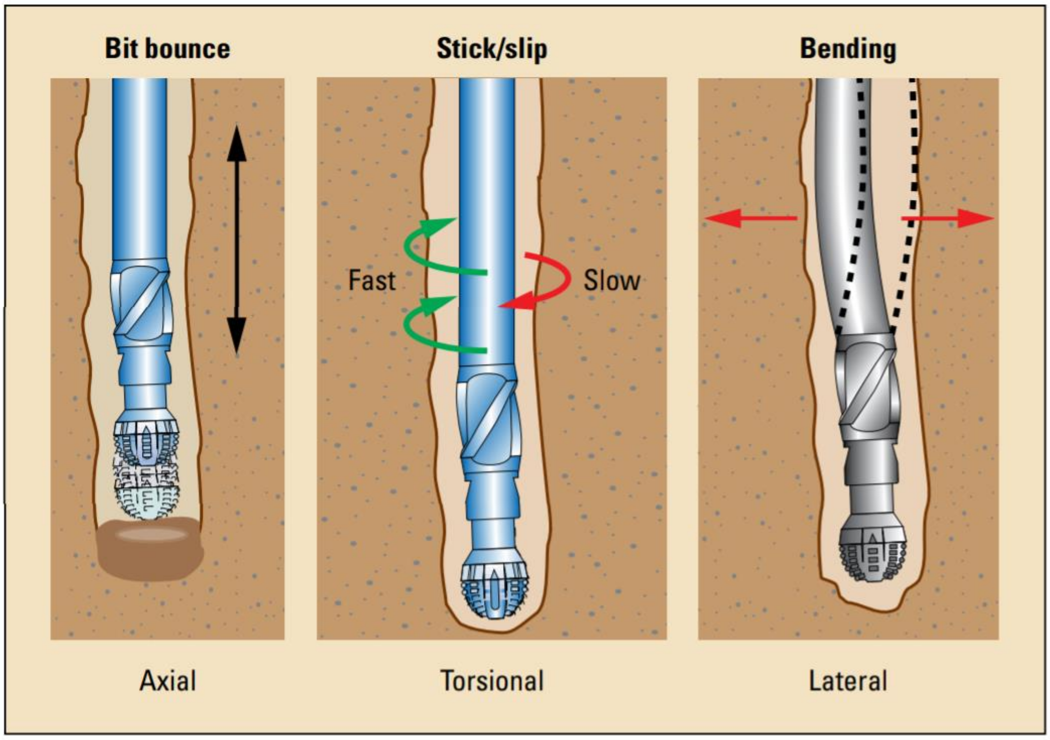


**Figure 1. Drill string vibrations (Schlumberger, 2010, p. 1).**

### *2.1.3 Bit Wear*

Bit wear is another common cause of drilling inefficiency. A worn bit requires more WOB to achieve the same ROP in a homogeneous formation. When ROP is reduced due to bit wear, it is generally advised to increase WOB before excessive RPM so the bits cutting structure can maintain a significant depth of cut to stabilize the bit and prevent whirl (IADC, 2015).

Both PDC bits and Roller cone bits are prone to bit wear, however PDC bits are more robust. A worn PDC bit becomes less aggressive, resulting in less torque for a sustained WOB. A PDC bit is exposed to both progressive bit wear with depth and sudden PDC cutter damage, the latter resulting in an instantaneous reduction in ROP. Vibrations are a common reason for bit wear. Both lateral and torsional vibrations can damage the cutters on a PDC bit (IADC, 2015). Bearing failure is the most common reason for bit failure among roller cone bits. A roller cone bit should be pulled if bearings failure is suspected, as leaving junk in the hole could lead to costly fishing jobs (IADC, 2015).

The IADC Dull bit grading system is a standard for grading the condition of a drilling bit after it has been pulled out of the borehole. There are 8 boxes characterising the bit condition and the reason for pulling the bit. By studying the pulled bits, the cause of wear can be identified. Bits are designed to wear relatively even, with slightly more wear on the outer cutters. If wear is seen on all cutters, abrasive formation, hard rock, or long run time is most likely the cause of wear. In the case of uneven bit wear, vibrations are the most likely cause of wear. By evaluating the location of the wear, the different vibrations can be differentiated, and the dominant vibration identified. Whirl is often identified by accelerated wear on the outside and no wear on the inside cutters, while stick slip is identified by accelerated wear on the inside cutters and no wear on the outside. (IADC, 2015).

## ***2.2 Unconfined Compressive Strength***

Unconfined compressive strength, UCS, is a measure of rock strength and is commonly determined through a uniaxial compressive test. In a uniaxial compressive test, the sample rock is compressed axially until it fails. The value in which the sample fails is defined as the unconfined compressive strength of the rock (Zoback, 2011). **Table 2** lists different UCS values per category of hardness.

**Table 2. Hardness and UCS Engineering Classification of Rock Materials (USDA, 2012).**

| **Hardness category** | **Typical Range UCS [MPa]** |
|---|---|
| Soil | < 0.6 |
| Very Soft Rock | 0.6 - 1.25 |
| Soft Rock | 1.25 – 5.0 |
| Moderately Soft Rock | 5 – 12.5 |
| Moderately Hard Rock | 12.5 - 50 |
| Hard Rock | 50-100 |
| Very Hard Rock | 100 - 250 |
| Extremely Hard Rock | > 250 |

As rock samples of overburden formations are rarely available for testing, numerous empirical correlations are proposed, relating rock strength to parameters available in geophysical well logs (Zoback, 2011). **Table 3** lists empirical relationships between UCS and Sonic interval velocity and common UCS values for different lithologies. An example is presented in **Figure 2**.

**Table 3. Estimations of UCS based on empirical correlations and common UCS values (Chang et al., 2006; Zhao, 2010).**

| **UCS [MPa]** | **Region Developed** | **Lithology** | **Common Values** |
|---|---|---|---|
| $1.35 * (\frac{304.8}{\Delta t})^{2.93}$ | Globally | Shale | Shale: 5-100 MPa |
| $1.4138 * 10^7 * \Delta t^{-3}$ | Gulf Coast | Weak Sand | Sand: 20-170 MPa |
| $1200 * e^{(-0.036*\Delta t)}$ | Bowen Basin Australia | Sand | |
| $\frac{(\frac{304.8}{\Delta t})^{1.82}}{145}$ | | Limestone/ Dolomites | Limestone: 30-250 MPa<br>Dolomite: 20-120 MPa |

Δt: Sonic interval velocity [μs/ft]

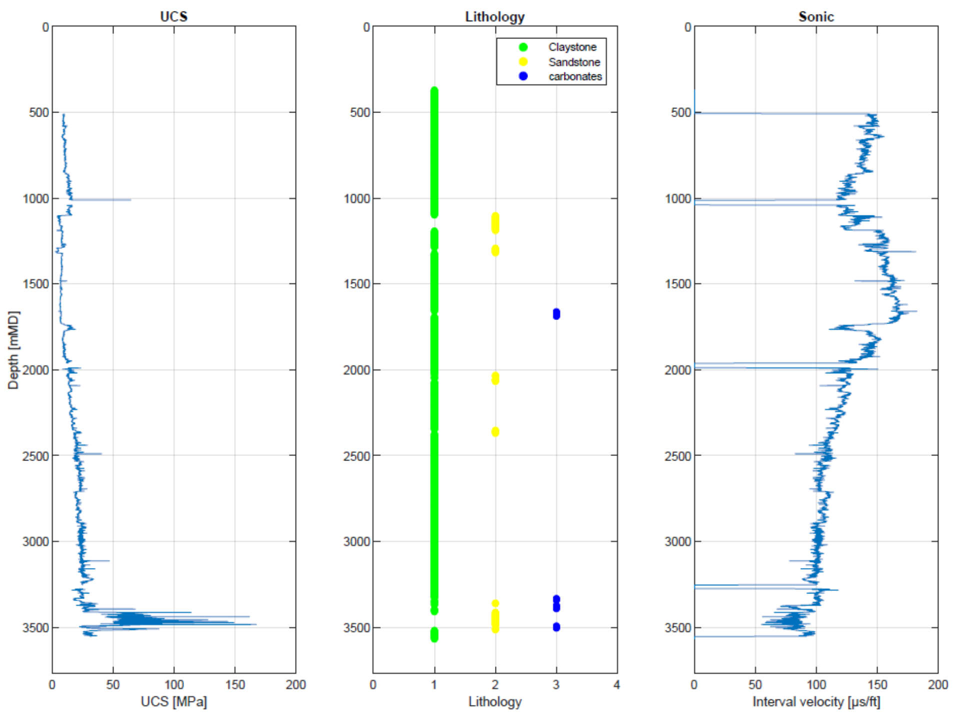


**Figure 2. Illustrating the relationship between Lithology, Sonic Interval Velocity and UCS estimations, well 34/4-15 S.**

### *2.3 Confined Compressive Strength*

Confined compressive strength, CCS, as opposed to unconfined compressive strength, includes the effect of confining pressure which affects the strength of the rock. A triaxial compression test is a common way to determine the confined compressive strength of a sample rock in the lab. The test simulates the conditions of the subsurface (Zoback, 2011). Confined compressive strength can be estimated as a function of unconfined compressive strength, differential pressure, and the angle of internal friction. CCS may be defined as:

$$CCS = UCS + D_p + 2D_p * \frac{sin\emptyset}{1 - sin\emptyset}$$

Where:

$$D_p = ECD_p - P_p$$

| | |
|---|---|
| UCS | Unconfined Compressive Strength [MPa] |
| $D_p$ | Differential Pressure [MPa] |
| $ECD_p$ | Equivalent Circulating Density [MPa] |
| $P_p$ | Pore Pressure [MPa] |
| $\emptyset$ | Angle of Internal Friction [°] |

The angle of internal friction can be estimated from (Lal, 1999):

$$\emptyset = \sin^{-1}\left(\frac{V_p - 1}{V_p + 1}\right)$$

Where:
$V_p$ P-wave velocity [km/s]

**Figure 3** illustrates the effect of wellbore pressures on the confined compressive strength. When the difference between the predicted pore pressure and the equivalent circulating density increases, the differential pressure is increased, thus increasing the CCS relative to the UCS.

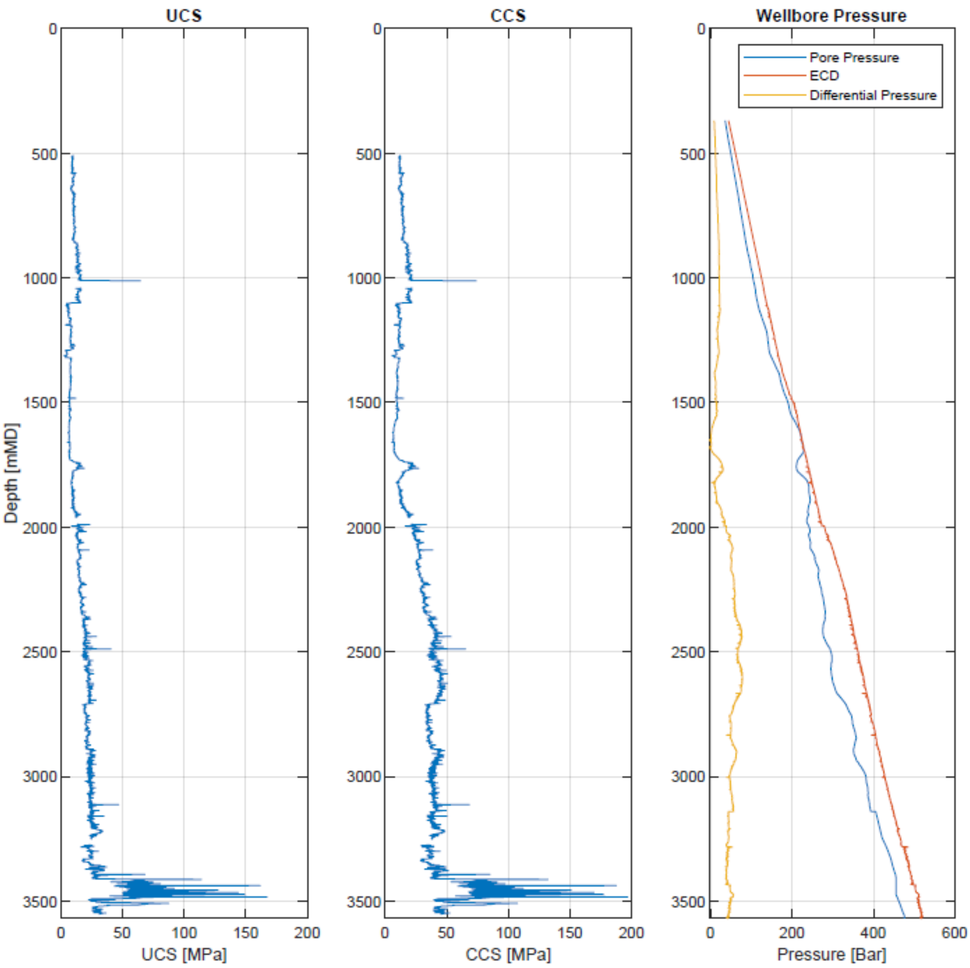


**Figure 3. Illustrating the effect of differential pressure on CCS, well 34/4-15 S.**

### *2.4 Mechanical Specific Energy*

Mechanical specific energy, introduced by Taele in 1965 (Teale, 1965), is defined as the work or energy being used per volume of rock drilled. Taele defined the equation for mechanical specific energy as:

$$MSE = \frac{WOB}{A_B} + \frac{120\pi * RPM * T}{A_B * ROP}$$

Where:

| | |
|---|---|
| MSE | Mechanical specific energy [psi] |
| $A_B$ | Bit area [in$^2$] |
| RPM | Bit rotation speed [rpm] |
| T | Torque [ft-lbs] |
| ROP | Rate of penetration [ft/hr] |

Taele's laboratory experiment showed that a perfectly efficient bit returns a MSE value numerically close to the unconfined compressive strength of the rock. However, the research was conducted under atmospheric conditions. Under borehole conditions, at maximum drilling efficiency, the MSE value is closer to the confined compressive strength of the rock.

The MSE model is composed of two parts: the weight on bit part and the rotary speed part (**Figure 5**). By definition, an increase in MSE value means more energy is required to remove a volume of rock. This may be due to a change in rock hardness, or due to drilling problems such as bit balling, vibrations or bit wear (**Figure 4**). Changes in rock hardness are less important when compared to changes in bit dysfunctions (IADC, 2015).

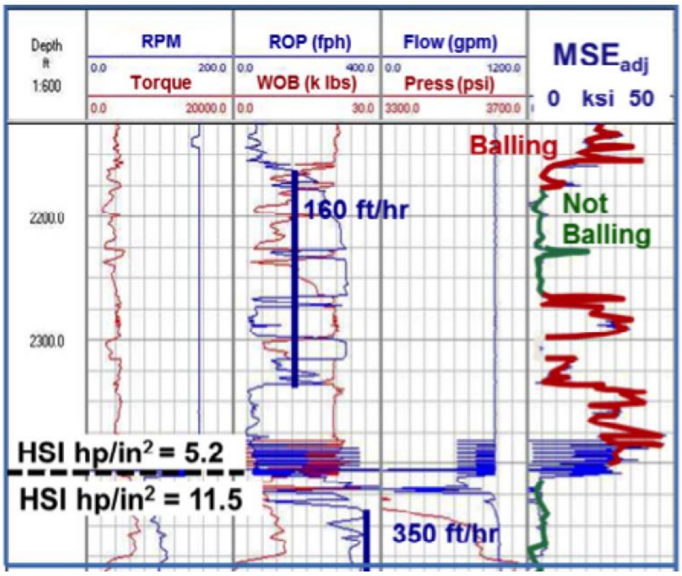


**Figure 4. Example of changes in Mechanical Specific Energy due to Bit Balling (IADC, 2015).**

On average, the bit consumes only 30-40% of the energy during drilling as a large amount of the energy is lost from the surface down to the bit (Pessier & Fear, 1992). The main source of energy loss is due to frictional forces, as the drill string tends to lay on the lower side of the borehole in deviated wells (Alsubaih & Albadran, 2018). Therefore, Dubriest et al. (2005) introduced an efficiency factor to the MSE model. Dupriest proposed an efficiency factor of 0.35 (Dupriest et al. 2005). Later efficiency factors of 0.26-0.64 and 0.125 were proposed (Hammoutene, 2012; Amadi & Iyalla, 2012). Due to the large number of factors influencing the bit efficiency, Hammoutene (2012) argued that the efficiency factor should be between 0.26 and 0.64. Amadi and Iyalla (2012) suggested a bit efficiency factor of 0.125 for directional and horizontal drilling.

$$MSE = E_f * \left(\frac{WOB}{A_B} + \frac{120\pi * RPM * T}{A_B * ROP}\right)$$

**Table 4. Efficiency factors Mechanical Specific Energy.**

| **Author** | **Efficiency factor, $E_f$** |
|---|---|
| Dupriest et al. 2005 | 0.35 |
| Hammoutene, 2012 | 0.26 - 0.64 |
| Amadi & Iyalla, 2012 | 0.125 |

In field practice, the MSE model is primarily used as a relative indicator (IADC, 2015). According to Guerrero (2007), operators commonly set the efficiency factor to 0.35. They argue that even though the value may not be completely accurate, the curve can still be used effectively as a trending tool.

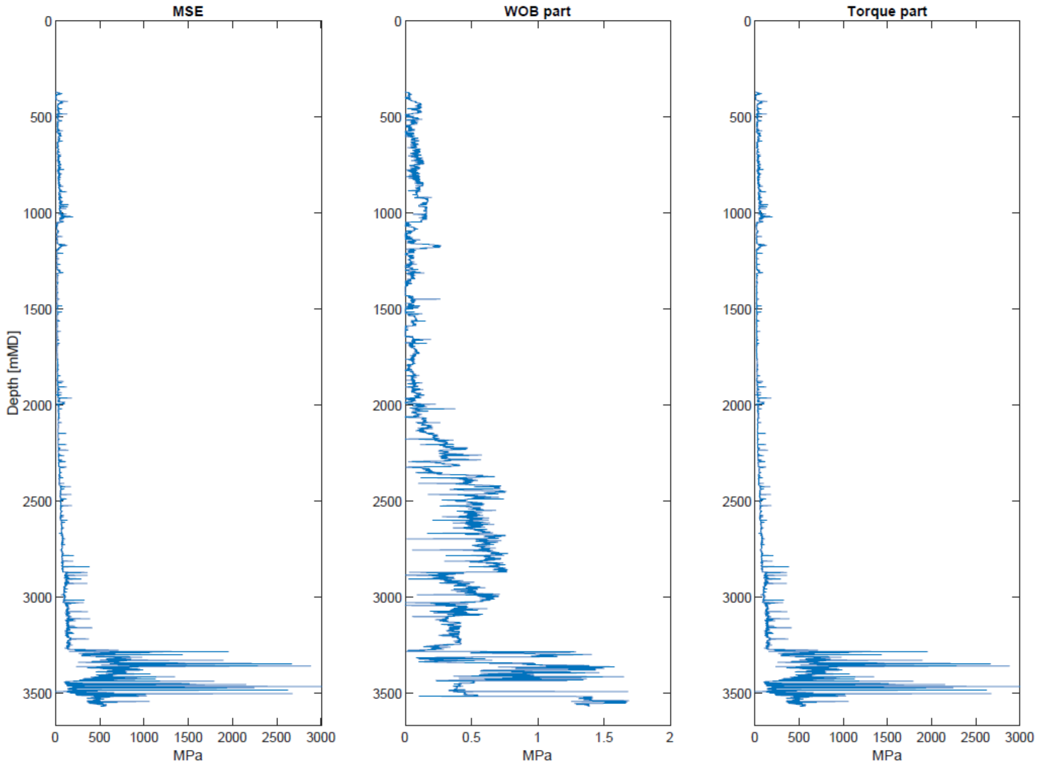


**Figure 5. Illustrating the effect of WOB and Torque on MSE, well 34/4-15 S. Note the minor influence of the WOB part compared to the Torque part on MSE.**

### *2.5 Drilling Strength*

Drilling strength is useful for analysing the contribution of WOB to the drilling efficiency since it presents a normalized pressure exercised by the drill bit on the rock. Drilling strength can be defined as:

$$DS = \frac{WOB}{R * DOC}$$

Where:

| | |
|---|---|
| DS | Drilling Strength [psi] |
| WOB | Weight on Bit [lbs] |
| DOC | Depth of Cut [in] |
| R | Bit radius [in] |

Depth of cut is defined as:

$$DOC = \frac{ROP}{RPM * 5}$$

Where:

| | |
|---|---|
| DOC | Depth of Cut [in] |
| ROP | Rate of Penetration [ft/hr] |
| RPM | Revolutions per minute |

Rearranging gives:

$$DS = \frac{WOB * RPM * 5}{R * ROP}$$

***2.6 MSE/DS ratio***

The ratio between mechanical specific energy and drilling strength is another method for detecting drilling problems. By comparing the MSE/DS ratio to MSE and DS, drilling problems can be detected and differentiated. During efficient drilling, the MSE/DS ratio lies between 1 and 1.5, reflecting an optimal balance between the energy used to drill the rock and the pressure exercised on the bit to have sufficient coupling. An increase in MSE/DS ratio indicates vibrations, while a decrease indicate bit balling or bit wear (Menand & Mills, 2017).

As illustrated in Figure 5, the WOB part in the MSE formula is almost neglectable when compared to the torque part. Menand and Mills demonstrated the difference in MSE without the WOB term to be less than 1% (Menand & Mills, 2017). By simplifying the mechanical specific energy formula by neglecting the WOB-part, the MSE/DS ratio can be expressed as:

$$\frac{MSE}{DS} = \frac{\left(\frac{120\pi * RPM * T}{A_B * ROP}\right)}{\left(\frac{WOB * RPM * 5}{R * ROP}\right)}$$

Simplifying, the above equation can be expressed as:

$$\frac{MSE}{DS} = \frac{24 * T}{R * WOB}$$

Where:

| | |
|---|---|
| MSE/DS | Ratio between mechanical specific energy and drilling strength, dimensionless |
| T | Torque [ft-lbs] |
| WOB | Weight on Bit [lbs] |
| R | Bit radius [in] |

This simplified equation illustrates how an increase in torque increases the MSE/DS ratio, while an increase in WOB decreases the MSE/DS ratio.

### *2.7 Drilling Mechanics and Performance*

Drilling rate is a function of WOB and RPM. When RPM and/or WOB are increased, the ROP should increase proportionately. A proportionate increase means the drilling is efficient, and a straight line will be formed in a ROP vs WOB plot, as illustrated in **Figure 6**. This proportionate line is known as the efficient bit line. Founder point is the point on the efficient bit line where a further increase in WOB does not give a proportionate increase in ROP. In other words, if the increase is not proportionate, the drilling is inefficient. Bit aggressiveness is another factor influencing the penetration rate. A more aggressive bit has a lower angle on the cutters and will thus achieve a greater depth of cut per revolution, consequently increasing the ROP (IADC, 2015).

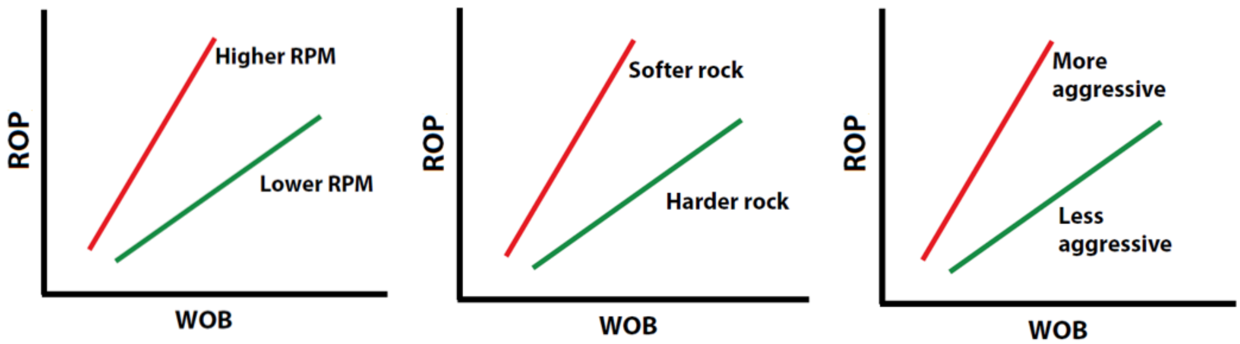


**Figure 6. Effect of WOB, RPM, rock hardness and bit aggressiveness on ROP for an efficient bit. (IADC, 2015).**

If the bit is not drilling efficiently, there is a specific reason causing the bit to drill inefficient. **Figure 7** illustrates potential causes and effects of founder. Bit balling, whirl vibrations, axial vibrations and stick-slip vibrations are the most common forms of dysfunctions. Other dysfunctions include interfacial severity: hard rock causing axial vibrations and break cutters, and bottomhole balling: differential pressure causing cuttings to be held at the bottom of the borehole (IADC, 2015).

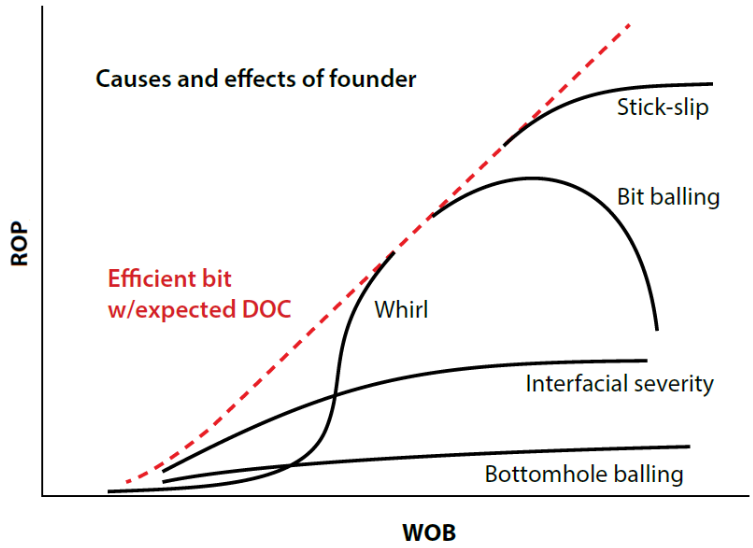


**Figure 7. Efficient bit and bit dysfunctions (IADC, 2015).**

Hamrick (Hamrick, 2011) developed a model for expressing torque as a function of WOB under the assumption that WOB, Torque and ROP are related to each other. He explained the relationship as an increase in axial force results in increased penetration rate and torsional force. Further on, he assumed a linear relationship between Torque and WOB within a normal processing range. Millan and Ringer (Millan & Ringer, 2018) developed a model for real-time bit wear estimation based on the relationship between TOB and WOB. The proposed model illustrates a perfect linear relationship between TOB and WOB for a new bit, but as the bit wears, the relationship becomes curved and less efficient (**Figure 8**).

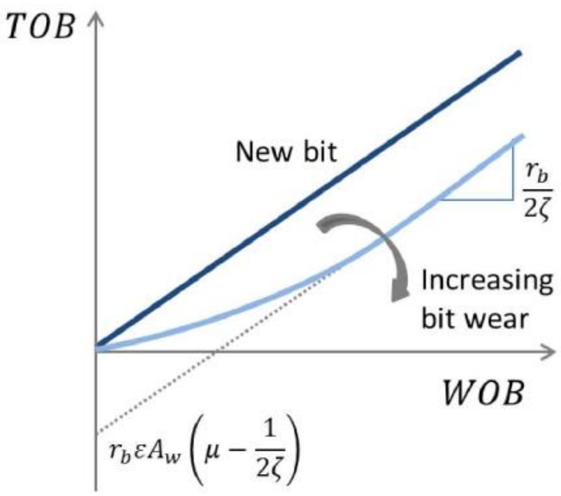


**Figure 8. Illustrating the relationship between TOB and WOB for Bit Wear (Millan & Ringer, 2018).**

### *2.8 Drilling Fluids*

Drilling fluids have several important functions for a drilling operation, including balancing the formation pressure, protect the drilled formation, cleaning the well for cuttings and transporting the cuttings to the surface. Drilling fluids furthermore lubricates and cool the drill string and drill bit and avoid pipe corrosion (e.g. Romagnoli, 2017). Drilling fluids are important for NPT as choosing the correct drilling fluid design may help mitigate drilling events such as bit balling.

Drilling fluids are commonly separated into two groups: water-based drilling fluids and oil-based drilling fluids. In water-based drilling fluids, water is the continuous phase, and additives such as barite, bentonite, polymers, alcohols and salts are added to the water-phase to achieve the required density, rheology and inhibition. Water-based drilling fluids have the advantages of being more environmentally friendly compared to their oil-based counterpart.

The continuous phase in oil-based drilling fluids is mineral oil. Crude oil and diesel have historically been used as the continuous phase, but have in later years been substituted with low aromatic mineral oil due to health, safety and environmental reasons (Young & Rabke, 2006). Additives include water, barite, bentonite and salt to achieve the desired density and rheology. Oil-based drilling fluids have the advantages of reducing the friction between the borehole and the drill string and being less reactive with the formation, thus increasing the wellbore stability, reducing formation damage and reducing the potential for differential sticking (Soliman, 1995). Another advantage is increased ROP due to the reduced wellbore friction.

# 3 METHODOLOGY AND WORKFLOW

The main objective of this thesis is to develop a consistent data analytics workflow in MATLAB analysing depth-based drilling data and composite logs to discover drilling events in offset wells. Based on the detected events, non-productive time is estimated. The workflow is developed in a MATLAB live script. Further on, the analysed wells are compared in Excel, and the costs associated with NPT are estimated. All examples presented in this chapter are from well 34/4-15 S.

An initial case study is conducted to develop a prototype MATLAB workflow and test different approaches. Based on the results from the case study, the MATLAB workflow is updated (**Figure 9**) and a total of 20 wells from the North Sea and the Norwegian Sea are analysed.

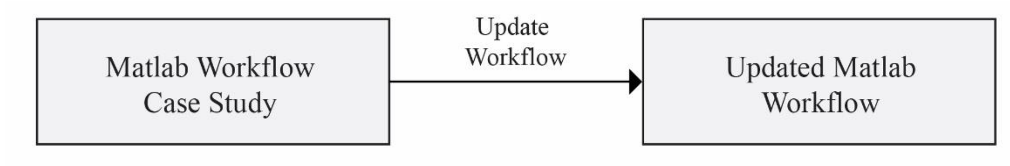


**Figure 9. Illustrating the work process in this thesis.**

The developed workflow for drilling events detection is inspired by the work by Menand and Mills (2017). The Menand and Mills method involved studying mechanical specific energy, drilling strength and the ratio between the two, MSE/DS. They proposed a method where an increase in MSE and DS and a decrease in MSE/DS indicated bit balling or bit wear, while an increase in MSE, DS and MSE/DS indicated vibrations. They also proposed UCS, or a change in rock hardness, as a reason for the change in MSE, DS and MSE/DS (**Figure 10**).

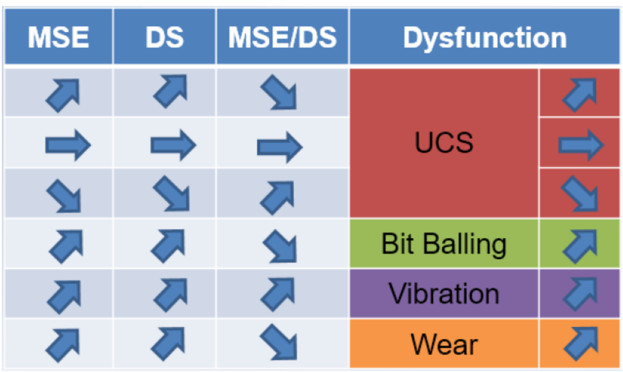


**Figure 10. MSE, DS and MSE/DS interpretation guide (Menand & Mills, 2017).**

The MATLAB workflow presented in this thesis utilizes cut-off values (**Table 5**) and lithology restrictions to allow differentiation between the drilling events. Further on, UCS estimated from Sonic transit time is used to differentiate hard rock from the other events. The cut-off values are based on the results obtained in the case study and are validated against real reported events.

**Table 5. Applied cut-off values for detecting drilling events.**

| MSE | DS | MSE/DS | UCS | Event |
|---|---|---|---|---|
| ↑ 150 MPa | ↑ 150 MPa | ↓ 1.5 | | Bit Balling |
| ↑ 150 MPa | ↑ 150 MPa | ↓ 1.5 | | Bit Wear |
| ↑ 150 MPa | ↑ 150 MPa | ↑ 1.5 | | Vibrations |
| | | | ↑ 100 MPa | Hard Rock |

The NPT estimates presented in the developed workflow utilize the relationship between ROP-WOB and Torque-WOB, presented in chapter *2.7 Drilling Mechanics and Performance*. Two methods are developed, the WOB-method and the Torque-method (**Figure 11**). Each method is presented in separate MATLAB Live Scripts. The WOB-method utilize the linear "efficient bit" relationship between ROP and WOB to estimate lost drilling time for inefficient drilling. The Torque-method utilize the "efficient bit" linear relationship between both torque and WOB as well as ROP and WOB to estimate lost drilling time for inefficient drilling. Lastly, lost time due to unplanned bit changes are estimated by comparing bit changes to the detected events. Lost drilling time and lost time due to unplanned bit changes are summed up and total lost time is estimated.

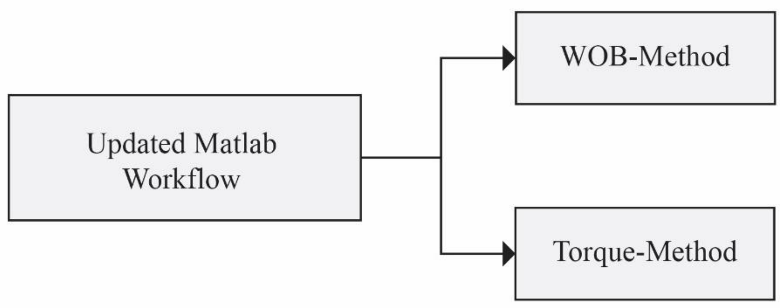


**Figure 11. The updated MATLAB workflow is separated into the WOB-Method and the Torque-Method for lost drilling time estimations.**

### *3.1 Input Data*

The developed MATLAB workflow is based on three input excel files: Drilling data, Composite data and bit change data. First, the well name for the well in question is entered and the variable *wellname* is created. The input data folder path is entered under the *path* function, along with the file names for the three excel files. Using the *xlsread* function, the variables *drilling_data*, *composite_data* and *bit_data* are created (**Figure 12**).

```
% North Sea

% 34/4-15 S
wellname = '34_4-15 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\34_4-15 S\34_4-15 S Ready');
drilling_data = xlsread('Ready_34_4-15 S  DRILLING DATA - MUDLOG.xlsx'); % Drilling data
composite_data =xlsread('Ready_34_4-15 S COMPOSITE_LOG.xlsx'); % Composite data
bit_data = xlsread('Ready_34_4-15 S  BIT_CHANGES.xlsx'); % Bit data
```

**Figure 12. Illustrating the import of input data and creation of input variables.**

The drilling data excel file is a depth-based data set listing various drilling data, generally listed for every 1-meter measured depth. The drilling data set contains: True vertical depth, measured depth, bit-size, rate of penetration, hook load, weight on bit, torque, revolutions per minute bit, stand-pipe pressure, active tank volume, mud-flow, mud density, mud temperature, equivalent circulating density, pore-pressure, lithology and drilling exponent. **Figure 13** illustrates an example excel file containing drilling data and **Figure 14** shows the creation of drilling data variables in the MATLAB workflow.

| DVER | DEPT | BDIA | ROPA | HKLA | HKLX | WOBA | TQA | TQX | RPMB | SPPA | TVA | MFOA | MFIA | MDOA | MDIA | MTOA | MTIA | ECDT | FPPG | LITH | DXC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| m | m | in | M/HR | TON | TON | TON | KNM | KNM | rpm | bar | $m^3$ | L/MN | L/MN | $g/cm^3$ | $g/cm^3$ | °C | °C | $g/cm^3$ | $g/cm^3$ | .... | DEXP |
| 370.0 | 370 | 26 | 18.23 | 0 | 0 | 0 | 2.189 | 0 | 74.86 | 21.92 | 0 | 0 | 1475.18 | 0.00 | 1.29 | 0 | 0 | 1.298 | 0 | 600 | 0 |
| 371.0 | 371 | 26 | 10.94 | 0 | 0 | 0 | 3.323 | 0 | 75.69 | 23.88 | 0 | 0 | 1476.62 | 0.00 | 1.29 | 0 | 0 | 1.298 | 0 | 600 | 0 |
| 372.0 | 372 | 26 | 3.67 | 0 | 0 | 0.436 | 2.361 | 0 | 84.72 | 20.78 | 0 | 0 | 1476.26 | 0.00 | 1.29 | 0 | 0 | 1.298 | 0 | 600 | 0 |
| 373.0 | 373 | 26 | 4.44 | 0 | 0 | 1.866 | 2.292 | 0 | 84.34 | 21.57 | 0 | 0 | 1476.25 | 0.00 | 1.29 | 0 | 0 | 1.298 | 0 | 600 | 0 |
| 374.0 | 374 | 26 | 3.60 | 0 | 0 | 2.884 | 3.081 | 0 | 83.90 | 23.40 | 0 | 0 | 1475.63 | 0.00 | 1.29 | 0 | 0 | 1.298 | 0 | 600 | 0 |
| 375.0 | 375 | 26 | 2.76 | 0 | 0 | 3.19 | 4.119 | 0 | 81.99 | 25.42 | 0 | 0 | 1476.82 | 0.00 | 1.29 | 0 | 0 | 1.298 | 0 | 600 | 0 |
| 376.0 | 376 | 26 | 3.65 | 0 | 0 | 2.602 | 3.327 | 0 | 83.66 | 23.68 | 0 | 0 | 1475.83 | 0.00 | 1.29 | 0 | 0 | 1.298 | 0 | 600 | 0 |
| 377.0 | 377 | 26 | 3.51 | 0 | 0 | 3.501 | 4.059 | 0 | 83.75 | 26.59 | 0 | 0 | 1475.42 | 0.00 | 1.29 | 0 | 0 | 1.298 | 0 | 600 | 0 |
| 378.0 | 378 | 26 | 3.97 | 0 | 0 | 3.232 | 5.022 | 0 | 85.42 | 27.88 | 0 | 0 | 1476.79 | 0.00 | 1.29 | 0 | 0 | 1.298 | 0 | 600 | 0 |
| 379.0 | 379 | 26 | 3.77 | 0 | 0 | 3.586 | 8.437 | 0 | 87.06 | 34.86 | 0 | 0 | 1471.94 | 0.00 | 1.29 | 0 | 0 | 1.298 | 0 | 600 | 0 |
| 380.0 | 380 | 26 | 3.74 | 0 | 0 | 2.395 | 3.124 | 0 | 82.83 | 24.20 | 0 | 0 | 1475.83 | 0.00 | 1.29 | 0 | 0 | 1.297 | 0 | 600 | 0 |
| 381.0 | 381 | 26 | 2.80 | 0 | 0 | 2.96 | 2.927 | 0 | 84.33 | 24.77 | 0 | 0 | 1502.49 | 0.00 | 1.29 | 0 | 0 | 1.297 | 0 | 600 | 0 |
| 382.0 | 382 | 26 | 2.20 | 0 | 0 | 4.481 | 3.273 | 0 | 83.87 | 25.16 | 0 | 0 | 1502.77 | 0.00 | 1.29 | 0 | 0 | 1.297 | 0 | 600 | 0 |
| 383.0 | 383 | 26 | 3.71 | 0 | 0 | 4.863 | 5.048 | 0 | 105.57 | 24.03 | 0 | 0 | 1501.05 | 0.00 | 1.29 | 0 | 0 | 1.297 | 0 | 600 | 0 |
| 384.0 | 384 | 26 | 9.34 | 0 | 0 | 2.284 | 0.77 | 0 | 112.47 | 94.15 | 0 | 0 | 2963.71 | 0.00 | 1.29 | 0 | 0 | 1.3 | 0 | 600 | 0 |
| 385.0 | 385 | 26 | 11.69 | 0 | 0 | 2.347 | 3.081 | 0 | 149.51 | 97.27 | 0 | 0 | 3005.85 | 0.00 | 1.29 | 0 | 0 | 1.302 | 0 | 600 | 0 |
| 386.0 | 386 | 26 | 19.47 | 0 | 0 | 3.766 | 4.169 | 0 | 150.16 | 101.04 | 0 | 0 | 3003.86 | 0.00 | 1.29 | 0 | 0 | 1.302 | 0 | 600 | 0 |
| 387.0 | 387 | 26 | 32.24 | 0 | 0 | 3.158 | 4.096 | 0 | 149.31 | 98.27 | 0 | 0 | 3005.85 | 0.00 | 1.29 | 0 | 0 | 1.301 | 0 | 600 | 0 |
| 388.0 | 388 | 26 | 28.84 | 0 | 0 | 2.894 | 3.664 | 0 | 153.20 | 95.25 | 0 | 0 | 3005.85 | 0.00 | 1.29 | 0 | 0 | 1.301 | 0 | 600 | 0 |
| 389.0 | 389 | 26 | 30.56 | 0 | 0 | 1.989 | 10.57 | 0 | 130.12 | 97.75 | 0 | 0 | 2999.48 | 0.00 | 1.29 | 0 | 0 | 1.301 | 0 | 600 | 0 |
| 390.0 | 390 | 26 | 30.03 | 0 | 0 | 3.136 | 2.311 | 0 | 131.51 | 96.90 | 0 | 0 | 3005.85 | 0.00 | 1.29 | 0 | 0 | 1.3 | 0 | 600 | 0 |
| 391.0 | 391 | 26 | 22.54 | 0 | 0 | 3.019 | 1.834 | 0 | 130.98 | 94.95 | 0 | 0 | 2998.48 | 0.00 | 1.29 | 0 | 0 | 1.3 | 0 | 600 | 0 |
| 392.0 | 392 | 26 | 30.67 | 0 | 0 | 2.932 | 2.067 | 0 | 131.73 | 95.29 | 0 | 0 | 3005.85 | 0.00 | 1.29 | 0 | 0 | 1.3 | 0 | 600 | 0 |
| 393.0 | 393 | 26 | 30.64 | 0 | 0 | 3.071 | 2.154 | 0 | 121.51 | 97.73 | 0 | 0 | 3001.69 | 0.00 | 1.29 | 0 | 0 | 1.3 | 0 | 600 | 0 |
| 394.0 | 394 | 26 | 26.36 | 0 | 0 | 3.519 | 2.564 | 0 | 121.80 | 99.40 | 0 | 0 | 3005.85 | 0.00 | 1.29 | 0 | 0 | 1.299 | 0 | 600 | 0 |
| 395.0 | 395 | 26 | 24.29 | 0 | 0 | 3.295 | 1.423 | 0 | 121.66 | 98.08 | 0 | 0 | 3001.82 | 0.00 | 1.29 | 0 | 0 | 1.3 | 0 | 600 | 0 |
| 396.0 | 396 | 26 | 22.20 | 0 | 0 | 2.052 | 1.878 | 0 | 121.86 | 98.53 | 0 | 0 | 2999.37 | 0.00 | 1.29 | 0 | 0 | 1.301 | 0 | 600 | 0 |

**Figure 13. Overview of the drilling data excel file.**

```
% Name drilling data
tvdrkb = drilling_data(:,1); % True Vertical Depth, meter RKB
mdrkb = drilling_data(:,2);  % Measured Depth, meter RKB
bdia = drilling_data(:,3);   % Bit Size, inches
rop = drilling_data(:,4);    % Rate of Penetration, m/hr
hkla = drilling_data(:,5);   % Hook-load average, ton
hklx = drilling_data(:,6);   % Hook-load max, ton
wob = drilling_data(:,7);    % Weight on Bit, ton
tqa = drilling_data(:,8);    % Torque average, KNm
tqx = drilling_data(:,9);    % Torque max, KNm
rpmb = drilling_data(:,10);  % Revolutions per minute bit, 1/min
sppa = drilling_data(:,11);  % StandPipe Pressure, bar
tva = drilling_data(:,12);   % Active Tank Volume, m^3
mfoa = drilling_data(:,13);  % Mud-flow out average, liter/min
mfia = drilling_data(:,14);  % Mud-flow in average, liter/min
mwout = drilling_data(:,15); % Mud-weight out, sg
mwin = drilling_data(:,16);  % Mud-weight in, sg
mtoa = drilling_data(:,17);  % Mud Temperature out average, °C
mtia = drilling_data(:,18);  % Mud Temperature in average, °C
ecd = drilling_data(:,19);   % Equivalent Circulating Density, sg
ppore = drilling_data(:,20); % Pore Pressure, sg
lith = drilling_data(:,21);  % Lithology
dexp = drilling_data(:,22);  % Drilling exponent
TD = max(mdrkb);             % Target Depth, meter MD RKB
```

**Figure 14. Illustrating the creation of drilling data variables in MATLAB workflow.**

The composite data excel file is a depth-based data set listing various composite data. Of interest for the developed MATLAB workflow is the Gamma-ray log and the Sonic log (**Figure 15**). As petrophysical well log measurements have higher vertical sampling rate than drilling data measurements, the Gamma-ray log and the Sonic-log are interpolated with respect to the measured depth of the drilling data, creating equal length arrays using the MATLAB function *interp1* (**Figure 16).** Empty cells, or NaN values, are removed prior to interpolation for the *interp1* function to work.

| DEPTH MD RKB | GR | DEPTH MD RKB | AC |
|---|---|---|---|
| m | gAPI | m | µs/ft |
| 356.0064 | 0 | 356.0064 | 0 |
| 356.1588 | 0 | 356.1588 | 0 |
| 356.3112 | 0 | 356.3112 | 0 |
| 356.4636 | 0 | 356.4636 | 0 |
| 356.616 | 0 | 356.616 | 0 |
| 356.7684 | 155.569 | 356.7684 | 0 |
| 356.9208 | 156.326 | 356.9208 | 0 |
| 357.0732 | 158.547 | 357.0732 | 0 |
| 357.2256 | 160.865 | 357.2256 | 0 |
| 357.378 | 161.169 | 357.378 | 0 |
| 357.5304 | 164.338 | 357.5304 | 0 |
| 357.6828 | 166.484 | 357.6828 | 0 |
| 357.8352 | 163.714 | 357.8352 | 0 |
| 357.9876 | 162.701 | 357.9876 | 0 |
| 358.14 | 162.775 | 358.14 | 0 |
| 358.2924 | 160.455 | 358.2924 | 0 |
| 358.4448 | 157.167 | 358.4448 | 0 |
| 358.5972 | 159.896 | 358.5972 | 0 |
| 358.7496 | 167.405 | 358.7496 | 0 |
| 358.902 | 168.612 | 358.902 | 0 |
| 359.0544 | 166.058 | 359.0544 | 0 |
| 359.2068 | 166.04 | 359.2068 | 0 |

**Figure 15. Overview of composite data excel file.**

```
% Name Composite Log Data
gammaray_raw = composite_data(:, 5);         % Gamma-ray, raw data, gAPI
gammaray_depth_raw = composite_data(:, 4);  % Gamma-ray, depth meter MD RKB
sonic_raw = composite_data(:, 7);           % Sonic log, raw data, µs/ft
sonic_depth_raw = composite_data(:, 6);     % Sonic log, depth meter MD RKB

% Create equal length arrays for composite logs and drilling data
gammaray_raw(isnan(gammaray_raw)) = []; % Remove NaN values
gammaray_depth_raw(isnan(gammaray_depth_raw)) = []; % Remove NaN values
sonic_raw(isnan(sonic_raw)) = []; % Remove NaN values
sonic_depth_raw(isnan(sonic_depth_raw)) = []; % Remove NaN values

gammaray = interp1(gammaray_depth_raw, gammaray_raw, mdrkb); % Gamma-ray, gAPI
sonic = interp1(sonic_depth_raw, sonic_raw, mdrkb);          % Sonic log, µs/ft
```

**Figure 16. Illustrating the creation of variables and interpolation of Gamma-ray and Sonic Data.**

The bit change data excel file is a depth-based data set listing various information related to bit data. Information include: bit number, depth bit in, depth bit out, bit size, bit type, track and coring (**Figure 17**). **Figure 18** illustrates the creation of bit data variables in MATLAB.

| Bit | Depth In (MD RKB) | Depth out (MD RKB) | Size [in] | Type | Track | Core |
|---|---|---|---|---|---|---|
| 1 | 370 | 1048 | 26 | .. | 1 | 1 |
| 2 | 1048 | 1995 | 17 1/2 | PDC | 1 | 1 |
| 3 | 1995 | 3282 | 12 1/4 | PDC | 1 | 1 |
| 4 | 3282 | 3435.2 | 8 1/2 | PDC | 1 | 1 |
| 5 | 3435.2 | 3466.5 | 8 1/2 | Core | 1 | 2 |
| 6 | 3466.5 | 3516.5 | 8 1/2 | Core | 1 | 2 |
| 7 | 3516.5 | 3569 | 8 1/2 | PDC | 1 | 1 |

**Figure 17. Overview of bit change data excel file.**

```
% Give variable names to bit runs
bit_nr = bit_data(:,1);                     % Bit number
bit_in = bit_data(:,2);                     % Depth in bit, [mMD]
bit_out = bit_data(:,3);                    % Depth out bit, [mMD]
bit_size = bit_data(:,4);                   % Bit size, [in]
bit_core = bit_data(:,7);                   % Coring. Core = 1: No Core. Core = 2: Core.
total_bits = length(bit_nr);                % Total number of bits
```

**Figure 18. Illustrating the creation of bit data variables.**

### *3.3 Initial MATLAB Workflow Case Study*

The initial MATLAB workflow is shown in the flowchart in **Figure 19**. The workflow is separated into several categories with the objective of achieving a best possible match with the reported events and high efficiency. Initially, drilling events are estimated from depth-based drilling data only and compared to WOB-ROP plots for each hole section. Furthermore, by including the sonic data and lithology data, the UCS can be estimated. Pore pressure predictions from Geoprovider is imported, enabling the estimation of CCS. The drilling events detection is updated by replacing UCS estimated from drilling data with CCS estimated from sonic data and differential pressure.

Further, estimated UCS from sonic data is separated into hardness categories. Based on cut-off values, UCS replaces CCS in the drilling events detection, enabling detection of Hard Rock directly from UCS estimations. Lastly, the boundaries for bit balling are based on lithology. This enables bit balling and bit wear to be differentiated. With the drilling events detection finalized, lost drilling time is estimated utilizing the updated WOB-ROP efficient bit-lines. Lost trip time is estimated by comparing the bit changes with the detected drilling events. Summing the two, total lost time is estimated, and the results are exported to Excel.

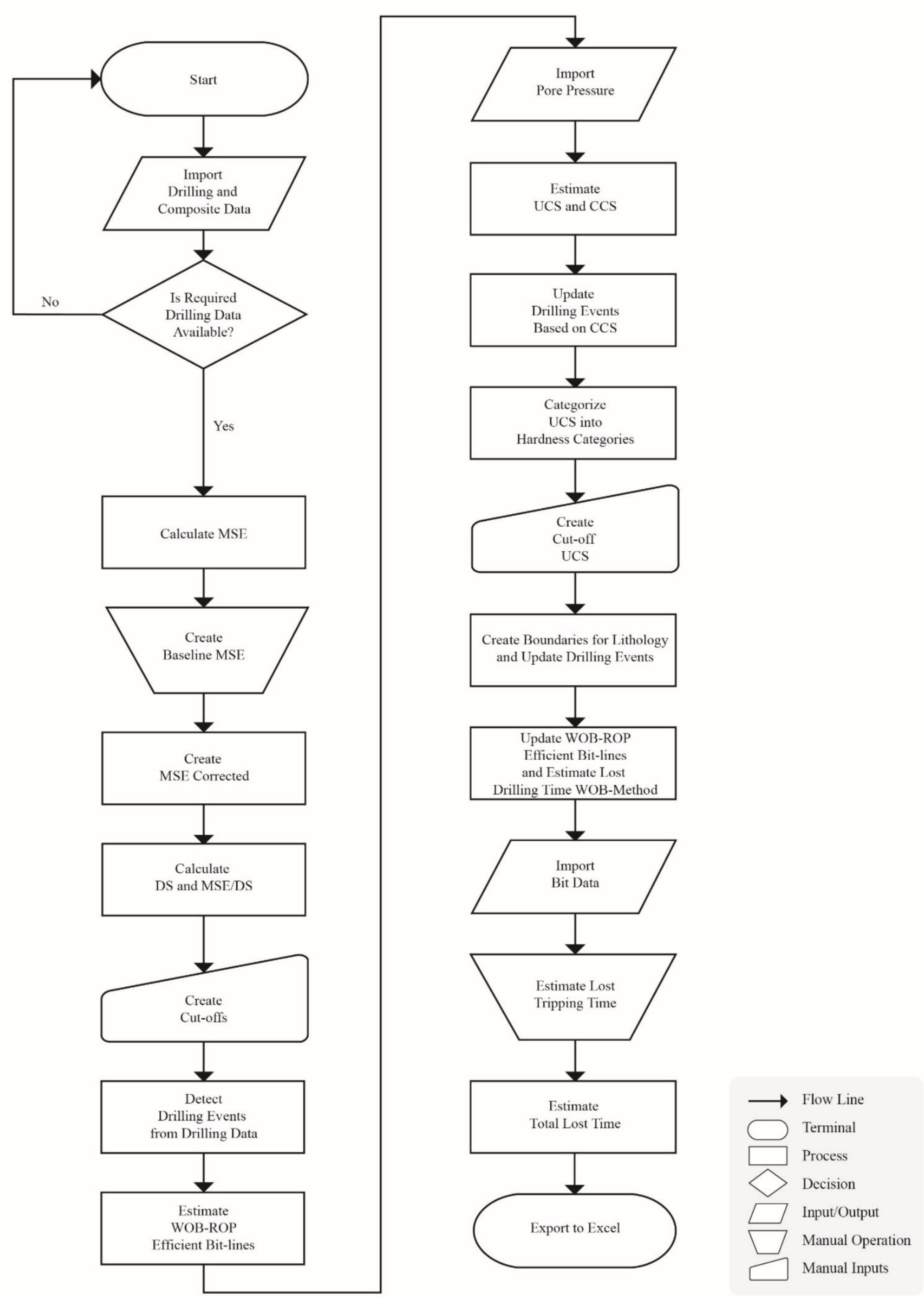


**Figure 19. Flow chart Case Study MATLAB workflow.**

### *3.3.1 Import Drilling and Composite Data*

The first step in the MATLAB workflow involves importing drilling and composite data and creating variables, as illustrated in chapter *3.1 Input Data*. Required drilling data for a full analysis are: ROP, WOB, RPM, Torque and Bit Size (**Figure 20**).

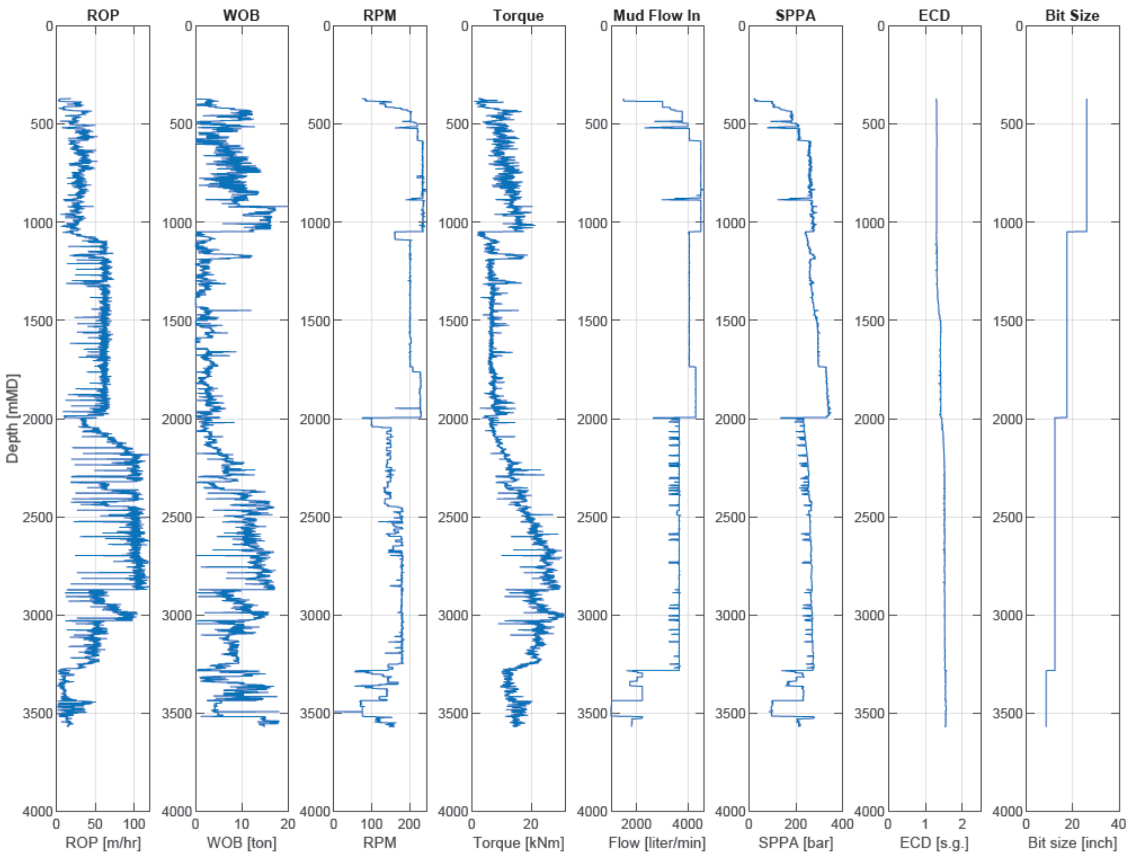


**Figure 20. Visualize drilling data.**

### *3.3.2 Mechanical Specific Energy and MSE Corrected*

The next step is to calculate the mechanical specific energy for the specified well (**Figure 21**). The formula from chapter *2.4 Mechanical Specific Energy* is utilized, by converting to the required units and multiplying by an efficiency factor of 0.35 (Dupriest et al. 2005).

```
wob_lbs = wob*2204.62; % Convert from tonnes to lbs
tqa_ft_lbs = tqa*737.56; % Convert from knm to ft-lbs
rop_ft_hr = rop*3.28; % Convert from m/hr to ft/hr
factor = 0.35; % Efficiency factor

mse_psi = factor * (wob_lbs./((pi/4).*(bdia.^2)) + (120*pi.*rpmb.*tqa_ft_lbs)./(((pi/4).*(bdia.^2)).*rop_ft_hr));
mse = mse_psi./145; % Convert MSE from PSI to MPa
```

**Figure 21. Illustrating the calculation of Mechanical Specific Energy in MATLAB workflow.**

Due to the increased friction with depth a baseline is needed along the minimum MSE, which is especially important for deviated wells. By subtracting the baseline from the calculated MSE, MSE Corrected is created. **Figure 22** illustrates the calculation of MSE and the creation of the baseline and MSE Corrected. While other studies (e.g. Dupriest et al. 2005) have observed this trend, no studies were found where the trend was removed.

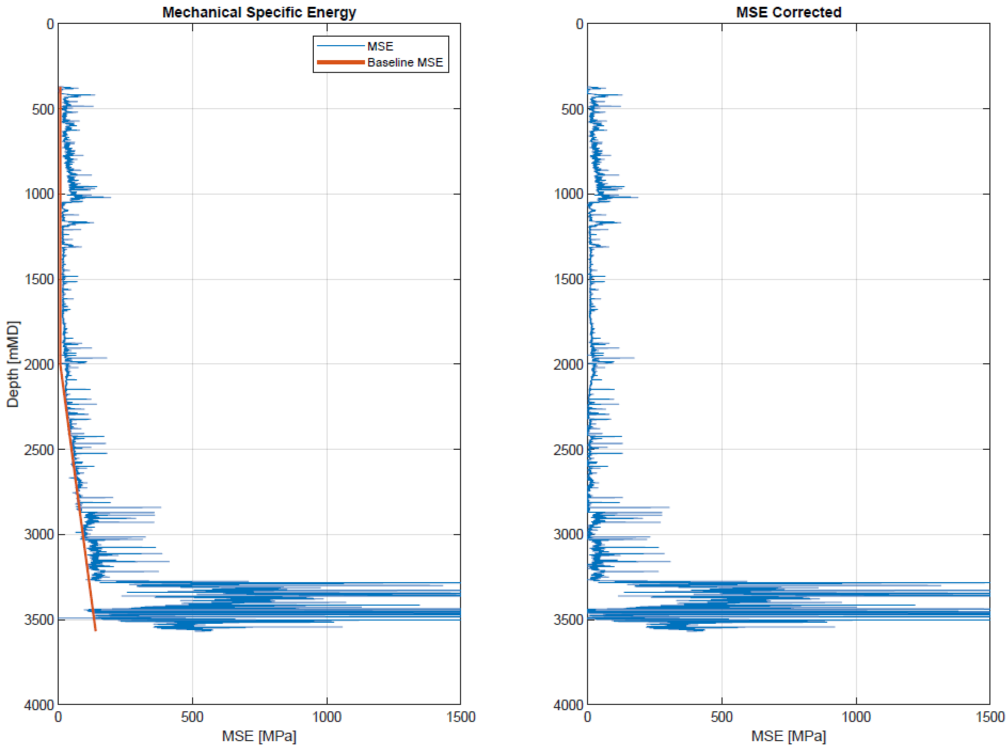


**Figure 22. Illustrating the calculated MSE, creation of MSE baseline and calculated MSE Corrected. Note how the trend is removed for MSE Corrected.**

To create the baseline, values are entered manually under the variable names *baseline_mse_min*, *baseline_mse_max*, *mse_min* and *mse_max*. The number of baseline segments created is entered under variable *baselines*, where minimum 1 and maximum 3 baseline segments can be created. **Figure 23** illustrates how the baseline values are entered in the MATLAB workflow and **Figure 24** shows how the baseline is created.

```
depth_interval = 1; % Interval for each measurement in the drilling data excel file. Default is 1 meter.

% North Sea
%%{
% 34/4-15 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2001]; % Depth Start Baseline
baseline_mse_max = [2000, max(mdrkb)]; % Depth Stop Baseline
mse_min = [8, 8];   % MSE Value Start Baseline
mse_max = [8, 140]; % MSE Value Stop Baseline
%}
```

**Figure 23. Illustrating input start and stop values for MSE baselines.**

```
% Create baselines
if baselines >= 1
    baseline_1 = baseline_mse_min(1):depth_interval:baseline_mse_max(1); % Default: baseline_mse_min(1):1:baseline_mse_max(1);
    baseline1 = [baseline_mse_min(1), baseline_mse_max(1)];
    mse_1 = [mse_min(1), mse_max(1)];
    baseline_mse_1 = (interp1(baseline1, (mse_1), baseline_1));
end

if baselines >= 2
    baseline_2 = baseline_mse_min(2):depth_interval:baseline_mse_max(2); % Default: baseline_mse_min(2):1:baseline_mse_max(2);
    baseline2 = [baseline_mse_min(2), baseline_mse_max(2)];
    mse_2 = [mse_min(2), mse_max(2)];
    baseline_mse_2 = (interp1(baseline2, (mse_2), baseline_2));
end

if baselines >= 3
    baseline_3 = baseline_mse_min(3):depth_interval:baseline_mse_max(3); % Default: baseline_mse_min(3):1:baseline_mse_max(3);
    baseline3 = [baseline_mse_min(3), baseline_mse_max(3)];
    mse_3 = [mse_min(3), mse_max(3)];
    baseline_mse_3 = (interp1(baseline3, (mse_3), baseline_3));
end

if baselines == 1
   baseline_mse = [baseline_mse_1];
elseif baselines == 2
    baseline_mse = [baseline_mse_1, baseline_mse_2];
elseif baselines == 3
   baseline_mse = [baseline_mse_1, baseline_mse_2, baseline_mse_3];
end
```

**Figure 24. Illustrating the creation of MSE baselines.**

### *3.3.3 Drilling Strength and MSE/DS Ratio*

Drilling strength is calculated using the formula from chapter *2.5 Drilling Strength*. The ratio between Mechanical specific energy and drilling strength is calculated by dividing MSE by DS, as illustrated in **Figure 25**. The calculation of DS and MSE/DS in the MATLAB workflow is shown in **Figure 26**.

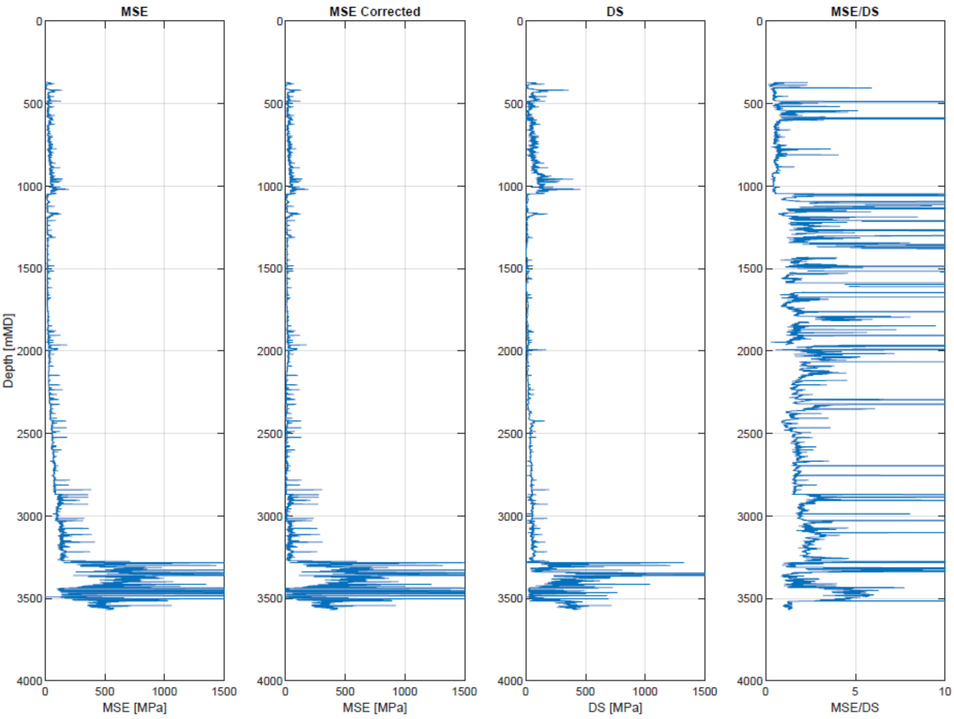


**Figure 25. Illustrating the calculated MSE, MSE Corrected, DS and MSE/DS.**

```
doc = rop_ft_hr./(rpmb.*5); % Depth of cut, [inch]
ds_psi = wob_lbs./(bdia.*doc); % Drilling strength, [psi]
ds = ds_psi /145; % Convert from psi to MPa
mse_ds = mse./ds; % Ratio between Mechanical spesific energy and drilling strength
```

**Figure 26. Showing the calculation of Drilling Strength and MSE/DS in MATLAB Workflow.**

### *3.3.4 Detecting Drilling Events from Drilling Data*

To detect drilling events and NPT from drilling data, cut-off values are selected and plotted as shown in **Figure 27**. The cut-off values and their corresponding events are listed in **Table 6**. When MSE Corrected, DS and MSE/DS values exceed their cut-off value, vibrations are plotted in the scatter plot to the right in Figure 27. When MSE Corrected and DS exceed their cut-off values while MSE/DS remain below, bit balling, bit wear and UCS are plotted. The MATLAB workflow for detecting drilling events from drilling data is shown in **Figure 28**.

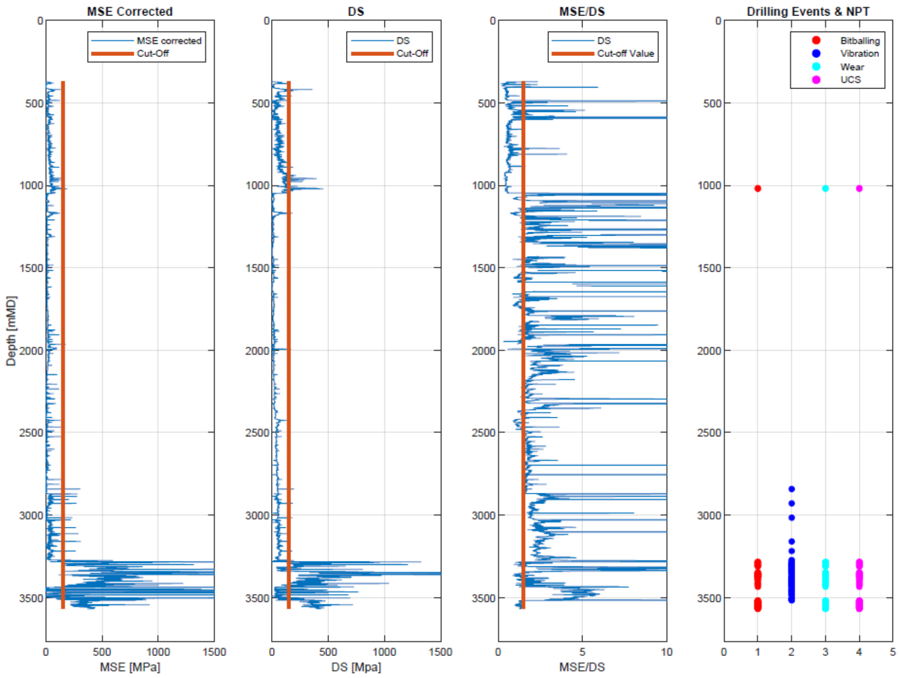


**Figure 27. MSE Corrected, DS, MSE/DS, cut-off values and detection of Drilling Events and NPT. Note how Bit Balling, Vibrations, Bit Wear and/or UCS are plotted in the scatter plot to the right when the cut-off values are crossed.**

**Table 6. Initial cut-off values Case Study MATLAB workflow.**

| MSE | DS | MSE/DS | Event |
|---|---|---|---|
| ↑ 150 MPa | ↑ 150 MPa | ↓ 1.5 | Bit Balling |
| ↑ 150 MPa | ↑ 150 MPa | ↓ 1.5 | Bit Wear |
| ↑ 150 MPa | ↑ 150 MPa | ↑ 1.5 | Vibrations |
| ↑ 150 MPa | ↑ 150 MPa | ↓ 1.5 | UCS |

```
cut_off = 150; % Cut-off value for bitballing, bit wear, vibration and UCS [MPa]
ineff_mse_ds = 1.5; % Cut off values for inefficient drilling mse/ds ratio

a = ones(length(mdrkb), 1); % Array of ones the length of the well
mse_cutoff = cut_off .* a; % Array of bitballing, bit wear, vibration and UCS cutoffs [MPa]
ineff_mse_ds_cutoff = ineff_mse_ds.*a; % Array of mse/ds cutoffs
```

```
b = zeros(length(mdrkb), 1); % Array of zeros
b0 = b - 1; % Not an event
b1 = b+1; % Bit balling
b2 = b + 2; % Vibration
b3 = b + 3; % Wear
b4 = b + 4; % UCS

k = 0;
    for i =1:length(mdrkb)
        k = k+1;
        ucs_event(k) = b0(i);
        bitballing (k) = b0 (i);
        wear(k) = b0(i);
        vibration (k) = b0(i);
    end
```

```
k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
            if mse_corr(k) > mse_cutoff(k) && ds(k) > mse_cutoff(k) && mse_ds(k) < ineff_mse_ds_cutoff(k)
                bitballing (k) = b1 (i);
                wear(k) = b3(i);
                ucs_event (k) = b4(i);
            else
               bitballing (k) = b0 (i);
               wear(k) = b0(i);
               ucs_event(k) = b0(i);
            end
    end

k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
            if mse_corr(k) > mse_cutoff(k) && ds(k) > mse_cutoff(k) && mse_ds(k) > ineff_mse_ds_cutoff(k)
                vibration (k) = b2 (i);
            end
    end
```

**Figure 28. Illustrating how drilling events are detected from drilling data using cut-off values in the initial MATLAB workflow.**

### *3.3.5 WOB - ROP and Efficient Bit-line*

Weight on bit is plotted against penetration rate for every hole section and compared to the detected drilling events (**Figure 29**). As an efficient bit should return a linear relationship between WOB and ROP, a large deviation in the WOB-ROP plot may indicate drilling issues.

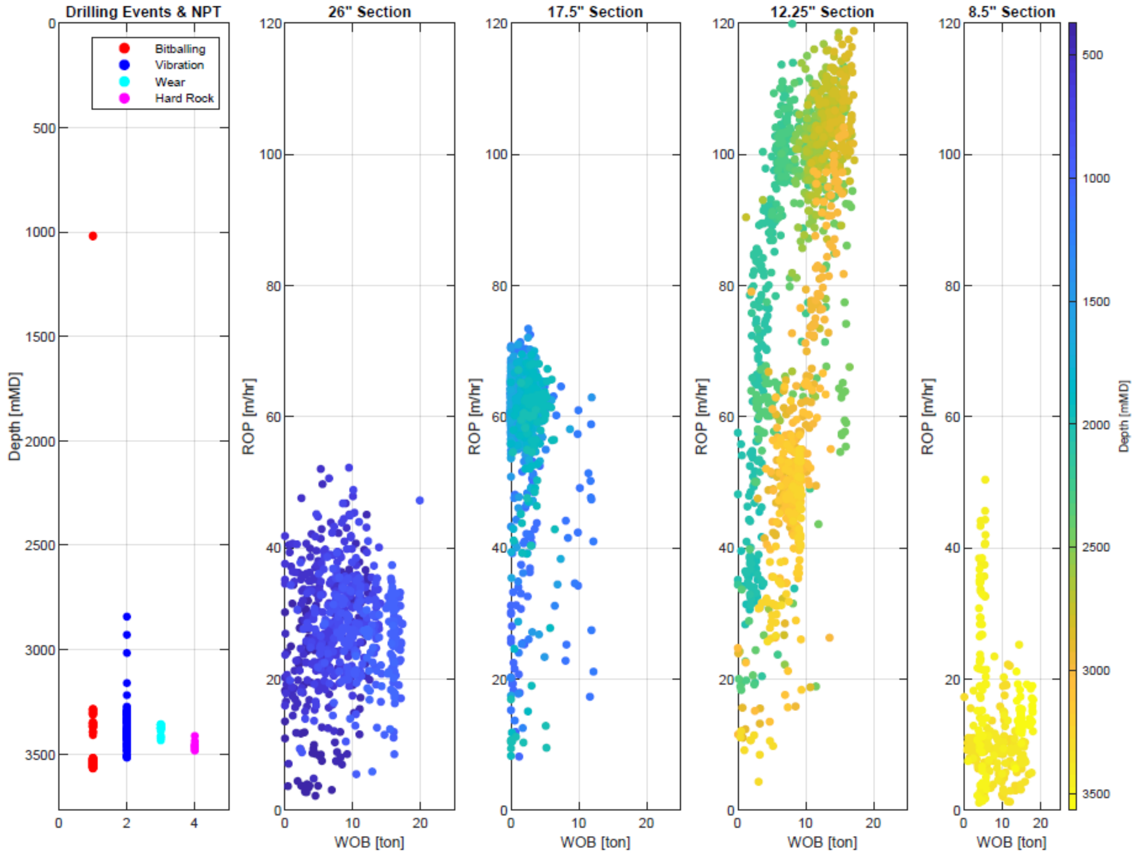


**Figure 29. Comparing WOB-ROP to detected Drilling Events.**

For each hole section, an efficient bit line is created. The efficient bit-line is created by filtering out the depths where drilling events are detected. The remaining depths are considered efficient. The ROP and WOB for the efficient depths are used to estimate the efficient bit-line by using the function *Polyfit* (**Figure 30**). The Polyfit function returns the best fit (least-square) coefficients a and b, and the linear function can be expressed as ROP = a*WOB + b. An example of an efficient bit-line is illustrated in WOB-ROP plot in **Figure 31**.

```
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1 % No events detected
       efficient_rop_8_5(k) = rop_8_5(i);
       efficient_wob_8_5(k) = wob_8_5(i);
    end
end

efficient_rop_8_5(isnan(efficient_rop_8_5)) = []; % Remove NaN values
efficient_wob_8_5(isnan(efficient_wob_8_5)) = []; % Remove NaN values
c = polyfit(efficient_wob_8_5, efficient_rop_8_5, 1); % Create trendline
disp(['ROP = ' num2str(c(1)) '*WOB + ' num2str(c(2))]) % % Display equation ROP = a*WOB + b

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline
```

**Figure 30. Creating efficient bit line using Polyfit function.**

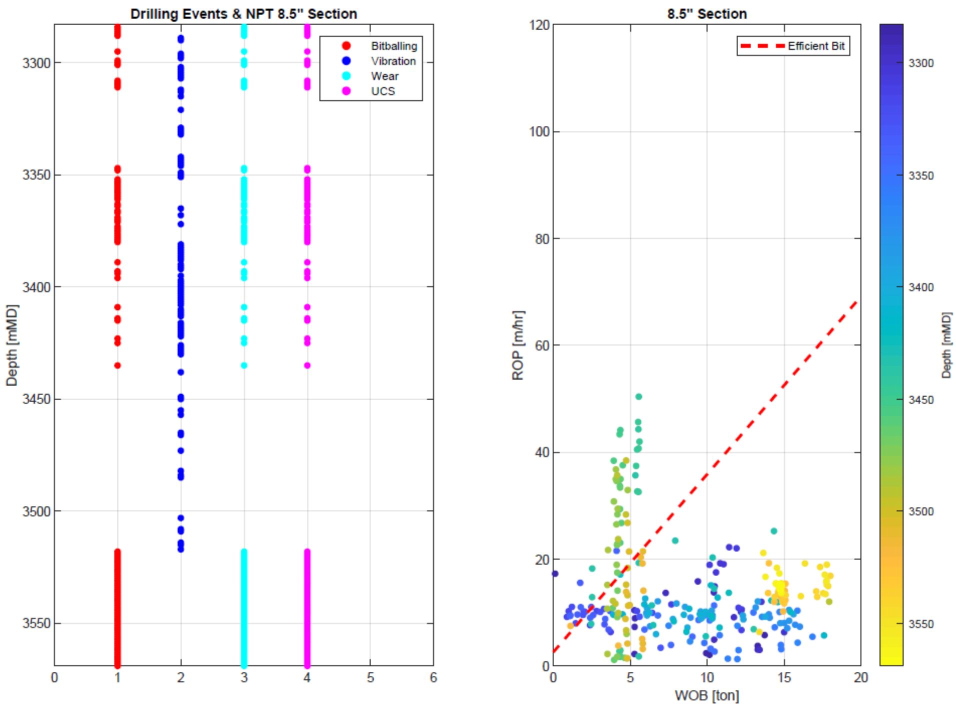


**Figure 31. Example of Drilling Events detection, WOB-ROP and efficient bit line 8 ½" hole section.**

### *3.3.6 Estimate UCS*

To estimate the UCS, lithology and sonic data is utilized. Lithology interpretation, generally found in the drilling data file, is also imported. Based on the lithology interpretation code, the lithology is separated into three categories: claystone, sandstone and carbonates (**Figure 33**). **Figure 32** illustrates the visualisation of Gamma-ray, Sonic and Lithology in the MATLAB workflow.

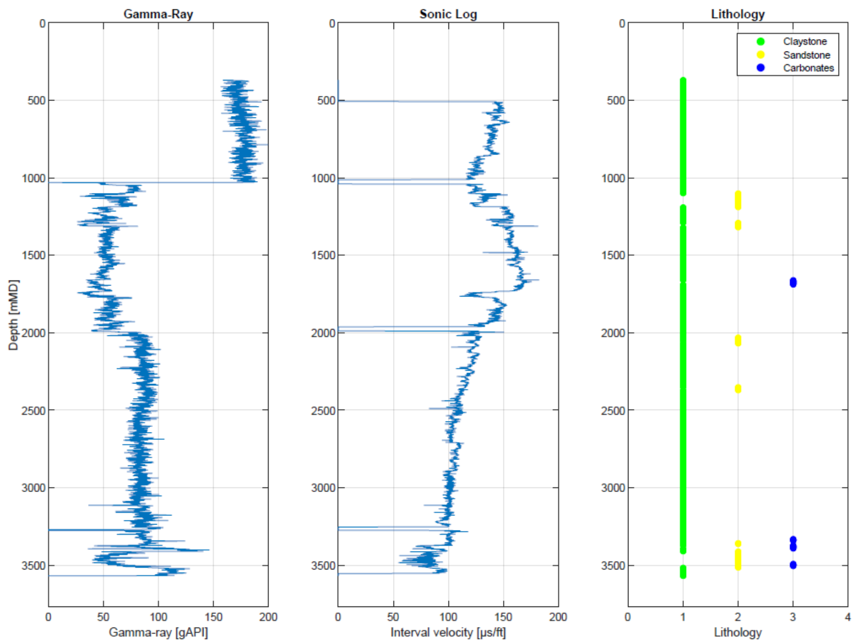


**Figure 32. Visualizing Gamma-ray, Sonic and Lithology.**

```
L = zeros(length(mdrkb), 1); % Array of zeros
k = 0;
for i = 1:length(mdrkb)
    k=k+1;
    if lith(k) >= 600 && lith(k) < 700 % Claystone lithology code
        claystone(k) = L(i)+1;
    else
        claystone(k) = L(i) - 1;
    end
end

k = 0;
for i = 1:length(mdrkb)
    k=k+1;
    if lith(k) >= 300 && lith(k) < 400 % Sandstone lithology code
        sandstone(k) = L(i)+2;
    else
        sandstone(k) = L(i) - 1;
    end
end

k = 0;
for i = 1:length(mdrkb)
    k=k+1;
    if lith(k) >= 700 && lith(k) < 800 % Carbonates lithology code
        carbonates(k) = L(i)+3;
    else
        carbonates(k) = L(i) - 1;
    end
end
```

**Figure 33. Separating the lithology interpretation code into Claystone, Sandstone and Carbonates.**

The UCS is estimated by utilizing the relationship between sonic interval velocity and lithology listed in Table 3 (**Figure 34**). For shallower depths, Gulf Coast Weak Sand is used to estimate sandstone. For depths greater than 2000-meter vertical depth RKB, Bowen Basin Australia Sand is used.

```
Vp = (1./sonic)./(3.2808*10^(-3)); % Convert Sonic from µs/ft to m/s

for i = 1:length(mdrkb)
    if claystone (i) == 1
        UCS(i) = 1.35*(304.8/sonic(i))^2.6;     % Globally Shale
    elseif carbonates(i) == 3
        UCS (i) = ((7682/sonic(i))^1.82)/145;   % Limestone/Dolomites
    elseif sandstone(i) == 2
        if tvdrkb(i) < 2000
            UCS(i) = 1.4138*10^7*(sonic(i)^-3); % Gulf Coast WeakSand
        elseif 2000 <= tvdrkb(i)
            UCS(i) = 1200*exp(-0.036*sonic(i)); % Bowen Basin Australia Sand
        end
    else
        UCS(i) = NaN;
    end
end
```

**Figure 34. Estimation of UCS in MATLAB workflow.**

### *3.3.7 Import Pore Pressure and Estimate CCS*

To estimate the CCS, differential pressure and angle of internal friction are applied. For the Case Study, predicted pore pressure by Geoprovider is imported and differential pressure in the wellbore is estimated by subtracting the predicted pore pressure from the ECD. The angle of internal friction is calculated from the sonic data and the CCS is estimated based on the UCS, differential pressure and the angle of internal friction (**Figure 35**). The angle of internal friction is visualized in **Figure 36** and **Figure 37** illustrates the effect of lithology, UCS and differential pressure on the CCS estimations.

```
% Calculate Angle of Internal Friction
Vp_km = (1./sonic)./(3.2808*10^(-3)); % Convert µs/ft to km/s
AIF = asind((Vp_km-1)./(Vp_km+1)); % Angle of Internal Friction [degrees]
```

```
% Estimate CCS
ecd_bar = ecd.*0.0981.*(tvdrkb-rkb); % Convert ecd from s.g. to bar
Dp_bar = ecd_bar - pp_bar; % Differential pressure [bar]
Dp = Dp_bar./10; % Differential Pressure MPa
ucs = transpose(UCS); % UCS MPa
ccs = ucs + Dp + 2.*Dp.*(sind(AIF)./(1-sind(AIF))); % CCS MPa
```

**Figure 35. Importing pore pressure, calculating Angle of Internal Friction and estimating CCS MATLAB workflow.**

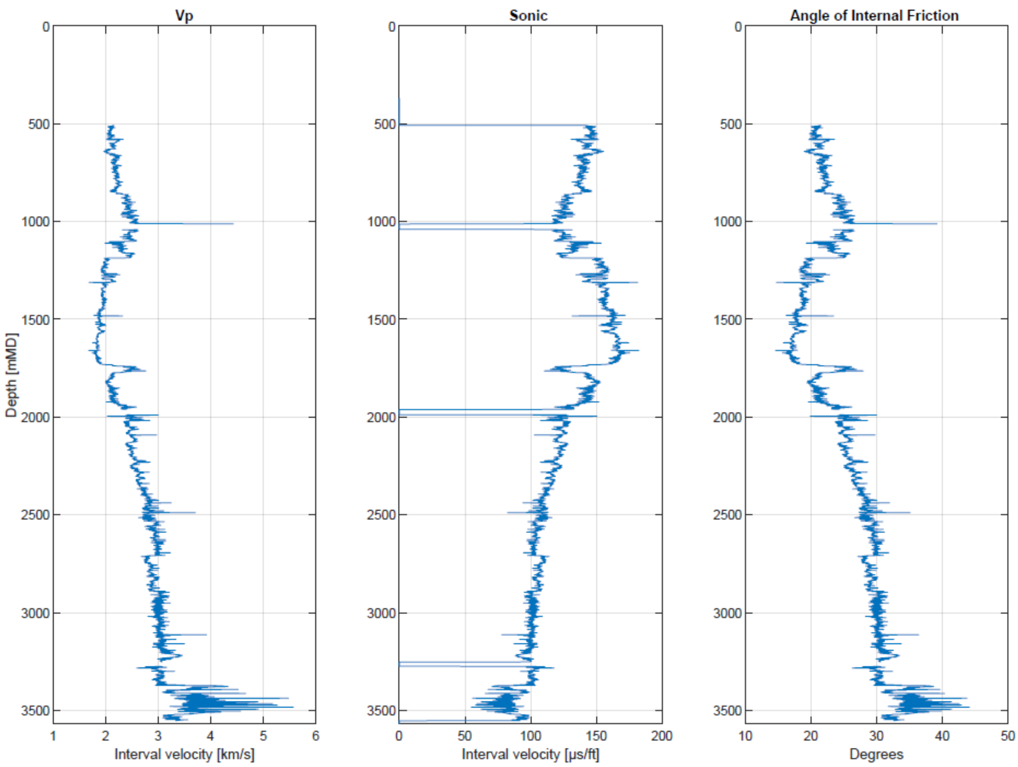


**Figure 36. Visualisation of Sonic Data and Angle of Internal Friction.**

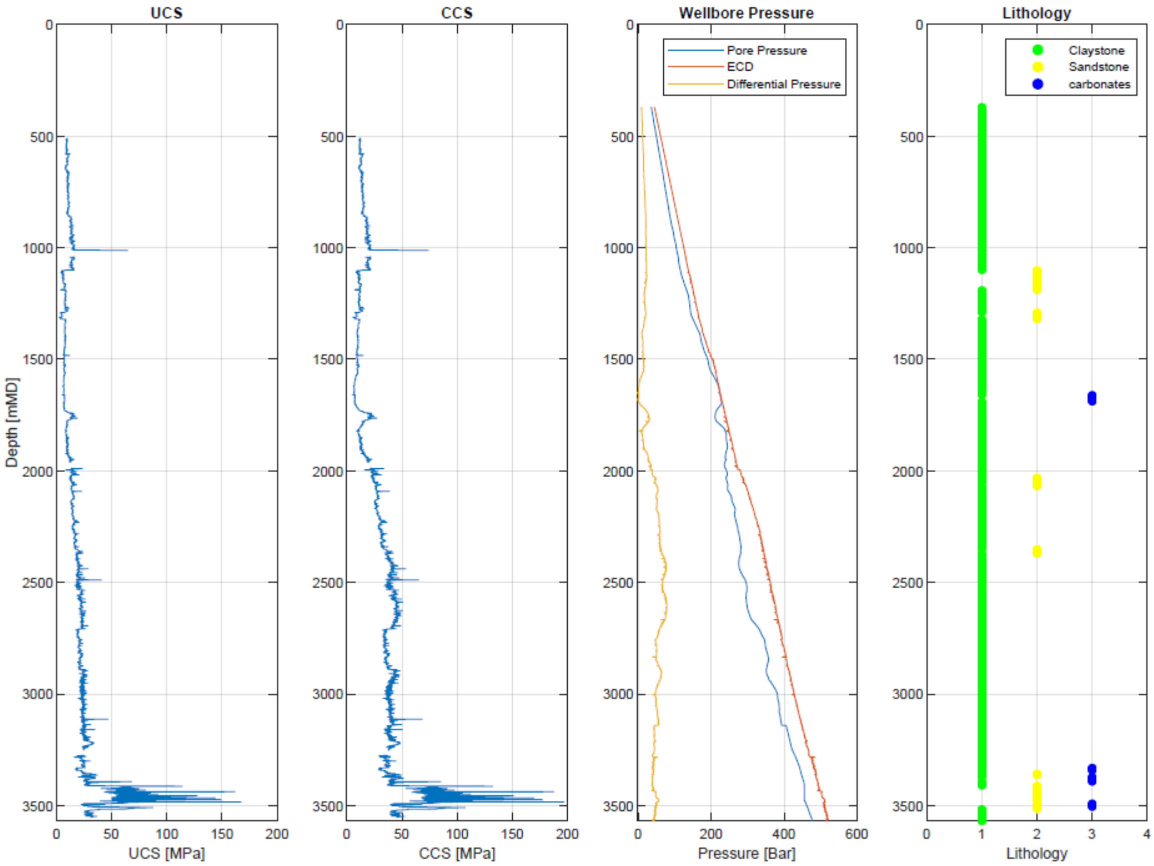


**Figure 37. Illustrating UCS, CCS, wellbore pressures and Lithology.**

*3.3.8 Update Drilling Events based on CCS*

As explained in chapter *2.4 Mechanical Specific Energy*, the MSE value in the subsurface during efficient drilling should be numerically close to the estimated CCS of the rock. By entering a cut-off value for CCS, the drilling events detection is updated (**Figure 38**). If UCS were previously detected from drilling data and the CCS value is above the cut-off value, UCS remains detected, while if the CCS value is below the cut-off value, UCS detection is removed.

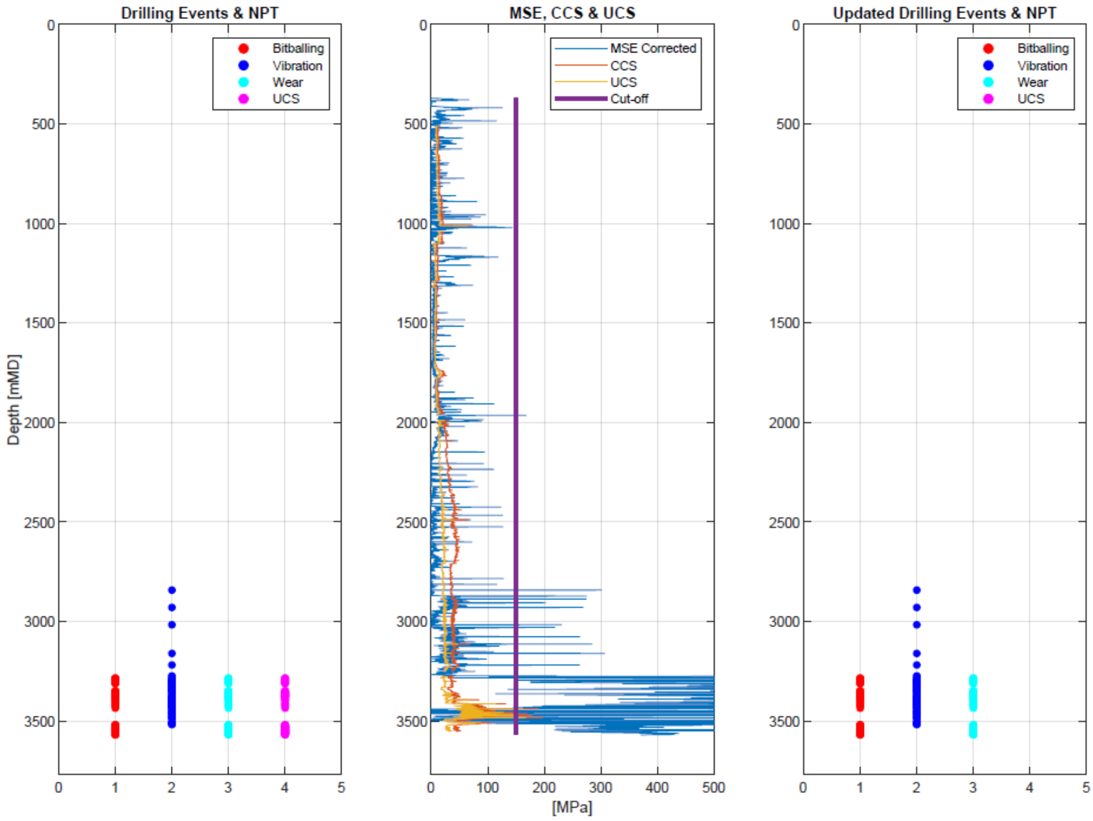


**Figure 38. Illustrating the updated Drilling Events detection based on cut-off values for CCS.**

### *3.3.9 Categorize UCS into Hardness Categories*

The estimated UCS is sorted into hardness categories, using the values listed in Table 2. **Figure 39** shows the creation of variable *hardness*, giving values from 1 to 8 according to hardness category. **Figure 40** visualize the rock hardness compared to UCS and MSE values.

```
% Sort UCS estimations according to Hardness Category

for i = 1:length(mdrkb)
    if ucs(i) <= 0.6 && ucs(i) > 0
        hardness(i) = 1;                  % Soil < 0.6 MPa
    elseif ucs(i) <= 1.25 && ucs(i) > 0.6
        hardness(i) = 2;                  % Very Soft Rock: 0.6 - 1.25 MPa
    elseif ucs(i) <= 5 && ucs(i) > 1.25
        hardness(i) = 3;                  % Soft Rock = 1.25 – 5.0 MPa
    elseif ucs(i) <= 12.5 && ucs(i) > 5
        hardness(i) = 4;                  % Moderately Soft Rock: 5 – 12.5 MPa
    elseif ucs(i) <= 50 && ucs(i) > 12.5
        hardness(i) = 5;                  % Moderately Hard Rock: 12.5 - 50 MPa
    elseif ucs(i) <= 100 && ucs(i) > 50
        hardness(i) = 6;                  % Hard Rock: 50-100 MPa
    elseif ucs(i) <= 250 && ucs(i) > 100
        hardness(i) = 7;                  % Very Hard Rock: 100 - 250 MPa
    elseif ucs(i) < 1000 && ucs(i) > 250
        hardness(i) = 8;                  % Extremely Hard Rock > 250 MPa
    else
        hardness(i) = NaN;
    end
end
```

**Figure 39. Sorting estimated UCS into hardness categories.**

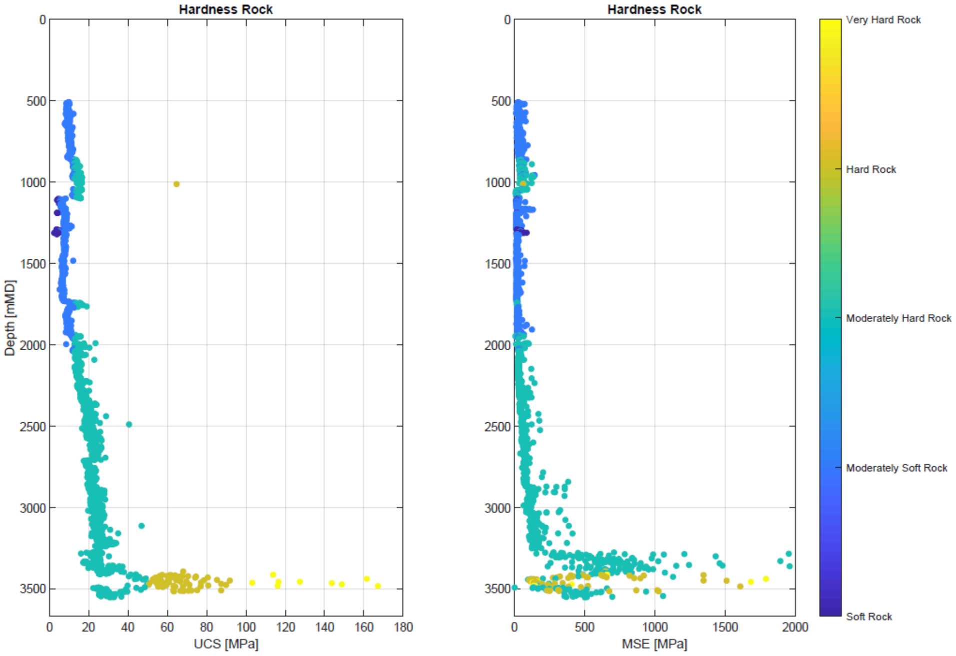


**Figure 40. Visualizing the estimated UCS and calculated MSE by introducing colour gradients representing the hardness categories of the rock.**

### *3.3.10 Create Boundaries for Lithology, Cut-off for UCS and Update Drilling Events*

As bit wear and bit balling are detected by the same drilling parameters, lithology boundaries need to be created to differentiate the two events. The workflow is updated to detect bit balling only if the lithology is claystone. For other lithologies, bit wear is detected. To detect Hard Rock, UCS estimations is used directly and cut-off values for UCS is created (**Figure 41**). **Figure 42** illustrates how lithology differentiates bit balling and bit wear and how hard rock is detected based on cut-off values for UCS.

```
% Create Cut-off for UCS
ucs_cutoff_1 = 100; % Very Hard Rock [MPa]
ucs_cutoff = ucs_cutoff_1.*a; % Array of cut-off
lim_ucs = 1000.*a; % Limit UCS array
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if ucs(i) > ucs_cutoff(i) && ucs(i) < lim_ucs(i) % Update UCS Drilling Events
        ucs_event_3(k) = b4(i);
    else
        ucs_event_3(k) = b0(i);
    end
end
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if claystone(i) == 1 && bitballing_2(i) == 1 % Update bit balling based on lithology
       bitballing_3(k) = b1(i);
    else
       bitballing_3(k) = b0(i);
    end
end
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing_3(i) == -1 && wear_2(i) == 3 && ucs_event_3(i) == -1
       wear_3(k) = b3(i);
    else
       wear_3(k) = b0(i);
    end
end
```

**Figure 41. Updating Drilling Events based on cut-offs for UCS and lithology boundaries for Bit Balling and Bit Wear.**

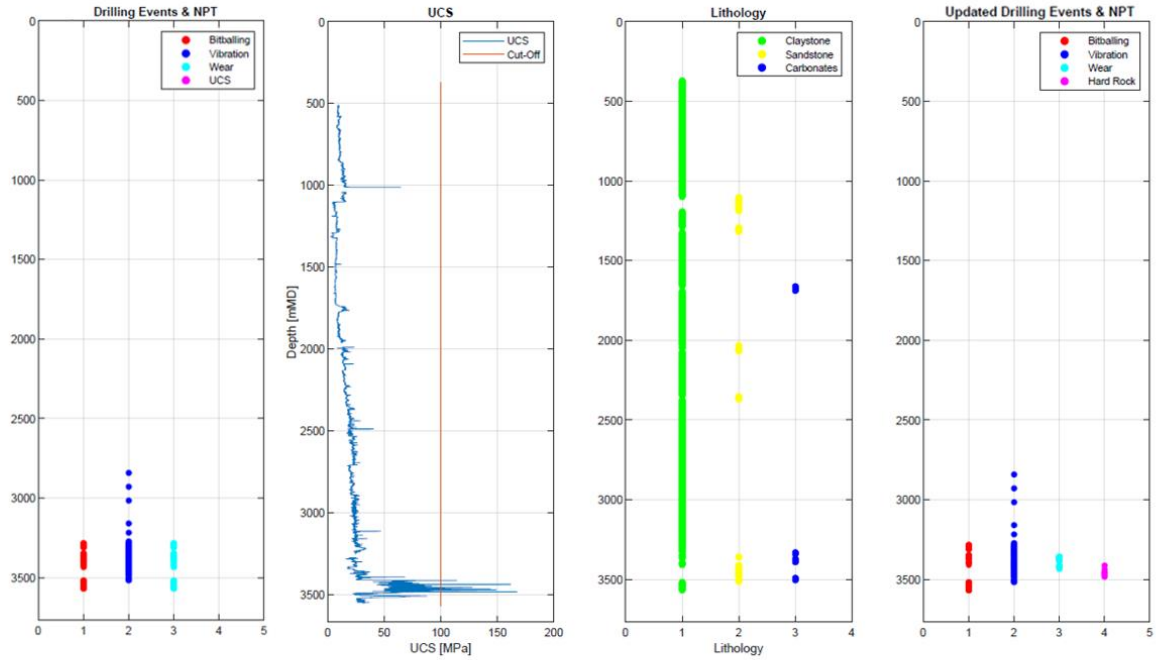


**Figure 42. Updating Drilling Events based on Lithology and UCS. Note how Bit Balling and Bit Wear are differentiated based on lithology and how Hard Rock is estimated from the cut-off for UCS.**

### *3.3.11 Estimate Lost Drilling Time WOB-Method*

To estimate lost drilling time, the estimated ROP-WOB efficient bit line is utilized and a new ROP is estimated (**Figure 43**). The efficient bit-line is estimated as in chapter *3.3.5 WOB - ROP and Efficient Bit-line*, based on the updated drilling events detection. If no event is detected, the new ROP remains the original ROP. If an event is detected, a new ROP is estimated from the efficient bit-line for the given hole section using the original WOB from the drilling data (**Figure 44**). The procedure is repeated for every hole section (**Figure 45**), concatenated and compared to the original ROP **(Figure 46)**.

```
% Calculate new ROP from efficient bit-line for 8.5" hole section
k = 0;
for i = 1:length(mdrkb_8_5)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
        rop_new_8_5(k) = rop_8_5(i); % No events flagged, use original ROP
    else
        rop_new_8_5(k) = c(1)*wob_8_5(i) + c(2); % Flagged event, calculate ROP from Efficient bit-line
    end
end
```

**Figure 43. Calculate New ROP for inefficient depths.**

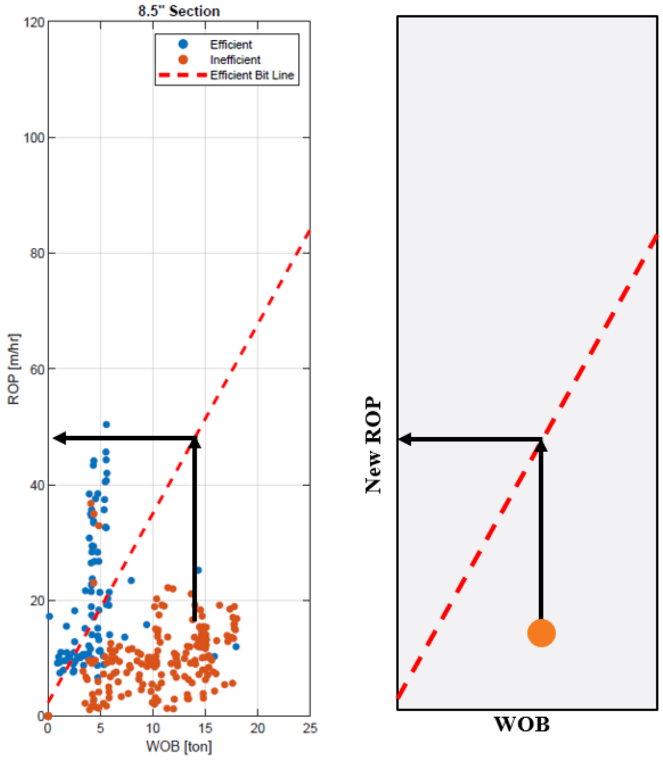


**Figure 44. Illustrating the efficient and inefficient depths and the WOB-Method for New ROP estimation. The orange circle represents an inefficient depth, and how a new ROP is created for this depth based on the original WOB and efficient bit-line.**

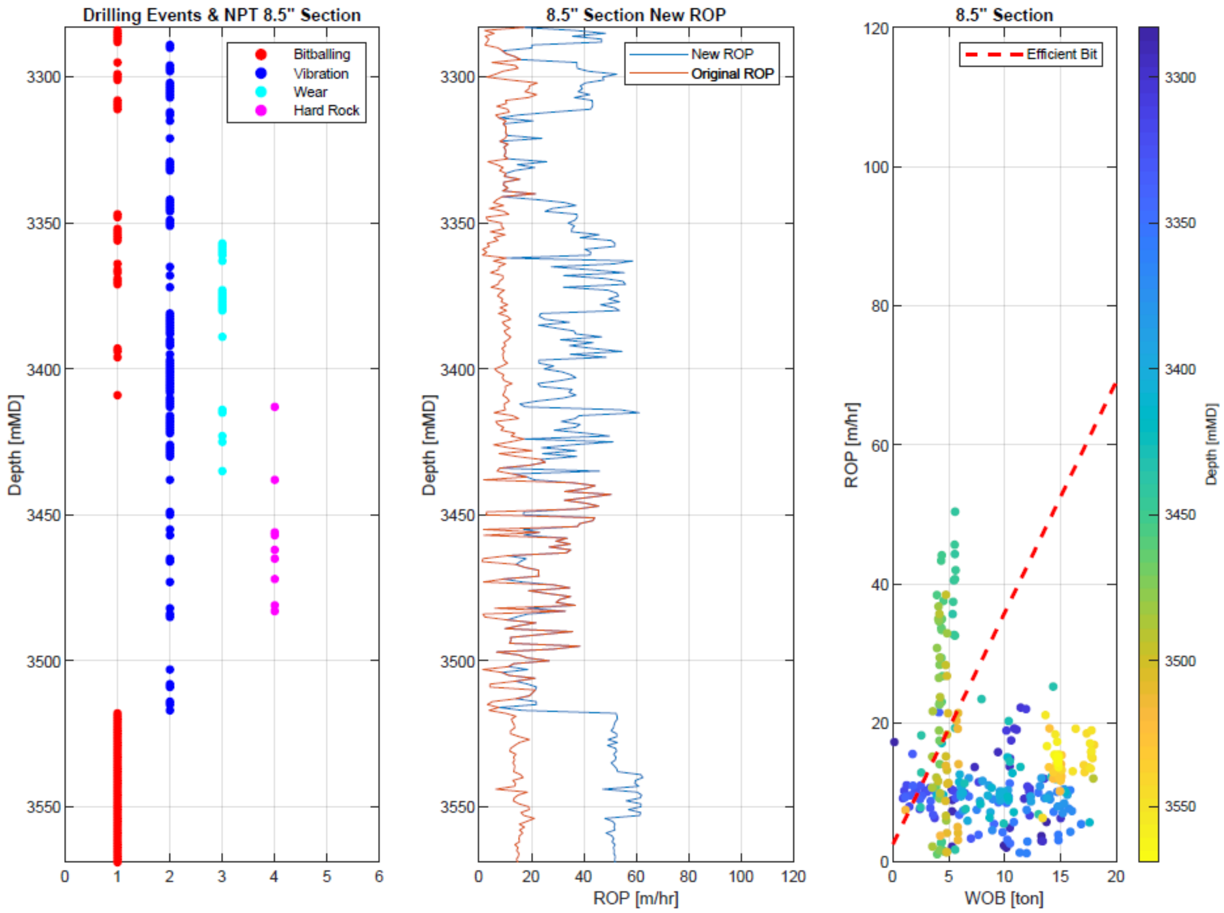


**Figure 45. Drilling Events detection, New ROP and efficient bit-line 8 ½" hole section.**

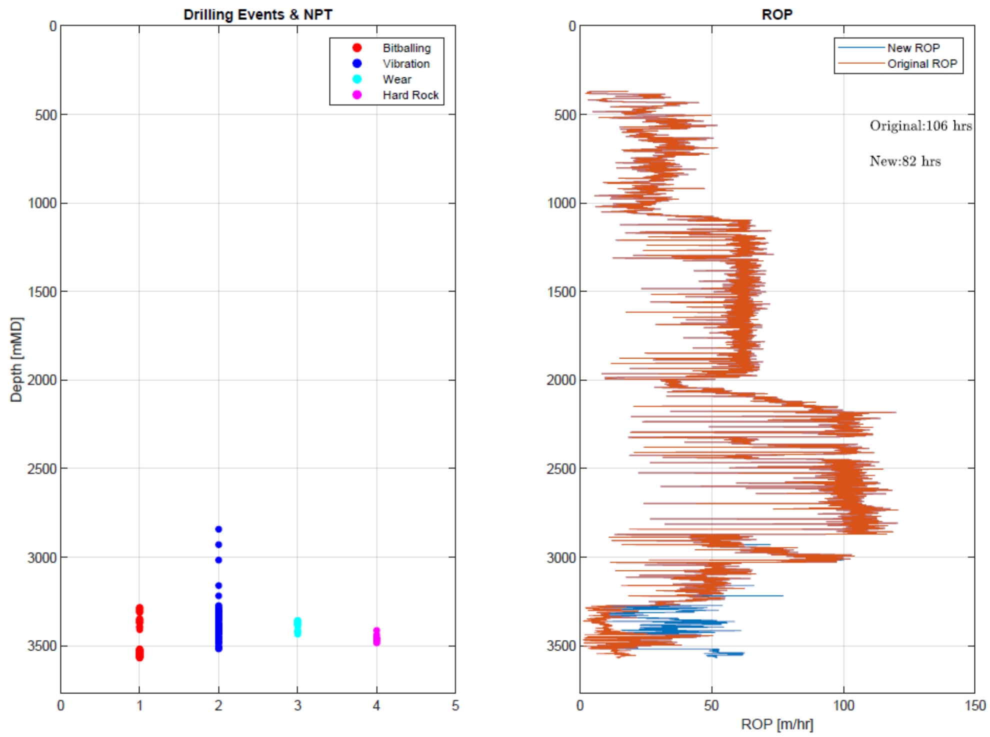


**Figure 46. Illustrating detected Drilling Events, concatenated ROPs, original and new drilling time. Note how a new ROP is estimated when a drilling event is detected and how the new ROP is higher compared to the original ROP.**

### *3.3.12 Estimate Lost Tripping Time*

To estimate lost tripping time due to unplanned bit changes; bit runs, bit size and drilling events are plotted next to each other. By comparing bit runs to bit size, it can be observed when multiple bit runs are needed for the same bit-size. If these bit changes correlate with the detected drilling events, then the bit changes are most likely due to the detected events. These bit changes are entered manually into the *change_bit* variable, where the bit number of the changed bit is entered in the corresponding location in the vector. If the bit change is not due to an event, 0 is entered (**Figure 47**). The depth of the unplanned bit is calculated, and the trip time is estimated based on an assumed tripping speed of 500 m/hr. Total lost time is estimated by summing up lost trip time and lost drilling time (**Figure 48**).

```matlab
% Calculate Tripping Time
trip_speed = 500; % Tripping speed assuming 500 m/hr

% Look at previous figure and study which bits are changed due to drilling
% events. Relate bit changes to bit size and drilling events plot.

change_bit = [0,0,0,0,0,0,0]; % Manually insert the bit number to calculate cost of bit change. Other bits are entered as zero.

if length(change_bit) ~= length(bit_nr)
    disp(['Error! The change bit vector must contain: ' num2str(total_bits) ' numbers'])
else
    disp('Correct bit change inputs.')
end

k=0;
for i = 1:length(bit_nr)
    k = k+1;
    if change_bit (k) == bit_nr(k);
        depth_bit_change(k) = bit_out(k);
        depth_bit_change_1(k) = bit_out(k);
    else
        depth_bit_change(k) = NaN;
        depth_bit_change_1(k) = NaN;
    end
end
```

**Figure 47. Calculating lost tripping time due to Drilling Events.**

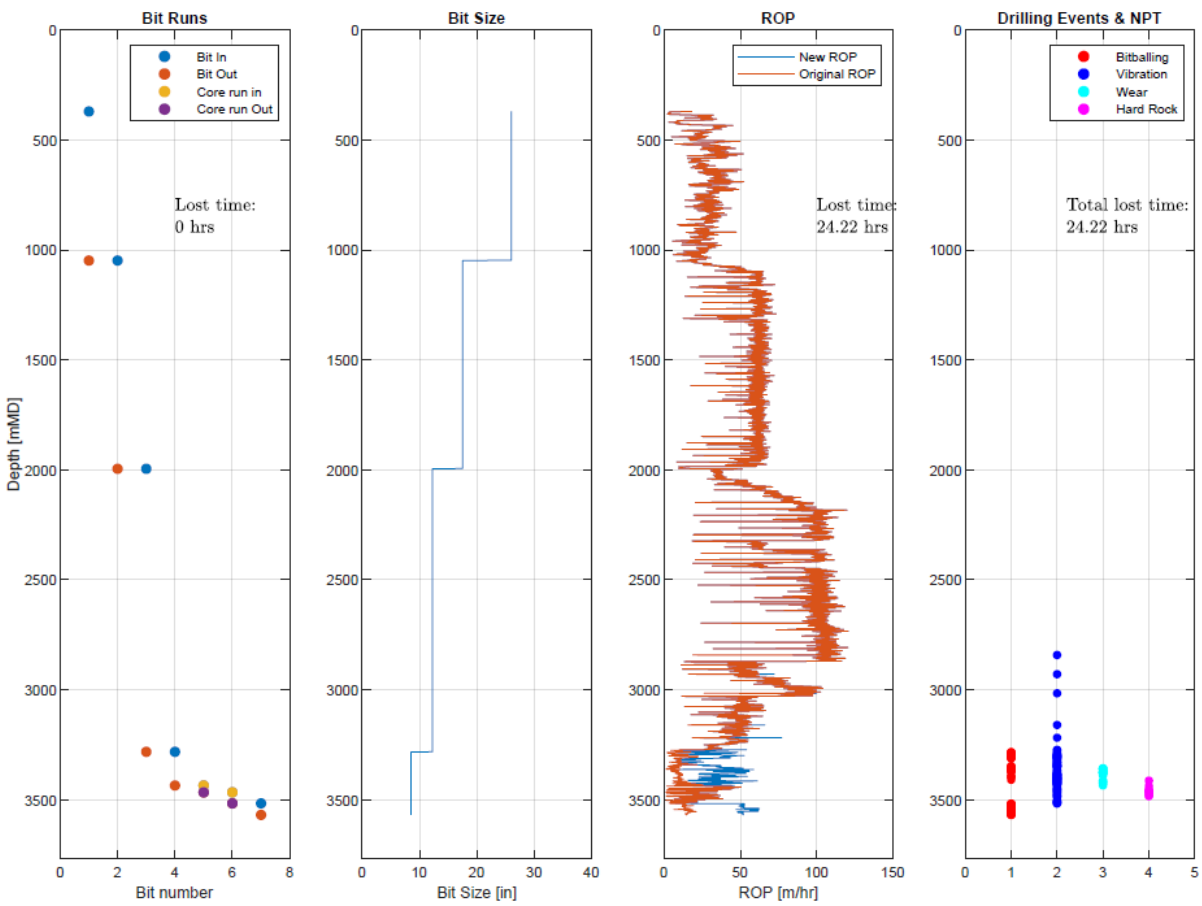


**Figure 48. Illustrating lost time due to bit changes, bit-size, ROPs, Drilling Events and total lost time.**

### *3.4 Updated MATLAB Workflow*

After the initial case study, the MATLAB workflow is updated based on the results from the case study. The updated workflow contains fewer steps, and include lithology restrictions and sonic data from the beginning to differentiate the drilling events (**Figure 49**). This allows a more efficient analysis of multiple wells. MSE, DS and MSE/DS are used to detect bit balling, bit wear and vibrations. To differentiate bit balling from bit wear, lithology restrictions are applied, allowing bit balling only to be detected in the presence of claystone, which coincides well with the theory from chapter *2.1.1 Bit Balling* and the results from the case study.

To detect Hard Rock, or UCS, the relationship between sonic data and lithology are used. The reason for using UCS estimations and not the more advanced CCS estimations when detecting Hard Rock in the updated workflow is due to the observations that for a large increase in UCS, the difference between UCS and CCS are marginal. Additionally, CCS estimations requires reliable pore pressure predictions, which are not readily available. Using UCS will therefore simplify the workflow while still provide good drilling events detections of hard rock. The applied cut-off values for drilling events detection are listed in Table 5. **Figure 50** shows the updated Drilling Events detection plot with the applied cut-off values and lithology restrictions.

To estimate lost drilling time, the initial WOB-method is utilized, as well as a new developed Torque-method. Both methods provide similar results and are presented in separate MATLAB Live scripts. The ability to differentiate between which drilling events contributed to what amount of non-productive time is another addition in the updated workflow.

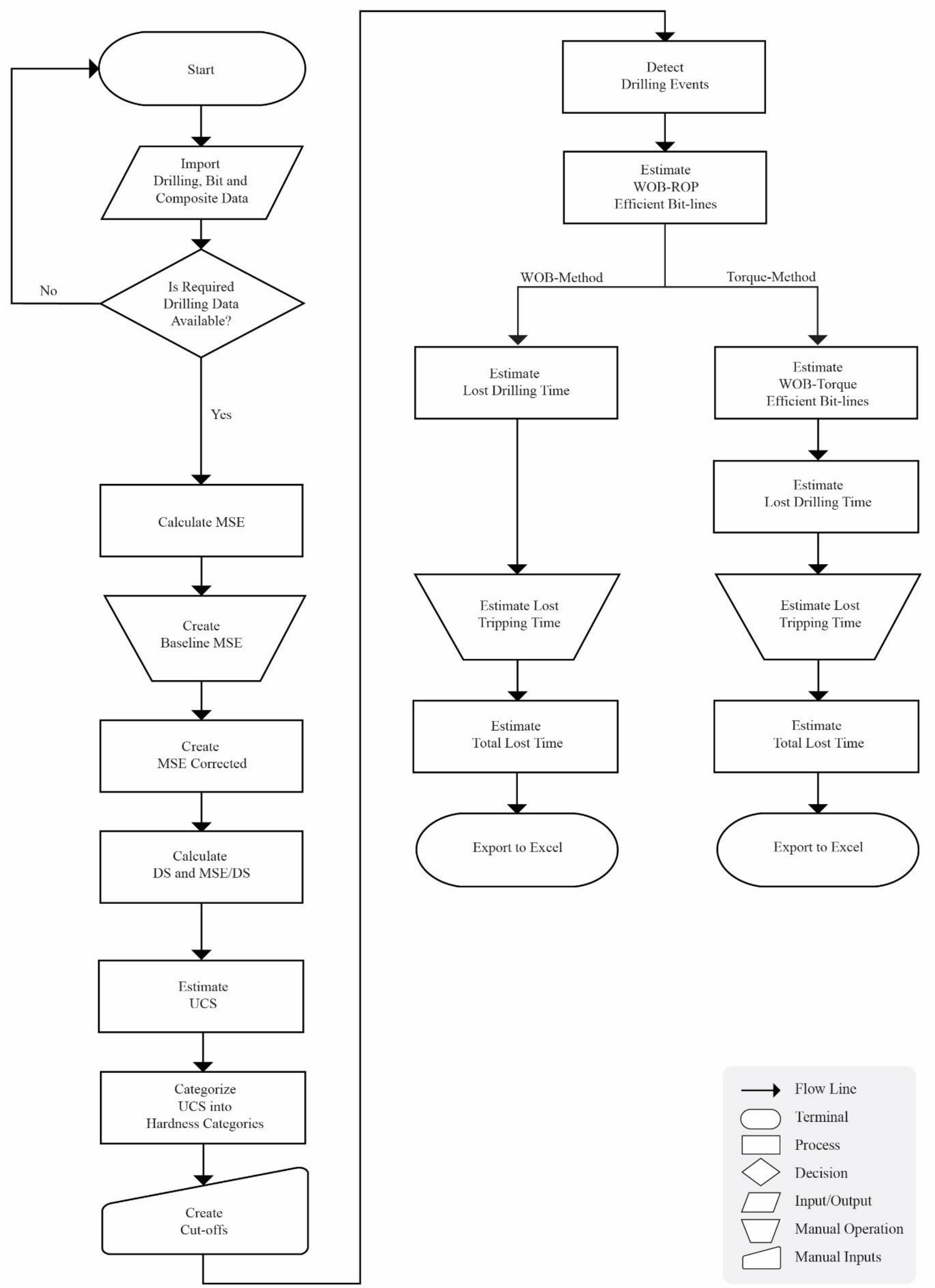


**Figure 49. Flow chart diagram updated MATLAB workflow.**

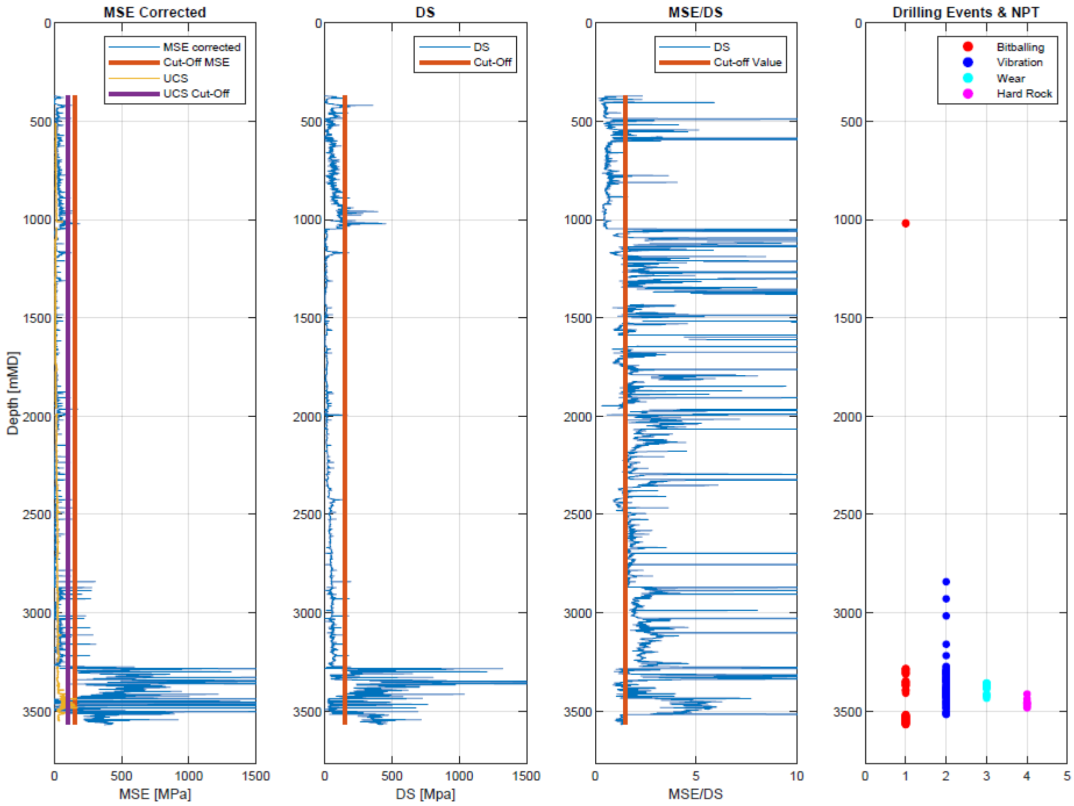


**Figure 50. MSE Corrected, UCS, DS, MSE/DS, Cut-off Values and Drilling Events detection updated MATLAB workflow.**

### *3.4.2 Estimate Lost Drilling Time Torque-Method*

The Torque-method utilize the linear relationship between WOB and Torque to estimate an efficient bit-line. As with the WOB-method, efficient depths are filtered out by the drilling events detection, and the depths are sorted into efficient and inefficient (**Figure 51**). Based on the efficient depths, MATLAB function *Polyfit* is used to create an efficient bit-line on the form Torque=a*WOB + b. Utilizing the Torque-WOB efficient bit-line, the original Torque from the drilling data is used to estimate a new WOB. Furthermore, the new WOB is used in the ROP-WOB efficient bit-line, and a new ROP is estimated (**Figure 52**). This is repeated for every hole section and the new ROPs are concatenated and compared to the original ROP (**Figure 53**).

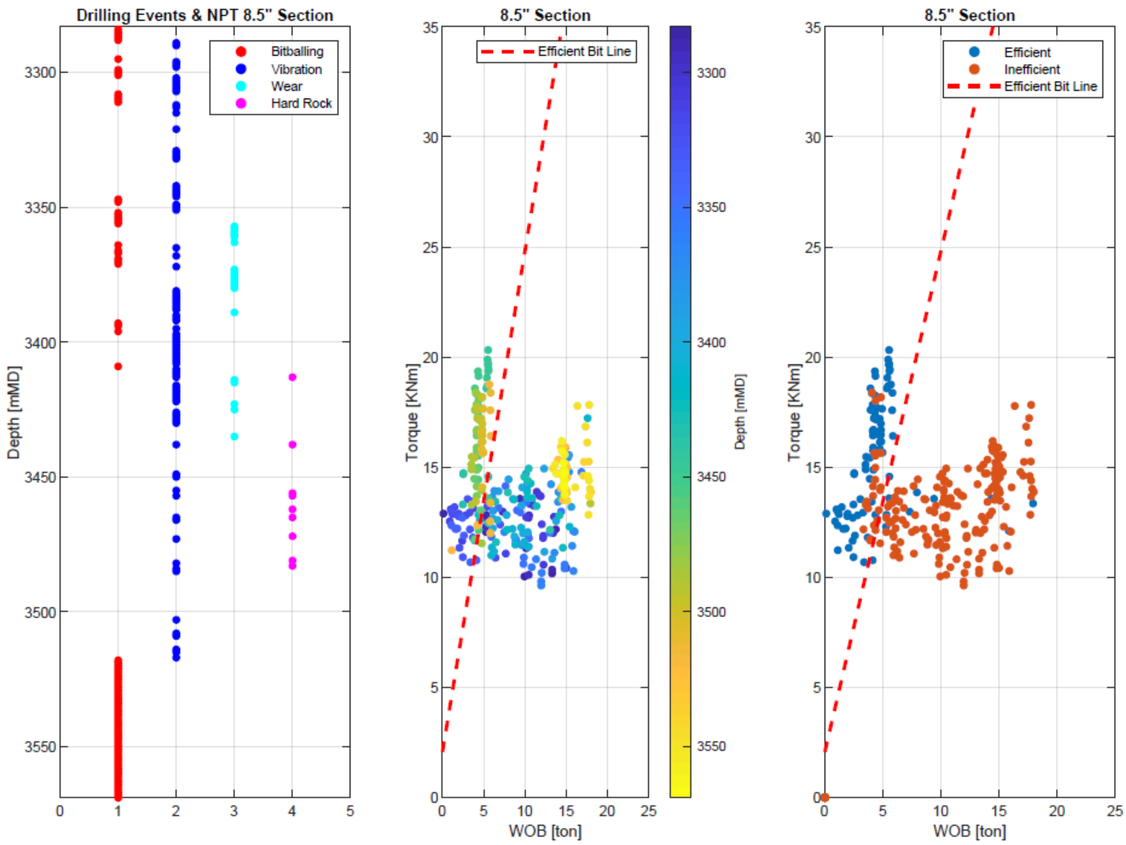


**Figure 51. Drilling Events, Torque-WOB and efficient and inefficient depths. Note how the efficient bit-line is a best-fit line of the blue efficient depth in the plot to the right.**

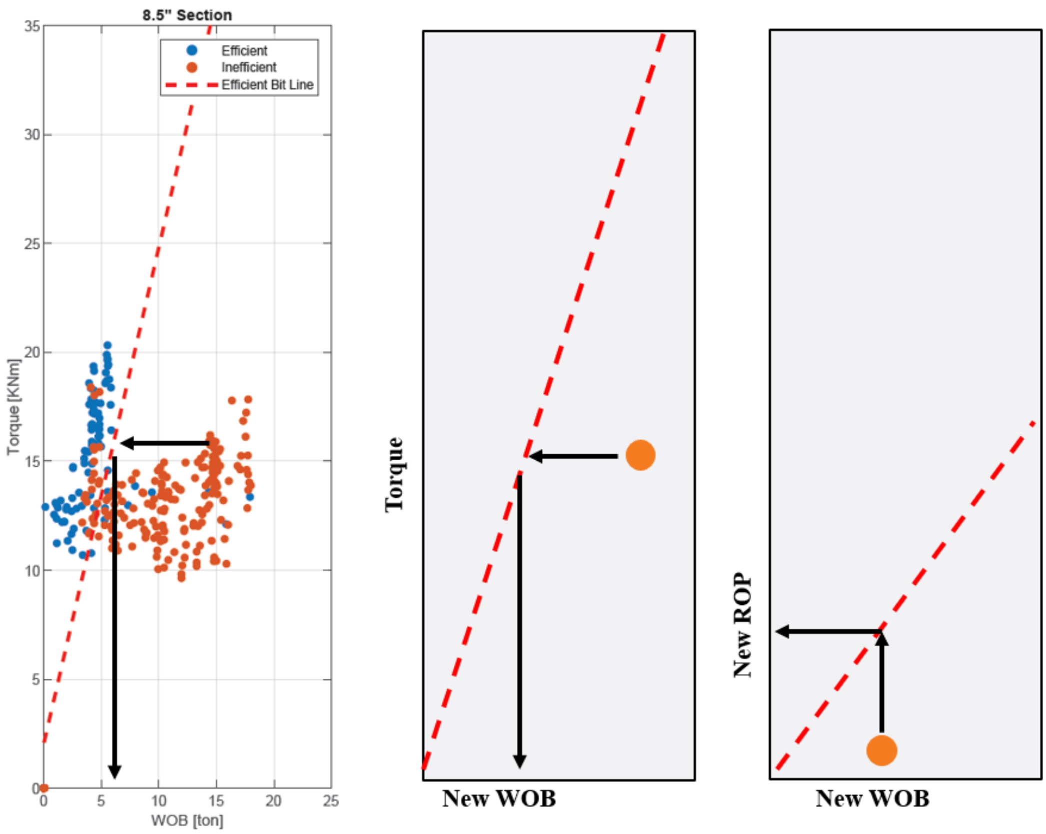


**Figure 52. Illustrating the Torque-Method for ROP estimation. The orange circle represent an inefficient depth, and how a new WOB and a new ROP are created for this depth based on the original Torque and efficient bit-lines.**

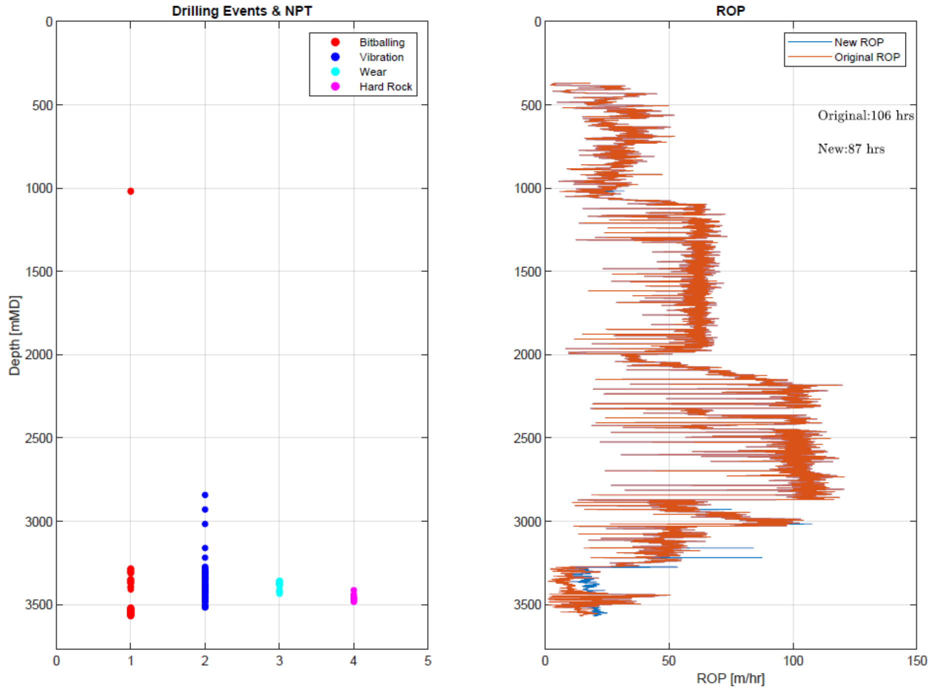


**Figure 53. Illustrating detection of Drilling Events and concatenated ROPs. Note the blue line in the plot to the right representing the New ROP from the Torque-Method estimation.**

### *3.4.3 Differentiate Drilling Events Contribution to NPT*

The updated workflow includes the ability to differentiate which drilling events contributed to what amount of lost time. When a specific drilling event is detected, a new ROP variable is created, estimated from the efficient bit-line using either the WOB-method or the Torque-method (**Figure 54**). The process is repeated for every hole section and concatenated, thus a "New ROP" is created, with a subsequent "Lost Drilling Time" estimation for each drilling event.

```
% Calculate which ROP change is due to which drilling event
k=0;
 for i = 1:length(mdrkb_8_5)
    k = k+1;
    if bitballing(k) == 1
        rop_bb_new_8_5(k) = c_8_5(1)*wob_new_8_5(i) + c_8_5(2);
    else
        rop_bb_new_8_5(k) = rop_8_5(i);
    end
    if vibration(k) == 2
        rop_vibration_new_8_5(k) = c_8_5(1)*wob_new_8_5(i) + c_8_5(2);
    else
        rop_vibration_new_8_5(k) = rop_8_5(i);
    end
    if wear(k) == 3
        rop_wear_new_8_5(k) = c_8_5(1)*wob_new_8_5(i) + c_8_5(2);
    else
        rop_wear_new_8_5(k) = rop_8_5(i);
    end
    if ucs_event(k) == 4
        rop_ucs_new_8_5(k) = c_8_5(1)*wob_new_8_5(i) + c_8_5(2);
    else
        rop_ucs_new_8_5(k) = rop_8_5(i);
    end
 end
```

**Figure 54. Illustrating the differentiation and calculation of Drilling Events contribution to NPT, 8 ½" Hole Section.**

*3.4.4 Estimate Costs due to Drilling Events and NPT*

To calculate the associated costs due to the detected drilling events and NPT, an average rig day-rate of USD 150 000 is assumed. The rig-day rate is based on the worldwide monthly rig-day rates for Semi-submersible rigs from January 2018 to November 2020 from IHS Markit shown in **Figure 55** (IHS Markit, 2021). The total daily well costs, or spread rate, is assumed to be double of the day-rate, at 300 000 dollars, or approx. 2 500 000 mill NOK/day (Schlumberger, 2021). The drilling events and NPT costs are estimated by multiplying the rig-day rate by the lost time estimates.

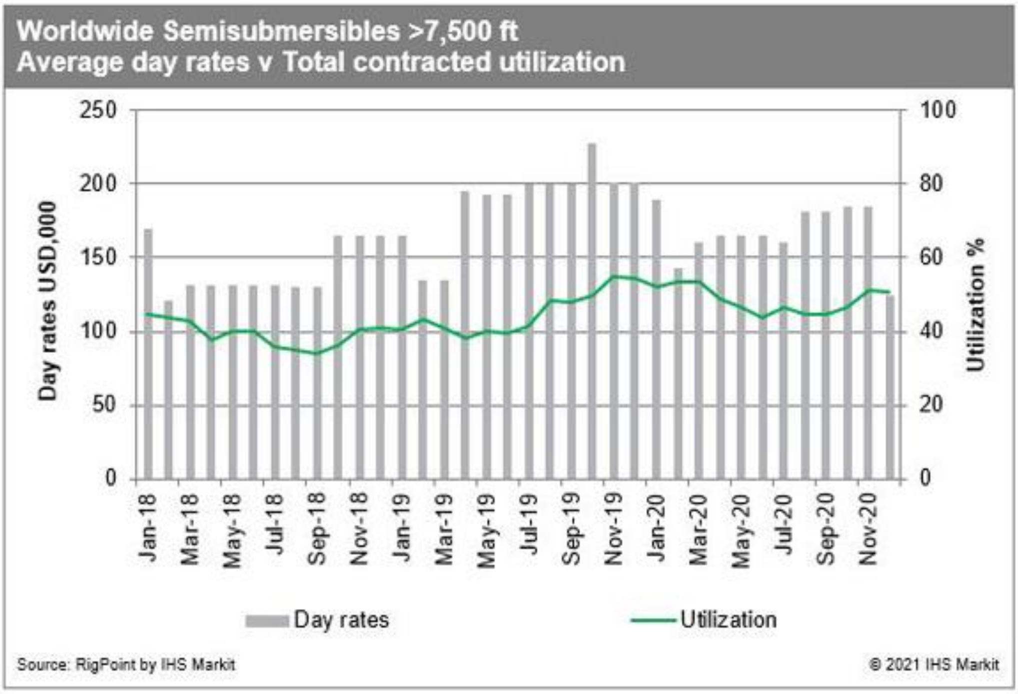


**Figure 55. Worldwide semisubmersibles average day-rates 2018-2020 (IHS Markit, 2021).**

# 4 RESULTS

An initial case study from the Northern North Sea was conducted to develop a prototype MATLAB Workflow and test different approaches. Based on the results from the case study, the MATLAB workflow was updated and a total of 20 wells from the North Sea and the Norwegian Sea were analysed.

### *4.1 Case Study*

The case study was performed on three wells in the Northern North Sea. The three wells were chosen due to their high-quality well data, reliable pore pressure predictions and available detailed final well reports. **Figure 56** presents the area of interest for the case study.

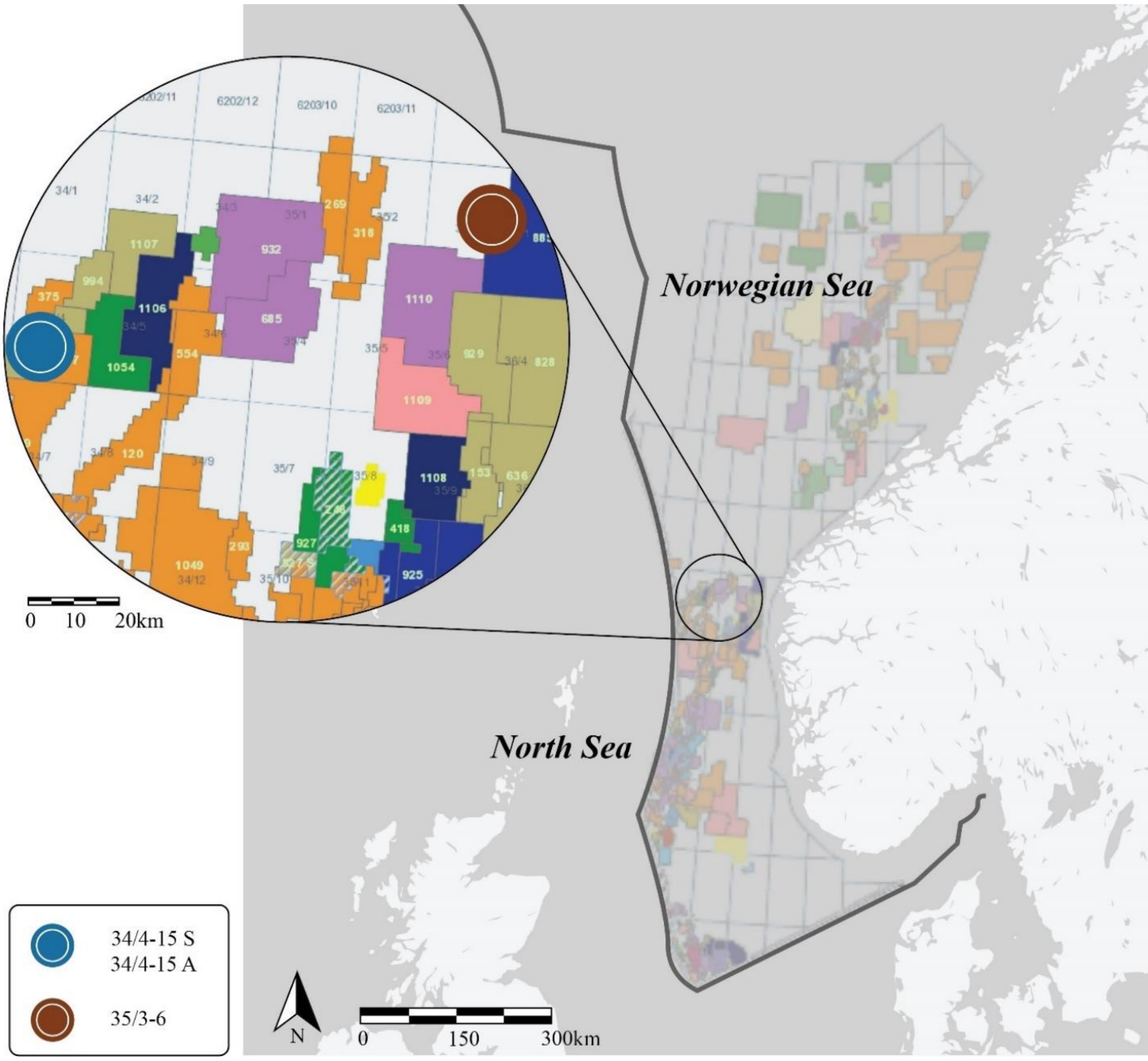


**Figure 56. Area of interest Case Study.**

*4.1.1 Well Overview*

The three chosen wells for the case study were: 34/4-15 S, 34/4-15 A and 35/3-6. An overview is presented in **Table 7**. The main well 34/4-15 S and its appraisal 34/4-15 A presented a good opportunity to develop the workflow from, as they were drilled with the same drilling rigg and drilling fluid, yet one well experienced bit balling and stick-slip vibrations, while the other experienced bit-bouncing vibrations. Well 35/3-6 also experienced bit balling, which occurred at shallower depths compared 34/4-15 S. Well 35/3-6 is 18 years older, drilled with water-based drilling fluid and drilled in a different area, giving good diversification to develop the workflow on. **Table 8**, **Table 9** and **Table 10** lists the reported drilling events from the Final Well Reports for the three wells.

**Table 7. Well overview Case Study.**

| **Well** | **TD** mMD | **TD** mTVD | **Year** | **Drilling Fluid** | **Drilling Days** | **Drilling Efficiency** m/day |
|---|---|---|---|---|---|---|
| 34/4-15 S | 3570 | 3455 | 2020 | OBM | 28 | 127.5 |
| 34/4-15 A | 3844 | 3560 | 2020 | OBM | 28 | 137.3 |
| 35/3-6 | 3366 | 3366 | 2002 | WBM | 56 | 60.1 |

*4.1.2 Summary Drilling Operation*

Well 34/4-15 S was drilled in the Dugong prospect north-west of the Snorre Field in the Northern part of the North Sea. The well spudded on the 17th June 2020 with the semi-submersible drilling rig Deepsea Yantai. The well was drilled with Glydril inhibitive water-based drilling fluid and RMR (Riserless Mud Return) down to 1048 mMD and Versatec Oil-based drilling fluid to TD at 3570 mMD. The well was drilled vertical down to kick-off point at 2010 mMD, with maximum inclination up to 30 degrees. Bit balling was experienced from 3276 mMD to 3436 mMD and stick-slip occurred from 3360 mMD to 3374 mMD.

Well 34/4-15 A side-tracked main well 34/4-15 S at 2012 mMD on the 20th July 2020. The well was drilled with Versatec Oil-based drilling fluid to TD at 3844 mMD, with maximum inclination up to 42 degrees. Bit bouncing was experienced during the second core-run.

Well 35/3-6 was drilled in the Måke Nord prospect in the Northern part of the North Sea. The well spudded on the 6th February 2002 with the semi-submersible drilling rig Deepsea Bergen and drilled vertical to TD at 3366 mMD. The well was drilled with Seawater and hi-vis pills down to 1300 mMD and with Glydril inhibitive water-based drilling fluid from 1300 mMD to TD. Bit balling was experienced in the 12 ¼” hole section, and an anti-bit balling pill was pumped at 1430 mMD, resulting in increased ROP. At 1990 mMD, the drill bit was pulled due to bit balling.

**Table 8. Reported Drilling Events well 34/4-15 S.**

| **Drilling Event** | **Depth** [mMD] |
|---|---|
| Boulders | 950 |
| Pack-off Tendencies | 959 - 1020 |
| Tight Hole | 1031 |
| Drilling Break Utsira fm. | 1191 |
| Bit-balling | 3276 - 3436 |
| Stick-slip | 3360 - 3374 |

**Table 9. Reported Drilling Events well 34/4-15 A.**

| **Drilling Event** | **Depth** [mMD] |
|---|---|
| Boulders | 950 |
| Pack-off Tendencies | 959 - 1020 |
| Tight Hole | 1031 |
| Drilling Break Utsira fm. | 1191 |
| Tight Spot | 3525 |
| Bit Bouncing 2nd core run | 3597 - 3634 |
| Tight Spot | 3751 - 3768 |

**Table 10. Reported Drilling Events well 35/3-6.**

| **Drilling Event** | **Depth** [mMD] |
|---|---|
| Boulder Bed | 270 - 275 |
| Shallow Water Flow | 587 |
| Bit-balling 12 1/4" | 1430 & 1990 |

*4.1.3 MSE Corrected*

The three wells were uploaded to the MATLAB workflow and **Figure 57** shows the calculation of the mechanical specific energy utilizing an efficiency factor of 0.35 (Dupriest et al. 2005) and the created baselines. By subtracting the baseline from the MSE, the MSE Corrected is created (**Figure 58**).

It was observed that for well 34/4-15 S and A, the MSE values were low down to 2000 mMD, where a steady increase in the trend was seen. This coincides well with the kick-off points, and the consequent increase in wellbore friction and torque (**Figure 90** and **Figure 91** in *8.2 Appendix B*). A new baseline was created along this trend for the two wells.

For well 35/3-6, an increase in the trend from near the start of the 12 ¼" hole section at 1300 mMD was observed, with a reduction in the trend near the end. New baselines were created along these trends. It was observed that the MSE values of well 35/3-6 were slightly higher than the other two wells, even though this well was drilled vertical as opposed to the other two with inclinations up to and exceeding 30 degrees. The increased MSE values may be related to the drilling fluid, as well 35/3-6 was drilled with water-based drilling fluid as opposed to oil-based drilling fluid for the other two wells. This observation was also seen at approx. 1000 mMD for well 34/4-15 S and A, where a change from water-based drilling fluid to oil-based drilling fluid occurred and a reduction in MSE value was observed. Drilling parameters for well 35/3-6 are shown in **Figure 92** in *8.2 Appendix B.*

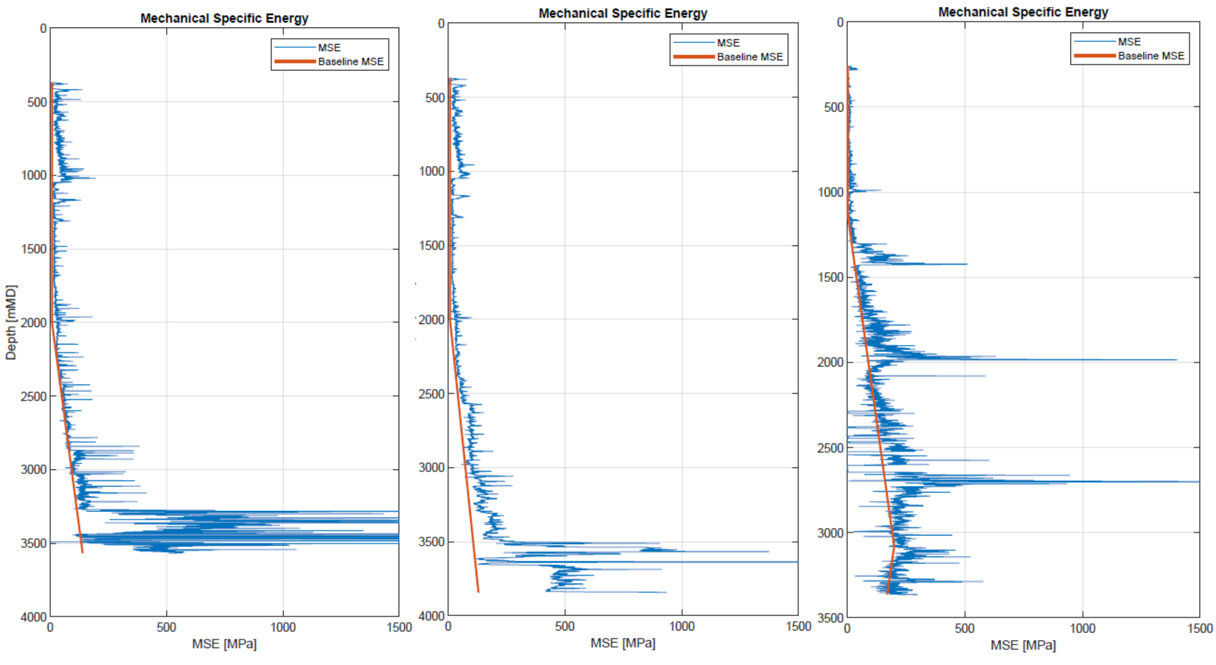


**Figure 57. Mechanical Specific Energy and baselines, well 34/4-15 S, 34/4-15 A and 35/3-6.**

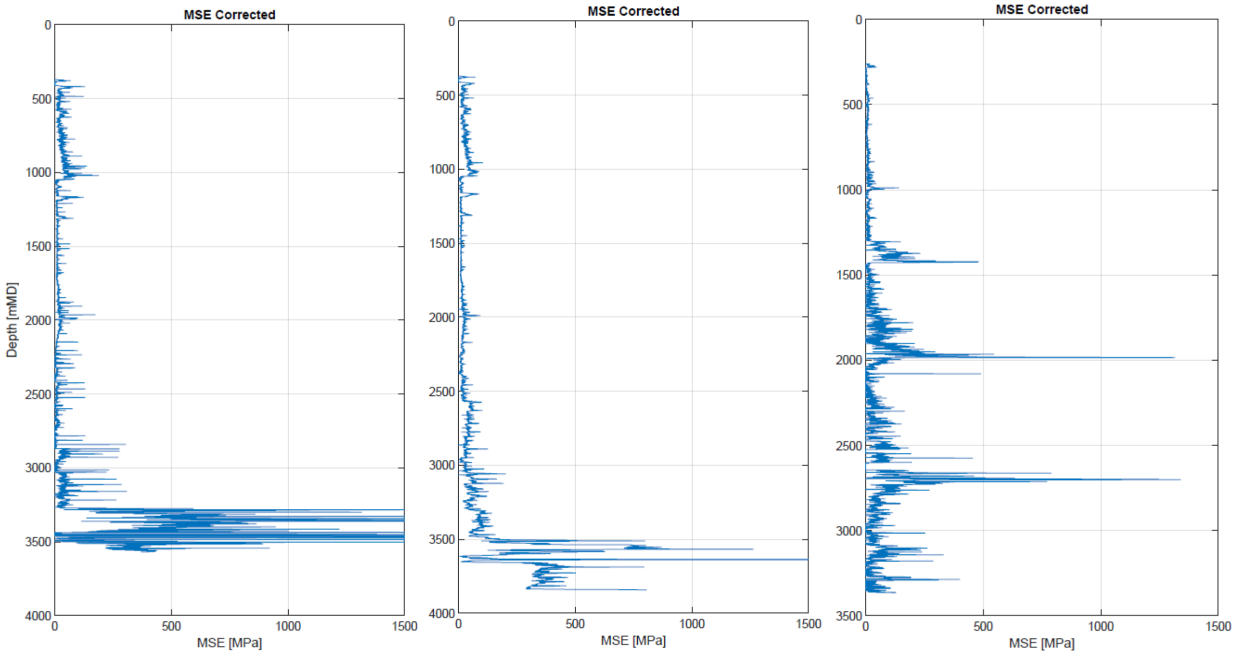


**Figure 58. MSE Corrected, well 34/4-15 S, 34/4-15 A and 35/3-6.**

*4.1.4 Detect Drilling Events and NPT from Depth-based Drilling Mechanics*

Further on, drilling strength and the MSE/DS ratio were calculated. By adjusting the cut-off values for MSE, DS and MSE/DS, drilling events were detected. For well 34/4-15 S and A, bit balling and vibrations were reported from below approx. 3200 mMD. Cut-off values of 150 MPa for MSE and DS matched well with the reported events for these two wells.

The MSE/DS ratio for efficient drilling lies between 1 and 1.5 (Menand & Mills, 2017). It was observed that for a cut-off value of 1, only vibrations were detected, which did not match the reported events for well 34/4-15 S. Using a cut-off value of 1.5, both bit balling and vibrations were detected for well 34/4-15 S, while only vibrations were detected for well 34/4-15 A, giving a better match with the reported events (**Figure 59** and **Figure 60**).

Utilizing the same cut-off values for well 35/3-6, it was observed that the reported bit balling at both approx. 1400 mMD and 2000 mMD were detected. Large increases in both MSE and DS values were observed at these depths, while the MSE/DS remained below the cut-off value of 1.5. A large increase in both MSE and DS values were also observed at 2700 mMD, and the MSE/DS value fluctuated above and below 1.5, detecting both bit balling and vibrations. No events were reported at this depth, making it unclear if the cut-off values should be adjusted (**Figure 61**). Bit wear or hard rock (UCS) could also be the cause of this drilling inefficiency.

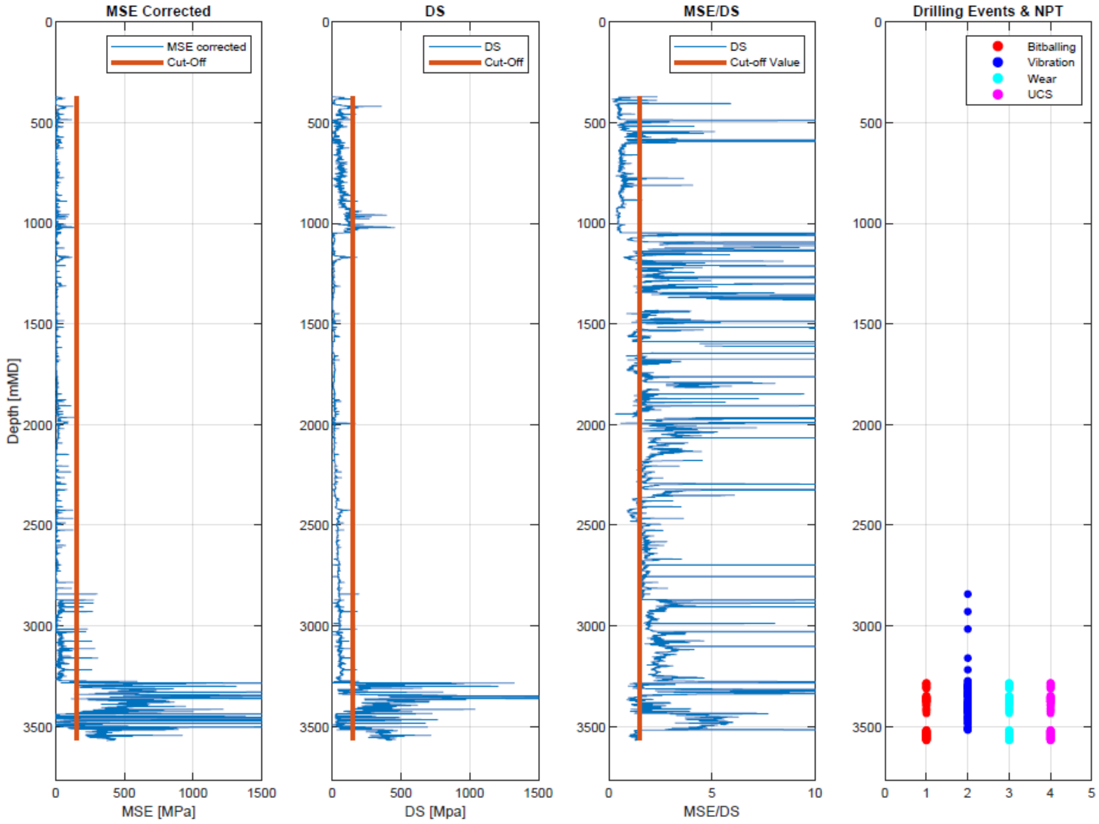


**Figure 59. MSE Corrected, DS, MSE/DS and Drilling Events detection, well 34/4-15 S. Note the increase in MSE and DS from 3200 to 3500 mMD where bit balling and stick-slip was reported.**

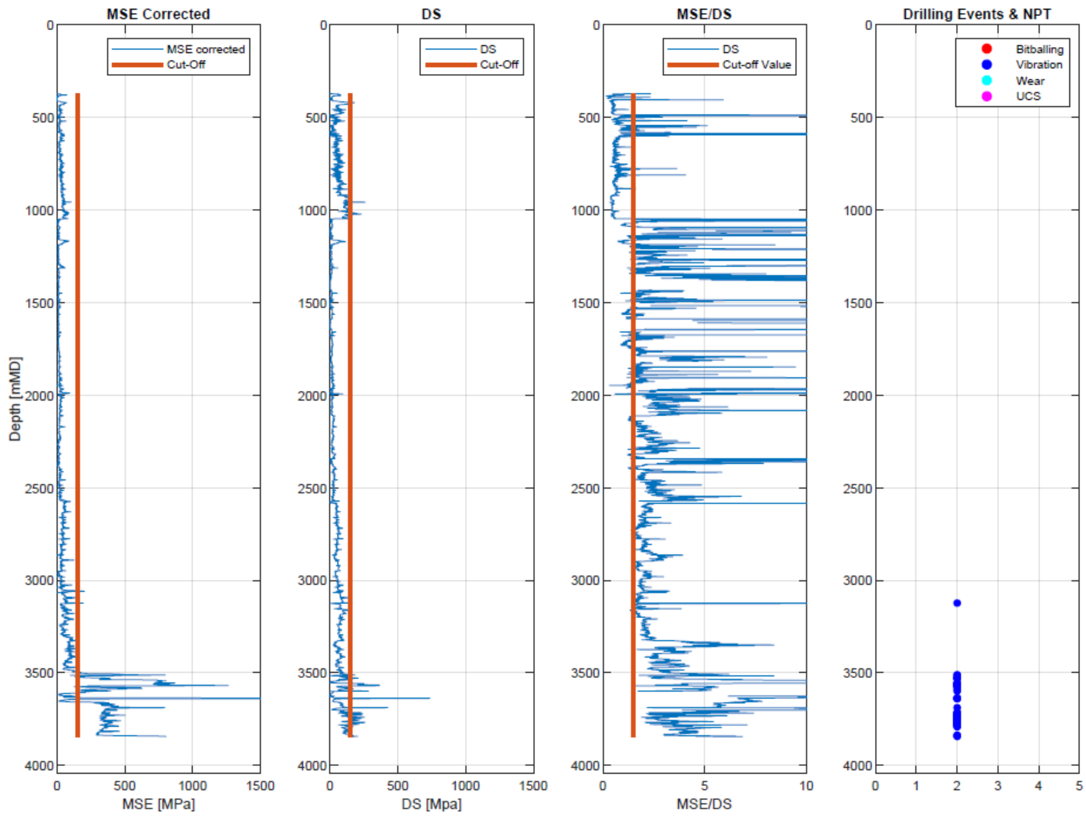


**Figure 60. MSE Corrected, DS, MSE/DS and Drilling Events detection, well 34/4-15 A. Note the increase in MSE and DS from 3500 to 3650 mMD where bit-bouncing was reported.**

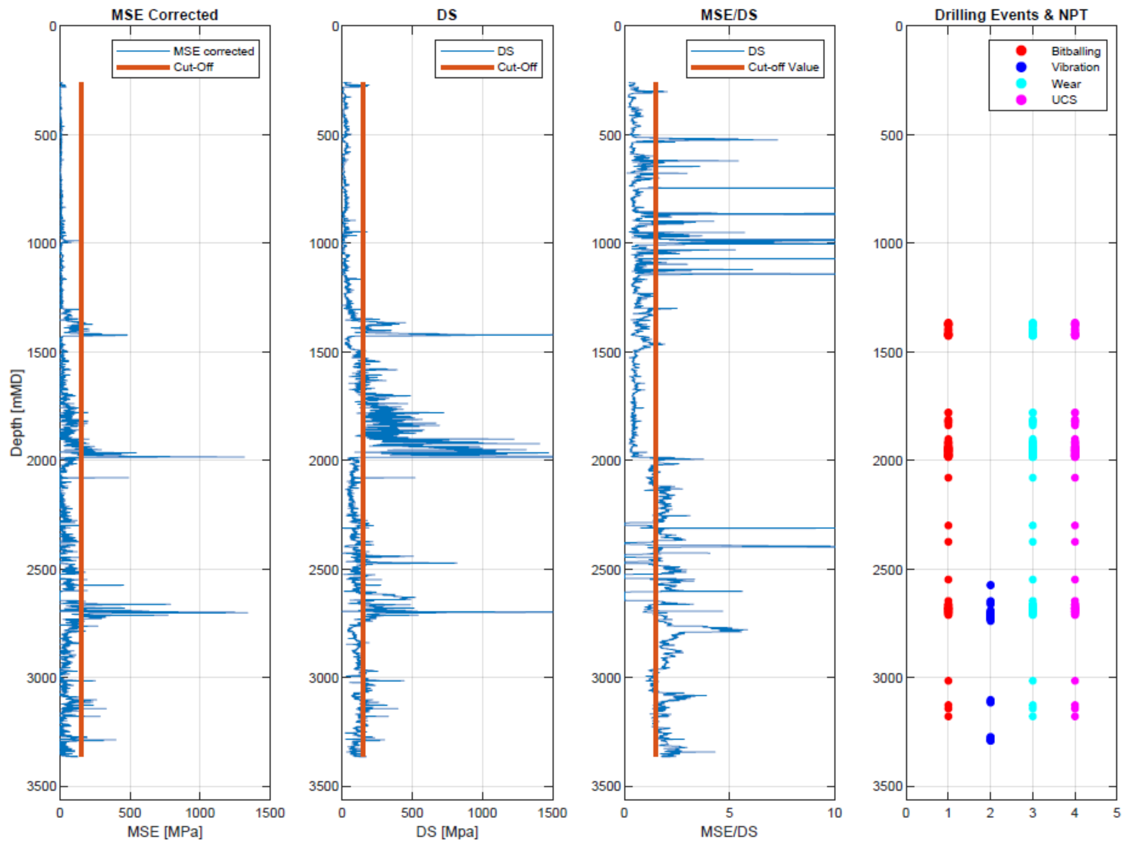


**Figure 61. MSE Corrected, DS, MSE/DS and Drilling Events detection, well 35/3-6. Note the increase in MSE and DS at 1400 and 2000 mMD where bit balling was reported.**

*4.1.5 Estimate CCS and Update Drilling Events and NPT*

To differentiate hard rock (UCS) from the other detected events, UCS and CCS values were estimated. As the MSE value should be numerically close to the estimated CCS value during efficient drilling, the same cut-off value of 150 MPa were utilized. **Figure 62**, **Figure 63** and **Figure 64** illustrates how the detected drilling events from chapter *4.1.4 Detect Drilling Events and NPT from Depth-based Drilling Mechanics* are updated if a previously detected hard rock (UCS) exceed the cut-off value for CCS of 150 MPa for the same depth. As the previously detected hard rock (UCS) never exceeded the cut-off value of 150 MPa, the UCS detection was removed from the drilling events detection for all three wells.

For well 34/4-15 S, a large increase in UCS and CCS values were observed in the reservoir section at approx. 3500 mMD, and both the UCS and the CCS values exceeded the cut-off value of 150 MPa. However, as the rock hardness increased, the MSE value decreased, indicating more efficient drilling for harder rock. The same trend was observed for well 34/4-15 A. For well 35/3-6, the cut-off value was never exceeded, but also for this well the MSE decreased with the increase in UCS and CCS at approx. 2700 – 3000 mMD. This trend is visualized in the MSE – Rock Hardness figures in *8.2 Appendix B* (**Figure 93**, **Figure 94** and **Figure 95**).

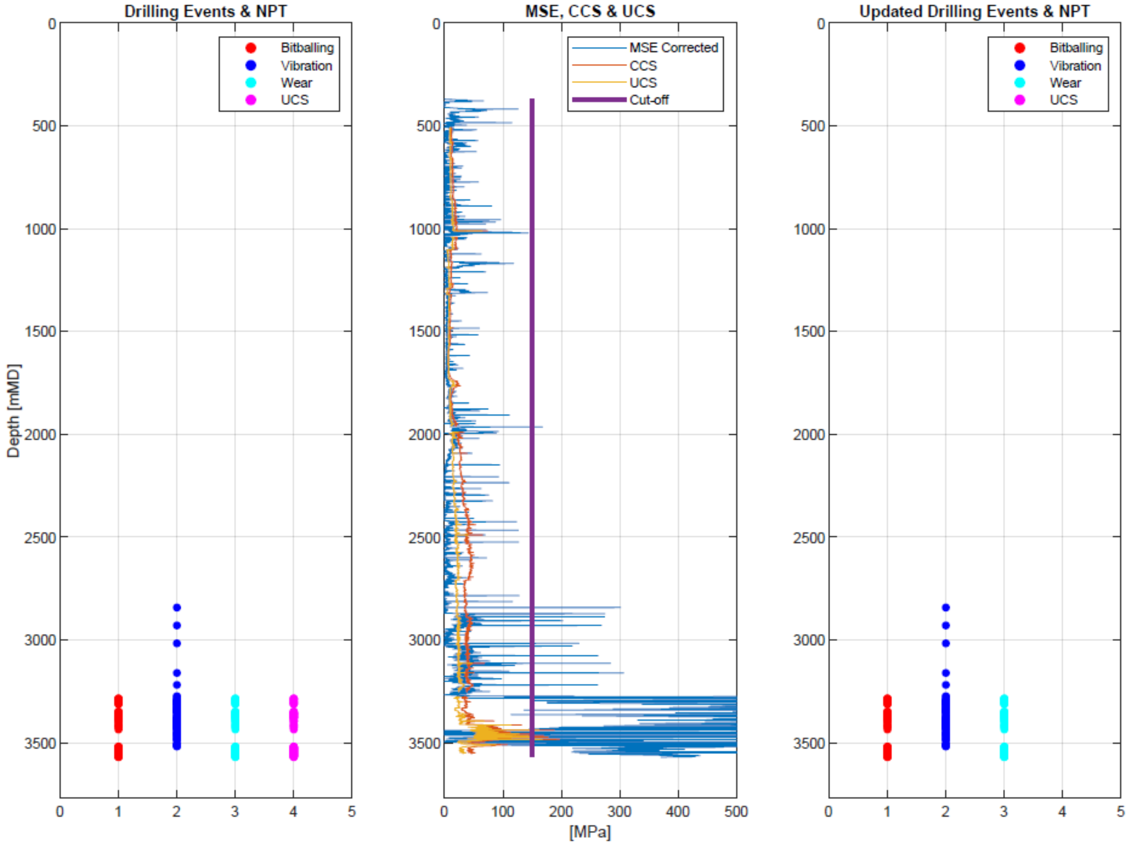


**Figure 62. Drilling Events, MSE, CCS, UCS, cut-offs and updated Drilling Events, well 34/4-15 S. Note the removal of the UCS detection.**

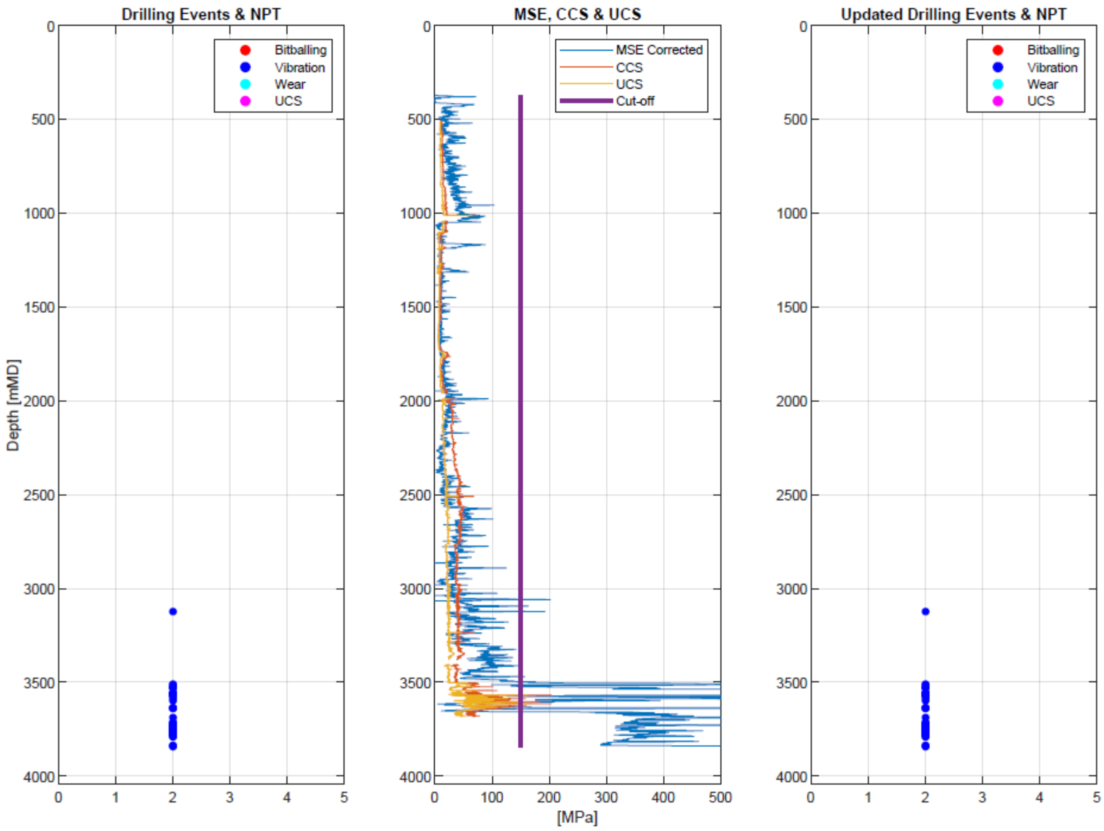


**Figure 63. Drilling Events, MSE, CCS, UCS, cut-offs and updated Drilling Events, well 34/4-15 A.**

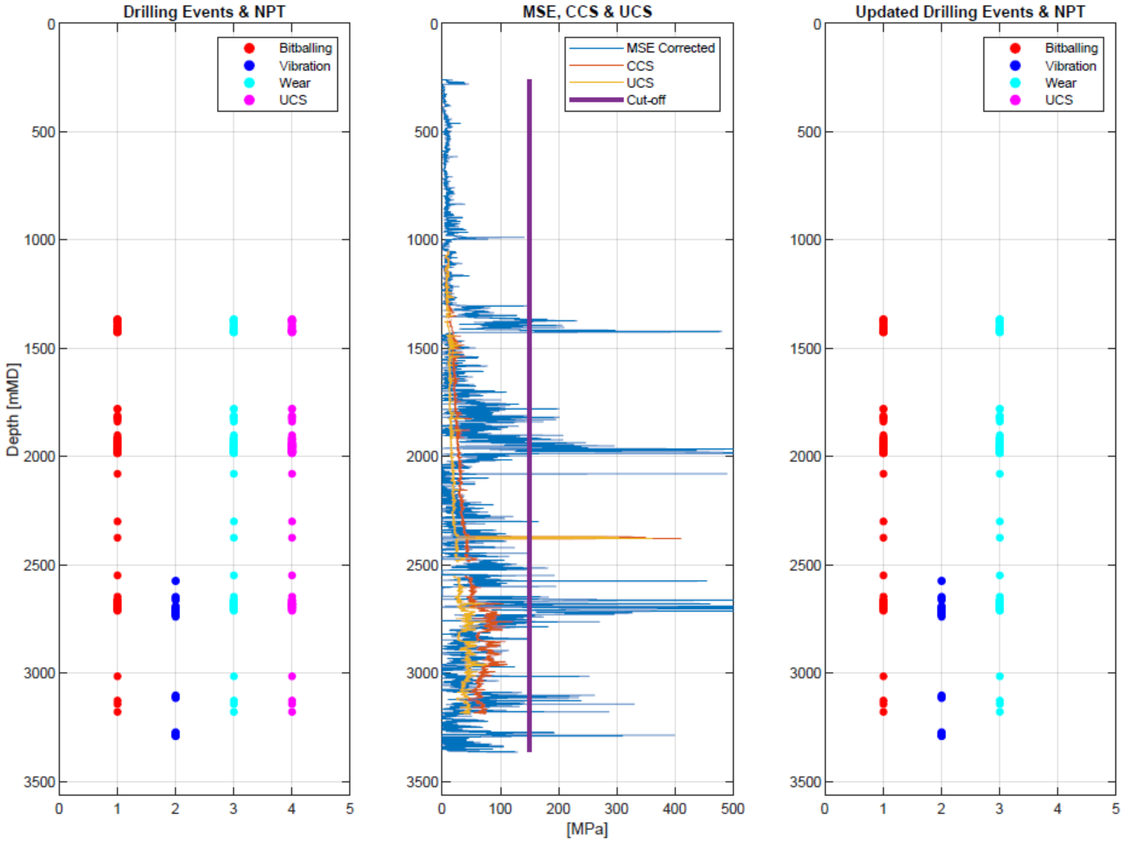


**Figure 64. Drilling Events, MSE, CCS, UCS, cut-offs and updated Drilling Events, well 35/3-6. Note the removal of the UCS detection.**

### *4.1.6 Final Drilling Events and NPT plot*

As the observed large increases in CCS values were followed by a similar increase in UCS values, it was decided to use UCS estimations directly to detect hard rock. As the effect of confined pressure is removed, a lower cut-off value of 100 MPa was utilized. Hard rock was consequently detected for well 34/4-15 S and A in the reservoir section.

Further, the detected drilling events were compared to the lithology of the well (**Figure 96**, **Figure 97** and **Figure 98** in *8.2 Appendix B*). It was observed that the reported bit balling occurred in claystone formations. Boundaries for bit balling were created, detecting bit balling only in the presence of claystone, hence enabling the differentiation of bit balling and bit wear.

The final drilling events detection procedure is shown in **Figure 65**, **Figure 66** and **Figure 67**. The detected events gave a good match with the reported events for all three wells.

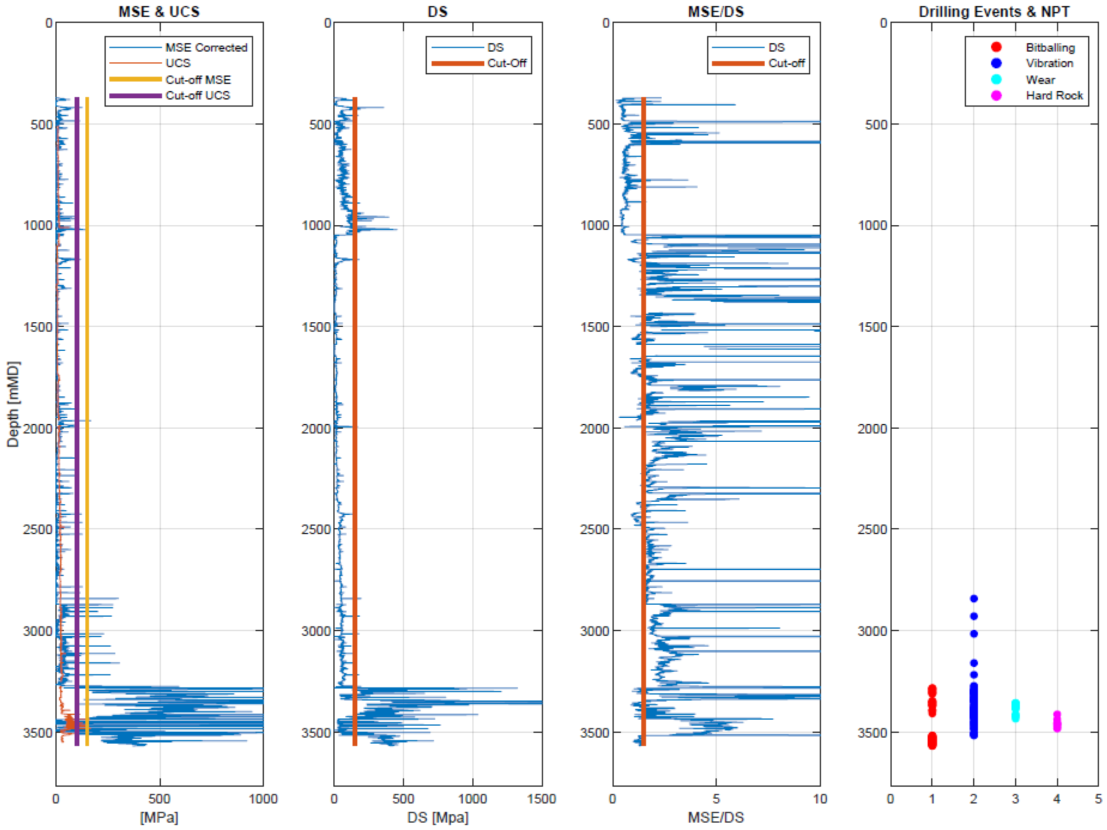


**Figure 65. MSE, UCS, DS, MSE/DS, cut-offs and final Drilling Events detection, well 34/4-15 S. Note the differentiation of Bit Balling and Bit Wear based on lithology and the detection of Hard Rock from estimated UCS and cut-off value.**

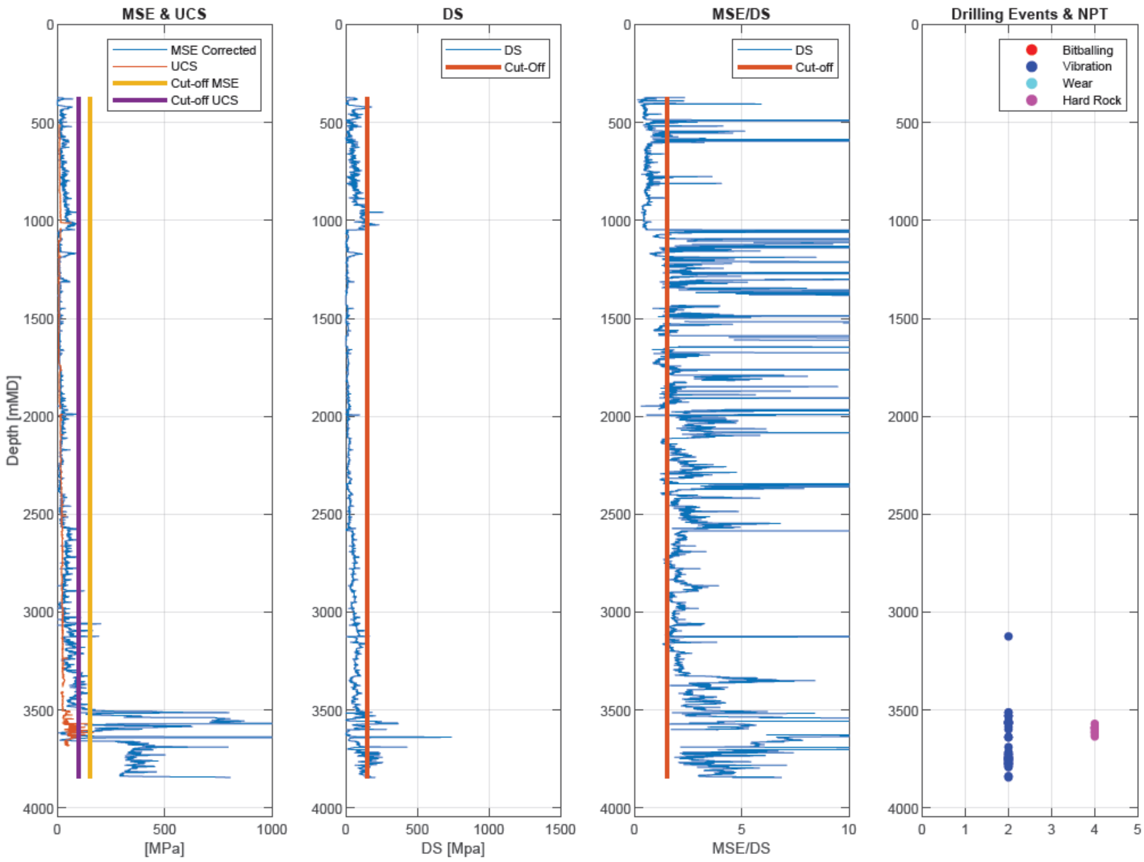


**Figure 66. MSE, UCS, DS, MSE/DS, cut-offs and final Drilling Events detection, well 34/4-15 A. Note the detection of Hard Rock from estimated UCS and cut-off value.**

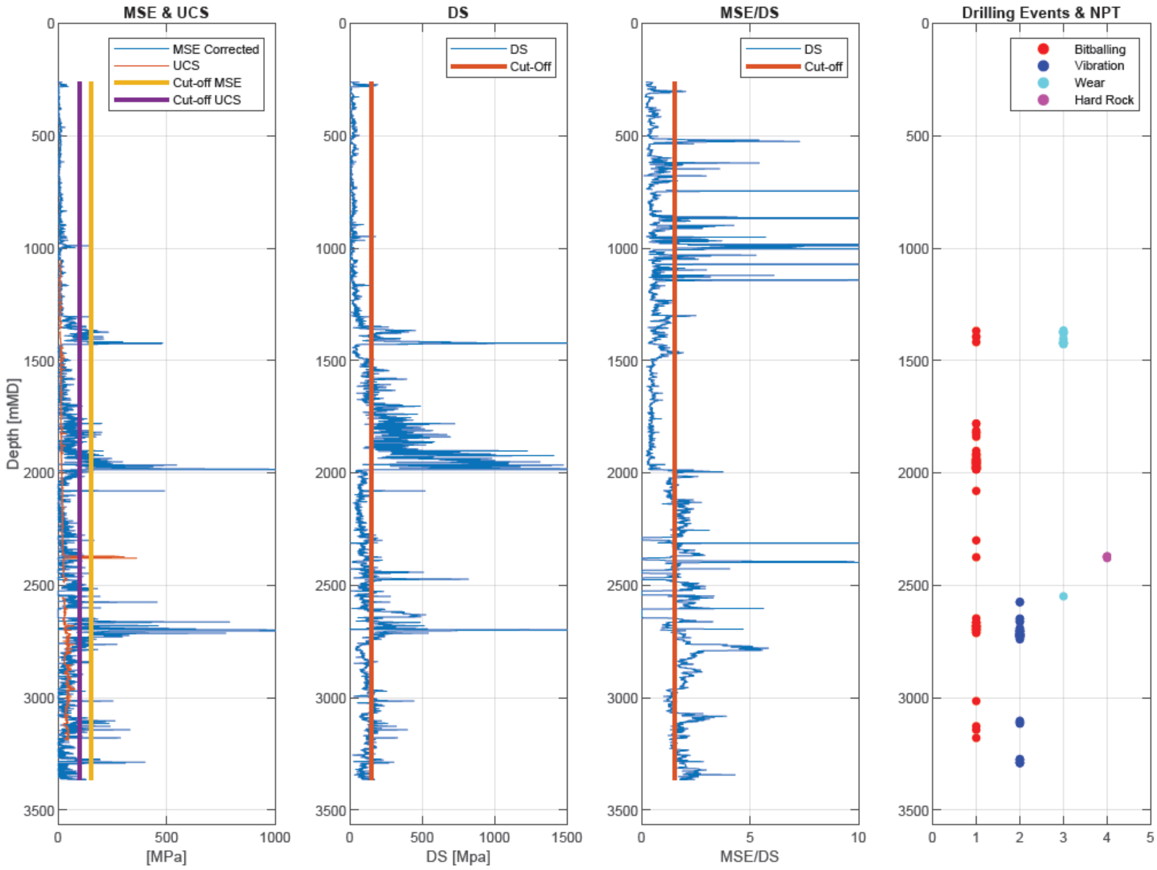


**Figure 67. MSE, UCS, DS, MSE/DS, cut-offs and final Drilling Events detection, well 35/3-6. Note the differentiation of Bit Balling and Bit Wear based on lithology and the detection of Hard Rock from estimated UCS and cut-off value.**

### *4.1.7 Estimate Lost Time*

To estimate the lost drilling time, the WOB-method was used. For every hole section: WOB was plotted against ROP, efficient bit-lines were created and efficient and non-efficient depths differentiated (**Figure 99** through **Figure 104** in *8.2 Appendix B*). It was observed that for a depth with a detected event, or non-efficient depth, the penetration rate would be lower for the same WOB compared to the efficient depths, increasing the confirmation in the drilling events detection workflow.

Further on, the new ROPs were estimated and concatenated (**Figure 68**, **Figure 69** and **Figure 70**). Well 34/4-15 S and A experienced efficient drilling in the overburden, achieving ROPs exceeding 100 m/hr. During drilling of the 8 ½” hole section, inefficient drilling due to bit balling and vibrations were experienced for the two wells. Approx. 24 hours and 4 hours drilling time were estimated lost for the two wells, with the greatest loss in time due to bit balling in well 34/4-15 S at 3300 – 3400 mMD. However, no extra bit changes were needed due to the events, thus no time was lost due to excessive tripping, yielding a high overall efficiency for the two wells.

Well 35/3-6 experienced several segments in the 12 ¼" hole section with inefficient drilling, leading to an estimated lost drilling time of 16 hours, with the majority of the lost time experienced at 1900 - 2000 mMD. It was also observed a bit change at this depth, which was reported to be due to bit balling. Studying Figure 103 and Figure 104 in *8.2 Appendix B*, a high WOB was observed with a consequent very low ROP at the depth prior to the bit-change, leading to the necessity for the bit change. Another bit change occurred at approx. 2400 mMD. An estimated 17 hours were lost due to the excess tripping, assuming a tripping speed of 500 m/hr, resulting in a total of approx. 33 hours lost.

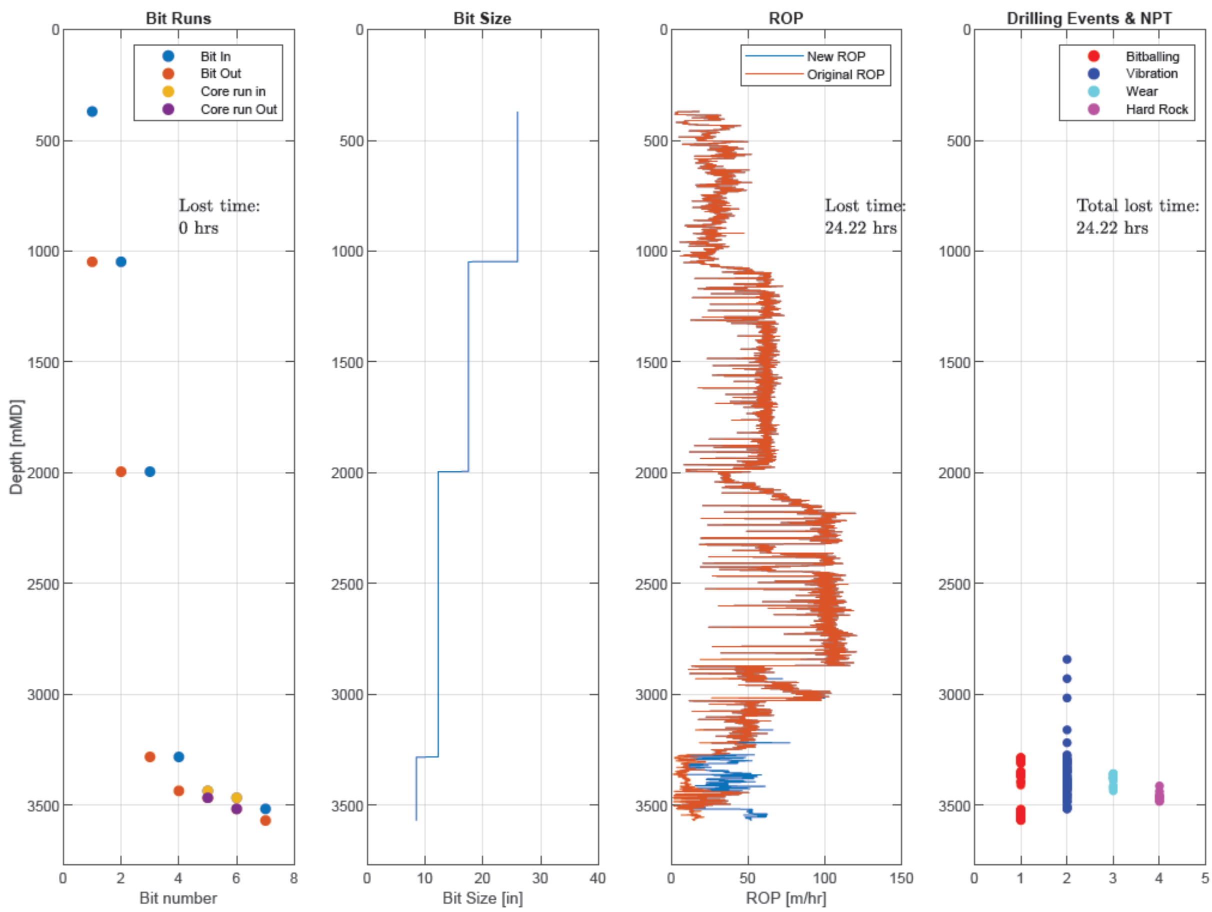


**Figure 68. Bit runs, bit size, new and original ROP, Drilling Events and lost time, well 34/4-15 S.**

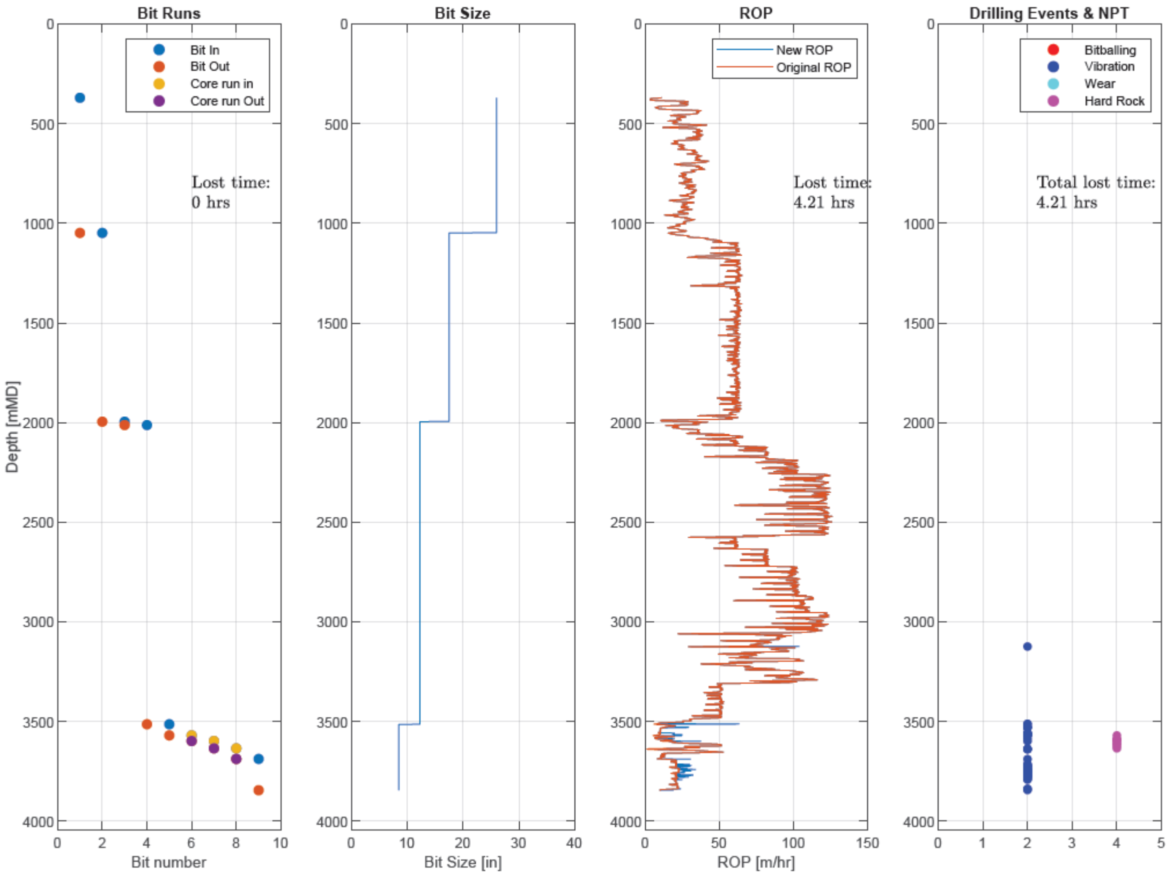


**Figure 69. Bit runs, bit size, new and original ROP, Drilling Events and lost time, well 34/4-15 A.**

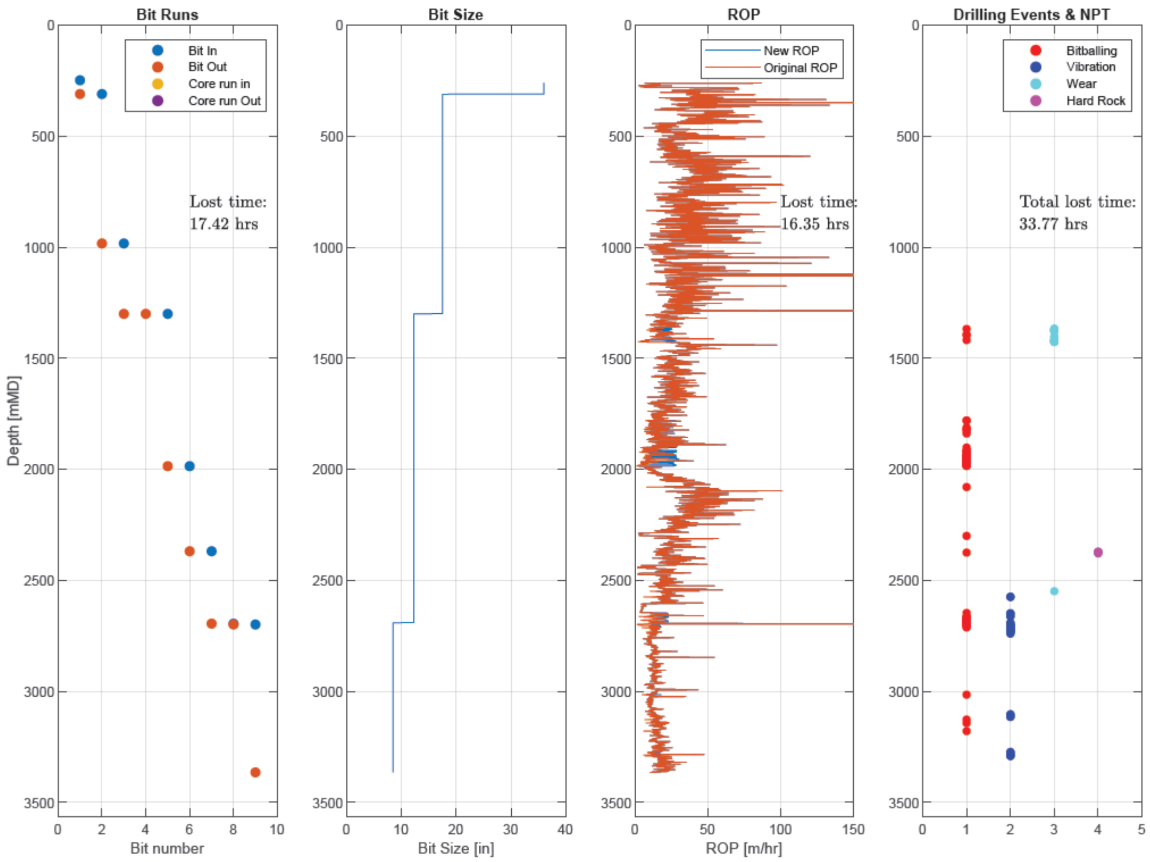


**Figure 70. Bit runs, bit size, new and original ROP, Drilling Events and lost time, well 35/3-6. Note the two bit changes at approx. 2000 and 2400 mMD.**

The Torque-method for lost time estimation was also used on the three wells, yielding similar results to the WOB-method. Estimation of lost time was 19, 4 and 15 hours of drilling time for wells 34/4-15 S, 34/4-15 A and 35/3-6 respectively. **Figure 105** through **Figure 110** in *8.2 Appendix B* illustrates the WOB-Torque cross plots. A near linear relationship was observed between WOB and Torque for well 34/4-15 S and A in the 26”, 17 ½” and 12 ¼” hole sections. For the 8 ½” hole sections, where drilling inefficiencies were detected, larger deviations were observed.

A linear relationship between WOB and Torque was also observed for well 35/3-6, but with larger deviation compared to the other two wells. The greatest deviation was observed in the 12 ¼” hole section. The 12 ¼” hole section experienced bit balling, explaining the large deviation in this hole section. As a result of the bit balling, two bit changes were required. As different bit-design vary in bit aggressiveness, different WOB-Torque profiles should be expected, further contributing to the large deviation in the Torque-WOB cross plot for the 12 ¼” hole section.

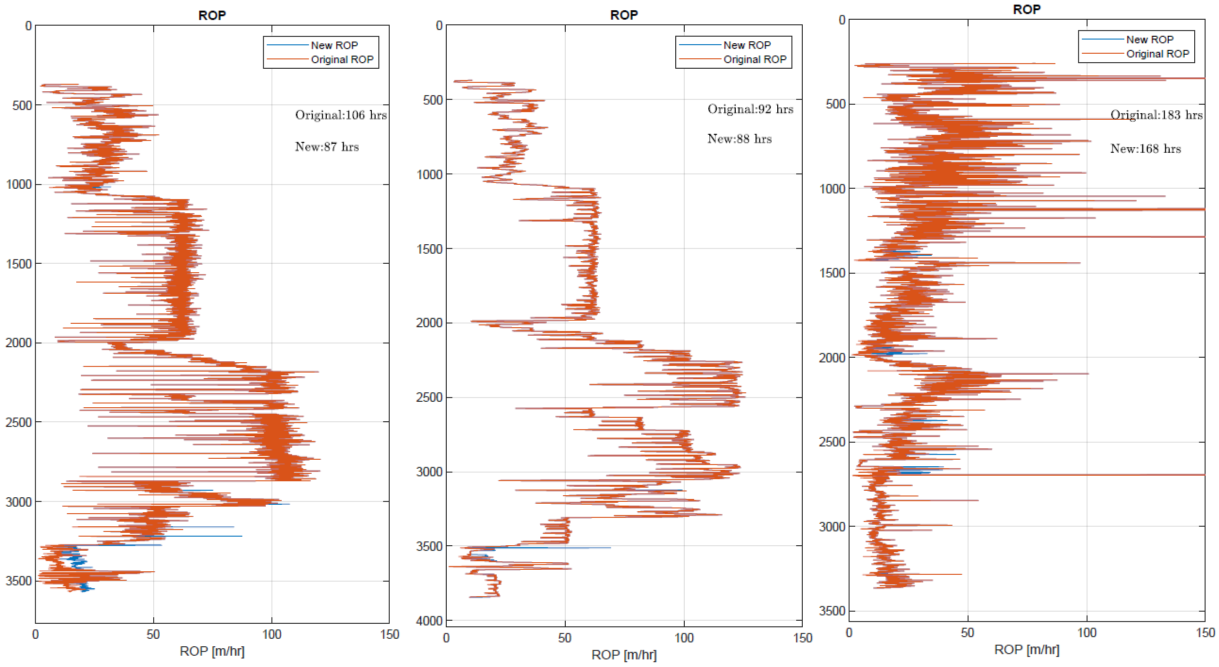


**Figure 71. Original and New ROP Torque-Method, well 34/4-15 S, 34/4-15 A and 35/3-6.**

*4.1.8 Estimate Costs due to Drilling Events and NPT*

**Figure 72** shows how many meters of drilling events were detected in each well, with the associated estimated costs shown in **Figure 73**. For well 34/4-15 S, similar lengths of vibrations and bit balling were detected, however the estimated costs associated with vibrations are double. The length of detected vibrations for well 34/4-15 A were similar to 34/4-15 S, yet the associated estimated costs are three times more for the S well. This may indicate that stick-slip vibrations, as was reported for well 34/4-15 S, reduces ROP more compared to bit-bouncing vibrations, as was reported for well 34/4-15 A.

Comparable lengths of bit balling were detected for well 35/3-6 as for well 34/4-15 S. However, well 35/3-6 needed two bit-changes to manage to break through the bit balling sections, resulting in much higher bit balling costs compared to well 34/4-15 S, where bit balling most likely diminished due to the change in lithology.

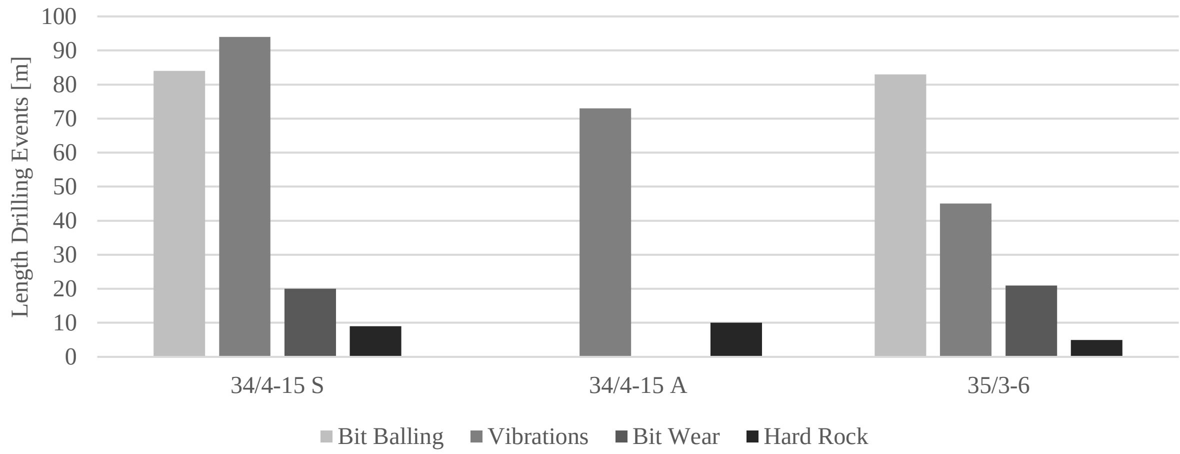


**Figure 72. Total meters of detected Drilling Events pr. well.**

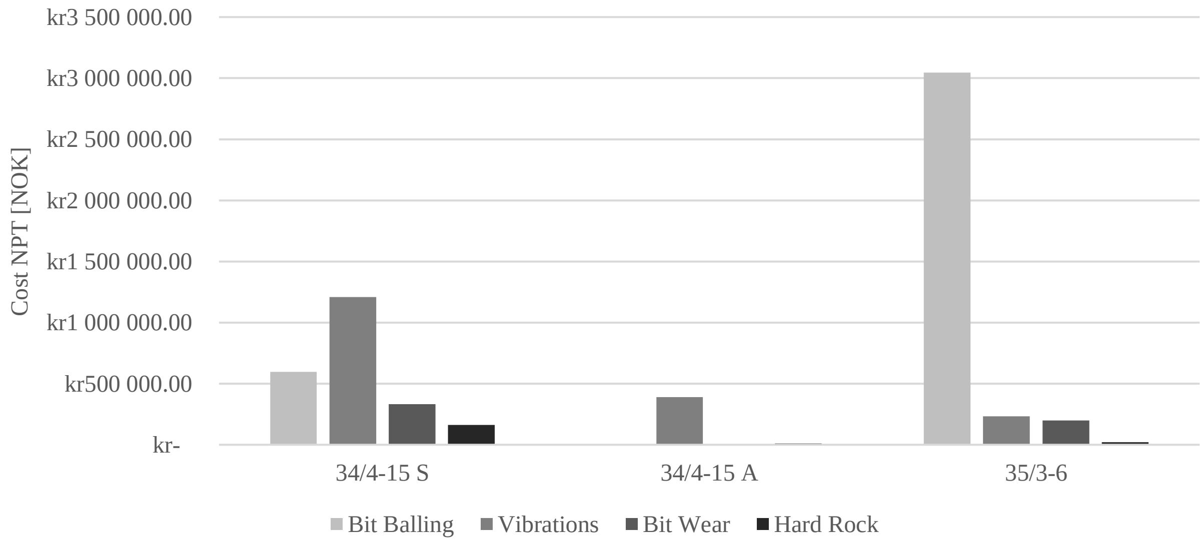


**Figure 73. Estimated cost detected Drilling Events pr. well.**

**Figure 74** separates the drilling time and the tripping time, illustrating how large percentage time was estimated lost due to drilling events. It was observed that for well 34/4-15 S and 34/4-15 A, approx. 12% and 2% of the total time were lost due to inefficient drilling, equal to 21% and 4% of the drilling time. For well 35/3-6, approx. 6% of the total time was lost due to inefficient drilling, equal to 9% of the drilling time. Additionally, 7% of the total time was lost due to unplanned bit changes, equal to 26% of the total tripping time. In total, the estimated associated costs due to drilling events and NPT for well 34/4-15 S, 34/4-15 A and 35/3-6 were 2.3, 0.4 and 3.5 mill NOK, respectively. Drilling cost were estimated to 600, 100 and 1000 NOK/meter.

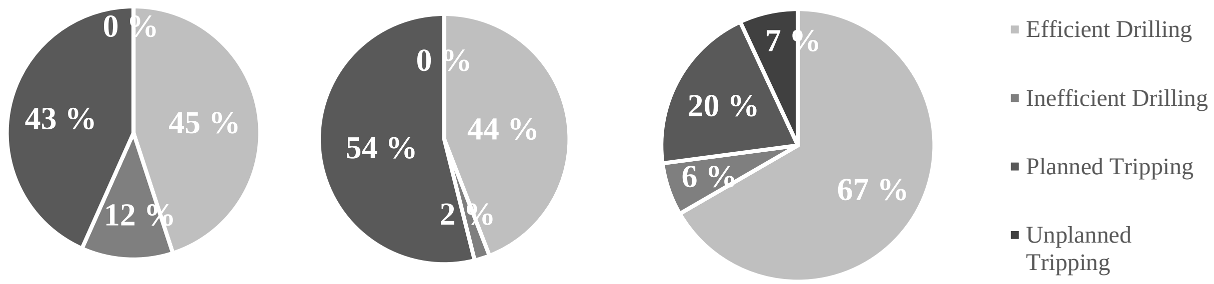


**Figure 74. Breakdown of drilling time and tripping time, well 34/4-15 S, 34/4-15 A and 35/3-6.**

### *4.2 Multi-well Analysis*

*4.2.1 Well Overview*

An overview of the 20 analysed wells are listed in **Table 11** and the results are presented in **Table 12** through **Table 17** in *8.1 Appendix A*. The analysed wells include wells from year 1997 to 2020 and range in number of drilling days from 15 to 180 days. Vertical depths ranges from 1920 m to 5438 m, and drilling efficiencies from 29 to 189 meters/day, providing a good diversity to test the analysis. Six wells were drilled with water-based drilling fluid, with the remaining 14 drilled with oil-based drilling fluids. Four of the wells are classified as HPHT wells. **Figure 75** presents an overview of the area of interest.

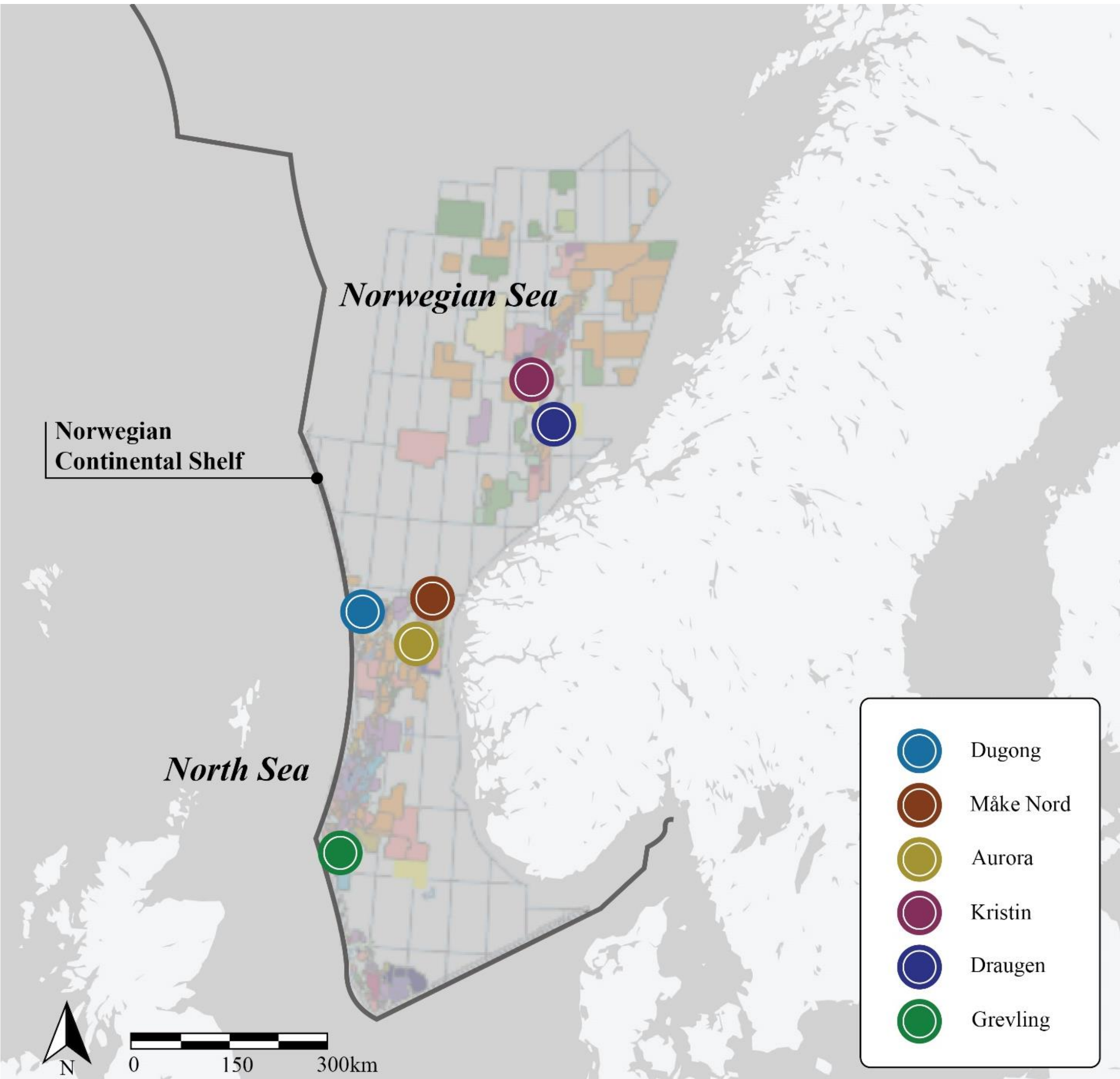


**Figure 75. Map of the area of interest.**

*4.2.2 Drilling Events Detection*

**Figure 76** illustrates the total meters of detected drilling events and the percentage of each event. From the 75 464 meters analysed, a total of 8640 meters of drilling events were detected, with the average drilling events detection being 432 meters. Vibrations were the most detected drilling event, constituting 39% of the detected events with 3400 meters detected. 2460 meters of bit balling was detected, with bit wear at 2315 meters detected and hard rock at 464 meters detected. The results corresponds with the theory from chapter *2.1.2 Vibrations*, where vibrations are stated as the most common form of drilling inefficiency (IADC, 2015). The drilling events detection and lost time estimations for the analysed wells, using both the WOB-method and the Torque-method, are shown in **Figure 111** through **Figure 161** in *8.3 Appendix C*.

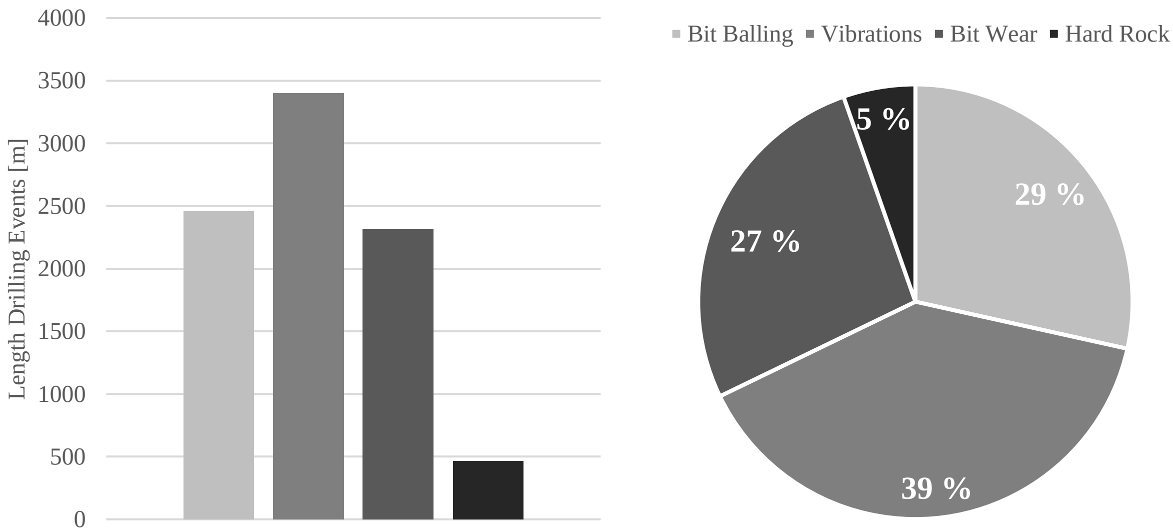


**Figure 76. Total meters of Drilling Events detected for all 20 wells and the percentage of each Drilling Event.**

**Figure 77** illustrates how many meters of drilling events were detected for each analysed well, as well as the vertical depth of the wells. Drilling events were detected for all the analysed wells, but the events and the severity of the events varied. A positive correlation between vertical depth and drilling events detection was observed.

As a deeper well could potentially detect more drilling events simply due to the more meters drilled, a comparison of meters detected pr vertical km is preferred (**Figure 78**). The three deepest wells analysed, with vertical depths exceeding 5000 meters, were the wells most prone to drilling events detection, indicating vertical depth may be a factor influencing drilling events. Increased pressure, temperature and compaction with depth can also be potential causes. Age could also be a factor, as the three wells were drilled in 1997, 1998 and 2001 for well 6406/2-5, 6506/11-6 and 6406/1-1, among the oldest analysed.

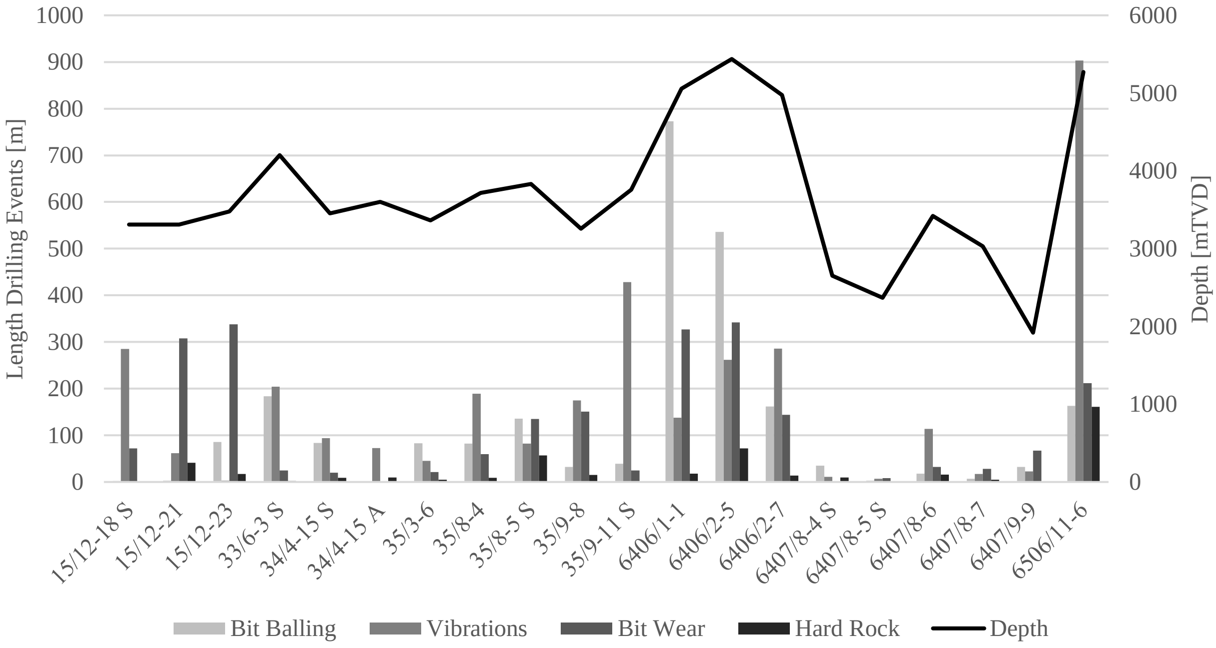


**Figure 77. Drilling Events detected and vertical depth. Note the correlation between vertical depth and detected Drilling Events.**

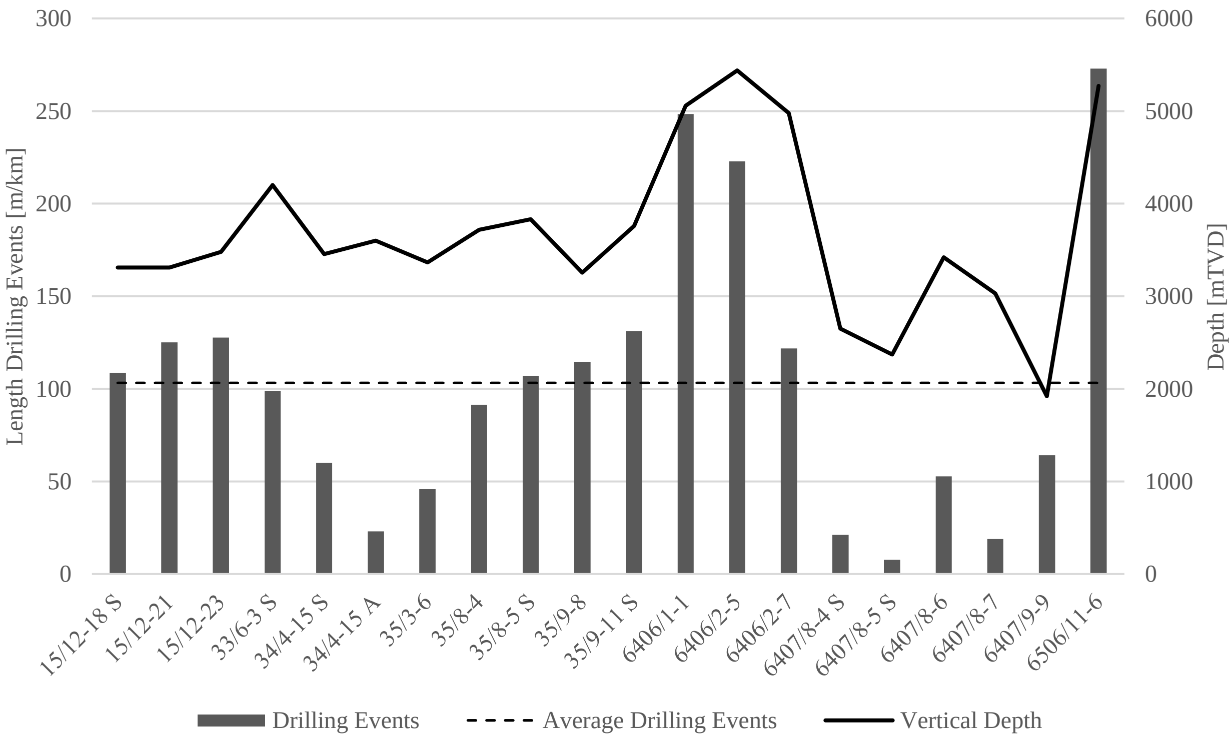


**Figure 78. Drilling Events detection in m/km and vertical depth.**

### *4.2.3 Lost Time*

The lost drilling time was estimated with two methods: the WOB-method and the Torque-method. The results from the two methods are presented in Table 12 and Table 13, and an average of the two is presented in Table 14 in *8.1 Appendix A*. On average, 24% of the drilling time was lost due to drilling events, with bit balling, vibrations and bit wear resulting in almost equal reduction in ROP (**Figure 79**).

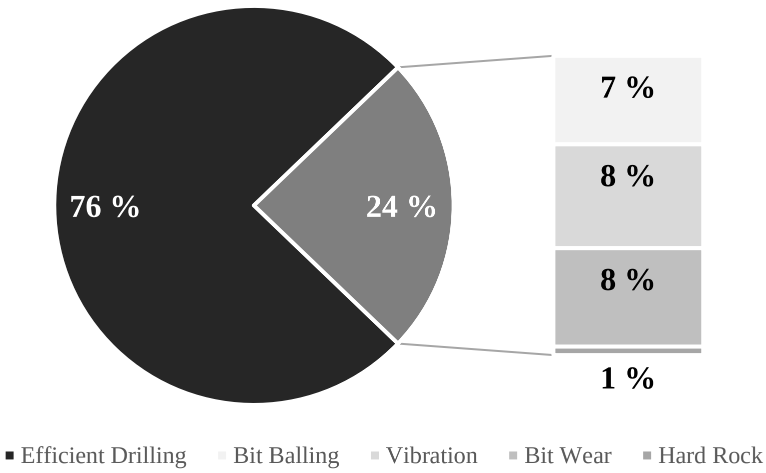


**Figure 79. Average lost drilling time with associated events.**

The estimated lost tripping time is presented in Table 15 in *8.1 Appendix A*. On average, 7% of the total tripping time was lost due to unplanned trip changes from drilling events, with bit balling being the major contributor to unplanned bit changes (**Figure 80**). However, the deviation in lost tripping time was observed to be large, with a standard deviation of 10.5%, or 12 hours. For 11 of the wells, no excess tripping was observed, while the greatest percentage loss in tripping time was estimated for well 33/6-3 S, where bit balling resulted in 38% lost tripping time, equal to an estimated 34 hours lost. The greatest time loss was estimated for well 6506/11-6, where 42 hours were lost due to bit balling. However, due to the great depth of the well, exceeding 5000 mTVD, and a high amount of bit changes, including 10 core runs, a total tripping time of 441 hours were estimated for this well, thus only resulting in an estimated 9% loss of tripping time due to bit balling.

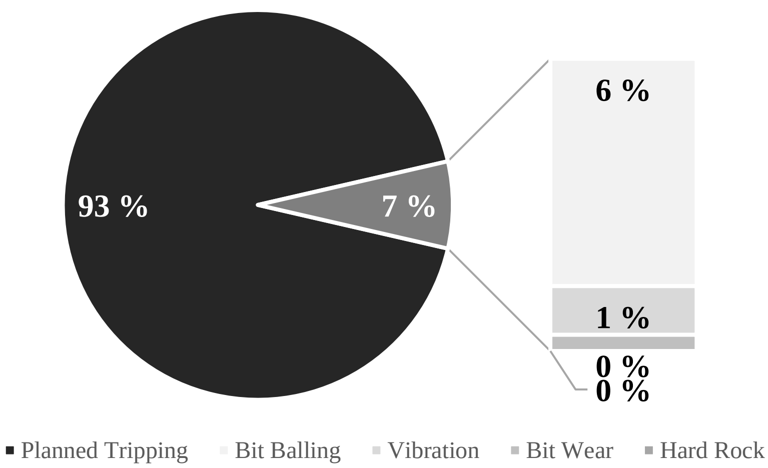


**Figure 80. Average lost tripping time with associated events.**

Summing up the average drilling time and average tripping time (**Figure 81**), it was observed that inefficient drilling was the largest contributor to non-productive time. 15% of the total time was estimated lost due to inefficient drilling and 3% was estimated lost due to unplanned tripping, resulting in a total of 18% lost time due to drilling events, equal to an estimated average of 69 hours lost.

On average, the number of drilling days listed on the NPD Factpages are 59 days, which includes setting of casings, cementing, plugging and more. The estimated 69 hours lost time due to drilling events equates to approx. 5% lost rig-time.

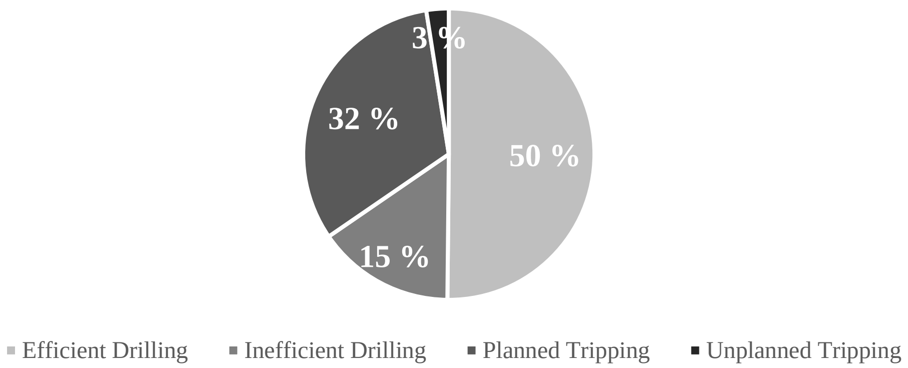


**Figure 81. Breakdown of average drilling time and average tripping time.**

*4.2.4 Estimated Costs*

The total contribution from each drilling event to non-productive time is listed in Table 16 in *8.1 Appendix A*. On average, the main contributor to NPT is bit balling, with an estimated 44% of the total lost time from lost ROP and lost tripping. Vibrations and bit wear followed at 25% and 28%, with hard rock at 3%. The associated estimated average costs for each drilling event was 3.2, 1.8, 2.0 and 0.2 mill NOK for bit balling, vibrations, bit wear and hard rock, respectively (**Figure 82**).

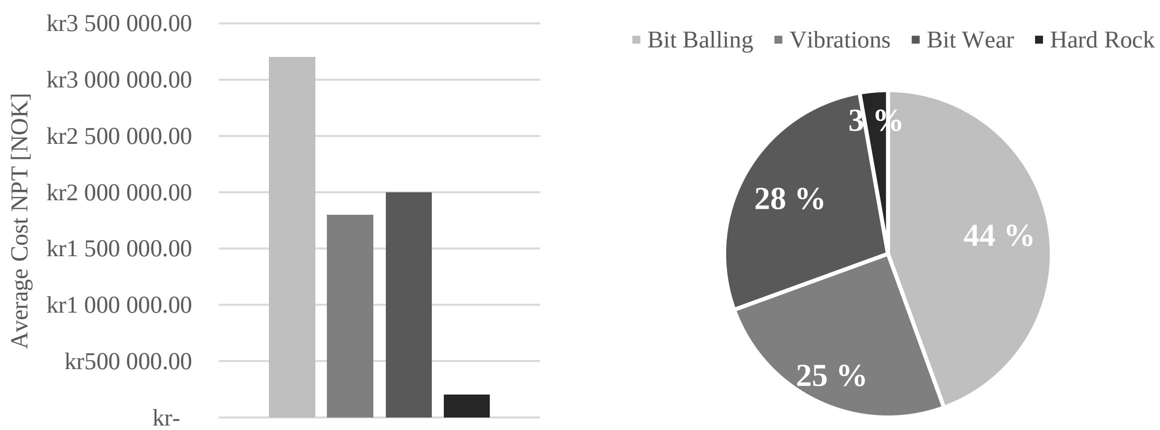


**Figure 82. Drilling Events contribution to NPT and associated average cost.**

**Figure 83** illustrates the estimated cost of NPT and the drilling efficiency for each well. Drilling efficiency is for the purpose of this thesis defined as the measured depth of the well in meters divided by the total number of drilling days listed on the NPD Factpages. Drilling efficiency is thus an independent calculation of the developed MATLAB Workflow.

A strong negative correlation was observed between the estimated cost of NPT and the drilling efficiency, increasing the confidence in the developed drilling events detection and lost time estimations from the MATLAB Workflow. As the estimated cost is a direct function of the estimated lost time, and therefore a function of the drilling events detection, a conclusion may be drawn as to increased drilling events results in a lower drilling efficiency.

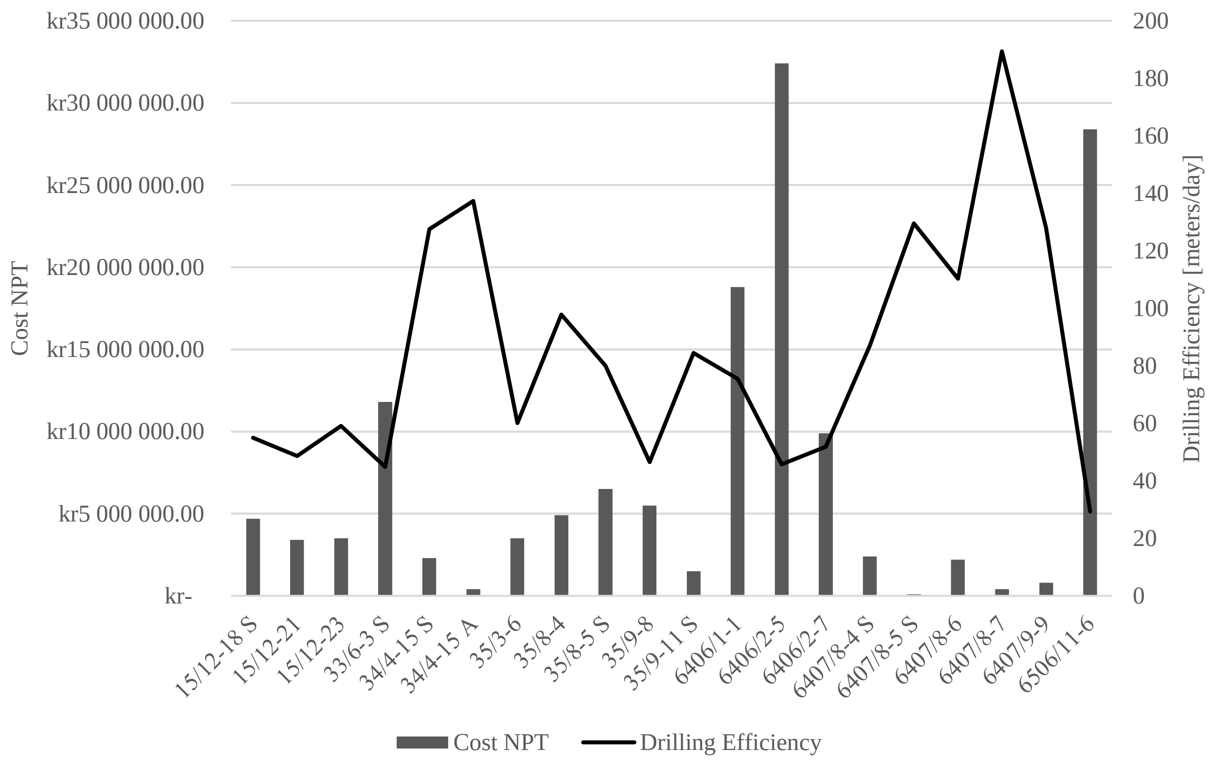


**Figure 83. Drilling efficiency and cost NPT. Note the negative correlation between drilling efficiency and cost NPT.**

*4.2.5 Other Observations*

**Figure 84** illustrates the detected drilling events for each well and compares it to the drilling fluid used. On average, more drilling events were detected in wells drilled with oil-based drilling fluid. However, the four wells categorized as HPHT wells were all drilled with oil-based drilling fluids: well 6406/1-1, 6406/2-5, 6406/2-7 and 6506/11-6. Excluding these wells, the average value for OBM was marginally lower compared to WBM, at 73 m/km detected for OBM and 78 m/km meters detected for WBM. No significant difference was therefore observed between OBM and WBM in the analysed wells with respect to drilling events detection. For HPHT wells, the average value was 216 m/km detected, almost 3 times higher than the other wells.

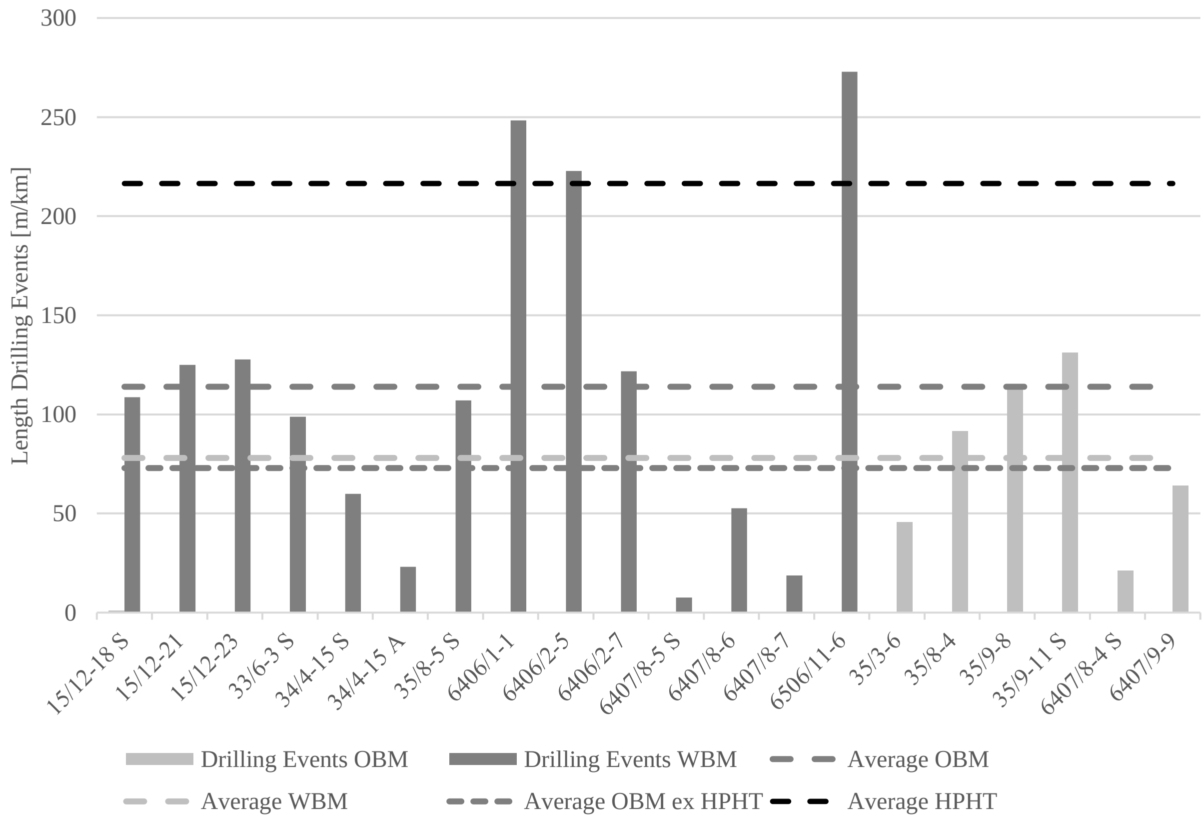


**Figure 84. Comparing Drilling Events detection per km to OBM, WBM and HPHT wells. The lines illustrate the average detected Drilling Events for OBM, WBM, OBM without HPHT wells and average for HPHT wells.**

*4.2.6 Summary Results*

The main findings from the analysis of the 20 wells from the North Sea and Norwegian Sea are summarized in **Figure 85**.

| | | |
|---|---|---|
| **Detected Drilling Events** | **103 m/km** | Average detected Drilling Events |
| | **41 m/km** | Average detected Vibrations |
| | **29 m/km** | Average detected Bit Wear |
| | **28 m/km** | Average detected Bit Balling |
| | **5 m/km** | Average detected Hard Rock |
| **Lost Time** | **24%** | Average lost drilling time |
| | **7%** | Average lost tripping time |
| | **5%** | Average lost rig-time |
| **Estimated Costs** | **7.2 mill. NOK** | Average estimated costs |
| | **3.2 mill. NOK** | Average estimated costs Bit Balling |
| | **2.0 mill. NOK** | Average estimated costs Bit Wear |
| | **1.8 mill. NOK** | Average estimated costs Vibrations |
| | **0.2 mill. NOK** | Average estimated costs Hard Rock |
| **HPHT** | **3x** | Three times more Drilling Events detected |

**Figure 85. Average per well of the main findings of the multi-well analysis of the 20 wells on the NCS.**

# 5 DISCUSSION

### *5.1 NPT & Drilling Events Detection*

The results obtained indicates that the developed workflow effectively manages to detect drilling events. The detected drilling events from the case study match well with the reported events from the three case study wells. The high negative correlation between the independently calculated drilling efficiency and the estimated costs for NPT from Figure 83 further increases the confidence in the developed workflow.

Through the development of the two ROP estimations, the WOB-method and the Torque-method, this thesis supports the theory of linearity between ROP-WOB and WOB-Torque for efficient drilling. However, as multiple factors influence these relationships, the relationships are shown not to be perfect. These imperfect relationships limit the accuracy of the estimated lost drilling time; however, it further emphasizes the importance of the developed drilling events detection using mechanical specific energy and drilling strength models instead of solely relying on these relationships.

**Figure 86** illustrates an approximation-method where inserting upper and lower bounds to the efficient bit-lines in the WOB-ROP and WOB-Torque cross plots for well 34/4-15 S, allowing efficient and inefficient depths to be detected. However, as the figure shows, some inefficient depths from the 8½" hole section could not be detected. Furthermore, a great amount of depths in the 12 ¼" hole section would have been marked as inefficient, a hole section which was mostly drilled with ROPs exceeding 100 m/hr.

The 8 ½" hole section in Figure 86 was drilled in 4 runs, including two core runs, explaining some of the deviation observed as different bits may have different aggressiveness and thus different WOB-ROP and WOB-Torque profiles. The 12 ¼" hole section was drilled in one bit run and therefore a more linear relationship was expected. The WOB-Torque relationship is observed to be more linear compared to the WOB-ROP relationship, indicating that the developed Torque-method for lost time estimation may be the preferred method (**Figure 87**).

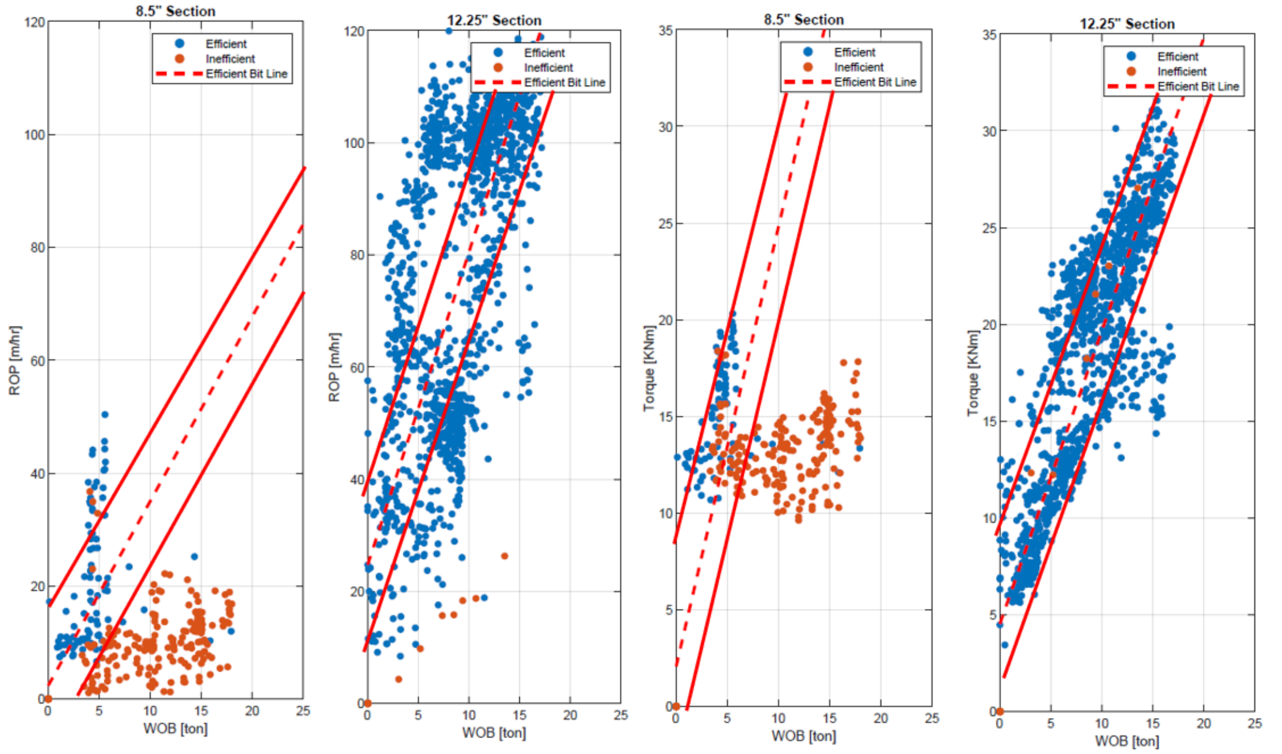


**Figure 86. WOB-ROP and WOB-Torque relationships for the 8 ½" and 12 ¼" hole sections, well 34/4-15 S. The red solid lines represent upper and lower bounds of the efficient bit line.**

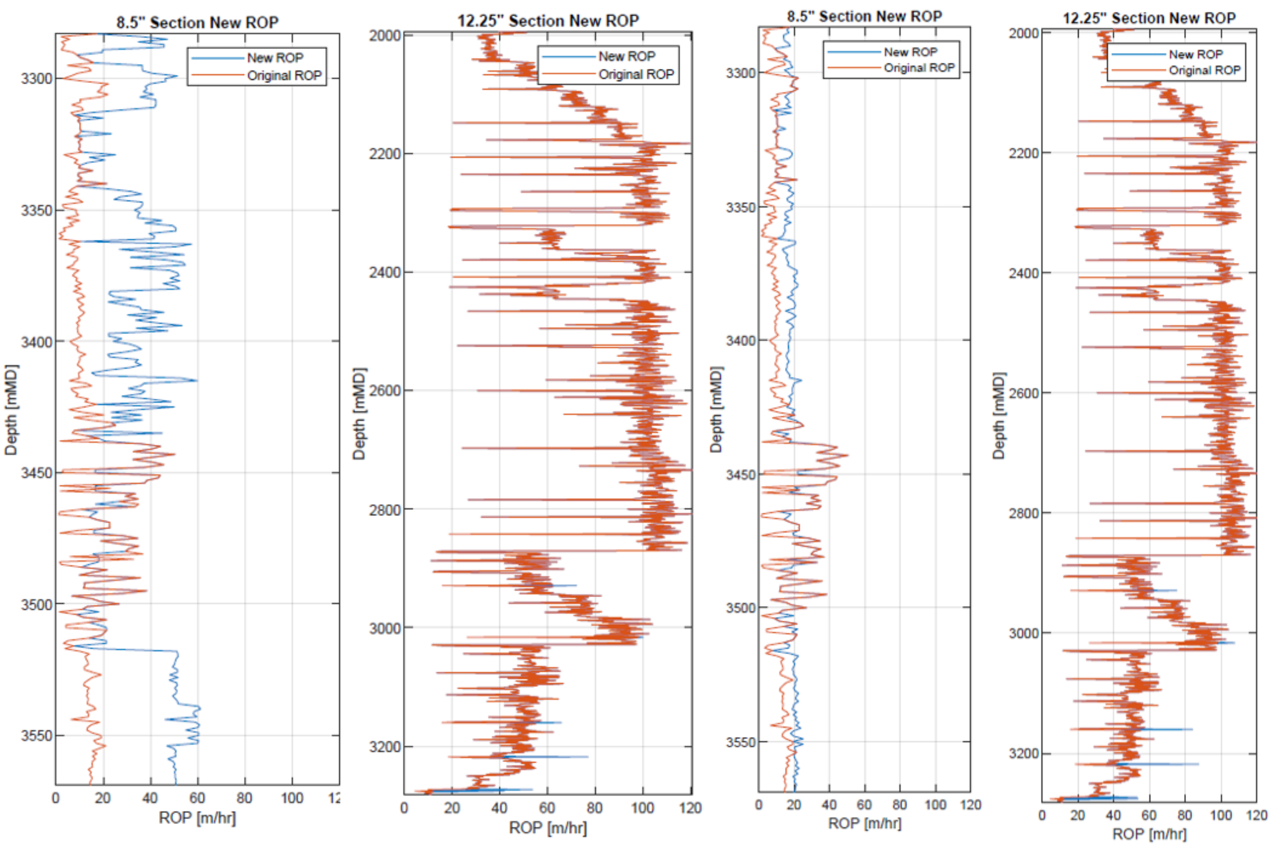


**Figure 87. WOB-method and Torque-method for lost time estimations for the 8 ½" and 12 ¼" hole sections, well 34/4-15 S. Note the more conservative and less fluctuating New ROP in the 8½" hole section for the Torque-method (right) compared to the WOB-method (left).**

It is observed from the results of the 20 analysed wells that the Torque-method estimates give a slightly more conservative lost time estimate compared to the WOB-method, however the results are generally very close. Figure 87 compares the WOB-method and the Torque-method for the 8 ½" and the 12 ¼" hole section for well 34/4-15 S. For the 8 ½" hole section, a more conservative and less fluctuating New ROP was estimated using the Torque-method compared to the WOB-method.

On average, there was an estimated 24% lost drilling time. However, due to the way the lost drilling time is estimated in this thesis, a high percentage loss in drilling time do not necessarily equal a low average ROP (**Figure 88**). This is due to a higher efficient ROP may return a higher lost time when the ROP is reduced due to inefficient drilling, as the estimated new ROP is a continuation of the previously efficient ROP.

For instance, wells 34/4-15 S and 35/3-6 estimated losses of 21% and 9% drilling time, yet well 34/4-15 S averaged 30 m/hr in ROP compared to 17 m/hr for well 35/3-6. It is thus important to study more than just the estimated lost drilling time to understand the performance of the drilled well.

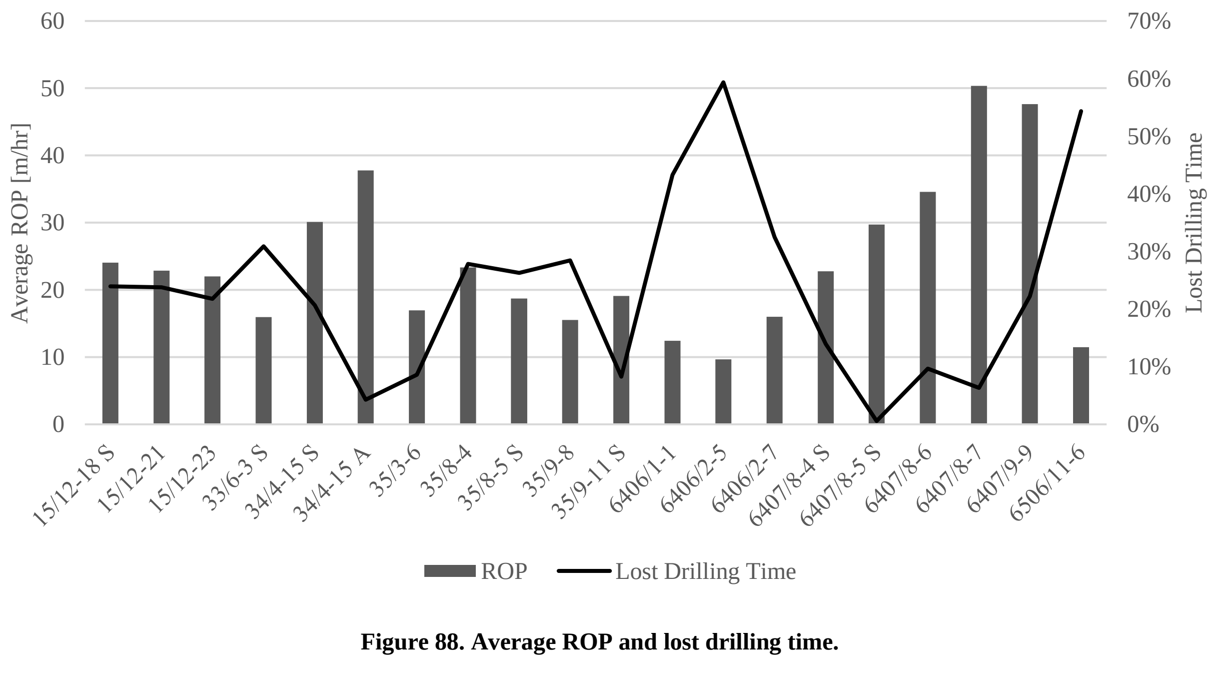


**Figure 88. Average ROP and lost drilling time.**

A major advantage to the developed ROP and lost time estimations is the ability to improve the understanding of the severity of a detected event. If the gap between the estimated new ROP and the original ROP is large, the severity of the event is most likely significant, and the event should be given more focus during the planning of this section in future wells. By comparing this gap to the bit changes, the severity of the event could be further understood. A large increase in lost time coupled with a bit change is a strong indicator of a severe event.

The results from the 20 analysed wells show that more time was lost due to low/inefficient ROP compared to unplanned bit changes, where 24% of the drilling time was estimated lost due to inefficient drilling and 7% of the tripping time was estimated lost due to unplanned trip changes, equal to 15% and 3% of the total time, respectively. This may be related to vibrations being the most common detected drilling event, where mitigations, explained in Chapter *2.1.2 Vibrations*, includes changing the rotary speed and/or weight on bit. Bit design is important when it comes to vibrations, but it does seem from the analysed wells, that it is preferred to alter the drilling practises before aiming for a bit change. Table 1 in the introduction lists the most mentioned drilling events on the NCS. Vibrations are only mentioned 43 times, or 3%. As vibrations are detected in every well analysed, with an average of 41 m detected per km, it seems as vibrations are an underreported event on the NCS.

The main reason for bit changes in the analysed wells was bit balling, in turn resulting in bit balling being the major contributor to NPT, with an estimated average cost of 3.2 mill NOK. It is worth noting the tripping speed is assumed constant at 500 m/hr for all wells. Limitations due to surge and swab pressures are not taken into consideration. The time to change the drill bit, or time to wait for the bit if it is not available on the rig, is also not accounted for. Neither is the extra cost associated with the need of an additional bit.

### *5.2 Economical Impact*

The cost estimation is a direct function of the estimated lost time and has therefore limitations related to the lost time estimations. Other limitations include the assumption that the rig-day rate is constant for every analysed well, even though the study period extends from 1997 to 2020. Assuming the same costs, however, makes the NPT costs more comparable between wells. It is also worth noting the rig-day cost assumed in this thesis is based on worldwide rig-prices (see *chapter 3.4.4 Estimate Costs due to Drilling Events and NPT*). This is a conservative assumption, as the rig-day prices on the NCS are generally higher than the worldwide rig-prices (Repsol, 2018).

An advantage of the cost estimations is the ability to estimate the price of the preventative measures while still reducing overall costs. If the developed workflow with mapped drilling events describes the events with such intensity that preventative measures are needed, the maximum cost of these preventative measures can be estimated.

From a net present value (NPV) point of view, reducing costs in the drilling phase of a project is important, as the drilling phase occurs in the initial phase of a project and the investment costs are thus less discounted compared to future income from oil and gas production. Applying the developed workflow on offset wells, knowledge of the subsurface is increased, decreasing the operational risk. Decreased operational risk may reduce the uncertainty of the investment costs. As the discount rate in the NPV calculation is a function of risk, an argument could be made that the discount rate could be reduced, further increasing the NPV of the project. Understanding the NPV associated with drilling events and NPT is of significant interest for various stakeholders as this would result in an increase/decrease in the overall value of the firm.

A total of 1807 exploration wells are drilled on the NCS, giving an average of 32 yearly drilled wells since the first well in 1966 (**Figure 89**). The estimated average costs due to drilling events and NPT of the wells analysed in this thesis is 7.2 mill NOK, resulting in an average estimated annual cost for drilling events and NPT of 230 mill NOK on the NCS. As the Petroleum tax system in Norway gives a 78% deduction from the investment costs for an operator and partners, there is a potential large cost saving for the society with a reduction in drilling events and NPT. It is worth noting the costs due to drilling events and NPT estimated in this thesis only takes into consideration the events related to drilling inefficiencies: bit balling, vibrations, bit wear and hard rock. The costs related to other events such as losses, shallow water flow, swelling and stuck pipe (Table 1) are not accounted for.

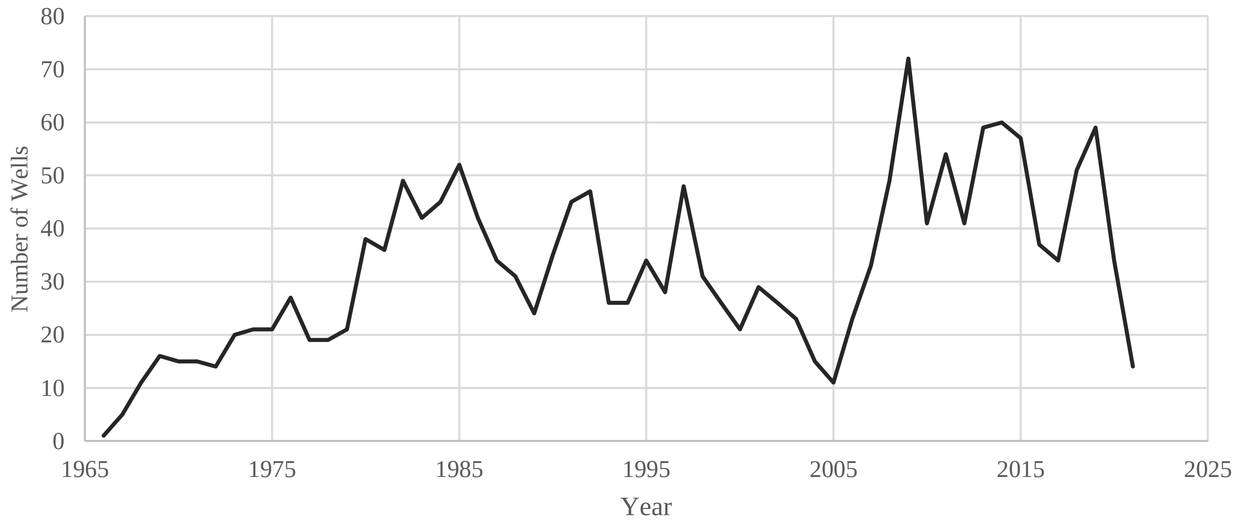


**Figure 89. Yearly drilled exploration wells on the NCS (NPD, 2021).**

### *5.3 Recommendations for Future Work*

By testing more wells, the developed workflow could be further improved. By comparing more wells to reported drilling events, further adjustments to the cut-off values could be done. A limiting factor with the cut-off value method is that depths are described as either efficient or inefficient. The difference between drilling at MSE and DS values of 149 MPa and 151 MPa are minor, but the first is detected as efficient and the latter as inefficient. Implementing more gradual cut-off values, depths could be described from “slightly inefficient” to “very inefficient”, for instance for cut-off values between 100 MPa and 200 MPa. The gradual cut-off values could then further be implemented in the drilling events detection and the severity and likeliness of the detected events could be described.

The two developed ROP estimations, the WOB-method and the Torque-method, give different results for different wells and for different parts of the wells, however the overall results are similar. A further improvement could be to utilize both methods simultaneously, and create a best case, worst case and average case for lost time and cost estimations.

The differentiation between bit balling and bit wear in the developed workflow is based on lithology restrictions. This assumption is based on the results obtained from the case study and the lack of obtained literature to differentiate the two events in another way from the available data. As bit wear also occurs in claystone formations, an improvement to the workflow would be to develop another method to differentiate the two events. By testing more wells with reported bit wear, other relationships may be discovered and implemented. Implementing time-based drilling mechanics is a potential way to identify bit wear and further improve the workflow (Millan & Ringer, 2018).

Chapter *4.2.2 Drilling Events Detection* show that the three wells with the most detected events are the deepest wells, with vertical depths exceeding 5000 meters. Figure 84 shows HPHT wells detect almost 3 times more drilling events compared to the other wells. Due to high pressure and temperature at these depths, more HPHT wells should be analysed and compared to reported events. Potential improvements for HPHT wells include changing cut-off values, changing MSE efficiency factors and different approaches to MSE baseline creation.

No wells from the Barents Sea were analysed in this thesis. Further work includes testing more wells in the Barents Sea and comparing the detected events to reported events. The geology in the Barents Sea differs from the North Sea and Norwegian Sea due to uplift and erosion, resulting in harder rock. This will most likely change the MSE and DS values, and different cut-off values may be necessary for the Barents Sea.

The developed workflow lays a foundation for real time drilling events detection and drilling optimization. Drilling parameters for calculating MSE, DS and MSE/DS are all available real-time, and hard rock could be detected from Sonic data, generally available for most parts of the well. Calculating MSE Corrected would be more difficult, as differentiating a trend from an event when creating baselines could be difficult real-time. Studying baselines from offset wells could help guide the baseline creation and improve the calculation of MSE Corrected.

# 6 CONCLUSION

The aim of this thesis has been to improve the understanding of drilling events to help mitigate non-productive time in drilling operations. A consistent data analytics workflow in MATLAB was developed to detect drilling events from depth-based drilling data and estimate non-productive time. The results obtained throughout this thesis indicate that the developed workflow effectively manages to detect bit balling, vibrations, and bit wear from drilling mechanics utilizing cut-off values for mechanical specific energy and drilling strength models. To detect hard rock, empirical correlations between sonic data and lithology were the preferred method. The detected drilling events from the developed workflow gave a good match with the reported drilling events from the conducted case study in the Northern North Sea.

Two methods were developed to estimate lost drilling time: the WOB-method and the Torque-method, utilizing the linear relationship between WOB-ROP and Torque-ROP for efficient drilling. Although these relationships were observed not to be perfectly linear, the two developed methods were shown to give good approximations of lost drilling time. Lost time due to unplanned bit changes was estimated from studying bit changes in conjunction with detected drilling events.

The developed workflow was tested on 20 wells from the North Sea and Norwegian Sea. A high negative correlation between the independently calculated drilling efficiency and the estimated costs for drilling events and NPT increased the confidence in the developed workflow. Vibrations were shown to be the most common drilling problem, constituting 39% of the detected events, followed by bit balling and bit wear at 29% and 27%, respectively. 5% of the detected events was due to hard rock. Vibrations were also the largest contributor to lost drilling time, closely followed by bit wear and bit balling. The major contributor to unplanned bit changes was bit balling.

As a result, 44% of the associated costs were estimated due to bit balling, followed by bit wear at 28%, vibrations at 25% and hard rock at 3%. 15% of the total time was estimated lost due to inefficient drilling and 3% was estimated lost due to unplanned bit changes, resulting in a total of 18% lost time due to drilling events, equal to an estimated average of 69 hours lost, or 5% lost rig-time.

The main advantage of the developed workflow is the ability to discover and map drilling events from limited available data in an effectively manner, also when no documentation is available or the events were not reported. By running the developed workflow on offset wells, the depth and intensity of the drilling events can be described and mapped in detail. If the drilling events are repeated with high intensity in several offset wells, preventative measures can be performed to limit drilling events and NPT prior to a drilling operation.

# 7 REFERENCES


Amadi, W. K. & Iyalla, I. (2012). Application of Mechanical Specific Energy Techniques in Reducing Drilling Cost in Deepwater Development. *Society of Petroleum Engineers*. https://doi.org/10.2118/156370-MS

Chang, C., Zoback, M. D., & Khaksar A. (2006). Empirical relations between rock strength and physical properties in sedimentary rocks. *Journal of Petroleum Science and Engineering, Volume 51*(Issues 3–4), 223-237.

Dupriest, F. E. & Koederitz, W. L. (23-25 February 2005). *Maximizing Drill Rates with Real-Time Surveillance of Mechanical Specific Energy*. [Conference presentation]. SPE/IADC Drilling Conference, Amsterdam, The Netherlands.

Guerrero, C. (2007, April 5). Drilling Optimization with Mechanical Specific Energy. *SPE Drilling Studies Group. Chevron.*

Hammoutene, C. (2012). FEA Modeled MSE/UCS Values Optimize PDC Design for Entire Hole Section. *Society of Petroleum Engineers*. https://doi.org/10.2118/149372-MS

Hamrick, T. R. (2011). *Optimization of Operating Parameters for Minimum Mechanical Specific Energy in Drilling*. [Ph.D. thesis, West Virginia University]. https://www.osti.gov/servlets/purl/1060223

*IADC Drilling Manual 12th Ed.* (2015). IADC.

IHS Markit. (2021). *Petrodata Offshore Rig Day Rate Trends*. IHS Markit. https://ihsmarkit.com/products/oil-gas-drilling-rigs-offshore-day-rates.html

Lal, M. (1999, April 20). *Shale stability: drilling fluid interaction and shale strength.* [Conference presentation]. SPE Latin American and Caribbean Petroleum Engineering Conference. Caracas, Venezuela.

Menand, S. & Mills, K. (2017). Use of Mechanical Specific Energy Calculation in Real-Time to Better Detect Vibrations and Bit Wear While Drilling. *American Association of Drilling Engineers.*

Millan, E. & Ringer, M. (2018, March 6-8). *A New Workflow for Estimating Bit Wear and Monitoring Drilling Efficiency in Real Time During Drilling Operations.* [Conference presentation]. IADC/SPE Drilling Conference and Exhibition Fort Worth. Texas.

Norwegian Petroleum Directorate. (2021, May 24). *Statistics*. NPD Factpages. https://factpages.npd.no/en/wellbore/statistics/completionyear

Norwegian Petroleum Directorate. (2021). *Factpages*. NPD Factpages. https://factpages.npd.no/en

Pessier, R. C. & Fear, M. J. (4-7 October 1992). *Quantifying Common Drilling Problems with Mechanical Specific Energy and a Bit-Specific Coefficient of Sliding Friction* [Conference presentation]. SPE Annual Technical Conference and Exhibition, Washington DC.

Romagnoli, R. (2017, March 29-31). *New Achievements Concerning Temperature Effects on Mud Rheology and Hydraulic Performance While Drilling.* [Conference presentation]. 13th Offshore Mediterranean Conference and Exhibition. Ravenna.

Repsol (2018). *Market Overview* [Brochure].

Schlumberger. (2021). *Day Rate*. Schlumberger Oilfield Glossary. https://glossary.oilfield.slb.com/en/Terms/d/day_rate.aspx

Schlumberger. (2010). *Drillstring Vibrations and Vibration Modeling* [Brochure]. https://www.slb.com/-/media/files/drilling/brochure/drillstring-vib-br

Soliman, A. A. (1995, March 11-14). *Oil Base Mud in High Pressure, High Temperature Wells.* [Conference presentation]. SPE Middle East Oil Show. Bahrain. https://doi.org/10.2118/29864-MS

Teale, R. (1965). The Concept of Specific Energy in Rock Drilling. *International Journal of Rock Mechanics and Mining Sciences Vol. 2* (No.1), 57-73.

United States Department of Agriculture. (2012). *National Engineering Handbook.* https://directives.sc.egov.usda.gov/OpenNonWebContent.aspx?content=31848.wba

Young, S. & Rabke, S. (2006, June 12-15). *Novel Fluid Design Can Eliminate OBM-Cuttings Waste*. SPE Europec/EAGE Annual Conference and Exhibition. Vienna.

Zhao, J., Labiouse V., Dudt J., & Mathier J. (Ed.). (2010). *Rock Mechanics in Civil and Environmental Engineering*. CRC Press.

# 8 APPENDIX

## *8.1 Appendix A - Tables*

**Table 11. Overview of the analysed wells from the NCS.**

| **Well** | **TD** mMD | **TD** mTVD | **Year** | **Drilling Fluid** | **Drilling Days** | **Efficiency** meter/day |
|---|---|---|---|---|---|---|
| 15/12-18 S | 3520 | 3311 | 2007 | OBM | 64 | 55.0 |
| 15/12-21 | 3310 | 3311 | 2009 | OBM | 68 | 48.7 |
| 15/12-23 | 3485 | 3478 | 2010 | OBM | 59 | 59.1 |
| 33/6-3 S | 4444 | 4200 | 2012 | OBM | 99 | 44.9 |
| 34/4-15 S | 3570 | 3455 | 2020 | OBM | 28 | 127.5 |
| 34/4-15 A | 3844 | 3600 | 2020 | OBM | 28 | 137.3 |
| 35/3-6 | 3366 | 3365 | 2002 | WBM | 56 | 60.1 |
| 35/8-4 | 3719 | 3718 | 1999 | WBM | 38 | 97.9 |
| 35/8-5 S | 4000 | 3832 | 2003 | OBM | 50 | 80.0 |
| 35/9-8 | 3256 | 3255 | 2013 | WBM | 70 | 46.5 |
| 35/9-11 S | 3800 | 3759 | 2014 | WBM | 45 | 84.4 |
| 6406/1-1 | 5057 | 5057 | 2001 | OBM | 67 | 75.5 |
| 6406/2-5 | 5439 | 5438 | 1997 | OBM | 119 | 45.7 |
| 6406/2-7 | 4981 | 4977 | 1999 | OBM | 96 | 51.9 |
| 6407/8-4 S | 2788 | 2650 | 2008 | WBM | 32 | 87.1 |
| 6407/8-5 S | 3240 | 2369 | 2009 | OBM | 25 | 129.6 |
| 6407/8-6 | 3420 | 3419 | 2013 | OBM | 31 | 110.3 |
| 6407/8-7 | 3030 | 3030 | 2015 | OBM | 16 | 189.4 |
| 6407/9-9 | 1920 | 1920 | 1999 | WBM | 15 | 128.0 |
| 6506/11-6 | 5275 | 5273 | 1998 | OBM | 180 | 29.3 |

**Table 12. Lost drilling time WOB-Method and contributing Drilling Events.**

| **Well** | **Drilling time** hours | **Lost Drilling time** hours | **Lost Drilling time** | **Bit balling** | **Vibrations** | **Bit Wear** | **Hard Rock** |
|---|---|---|---|---|---|---|---|
| 15/12-18 S | 141 | 34 | 24 % | 1 % | 44 % | 54 % | 1 % |
| 15/12-21 | 139 | 32 | 23 % | 1 % | 14 % | 81 % | 4 % |
| 15/12-23 | 153 | 34 | 22 % | 12 % | 0 % | 85 % | 2 % |
| 33/6-3 S | 256 | 81 | 32 % | 55 % | 41 % | 3 % | 0 % |
| 34/4-15 S | 106 | 24 | 23 % | 29 % | 50 % | 15 % | 6 % |
| 34/4-15 A | 92 | 4 | 5 % | 0 % | 100 % | 0 % | 0 % |
| 35/3-6 | 183 | 16 | 9 % | 74 % | 13 % | 13 % | 0 % |
| 35/8-4 | 142 | 40 | 28 % | 26 % | 48 % | 24 % | 2 % |
| 35/8-5 S | 192 | 50 | 26 % | 42 % | 5 % | 50 % | 3 % |
| 35/9-8 | 185 | 54 | 30 % | 12 % | 44 % | 43 % | 1 % |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 35/9-11 S | 179 | 13 | 7 % | 35 % | 46 % | 19 % | 0 % |
| 6406/1-1 | 378 | 165 | 44 % | 56 % | 5 % | 37 % | 2 % |
| 6406/2-5 | 524 | 327 | 62 % | 61 % | 14 % | 25 % | 0 % |
| 6406/2-7 | 291 | 95 | 33 % | 40 % | 25 % | 34 % | 1 % |
| 6407/8-4 S | 110 | 16 | 14 % | 88 % | 10 % | 0 % | 2 % |
| 6407/8-5 S | 100 | 1 | 1 % | 16 % | 39 % | 45 % | 0 % |
| 6407/8-6 | 90 | 9 | 10 % | 11 % | 63 % | 19 % | 7 % |
| 6407/8-7 | 54 | 3 | 6 % | 10 % | 26 % | 57 % | 7 % |
| 6407/9-9 | 34 | 8 | 22 % | 23 % | 36 % | 41 % | 0 % |
| 6506/11-6 | 425 | 235 | 55 % | 12 % | 48 % | 26 % | 14 % |
| **Average** | **189** | **62** | **24 %** | **30 %** | **34 %** | **34 %** | **3 %** |

**Table 13. Lost drilling time Torque-Method and contributing Drilling Events.**

| **Well** | **Drilling time** hours | **Lost Drilling time** hours | **Lost Drilling time** | **Bit balling** | **Vibrations** | **Bit Wear** | **Hard Rock** |
|---|---|---|---|---|---|---|---|
| 15/12-18 S | 141 | 34 | 24 % | 1 % | 46 % | 52 % | 1 % |
| 15/12-21 | 139 | 34 | 24 % | 1 % | 15 % | 80 % | 4 % |
| 15/12-23 | 153 | 33 | 22 % | 12 % | 0 % | 85 % | 3 % |
| 33/6-3 S | 256 | 77 | 30 % | 58 % | 38 % | 3 % | 0 % |
| 34/4-15 S | 106 | 20 | 18 % | 23 % | 55 % | 14 % | 8 % |
| 34/4-15 A | 92 | 4 | 4 % | 0 % | 95 % | 0 % | 5 % |
| 35/3-6 | 183 | 15 | 8 % | 72 % | 15 % | 11 % | 2 % |
| 35/8-4 | 142 | 39 | 27 % | 26 % | 48 % | 23 % | 2 % |
| 35/8-5 S | 192 | 51 | 27 % | 40 % | 8 % | 48 % | 4 % |
| 35/9-8 | 185 | 51 | 27 % | 11 % | 49 % | 39 % | 1 % |
| 35/9-11 S | 179 | 17 | 9 % | 20 % | 68 % | 12 % | 0 % |
| 6406/1-1 | 378 | 162 | 43 % | 56 % | 8 % | 35 % | 1 % |
| 6406/2-5 | 524 | 295 | 56 % | 58 % | 18 % | 24 % | 0 % |
| 6406/2-7 | 291 | 94 | 32 % | 39 % | 28 % | 32 % | 1 % |
| 6407/8-4 S | 110 | 15 | 14 % | 88 % | 10 % | 0 % | 2 % |
| 6407/8-5 S | 100 | 1 | 1 % | 16 % | 39 % | 45 % | 0 % |
| 6407/8-6 | 90 | 9 | 9 % | 11 % | 64 % | 18 % | 7 % |
| 6407/8-7 | 54 | 3 | 6 % | 11 % | 25 % | 57 % | 7 % |
| 6407/9-9 | 34 | 8 | 22 % | 23 % | 36 % | 41 % | 0 % |
| 6506/11-6 | 425 | 227 | 53 % | 12 % | 49 % | 24 % | 14 % |
| **Average** | **189** | **59** | **23 %** | **29 %** | **36 %** | **32 %** | **3 %** |

**Table 14. Lost drilling time average WOB- and Torque-Method and contributing Drilling Events.**

| Well | **Drilling time** hours | **Lost Drilling time** hours | **Lost Drilling time** | **Bit balling** | **Vibrations** | **Bit Wear** | **Hard Rock** |
|---|---|---|---|---|---|---|---|
| 15/12-18 S | 141 | 34 | 24 % | 1 % | 45 % | 53 % | 1 % |
| 15/12-21 | 139 | 33 | 24 % | 1 % | 14 % | 80 % | 4 % |
| 15/12-23 | 153 | 33 | 22 % | 12 % | 0 % | 85 % | 2 % |
| 33/6-3 S | 256 | 79 | 31 % | 57 % | 40 % | 3 % | 0 % |
| 34/4-15 S | 106 | 22 | 21 % | 26 % | 53 % | 14 % | 7 % |
| 34/4-15 A | 92 | 4 | 4 % | 0 % | 97 % | 0 % | 3 % |
| 35/3-6 | 183 | 16 | 9 % | 73 % | 14 % | 12 % | 1 % |
| 35/8-4 | 142 | 40 | 28 % | 26 % | 48 % | 24 % | 2 % |
| 35/8-5 S | 192 | 51 | 26 % | 41 % | 6 % | 49 % | 3 % |
| 35/9-8 | 185 | 53 | 28 % | 11 % | 47 % | 41 % | 1 % |
| 35/9-11 S | 179 | 15 | 8 % | 28 % | 57 % | 15 % | 0 % |
| 6406/1-1 | 378 | 163 | 43 % | 56 % | 6 % | 36 % | 1 % |
| 6406/2-5 | 524 | 311 | 59 % | 60 % | 16 % | 24 % | 0 % |
| 6406/2-7 | 291 | 95 | 32 % | 39 % | 27 % | 33 % | 1 % |
| 6407/8-4 S | 110 | 15 | 14 % | 88 % | 10 % | 0 % | 2 % |
| 6407/8-5 S | 100 | 1 | 1 % | 16 % | 39 % | 45 % | 0 % |
| 6407/8-6 | 90 | 9 | 10 % | 11 % | 64 % | 18 % | 7 % |
| 6407/8-7 | 54 | 3 | 6 % | 11 % | 26 % | 57 % | 7 % |
| 6407/9-9 | 34 | 8 | 22 % | 23 % | 36 % | 41 % | 0 % |
| 6506/11-6 | 425 | 231 | 54 % | 12 % | 49 % | 25 % | 14 % |
| **Average** | **189** | **61** | **23 %** | **30 %** | **35 %** | **33 %** | **3 %** |

**Table 15. Lost tripping time and contributing Drilling Events.**

| Well | **Tripping time** hours | **Lost Tripping time** hours | **Lost Tripping time** | **Bit balling** | **Vibrations** | **Bit Wear** | **Hard Rock** |
|---|---|---|---|---|---|---|---|
| 15/12-18 S | 60 | 12 | 19 % | 0 % | 50 % | 50 % | 0 % |
| 15/12-21 | 102 | 0 | 0 % | 0 % | 0 % | 0 % | 0 % |
| 15/12-23 | 78 | 0 | 0 % | 0 % | 0 % | 0 % | 0 % |
| 33/6-3 S | 90 | 34 | 38 % | 100 % | 0 % | 0 % | 0 % |
| 34/4-15 S | 81 | 0 | 0 % | 0 % | 0 % | 0 % | 0 % |
| 34/4-15 A | 108 | 0 | 0 % | 0 % | 0 % | 0 % | 0 % |
| 35/3-6 | 68 | 17 | 26 % | 100 % | 0 % | 0 % | 0 % |
| 35/8-4 | 59 | 7 | 13 % | 100 % | 0 % | 0 % | 0 % |
| 35/8-5 S | 169 | 12 | 7 % | 100 % | 0 % | 0 % | 0 % |
| 35/9-8 | 114 | 0 | 0 % | 0 % | 0 % | 0 % | 0 % |
| 35/9-11 S | 42 | 0 | 0 % | 0 % | 0 % | 0 % | 0 % |
| 6406/1-1 | 232 | 17 | 7 % | 100 % | 0 % | 0 % | 0 % |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6406/2-5 | 60 | 0 | 0 % | 0 % | 0 % | 0 % | 0 % |
| 6406/2-7 | 56 | 0 | 0 % | 0 % | 0 % | 0 % | 0 % |
| 6407/8-4 S | 78 | 8 | 10 % | 100 % | 0 % | 0 % | 0 % |
| 6407/8-5 S | 28 | 0 | 0 % | 0 % | 0 % | 0 % | 0 % |
| 6407/8-6 | 87 | 13 | 15 % | 0 % | 100 % | 0 % | 0 % |
| 6407/8-7 | 18 | 0 | 0 % | 0 % | 0 % | 0 % | 0 % |
| 6407/9-9 | 21 | 0 | 0 % | 0 % | 0 % | 0 % | 0 % |
| 6506/11-6 | 441 | 42 | 9 % | 100 % | 0 % | 0 % | 0 % |
| **Average** | **100** | **8** | **7 %** | **35 %** | **8 %** | **3 %** | **0 %** |

**Table 16. Total meters of detected Drilling Events and the Drilling Events contribution to total NPT.**

| **Well** | **Total Bit balling** m | **Total Vibrations** m | **Total Bit Wear** m | **Total Hard Rock** m | **NPT Bit Balling** | **NPT Vibrations** | **NPT Bit Wear** | **NPT Hard Rock** |
|---|---|---|---|---|---|---|---|---|
| 15/12-18 S | 2 | 285 | 72 | 1 | 1 % | 46 % | 52 % | 1 % |
| 15/12-21 | 3 | 62 | 308 | 41 | 1 % | 14 % | 80 % | 4 % |
| 15/12-23 | 86 | 3 | 338 | 17 | 12 % | 0 % | 85 % | 2 % |
| 33/6-3 S | 184 | 204 | 25 | 2 | 70 % | 28 % | 2 % | 0 % |
| 34/4-15 S | 84 | 94 | 20 | 9 | 26 % | 53 % | 14 % | 7 % |
| 34/4-15 A | 0 | 73 | 0 | 10 | 0 % | 97 % | 0 % | 3 % |
| 35/3-6 | 83 | 45 | 21 | 5 | 87 % | 7 % | 6 % | 1 % |
| 35/8-4 | 82 | 189 | 60 | 9 | 38 % | 41 % | 20 % | 2 % |
| 35/8-5 S | 136 | 82 | 135 | 57 | 52 % | 5 % | 40 % | 3 % |
| 35/9-8 | 32 | 175 | 151 | 15 | 11 % | 47 % | 41 % | 1 % |
| 35/9-11 S | 154 | 1711 | 100 | 1 | 28 % | 57 % | 15 % | 0 % |
| 6406/1-1 | 773 | 138 | 327 | 18 | 60 % | 6 % | 33 % | 1 % |
| 6406/2-5 | 536 | 262 | 342 | 72 | 60 % | 16 % | 24 % | 0 % |
| 6406/2-7 | 162 | 286 | 144 | 14 | 39 % | 27 % | 33 % | 1 % |
| 6407/8-4 S | 35 | 11 | 0 | 10 | 92 % | 6 % | 0 % | 1 % |
| 6407/8-5 S | 3 | 7 | 8 | 0 | 16 % | 39 % | 45 % | 0 % |
| 6407/8-6 | 18 | 114 | 32 | 16 | 4 % | 85 % | 7 % | 3 % |
| 6407/8-7 | 7 | 17 | 28 | 5 | 11 % | 26 % | 57 % | 7 % |
| 6407/9-9 | 32 | 23 | 67 | 1 | 23 % | 36 % | 41 % | 0 % |
| 6506/11-6 | 163 | 903 | 212 | 161 | 26 % | 41 % | 21 % | 12 % |
| **Average** | 128.75 | 234.2 | 119.5 | 23.2 | 33 % | 34 % | 31 % | 2 % |
| **Totals** | 2575 | 4684 | 2390 | 464 | | | | |

**Table 17. Costs NPT.**

| **Well** | **Cost NPT** NOK | **Cost NPT** NOK/meter | **Cost Bit Balling** NOK | **Cost Vibrations** NOK | **Cost Bit Wear** NOK | **Cost Hard Rock** NOK |
|---|---|---|---|---|---|---|
| 15/12-18 S | kr 4 700 000 | kr 1 300 | kr 35 000 | kr 2 181 000 | kr 2 459 000 | kr 25 000 |
| 15/12-21 | kr 3 400 000 | kr 1 000 | kr 35 000 | kr 491 000 | kr 2 733 000 | kr 141 000 |
| 15/12-23 | kr 3 500 000 | kr 1 000 | kr 421 000 | kr 12 000 | kr 2 982 000 | kr 86 000 |
| 33/6-3 S | kr 11 800 000 | kr 2 700 | kr 8 230 000 | kr 3 269 000 | kr 277 000 | kr 24 000 |
| 34/4-15 S | kr 2 300 000 | kr 600 | kr 597 000 | kr 1 210 000 | kr 333 000 | kr 161 000 |
| 34/4-15 A | kr 400 000 | kr 100 | kr 0 | kr 389 000 | kr 0 | kr 11 000 |
| 35/3-6 | kr 3 500 000 | kr 1 000 | kr 3 045 000 | kr 234 000 | kr 200 000 | kr 21 000 |
| 35/8-4 | kr 4 900 000 | kr 1 300 | kr 1 855 000 | kr 1 986 000 | kr 978 000 | kr 81 000 |
| 35/8-5 S | kr 6 500 000 | kr 1 600 | kr 3 402 000 | kr 330 000 | kr 2 585 000 | kr 182 000 |
| 35/9-8 | kr 5 500 000 | kr 1 700 | kr 629 000 | kr 2 566 000 | kr 2 250 000 | kr 56 000 |
| 35/9-11 S | kr 1 500 000 | kr 400 | kr 413 000 | kr 855 000 | kr 231 000 | kr 1 000 |
| 6406/1-1 | kr 18 800 000 | kr 3 700 | kr 11 345 000 | kr 1 091 000 | kr 6 127 000 | kr 237 000 |
| 6406/2-5 | kr 32 400 000 | kr 6 000 | kr 19 318 000 | kr 5 173 000 | kr 7 909 000 | kr 0 |
| 6406/2-7 | kr 9 900 000 | kr 2 000 | kr 3 895 000 | kr 2 661 000 | kr 3 270 000 | kr 74 000 |
| 6407/8-4 S | kr 2 400 000 | kr 900 | kr 2 216 000 | kr 155 000 | kr 0 | kr 30 000 |
| 6407/8-5 S | kr 100 000 | kr 30 | kr 16 000 | kr 39 000 | kr 45 000 | kr 0 |
| 6407/8-6 | kr 2 200 000 | kr 600 | kr 97 000 | kr 1 879 000 | kr 163 000 | kr 61 000 |
| 6407/8-7 | kr 400 000 | kr 100 | kr 42 000 | kr 103 000 | kr 228 000 | kr 27 000 |
| 6407/9-9 | kr 800 000 | kr 400 | kr 182 000 | kr 287 000 | kr 329 000 | kr 2 000 |
| 6506/11-6 | kr 28 400 000 | kr 5 400 | kr 7 293 000 | kr 11 679 000 | kr 6 077 000 | kr 3 351 000 |
| **Average** | kr 7 170 000 | kr 1 600 | kr 3 200 000 | kr 1 800 000 | kr 2 000 000 | kr 200 000 |
| **Totals** | kr 143 400 000 | | kr 63 100 000 | kr 36 600 000 | kr 39 200 000 | kr 4 600 000 |

## *8.2 Appendix B – Case Study Figures*

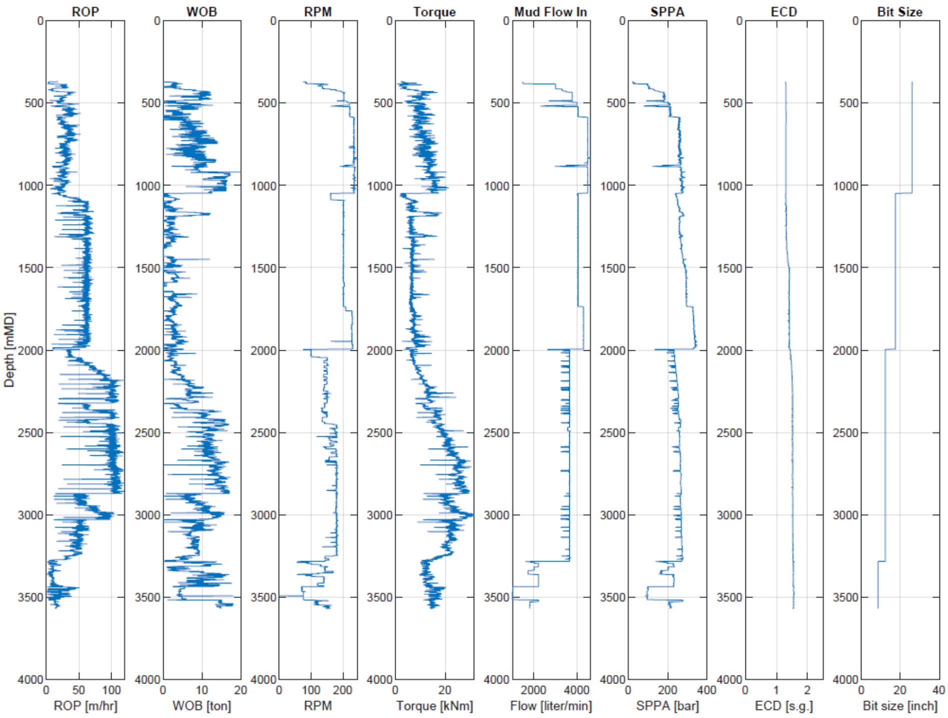


**Figure 90. Drilling Parameters well 34/4-15 S.**

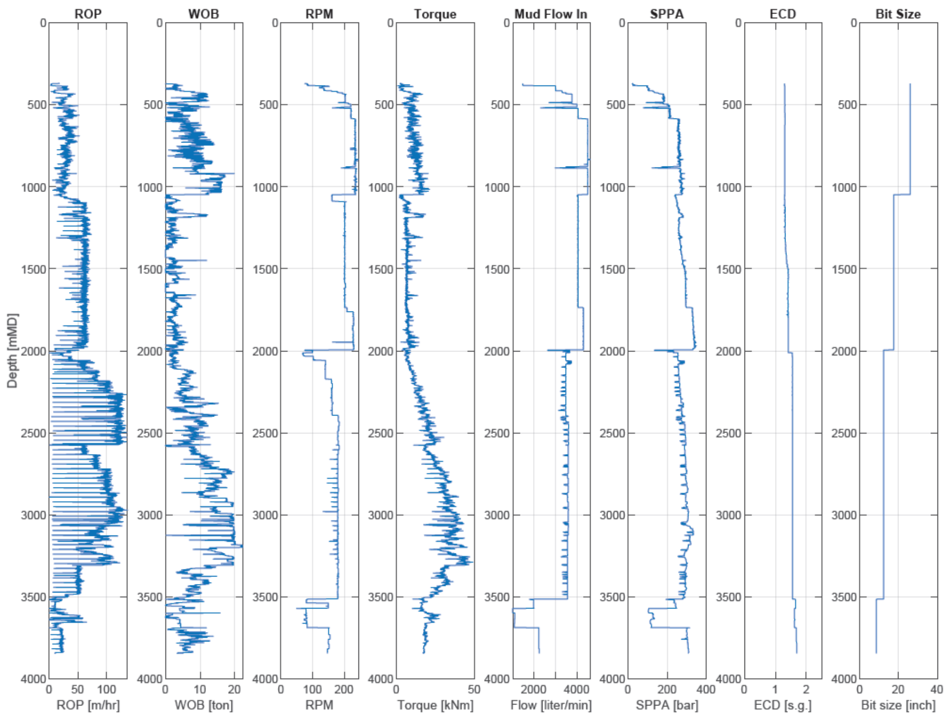


**Figure 91. Drilling Parameters well 34/4-15 A.**

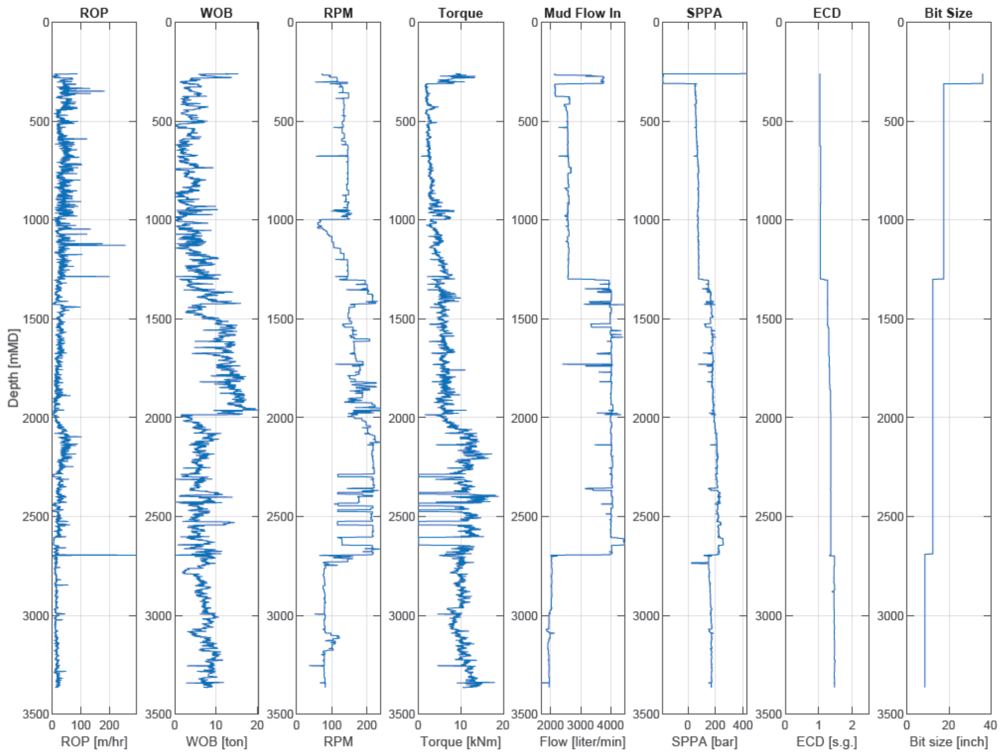


Figure 92. Drilling Parameters well 35/3-6.

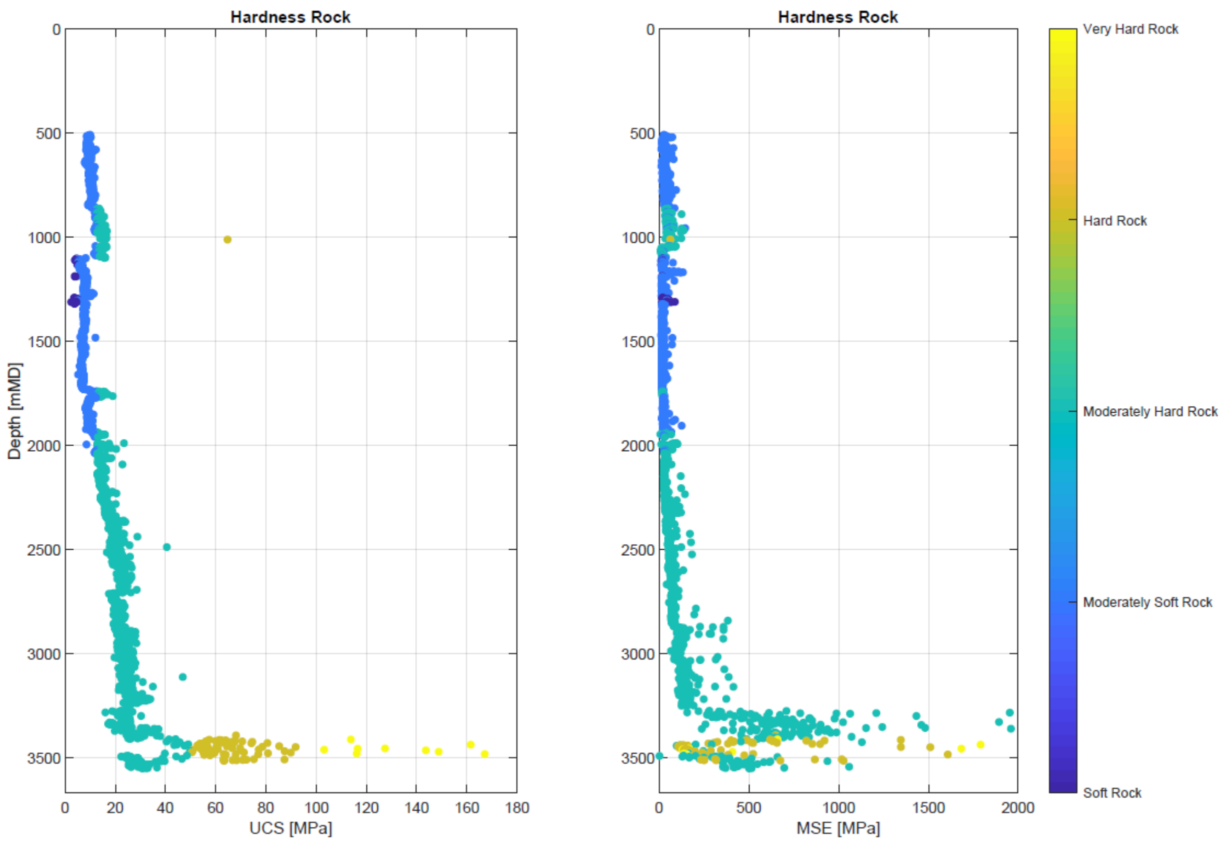


Figure 93. Hardness Rock, UCS and MSE, well 34/4-15 S.

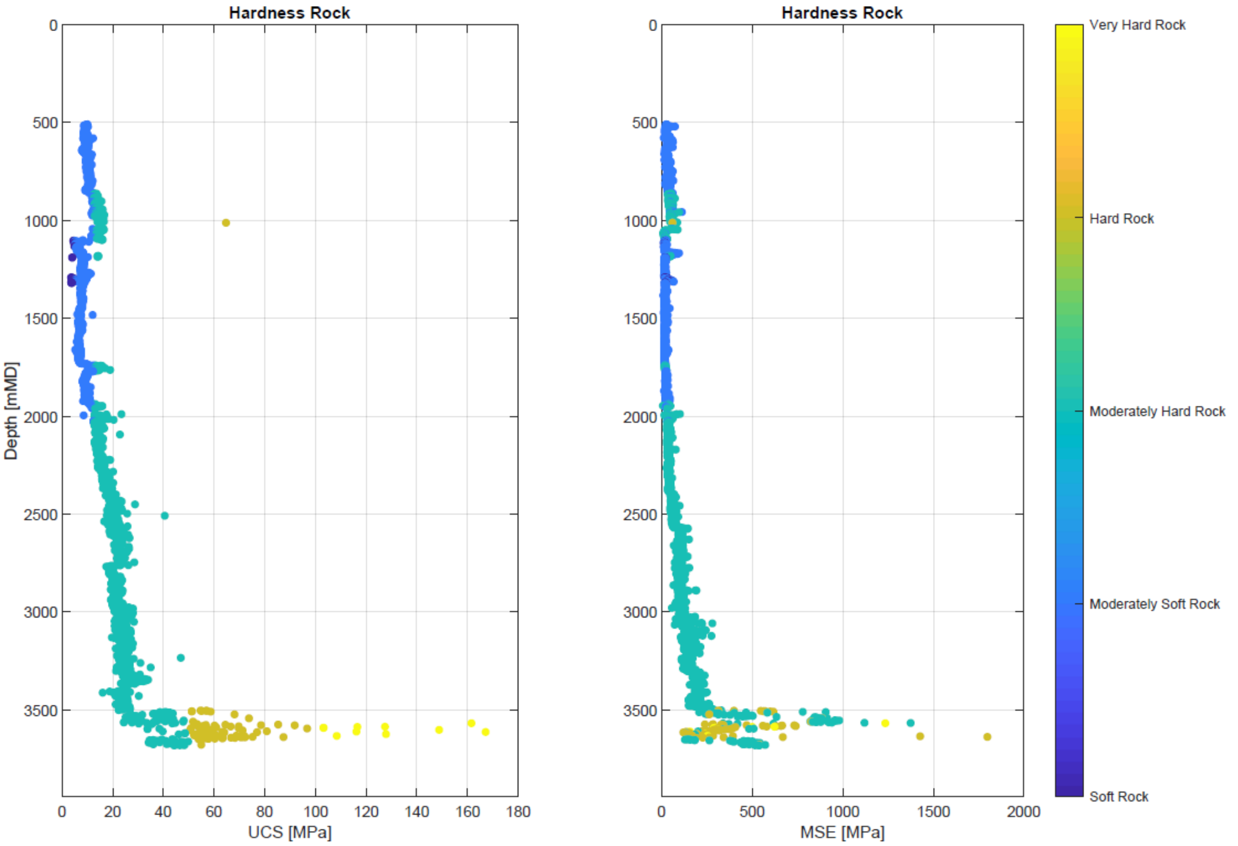


**Figure 94. Hardness Rock, UCS and MSE, well 34/4-15 A.**

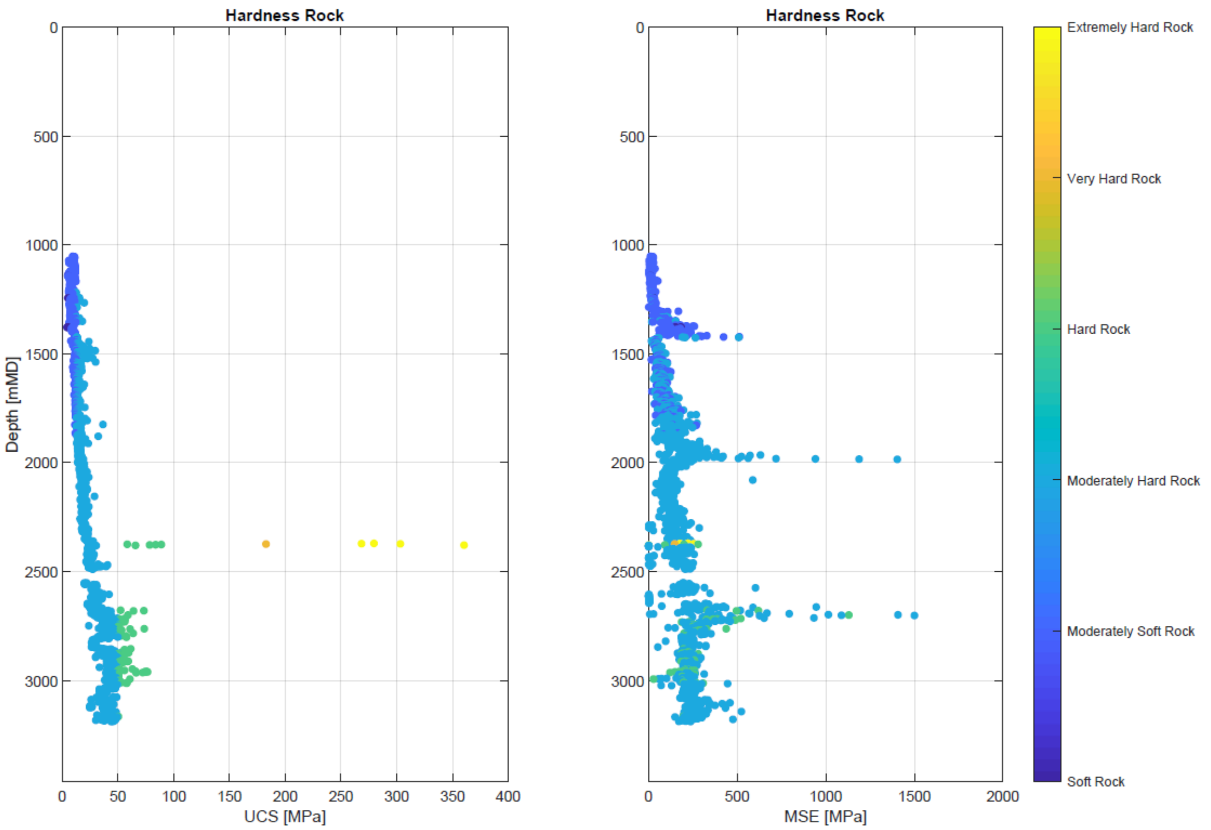


**Figure 95. Hardness Rock, UCS and MSE, well 35/3-6.**

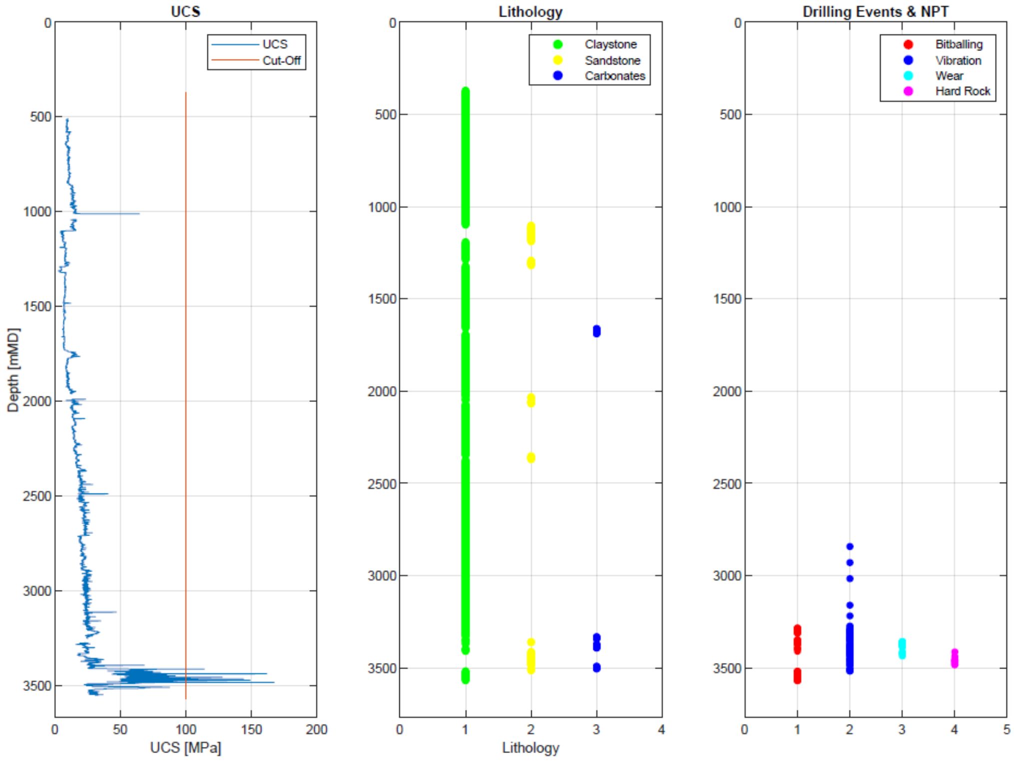


**Figure 96. Updated Drilling Events Detection based on UCS Cut-offs and Lithology Restrictions, well 34/4-15 S.**

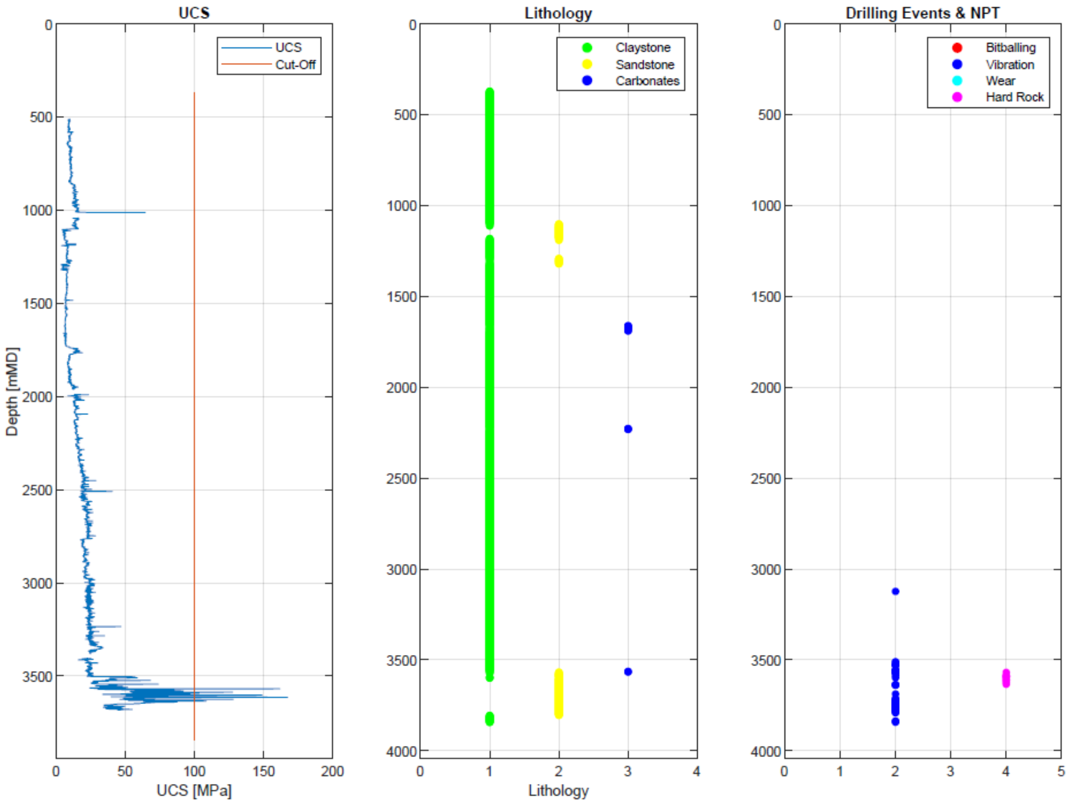


**Figure 97. Updated Drilling Events Detection based on UCS Cut-offs and Lithology Restrictions, well 34/4-15 A.**

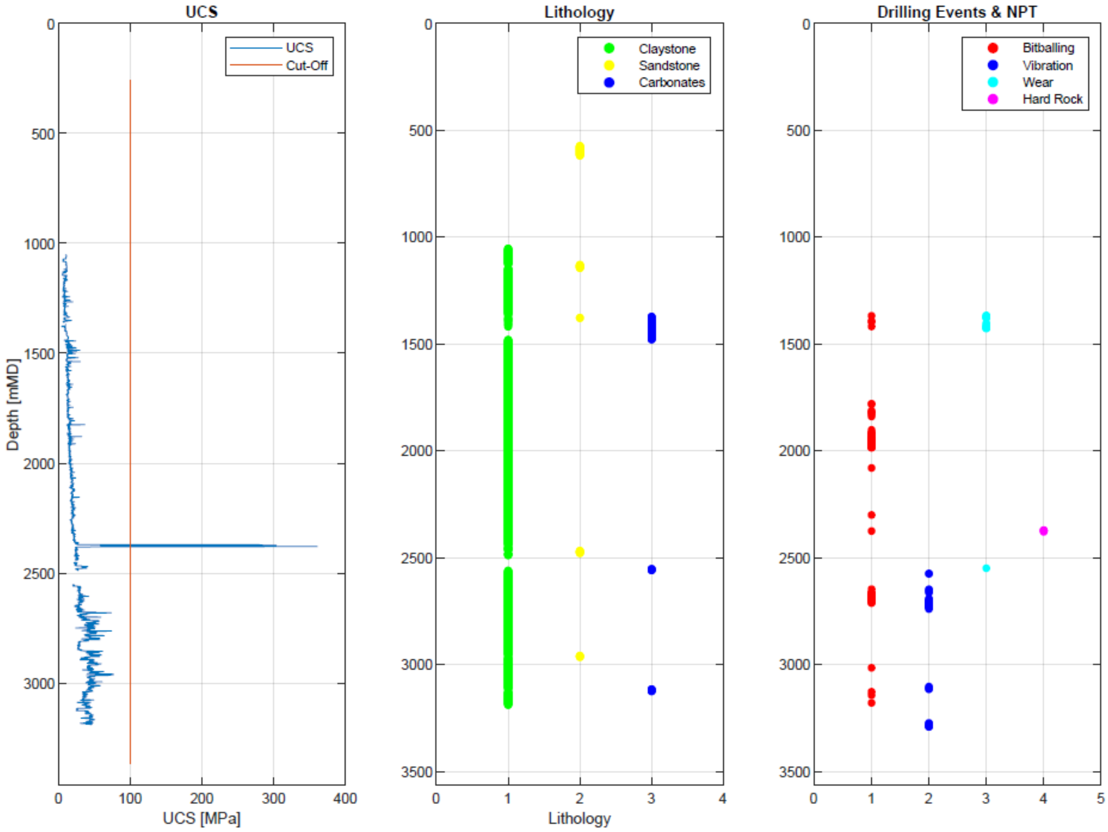


**Figure 98. Updated Drilling Events Detection based on UCS Cut-offs and Lithology Restrictions, well 35/3-6.**

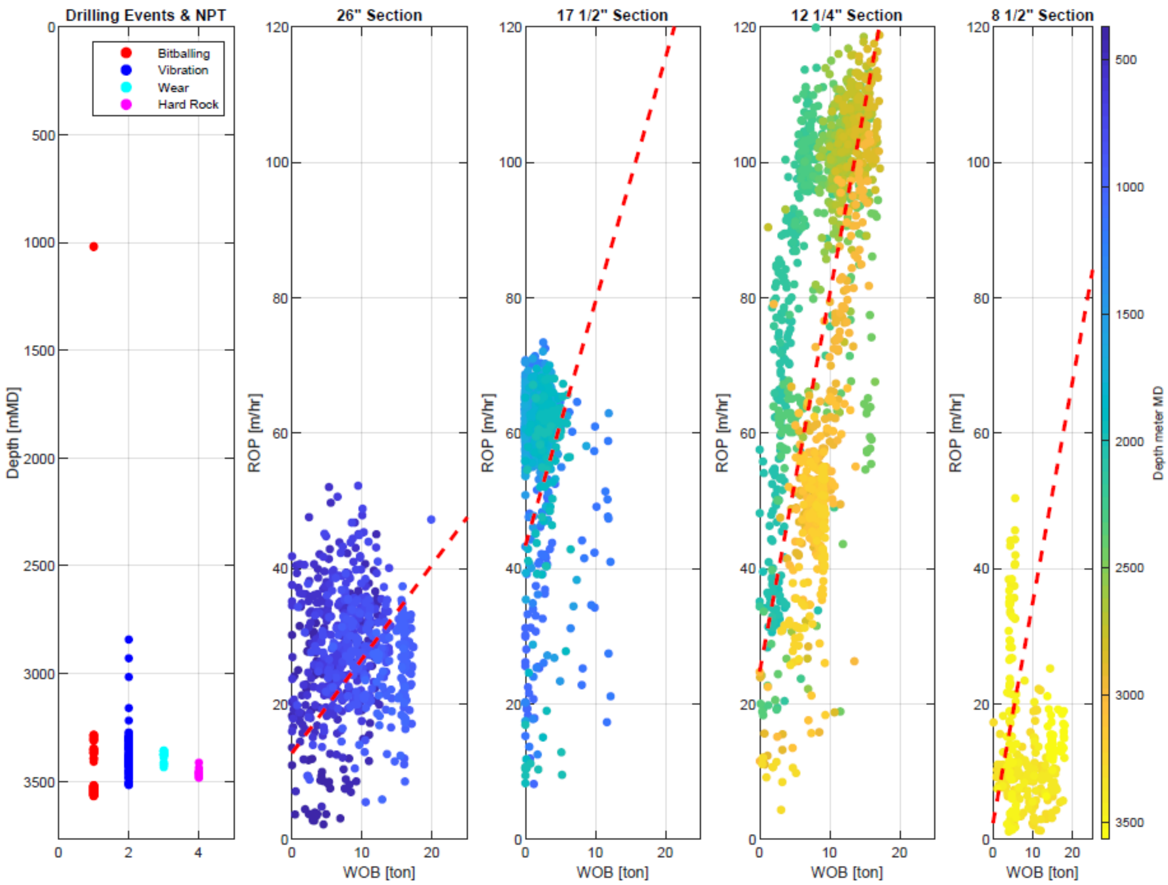


**Figure 99. Drilling Events Detection, WOB-ROP Cross plots vs Depth and Efficient Bit-lines (red lines), well 34/4-15 S.**

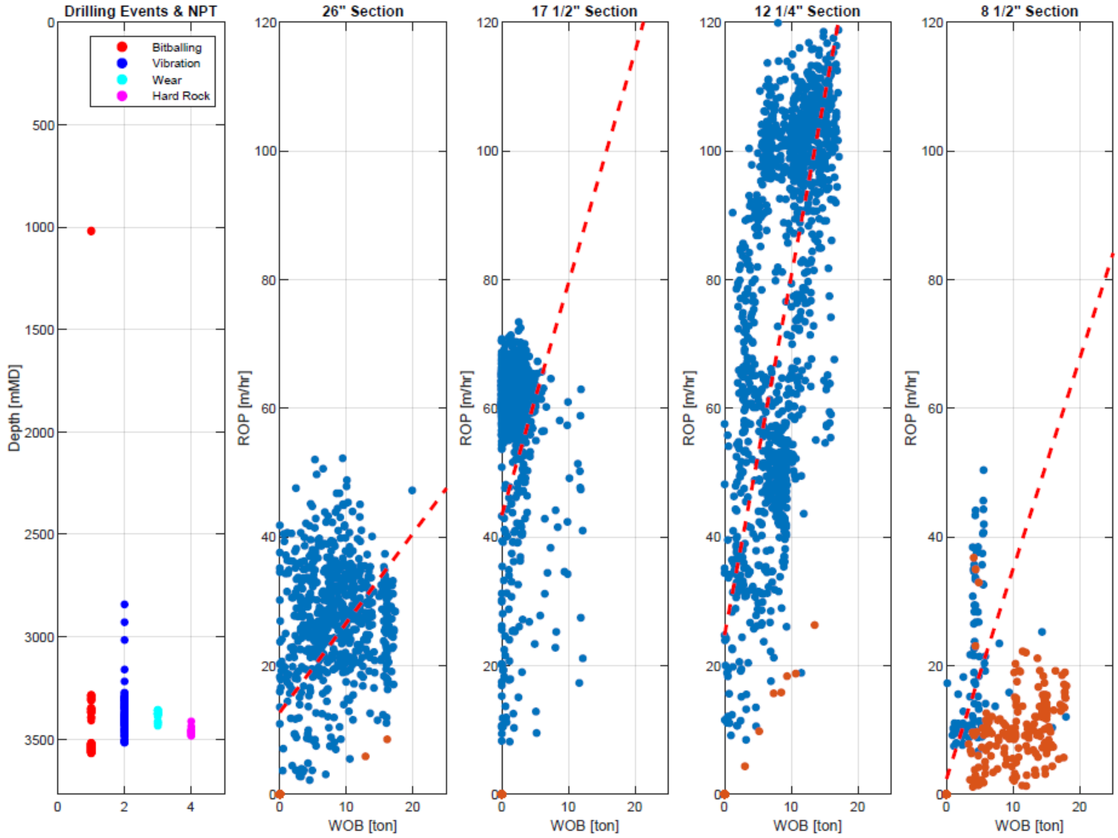


**Figure 100. Drilling Events Detection, WOB-ROP Cross plots, Efficient (blue) and Non-Efficient (orange) Depths and Efficient Bit-lines (red), well 34/4-15 S.**

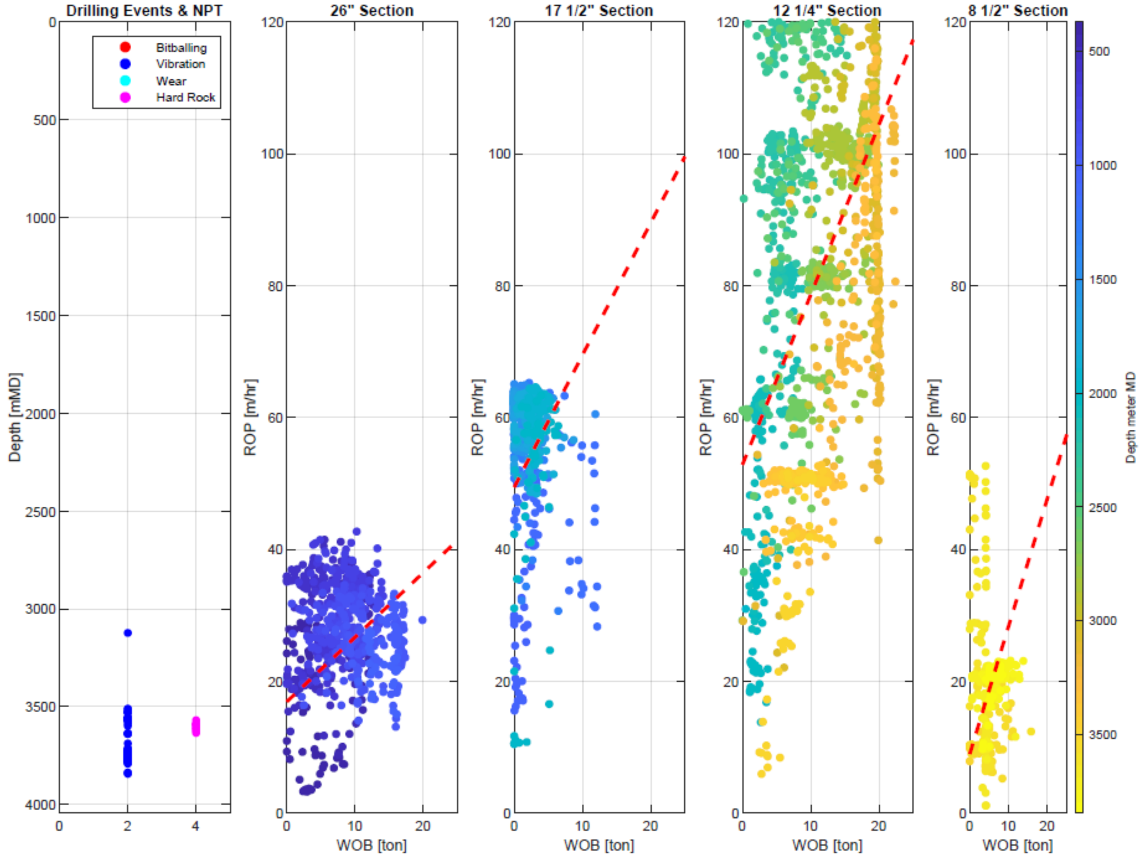


**Figure 101. Drilling Events Detection, WOB-ROP Cross plots vs Depth and Efficient Bit-lines (red lines), well 34/4-15 A.**

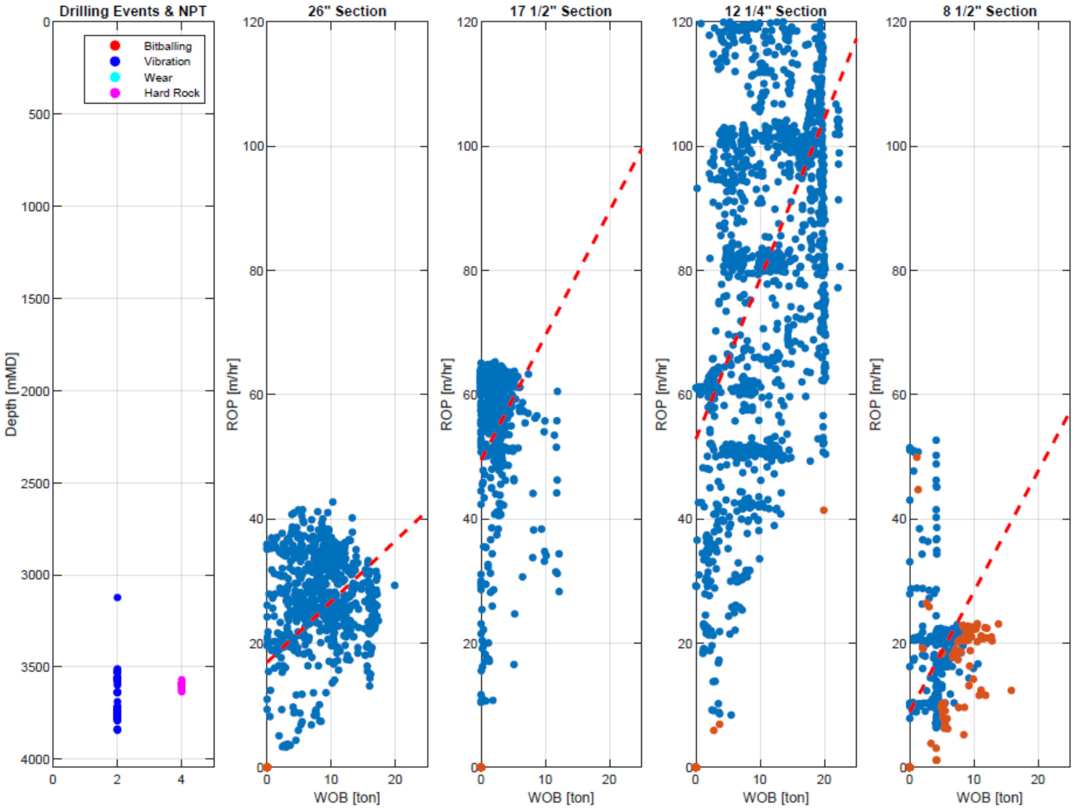


**Figure 102. Drilling Events Detection, WOB-ROP Cross plots, Efficient (blue) and Non-Efficient (orange) Depths and Efficient Bit-lines (red), well 34/4-15 A.**

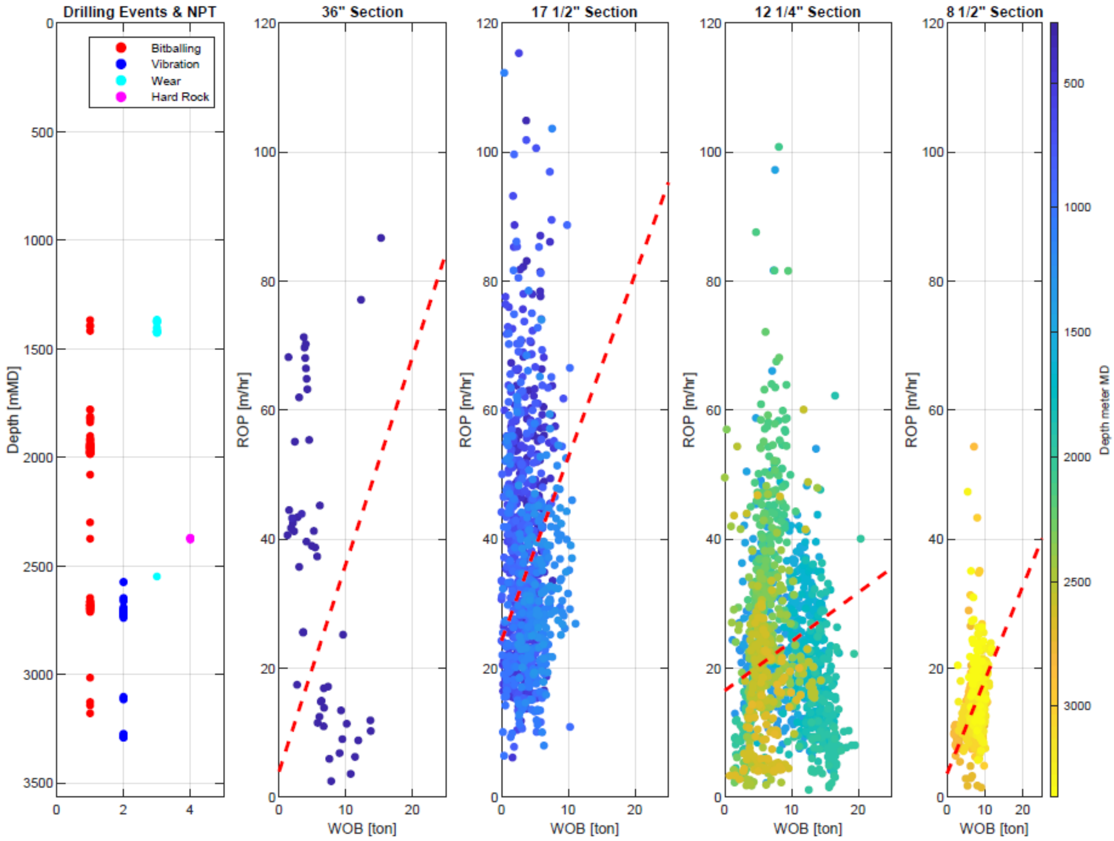


**Figure 103. Drilling Events Detection, WOB-ROP Cross plots vs Depth and Efficient Bit-lines (red lines), well 35/3-6.**

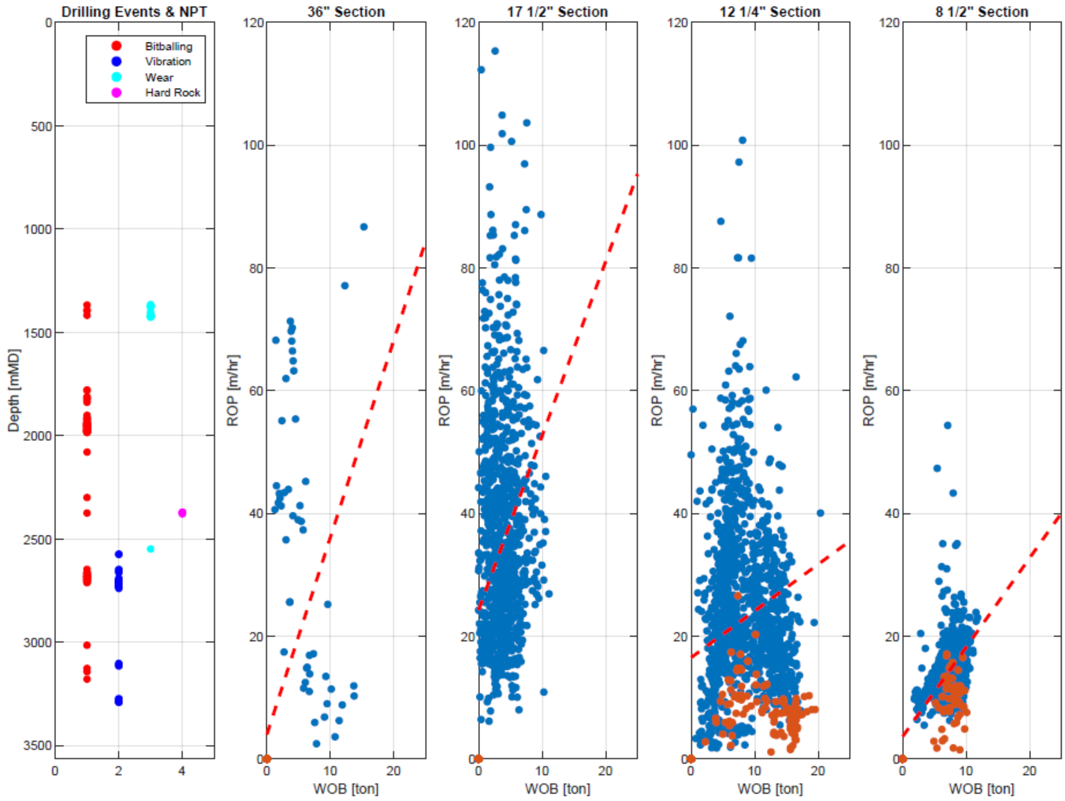


**Figure 104. Drilling Events Detection, WOB-ROP Cross plots, Efficient (blue) and Non-Efficient (orange) Depths and Efficient Bit-lines (red), well 35/3-6.**

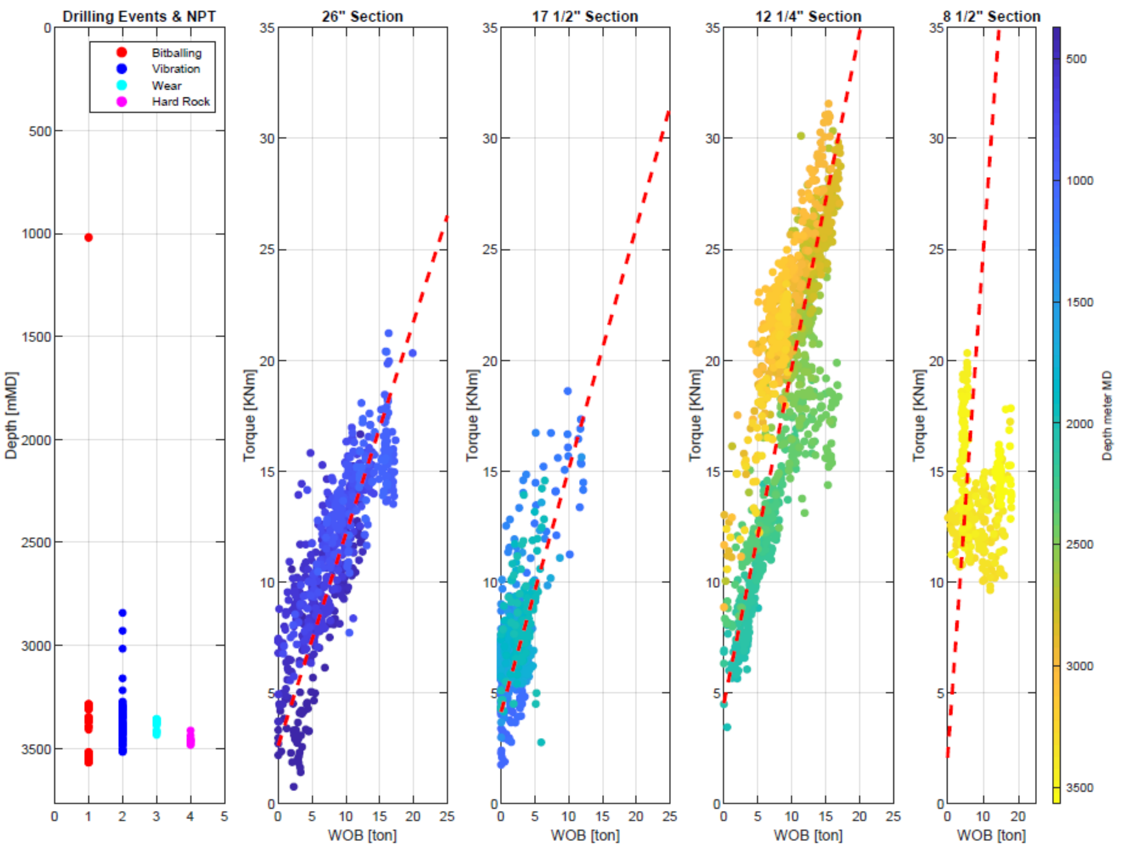


**Figure 105. Drilling Events Detection, WOB-Torque Cross plots vs Depth and Efficient Bit-lines (red lines), well 34/4-15 S.**

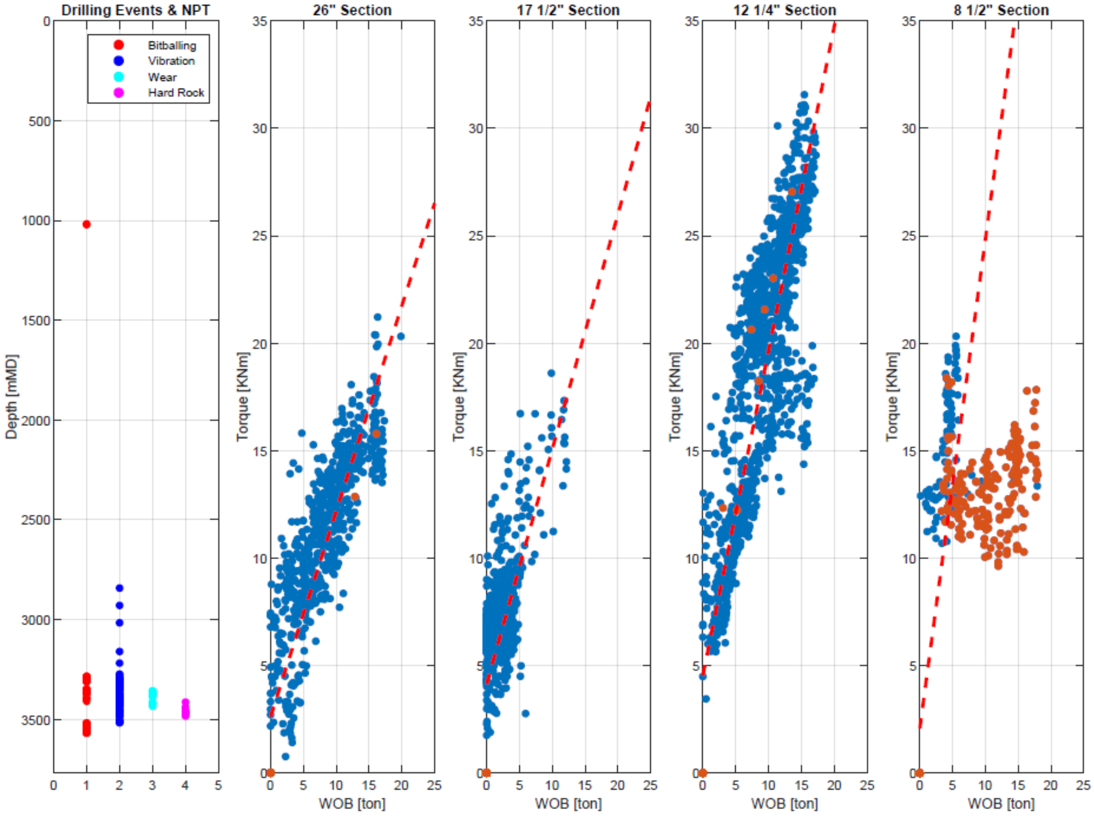


**Figure 106. Drilling Events Detection, WOB-Torque Cross plots, Efficient (blue) and Non-Efficient (orange) Depths and Efficient Bit-lines (red), well 34/4-15 S.**

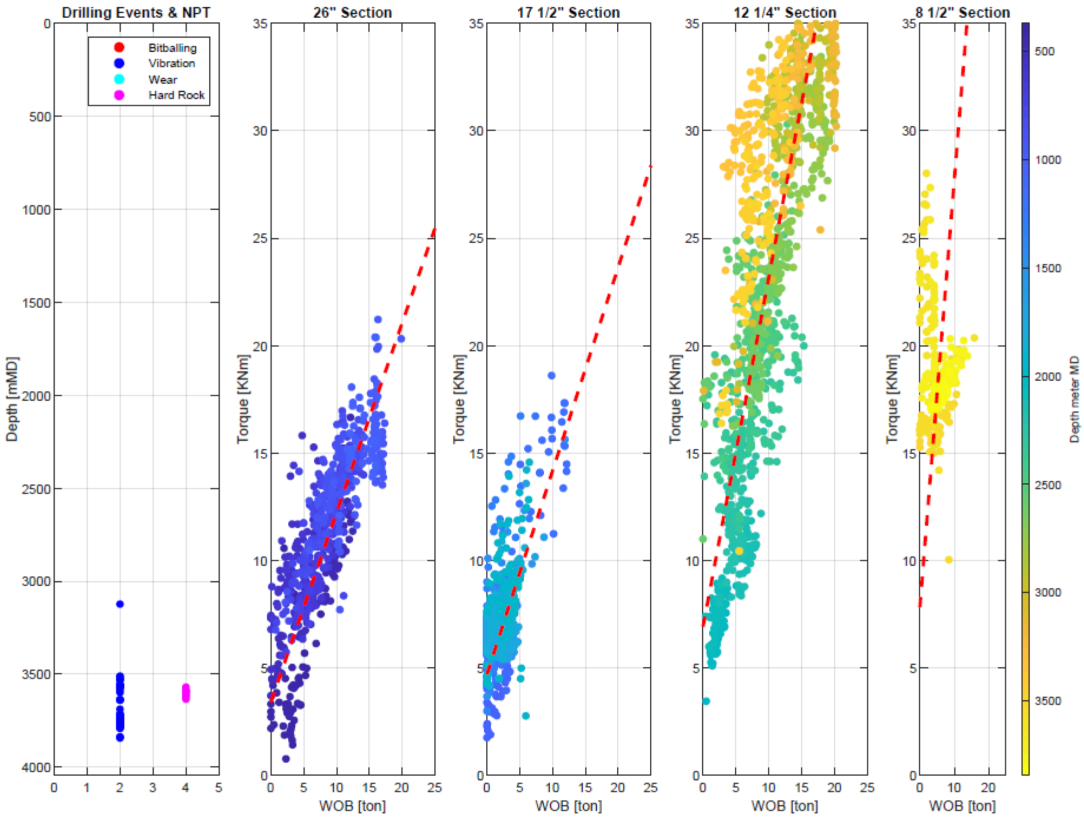


**Figure 107. Drilling Events Detection, WOB-Torque Cross plots vs Depth and Efficient Bit-lines (red lines), well 34/4-15 A.**

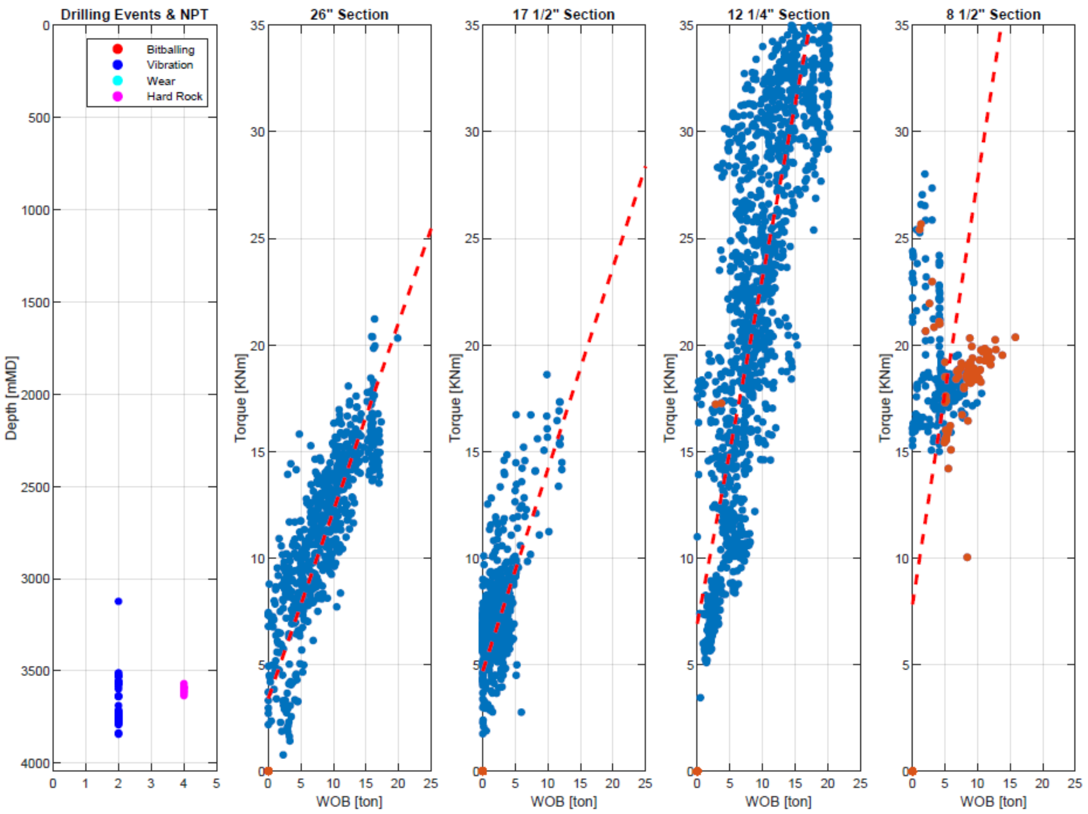


**Figure 108. Drilling Events Detection, WOB-Torque Cross plots, Efficient (blue) and Non-Efficient (orange) Depths and Efficient Bit-lines (red), well 34/4-15 A.**

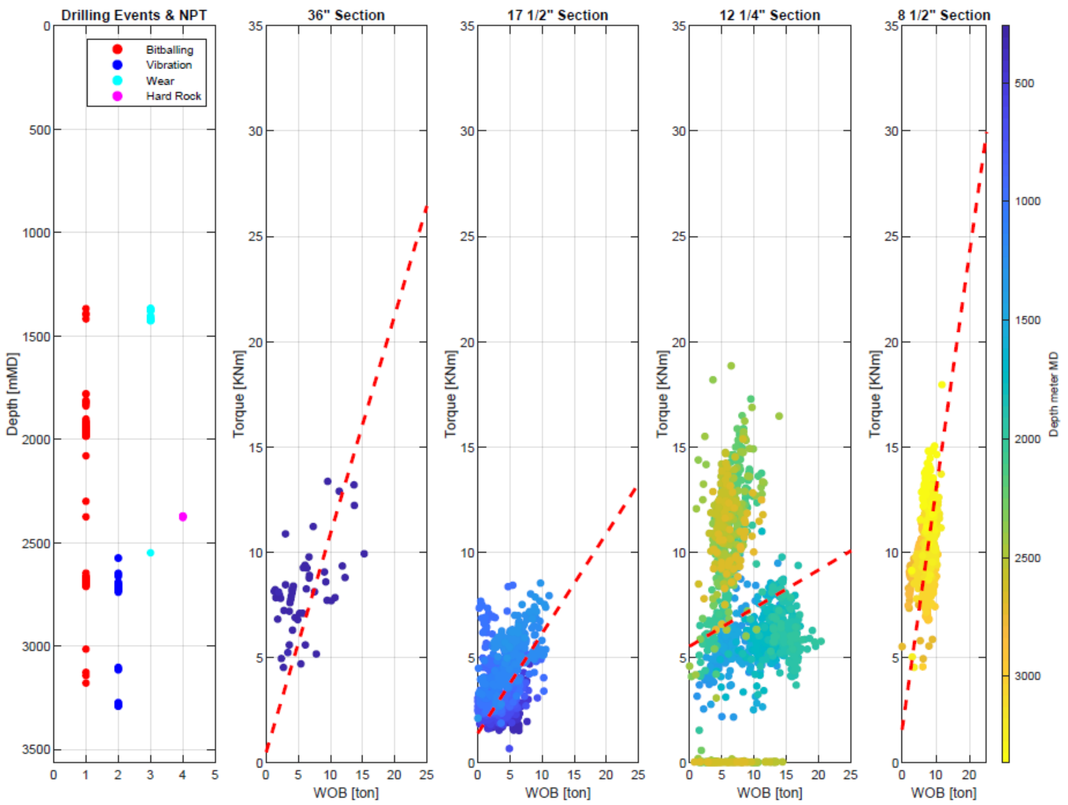


**Figure 109. Drilling Events Detection, WOB-Torque Cross plots vs Depth and Efficient Bit-lines (red lines), well 35/3-6.**

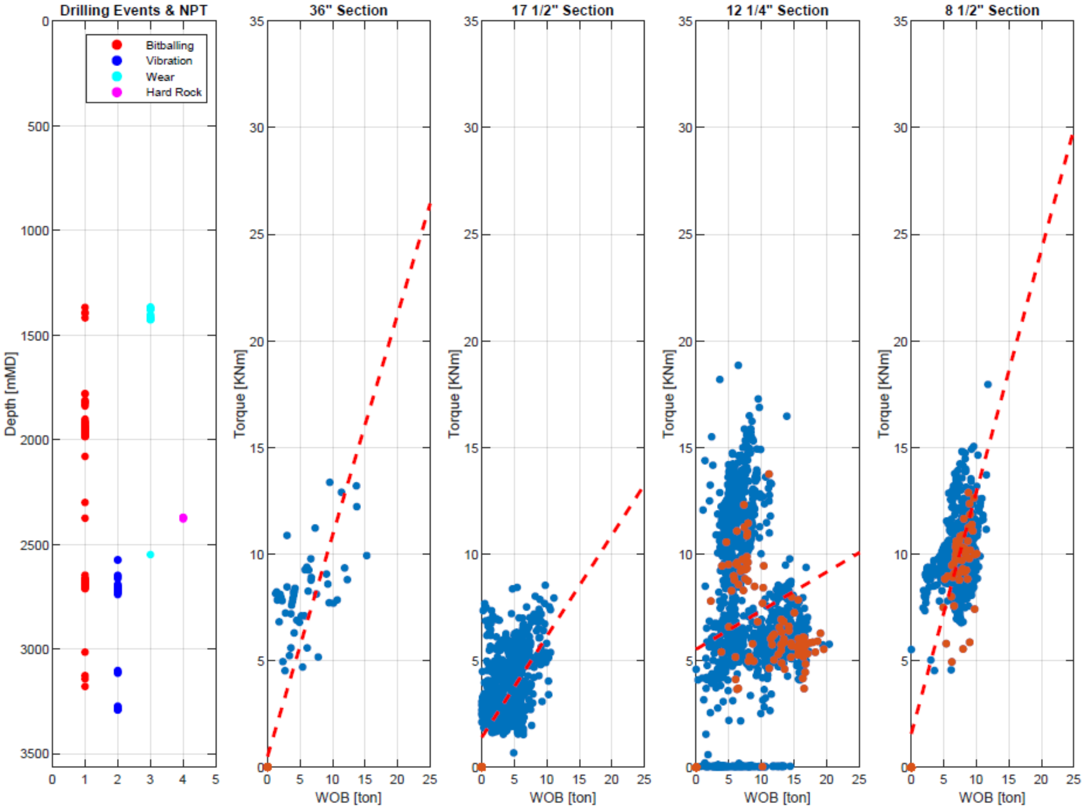


**Figure 110. Drilling Events Detection, WOB-Torque Cross plots, Efficient (blue) and Non-Efficient (orange) Depths and Efficient Bit-lines (red), well 35/3-6.**

## *8.3 Appendix C – Figures Multi-well Analysis*

*Well 15/12-18 S*

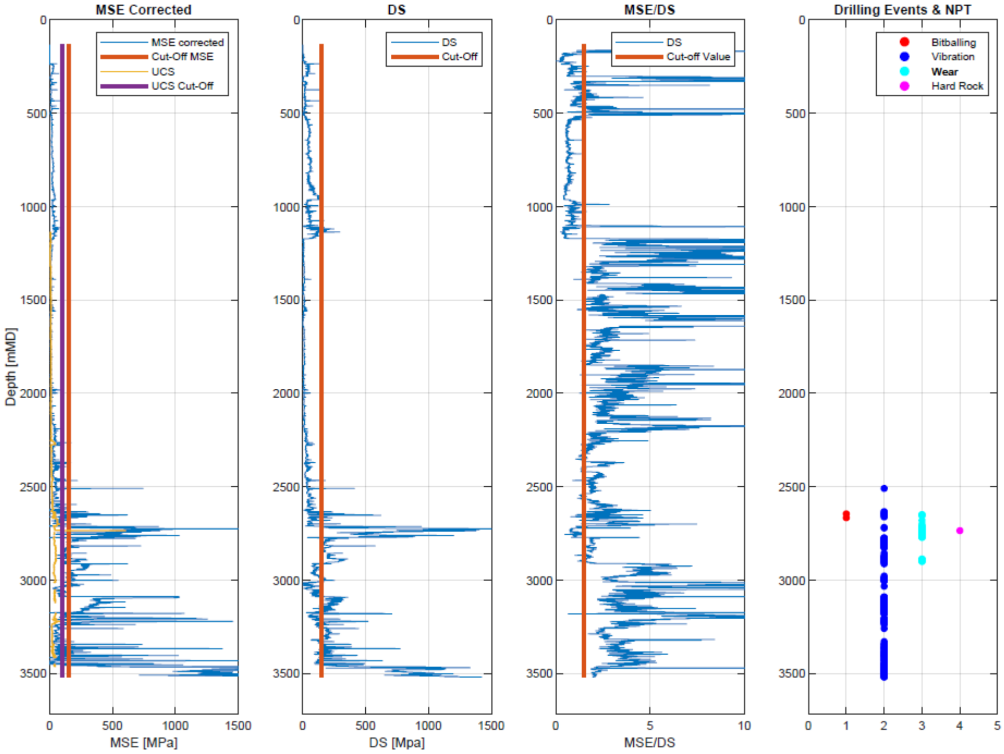


**Figure 111. Drilling Events Detection, Well 15/12-18 S.**

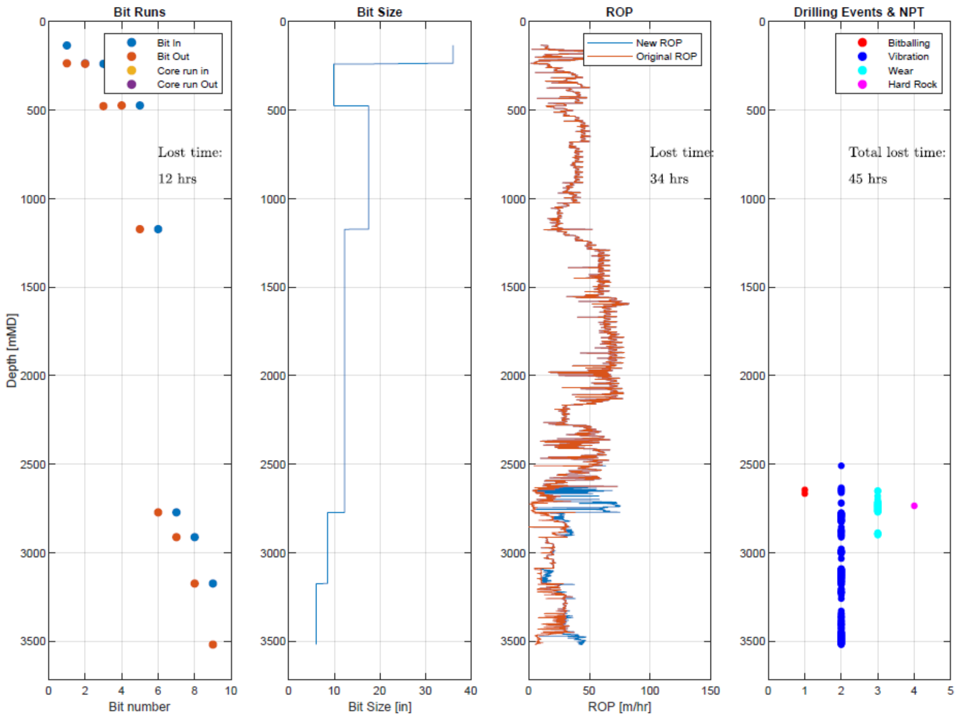


**Figure 112. Lost Time WOB-Method, Well 15/12-18 S.**

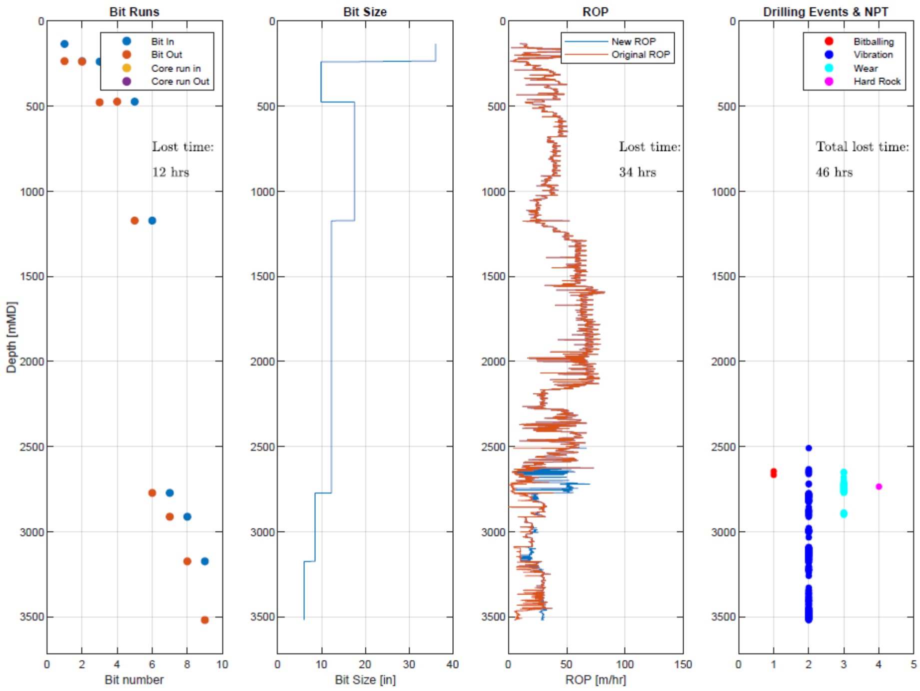


**Figure 113. Lost Time Torque-Method, Well 15/12-18 S.**

*Well 15/12-21*

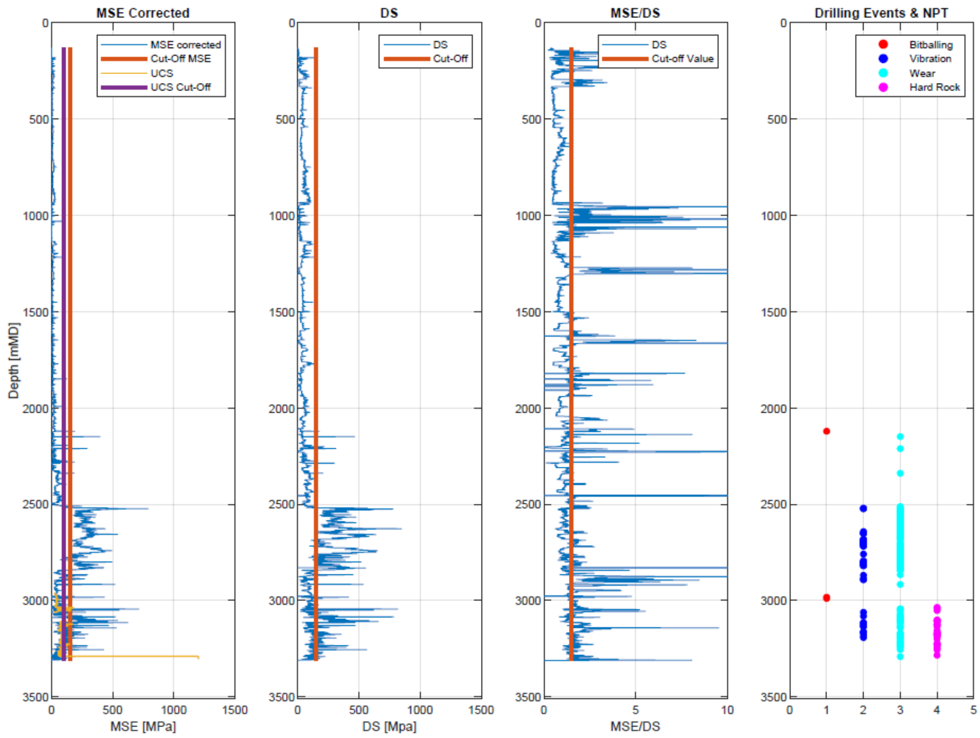


**Figure 114. Drilling Events Detection, Well 15/12-21.**

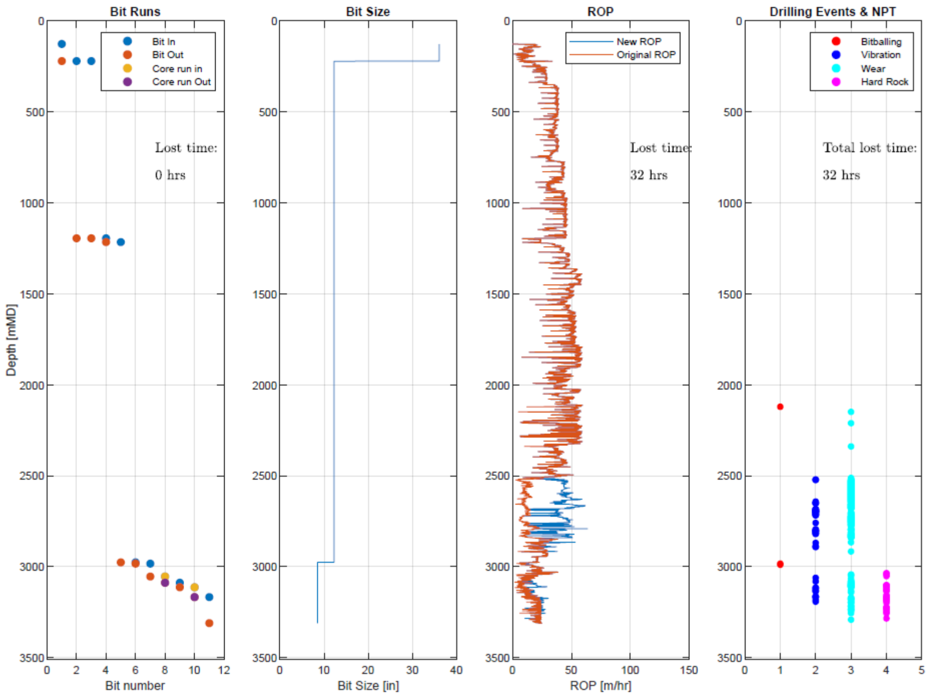


**Figure 115. Lost Time WOB-Method, Well 15/12-21.**

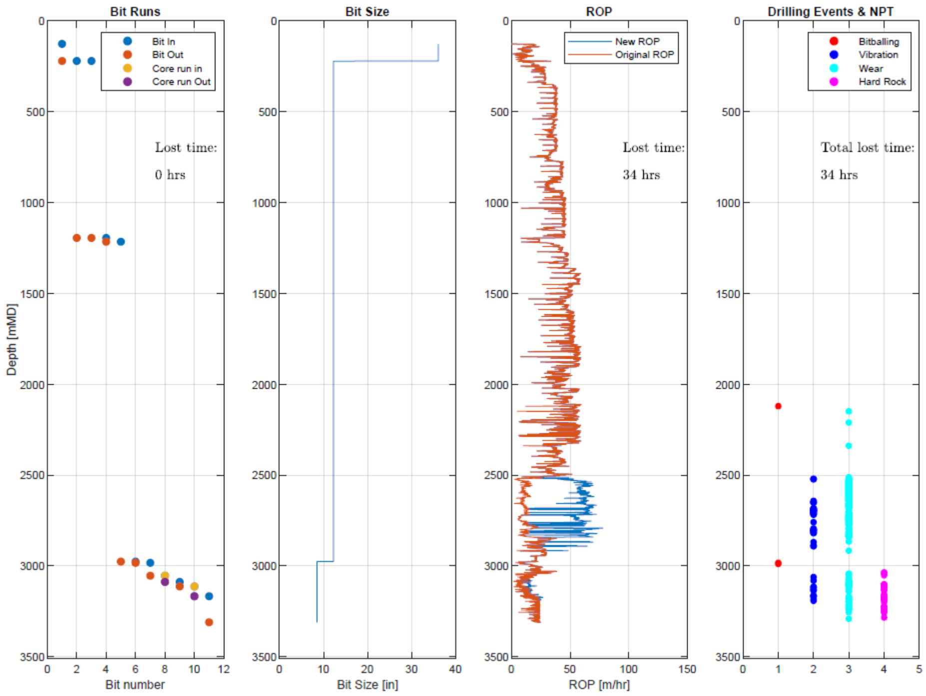


**Figure 116. Lost Time Torque-Method, Well 15/12-21.**

*Well 15/12-23*

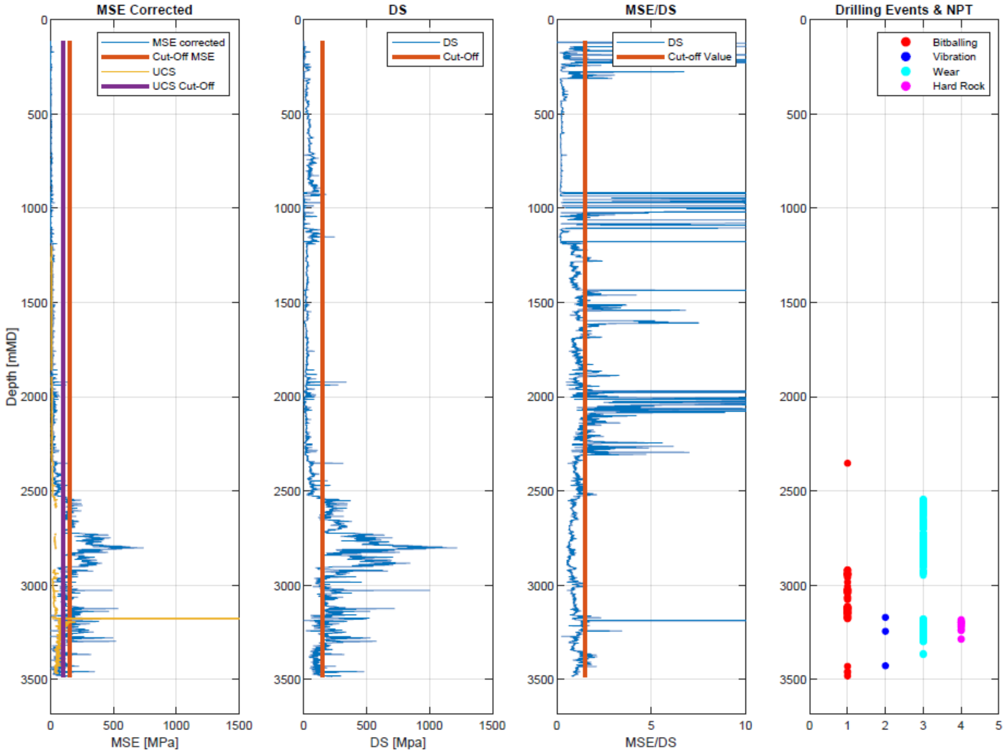


**Figure 117. Drilling Events Detection, Well 15/12-23.**

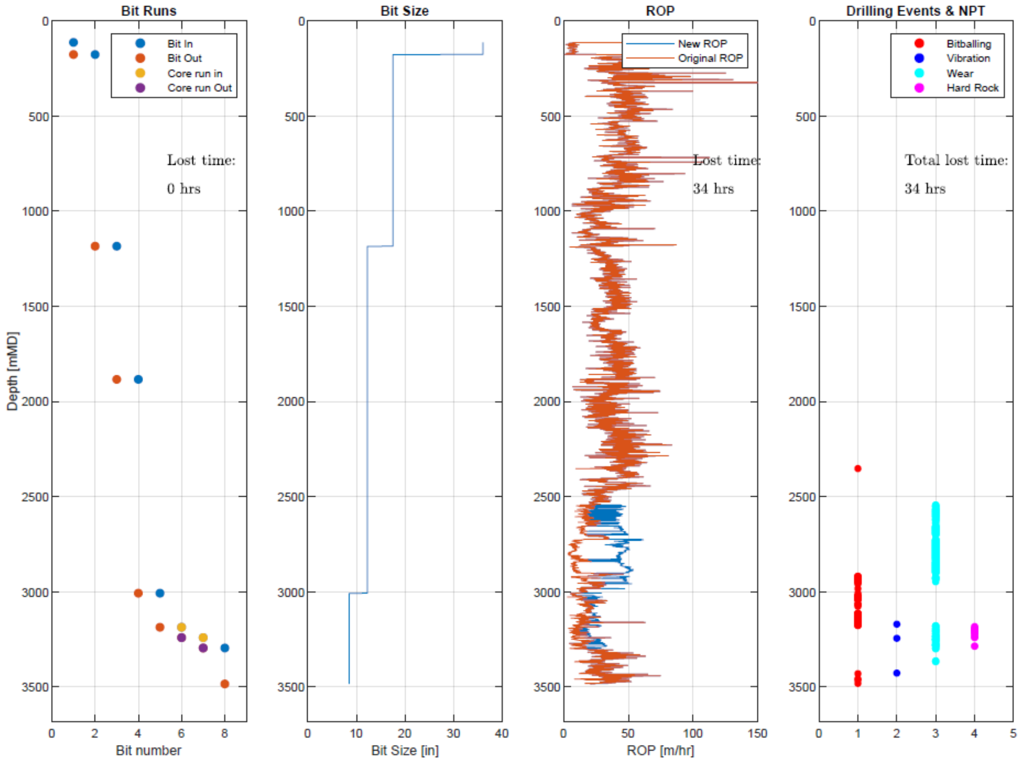


**Figure 118. Lost Time WOB-Method, Well 15/12-23.**

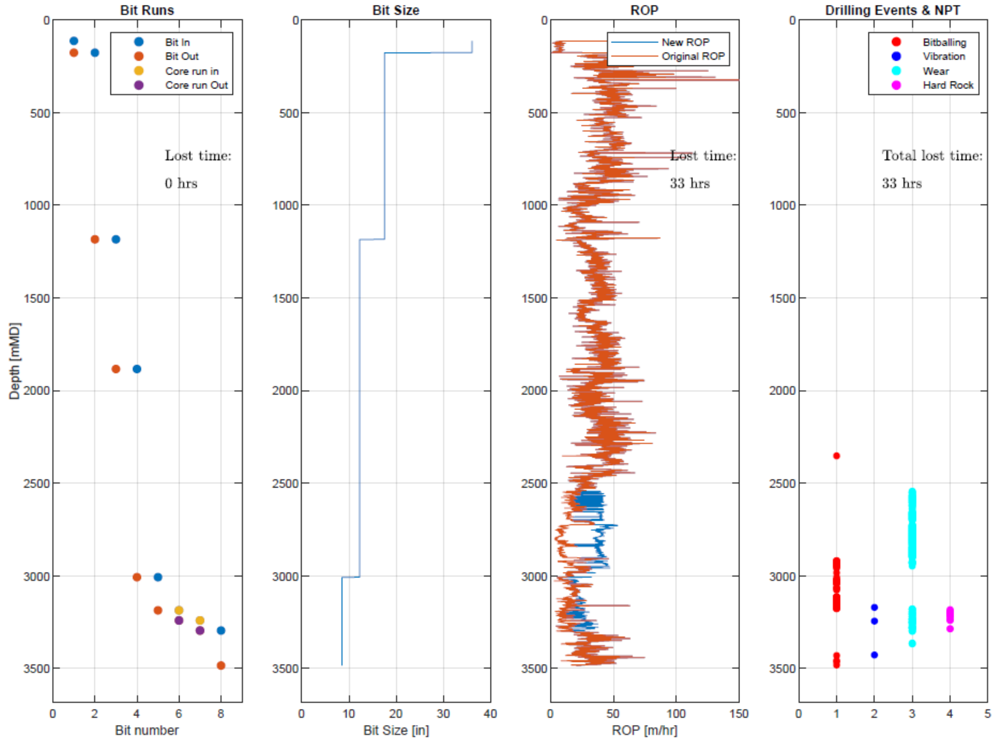


**Figure 119. Lost Time Torque-Method, Well 15/12-23.**

## *Well 33/6-3 S*

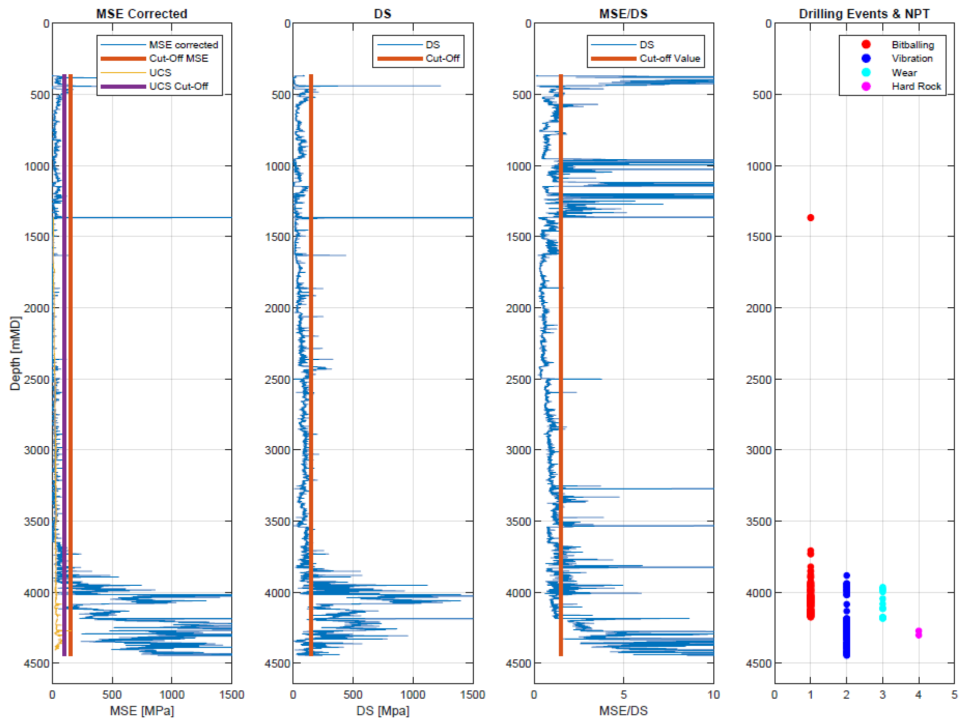


**Figure 120. Drilling Events Detection, Well 33/6-3 S.**

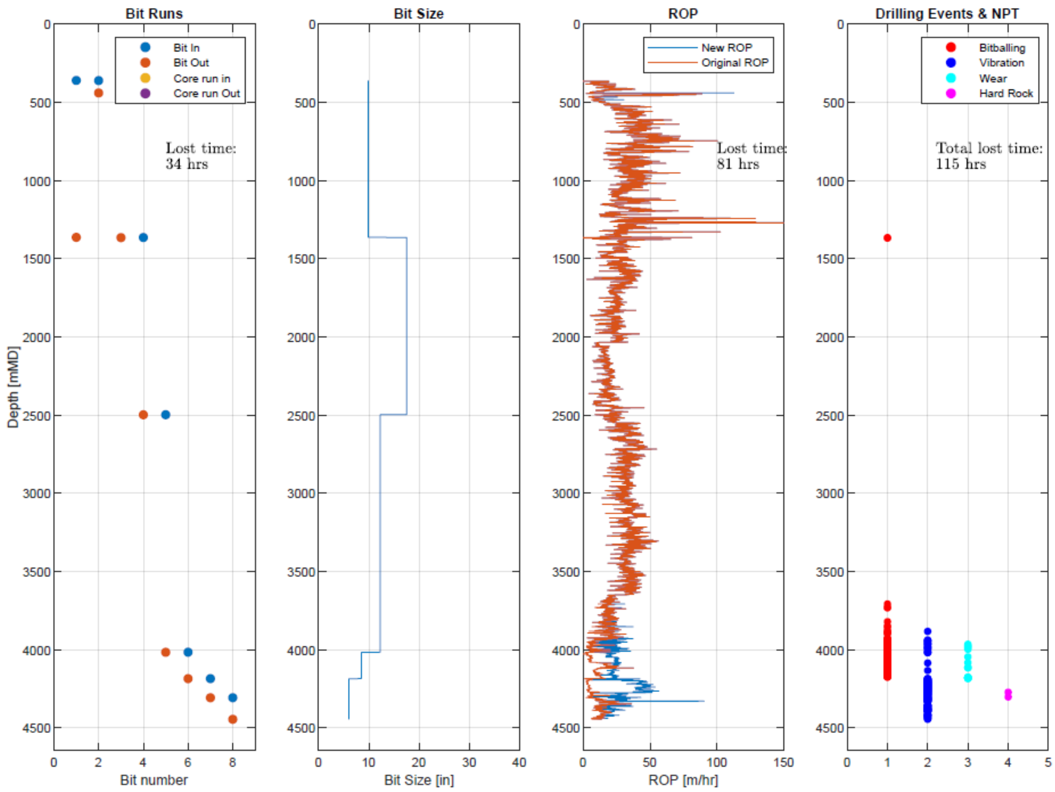


**Figure 121. Lost Time WOB-Method, Well 33/6-3 S.**

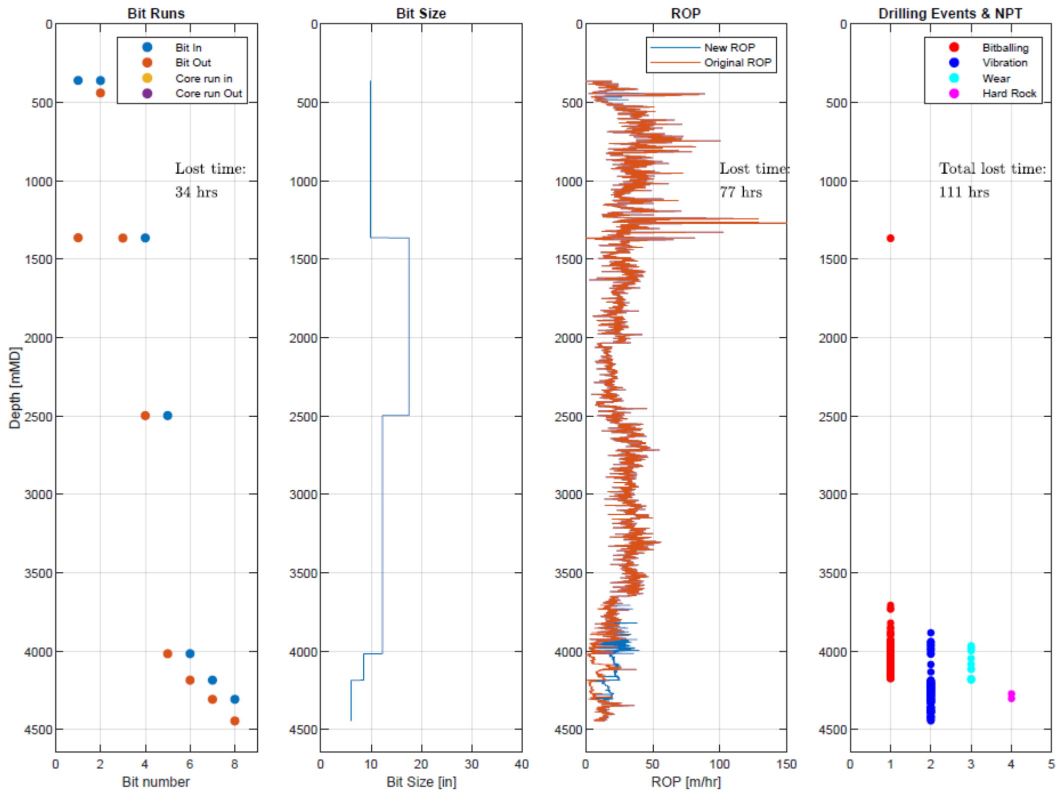


**Figure 122. Lost Time Torque-Method, Well 33/6-3 S.**

*Well 35/8-4*

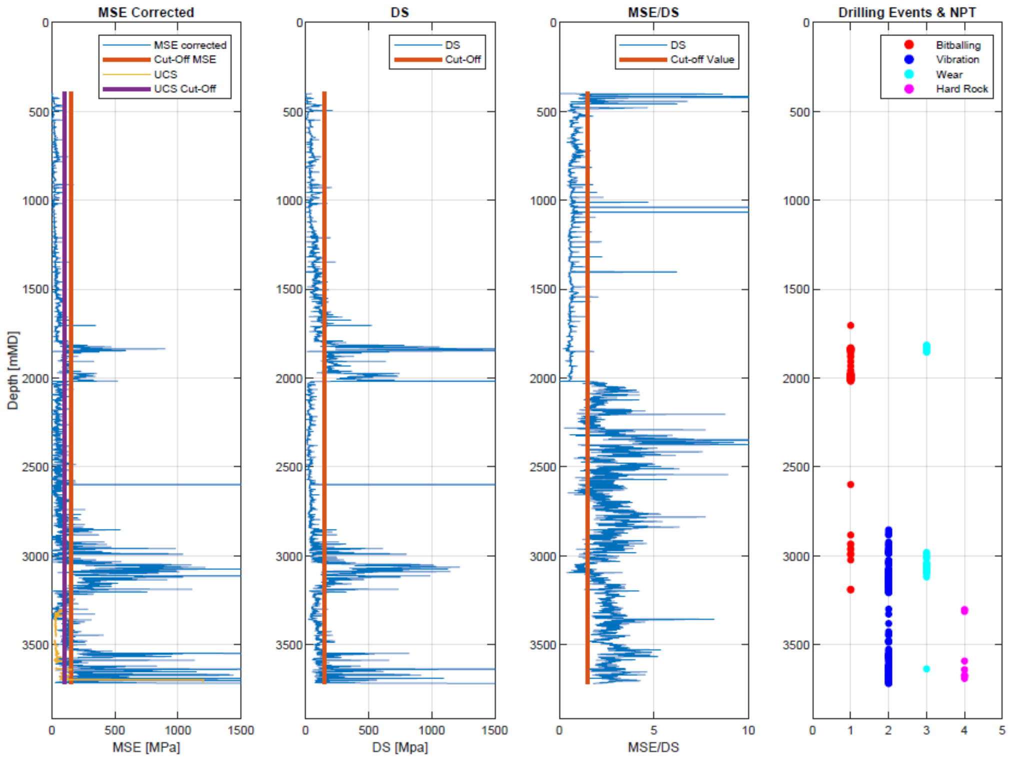


**Figure 123. Drilling Events Detection, well 35/8-4.**

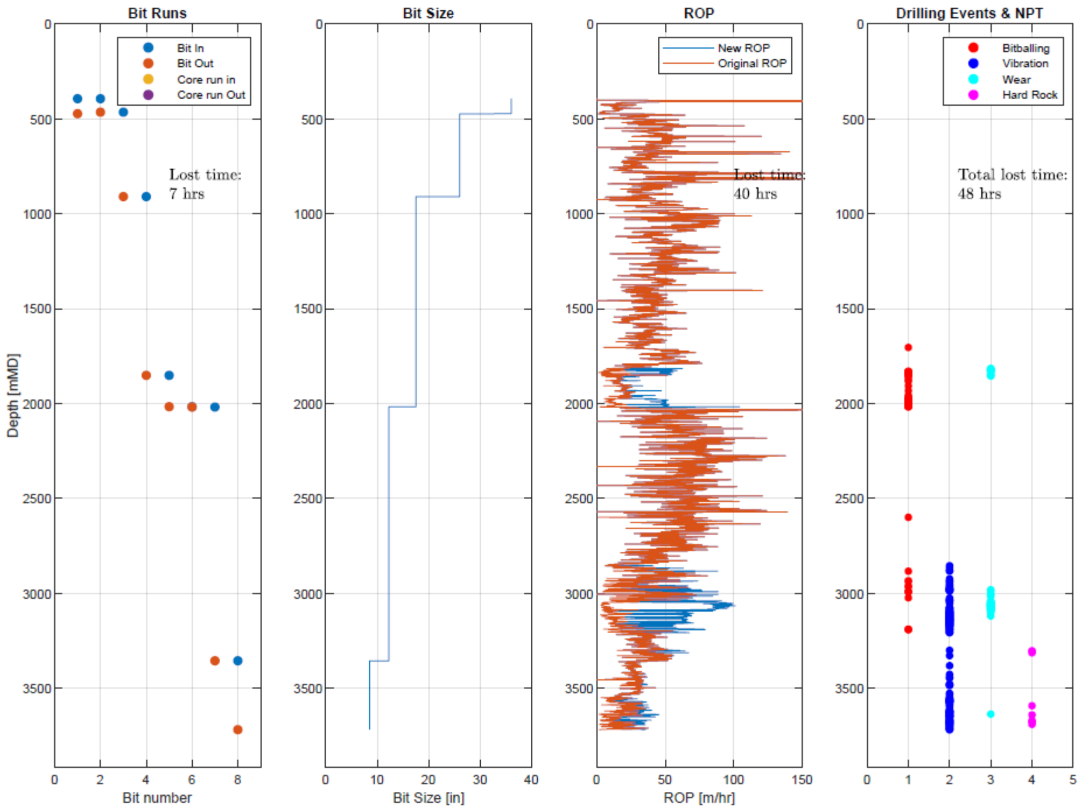


**Figure 124. Lost Time WOB-Method, Well 35/8-4.**

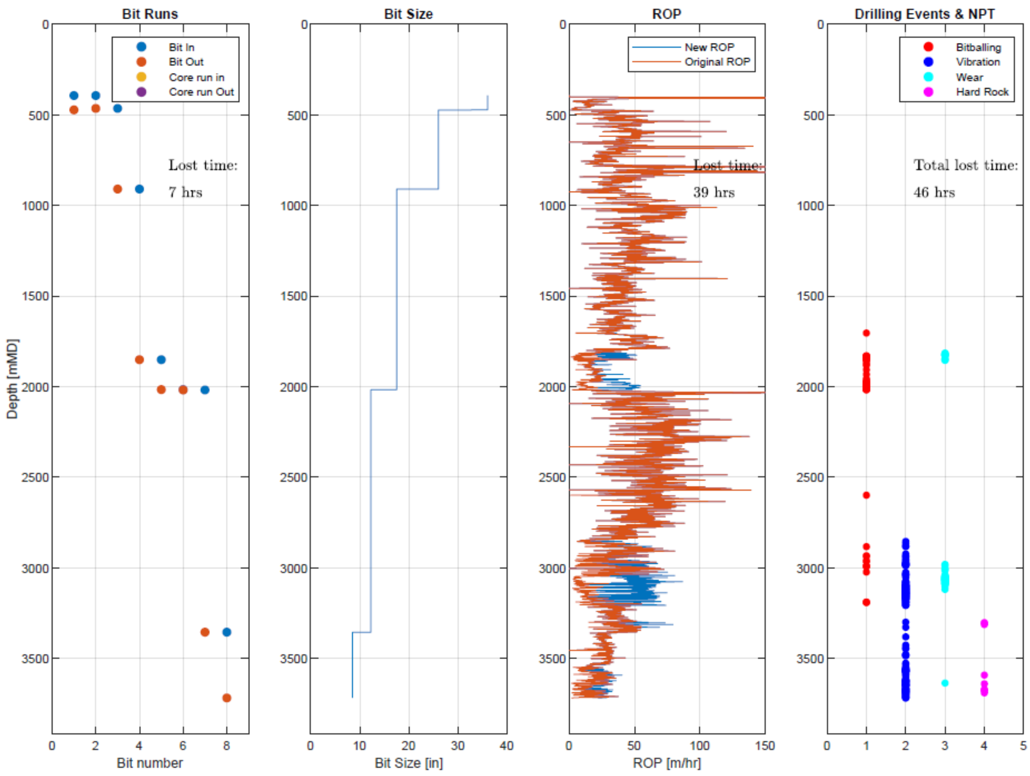


**Figure 125. Lost Time Torque-Method, Well 35/8-4.**

*Well 35/8-5 S*

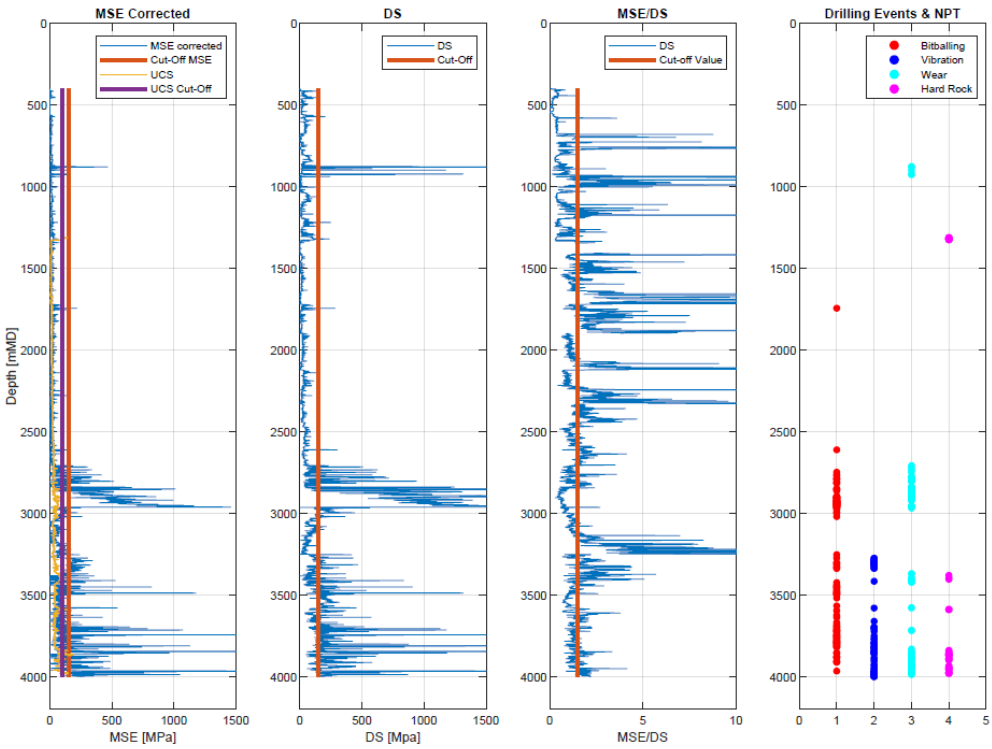


**Figure 126. Drilling Events Detection, Well 35/8-5 S.**

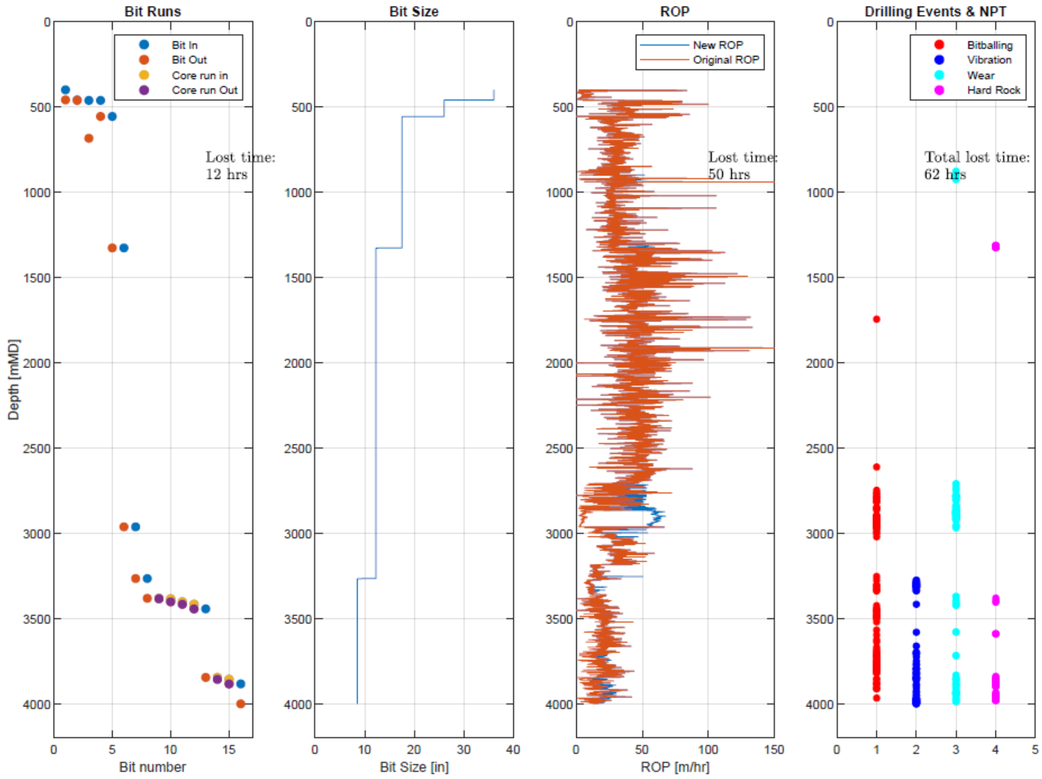


**Figure 127. Lost Time WOB-Method, Well 35/8-5 S.**

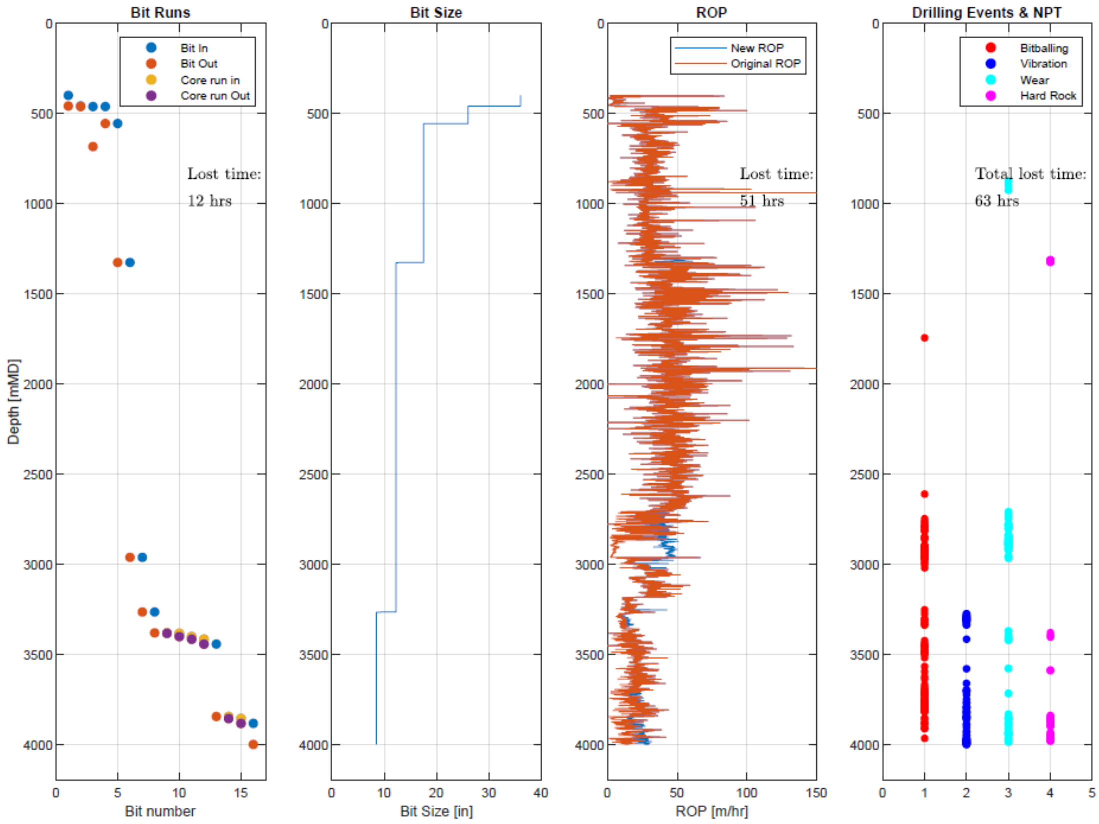


**Figure 128. Lost Time Torque-Method, Well 35/8-5 S.**

*Well 35/9-8*

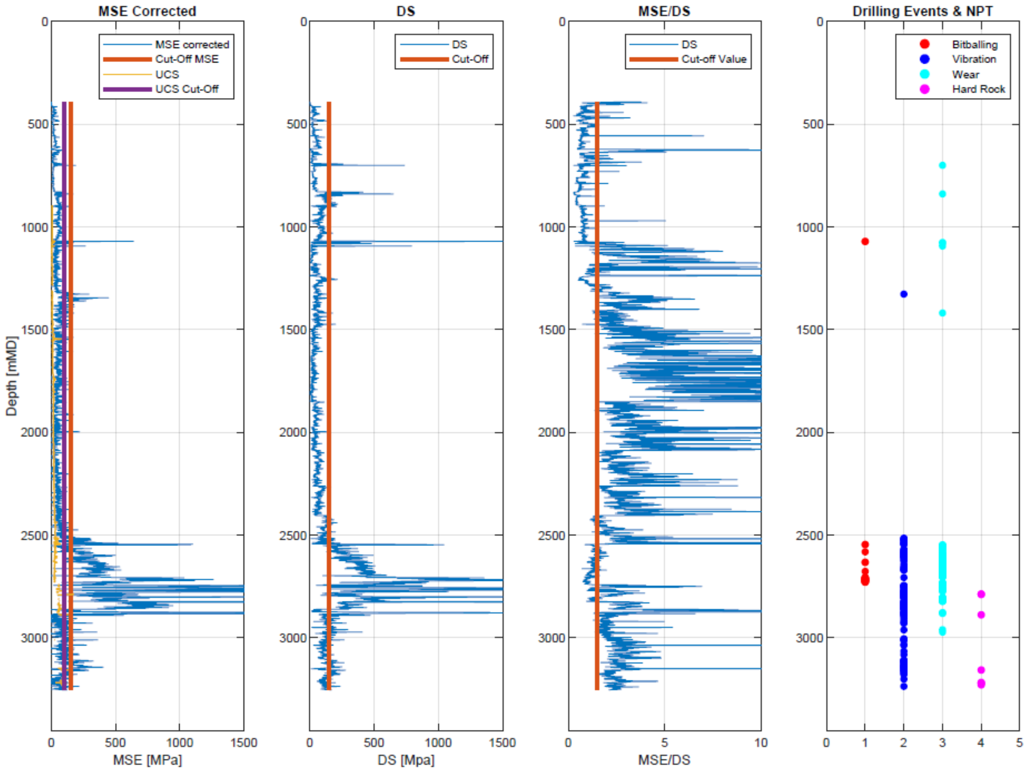


**Figure 129. Drilling Events Detection, Well 35/9-8.**

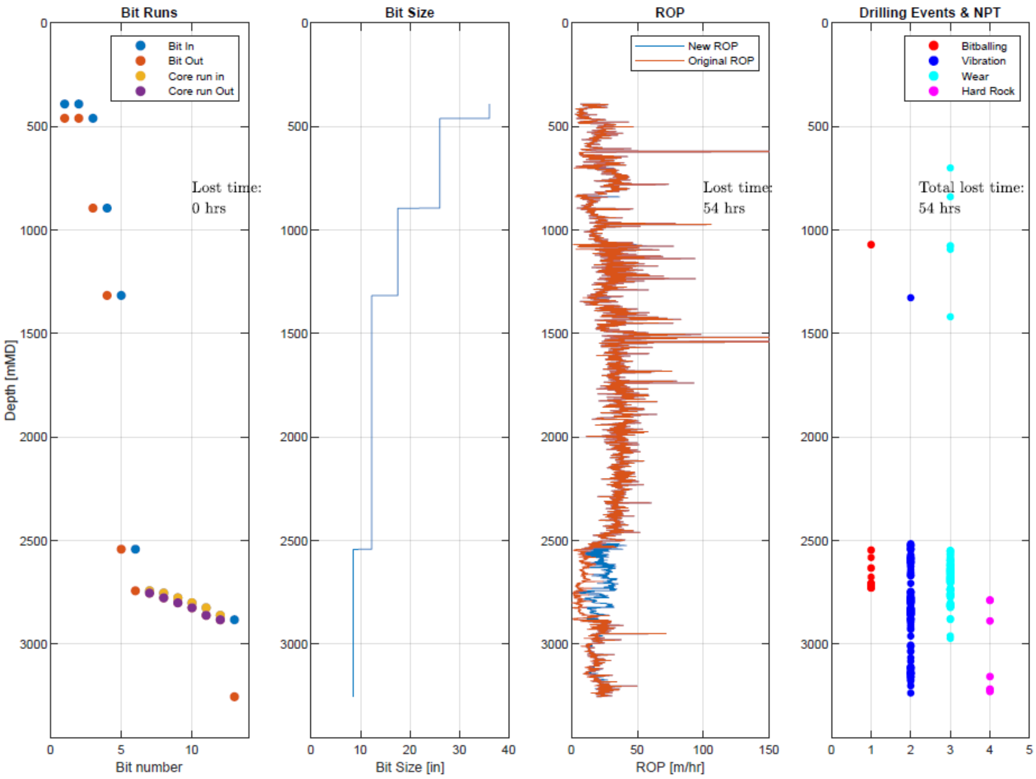


**Figure 130. Lost Time WOB-Method, Well 35/9-8.**

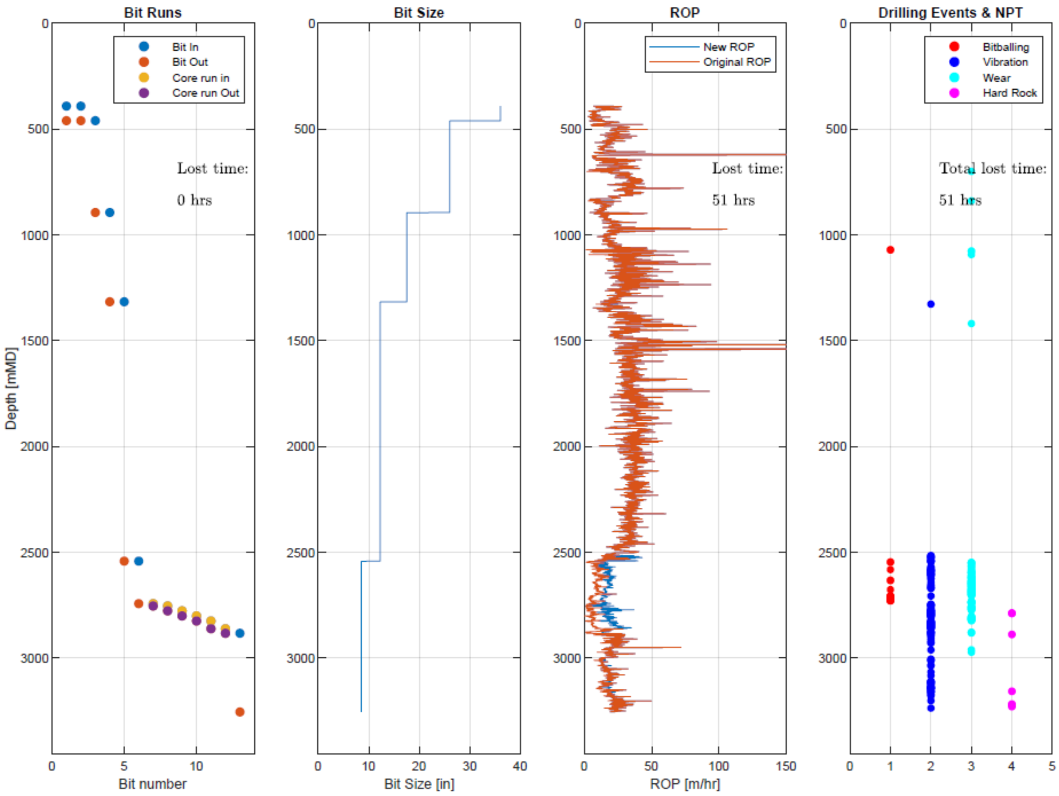


**Figure 131. Lost Time Torque-Method, Well 35/9-8.**

*Well 35/9-11 S*

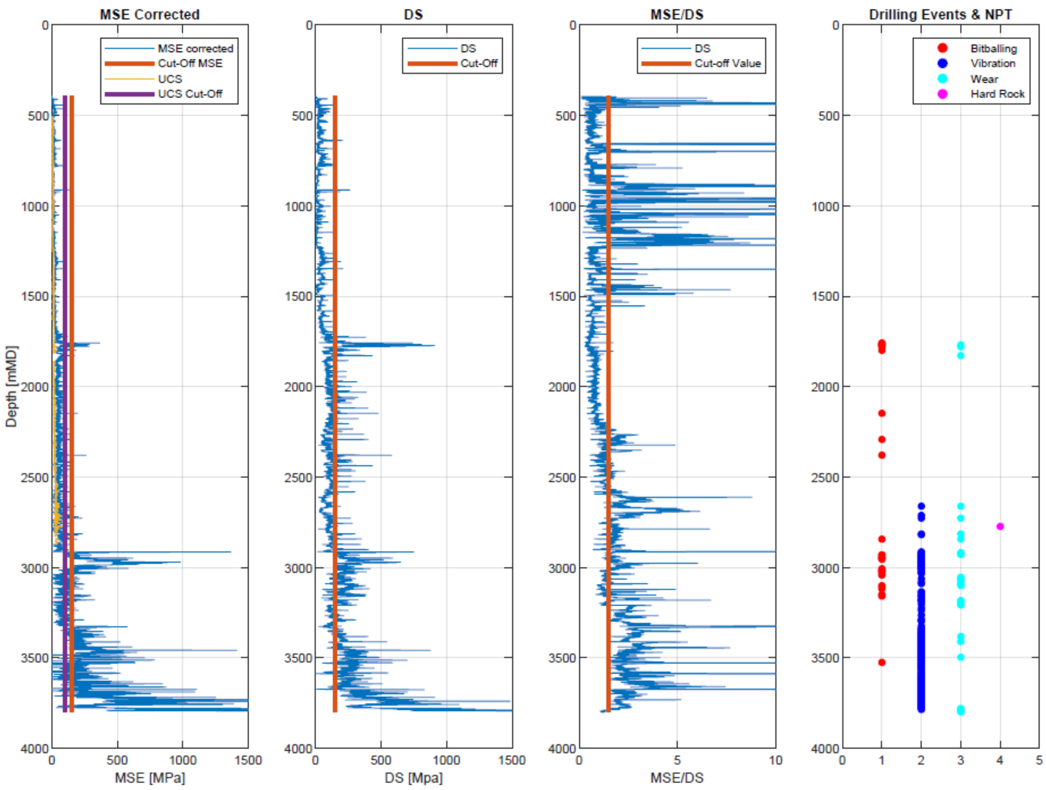


**Figure 132. Drilling Events Detection, Well 35/9-11 S.**

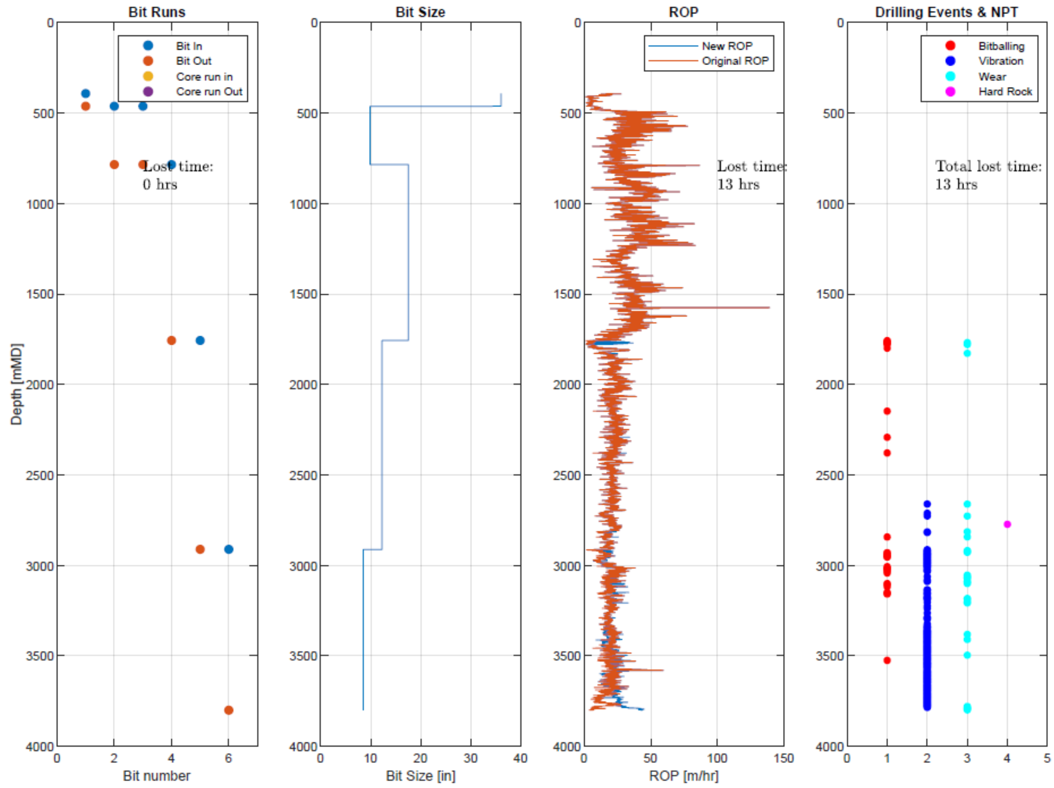


**Figure 133. Lost Time WOB-Method, Well 35/9-11 S.**

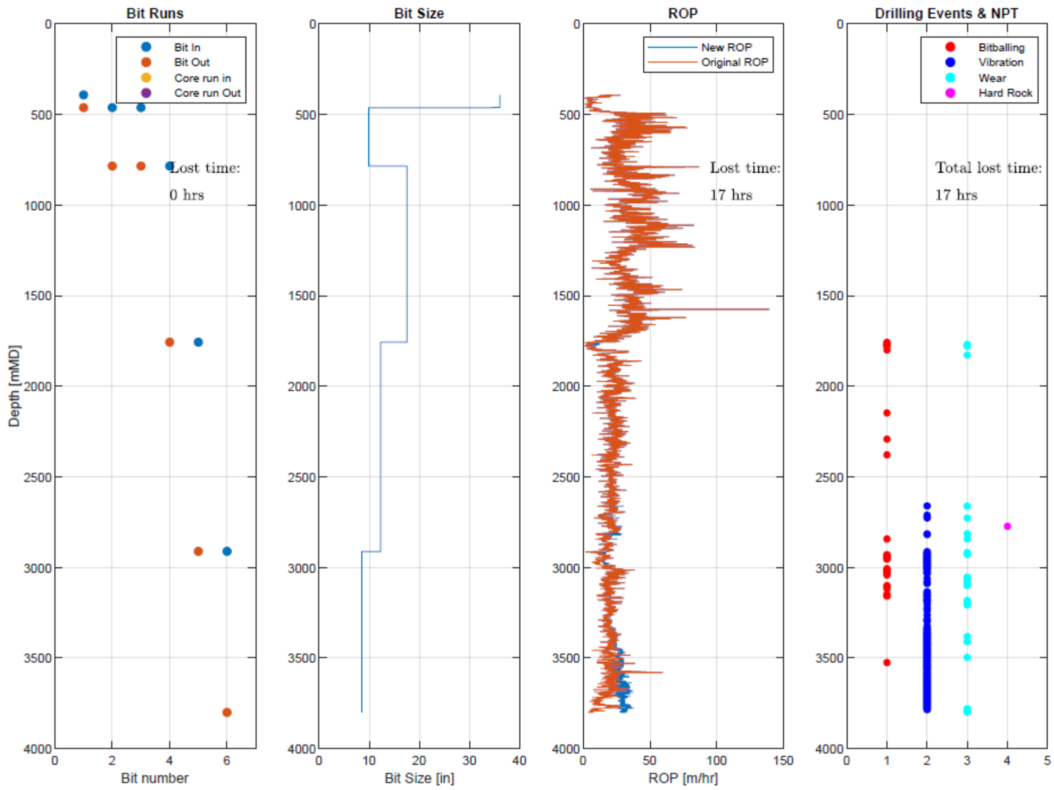


**Figure 134. Lost Time Torque-Method, Well 35/9-11 S.**

*Well 6406/1-1*

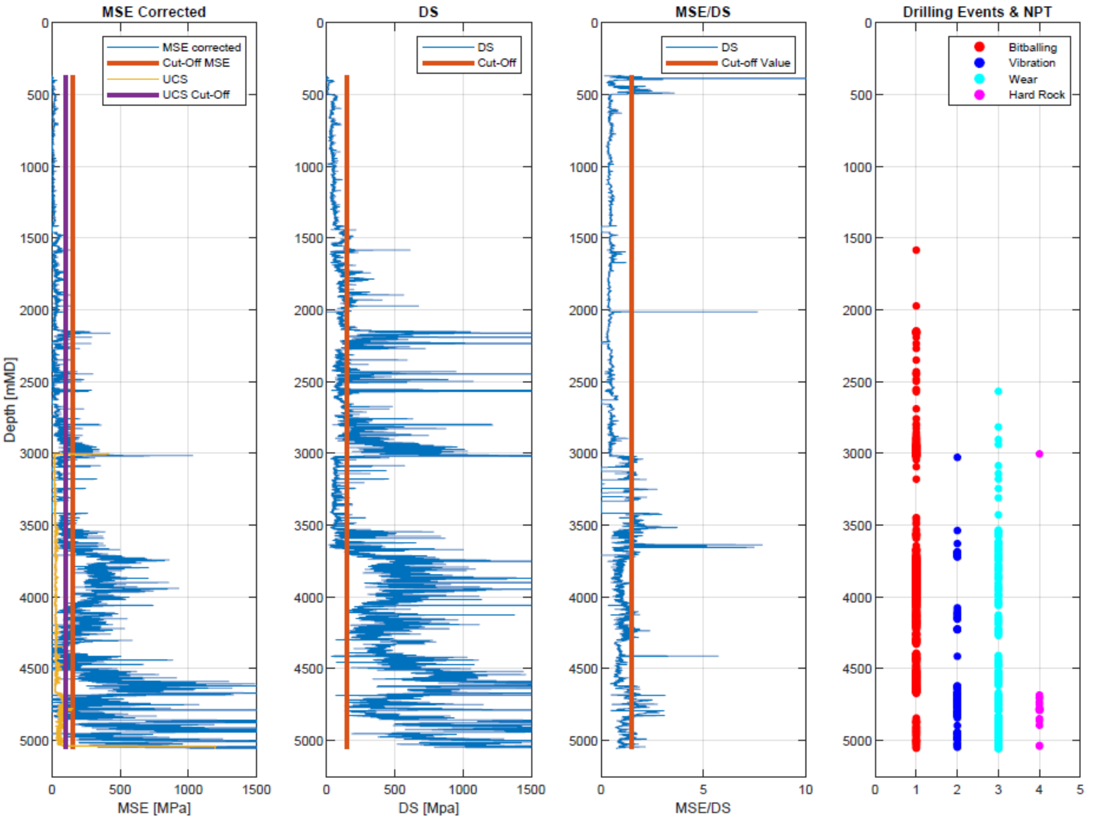


**Figure 135. Drilling Events Detection, well 6406/1-1.**

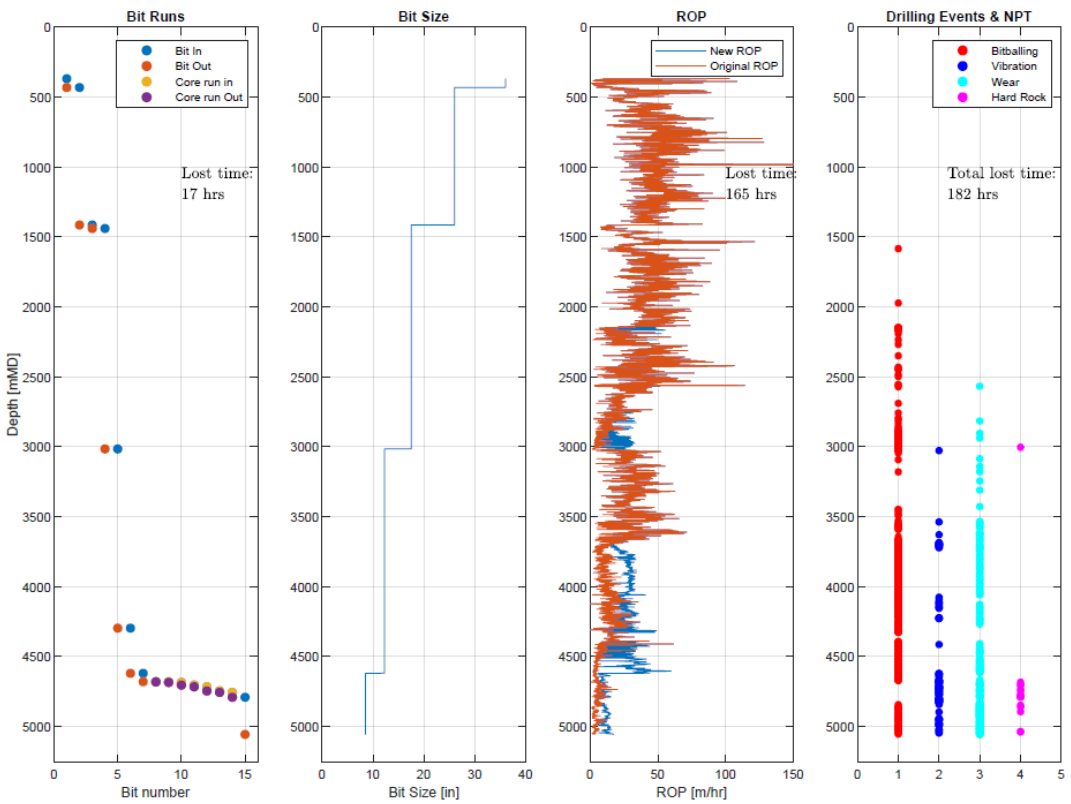


**Figure 136. Lost Time WOB-Method, Well 6406/1-1.**

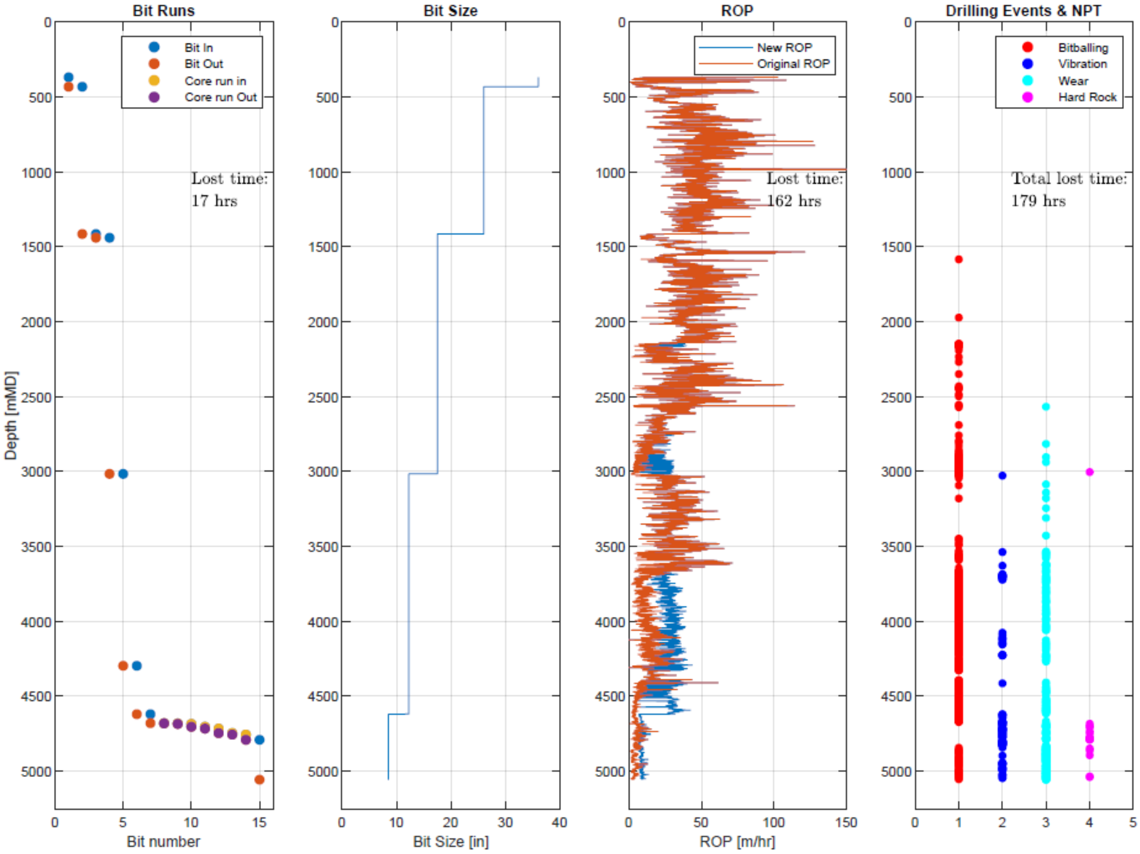


**Figure 137. Lost Time Torque-Method, Well 6406/1-1.**

*Well 6406/2-5*

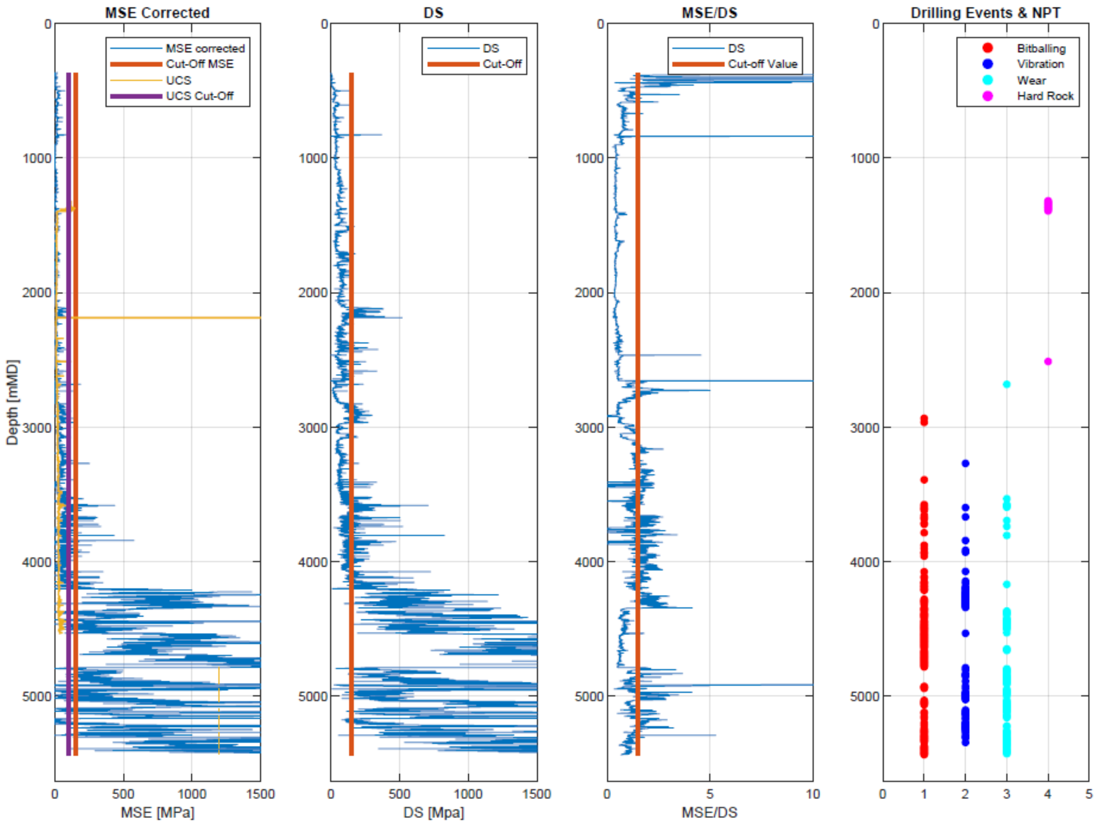


**Figure 138. Drilling Events Detection, well 6406/2-5.**

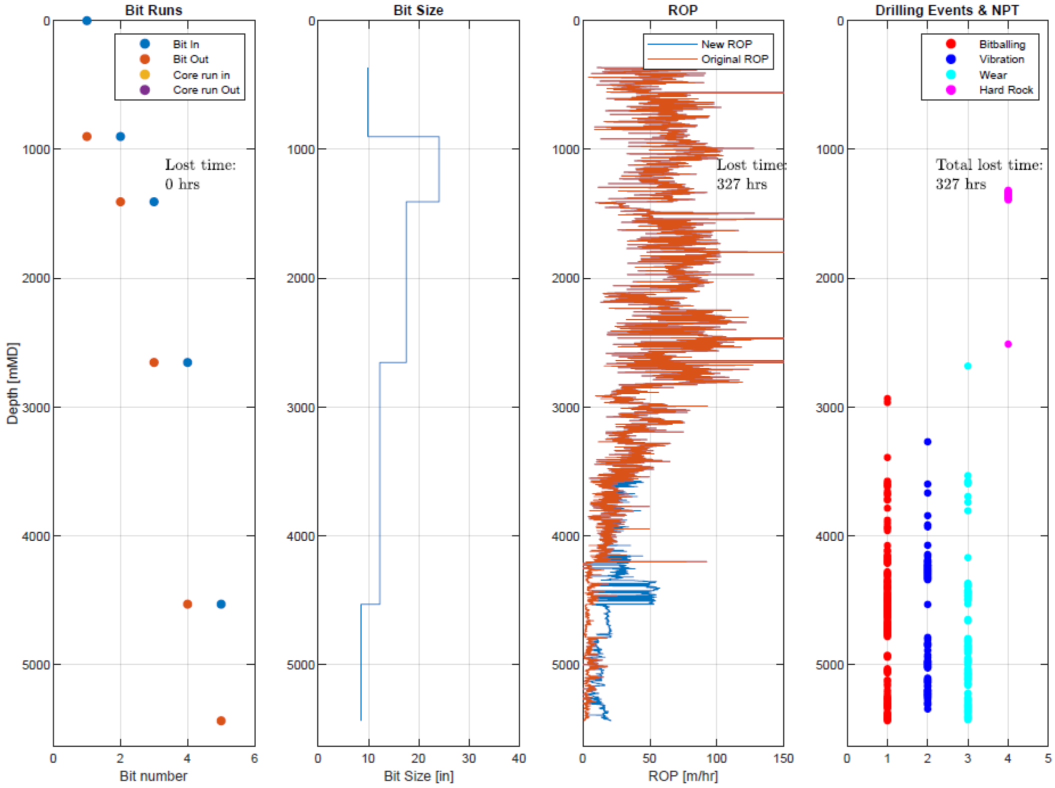


**Figure 139. Lost Time WOB-Method, Well 6406/2-5.**

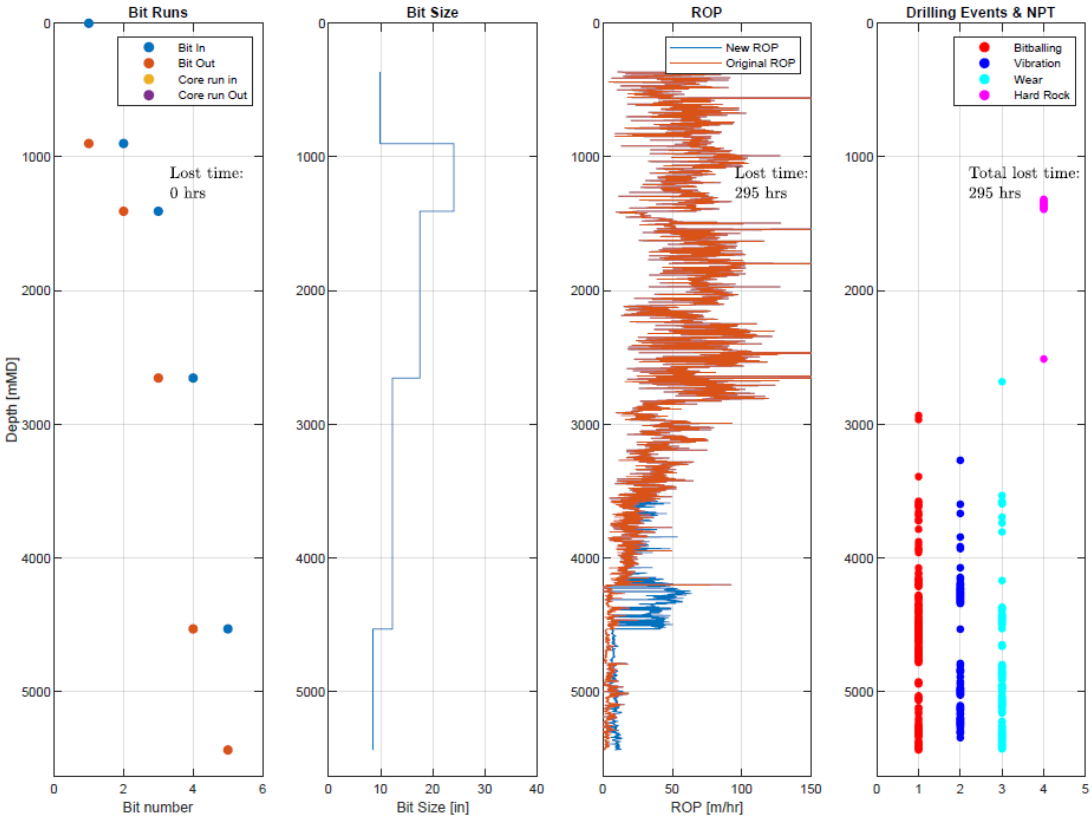


**Figure 140. Lost Time Torque-Method, Well 6406/2-5.**

*Well 6406/2-7*

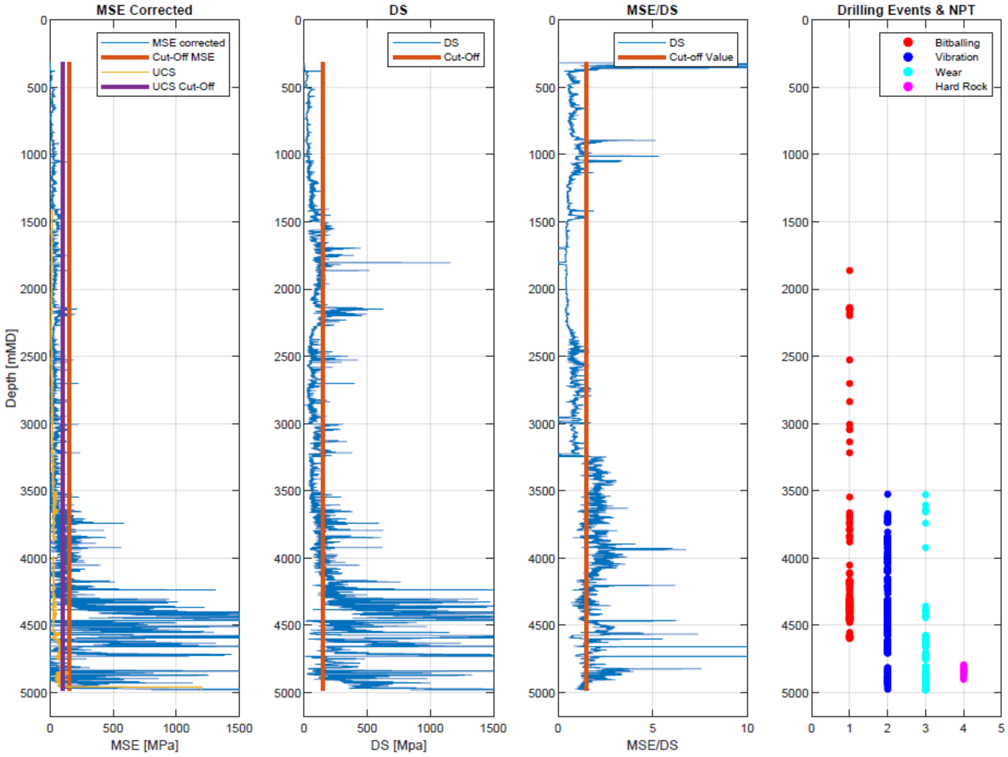


**Figure 141. Drilling Events Detection, Well 6406/2-7.**

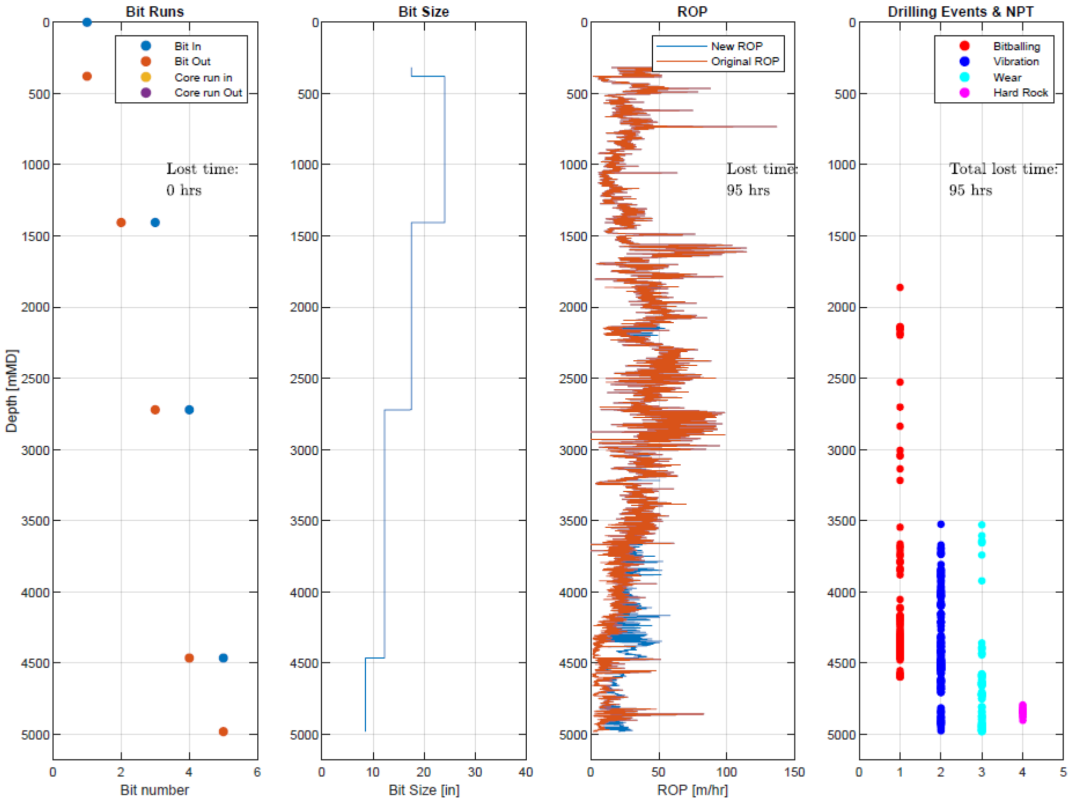


**Figure 142. Lost Time WOB-Method, Well 6406/2-7.**

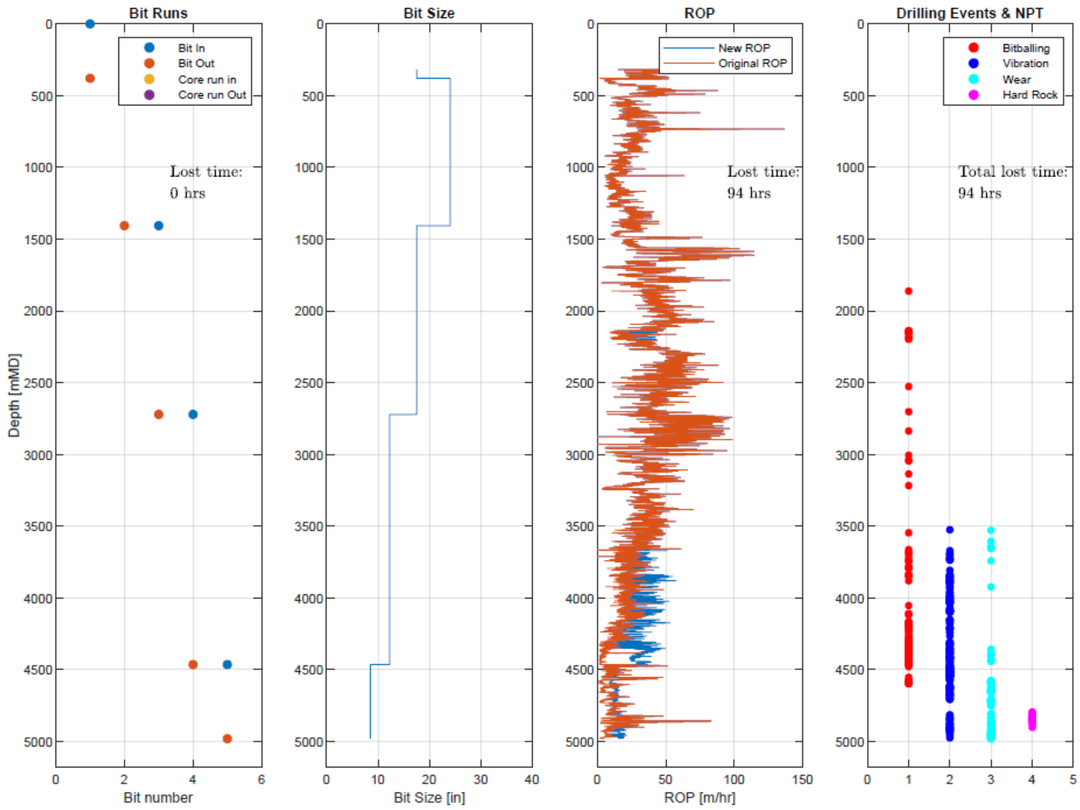


**Figure 143. Lost Time Torque-Method, Well 6406/2-7.**

## *Well 6407/8-4 S*

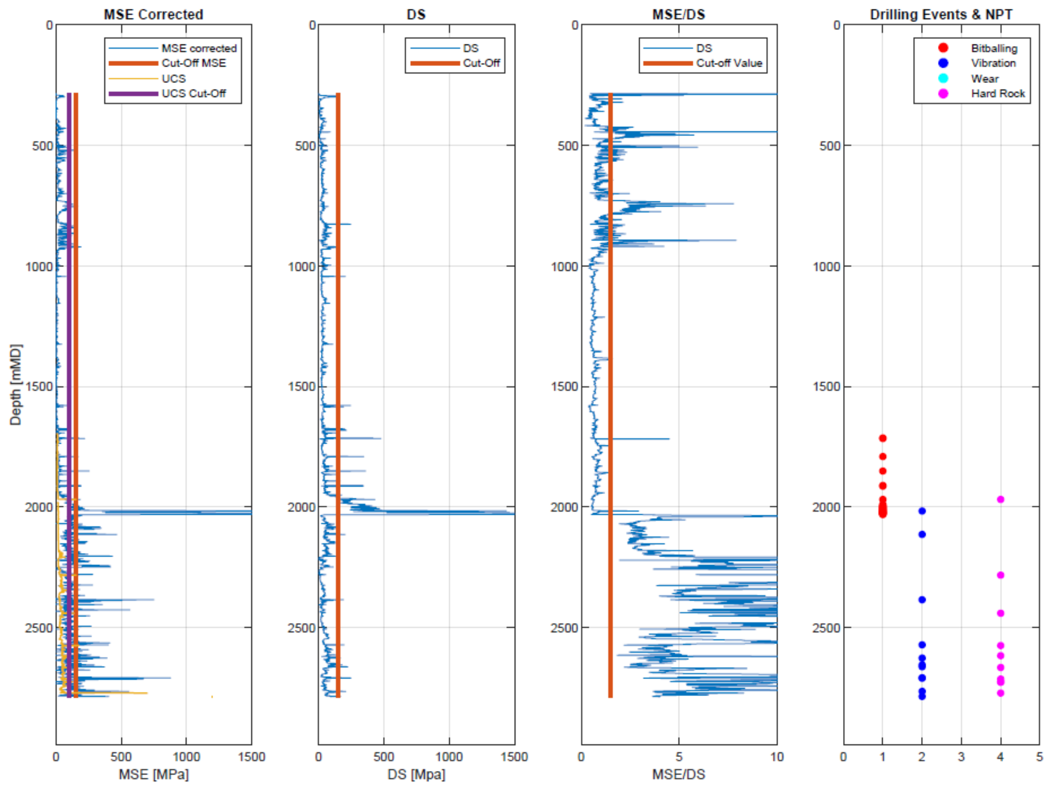


**Figure 144. Drilling Events Detection, Well 6407/8-4 S.**

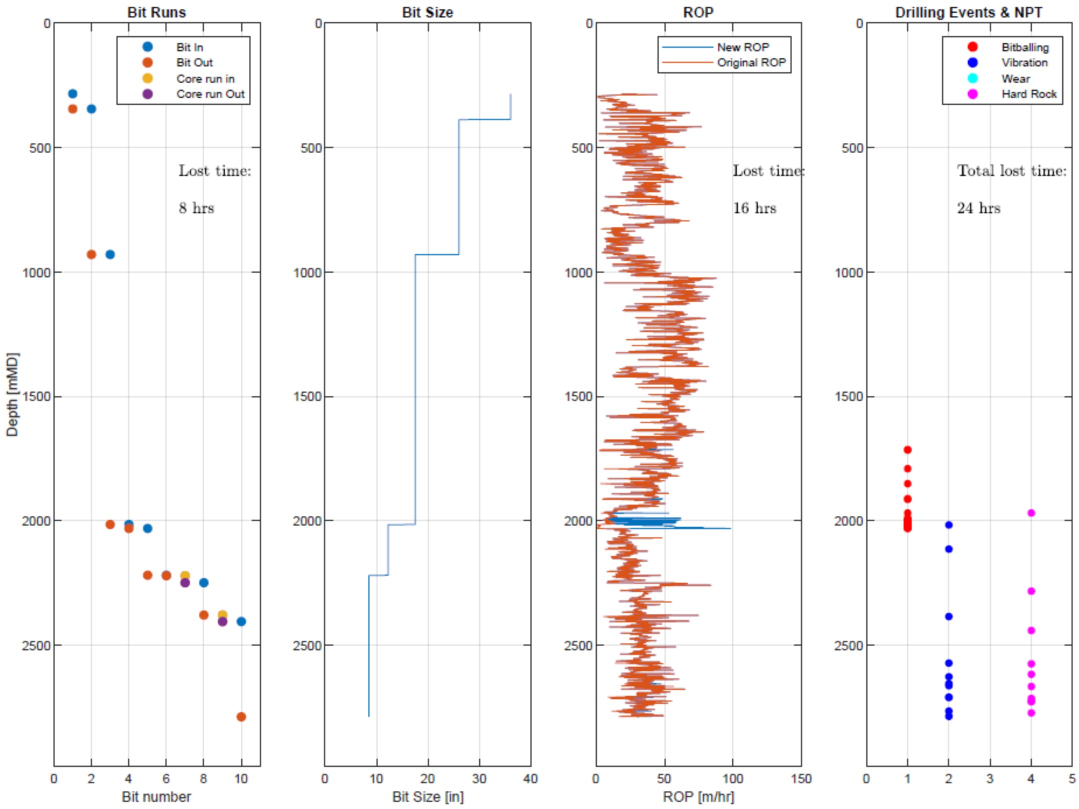


**Figure 145. Lost Time WOB-Method, Well 6407/8-4 S.**

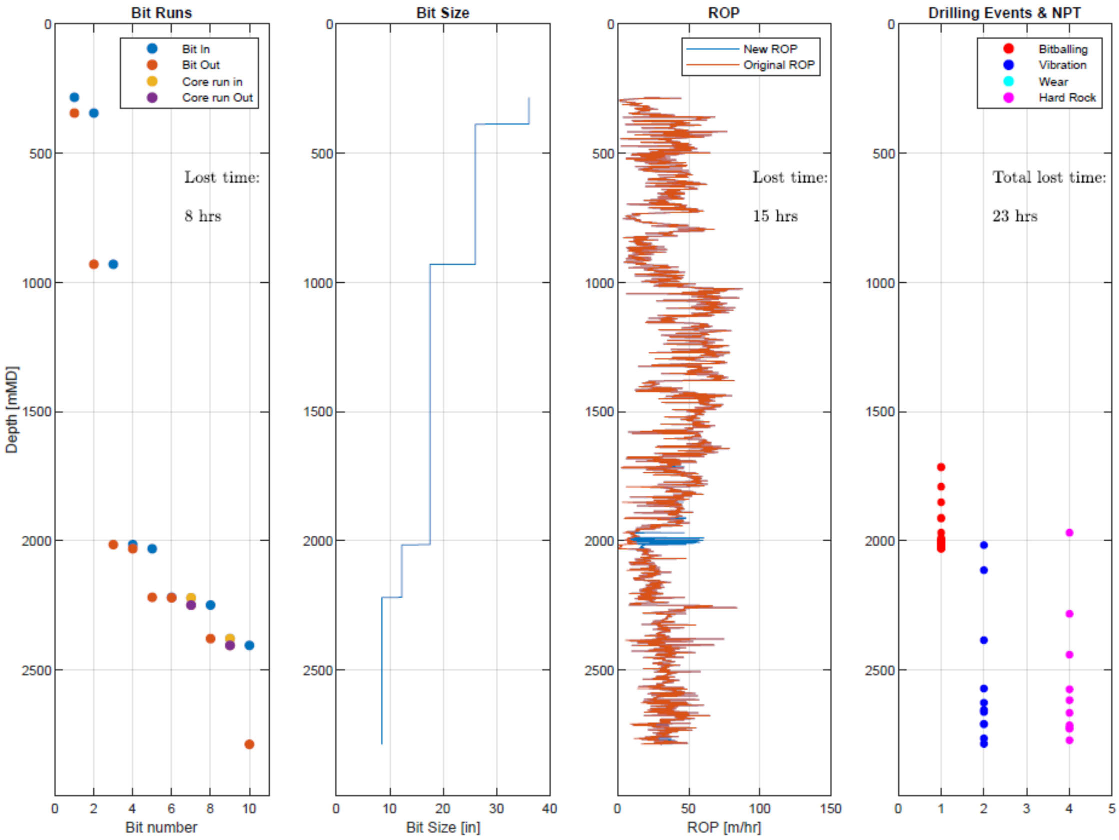


**Figure 146. Lost Time Torque-Method, Well 6407/8-4 S.**

*Well 6407/8-5 S*

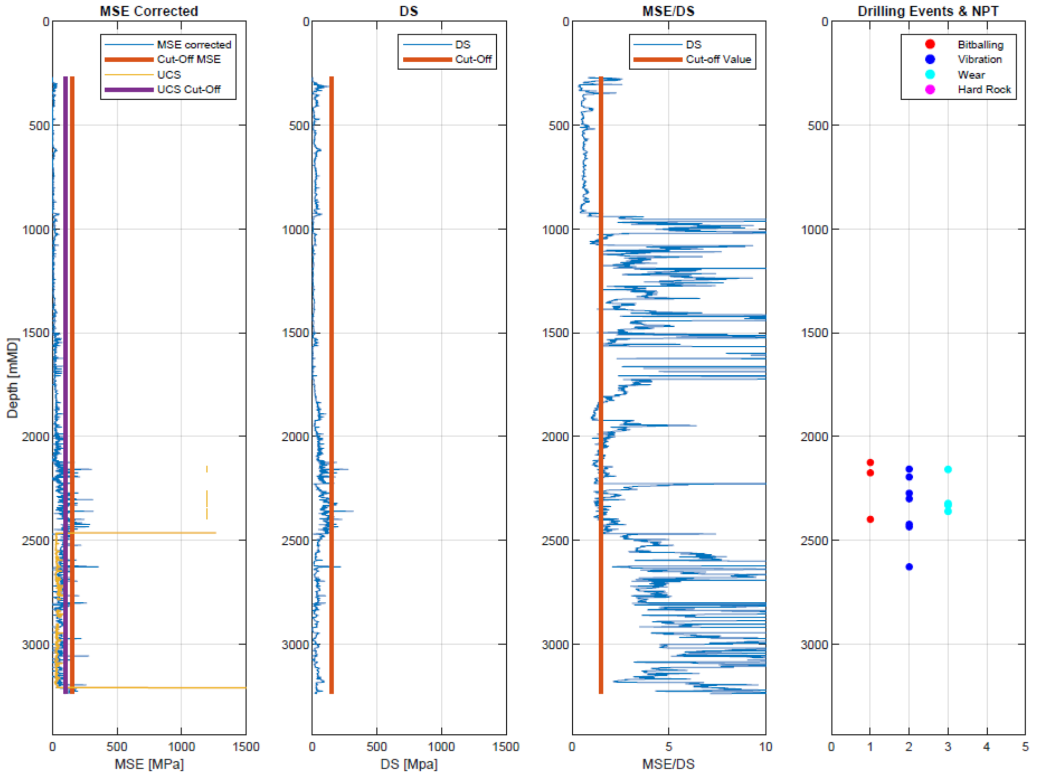


**Figure 147. Drilling Events Detection, well 6407/8-5 S.**

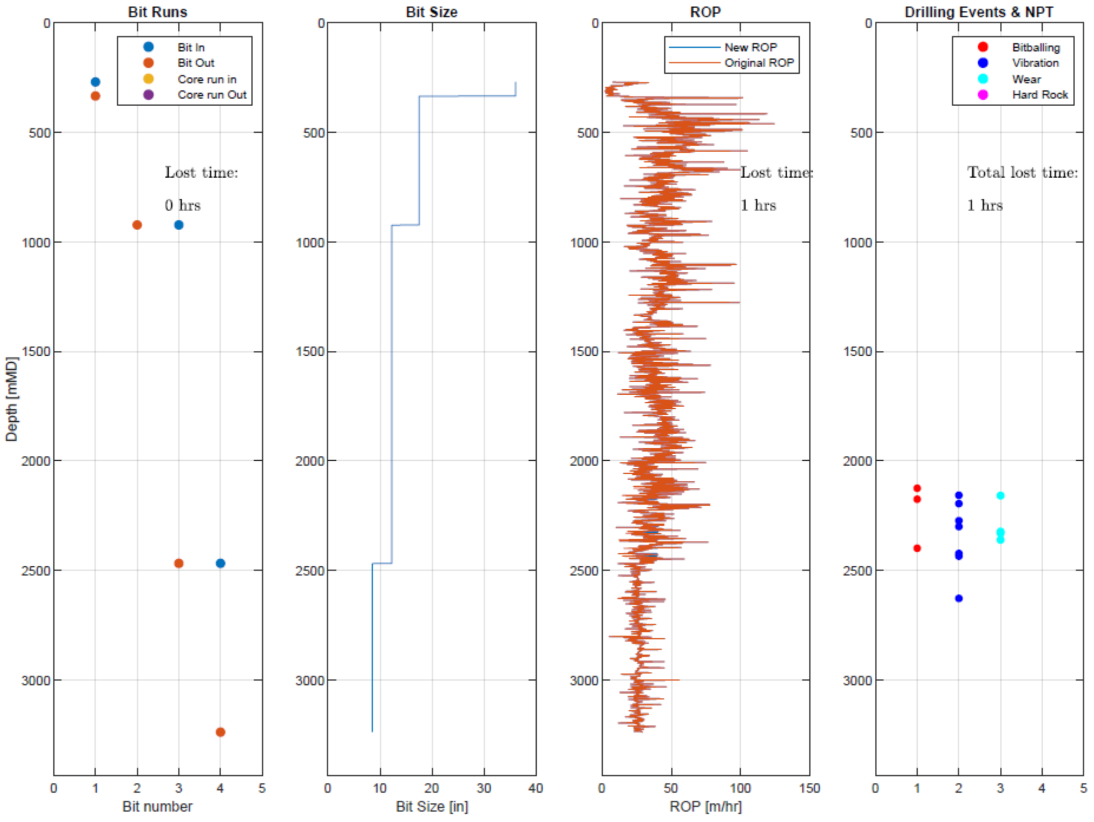


**Figure 148. Lost Time WOB-Method, Well 6407/8-5 S.**

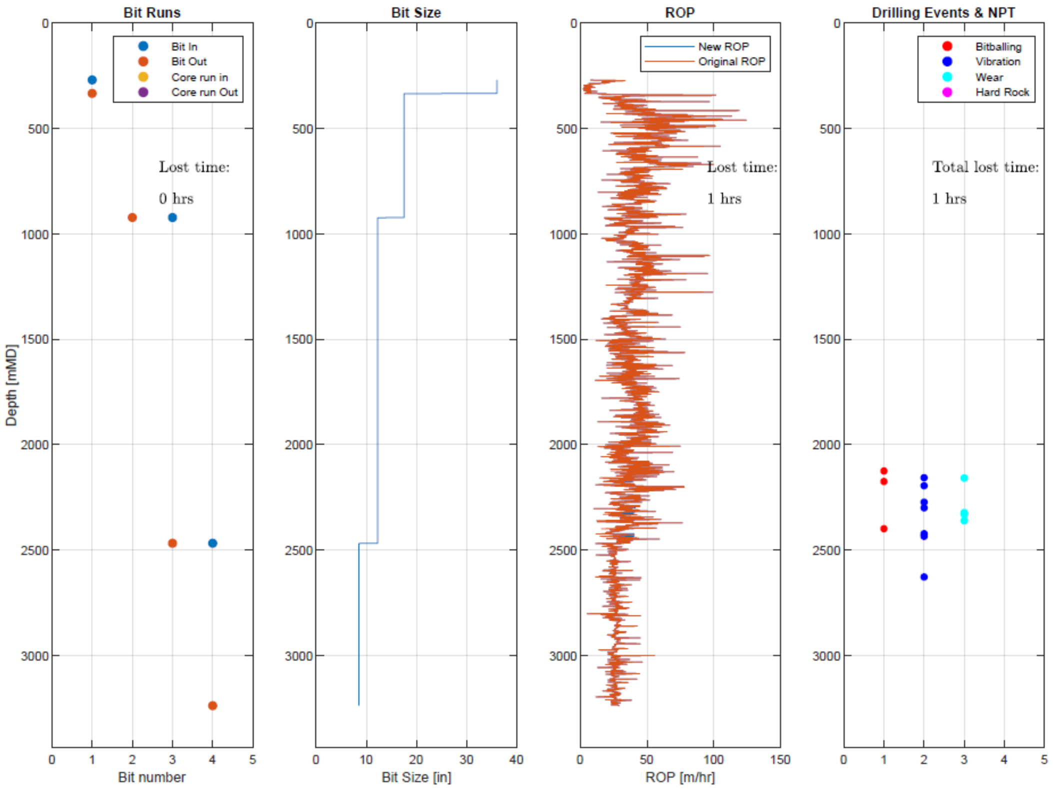


**Figure 149. Lost Time Torque-Method, Well 6407/8-5 S.**

## *Well 6407/8-6*

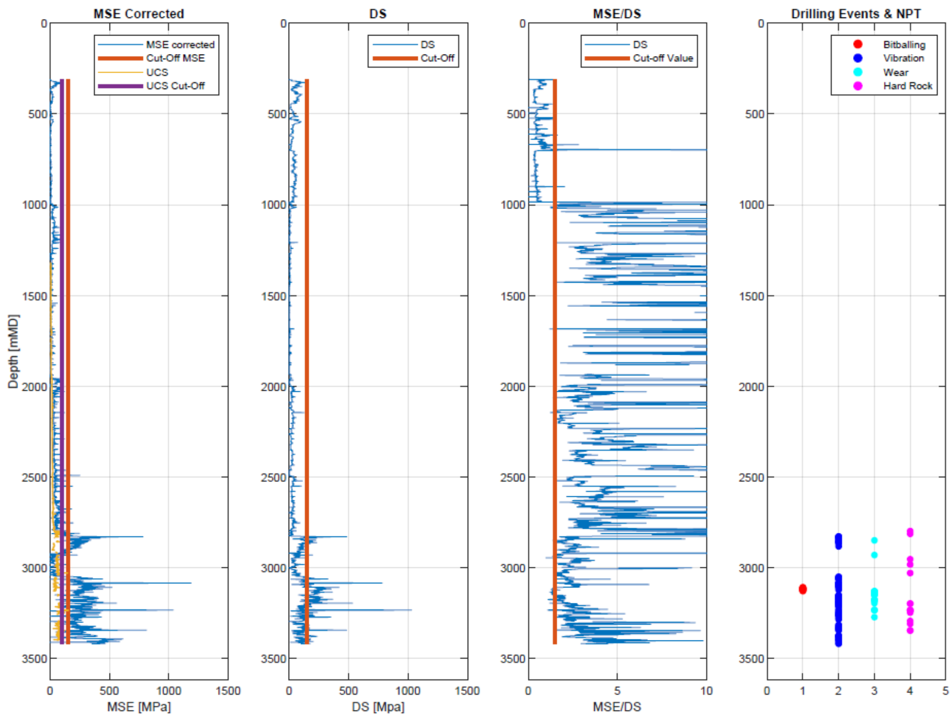


**Figure 150. Drilling Events Detection, Well 6407/8-6.**

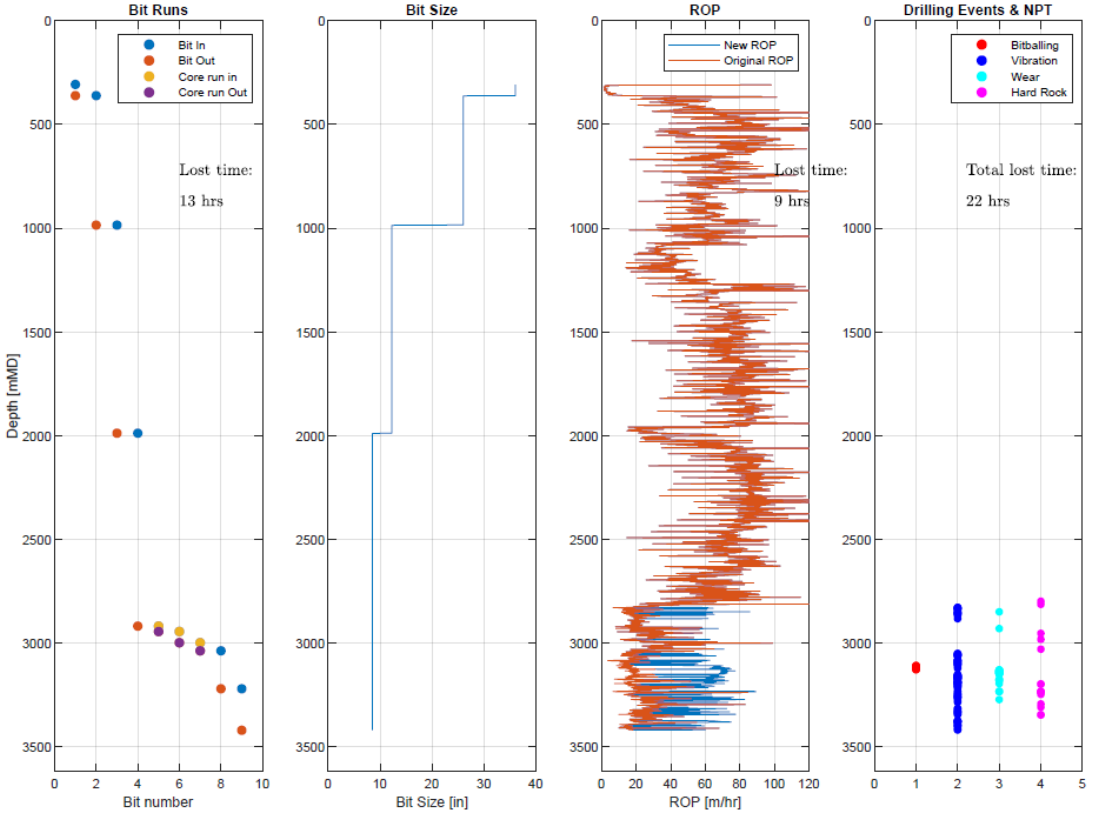


**Figure 151. Lost Time WOB-Method, Well 6407/8-6.**

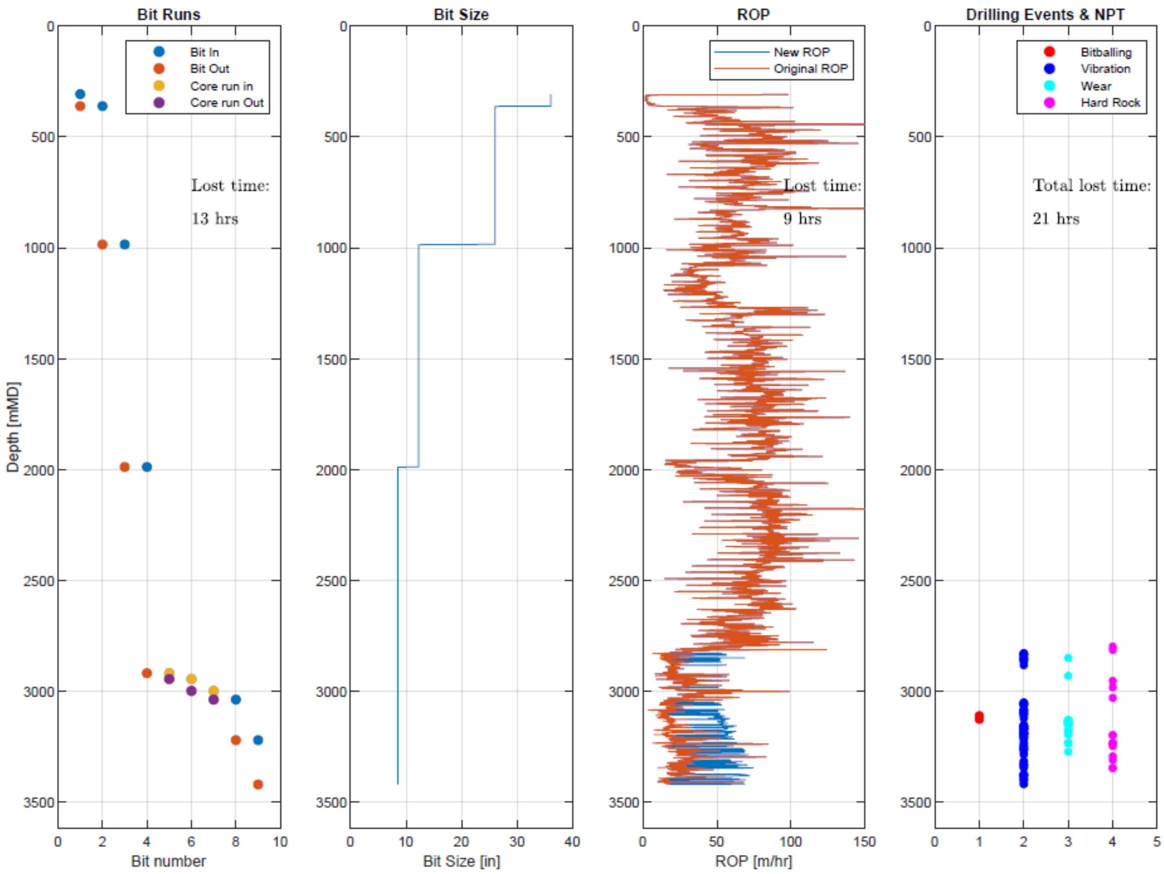


**Figure 152. Lost Time Torque-Method, Well 6407/8-6.**

*Well 6407/8-7*

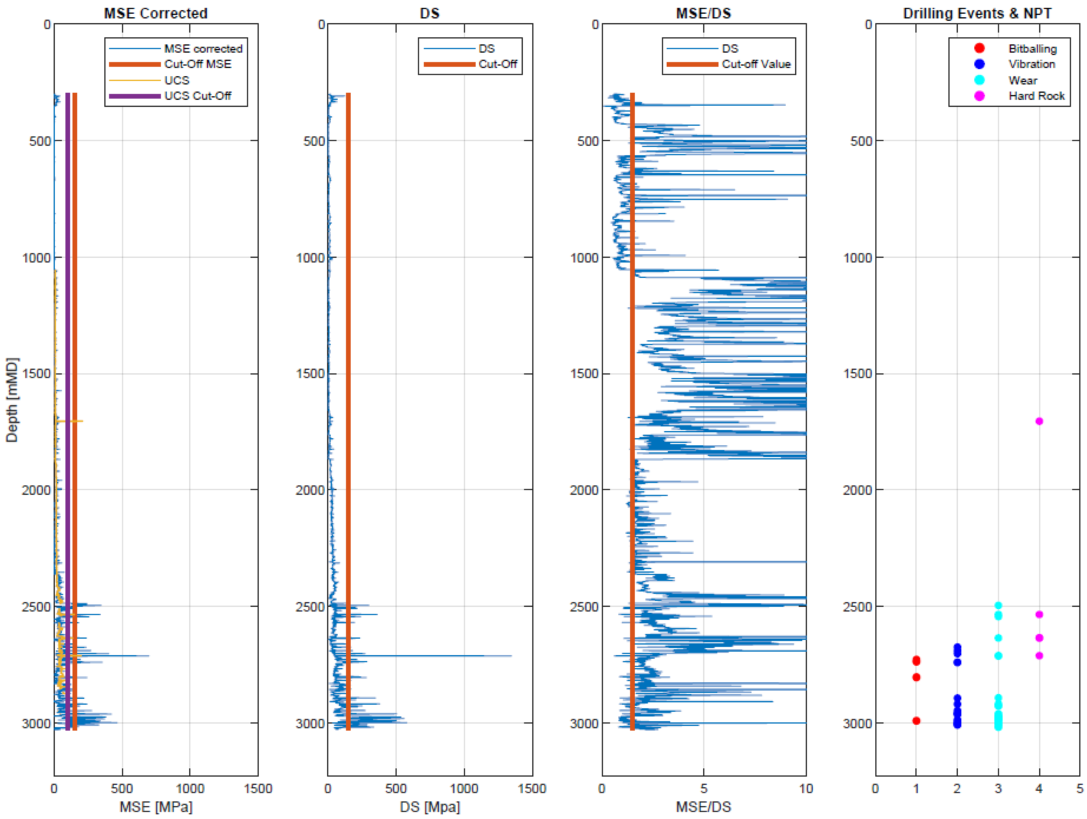


**Figure 153. Drilling Events Detection, Well 6407/8-7.**

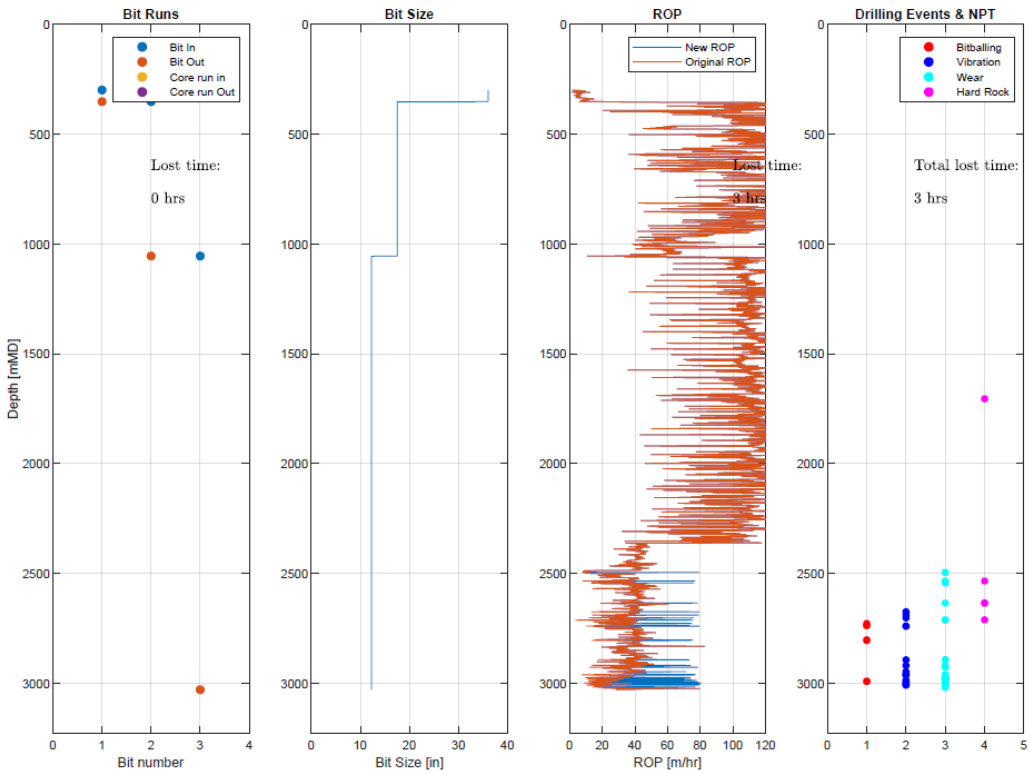


**Figure 154. Lost Time WOB-Method, Well 6407/8-7.**

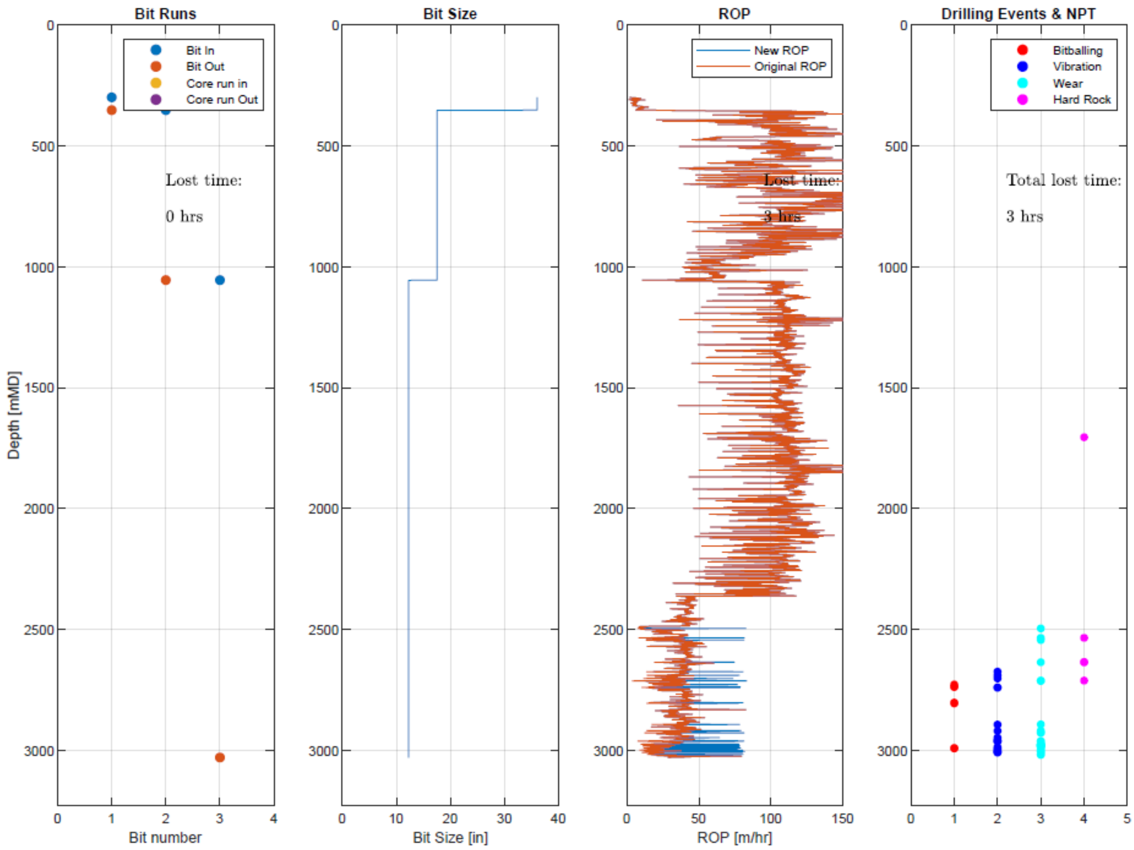


**Figure 155. Lost Time Torque-Method, Well 6407/8-7.**

*Well 6407/9-9*

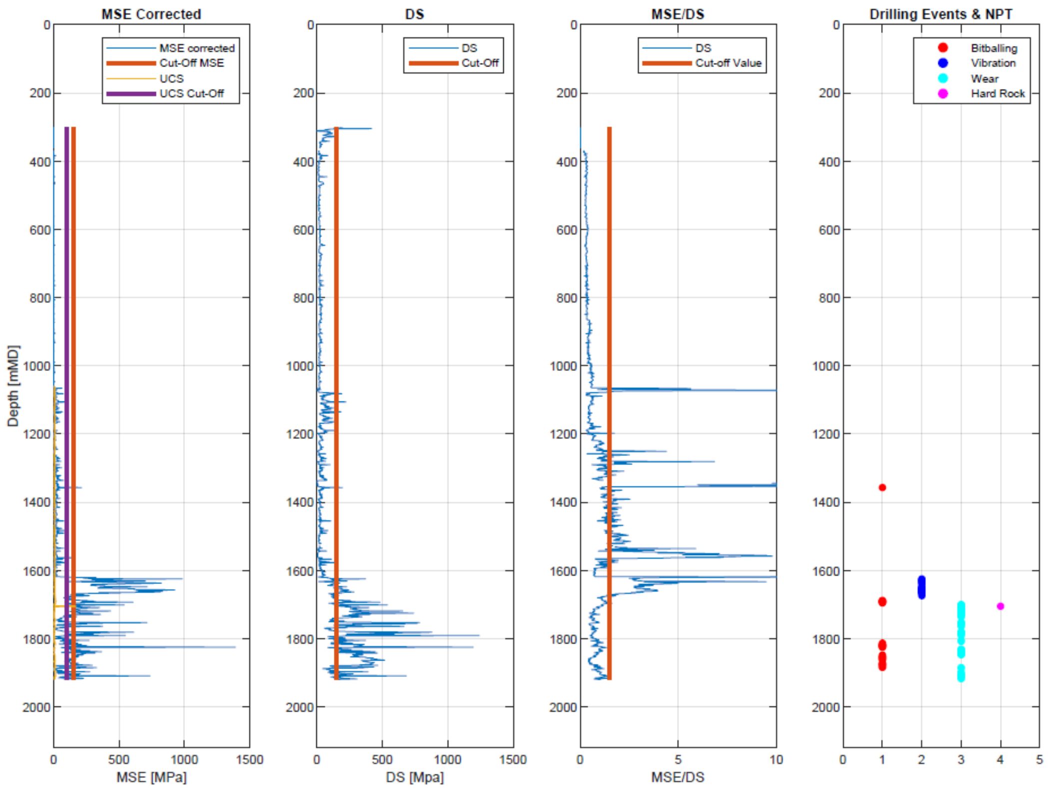


**Figure 156. Drilling Events Detection, Well 6407/9-9.**

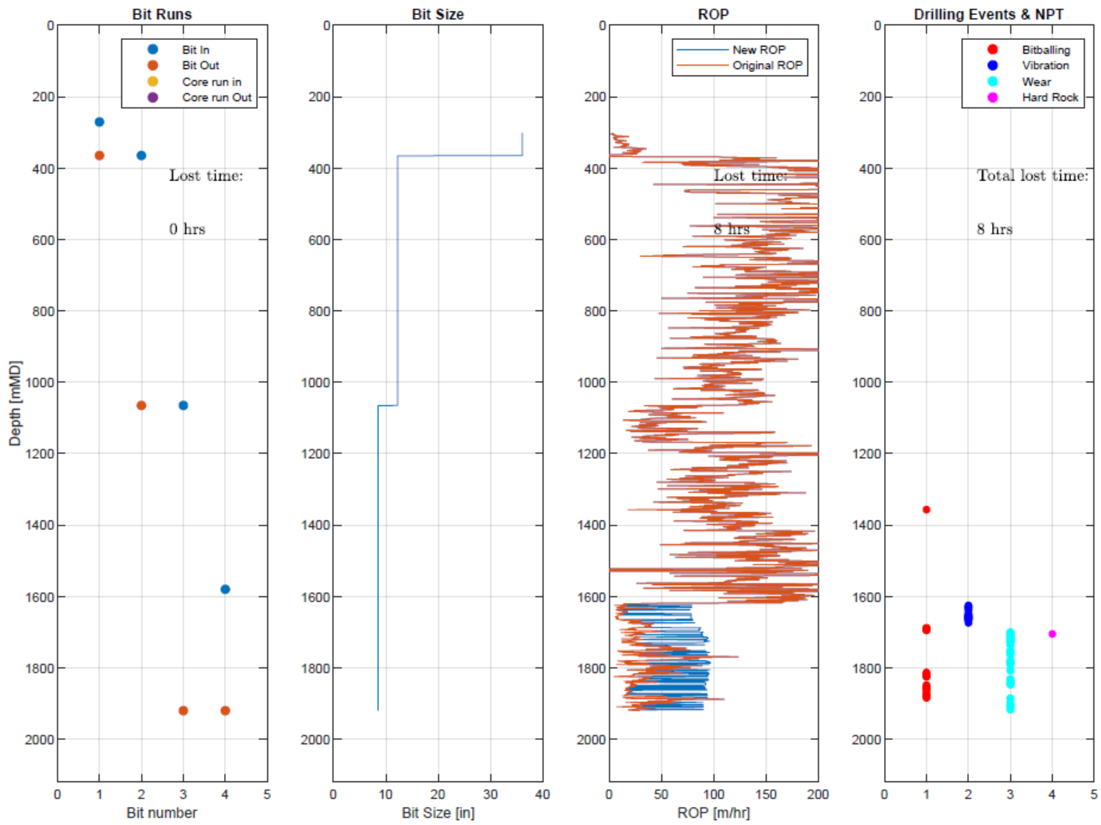


**Figure 157. Lost Time WOB-Method, Well 6407/9-9.**

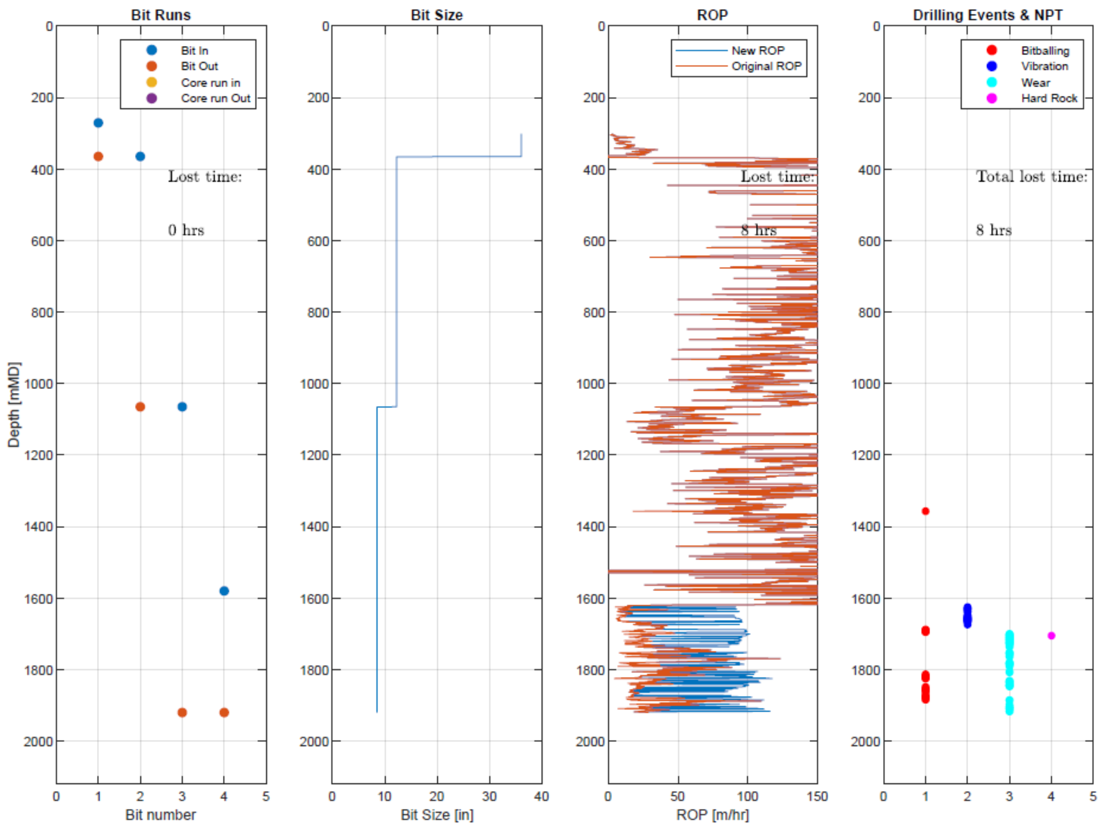


**Figure 158. Lost Time Torque-Method, Well 6407/9-9.**

*Well 6506/11-6*

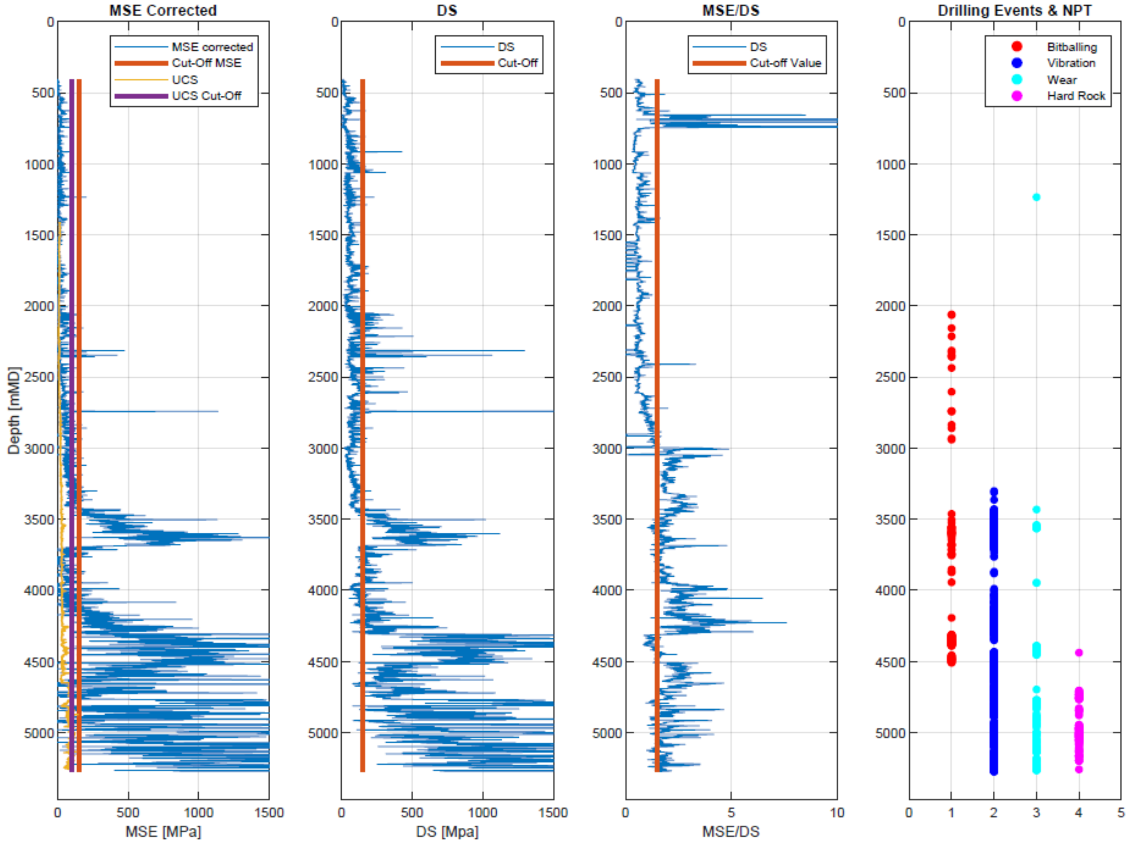


**Figure 159. Drilling Events Detection, Well 6506/11-6.**

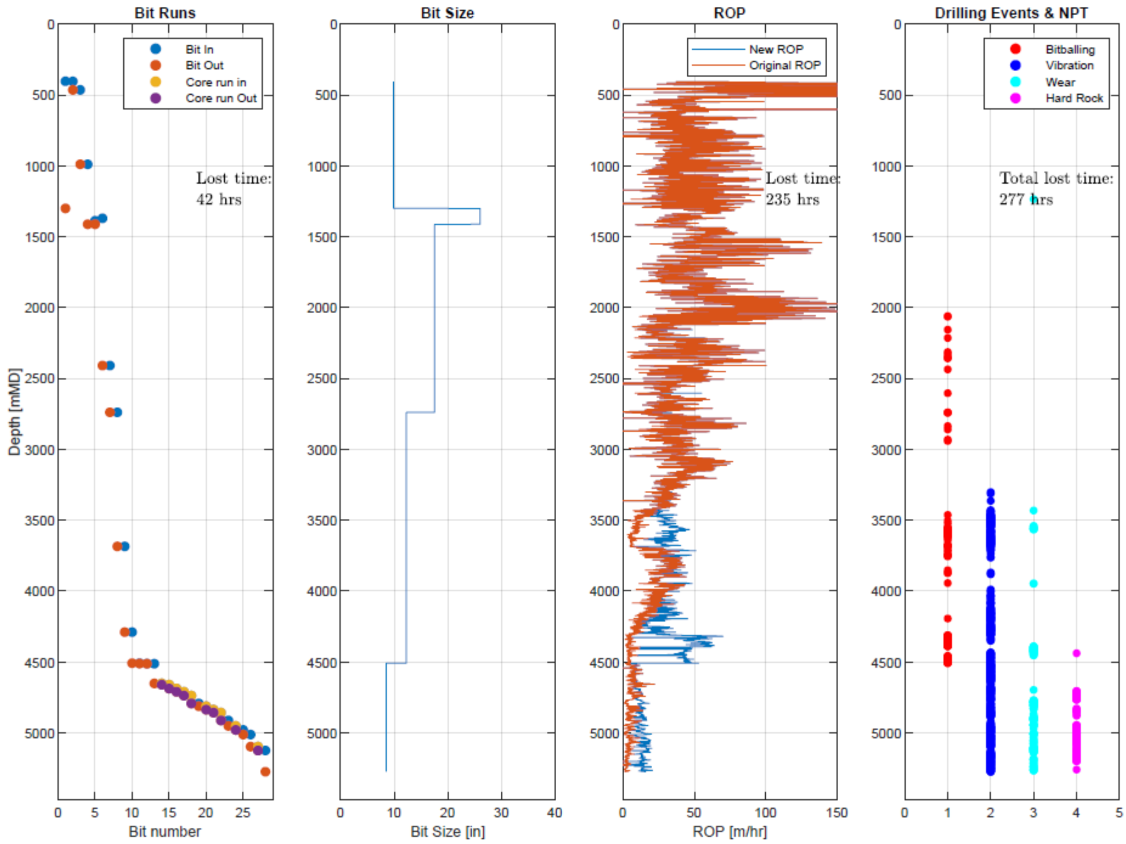


**Figure 160. Lost Time WOB-Method, Well 6506/11-6.**

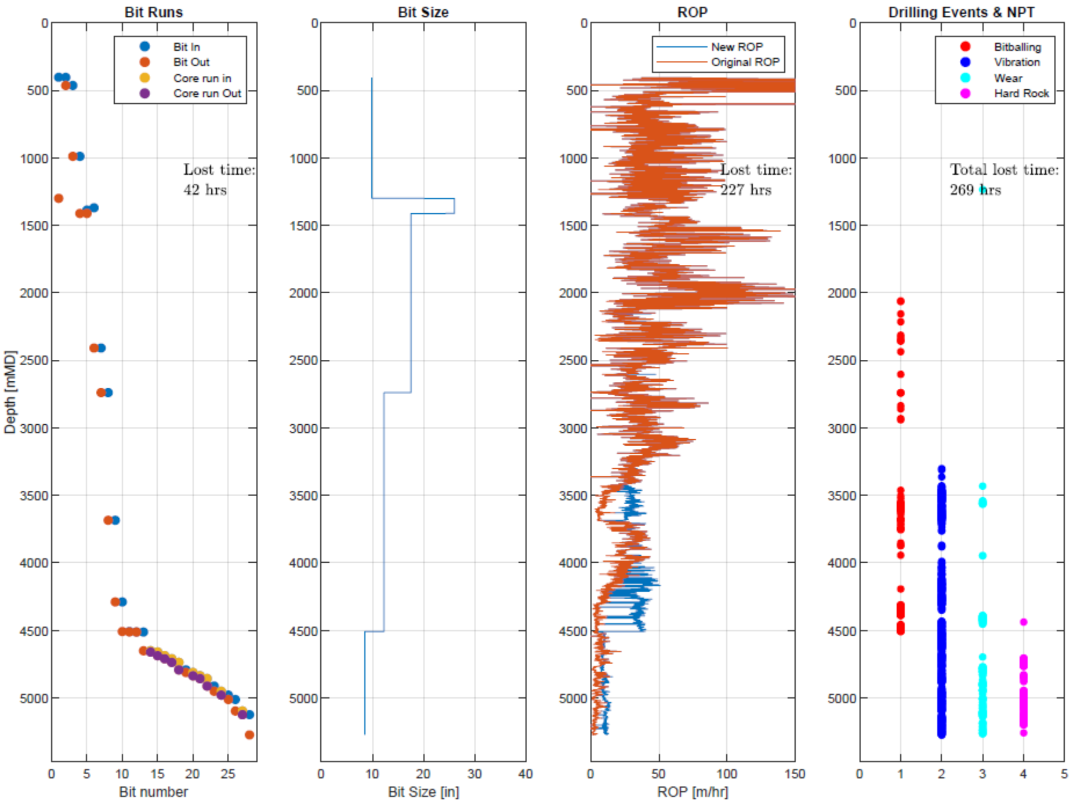


**Figure 161. Lost Time Torque-Method, Well 6506/11-6.**

# 9 MATLAB

## *9.1 MATLAB Prototype Workflow Case Study*

# Detection of NPT and Drilling Events on the NCS from Depth-based Drilling Mechanics Data

# Matlab Prototype Workflow for Case Study

```
clear all
clc
```

Import Well Data

```
% 34/4-15 S
wellname = '34_4-15 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\34_4-15 S\34_4-15 S
Ready');
drilling_data = xlsread('Ready_34_4-15 S  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_34_4-15 S COMPOSITE_LOG.xlsx');

% 34/4-15 A
%wellname = '34_4-15 A';
%path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\34_4-15 A\34_4-15 A
Ready');
%drilling_data = xlsread('Ready_34_4-15 A  DRILLING DATA - MUDLOG.xlsx');
%composite_data =xlsread('34_4-15 A COMPOSITE_LOG.xlsx');

% 35/3-6
%wellname = '35_3-6';
%path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_3-6\35_3-6 Ready');
%drilling_data = xlsread('Ready_35_3-6  DRILLING DATA - MUDLOG.xlsx');
%composite_data =xlsread('Ready_35_3-6 COMPOSITE_LOG.xlsx');
```

Name Drilling Data Variables

```
tvdrkb = drilling_data(:,1);
mdrkb = drilling_data(:,2);
bdia = drilling_data(:,3);
rop = drilling_data(:,4);
hkla = drilling_data(:,5);
hklx = drilling_data(:,6);
wob = drilling_data(:,7);
tqa = drilling_data(:,8);
tqx = drilling_data(:,9);
rpmb = drilling_data(:,10);
sppa = drilling_data(:,11);
tva = drilling_data(:,12);
mfoa = drilling_data(:,13);
mfia = drilling_data(:,14);
mwout = drilling_data(:,15);
mwin = drilling_data(:,16);
mtoa = drilling_data(:,17);
mtia = drilling_data(:,18);
ecd = drilling_data(:,19);
ppore = drilling_data(:,20);
lith = drilling_data(:,21);
dexp = drilling_data(:,22);
TD = max(mdrkb);
```

Name Composite Log Data

```
gammaray_raw = composite_data(:, 5);
gammaray_depth_raw = composite_data(:, 4);
sonic_raw = composite_data(:, 7);
sonic_depth_raw = composite_data(:, 6);
```

*Create equal length arrays*

```
gammaray_raw(isnan(gammaray_raw)) = []; % Remove NaN values
gammaray_depth_raw(isnan(gammaray_depth_raw)) = []; % Remove NaN values
sonic_raw(isnan(sonic_raw)) = []; % Remove NaN values
sonic_depth_raw(isnan(sonic_depth_raw)) = []; % Remove NaN values

gammaray = interp1(gammaray_depth_raw, gammaray_raw, mdrkb);
```

```
sonic = interp1(sonic_depth_raw, sonic_raw, mdrkb); % Microseconds/feet
```

Plot the Variables

```
figure(1) % Plot drilling parameters: ROP, WOB, RPM, Torque, Mud flow in, SPPA and ECD, Bit-size

subplot(1,8,1)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
grid on
box on

subplot(1,8,2)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
grid on
box on

subplot(1,8,3)
plot(rpmb, mdrkb)
set(gca, 'YDir','reverse')
title('RPM')
xlabel('RPM')
grid on
box on

subplot(1,8,4)
plot(tqa, mdrkb)
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
grid on
box on

subplot(1,8,5)
plot(mfia, mdrkb)
set(gca, 'YDir','reverse')
title('Mud Flow In')
xlabel('Flow [liter/min]')
grid on
box on

subplot(1,8,6)
plot(sppa, mdrkb)
set(gca, 'YDir','reverse')
title('SPPA')
xlabel('SPPA [bar]')
grid on
box on

subplot(1,8,7)
plot(ecd, mdrkb)
set(gca, 'YDir','reverse')
title('ECD')
xlabel('ECD [s.g.]')
grid on
box on

subplot(1,8,8)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [inch]')
xlim([0 40])
grid on
box on

nameofplot = 'Drilling_Data_Casestudy_';
fig1 = figure(1);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% Check if required drilling data is available
if sum(tqa)>0 && sum(rpmb)>0 && sum(rop)>0 && sum(wob)>0 && sum(bdia)>0
    disp('All required Drilling Data is available')
else
    disp('Data missing')
```

```
end
figure(2) % Plot Gamma-ray, Sonic log and Lithology
subplot(1,3,1)
plot(gammaray, mdrkb)
set(gca, 'YDir','reverse')
title('Gamma-Ray')
xlabel('Gamma-ray [gAPI]')
ylabel('Depth [mMD]')
ylim([0 TD+200])
grid on
box on

subplot(1,3,2)
plot(sonic, mdrkb)
set(gca, 'YDir','reverse')
title('Sonic Log')
xlabel('Interval velocity [µs/ft]')
ylim([0 TD+200])
grid on
box on

L = zeros(length(mdrkb), 1);
k = 0;
for i = 1:length(mdrkb);
    k=k+1;
    if lith(k) >= 600 && lith(k) < 700 % Claystone
        claystone(k) = L(i)+1;
    else
        claystone(k) = L(i) - 1;
    end
end

k = 0;
for i = 1:length(mdrkb);
    k=k+1;
    if lith(k) >= 300 && lith(k) < 400 % Sandstone
        sandstone(k) = L(i)+2;
    else
        sandstone(k) = L(i) - 1;
    end
end

k = 0;
for i = 1:length(mdrkb);
    k=k+1;
    if lith(k) >= 700 && lith(k) < 800 % Carbonates
        carbonates(k) = L(i)+3;
    else
        carbonates(k) = L(i) - 1;
    end
end

subplot(1,3,3)
scatter(claystone, mdrkb, 20, 'filled', 'g')
hold on
scatter(sandstone, mdrkb, 20, 'filled', 'y')
scatter(carbonates, mdrkb, 20, 'filled', 'b')
set(gca, 'YDir','reverse')
title('Lithology')
xlabel('Lithology')
hold off
xlim([0, 4])
legend('Claystone', 'Sandstone', 'Carbonates')
ylim([0 TD+200])
grid on
box on

nameofplot = 'Gamma_sonic_lithology_';
fig1 = figure(2);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Calculate Mechanical Specific Energy

```
wob_lbs = wob*2204.62; % Convert from tonnes to lbs
tqa_ft_lbs = tqa*737.56; % Convert from knm to ft-lbs
rop_ft_hr = rop*3.28; % Convert from m/hr to ft/hr
factor = 0.35; % Efficiency factor

mse_psi = factor * (wob_lbs./((pi/4).*(bdia.^2)) + (120*pi.*rpmb.*tqa_ft_lbs)./(((pi/4).*(bdia.^2)).*rop_ft_hr)); %
Mechanical specific energy, psi
```

```
mse = mse_psi./145; % Convert MSE from PSI to MPa
mse_psi_2 = factor * ( (120*pi.*rpmb.*tqa_ft_lbs)./(((pi/4).*(bdia.^2)).*rop_ft_hr)); % Mechanical specific energy, psi
mse_torque = mse_psi_2./145; % Convert MSE from PSI to MPa

mse_psi_3 = factor * (wob_lbs./((pi/4).*(bdia.^2))); % Mechanical specific energy, psi
mse_wob = mse_psi_3./145; % Convert MSE from PSI to MPa

figure(342423)
subplot(1,3,1)
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
ylim([0 TD+100])
xlabel('MPa')
ylabel('Depth [mMD]')
title('MSE')
subplot(1,3,2)
plot(mse_wob, mdrkb)
set(gca, 'YDir','reverse')
title ('WOB part')
xlabel('MPa')
ylim([0 TD+100])
subplot(1,3,3)
plot(mse_torque, mdrkb)
set(gca, 'YDir','reverse')
title('Torque part')
xlabel('MPa')
ylim([0 TD+100])

nameofplot = 'MSE_withWOB_withTorque_';
fig1 = figure(342423);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Plot MSE and Drilling Data

```
figure(3) % Plot drilling parameters: MSE, ROP, WOB, Torque and bit size
subplot(1,5,1)
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('MSE')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
xlim([0 1500])
grid on
box on

subplot(1,5,2)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
grid on
box on

subplot(1,5,3)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
grid on
box on

subplot(1,5,4)
plot(tqa, mdrkb)
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
grid on
box on

subplot(1,5,5)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [Inch]')
xlim([0 40])
grid on
box on

nameofplot = 'drillingparameters_';
```

```
fig1 = figure(3);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
figure(4) % Plot drilling parameters: MSE, ROP, WOB, RPM, Torque, Mud flow in, SPPA and ECD, Bit-size
subplot(1,9,1)
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('MSE')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
xlim([0 1500])
grid on
box on

subplot(1,9,2)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
grid on
box on

subplot(1,9,3)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
grid on
box on

subplot(1,9,4)
plot(rpmb, mdrkb)
set(gca, 'YDir','reverse')
title('RPM')
xlabel('RPM')
grid on
box on

subplot(1,9,5)
plot(tqa, mdrkb)
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
grid on
box on

subplot(1,9,6)
plot(mfia, mdrkb)
set(gca, 'YDir','reverse')
title('Mud Flow In')
xlabel('Flow [liter/min]')
grid on
box on

subplot(1,9,7)
plot(sppa, mdrkb)
set(gca, 'YDir','reverse')
title('SPPA')
xlabel('SPPA [bar]')
grid on
box on

subplot(1,9,8)
plot(ecd, mdrkb)
set(gca, 'YDir','reverse')
title('ECD')
xlabel('ECD [s.g.]')
grid on
box on

subplot(1,9,9)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [inch]')
xlim([0 40])
grid on
box on

nameofplot = 'drillingparameters2_';
fig1 = figure(4);
fig1.Renderer = 'Painters';
```

```
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Plot MSE and Create MSE Corrected

```
figure(5) % Plot Mechanical specific energy
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('Mechanical specific energy')
xlabel('Mechanical specific energy [MPa]')
ylabel('Depth [mMD]')
xlim([0 1500])
grid on
box on

nameofplot = 'Mechanical_specific_energy_';
fig1 = figure(5);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Create Baseline for MSE*

```
% 34/4-15 S
baselines = 3; % max 6
baseline_mse_min = [min(mdrkb), 1051, 2001];
baseline_mse_max = [1050, 2000, max(mdrkb)];
mse_min = [8, 14, 14];
mse_max = [55, 14, 140];

% 34/4-15 A
%baselines = 2; % max 6
%baseline_mse_min = [min(mdrkb), 2001];
%baseline_mse_max = [2000, max(mdrkb)];
%mse_min = [8, 8];
%mse_max = [8, 130];

% 35/3-6
%baselines = 3; % max 6
%baseline_mse_min = [min(mdrkb), 1130, 3081];
%baseline_mse_max = [1129, 3080, max(mdrkb)];
%mse_min = [2, 2, 200];
%mse_max = [2, 200, 170];

[baseline_mse] = funk_baseline_mse(baselines,baseline_mse_min, baseline_mse_max, mse_min, mse_max); % Baseline Function
```

*Plot Baseline and MSE*

```
figure(6)
subplot(1,2,1)
plot(mse, mdrkb)
hold on
plot(baseline_mse, mdrkb, 'linewidth', 1.5)
hold off
set(gca, 'YDir','reverse')
title('Mechanical specific energy')
xlabel('Mechanical specific energy [MPa]')
ylabel('Depth [mMD]')
legend('MSE', 'Baseline MSE')
xlim([0 1500])
grid on
box on

subplot(1,2,2)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit Size [in]')
xlim([0 40])
grid on
box on

nameofplot = 'Mechanical_specific_energy_baseline_';
fig1 = figure(6);
```

```
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Create and Plot MSE Corrected*

```
baseline_mse_transposed = transpose(baseline_mse); % Transpose the baseline vector
mse_corr = mse - baseline_mse_transposed; % Create MSE corrected
mse_corr(mse_corr < 0) = 0; % Remove negative MSE values

figure(7)
subplot(1,2,1)
plot(mse, mdrkb)
hold on
plot(baseline_mse, mdrkb, 'linewidth', 1.5)
hold off
set(gca, 'YDir','reverse')
title('Mechanical specific energy')
xlabel('Mechanical specific energy [MPa]')
ylabel('Depth [mMD]')
legend('MSE', 'Baseline MSE')
xlim([0 1500])
grid on
box on

subplot(1,2,2)
plot(mse_corr, mdrkb)
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('Mechanical specific energy [MPa]')
xlim([0 1500])
grid on
box on

nameofplot = 'MSE_Corrected_';
fig1 = figure(7);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

## Calculate Drilling Strength and MSE/DS ratio

*Calculate Depth of Cut and Drilling Strength*

```
doc = rop_ft_hr./(rpmb.*5); % Depth of cut, [inch]
ds_psi = wob_lbs./(bdia.*doc); % Drilling strength, [psi]
ds = ds_psi /145; % Convert from psi to MPa
mse_ds = mse./ds; % Ratio between Mechanical spesific energy and drilling strength
```

*Plot MSE, MSE Corrected, DS and MSE/DS*

```
figure(8)
subplot(1,4,1)
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('MSE')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
xlim([0 1500])
grid on
box on

subplot(1,4,2)
plot(mse_corr, mdrkb)
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('MSE [MPa]')
xlim([0 1500])
grid on
box on

subplot(1,4,3)
plot(ds, mdrkb)
set(gca, 'YDir','reverse')
title('DS')
xlabel('DS [MPa]')
```

```
xlim([0 1500])
grid on
box on

subplot(1,4,4)
plot(mse_ds, mdrkb)
set(gca, 'YDir','reverse')
title('MSE/DS')
xlabel('MSE/DS')
xlim([0, 10])
grid on
box on

nameofplot = 'MSE_MSE_Corr_DS_MSEDS_';
fig1 = figure(8);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

### Detect Drilling Events and NPT

*Create Cut-offs*

```
cut_off = 150; % Cut-off value for bitballing, bit wear and UCS [MPa]

ineff_mse_ds = 1.5; % Cut off values for inefficient drilling mse/ds ratio

a = ones(length(mdrkb), 1); % Array of ones the length of the well
mse_cutoff = cut_off .* a; % Array of bitballing cutoffs [MPa]
ineff_mse_ds_cutoff = ineff_mse_ds.*a; % Array of mse/ds cutoffs
```

*Plot MSE corrected with cut-off values, DS and MSE/DS*

```
figure(9)
subplot(1,3,1)
plot(mse_corr, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
legend('MSE corrected', 'Cut-Off')
xlim([0 1500])
grid on
box on

subplot(1,3,2)
plot(ds, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('DS')
xlabel('DS [MPa]')
xlim([0 1500])
legend('DS', 'Cut-Off')
grid on
box on

subplot(1,3,3)
plot(mse_ds, mdrkb)
hold on
plot(ineff_mse_ds_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE/DS')
xlabel('MSE/DS')
legend('DS', 'Cut-Off')
xlim([0, 10])
grid on
box on

nameofplot = 'Cutoff_values_';
fig1 = figure(9);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
```

```
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Detecting Bit Balling, Vibration, Wear or UCS*

```
b = zeros(length(mdrkb), 1); % Array of zeros
b0 = b - 1; % Not an event
b1 = b+1; % Bit balling
b2 = b + 2; % Vibration
b3 = b + 3; % Wear
b4 = b + 4; % UCS

k = 0;
    for i =1:length(mdrkb);
        k = k+1;
        ucs_event(k) = b0(i);
        bitballing (k) = b0 (i);
        wear(k) = b0(i);
        vibration (k) = b0(i);
    end

k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
            if mse_corr(k) < mse_cutoff(k) && ds(k) < mse_cutoff(k) && mse_ds(k) > ineff_mse_ds_cutoff(k)
                ucs_event (k) = b4(i);
            end
    end

k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
            if mse_corr(k) > mse_cutoff(k) && ds(k) > mse_cutoff(k) && mse_ds(k) < ineff_mse_ds_cutoff(k)
                bitballing (k) = b1 (i);
                wear(k) = b3(i);
                ucs_event (k) = b4(i);
            else
               bitballing (k) = b0 (i);
               wear(k) = b0(i);
               ucs_event(k) = b0(i);
            end
    end

k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
            if mse_corr(k) > mse_cutoff(k) && ds(k) > mse_cutoff(k) && mse_ds(k) > ineff_mse_ds_cutoff(k)
                vibration (k) = b2 (i);
            end
    end
```

*Plot MSE Corrected, DS, MSE/DS and Drilling Events*

```
figure(10)
subplot(1,4,1)
plot(mse_corr, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
legend('MSE corrected', 'Cut-Off')
xlim([0 1500])
ylim([0, TD+200])
grid on
box on

subplot(1,4,2)
plot(ds, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('DS')
xlabel('DS [Mpa]')
xlim([0 1500])
legend('DS', 'Cut-Off')
ylim([0, TD+200])
grid on
box on

subplot(1,4,3)
plot(mse_ds, mdrkb)
hold on
plot(ineff_mse_ds_cutoff, mdrkb, 'linewidt', 2)
```

```
hold off
set(gca, 'YDir','reverse')
title('MSE/DS')
xlabel('MSE/DS')
legend('DS', 'Cut-off Value')
xlim([0, 10])
ylim([0, TD+200])
grid on
box on

subplot(1,4,4)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
grid on
box on
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 5])

nameofplot = 'Drilling events & NPT detection_';
fig1 = figure(10);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Plot Drilling Events and NPT Detection and Drilling Parameters*

```
figure(11) % Plot Drilling events & NPT, ROP, WOB, RPM, Torque, Mud flow in, SPPA and ECD, Bit-size
subplot(1,9,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,9,2)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylim([0, TD+200])
grid on
box on


subplot(1,9,3)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,4)
plot(rpmb, mdrkb)
set(gca, 'YDir','reverse')
title('RPM')
xlabel('RPM')
ylim([0, TD+200])
grid on
box on

subplot(1,9,5)
plot(tqa, mdrkb)
```

```
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,6)
plot(mfia, mdrkb)
set(gca, 'YDir','reverse')
title('Mud Flow In')
xlabel('Flow [liter/min]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,7)
plot(sppa, mdrkb)
set(gca, 'YDir','reverse')
title('SPPA')
xlabel('SPPA [bar]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,8)
plot(ecd, mdrkb)
set(gca, 'YDir','reverse')
title('ECD')
xlabel('ECD [s.g.]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,9)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [inch]')
xlim([0 40])
ylim([0, TD+200])
grid on
box on

nameofplot = 'Detection NPT & events and drillingparameters_';
fig1 = figure(11);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

## Investigate NPT and drilling events for each hole section

*Create new variables for each hole section*

```
k = 0;

for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 6
        wob_6(k) = wob(i);
        rop_6(k) = rop(i) ;
        mdrkb_6 (k) = mdrkb(i);
        tqa_6 (k) = tqa(i);
        rpmb_6(k) = rpmb(i);
        bitsize_6(k) = 6;
    else
        wob_6(k) = NaN;
        rop_6(k) = NaN ;
        mdrkb_6 (k) = NaN;
        tqa_6 (k) = NaN;
        rpmb_6(k) = NaN;
        bitsize_6(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 8.5
        wob_8_5(k) = wob(i);
        rop_8_5(k) = rop(i) ;
        mdrkb_8_5 (k) = mdrkb(i);
        tqa_8_5 (k) = tqa(i);
```

```
        rpmb_8_5(k) = rpmb(i);
        bitsize_8_5(k) = 8.5;
    else
        wob_8_5(k) = NaN;
        rop_8_5(k) = NaN ;
        mdrkb_8_5 (k) = NaN;
        tqa_8_5 (k) = NaN;
        rpmb_8_5(k) = NaN;
        bitsize_8_5(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 9.88
        wob_9_88(k) = wob(i);
        rop_9_88(k) = rop(i) ;
        mdrkb_9_88 (k) = mdrkb(i);
        tqa_9_88 (k) = tqa(i);
        rpmb_9_88(k) = rpmb(i);
        bitsize_9_88(k) = 9.88;
    else
        wob_9_88(k) = NaN;
        rop_9_88(k) = NaN ;
        mdrkb_9_88 (k) = NaN;
        tqa_9_88 (k) = NaN;
        rpmb_9_88(k) = NaN;
        bitsize_9_88(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 12.25
        wob_12_25(k) = wob(i);
        rop_12_25(k) = rop(i) ;
        mdrkb_12_25 (k) = mdrkb(i);
        tqa_12_25 (k) = tqa(i);
        rpmb_12_25(k) = rpmb(i);
        bitsize_12_25(k) = 12.25;
    else
        wob_12_25(k) = NaN;
        rop_12_25(k) = NaN ;
        mdrkb_12_25 (k) = NaN;
        tqa_12_25 (k) = NaN;
        rpmb_12_25(k) = NaN;
        bitsize_12_25(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 17.5
        wob_17_5(k) = wob(i);
        rop_17_5(k) = rop(i) ;
        mdrkb_17_5 (k) = mdrkb(i);
        tqa_17_5 (k) = tqa(i);
        rpmb_17_5(k) = rpmb(i);
        bitsize_17_5(k) = 17.5;
    else
        wob_17_5(k) = NaN;
        rop_17_5(k) = NaN ;
        mdrkb_17_5 (k) = NaN;
        tqa_17_5 (k) = NaN;
        rpmb_17_5(k) = NaN;
        bitsize_17_5(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 26
        wob_26(k) = wob(i);
        rop_26(k) = rop(i) ;
        mdrkb_26 (k) = mdrkb(i);
        tqa_26 (k) = tqa(i);
        rpmb_26(k) = rpmb(i);
        bitsize_26(k) = 26;
    else
        wob_26(k) = NaN;
        rop_26(k) = NaN ;
        mdrkb_26 (k) = NaN;
        tqa_26 (k) = NaN;
        rpmb_26(k) = NaN;
```

```
        bitsize_26(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 36
        wob_36(k) = wob(i);
        rop_36(k) = rop(i) ;
        mdrkb_36 (k) = mdrkb(i);
        tqa_36 (k) = tqa(i);
        rpmb_36(k) = rpmb(i);
        bitsize_36(k) = 36;
    else
        wob_36(k) = NaN;
        rop_36(k) = NaN ;
        mdrkb_36 (k) = NaN;
        tqa_36 (k) = NaN;
        rpmb_36(k) = NaN;
        bitsize_36(k) = 0;
    end
end
```

*Plot Drilling events & NPT with WOB-ROP Crossplot for Each Bit Size*

```
% Find out which bit-sizes are in use in the drilling data excel file, and comment out the bit sizes that
% are not used for this well.
bits=0;
disp('The bit-sizes used for this well are:')
if sum(bitsize_9_88)>0
    disp('9 7/8"')
    bits=bits+1;
end
if sum(bitsize_36)>0
    disp('36"')
    bits=bits+1;
end
if sum(bitsize_26)>0
    disp('26"')
    bits=bits+1;
end
if sum(bitsize_17_5)>0
    disp('17 1/2"')
    bits=bits+1;
end
if sum(bitsize_12_25)>0
    disp('12 1/4"')
    bits=bits+1;
end
if sum(bitsize_8_5)>0
    disp('8 1/2"')
    bits=bits+1;
end
if sum(bitsize_6)>0
    disp('6"')
    bits=bits+1;
end

disp(['The drilling data contains ' num2str(bits) ' different bit-sizes'])
figure(12)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 5])
grid on
box on
%{
subplot(1,(bits+1),2)
scatter(wob_9_88, rop_9_88, 20, mdrkb, 'filled')
title('9 7/8" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
set(depth, 'YDir', 'reverse' );
axis([0 20 0 120])
```

```
grid on
box on
colorbar off
%}

%{
subplot(1,(bits+1),2)
scatter(wob_36, rop_36, 20, mdrkb, 'filled')
title('36" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
set(depth, 'YDir', 'reverse' );
axis([0 20 0 120])
grid on
box on
colorbar off
%}


subplot(1,(bits+1),2)
scatter(wob_26, rop_26, 20, mdrkb, 'filled')
title('26" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
set(depth, 'YDir', 'reverse' );
axis([0 20 0 120])
grid on
box on
colorbar off


subplot(1,(bits+1),3)
scatter(wob_17_5, rop_17_5, 20, mdrkb, 'filled')
title('17.5" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
grid on
box on
colorbar off

subplot(1,(bits+1),4)
scatter(wob_12_25, rop_12_25, 20, mdrkb, 'filled')
title('12.25" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
grid on
box on
colorbar off

subplot(1,(bits+1),5)
scatter(wob_8_5, rop_8_5, 20, mdrkb, 'filled')
title('8.5" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth [mMD]');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
grid on
box on
colorbar off

%{
subplot(1,(bits+1),6)
scatter(wob_6, rop_6, 20, mdrkb, 'filled')
title('6" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth [mMD]');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
grid on
box on
```

```
%}

nameofplot = 'wob_rop_depth_bitsize_events_NPT';
fig1 = figure(12);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Look at each hole section separately

6" Hole Section

```
%{
% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1
       efficient_rop_6(k) = rop_6(i);
       efficient_wob_6(k) = wob_6(i);
    end
end

efficient_rop_6(isnan(efficient_rop_6)) = []; % Remove NaN values
efficient_wob_6(isnan(efficient_wob_6)) = []; % Remove NaN values
c = polyfit(efficient_wob_6, efficient_rop_6, 1); % Create trendline
disp(['y = ' num2str(c(1)) 'x + ' num2str(c(2))]) % % Display equation y = ax + b

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline

figure(133)
subplot(1,2,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 6" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_6), max(mdrkb_6)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,2,2)
scatter(wob_6, rop_6, 20, mdrkb_6, 'filled')
title('6" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth [mMD]');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

nameofplot = 'wob_rop_depth_bitsize_6_events_NPT';
fig1 = figure(133);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
%}
```

*8.5" Hole Section*

```
% Find average value of efficient bit and plot
```

```
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1
       efficient_rop_8_5(k) = rop_8_5(i);
       efficient_wob_8_5(k) = wob_8_5(i);
    end
end

efficient_rop_8_5(isnan(efficient_rop_8_5)) = []; % Remove NaN values
efficient_wob_8_5(isnan(efficient_wob_8_5)) = []; % Remove NaN values
c = polyfit(efficient_wob_8_5, efficient_rop_8_5, 1); % Create trendline
disp(['y = ' num2str(c(1)) 'x + ' num2str(c(2))]) % % Display equation y = ax + b

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline

figure(13)
subplot(1,2,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 8.5" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_8_5), max(mdrkb_8_5)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,2,2)
scatter(wob_8_5, rop_8_5, 20, mdrkb_8_5, 'filled')
title('8.5" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth [mMD]');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

nameofplot = 'wob_rop_depth_bitsize_8_5_events_NPT';
fig1 = figure(13);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*12.25" Hole Section*

```
% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1
       efficient_rop_12_25(k) = rop_12_25(i);
       efficient_wob_12_25(k) = wob_12_25(i);
    end
end

efficient_rop_12_25(isnan(efficient_rop_12_25)) = []; % Remove NaN values
efficient_wob_12_25(isnan(efficient_wob_12_25)) = []; % Remove NaN values
c = polyfit(efficient_wob_12_25, efficient_rop_12_25, 1); % Create trendline
disp(['y = ' num2str(c(1)) '*WOB + ' num2str(c(2))]) % % Display equation in the form y = a*x + b

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline

figure(14)
subplot(1,2,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
```

```
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 12.25" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_12_25), max(mdrkb_12_25)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,2,2)
scatter(wob_12_25, rop_12_25, 20, mdrkb_12_25, 'filled')
title('12.25" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth [mMD]');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

nameofplot = 'wob_rop_depth_bitsize_12_25_events_NPT';
fig1 = figure(14);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*17.5" Hole Section*

```
% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1
       efficient_rop_17_5(k) = rop_17_5(i);
       efficient_wob_17_5(k) = wob_17_5(i);
    end
end

efficient_rop_17_5(isnan(efficient_rop_17_5)) = []; % Remove NaN values
efficient_wob_17_5(isnan(efficient_wob_17_5)) = []; % Remove NaN values
c = polyfit(efficient_wob_17_5, efficient_rop_17_5, 1); % Create trendline
disp(['y = ' num2str(c(1)) '*WOB + ' num2str(c(2))]) % % Display equation in the form y = a*x + b

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline


figure(15)
subplot(1,2,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 17.5" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_17_5), max(mdrkb_17_5)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,2,2)
scatter(wob_17_5, rop_17_5, 20, mdrkb_17_5, 'filled')
title('17.5" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
```

```
colorbar
depth = colorbar;
ylabel(depth, 'Depth [mMD]');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

nameofplot = 'wob_rop_depth_bitsize_17_5_events_NPT';
fig1 = figure(15);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*26" Hole Section*

```
%

% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1
       efficient_rop_26(k) = rop_26(i);
       efficient_wob_26(k) = wob_26(i);
    end
end

efficient_rop_26(isnan(efficient_rop_26)) = []; % Remove NaN values
efficient_wob_26(isnan(efficient_wob_26)) = []; % Remove NaN values
c = polyfit(efficient_wob_26, efficient_rop_26, 1); % Create trendline
disp(['y = ' num2str(c(1)) '*x + ' num2str(c(2))]) % % Display equation in the form y = a*x + b

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline

figure(16)
subplot(1,2,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 26" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_26) max(mdrkb_26)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,2,2)
scatter(wob_26, rop_26, 20, mdrkb_26, 'filled')
title('26" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth meter MD');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

nameofplot = 'wob_rop_depth_bitsize_26_events_NPT';
fig1 = figure(16);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
```

```
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);

%
```

36" Hole Section

```
%{

% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1
       efficient_rop_36(k) = rop_36(i);
       efficient_wob_36(k) = wob_36(i);
    end
end

efficient_rop_36(isnan(efficient_rop_36)) = []; % Remove NaN values
efficient_wob_36(isnan(efficient_wob_36)) = []; % Remove NaN values
c = polyfit(efficient_wob_36, efficient_rop_36, 1); % Create trendline
disp(['y = ' num2str(c(1)) '*x + ' num2str(c(2))]) % % Display equation in the form y = a*x + b

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline

figure(16)
subplot(1,2,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 36" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_36) max(mdrkb_36)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,2,2)
scatter(wob_36, rop_36, 20, mdrkb_36, 'filled')
title('36" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth meter MD');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

nameofplot = 'wob_rop_depth_bitsize_36_events_NPT';
fig1 = figure(99);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);

%}
```

9 7/8" Section

```
%{
% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1
```

```
        efficient_rop_9_88(k) = rop_9_88(i);
        efficient_wob_9_88(k) = wob_9_88(i);
    end
end

efficient_rop_9_88(isnan(efficient_rop_9_88)) = []; % Remove NaN values
efficient_wob_9_88(isnan(efficient_wob_9_88)) = []; % Remove NaN values
c = polyfit(efficient_wob_9_88, efficient_rop_9_88, 1); % Create trendline
disp(['y = ' num2str(c(1)) '*x + ' num2str(c(2))]) % % Display equation in the form y = a*x + b

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline

figure(16)
subplot(1,2,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 9 7/8" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_9_88) max(mdrkb_9_88)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,2,2)
scatter(wob_9_88, rop_9_88, 20, mdrkb_9_88, 'filled')
title('9 7/8" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth meter MD');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

nameofplot = 'wob_rop_depth_bitsize_9_88_events_NPT';
fig1 = figure(999);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);

%}
```

### Calculate and Plot Confined Compressive Strength

*Import Predicted Pore Pressure*

```
rkb = [27]; % Distance from RKB to seafloor
pressures = xlsread('34_4-15 S porepressure.xlsx'); % Import pore pressure file
pp_sg = pressures(:,3); % Pore Pressure in s.g.
pp_bar = pressures(:,4); % Pore Pressure in bar

figure(100)
subplot(1,2,1)
plot(pp_sg, mdrkb)
set(gca, 'YDir', 'reverse' );
xlabel('Pore Pressure [s.g.]')
ylabel('Depth [mMD]')
title('Pore Pressure s.g.')
xlim([0 2])
subplot(1,2,2)
plot(pp_bar, mdrkb)
set(gca, 'YDir', 'reverse' );
xlabel('Pore Pressure [bar]')
title('Pore Pressure bar')

nameofplot = 'pore_pressure_';
fig1 = figure(100);
fig1.Renderer = 'Painters';
filetype = '.pdf';
```

```
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Calculate Angle of Internal Friction*

```
Vp_km = (1./sonic)./(3.2808*10^(-3)); % Convert µs/ft to km/s
AIF = asind((Vp_km-1)./(Vp_km+1)); % Angle of Internal Friction [degrees]

figure(17)
subplot(1,3,1)
plot(Vp_km, mdrkb)
set(gca, 'YDir','reverse')
title('Vp')
xlabel('Interval velocity [km/s]')
ylabel('Depth [mMD]')
ylim([0, TD])
grid on
box on

subplot(1,3,2)
plot(sonic, mdrkb)
set(gca, 'YDir','reverse')
title('Sonic ')
xlabel('Interval velocity [µs/ft]')
ylim([0, TD])
grid on
box on

subplot(1,3,3)
plot(AIF, mdrkb)
set(gca, 'YDir','reverse')
title('Angle of Internal Friction ')
xlabel('Degrees')
ylim([0, TD])
grid on
box on

nameofplot = 'Angle_AIF_Sonic_Vp_';
fig1 = figure(17);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Calculate UCS*

```
Vp = (1./sonic)./(3.2808*10^(-3)); % Convert µs/ft to m/s

for i = 1:length(mdrkb)
    if claystone (i) == 1
        UCS(i) = 1.35*(304.8/sonic(i))^2.6; % Globally Shale
    elseif carbonates(i) == 3
        UCS (i) = ((7682/sonic(i))^1.82)/145; % MilitzerStoll Lime
    elseif sandstone(i) == 2
        if tvdrkb(i) < 2000
            UCS(i) = 1.4138*10^7*(sonic(i)^-3); % GulfCoast WeakSand
        elseif 2000 <= tvdrkb(i)
            UCS(i) = 1200*exp(-0.036*sonic(i)); % BowenBasinAustralia_Sand
        end
    else
        UCS(i) = NaN;
    end
end

figure(18)
plot(UCS, mdrkb)
title('UCS')
set(gca, 'YDir','reverse')
xlabel('UCS [MPa]')
ylabel('Depth [mMD]')
ylim([0, TD])
grid on
box on
```

*Calculate CCS*

```
ecd_bar = ecd.*0.0981.*(tvdrkb-rkb); % Convert ecd from s.g. to bar
```

```
Dp_bar = ecd_bar - pp_bar; % Differential pressure [bar]
Dp = Dp_bar./10;
ucs = transpose(UCS);
ccs = ucs + Dp + 2.*Dp.*(sind(AIF)./(1-sind(AIF)));

figure(19)
subplot(1,4,1)
plot(ucs, mdrkb)
title('UCS')
set(gca, 'YDir','reverse')
xlabel('UCS [MPa]')
ylabel('Depth [mMD]')
ylim([0, TD])
grid on
box on

subplot(1,4,2)
plot(ccs, mdrkb)
title('CCS')
set(gca, 'YDir','reverse')
xlabel('CCS [MPa]')
ylim([0, TD])
grid on
box on

subplot(1,4,3)
plot(pp_bar, mdrkb)
hold on
plot(ecd_bar, mdrkb)
plot(Dp_bar, mdrkb)
hold off
set(gca, 'YDir','reverse')
legend('Pore Pressure','ECD', 'Differential Pressure')
ylim([0 TD])
xlabel('Pressure [Bar]')
title('Wellbore Pressure')
grid on
box on

subplot(1,4,4)
scatter(claystone, mdrkb, 20, 'filled', 'g')
hold on
scatter(sandstone, mdrkb, 20, 'filled', 'y')
scatter(carbonates, mdrkb, 20, 'filled', 'b')
set(gca, 'YDir','reverse')
title('Lithology')
xlabel('Lithology')
hold off
xlim([0, 4])
ylim([0 TD])
legend('Claystone', 'Sandstone', 'carbonates')
grid on
box on

nameofplot = 'UCS_CCS_Pressures_lithology_';
fig1 = figure(19);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Compare CCS and UCS to Drilling events and NPT

```
figure(20)
plot (mse, mdrkb)
hold on
plot(ccs, mdrkb)
plot(ucs, mdrkb)
hold off
title('MSE, CCS & UCS')
xlabel('[MPa]')
ylabel('Depth [mMD]')
xlim([0, 1500])
ylim([0 TD])
set(gca, 'YDir','reverse')
legend('Mechanical Specific Energy', 'Confined Compressive Strength', 'Unconfined Compressive Strength')
grid on
box on

nameofplot = 'MSE_CCS_UCS_';
fig1 = figure(20);
fig1.Renderer = 'Painters';
```

```
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
figure(21)
subplot(1,2,1)
plot (mse_corr, mdrkb)
hold on
plot(ccs, mdrkb)
plot(ucs, mdrkb)
hold off
title('MSE, CCS & UCS')
xlabel('[MPa]')
ylabel('Depth [mMD]')
ylim([0 TD+200])
xlim([0 1500])
set(gca, 'YDir','reverse')
legend('MSE Corrected', 'CCS', 'UCS')
grid on
box on

subplot(1,2,2)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 5])
grid on
box on

nameofplot = 'MSE_CCS_UCS_Drillingevents_NPT_';
fig1 = figure(21);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Update Drilling Events and NPT based on CCS results*

```
% Update Drilling Events and NPT based on CCS results
ccs_cutoff_1 = 150; % Cut-off value for ccs [Mpa]
ccs_cutoff = ccs_cutoff_1 .*a; % Array of cutoff values [MPa]

b = zeros(length(mdrkb), 1); % Array of zeros
b0 = b - 1; % Not an event
b1 = b+1; % Bit balling
b2 = b + 2; % Vibration
b3 = b + 3; % Wear
b4 = b + 4; % UCS

ucs_event_2 = ucs_event;
bitballing_2 = bitballing;
vibration_2 = vibration;
wear_2 = wear;

k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if ucs_event(k) == 4 && ccs(k) > ccs_cutoff(k)
            ucs_event_2(k) = b4(i);
            bitballing_2(k) = b0(i);
            vibration_2(k) = b0(i);
            wear_2(k) = b0(i);
    else
        ucs_event_2(k) = b0(i);

    end
end

figure(22)
subplot(1,3,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
```

```
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 5])
grid on
box on

subplot(1,3,2)
plot (mse_corr, mdrkb)
hold on
plot(ccs, mdrkb)
plot(ucs, mdrkb)
plot(ccs_cutoff, mdrkb, 'linewidt', 2)
hold off
title('MSE, CCS & UCS')
xlabel('[MPa]')
ylim([0, TD+200])
xlim([0 500])
set(gca, 'YDir','reverse')
legend('MSE Corrected', 'CCS', 'UCS', 'Cut-off')
grid on
box on

subplot(1,3,3)
scatter(bitballing_2, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration_2, mdrkb, 20, 'filled', 'b')
scatter(wear_2, mdrkb, 20, 'filled', 'c')
scatter(ucs_event_2, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Updated Drilling Events & NPT')
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 5])
grid on
box on

nameofplot = 'UCS_CCS_MSE_updated_Drillingevents_NPT_';
fig1 = figure(22);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Plot UCS estimation according to Hardness*

```
% Sort UCS estimations according to Hardness Category

for i = 1:length(mdrkb)
    if ucs(i) <= 0.6 && ucs(i) > 0
        hardness(i) = 1;                % Soil < 0.6 MPa
    elseif ucs(i) <= 1.25 && ucs(i) > 0.6
        hardness(i) = 2;                % Very Soft Rock: 0.6 - 1.25 MPa
    elseif ucs(i) <= 5 && ucs(i) > 1.25
        hardness(i) = 3;                % Soft Rock = 1.25 – 5.0 MPa
    elseif ucs(i) <= 12.5 && ucs(i) > 5
        hardness(i) = 4;                % Moderately Soft Rock: 5 – 12.5 MPa
    elseif ucs(i) <= 50 && ucs(i) > 12.5
        hardness(i) = 5;                % Moderately Hard Rock: 12.5 - 50 MPa
    elseif ucs(i) <= 100 && ucs(i) > 50
        hardness(i) = 6;                % Hard Rock: 50-100 MPa
    elseif ucs(i) <= 250 && ucs(i) > 100
        hardness(i) = 7;                % Very Hard Rock: 100 - 250 MPa
    elseif ucs(i) < 1000 && ucs(i) > 250
        hardness(i) = 8;                % Extremely Hard Rock > 250 MPa
    else
        hardness(i) = NaN;
    end
end
figure(23)
subplot(1,2,1)
scatter(ucs, mdrkb, 20, hardness, 'filled')
title('Hardness Rock')
xlabel('UCS [MPa]')
```

```
ylabel('Depth [mMD]')
set(gca, 'YDir','reverse')
colorbar
hard = colorbar;
ylabel(hard, 'Hardness Category');
ylim([0 TD+100])
colorbar('Ticks',[1,2,3,4,5,6,7,8], 'TickLabels', {'','','','','','','',''}) %{'1','2','3','4','5','6','7','8'})
colorbar off
grid on
box on

subplot(1,2,2)
scatter(mse, mdrkb, 20, hardness, 'filled')
title('Hardness Rock')
xlabel('MSE [MPa]')
set(gca, 'YDir','reverse')
colorbar
hard = colorbar;
ylabel(hard, 'Hardness Category');
ylim([0 TD+100])
xlim([0 2000])
colorbar('Ticks',[1,2,3,4,5,6,7,8], 'TickLabels',{'Soil','Very Soft Rock','Soft Rock','Moderately Soft Rock','Moderately
Hard Rock', 'Hard Rock', 'Very Hard Rock', 'Extremely Hard Rock'})
grid on
box on

nameofplot = 'Hardness_UCS_MSE_';
fig1 = figure(23);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
figure(3323)
scatter(wob_8_5, rop_8_5, 20, hardness, 'filled')
title('8.5" Hole Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
hard = colorbar;
ylabel(hard, 'Hardness Category');
colorbar('Ticks',[1,2,3,4,5,6,7,8], 'TickLabels',{'Soil','Very Soft Rock','Soft Rock','Moderately Soft Rock','Moderately
Hard Rock', 'Hard Rock', 'Very Hard Rock', 'Extremely Hard Rock'})
grid on
box on

for i = 1:length(mdrkb)
    if claystone(i) == 1;
        lithology(i) = 2;
    elseif sandstone(i) == 2;
        lithology(i) = 3;
    elseif carbonates == 3;
        lithology(i) = 1;
    else
        lithology(i) = NaN;
    end
end


figure(43223)
scatter(wob_8_5, rop_8_5, 20, lithology, 'filled')
title('8.5" Hole Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
hard = colorbar;
ylabel(hard, 'Lithology');
colorbar('Ticks',[1,2,3], 'TickLabels',{'Carbonates', ' Claystone', 'Sandstone'})
grid on
box on
```

*Create Cut-off UCS and Update Drilling events and NPT based on lithology and UCS estimations*

```
% Create Cut-off for UC
ucs_cutoff_1 = 100; % Very Hard Rock [MPa]
ucs_cutoff = ucs_cutoff_1.*a; % Array of cut-off
lim_ucs = 1000.*a; % Limit UCS array
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if ucs(i) > ucs_cutoff(i) && ucs(i) < lim_ucs(i) % Update UCS Drilling Events
        ucs_event_3(k) = b4(i);
    else
```

```
        ucs_event_3(k) = b0(i);
    end
end
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if claystone(i) == 1 && bitballing_2(i) == 1 % Update bit balling based on lithology
       bitballing_3(k) = b1(i);
    else
       bitballing_3(k) = b0(i);
    end
end
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing_3(i) == -1 && wear_2(i) == 3 && ucs_event_3(i) == -1
       wear_3(k) = b3(i);
    else
       wear_3(k) = b0(i);
    end
end
figure(24)
subplot(1,3,1)
plot(ucs, mdrkb)
hold on
plot(ucs_cutoff, mdrkb)
hold off
title('UCS')
xlabel('UCS [MPa]')
ylabel('Depth [mMD]')
set(gca, 'YDir','reverse')
legend('UCS', 'Cut-Off')
ylim([0 TD+100])
grid on
box on

subplot(1,3,2)
scatter(claystone, mdrkb, 20, 'filled', 'g')
hold on
scatter(sandstone, mdrkb, 20, 'filled', 'y')
scatter(carbonates, mdrkb, 20, 'filled', 'b')
set(gca, 'YDir','reverse')
title('Lithology')
xlabel('Lithology')
hold off
xlim([0, 4])
legend('Claystone', 'Sandstone', 'Carbonates')
ylim([0, TD+200])
grid on
box on

subplot(1,3,3)
scatter(bitballing_3, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration_2, mdrkb, 20, 'filled', 'b')
scatter(wear_3, mdrkb, 20, 'filled', 'c')
scatter(ucs_event_3, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

nameofplot = 'Final_UCS_Lithology_Drilling events & NPT_';
fig1 = figure(24);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
figure(25)
subplot(1,4,1)
plot (mse_corr, mdrkb)
hold on
plot(ucs, mdrkb)
plot(mse_cutoff, mdrkb, 'linewidt', 2)
plot(ucs_cutoff, mdrkb, 'linewidt', 2)
hold off
title('MSE & UCS')
xlabel('[MPa]')
ylabel('Depth [mMD]')
```

```
ylim([0, TD+200])
xlim([0 1000])
set(gca, 'YDir','reverse')
legend('MSE Corrected', 'UCS', 'Cut-off MSE', 'Cut-off UCS')
grid on
box on

subplot(1,4,2)
plot(ds, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('DS')
xlabel('DS [Mpa]')
xlim([0 1500])
legend('DS', 'Cut-Off')
ylim([0, TD+200])
grid on
box on

subplot(1,4,3)
plot(mse_ds, mdrkb)
hold on
plot(ineff_mse_ds_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE/DS')
xlabel('MSE/DS')
legend('DS', 'Cut-off')
xlim([0, 10])
ylim([0, TD+200])
grid on
box on

subplot(1,4,4)
scatter(bitballing_3, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration_2, mdrkb, 20, 'filled', 'b')
scatter(wear_3, mdrkb, 20, 'filled', 'c')
scatter(ucs_event_3, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

nameofplot = 'Final_MSE_UCS_DS_MSEDS_Drilling events & NPT_';
fig1 = figure(25);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

## Calculate Lost Time

6" Hole Section

```
%{
% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing_3(k) == -1  && vibration_2(k) == -1  && wear_3(k) == -1
       efficient_rop_6(k) = rop_6(i);
       efficient_wob_6(k) = wob_6(i);
    end
end

efficient_rop_6(isnan(efficient_rop_6)) = []; % Remove NaN values
efficient_wob_6(isnan(efficient_wob_6)) = []; % Remove NaN values
c = polyfit(efficient_wob_6, efficient_rop_6, 1); % Create trendline
disp(['ROP = ' num2str(c(1)) '*WOB + ' num2str(c(2))]) % % Display efficient bit line

x= linspace(0,50); % WOB
y = c(1)*x + c(2); % Efficient Bit-line
```

```
figure(266)
subplot(1,3,1)
scatter(bitballing_3, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration_2, mdrkb, 20, 'filled', 'b')
scatter(wear_3, mdrkb, 20, 'filled', 'c')
scatter(ucs_event_3, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 6" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_6), max(mdrkb_6)])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 6])
grid on
box on

subplot(1,3,3)
scatter(wob_6, rop_6, 20, mdrkb_6, 'filled')
title('6" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth [mMD]');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

% Calculate new ROP from efficient bit-line for 8.5" hole section
k = 0;
for i = 1:length(mdrkb_6)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
        rop_new_6(k) = rop_6(i); % No events flagged, use original ROP
    else
        rop_new_6(k) = c(1)*wob_6(i) + c(2); % Flagged event, calculate ROP from Efficient bit-line
    end
end

subplot(1,3,2)
plot(rop_new_6, mdrkb)
hold on
plot(rop_6, mdrkb)
hold off
set(gca, 'YDir','reverse')
title('6" Section New ROP')
xlabel('ROP [m/hr]')
legend('New ROP', 'Original ROP')
xlim([0 120])
ylim([min(mdrkb_6), max(mdrkb_6)])
ylabel('Depth [mMD]')
box on
grid on

nameofplot = 'New_ROP_6_';
fig1 = figure(266);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);

%}
```

8.5" Hole Section

```
% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing_3(k) == -1  && vibration_2(k) == -1  && wear_3(k) == -1
       efficient_rop_8_5(k) = rop_8_5(i);
       efficient_wob_8_5(k) = wob_8_5(i);
    end
end
```

```
efficient_rop_8_5(isnan(efficient_rop_8_5)) = []; % Remove NaN values
efficient_wob_8_5(isnan(efficient_wob_8_5)) = []; % Remove NaN values
c = polyfit(efficient_wob_8_5, efficient_rop_8_5, 1); % Create trendline
disp(['ROP = ' num2str(c(1)) '*WOB + ' num2str(c(2))]) % % Display efficient bit line

x= linspace(0,50); % WOB
y = c(1)*x + c(2); % Efficient Bit-line

figure(26)
subplot(1,3,1)
scatter(bitballing_3, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration_2, mdrkb, 20, 'filled', 'b')
scatter(wear_3, mdrkb, 20, 'filled', 'c')
scatter(ucs_event_3, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 8.5" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_8_5), max(mdrkb_8_5)])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 6])
grid on
box on

subplot(1,3,3)
scatter(wob_8_5, rop_8_5, 20, mdrkb_8_5, 'filled')
title('8.5" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth [mMD]');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

% Calculate new ROP from efficient bit-line for 8.5" hole section
k = 0;
for i = 1:length(mdrkb_8_5)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
        rop_new_8_5(k) = rop_8_5(i); % No events flagged, use original ROP
    else
        rop_new_8_5(k) = c(1)*wob_8_5(i) + c(2); % Flagged event, calculate ROP from Efficient bit-line
    end
end

subplot(1,3,2)
plot(rop_new_8_5, mdrkb)
hold on
plot(rop_8_5, mdrkb)
hold off
set(gca, 'YDir','reverse')
title('8.5" Section New ROP')
xlabel('ROP [m/hr]')
legend('New ROP', 'Original ROP')
xlim([0 120])
ylim([min(mdrkb_8_5), max(mdrkb_8_5)])
ylabel('Depth [mMD]')
box on
grid on

nameofplot = 'New_ROP_8_5_';
fig1 = figure(26);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

12.25" Hole section

```
% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
```

```
    if bitballing_3(k) == -1  && vibration_2(k) == -1  && wear_3(k) == -1
       efficient_rop_12_25(k) = rop_12_25(i);
       efficient_wob_12_25(k) = wob_12_25(i);
    end
end

efficient_rop_12_25(isnan(efficient_rop_12_25)) = []; % Remove NaN values
efficient_wob_12_25(isnan(efficient_wob_12_25)) = []; % Remove NaN values
c = polyfit(efficient_wob_12_25, efficient_rop_12_25, 1); % Create trendline
disp(['ROP = ' num2str(c(1)) '*WOB + ' num2str(c(2))]) % % Display efficient bit line

x= linspace(0,50); % WOB
y = c(1)*x + c(2); % Trendline

figure(28)
subplot(1,3,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 12.25" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_12_25), max(mdrkb_12_25)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,3,3)
scatter(wob_12_25, rop_12_25, 20, mdrkb_12_25, 'filled')
title('12.25" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth [mMD]');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

% Calculate new ROP from efficient bit-line for 12.25" hole section
k = 0;
for i = 1:length(mdrkb_12_25)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
        rop_new_12_25(k) = rop_12_25(i); % No events flagged, use original ROP
    else
        rop_new_12_25(k) = c(1)*wob_12_25(i) + c(2); % Flagged event, calculate ROP from Efficient bit-line
    end
end

subplot(1,3,2)
plot(rop_new_12_25, mdrkb)
hold on
plot(rop_12_25, mdrkb)
hold off
set(gca, 'YDir','reverse')
title('12.25" Section New ROP')
xlabel('ROP [m/hr]')
ylim([min(mdrkb_12_25), max(mdrkb_12_25)])
ylabel('Depth [mMD]')
legend('New ROP', 'Original ROP')
grid on
box on

nameofplot = 'New_ROP_12_25_';
fig1 = figure(28);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

17.5" Hole Section

```
% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
       efficient_rop_17_5(k) = rop_17_5(i);
       efficient_wob_17_5(k) = wob_17_5(i);
    end
end

efficient_rop_17_5(isnan(efficient_rop_17_5)) = []; % Remove NaN values
efficient_wob_17_5(isnan(efficient_wob_17_5)) = []; % Remove NaN values
c = polyfit(efficient_wob_17_5, efficient_rop_17_5, 1); % Create trendline
disp(['ROP = ' num2str(c(1)) '*WOB + ' num2str(c(2))]) % % Display efficient bit line

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline


figure(30)
subplot(1,3,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 17.5" Section')
xlabel('')
ylabel('Depth [meter MD]')
ylim([min(mdrkb_17_5), max(mdrkb_17_5)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,3,3)
scatter(wob_17_5, rop_17_5, 20, mdrkb_17_5, 'filled')
title('17.5" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth meter MD');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

% Calculate new ROP from efficient bit-line for 12.25" hole section
k = 0;
for i = 1:length(mdrkb_17_5)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
        rop_new_17_5(k) = rop_17_5(i); % No events flagged, use original ROP
    else
        rop_new_17_5(k) = c(1)*wob_17_5(i) + c(2); % Flagged event, calculate ROP from Efficient bit-line
    end
end

subplot(1,3,2)
plot(rop_new_17_5, mdrkb)
hold on
plot(rop_17_5, mdrkb)
hold off
set(gca, 'YDir','reverse')
title('17.5" Section New ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
legend('New ROP', 'Original ROP')
ylim([min(mdrkb_17_5), max(mdrkb_17_5)])
grid on
box on

nameofplot = 'New_ROP_17_5_';
fig1 = figure(30);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
```

```
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

26" Hole Section

```
%

% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
       efficient_rop_26(k) = rop_26(i);
       efficient_wob_26(k) = wob_26(i);
    end
end

efficient_rop_26(isnan(efficient_rop_26)) = []; % Remove NaN values
efficient_wob_26(isnan(efficient_wob_26)) = []; % Remove NaN values
c = polyfit(efficient_wob_26, efficient_rop_26, 1); % Create trendline
disp(['ROP = ' num2str(c(1)) '*WOB + ' num2str(c(2))]) % % Display efficient bit line

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline

figure(32)
subplot(1,3,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 26" Section')
xlabel('')
ylabel('Depth [meter MD]')
ylim([min(mdrkb_26), max(mdrkb_26)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,3,3)
scatter(wob_26, rop_26, 20, mdrkb_26, 'filled')
title('26" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth meter MD');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

% Calculate new ROP from efficient bit-line for 26" hole section

k = 0;
for i = 1:length(mdrkb_26)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
        rop_new_26(k) = rop_26(i); % No events flagged, use original ROP
    else
        rop_new_26(k) = c(1)*wob_26(i) + c(2); % Flagged event, calculate ROP from Efficient bit-line
    end
end

subplot(1,3,2)
plot(rop_new_26, mdrkb)
hold on
plot(rop_26, mdrkb)
hold off
set(gca, 'YDir','reverse')
title('26" Section New ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
legend('New ROP', 'Original ROP')
ylim([min(mdrkb_26), max(mdrkb_26)])
grid on
box on
```

```
nameofplot = 'New_ROP_26_';
fig1 = figure(32);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);

%
```

36" Hole Section

```
%{

% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
       efficient_rop_36(k) = rop_36(i);
       efficient_wob_36(k) = wob_36(i);
    end
end

efficient_rop_36(isnan(efficient_rop_36)) = []; % Remove NaN values
efficient_wob_36(isnan(efficient_wob_36)) = []; % Remove NaN values
c = polyfit(efficient_wob_36, efficient_rop_36, 1); % Create trendline
disp(['ROP = ' num2str(c(1)) '*WOB + ' num2str(c(2))]) % % Display efficient bit line

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline

figure(33)
subplot(1,3,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 36" Section')
xlabel('')
ylabel('Depth [meter MD]')
ylim([min(mdrkb_36), max(mdrkb_36)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,3,3)
scatter(wob_36, rop_36, 20, mdrkb_36, 'filled')
title('36" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth meter MD');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

% Calculate new ROP from efficient bit-line for 36" hole section

k = 0;
for i = 1:length(mdrkb_36)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
        rop_new_36(k) = rop_36(i); % No events flagged, use original ROP
    else
        rop_new_36(k) = c(1)*wob_36(i) + c(2); % Flagged event, calculate ROP from Efficient bit-line
    end
end

subplot(1,3,2)
plot(rop_new_36, mdrkb)
hold on
plot(rop_36, mdrkb)
hold off
```

```
set(gca, 'YDir','reverse')
title('36" Section New ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
legend('New ROP', 'Original ROP')
ylim([min(mdrkb_36), max(mdrkb_36)])
grid on
box on

nameofplot = 'New_ROP_36_';
fig1 = figure(33);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);

%}
```

9 7/8" Hole Section

```
%{

% Find average value of efficient bit and plot
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
       efficient_rop_9_88(k) = rop_9_88(i);
       efficient_wob_9_88(k) = wob_9_88(i);
    end
end

efficient_rop_9_88(isnan(efficient_rop_9_88)) = []; % Remove NaN values
efficient_wob_9_88(isnan(efficient_wob_9_88)) = []; % Remove NaN values
c = polyfit(efficient_wob_9_88, efficient_rop_9_88, 1); % Create trendline
disp(['ROP = ' num2str(c(1)) '*WOB + ' num2str(c(2))]) % % Display efficient bit line

x= linspace(0,20); % WOB
y = c(1)*x + c(2); % Trendline

figure(333)
subplot(1,3,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 9 7/8" Section')
xlabel('')
ylabel('Depth [meter MD]')
ylim([min(mdrkb_9_88), max(mdrkb_9_88)])
legend('Bitballing', 'Vibration', 'Wear', 'UCS')
xlim([0 6])
grid on
box on

subplot(1,3,3)
scatter(wob_9_88, rop_9_88, 20, mdrkb_9_88, 'filled')
title('9 7/8" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth meter MD');
set( depth, 'YDir', 'reverse' );
axis([0 20 0 120])
hold on
h = plot(x,y, '--r','linewidth', 2);
legend(h, 'Efficient Bit')
hold off
grid on
box on

% Calculate new ROP from efficient bit-line for 9_88" hole section

k = 0;
for i = 1:length(mdrkb_9_88)
    k = k+1;
    if bitballing_3(k) == -1 && vibration_2(k) == -1 && wear_3(k) == -1
        rop_new_9_88(k) = rop_9_88(i); % No events flagged, use original ROP
    else
```

```
        rop_new_9_88(k) = c(1)*wob_9_88(i) + c(2); % Flagged event, calculate ROP from Efficient bit-line
    end
end

subplot(1,3,2)
plot(rop_new_9_88, mdrkb)
hold on
plot(rop_9_88, mdrkb)
hold off
set(gca, 'YDir','reverse')
title('9 7/8" Section New ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
legend('New ROP', 'Original ROP')
ylim([min(mdrkb_9_88), max(mdrkb_9_88)])
grid on
box on

nameofplot = 'New_ROP_9_88_';
fig1 = figure(333);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);

%}
```

Concatinate ROPs, Calculate new time and plot new ROP

```
% Concatinate the new ROPs
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 9.88;
        rop_new(k) = rop_new_9_88(i);
    elseif bdia(k) == 36;
        rop_new(k) = rop_new_36(i);
    elseif bdia(k) == 26;
        rop_new(k) = rop_new_26(i);
    elseif bdia(k) == 17.5;
        rop_new(k) = rop_new_17_5(i);
    elseif bdia(k) == 12.25;
        rop_new(k) = rop_new_12_25(i);
    elseif bdia(k) == 8.5;
        rop_new(k) = rop_new_8_5(i);
    elseif bdia(k) == 6;
        rop_new(k) = rop_new_6(i);
    else
        rop_new(k) = rop(i);
    end
end

figure(34)
plot(rop_new, mdrkb)
hold on
plot(rop, mdrkb)
hold off
set(gca, 'YDir','reverse')
title(' New ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
legend('New ROP', 'Original ROP')
grid on
box on
% Calculate time difference between original and new ROP

k=0;
for i=1:length(mdrkb)
    k=k+1;
    time_orig(k)= 1/rop(i);
    time_n(k) = 1/rop_new(i);
end

time_orig(isinf(time_orig)) = []; % Remove inf values
time_n(isinf(time_n)) = []; % Remove inf values

time_original = sum(time_orig)
time_new = sum(time_n)
figure(35)
subplot(1,2,1)
scatter(bitballing_3, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration_2, mdrkb, 20, 'filled', 'b')
```

```
scatter(wear_3, mdrkb, 20, 'filled', 'c')
scatter(ucs_event_3, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

subplot(1,2,2)
plot(rop_new, mdrkb, 'linewidt', 0.5)
hold on
plot(rop, mdrkb, 'linewidt', 0.5)

num1 = num2str(round(time_original));
num2 = num2str(round(time_new));

   txt1 = ['Original:  '];
   txt2 = ['New: '];
   txt3 = [' hrs'];
   txt4 = [' '];

txt32 = strcat(txt1, txt4, num1, txt4, txt3);
text(110, 570, txt32, 'fontsize', 10,'Interpreter','latex')

txt42 = strcat(txt2, txt4, num2, txt3);
text(110, 770, txt42, 'fontsize', 10,'Interpreter','latex')

hold off

set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylim([0, TD+200])
xlim([0, 150])
legend('New ROP', 'Original ROP')
grid on
box on

nameofplot = 'time_efficient_drilling_';
fig1 = figure(35);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Bit runs and NPT

```
% Import bit runs

 bit_data = xlsread('Ready_34_4-15 S  BIT_CHANGES.xlsx'); % Excel file containing bit changes

% bit_data = xlsread('Ready_35_3-6  BIT_CHANGES.xlsx'); % Excel file containing bit changes

%bit_data = xlsread('Ready_34_4-15 S  BIT_CHANGES.xlsx'); % Excel file containing bit changes

bit_nr = bit_data(:,1);                                 % Bit number
bit_in = bit_data(:,2);                                 % Depth in bit, [mMD]
bit_out = bit_data(:,3);                                % Depth out bit, [mMD]
bit_size = bit_data(:,4);                               % Bit size, [in]
bit_core = bit_data(:,7);                               % Coring or not
total_bits = length(bit_nr);                            % Total number of bits

k=0;
for i = 1:length(bit_nr)
    k=k+1;
    if bit_core(k) == 2
        bit_in_core(k) = bit_in(i);
        bit_out_core(k) = bit_out(i);
    else
        bit_in_core(k) = NaN;
        bit_out_core(k) = NaN;
    end
end

% Plot bit-runs and drilling events/NPT
figure(36)
subplot(1,3,1)
scatter(bit_nr, bit_in, 30, 'filled')
```

```
  hold on
  scatter(bit_nr, bit_out, 30, 'filled')
  scatter(bit_nr, bit_in_core, 30, 'filled')
  scatter(bit_nr, bit_out_core, 30, 'filled')
  hold off
  set(gca, 'YDir','reverse')
  xlim([0 length(bit_nr)+1])
  title('Bit Runs')
  xlabel('Bit number')
  ylabel('Depth [mMD]')
  ylim([0, TD+200])
  legend('Bit In', 'Bit Out', 'Core run in', 'Core run Out')
  grid on
  box on

  subplot(1,3,2)
  plot(bdia, mdrkb)
  set(gca, 'YDir','reverse')
  xlim([0 40])
  title('Bit Size')
  xlabel('Sit size [in]')
  ylim([0, TD+200])
  grid on
  box on

  subplot(1,3,3)
  scatter(bitballing_3, mdrkb, 20, 'filled', 'r')
  hold on
  scatter(vibration_2, mdrkb, 20, 'filled', 'b')
  scatter(wear_3, mdrkb, 20, 'filled', 'c')
  scatter(ucs_event_3, mdrkb, 20, 'filled', 'm')
  hold off
  set(gca, 'YDir','reverse')
  title('Drilling Events & NPT')
  xlabel('')
  ylim([0, TD+200])
  legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
  xlim([0 5])
  grid on
  box on

  nameofplot = 'core_bit_drillingevents_NPT_';
  fig1 = figure(36);
  fig1.Renderer = 'Painters';
  filetype = '.pdf';
  txtconcat = strcat(nameofplot, wellname, filetype);

  h=gcf;
  set(h, 'PaperOrientation', 'landscape');
  set(h, 'Paperposition', [0 0 13 8]);
  print(gcf, '-dpdf', '-fillpage', txtconcat);
  % Calculate time for bit change
  trip_speed = 500; % Tripping speed assuming 500 m/hr

  % Look at figure 36 and study which bits are changed due to drilling
  % events. Relate bit changes to bit size and drilling events plot.

  change_bit = [0,0,0,0,0,0,0]; % Manually insert the bit number to calculate cost of bit change. Other bits are entered
as zero.

  if length(change_bit) ~= length(bit_nr)
      disp(['Error! The change bit vector must contain: ' num2str(total_bits) ' numbers'])
  else
      disp('Correct bit change inputs.')
  end

  k=0;
  for i = 1:length(bit_nr)
      k = k+1;
      if change_bit (k) == bit_nr(k);
          depth_bit_change(k) = bit_out(k);
          depth_bit_change_1(k) = bit_out(k);
      else
          depth_bit_change(k) = NaN;
          depth_bit_change_1(k) = NaN;
      end
  end

  depth_bit_change_1(isnan(depth_bit_change_1)) = []; % Remove NaN values
  disp(['Depth of bit change: ' num2str(depth_bit_change_1) ' mMD'])

  k = 0;
  for i = 1:length(bit_nr)
      k=k+1;
      if depth_bit_change(k) == bit_out(k);
          trip_time(k) = (bit_out(k)*2)/trip_speed;
      else
          trip_time(k) = 0;
```

```
    end
end

disp(['Trip times: ' num2str(round(trip_time,1))])
total_trip_time = sum(trip_time);
disp(['Total tripping time: ' num2str(round(total_trip_time,2)) ' hours'])
```

Calculate total lost time due to inefficient drilling and bit changes

```
% Lost time due to inefficient drilling
rop_lost_time = time_original - time_new;
disp(['Lost time due to inefficient drilling: ' num2str(round(rop_lost_time,1)) ' hours'])

% Lost time due to bit change
disp(['Lost time due to bit change: ' num2str(round(total_trip_time,1)) ' hours'])

% Total lost time
total_lost_time = rop_lost_time + total_trip_time;
disp(['Total lost time: ' num2str(round(total_lost_time,1)) ' hours'])
% Plot bit runs, bit size, new rop, drilling events and NPT and lost time
figure(37)
subplot(1,4,1)
scatter(bit_nr, bit_in, 30, 'filled')
hold on
scatter(bit_nr, bit_out, 30, 'filled')
scatter(bit_nr, bit_in_core, 30, 'filled')
scatter(bit_nr, bit_out_core, 30, 'filled')
set(gca, 'YDir','reverse')
xlim([0 length(bit_nr)+1])
title('Bit Runs')
xlabel('Bit number')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bit In', 'Bit Out', 'Core run in', 'Core run Out')
num1 = num2str(round(total_trip_time,2));
txt1 = ['Lost time:  '];
txt3 = [' hrs'];
txt4 = [' '];
txt32 = strcat(txt1, txt4);
text((length(bit_nr)-3), 800, txt32, 'fontsize', 10,'Interpreter','latex')
txt33 = strcat(num1, txt4, txt3);
text((length(bit_nr)-3), 900, txt33, 'fontsize', 10,'Interpreter','latex')
hold off
box on
grid on

subplot(1,4,2)
plot(bdia, mdrkb)
xlim([0 40])
title('Bit Size')
xlabel('Bit Size [in]')
ylim([0, TD+200])
set(gca, 'YDir','reverse')
grid on
box on

subplot(1,4,3)
plot(rop_new, mdrkb, 'linewidt', 0.5)
hold on
plot(rop, mdrkb, 'linewidt', 0.5)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylim([0, TD+200])
xlim([0, 150])
legend('New ROP', 'Original ROP')
num1 = num2str(round(rop_lost_time,2));
txt1 = ['Lost time:  '];
txt3 = [' hrs'];
txt4 = [' '];
txt32 = strcat(txt1, txt4);
text(100, 800, txt32, 'fontsize', 10,'Interpreter','latex')
txt33 = strcat(num1, txt4, txt3);
text(100, 900, txt33, 'fontsize', 10,'Interpreter','latex')
hold off
grid on
box on

subplot(1,4,4)
scatter(bitballing_3, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration_2, mdrkb, 20, 'filled', 'b')
scatter(wear_3, mdrkb, 20, 'filled', 'c')
scatter(ucs_event_3, mdrkb, 20, 'filled', 'm')
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
```

```
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
num1 = num2str(round(total_lost_time,2));
txt1 = ['Total lost time:  '];
txt3 = [' hrs'];
txt4 = [' '];
txt32 = strcat(txt1, txt4);
text(2.2, 800, txt32, 'fontsize', 10,'Interpreter','latex')
txt33 = strcat(num1, txt4, txt3);
text(2.2, 900, txt33, 'fontsize', 10,'Interpreter','latex')
hold off
grid on
box on

nameofplot = 'total_lost_time_';
fig1 = figure(37);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Export to Excel

```
% Transpose vectors before exporting
Bitballing = transpose(bitballing_3);
Vibration = transpose(vibration_2);
Bit_Wear = transpose(wear_3);
Hard_Rock = transpose(ucs_event_3);

Bitballing(Bitballing == -1) = 0; % Set non flagged events equal to zero.
Vibration(Vibration == -1) = 0; % Set non flagged events equal to zero.
Bit_Wear(Bit_Wear == -1) = 0; % Set non flagged events equal to zero.
Hard_Rock(Hard_Rock == -1) = 0; % Set non flagged events equal to zero.
T = table(mdrkb, tvdrkb, mse, mse_corr, ds, mse_ds, Bitballing, Vibration, Bit_Wear, Hard_Rock, rop, wob, tqa, bdia, 
ecd, sppa, sonic);
t = table(total_trip_time, rop_lost_time, total_lost_time, cut_off, ucs_cutoff_1);
name = '_NPT_Drilling_Events_Detection';
file = '.xlsx';
filename = strcat(wellname, name, file);
writetable(T,filename,'Sheet',1)
writetable(t, filename, 'sheet',2)
```

### *9.2 Updated MATLAB Workflow WOB-Method*

# Detection of NPT and Drilling Events on the NCS from Depth-based Drilling Mechanics Data

## Updated Workflow WOB-Method

```
clear all
clc
```

Import Well Data

```
% North Sea

%{
% 34/4-15 S
wellname = '34_4-15 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\34_4-15 S\34_4-15 S
Ready');
drilling_data = xlsread('Ready_34_4-15 S  DRILLING DATA - MUDLOG.xlsx'); % Drilling data
composite_data =xlsread('Ready_34_4-15 S COMPOSITE_LOG.xlsx'); % Composite data
bit_data = xlsread('Ready_34_4-15 S  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 34/4-15 A
wellname = '34_4-15 A';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\34_4-15 A\34_4-15 A
Ready');
drilling_data = xlsread('Ready_34_4-15 A  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('34_4-15 A COMPOSITE_LOG.xlsx');
bit_data = xlsread('Ready_34_4-15 A  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 34/4-16 S
wellname = '34_4-16 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\34_4-16 S');
drilling_data = xlsread('Ready_34_4-16 S  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_34_4-16 S COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_34_4-16 S  BIT_CHANGES.xlsx'); % Bit data
%}


%{
% 33/6-3 S
wellname = '33_6-3 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\33_6-3 S\33_6-3 S Ready');
drilling_data = xlsread('Ready_33_6-3 S  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_33_6-3 S COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_33_6-3 S  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 35/3-6
wellname = '35_3-6';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_3-6\35_3-6 Ready');
drilling_data = xlsread('Ready_35_3-6  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_35_3-6 COMPOSITE_LOG.xlsx');
bit_data = xlsread('Ready_35_3-6  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 35/3-4
wellname = '35_3-4';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_3-4\35_3-4 Ready');
drilling_data = xlsread('Ready_35_3-4  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_35_3-4 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_35_3-4  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 35/8-4
wellname = '35_8-4';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_8-4\35_8-4 Ready');
drilling_data = xlsread('Ready_35_8-4  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_35_8-4 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_35_8-4  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 35/8-5 S
```

```
wellname = '35_8-5 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_8-5 S\35_8-5 S Ready');
drilling_data = xlsread('Ready_35_8-5 S  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_35_8-5 S COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_35_8-5 S  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 35/9-8
wellname = '35_9-8';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_9-8\35_9-8 Ready');
drilling_data = xlsread('Ready_35_9-8  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_35_9-8 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_35_9-8  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 35/9-11 S
wellname = '35_9-11 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_9-11 S\35_9-11 S
Ready');
drilling_data = xlsread('Ready_35_9-11 S  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_35_9-11 S COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_35_9-11 S  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 15/12-18 S
wellname = '15_12-18 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\15_12-18 S Ready');
drilling_data = xlsread('Ready_15_12-18 S DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_15_12-18 S COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_15_12-18 S BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 15/12-21
wellname = '15_12-21';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\15_12-21 Ready');
drilling_data = xlsread('Ready_15_12-21  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_15_12-21 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_15_12-21  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 15/12-23
wellname = '15_12-23';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\15_12-23 Ready');
drilling_data = xlsread('Ready_15_12-23 DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_15_12-23 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_15_12-23 BIT_CHANGES.xlsx'); % Bit data
%}


%%{
% 15/12-21

%%{
% 15/12-23


% Norwegian Sea

%{
% 6406/1-1
wellname = '6406_1-1';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6406_1-1 Ready');
drilling_data = xlsread('Ready_6406_1-1  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('READY_6406_1-1 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_6406_1-1  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 6406/2-3
wellname = '6406_2-3';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6406_2-3 Ready');
drilling_data = xlsread('Ready_6406_2-3  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_6406_2-3 Composite.xlsx');
bit_data = xlsread('Ready_6406_2-3  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 6406/2-5
wellname = '6406_2-5';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6406_2-5 Ready');
drilling_data = xlsread('Ready_6406_2-5  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_6406_2-5 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_6406_2-5  BIT_CHANGES.xlsx'); % Bit data
```

```
%}


%{
% 6406/2-7
wellname = '6406_2-7';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6406_2-7 Ready');
drilling_data = xlsread('Ready_6406_2-7  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_6406_2-7 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_6406_2-7  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 6506/11-6
wellname = '6506_11-6';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6506_11-6 Ready');
drilling_data = xlsread('Ready_6506_11-6  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_6506_11-6 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_6506_11-6  BIT_CHANGES.xlsx'); % Bit data
%}


%{
% 6407/8-4 S
wellname = '6407_8-4 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6407_8-4 S Ready');
drilling_data = xlsread('Ready_6407_8-4 S  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_6407_8-4 S COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_6407_8-4 S  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 6407/8-5 S
wellname = '6407_8-5 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6407_8-5 S Ready');
drilling_data = xlsread('Ready_6407_8-5 S  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_6407_8-5 S COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_6407_8-5 S  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 6407/8-6
wellname = '6407_8-6';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6407_8-6 Ready');
drilling_data = xlsread('Ready_6407_8-6 DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_6407_8-6 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_6407_8-6  BIT_CHANGES.xlsx'); % Bit data
%}




%{
% 6407/8-7
wellname = '6407_8-7';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6407_8-7 Ready');
drilling_data = xlsread('Ready_6407_8-7  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_6407_8-7 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_6407_8-7  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 6407/9-9
wellname = '6407_9-9';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6407_9-9 Ready');
drilling_data = xlsread('Ready_6407_9-9  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_6407_8-7 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_6407_9-9  BIT_CHANGES.xlsx'); % Bit data
%}
```

Name Drilling Data and Composite logs variables

```
% Name drilling data
tvdrkb = drilling_data(:,1);
mdrkb = drilling_data(:,2);
bdia = drilling_data(:,3);
rop = drilling_data(:,4);
hkla = drilling_data(:,5);
hklx = drilling_data(:,6);
wob = drilling_data(:,7);
tqa = drilling_data(:,8);
tqx = drilling_data(:,9);
rpmb = drilling_data(:,10);
sppa = drilling_data(:,11);
tva = drilling_data(:,12);
mfoa = drilling_data(:,13);
```

```
mfia = drilling_data(:,14);
mwout = drilling_data(:,15);
mwin = drilling_data(:,16);
mtoa = drilling_data(:,17);
mtia = drilling_data(:,18);
ecd = drilling_data(:,19);
ppore = drilling_data(:,20);
lith = drilling_data(:,21);
dexp = drilling_data(:,22);
TD = max(mdrkb);
% Name Composite Log Data
gammaray_raw = composite_data(:, 5);
gammaray_depth_raw = composite_data(:, 4);
sonic_raw = composite_data(:, 7);
sonic_depth_raw = composite_data(:, 6);

% Create equal length arrays for composite logs and drilling data
gammaray_raw(isnan(gammaray_raw)) = []; % Remove NaN values
gammaray_depth_raw(isnan(gammaray_depth_raw)) = []; % Remove NaN values
sonic_raw(isnan(sonic_raw)) = []; % Remove NaN values
sonic_depth_raw(isnan(sonic_depth_raw)) = []; % Remove NaN values

gammaray = interp1(gammaray_depth_raw, gammaray_raw, mdrkb);
sonic = interp1(sonic_depth_raw, sonic_raw, mdrkb);
```

Plot the Variables

```
figure(1) % Plot drilling parameters: ROP, WOB, RPM, Torque, Mud flow in, SPPA and ECD, Bit-size

subplot(1,8,1)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
grid on
box on

subplot(1,8,2)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
grid on
box on

subplot(1,8,3)
plot(rpmb, mdrkb)
set(gca, 'YDir','reverse')
title('RPM')
xlabel('RPM')
grid on
box on

subplot(1,8,4)
plot(tqa, mdrkb)
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
grid on
box on

subplot(1,8,5)
plot(mfia, mdrkb)
set(gca, 'YDir','reverse')
title('Mud Flow In')
xlabel('Flow [liter/min]')
grid on
box on

subplot(1,8,6)
plot(sppa, mdrkb)
set(gca, 'YDir','reverse')
title('SPPA')
xlabel('SPPA [bar]')
grid on
box on

subplot(1,8,7)
plot(ecd, mdrkb)
set(gca, 'YDir','reverse')
title('ECD')
xlabel('ECD [s.g.]')
xlim ([0 2.5])
```

```
grid on
box on

subplot(1,8,8)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [inch]')
xlim([0 40])
grid on
box on

nameofplot = 'Drilling_Data_figure_';
fig1 = figure(1);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% Check if torque is available
if sum(tqa)>0
    disp('Torque is available')
else
    disp('Torque is not available')
end
figure(2) % Plot Gamma-ray, Sonic log and Lithology
subplot(1,3,1)
plot(gammaray, mdrkb)
set(gca, 'YDir','reverse')
title('Gamma-Ray')
xlabel('Gamma-ray [gAPI]')
ylabel('Depth [mMD]')
ylim([0 TD+200])
grid on
box on

subplot(1,3,2)
plot(sonic, mdrkb)
set(gca, 'YDir','reverse')
title('Sonic Log')
xlabel('Interval velocity [µs/ft]')
ylim([0 TD+200])
grid on
box on

L = zeros(length(mdrkb), 1);
k = 0;
for i = 1:length(mdrkb);
    k=k+1;
    if lith(k) >= 500 && lith(k) < 700 % Claystone
        claystone(k) = L(i)+1;
    else
        claystone(k) = L(i) - 1;
    end
end

k = 0;
for i = 1:length(mdrkb);
    k=k+1;
    if lith(k) >= 300 && lith(k) < 400 % Sandstone
        sandstone(k) = L(i)+2;
    else
        sandstone(k) = L(i) - 1;
    end
end

k = 0;
for i = 1:length(mdrkb);
    k=k+1;
    if lith(k) >= 700 && lith(k) < 800 % Carbonates
        carbonates(k) = L(i)+3;
    else
        carbonates(k) = L(i) - 1;
    end
end

subplot(1,3,3)
scatter(claystone, mdrkb, 20, 'filled', 'g')
hold on
scatter(sandstone, mdrkb, 20, 'filled', 'y')
scatter(carbonates, mdrkb, 20, 'filled', 'b')
set(gca, 'YDir','reverse')
title('Lithology')
xlabel('Lithology')
hold off
```

```
xlim([0, 4])
legend('Claystone', 'Sandstone', 'Carbonates')
ylim([0 TD+200])
grid on
box on

nameofplot = 'Gamma_sonic_lithology_';
fig1 = figure(2);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Calculate Mechanical Specific Energy

```
wob_lbs = wob*2204.62; % Convert from tonnes to lbs
tqa_ft_lbs = tqa*737.56; % Convert from knm to ft-lbs
rop_ft_hr = rop*3.28; % Convert from m/hr to ft/hr
factor = 0.35; % Efficiency factor

mse_psi = factor * (wob_lbs./((pi/4).*(bdia.^2)) + (120*pi.*rpmb.*tqa_ft_lbs)./(((pi/4).*(bdia.^2)).*rop_ft_hr)); %
Mechanical specific energy, psi
mse = mse_psi./145; % Convert MSE from PSI to MPa
```

Plot MSE and Drilling Data

```
figure(3) % Plot drilling parameters: MSE, ROP, WOB, Torque and bit size
subplot(1,5,1)
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('MSE')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
xlim([0 500]) % xlim([0 1500])
grid on
box on

subplot(1,5,2)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
grid on
box on

subplot(1,5,3)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
grid on
box on

subplot(1,5,4)
plot(tqa, mdrkb)
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
grid on
box on

subplot(1,5,5)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [Inch]')
xlim([0 40])
grid on
box on

nameofplot = 'drillingparameters_';
fig1 = figure(3);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
figure(4) % Plot drilling parameters: MSE, ROP, WOB, RPM, Torque, Mud flow in, SPPA, ECD and Bit-size
subplot(1,9,1)
```

```
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('MSE')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
xlim([0 500]) % xlim([0 1500])
grid on
box on

subplot(1,9,2)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
xlim([0 200])
grid on
box on

subplot(1,9,3)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
grid on
box on

subplot(1,9,4)
plot(rpmb, mdrkb)
set(gca, 'YDir','reverse')
title('RPM')
xlabel('RPM')
grid on
box on

subplot(1,9,5)
plot(tqa, mdrkb)
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
grid on
box on

subplot(1,9,6)
plot(mfia, mdrkb)
set(gca, 'YDir','reverse')
title('Mud Flow In')
xlabel('Flow [liter/min]')
grid on
box on

subplot(1,9,7)
plot(sppa, mdrkb)
set(gca, 'YDir','reverse')
title('SPPA')
xlabel('SPPA [bar]')
grid on
box on

subplot(1,9,8)
plot(ecd, mdrkb)
set(gca, 'YDir','reverse')
title('ECD')
xlabel('ECD [s.g.]')
grid on
box on

subplot(1,9,9)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [inch]')
xlim([0 40])
grid on
box on

nameofplot = 'drillingparameters2_';
fig1 = figure(4);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

### Plot MSE and Create MSE Corrected

```
figure(5) % Plot Mechanical specific energy
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('Mechanical specific energy')
xlabel('Mechanical specific energy [MPa]')
ylabel('Depth [mMD]')
xlim([0 1500])
grid on
box on

nameofplot = 'Mechanical_specific_energy_';
fig1 = figure(5);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Create Baseline for MSE*

```
% Manually insert the baselines.

% North Sea

%{
% 34/4-15 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2001];
baseline_mse_max = [2000, max(mdrkb)];
mse_min = [8, 8];
mse_max = [8, 140];
%}

%{
% 34/4-15 A
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2001];
baseline_mse_max = [2000, max(mdrkb)];
mse_min = [8, 8];
mse_max = [8, 130];
%}

%{
% 34/4-16 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2300];
baseline_mse_max = [2300, max(mdrkb)];
mse_min = [8, 8];
mse_max = [8, 100];
%}

%{
% 33/6-3 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1601];
baseline_mse_max = [1600, max(mdrkb)];
mse_min = [10, 12];
mse_max = [12, 110];
%}

%{
% 35/3-6
baselines = 3; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1130, 3081];
baseline_mse_max = [1129, 3080, max(mdrkb)];
mse_min = [2, 2, 200];
mse_max = [2, 200, 170];
%}

%{
% 35/3-4
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2001];
baseline_mse_max = [2000, max(mdrkb)];
mse_min = [2, 2];
mse_max = [2, 70];
%}

%{
% 35/8-4
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2501];
baseline_mse_max = [2500, max(mdrkb)];
```

```
mse_min = [5, 20];
mse_max = [20, 160];
%}

%{
% 35/8-5 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2001];
baseline_mse_max = [2000, max(mdrkb)];
mse_min = [5, 5];
mse_max = [5, 140];
%}

%{
% 35/9-8
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1001];
baseline_mse_max = [1000, max(mdrkb)];
mse_min = [5, 5];
mse_max = [5, 230];
%}

%{
% 35/9-11 S
baselines = 3; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1700.25, 3200.25];
baseline_mse_max = [1700, 3200, max(mdrkb)];
mse_min = [5, 5, 190];
mse_max = [5, 190, 550];
%}

%{
% 15/12-18 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2501];
baseline_mse_max = [2500, max(mdrkb)];
mse_min = [10, 15];
mse_max = [15, 550];
%}

%{
% 15/12-21
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2301];
baseline_mse_max = [2300, max(mdrkb)];
mse_min = [10, 35];
mse_max = [35, 140];
%}

%{
% 15/12-23
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2301];
baseline_mse_max = [2300, max(mdrkb)];
mse_min = [1, 15];
mse_max = [15, 60];
%}


% Norwegian Sea

%{
% 6406/1-1
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 3601];
baseline_mse_max = [3600, max(mdrkb)];
mse_min = [1, 50];
mse_max = [50, 500];
%}

%{
% 6406/2-3
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 3251];
baseline_mse_max = [3250, max(mdrkb)];
mse_min = [5, 5];
mse_max = [5, 500];
%}

%{
% 6406/2-5
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 3201];
baseline_mse_max = [3200, max(mdrkb)];
mse_min = [5, 40];
mse_max = [40, 400];
%}
```

```
%{
% 6406/2-7
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 3001];
baseline_mse_max = [3000, max(mdrkb)];
mse_min = [5, 25];
mse_max = [25, 400];
%}

%{
% 6506/11-6
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 3101];
baseline_mse_max = [3100, max(mdrkb)];
mse_min = [5, 40];
mse_max = [40, 600];
%}

%{
% 6407/8-4 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1901];
baseline_mse_max = [1900, max(mdrkb)];
mse_min = [5, 15];
mse_max = [15, 200];
%}

%{
% 6407/8-5 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1801];
baseline_mse_max = [1800, max(mdrkb)];
mse_min = [5, 20];
mse_max = [20, 150];
%}

%{
% 6407/8-6
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1901];
baseline_mse_max = [1900, max(mdrkb)];
mse_min = [5, 20];
mse_max = [20, 80];
%}

%{
% 6407/8-7
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1901];
baseline_mse_max = [1900, max(mdrkb)];
mse_min = [5, 15];
mse_max = [15, 120];
%}

%{
% 6407/9-9
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1619];
baseline_mse_max = [1618, max(mdrkb)];
mse_min = [5, 15];
mse_max = [15, 70];
%}


% Create baselines
if baselines >= 1
    baseline_1 = baseline_mse_min(1):1:baseline_mse_max(1); % baseline_mse_min(1):1:baseline_mse_max(1);
    baseline1 = [baseline_mse_min(1), baseline_mse_max(1)];
    mse_1 = [mse_min(1), mse_max(1)];
    baseline_mse_1 = (interp1(baseline1, (mse_1), baseline_1));
end

if baselines >= 2
    baseline_2 = baseline_mse_min(2):1:baseline_mse_max(2); % baseline_mse_min(2):1:baseline_mse_max(2);
    baseline2 = [baseline_mse_min(2), baseline_mse_max(2)];
    mse_2 = [mse_min(2), mse_max(2)];
    baseline_mse_2 = (interp1(baseline2, (mse_2), baseline_2));
end

if baselines >= 3
    baseline_3 = baseline_mse_min(3):1:baseline_mse_max(3); % baseline_mse_min(3):1:baseline_mse_max(3);
    baseline3 = [baseline_mse_min(3), baseline_mse_max(3)];
    mse_3 = [mse_min(3), mse_max(3)];
    baseline_mse_3 = (interp1(baseline3, (mse_3), baseline_3));
end

if baselines == 1
   baseline_mse = [baseline_mse_1];
```

```
elseif baselines == 2
    baseline_mse = [baseline_mse_1, baseline_mse_2];
elseif baselines == 3
   baseline_mse = [baseline_mse_1, baseline_mse_2, baseline_mse_3];
end
```

*Plot Baseline and MSE*

```
figure(6)
subplot(1,2,1)
plot(mse, mdrkb)
hold on
plot(baseline_mse, mdrkb, 'linewidth', 1.5)
hold off
set(gca, 'YDir','reverse')
title('Mechanical specific energy')
xlabel('Mechanical specific energy [MPa]')
ylabel('Depth [mMD]')
legend('MSE', 'Baseline MSE')
xlim([0 1500])
grid on
box on

subplot(1,2,2)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit Size [in]')
xlim([0 40])
grid on
box on

nameofplot = 'Mechanical_specific_energy_baseline_';
fig1 = figure(6);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Create and Plot MSE Corrected*

```
baseline_mse_transposed = transpose(baseline_mse); % Transpose the baseline vector
mse_corr = mse - baseline_mse_transposed; % Create MSE corrected
mse_corr(mse_corr < 0) = 0; % Remove negative MSE values

figure(7)
subplot(1,2,1)
plot(mse, mdrkb)
hold on
plot(baseline_mse, mdrkb, 'linewidth', 1.5)
hold off
set(gca, 'YDir','reverse')
title('Mechanical Specific Energy')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
legend('MSE', 'Baseline MSE')
xlim([0 1500])
grid on
box on

subplot(1,2,2)
plot(mse_corr, mdrkb)
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('MSE [MPa]')
xlim([0 1500])
grid on
box on

nameofplot = 'MSE_Corrected_';
fig1 = figure(7);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

### Calculate Drilling Strength and MSE/DS ratio

*Calculate Depth of Cut and Drilling Strength*

```
doc = rop_ft_hr./(rpmb.*5); % Depth of cut, [inch]
ds_psi = wob_lbs./(bdia.*doc); % Drilling strength, [psi]
ds = ds_psi /145; % Convert from psi to MPa
mse_ds = mse./ds; % Ratio between Mechanical spesific energy and drilling strength
```

*Plot MSE, MSE Corrected, DS and MSE/DS*

```
figure(8)
subplot(1,4,1)
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('MSE')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
xlim([0 1500])
grid on
box on

subplot(1,4,2)
plot(mse_corr, mdrkb)
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('MSE [MPa]')
xlim([0 1500])
grid on
box on

subplot(1,4,3)
plot(ds, mdrkb)
set(gca, 'YDir','reverse')
title('DS')
xlabel('DS [MPa]')
xlim ([0 3000])  %([0 1500])
grid on
box on

subplot(1,4,4)
plot(mse_ds, mdrkb)
set(gca, 'YDir','reverse')
title('MSE/DS')
xlabel('MSE/DS')
xlim([0, 10])
grid on
box on

nameofplot = 'MSE_MSE_Corr_DS_MSEDS_';
fig1 = figure(8);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

## Estimate UCS

```
% Estimate UCS according to lithology
%Vp = (1./sonic)./(3.2808*10^(-3)); % Convert µs/ft to m/s

for i = 1:length(mdrkb)
    if claystone (i) == 1
        UCS(i) = 1.35*(304.8/sonic(i))^2.6; % Globally Shale
    elseif carbonates(i) == 3
        UCS (i) = ((7682/sonic(i))^1.82)/145; % MilitzerStoll Lime
    elseif sandstone(i) == 2
        if tvdrkb(i) < 2000
            UCS(i) = 1.4138*10^7*(sonic(i)^-3); % GulfCoast WeakSand
        elseif 2000 <= tvdrkb(i)
            UCS(i) = 1200*exp(-0.036*sonic(i)); % BowenBasinAustralia_Sand
        end
    else
        UCS(i) = NaN;
    end
end

for i = 1:length(mdrkb)
    if UCS(i) >= 1000;
        UCS(i) = NaN;
    end
end

figure(18)
```

```
subplot(1,3,1)
plot(UCS, mdrkb)
title('UCS')
set(gca, 'YDir','reverse')
xlabel('UCS [MPa]')
ylabel('Depth [mMD]')
ylim([0, TD+200])
grid on
box on

subplot(1,3,2)
scatter(claystone, mdrkb, 20, 'filled', 'g')
hold on
scatter(sandstone, mdrkb, 20, 'filled', 'y')
scatter(carbonates, mdrkb, 20, 'filled', 'b')
set(gca, 'YDir','reverse')
title('Lithology')
xlabel('Lithology')
hold off
xlim([0, 4])
ylim([0 TD+200])
legend('Claystone', 'Sandstone', 'carbonates')
grid on
box on

subplot(1,3,3)
plot(sonic, mdrkb)
set(gca, 'YDir','reverse')
title('Sonic')
xlabel('Interval velocity [µs/ft]')
ylim([0 TD+200])
grid on
box on

nameofplot = 'UCS_lithology_sonic_';
fig1 = figure(18);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% Sort UCS estimations according to Hardness Category
ucs = transpose(UCS);
for i = 1:length(mdrkb)
    if ucs(i) <= 0.6 && ucs(i) > 0
        hardness(i) = 1;                % Soil < 0.6 MPa
    elseif ucs(i) <= 1.25 && ucs(i) > 0.6
        hardness(i) = 2;                % Very Soft Rock: 0.6 - 1.25 MPa
    elseif ucs(i) <= 5 && ucs(i) > 1.25
        hardness(i) = 3;                % Soft Rock = 1.25 – 5.0 MPa
    elseif ucs(i) <= 12.5 && ucs(i) > 5
        hardness(i) = 4;                % Moderately Soft Rock: 5 – 12.5 MPa
    elseif ucs(i) <= 50 && ucs(i) > 12.5
        hardness(i) = 5;                % Moderately Hard Rock: 12.5 - 50 MPa
    elseif ucs(i) <= 100 && ucs(i) > 50
        hardness(i) = 6;                % Hard Rock: 50-100 MPa
    elseif ucs(i) <= 250 && ucs(i) > 100
        hardness(i) = 7;                % Very Hard Rock: 100 - 250 MPa
    elseif ucs(i) < 1000 && ucs(i) > 250
        hardness(i) = 8;                % Extremely Hard Rock > 250 MPa
    else
        hardness(i) = NaN;
    end
end
figure(23)
subplot(1,2,1)
scatter(ucs, mdrkb, 20, hardness, 'filled')
title('Hardness Rock')
xlabel('UCS [MPa]')
ylabel('Depth [mMD]')
set(gca, 'YDir','reverse')
colorbar
hard = colorbar;
ylabel(hard, 'Hardness Category');
ylim([0 TD+100])
colorbar('Ticks',[1,2,3,4,5,6,7,8], 'TickLabels', {'','','','','','','',''})
colorbar off
grid on
box on

subplot(1,2,2)
scatter(mse, mdrkb, 20, hardness, 'filled')
title('Hardness Rock')
xlabel('MSE [MPa]')
set(gca, 'YDir','reverse')
```

```
colorbar
hard = colorbar;
ylabel(hard, 'Hardness Category');
ylim([0 TD+100])
xlim([0 2000])
colorbar('Ticks',[1,2,3,4,5,6,7,8], 'TickLabels',{'Soil','Very Soft Rock','Soft Rock','Moderately Soft Rock','Moderately
Hard Rock', 'Hard Rock', 'Very Hard Rock', 'Extremely Hard Rock'})
grid on
box on

nameofplot = 'Hardness_UCS_MSE_';
fig1 = figure(23);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

## Detect Drilling Events and NPT

*Create Cut-offs*

```
cut_off = 150; % Cut-off value for bitballing, bit wear and Vibrations [MPa]
ucs_cutoff_1 = 100; % Cut-off value for Very Hard Rock [MPa]
ineff_mse_ds = 1.5; % Cut off values for inefficient drilling mse/ds ratio

a = ones(length(mdrkb), 1); % Array of ones the length of the well
mse_cutoff = cut_off .* a; % Array of MSE cut-off [MPa]
ucs_cutoff = ucs_cutoff_1.*a; % Array of UCS cut-off
ineff_mse_ds_cutoff = ineff_mse_ds.*a; % Array of mse/ds cut-off
```

*Plot MSE corrected with cut-off values, DS and MSE/DS*

```
figure(9)
subplot(1,3,1)
plot(mse_corr, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
plot(ucs, mdrkb, 'linewidt',1)
plot(ucs_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
legend('MSE corrected', 'Cut-Off MSE', 'UCS', 'Cut-Off UCS')
xlim([0 1500])
grid on
box on

subplot(1,3,2)
plot(ds, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('DS')
xlabel('DS [MPa]')
xlim([0 1500])
legend('DS', 'Cut-Off')
grid on
box on

subplot(1,3,3)
plot(mse_ds, mdrkb)
hold on
plot(ineff_mse_ds_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE/DS')
xlabel('MSE/DS')
legend('DS', 'Cut-Off')
xlim([0, 10])
grid on
box on

nameofplot = 'Cutoff_values_mse_ds_ucs_';
fig1 = figure(9);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
```

```
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Detecting Bit Balling, Vibration, Wear or UCS*

```
b = zeros(length(mdrkb), 1); % Array of zeros
b0 = b - 1; % Not an event
b1 = b+1; % Bit balling
b2 = b + 2; % Vibration
b3 = b + 3; % Wear
b4 = b + 4; % Hard Rock
lim_ucs = 1000.*a; % Limit UCS array

k = 0;
    for i =1:length(mdrkb);
        k = k+1;
        ucs_event(k) = b0(i);
        bitballing (k) = b0 (i);
        wear(k) = b0(i);
        vibration (k) = b0(i);
    end

k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if ucs(i) >= ucs_cutoff(i) && ucs(i) <= lim_ucs(i)
        ucs_event(k) = b4(i);
    end
end

k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if mse_corr(k) >= mse_cutoff(k) && ds(k) >= mse_cutoff(k) && mse_ds(k) <= ineff_mse_ds_cutoff(k) && claystone(k) ==
1
            bitballing (k) = b1 (i);
    end
end

k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if mse_corr(k) >= mse_cutoff(k) && ds(k) >= mse_cutoff(k) && mse_ds(k) <= ineff_mse_ds_cutoff(k) && bitballing(k) ==
-1 %claystone(k) ~= 1
            wear (k) = b3 (i);
    end
end

k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if mse_corr(k) >= mse_cutoff(k) && ds(k) >= mse_cutoff(k) && mse_ds(k) > ineff_mse_ds_cutoff(k)
            vibration (k) = b2 (i);
    end
end
```

*Plot MSE Corrected, DS, MSE/DS and Drilling Events*

```
figure(10)
subplot(1,4,1)
plot(mse_corr, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
plot(ucs, mdrkb, 'linewidt',1)
plot(ucs_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
legend('MSE corrected', 'Cut-Off MSE', 'UCS', 'UCS Cut-Off')
xlim([0 1500])
ylim([0, TD+200])
grid on
box on

subplot(1,4,2)
plot(ds, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('DS')
xlabel('DS [Mpa]')
xlim([0 1500])
legend('DS', 'Cut-Off')
ylim([0, TD+200])
```

```
grid on
box on

subplot(1,4,3)
plot(mse_ds, mdrkb)
hold on
plot(ineff_mse_ds_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE/DS')
xlabel('MSE/DS')
legend('DS', 'Cut-off Value')
xlim([0, 10])
ylim([0, TD+200])
grid on
box on

subplot(1,4,4)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
grid on
box on
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])

nameofplot = 'Drilling events & NPT detection_';
fig1 = figure(10);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Plot Drilling Events and NPT Detection and Drilling Parameters*

```
figure(11) % Plot Drilling events & NPT, ROP, WOB, RPM, Torque, Mud flow in, SPPA and ECD, Bit-size
subplot(1,9,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 6])
grid on
box on

subplot(1,9,2)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylim([0, TD+200])
grid on
box on


subplot(1,9,3)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,4)
plot(rpmb, mdrkb)
set(gca, 'YDir','reverse')
title('RPM')
```

```
xlabel('RPM')
ylim([0, TD+200])
grid on
box on

subplot(1,9,5)
plot(tqa, mdrkb)
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,6)
plot(mfia, mdrkb)
set(gca, 'YDir','reverse')
title('Mud Flow In')
xlabel('Flow [liter/min]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,7)
plot(sppa, mdrkb)
set(gca, 'YDir','reverse')
title('SPPA')
xlabel('SPPA [bar]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,8)
plot(ecd, mdrkb)
set(gca, 'YDir','reverse')
title('ECD')
xlabel('ECD [s.g.]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,9)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [inch]')
xlim([0 40])
ylim([0, TD+200])
grid on
box on

nameofplot = 'Detection NPT & events and drillingparameters_';
fig1 = figure(11);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

## Investigate NPT and drilling events for each hole section

*Create new variables for each hole section*

```
% For each possible hole section diameter, new create variables for wob, rop,
% mdrkb, torque, rpm and bit-size. If hole section is not present, set
% equal to NaN.

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 6
        wob_6(k) = wob(i);
        rop_6(k) = rop(i) ;
        mdrkb_6 (k) = mdrkb(i);
        tqa_6 (k) = tqa(i);
        rpmb_6(k) = rpmb(i);
        bitsize_6(k) = 6;
    else
        wob_6(k) = NaN;
        rop_6(k) = NaN ;
        mdrkb_6 (k) = NaN;
        tqa_6 (k) = NaN;
        rpmb_6(k) = NaN;
        bitsize_6(k) = 0;
    end
```

```
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 8.5
        wob_8_5(k) = wob(i);
        rop_8_5(k) = rop(i) ;
        mdrkb_8_5 (k) = mdrkb(i);
        tqa_8_5 (k) = tqa(i);
        rpmb_8_5(k) = rpmb(i);
        bitsize_8_5(k) = 8.5;
    else
        wob_8_5(k) = NaN;
        rop_8_5(k) = NaN ;
        mdrkb_8_5 (k) = NaN;
        tqa_8_5 (k) = NaN;
        rpmb_8_5(k) = NaN;
        bitsize_8_5(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 9.88
        wob_9_88(k) = wob(i);
        rop_9_88(k) = rop(i) ;
        mdrkb_9_88 (k) = mdrkb(i);
        tqa_9_88 (k) = tqa(i);
        rpmb_9_88(k) = rpmb(i);
        bitsize_9_88(k) = 9.88;
    else
        wob_9_88(k) = NaN;
        rop_9_88(k) = NaN ;
        mdrkb_9_88 (k) = NaN;
        tqa_9_88 (k) = NaN;
        rpmb_9_88(k) = NaN;
        bitsize_9_88(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 12.25
        wob_12_25(k) = wob(i);
        rop_12_25(k) = rop(i) ;
        mdrkb_12_25 (k) = mdrkb(i);
        tqa_12_25 (k) = tqa(i);
        rpmb_12_25(k) = rpmb(i);
        bitsize_12_25(k) = 12.25;
    else
        wob_12_25(k) = NaN;
        rop_12_25(k) = NaN ;
        mdrkb_12_25 (k) = NaN;
        tqa_12_25 (k) = NaN;
        rpmb_12_25(k) = NaN;
        bitsize_12_25(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 17.5
        wob_17_5(k) = wob(i);
        rop_17_5(k) = rop(i) ;
        mdrkb_17_5 (k) = mdrkb(i);
        tqa_17_5 (k) = tqa(i);
        rpmb_17_5(k) = rpmb(i);
        bitsize_17_5(k) = 17.5;
    else
        wob_17_5(k) = NaN;
        rop_17_5(k) = NaN ;
        mdrkb_17_5 (k) = NaN;
        tqa_17_5 (k) = NaN;
        rpmb_17_5(k) = NaN;
        bitsize_17_5(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 26
        wob_26(k) = wob(i);
        rop_26(k) = rop(i) ;
```

```
        mdrkb_26 (k) = mdrkb(i);
        tqa_26 (k) = tqa(i);
        rpmb_26(k) = rpmb(i);
        bitsize_26(k) = 26;
    else
        wob_26(k) = NaN;
        rop_26(k) = NaN ;
        mdrkb_26 (k) = NaN;
        tqa_26 (k) = NaN;
        rpmb_26(k) = NaN;
        bitsize_26(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 36
        wob_36(k) = wob(i);
        rop_36(k) = rop(i) ;
        mdrkb_36 (k) = mdrkb(i);
        tqa_36 (k) = tqa(i);
        rpmb_36(k) = rpmb(i);
        bitsize_36(k) = 36;
    else
        wob_36(k) = NaN;
        rop_36(k) = NaN ;
        mdrkb_36 (k) = NaN;
        tqa_36 (k) = NaN;
        rpmb_36(k) = NaN;
        bitsize_36(k) = 0;
    end
end
```

*Plot Drilling events & NPT with WOB-ROP Crossplot for Each Bit Size*

```
% Find out which bit-sizes are in use in the drilling data excel file
bits=0;
disp('The bit-sizes used for this well are:')
if sum(bitsize_9_88)>0
    disp('9 7/8"')
    bits=bits+1;
end
if sum(bitsize_36)>0
    disp('36"')
    bits=bits+1;
end
if sum(bitsize_26)>0
    disp('26"')
    bits=bits+1;
end
if sum(bitsize_17_5)>0
    disp('17 1/2"')
    bits=bits+1;
end
if sum(bitsize_12_25)>0
    disp('12 1/4"')
    bits=bits+1;
end
if sum(bitsize_8_5)>0
    disp('8 1/2"')
    bits=bits+1;
end
if sum(bitsize_6)>0
    disp('6"')
    bits=bits+1;
end

disp(['The drilling data contains ' num2str(bits) ' different bit-sizes'])
rop_lim = 120; % Limit on y-axis for ROP. Default 120 m/hr.
wob_lim = 25; % Limit on x-axis for WOB. Default 25 ton.
% Plot drilling events and NPT and WOB/ROP plot for each hole section
figure(12)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
```

```
grid on
box on


p=1; % Subplot number
if sum(bitsize_9_88)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_9_88, rop_9_88, 20, mdrkb, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_36)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_36, rop_36, 20, mdrkb, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_26)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_26, rop_26, 20, mdrkb, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_17_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_17_5, rop_17_5, 20, mdrkb, 'filled')
    title('17.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_12_25)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_12_25, rop_12_25, 20, mdrkb, 'filled')
    title('12.25" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
```

```
        grid on
        box on
        if bits + 1 ~= p
            colorbar off
        end
    end

    if sum(bitsize_8_5)>0
        p=p+1;
        subplot(1,(bits+1),p)
        scatter(wob_8_5, rop_8_5, 20, mdrkb, 'filled')
        title('8.5" Section')
        xlabel('WOB [ton]')
        ylabel('ROP [m/hr]')
        colorbar
        depth = colorbar;
        ylabel(depth, 'Depth [mMD]');
        set( depth, 'YDir', 'reverse' );
        axis([0 wob_lim 0 rop_lim])
        grid on
        box on
        if bits + 1 ~= p
            colorbar off
        end
    end

    if sum(bitsize_6)>0
        p=p+1;
        subplot(1,(bits+1),p)
        scatter(wob_6, rop_6, 20, mdrkb, 'filled')
        title('6" Section')
        xlabel('WOB [ton]')
        ylabel('ROP [m/hr]')
        colorbar
        depth = colorbar;
        ylabel(depth, 'Depth [mMD]');
        set( depth, 'YDir', 'reverse' );
        axis([0 wob_lim 0 rop_lim])
        grid on
        box on
        if bits + 1 ~= p
            colorbar off
        end
    end

    nameofplot = 'wob_rop_depth_bitsize_events_NPT_';
    fig1 = figure(12);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);

    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Look at each hole section separately

6" Hole Section

```
    % If 6" hole section exist, calculate efficient bit line and plot
    if sum(bitsize_6)>0
        k = 0;
        for i = 1:length(mdrkb)
            k = k+1;
            if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
                efficient_rop_6(k) = rop_6(i);
                efficient_wob_6(k) = wob_6(i);
                eff_rop_6 (k) = rop_6(i);
                eff_wob_6 (k) = wob_6(i);
            else
                ineff_rop_6(k) = rop_6(i);
                ineff_wob_6(k) = wob_6(i);
            end
        end

        efficient_rop_6(isnan(efficient_rop_6)) = []; % Remove NaN values
        efficient_wob_6(isnan(efficient_wob_6)) = []; % Remove NaN values
        c_6 = polyfit(efficient_wob_6, efficient_rop_6, 1); % Create trendline
        disp(['ROP = ' num2str(c_6(1)) '*WOB + ' num2str(c_6(2))]) % % Display equation ROP = a*WOB + b

        x= linspace(0,wob_lim); % WOB
        y_6 = c_6(1)*x + c_6(2); % Trendline

        figure(133)
        subplot(1,3,1)
```

```
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 6" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_6), max(mdrkb_6)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_6, rop_6, 20, mdrkb_6, 'filled')
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_6 = plot(x,y_6, '--r','linewidth', 2);
    legend(h_6, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_6, eff_rop_6, 20, 'filled')
    hold on
    scatter(ineff_wob_6, ineff_rop_6, 20, 'filled')
    plot(x,y_6, '--r','linewidth', 2);
    hold off
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on

    nameofplot = 'wob_rop_depth_bitsize_6_events_NPT';
    fig1 = figure(133);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 6" hole section do not exist');
end
```

*8 1/2" Hole Section*

```
% If 8 1/2" hole section exist, calculate efficient bit line and plot
if sum(bitsize_8_5)>0
    k = 0;
    for i = 1:length(mdrkb)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
       efficient_rop_8_5(k) = rop_8_5(i);
       efficient_wob_8_5(k) = wob_8_5(i);
       eff_rop_8_5 (k) = rop_8_5(i);
       eff_wob_8_5 (k) = wob_8_5(i);
    else
        ineff_rop_8_5(k) = rop_8_5(i);
        ineff_wob_8_5(k) = wob_8_5(i);
    end
    end

    efficient_rop_8_5(isnan(efficient_rop_8_5)) = []; % Remove NaN values
    efficient_wob_8_5(isnan(efficient_wob_8_5)) = []; % Remove NaN values
    c_8_5 = polyfit(efficient_wob_8_5, efficient_rop_8_5, 1); % Create trendline
    disp(['ROP = ' num2str(c_8_5(1)) 'WOB + ' num2str(c_8_5(2))]) % % Display equation ROP = a*WOB + b

    x= linspace(0,wob_lim); % WOB
    y_8_5 = c_8_5(1)*x + c_8_5(2); % Trendline

    figure(13)
```

```
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 8.5" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_8_5), max(mdrkb_8_5)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_8_5, rop_8_5, 20, mdrkb_8_5, 'filled')
    title('8.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_8_5 = plot(x,y_8_5, '--r','linewidth', 2);
    legend(h_8_5, 'Efficient Bit Line')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_8_5, eff_rop_8_5, 20, 'filled')
    hold on
    scatter(ineff_wob_8_5, ineff_rop_8_5, 20, 'filled')
    plot(x,y_8_5, '--r','linewidth', 2);
    hold off
    title('8.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on

    nameofplot = 'wob_rop_depth_bitsize_8_5_events_NPT';
    fig1 = figure(13);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 8 1/2" hole section do not exist');
end
```

*12 1/4" Hole Section*

```
% If 12 1/4" hole section exist, calculate efficient bit line and plot
if sum(bitsize_12_25)>0
    k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
           efficient_rop_12_25(k) = rop_12_25(i);
           efficient_wob_12_25(k) = wob_12_25(i);
           eff_rop_12_25 (k) = rop_12_25(i);
           eff_wob_12_25 (k) = wob_12_25(i);
        else
            ineff_rop_12_25(k) = rop_12_25(i);
            ineff_wob_12_25(k) = wob_12_25(i);
        end
    end

    efficient_rop_12_25(isnan(efficient_rop_12_25)) = []; % Remove NaN values
    efficient_wob_12_25(isnan(efficient_wob_12_25)) = []; % Remove NaN values
    c_12_25 = polyfit(efficient_wob_12_25, efficient_rop_12_25, 1); % Create trendline
    disp(['ROP = ' num2str(c_12_25(1)) '*WOB + ' num2str(c_12_25(2))]) % % Display equation ROP = a*WOB + b

    x= linspace(0,wob_lim); % WOB
    y_12_25 = c_12_25(1)*x + c_12_25(2); % Trendline
```

```
    figure(14)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 12.25" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_12_25), max(mdrkb_12_25)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_12_25, rop_12_25, 20, mdrkb_12_25, 'filled')
    title('12.25" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_12_25 = plot(x,y_12_25, '--r','linewidth', 2);
    legend(h_12_25, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_12_25, eff_rop_12_25, 20, 'filled')
    hold on
    scatter(ineff_wob_12_25, ineff_rop_12_25, 20, 'filled')
    plot(x,y_12_25, '--r','linewidth', 2);
    hold off
    title('12.25" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on

    nameofplot = 'wob_rop_depth_bitsize_12_25_events_NPT';
    fig1 = figure(14);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 12 1/4" hole section do not exist');
end
```

*17 1/2" Hole Section*

```
% If 17 1/2" hole section exist, calculate efficient bit line and plot
if sum(bitsize_17_5)>0
    k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
           efficient_rop_17_5(k) = rop_17_5(i);
           efficient_wob_17_5(k) = wob_17_5(i);
           eff_rop_17_5 (k) = rop_17_5(i);
           eff_wob_17_5 (k) = wob_17_5(i);
        else
            ineff_rop_17_5(k) = rop_17_5(i);
            ineff_wob_17_5(k) = wob_17_5(i);
        end
    end

    efficient_rop_17_5(isnan(efficient_rop_17_5)) = []; % Remove NaN values
    efficient_wob_17_5(isnan(efficient_wob_17_5)) = []; % Remove NaN values
    c_17_5 = polyfit(efficient_wob_17_5, efficient_rop_17_5, 1); % Create trendline
    disp(['ROP = ' num2str(c_17_5(1)) '*WOB + ' num2str(c_17_5(2))]) % % Display equation in the form y = a*x + b

    x= linspace(0,25); % WOB
    y_17_5 = c_17_5(1)*x + c_17_5(2); % Trendline
```

```
    figure(15)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 17.5" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_17_5), max(mdrkb_17_5)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_17_5, rop_17_5, 20, mdrkb_17_5, 'filled')
    title('17.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_17_5 = plot(x,y_17_5, '--r','linewidth', 2);
    legend(h_17_5, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_17_5, eff_rop_17_5, 20, 'filled')
    hold on
    scatter(ineff_wob_17_5, ineff_rop_17_5, 20, 'filled')
    plot(x,y_17_5, '--r','linewidth', 2);
    hold off
    title('17.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on

    nameofplot = 'wob_rop_depth_bitsize_17_5_events_NPT_';
    fig1 = figure(15);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 17 1/2" hole section do not exist');
end
```

*26" Hole Section*

```
% If 26" hole section exist, calculate efficient bit line and plot
if sum(bitsize_26)>0;
    k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
           efficient_rop_26(k) = rop_26(i);
           efficient_wob_26(k) = wob_26(i);
           eff_rop_26 (k) = rop_26(i);
           eff_wob_26 (k) = wob_26(i);
        else
            ineff_rop_26(k) = rop_26(i);
            ineff_wob_26(k) = wob_26(i);
        end
    end

    efficient_rop_26(isnan(efficient_rop_26)) = []; % Remove NaN values
    efficient_wob_26(isnan(efficient_wob_26)) = []; % Remove NaN values
    c_26 = polyfit(efficient_wob_26, efficient_rop_26, 1); % Create trendline
    disp(['ROP = ' num2str(c_26(1)) '*WOB + ' num2str(c_26(2))]) % % Display equation ROP=a*WOB + b
```

```
    x= linspace(0,25); % WOB
    y_26 = c_26(1)*x + c_26(2); % Trendline

    figure(16)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 26" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_26) max(mdrkb_26)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_26, rop_26, 20, mdrkb_26, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_26 = plot(x,y_26, '--r','linewidth', 2);
    legend(h_26, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_26, eff_rop_26, 20, 'filled')
    hold on
    scatter(ineff_wob_26, ineff_rop_26, 20, 'filled')
    plot(x,y_26, '--r','linewidth', 2);
    hold off
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on

    nameofplot = 'wob_rop_depth_bitsize_26_events_NPT_';
    fig1 = figure(16);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 26" hole section do not exist');
end
%
```

36" Hole Section

```
% If 36" hole section exist, calculate efficient bit line and plot
if sum(bitsize_36)>0;
    k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
           efficient_rop_36(k) = rop_36(i);
           efficient_wob_36(k) = wob_36(i);
           eff_rop_36 (k) = rop_36(i);
           eff_wob_36 (k) = wob_36(i);
        else
            ineff_rop_36(k) = rop_36(i);
            ineff_wob_36(k) = wob_36(i);
        end
    end

    efficient_rop_36(isnan(efficient_rop_36)) = []; % Remove NaN values
    efficient_wob_36(isnan(efficient_wob_36)) = []; % Remove NaN values
    c_36 = polyfit(efficient_wob_36, efficient_rop_36, 1); % Create trendline
```

```
    disp(['ROP = ' num2str(c_36(1)) '*WOB + ' num2str(c_36(2))]) % % Display equation ROP = a*WOB + b

    x= linspace(0,25); % WOB
    y_36 = c_36(1)*x + c_36(2); % Trendline

    figure(16)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 36" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_36) max(mdrkb_36)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_36, rop_36, 20, mdrkb_36, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_36 = plot(x,y_36, '--r','linewidth', 2);
    legend(h_36, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_36, eff_rop_36, 20, 'filled')
    hold on
    scatter(ineff_wob_36, ineff_rop_36, 20, 'filled')
    plot(x,y_36, '--r','linewidth', 2);
    hold off
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on

    nameofplot = 'wob_rop_depth_bitsize_36_events_NPT';
    fig1 = figure(99);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);

    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 36" hole section do not exist');
end
```

9 7/8" Section

```
% If 9 7/8" hole section exist, calculate efficient bit line and plot
if sum(bitsize_9_88)>0;
    k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
           efficient_rop_9_88(k) = rop_9_88(i);
           efficient_wob_9_88(k) = wob_9_88(i);
           eff_rop_9_88 (k) = rop_9_88(i);
           eff_wob_9_88 (k) = wob_9_88(i);
        else
            ineff_rop_9_88(k) = rop_9_88(i);
            ineff_wob_9_88(k) = wob_9_88(i);
        end
    end

    efficient_rop_9_88(isnan(efficient_rop_9_88)) = []; % Remove NaN values
```

```
    efficient_wob_9_88(isnan(efficient_wob_9_88)) = []; % Remove NaN values
    c_9_88 = polyfit(efficient_wob_9_88, efficient_rop_9_88, 1); % Create trendline
    disp(['ROP = ' num2str(c_9_88(1)) '*WOB + ' num2str(c_9_88(2))]) % % Display equation ROP = a*WOB + b

    x= linspace(0,25); % WOB
    y_9_88 = c_9_88(1)*x + c_9_88(2); % Trendline

    figure(16)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 9 7/8" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_9_88) max(mdrkb_9_88)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_9_88, rop_9_88, 20, mdrkb_9_88, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_9_88 = plot(x,y_9_88, '--r','linewidth', 2);
    legend(h_9_88, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_9_88, eff_rop_9_88, 20, 'filled')
    hold on
    scatter(ineff_wob_9_88, ineff_rop_9_88, 20, 'filled')
    plot(x,y_9_88, '--r','linewidth', 2);
    hold off
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on

    nameofplot = 'wob_rop_depth_bitsize_9_88_events_NPT';
    fig1 = figure(999);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 9 7/8" hole section do not exist');
end
```

Plot hole sections together

```
% Plot drilling events and NPT and WOB/ROP plot for each hole section in
% one plot including efficient bit line
figure(8876)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
```

```
xlim([0 5])
grid on
box on

p=1; % Subplot number
if sum(bitsize_9_88)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_9_88, rop_9_88, 20, mdrkb, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_9_88 = plot(x,y_9_88, '--r','linewidth', 2);
    %legend(h_9_88, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_36)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_36, rop_36, 20, mdrkb, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_36 = plot(x,y_36, '--r','linewidth', 2);
    %legend(h_36, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_26)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_26, rop_26, 20, mdrkb, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_26 = plot(x,y_26, '--r','linewidth', 2);
    %legend(h_26, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_17_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_17_5, rop_17_5, 20, mdrkb, 'filled')
    title('17 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_17_5 = plot(x,y_17_5, '--r','linewidth', 2);
```

```
    %legend(h_17_5, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_12_25)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_12_25, rop_12_25, 20, mdrkb, 'filled')
    title('12 1/4" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_12_25 = plot(x,y_12_25, '--r','linewidth', 2);
    %legend(h_12_25, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_8_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_8_5, rop_8_5, 20, mdrkb, 'filled')
    title('8 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_8_5 = plot(x,y_8_5, '--r','linewidth', 2);
    %legend(h_8_5, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_6)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_6, rop_6, 20, mdrkb, 'filled')
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_6 = plot(x,y_6, '--r','linewidth', 2);
    %legend(h_6, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

nameofplot = 'eff_wob_rop_depth_bitsize_events_NPT_';
fig1 = figure(8876);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

```
% Plot drilling events and NPT and WOB/ROP plot for each hole section in
% one plot including efficient bit line.
figure(54363)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

p=1; % Subplot number
if sum(bitsize_9_88)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_9_88, eff_rop_9_88, 20, 'filled')
    hold on
    scatter(ineff_wob_9_88, ineff_rop_9_88, 20, 'filled')
    plot(x,y_9_88, '--r','linewidth', 2);
    hold off
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    %legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_36)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_36, eff_rop_36, 20, 'filled')
    hold on
    scatter(ineff_wob_36, ineff_rop_36, 20, 'filled')
    plot(x,y_36, '--r','linewidth', 2);
    hold off
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    %legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_26)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_26, eff_rop_26, 20, 'filled')
    hold on
    scatter(ineff_wob_26, ineff_rop_26, 20, 'filled')
    plot(x,y_26, '--r','linewidth', 2);
    hold off
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    %legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_17_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_17_5, eff_rop_17_5, 20, 'filled')
    hold on
    scatter(ineff_wob_17_5, ineff_rop_17_5, 20, 'filled')
    plot(x,y_17_5, '--r','linewidth', 2);
```

```
    hold off
    title('17 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    %legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_12_25)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_12_25, eff_rop_12_25, 20, 'filled')
    hold on
    scatter(ineff_wob_12_25, ineff_rop_12_25, 20, 'filled')
    plot(x,y_12_25, '--r','linewidth', 2);
    hold off
    title('12 1/4" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    %legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_8_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_8_5, eff_rop_8_5, 20, 'filled')
    hold on
    scatter(ineff_wob_8_5, ineff_rop_8_5, 20, 'filled')
    plot(x,y_8_5, '--r','linewidth', 2);
    hold off
    title('8 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    %legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_6)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_6, eff_rop_6, 20, 'filled')
    hold on
    scatter(ineff_wob_6, ineff_rop_6, 20, 'filled')
    plot(x,y_6, '--r','linewidth', 2);
    hold off
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    %legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

nameofplot = 'eff_noneff_wob_rop_depth_bitsize_events_NPT_';
fig1 = figure(54363);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Calculate Lost Time

6" Hole Section

```
% If 6" hole section exist, calculate new ROP from efficient bit-line
% based on drilling events.

if sum(bitsize_6)>0
    figure(266)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 6" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_6), max(mdrkb_6)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    % Calculate new ROP from efficient bit-line for 6" hole section
    k = 0;
    for i = 1:length(mdrkb_6)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            rop_new_6(k) = rop_6(i); % No events flagged, use original ROP
        else
            rop_new_6(k) = c_6(1)*wob_6(i) + c_6(2); % Flagged event, calculate ROP from Efficient bit-line
        end
    end

    subplot(1,3,2)
    plot(rop_new_6, mdrkb)
    hold on
    plot(rop_6, mdrkb)
    hold off
    set(gca, 'YDir','reverse')
    title('6" Section New ROP')
    xlabel('ROP [m/hr]')
    legend('New ROP', 'Original ROP')
    xlim([0 rop_lim])
    ylim([min(mdrkb_6), max(mdrkb_6)])
    ylabel('Depth [mMD]')
    box on
    grid on

    subplot(1,3,3)
    scatter(wob_6, rop_6, 20, mdrkb_6, 'filled')
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_6 = plot(x,y_6, '--r','linewidth', 2);
    legend(h_6, 'Efficient Bit')
    hold off
    grid on
    box on

    nameofplot = 'New_ROP_6_';
    fig1 = figure(266);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

     % Calculate which ROP change is due to which drilling event
     k=0;
     for i = 1:length(mdrkb_6)
        k = k+1;
        if bitballing(k) == 1
            rop_bb_new_6(k) = c_6(1)*wob_6(i) + c_6(2);
        else
            rop_bb_new_6(k) = rop_6(i);
        end
        if vibration(k) == 2
            rop_vibration_new_6(k) = c_6(1)*wob_6(i) + c_6(2);
```

```
        else
            rop_vibration_new_6(k) = rop_6(i);
        end
        if wear(k) == 3
            rop_wear_new_6(k) = c_6(1)*wob_6(i) + c_6(2);
        else
            rop_wear_new_6(k) = rop_6(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_6(k) = c_6(1)*wob_6(i) + c_6(2);
        else
            rop_ucs_new_6(k) = rop_6(i);
        end
     end
else
    disp('The 6" hole section do not exist');
end
```

8 1/2" Hole Section

```
% If 8 1/2" hole section exist, calculate new ROP from efficient bit-line
% based on drilling events.

if sum(bitsize_8_5)>0
    figure(26)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 8.5" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_8_5), max(mdrkb_8_5)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    % Calculate new ROP from efficient bit-line for 8.5" hole section
    k = 0;
    for i = 1:length(mdrkb_8_5)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            rop_new_8_5(k) = rop_8_5(i); % No events flagged, use original ROP
        else
            rop_new_8_5(k) = c_8_5(1)*wob_8_5(i) + c_8_5(2); % Flagged event, calculate ROP from Efficient bit-line
        end
    end

    subplot(1,3,2)
    plot(rop_new_8_5, mdrkb)
    hold on
    plot(rop_8_5, mdrkb)
    hold off
    set(gca, 'YDir','reverse')
    title('8.5" Section New ROP')
    xlabel('ROP [m/hr]')
    legend('New ROP', 'Original ROP')
    xlim([0 rop_lim])
    ylim([min(mdrkb_8_5), max(mdrkb_8_5)])
    ylabel('Depth [mMD]')
    box on
    grid on

    subplot(1,3,3)
    scatter(wob_8_5, rop_8_5, 20, mdrkb_8_5, 'filled')
    title('8.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_8_5 = plot(x,y_8_5, '--r','linewidth', 2);
    legend(h_8_5, 'Efficient Bit')
    hold off
    grid on
    box on

    nameofplot = 'New_ROP_8_5_';
    fig1 = figure(26);
```

```
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);


    k=0;
     for i = 1:length(mdrkb_8_5)
        k = k+1;
        if bitballing(k) == 1
            rop_bb_new_8_5(k) = c_8_5(1)*wob_8_5(i) + c_8_5(2);
        else
            rop_bb_new_8_5(k) = rop_8_5(i);
        end
        if vibration(k) == 2
            rop_vibration_new_8_5(k) = c_8_5(1)*wob_8_5(i) + c_8_5(2);
        else
            rop_vibration_new_8_5(k) = rop_8_5(i);
        end
        if wear(k) == 3
            rop_wear_new_8_5(k) = c_8_5(1)*wob_8_5(i) + c_8_5(2);
        else
            rop_wear_new_8_5(k) = rop_8_5(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_8_5(k) = c_8_5(1)*wob_8_5(i) + c_8_5(2);
        else
            rop_ucs_new_8_5(k) = rop_8_5(i);
        end
     end
else
    disp('The 8 1/2" hole section do not exist');
end
```

12 1/4" Hole section

```
% If 12 1/4" hole section exist, calculate new ROP from efficient bit-line
% based on drilling events.

if sum(bitsize_12_25)>0
    figure(28)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 12.25" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_12_25), max(mdrkb_12_25)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    % Calculate new ROP from efficient bit-line for 12.25" hole section
    k = 0;
    for i = 1:length(mdrkb_12_25)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            rop_new_12_25(k) = rop_12_25(i); % No events flagged, use original ROP
        else
            rop_new_12_25(k) = c_12_25(1)*wob_12_25(i) + c_12_25(2); % Flagged event, calculate ROP from Efficient bit-
line
        end
    end

    subplot(1,3,2)
    plot(rop_new_12_25, mdrkb)
    hold on
    plot(rop_12_25, mdrkb)
    hold off
    set(gca, 'YDir','reverse')
    title('12.25" Section New ROP')
    xlabel('ROP [m/hr]')
    ylim([min(mdrkb_12_25), max(mdrkb_12_25)])
    ylabel('Depth [mMD]')
    legend('New ROP', 'Original ROP')
    grid on
    box on
```

```
    subplot(1,3,3)
    scatter(wob_12_25, rop_12_25, 20, mdrkb_12_25, 'filled')
    title('12.25" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_12_25 = plot(x,y_12_25, '--r','linewidth', 2);
    legend(h_12_25, 'Efficient Bit')
    hold off
    grid on
    box on

    nameofplot = 'New_ROP_12_25_';
    fig1 = figure(28);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

     % Calculate which ROP change is due to which drilling event
   k=0;
     for i = 1:length(mdrkb_12_25)
        k = k+1;
        if bitballing(k) == 1
            rop_bb_new_12_25(k) = c_12_25(1)*wob_12_25(i) + c_12_25(2);
        else
            rop_bb_new_12_25(k) = rop_12_25(i);
        end
        if vibration(k) == 2
            rop_vibration_new_12_25(k) = c_12_25(1)*wob_12_25(i) + c_12_25(2);
        else
            rop_vibration_new_12_25(k) = rop_12_25(i);
        end
        if wear(k) == 3
            rop_wear_new_12_25(k) = c_12_25(1)*wob_12_25(i) + c_12_25(2);
        else
            rop_wear_new_12_25(k) = rop_12_25(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_12_25(k) = c_12_25(1)*wob_12_25(i) + c_12_25(2);
        else
            rop_ucs_new_12_25(k) = rop_12_25(i);
        end
     end
else
    disp('The 12 1/4" hole section do not exist');
end
```

17 1/2" Hole Section

```
% If 17 1/2" hole section exist, calculate new ROP from efficient bit-line
% based on drilling events.

if sum(bitsize_17_5)>0
    figure(30)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 17.5" Section')
    xlabel('')
    ylabel('Depth [meter MD]')
    ylim([min(mdrkb_17_5), max(mdrkb_17_5)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    % Calculate new ROP from efficient bit-line for 12.25" hole section
    k = 0;
    for i = 1:length(mdrkb_17_5)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            rop_new_17_5(k) = rop_17_5(i); % No events flagged, use original ROP
        else
```

```
            rop_new_17_5(k) = c_17_5(1)*wob_17_5(i) + c_17_5(2); % Flagged event, calculate ROP from Efficient bit-line
        end
    end

    subplot(1,3,2)
    plot(rop_new_17_5, mdrkb)
    hold on
    plot(rop_17_5, mdrkb)
    hold off
    set(gca, 'YDir','reverse')
    title('17.5" Section New ROP')
    xlabel('ROP [m/hr]')
    ylabel('Depth [mMD]')
    legend('New ROP', 'Original ROP')
    ylim([min(mdrkb_17_5), max(mdrkb_17_5)])
    grid on
    box on

    subplot(1,3,3)
    scatter(wob_17_5, rop_17_5, 20, mdrkb_17_5, 'filled')
    title('17.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_17_5 = plot(x,y_17_5, '--r','linewidth', 2);
    legend(h_17_5, 'Efficient Bit')
    hold off
    grid on
    box on

    nameofplot = 'New_ROP_17_5_';
    fig1 = figure(30);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

     % Calculate which ROP change is due to which drilling event
    k=0;
     for i = 1:length(mdrkb_17_5)
        k = k+1;
        if bitballing(k) == 1
            rop_bb_new_17_5(k) = c_17_5(1)*wob_17_5(i) + c_17_5(2);
        else
            rop_bb_new_17_5(k) = rop_17_5(i);
        end
        if vibration(k) == 2
            rop_vibration_new_17_5(k) = c_17_5(1)*wob_17_5(i) + c_17_5(2);
        else
            rop_vibration_new_17_5(k) = rop_17_5(i);
        end
        if wear(k) == 3
            rop_wear_new_17_5(k) = c_17_5(1)*wob_17_5(i) + c_17_5(2);
        else
            rop_wear_new_17_5(k) = rop_17_5(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_17_5(k) = c_17_5(1)*wob_17_5(i) + c_17_5(2);
        else
            rop_ucs_new_17_5(k) = rop_17_5(i);
        end
     end
else
    disp('The 17 1/2" hole section do not exist');
end
```

26" Hole Section

```
% If 26" hole section exist, calculate new ROP from efficient bit-line
% based on drilling events.

if sum(bitsize_26)>0
    figure(32)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
```

```
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 26" Section')
    xlabel('')
    ylabel('Depth [meter MD]')
    ylim([min(mdrkb_26), max(mdrkb_26)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    % Calculate new ROP from efficient bit-line for 26" hole section

    k = 0;
    for i = 1:length(mdrkb_26)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            rop_new_26(k) = rop_26(i); % No events flagged, use original ROP
        else
            rop_new_26(k) = c_26(1)*wob_26(i) + c_26(2); % Flagged event, calculate ROP from Efficient bit-line
        end
    end

    subplot(1,3,2)
    plot(rop_new_26, mdrkb)
    hold on
    plot(rop_26, mdrkb)
    hold off
    set(gca, 'YDir','reverse')
    title('26" Section New ROP')
    xlabel('ROP [m/hr]')
    ylabel('Depth [mMD]')
    legend('New ROP', 'Original ROP')
    ylim([min(mdrkb_26), max(mdrkb_26)])
    grid on
    box on

    subplot(1,3,3)
    scatter(wob_26, rop_26, 20, mdrkb_26, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_26 = plot(x,y_26, '--r','linewidth', 2);
    legend(h_26, 'Efficient Bit')
    hold off
    grid on
    box on

    nameofplot = 'New_ROP_26_';
    fig1 = figure(32);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

     % Calculate which ROP change is due to which drilling event
   k=0;
     for i = 1:length(mdrkb_26)
        k = k+1;
        if bitballing(k) == 1
            rop_bb_new_26(k) = c_26(1)*wob_26(i) + c_26(2);
        else
            rop_bb_new_26(k) = rop_26(i);
        end
        if vibration(k) == 2
            rop_vibration_new_26(k) = c_26(1)*wob_26(i) + c_26(2);
        else
            rop_vibration_new_26(k) = rop_26(i);
        end
        if wear(k) == 3
            rop_wear_new_26(k) = c_26(1)*wob_26(i) + c_26(2);
        else
            rop_wear_new_26(k) = rop_26(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_26(k) = c_26(1)*wob_26(i) + c_26(2);
        else
            rop_ucs_new_26(k) = rop_26(i);
        end
```

```
        end
    else
        disp('The 26" hole section do not exist');
    end
```

36" Hole Section

```
% If 36" hole section exist, calculate new ROP from efficient bit-line
% based on drilling events.

if sum(bitsize_36)>0
    figure(33)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 36" Section')
    xlabel('')
    ylabel('Depth [meter MD]')
    ylim([min(mdrkb_36), max(mdrkb_36)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    % Calculate new ROP from efficient bit-line for 36" hole section

    k = 0;
    for i = 1:length(mdrkb_36)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            rop_new_36(k) = rop_36(i); % No events flagged, use original ROP
        else
            rop_new_36(k) = c_36(1)*wob_36(i) + c_36(2); % Flagged event, calculate ROP from Efficient bit-line
        end
    end

    subplot(1,3,2)
    plot(rop_new_36, mdrkb)
    hold on
    plot(rop_36, mdrkb)
    hold off
    set(gca, 'YDir','reverse')
    title('36" Section New ROP')
    xlabel('ROP [m/hr]')
    ylabel('Depth [mMD]')
    legend('New ROP', 'Original ROP')
    ylim([min(mdrkb_36), max(mdrkb_36)])
    grid on
    box on

    subplot(1,3,3)
    scatter(wob_36, rop_36, 20, mdrkb_36, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_36 = plot(x,y_36, '--r','linewidth', 2);
    legend(h_36, 'Efficient Bit')
    hold off
    grid on
    box on

    nameofplot = 'New_ROP_36_';
    fig1 = figure(33);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

     % Calculate which ROP change is due to which drilling event
    k=0;
     for i = 1:length(mdrkb_36)
        k = k+1;
        if bitballing(k) == 1
```

```
            rop_bb_new_36(k) = c_36(1)*wob_36(i) + c_36(2);
        else
            rop_bb_new_36(k) = rop_36(i);
        end
        if vibration(k) == 2
            rop_vibration_new_36(k) = c_36(1)*wob_36(i) + c_36(2);
        else
            rop_vibration_new_36(k) = rop_36(i);
        end
        if wear(k) == 3
            rop_wear_new_36(k) = c_36(1)*wob_36(i) + c_36(2);
        else
            rop_wear_new_36(k) = rop_36(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_36(k) = c_36(1)*wob_36(i) + c_36(2);
        else
            rop_ucs_new_36(k) = rop_36(i);
        end
     end
else
    disp('The 36" hole section do not exist');
end
```

9 7/8" Hole Section

```
% If 9 7/8" hole section exist, calculate new ROP from efficient bit-line
% based on drilling events.

if sum(bitsize_9_88)>0
    figure(333)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 9 7/8" Section')
    xlabel('')
    ylabel('Depth [meter MD]')
    ylim([min(mdrkb_9_88), max(mdrkb_9_88)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    % Calculate new ROP from efficient bit-line for 9_88" hole section

    k = 0;
    for i = 1:length(mdrkb_9_88)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            rop_new_9_88(k) = rop_9_88(i); % No events flagged, use original ROP
        else
            rop_new_9_88(k) = c_9_88(1)*wob_9_88(i) + c_9_88(2); % Flagged event, calculate ROP from Efficient bit-line
        end
    end

    subplot(1,3,2)
    plot(rop_new_9_88, mdrkb)
    hold on
    plot(rop_9_88, mdrkb)
    hold off
    set(gca, 'YDir','reverse')
    title('9 7/8" Section New ROP')
    xlabel('ROP [m/hr]')
    ylabel('Depth [mMD]')
    legend('New ROP', 'Original ROP')
    ylim([min(mdrkb_9_88), max(mdrkb_9_88)])
    grid on
    box on

    subplot(1,3,3)
    scatter(wob_9_88, rop_9_88, 20, mdrkb_9_88, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_9_88 = plot(x,y_9_88, '--r','linewidth', 2);
    legend(h_9_88, 'Efficient Bit')
```

```
    hold off
    grid on
    box on

    nameofplot = 'New_ROP_9_88_';
    fig1 = figure(333);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

     % Calculate which ROP change is due to which drilling event
    k=0;
     for i = 1:length(mdrkb_9_88)
        k = k+1;
        if bitballing(k) == 1
            rop_bb_new_9_88(k) = c_9_88(1)*wob_9_88(i) + c_9_88(2);
        else
            rop_bb_new_9_88(k) = rop_9_88(i);
        end
        if vibration(k) == 2
            rop_vibration_new_9_88(k) = c_9_88(1)*wob_9_88(i) + c_9_88(2);
        else
            rop_vibration_new_9_88(k) = rop_9_88(i);
        end
        if wear(k) == 3
            rop_wear_new_9_88(k) = c_9_88(1)*wob_9_88(i) + c_9_88(2);
        else
            rop_wear_new_9_88(k) = rop_9_88(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_9_88(k) = c_9_88(1)*wob_9_88(i) + c_9_88(2);
        else
            rop_ucs_new_9_88(k) = rop_9_88(i);
        end
     end
else
    disp('The 9 7/8" hole section do not exist');
end
```

Concatinate ROPs, Calculate new time and plot new ROP

```
% Concatinate the new ROPs
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 9.88
        rop_new(k) = rop_new_9_88(i);
    elseif bdia(k) == 36
        rop_new(k) = rop_new_36(i);
    elseif bdia(k) == 26
        rop_new(k) = rop_new_26(i);
    elseif bdia(k) == 17.5
        rop_new(k) = rop_new_17_5(i);
    elseif bdia(k) == 12.25
        rop_new(k) = rop_new_12_25(i);
    elseif bdia(k) == 8.5
        rop_new(k) = rop_new_8_5(i);
    elseif bdia(k) == 6
        rop_new(k) = rop_new_6(i);
    else
        rop_new(k) = rop(i);
    end
end

figure(34)
plot(rop_new, mdrkb)
hold on
plot(rop, mdrkb)
hold off
set(gca, 'YDir','reverse')
title(' New ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
legend('New ROP', 'Original ROP')
grid on
box on


% Concatinate the new ROPs for each drilling event
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 9.88
```

```
            rop_new_bb(k) = rop_bb_new_9_88(i);
            rop_new_vibration(k) = rop_vibration_new_9_88(i);
            rop_new_wear(k) = rop_wear_new_9_88(i);
            rop_new_ucs(k) = rop_ucs_new_9_88(i);
        elseif bdia(k) == 36
            rop_new_bb(k) = rop_bb_new_36(i);
            rop_new_vibration(k) = rop_vibration_new_36(i);
            rop_new_wear(k) = rop_wear_new_36(i);
            rop_new_ucs(k) = rop_ucs_new_36(i);
        elseif bdia(k) == 26
            rop_new_bb(k) = rop_bb_new_26(i);
            rop_new_vibration(k) = rop_vibration_new_26(i);
            rop_new_wear(k) = rop_wear_new_26(i);
            rop_new_ucs(k) = rop_ucs_new_26(i);
        elseif bdia(k) == 17.5
            rop_new_bb(k) = rop_bb_new_17_5(i);
            rop_new_vibration(k) = rop_vibration_new_17_5(i);
            rop_new_wear(k) = rop_wear_new_17_5(i);
            rop_new_ucs(k) = rop_ucs_new_17_5(i);
        elseif bdia(k) == 12.25
            rop_new_bb(k) = rop_bb_new_12_25(i);
            rop_new_vibration(k) = rop_vibration_new_12_25(i);
            rop_new_wear(k) = rop_wear_new_12_25(i);
            rop_new_ucs(k) = rop_ucs_new_12_25(i);
        elseif bdia(k) == 8.5
            rop_new_bb(k) = rop_bb_new_8_5(i);
            rop_new_vibration(k) = rop_vibration_new_8_5(i);
            rop_new_wear(k) = rop_wear_new_8_5(i);
            rop_new_ucs(k) = rop_ucs_new_8_5(i);
        elseif bdia(k) == 6
            rop_new_bb(k) = rop_bb_new_6(i);
            rop_new_vibration(k) = rop_vibration_new_6(i);
            rop_new_wear(k) = rop_wear_new_6(i);
            rop_new_ucs(k) = rop_ucs_new_6(i);
        else
            rop_new_bb(k) = rop(i);
            rop_new_vibration(k) = rop(i);
            rop_new_wear(k) = rop(i);
            rop_new_ucs(k) = rop(i);
        end
    end

    figure(343334)
    plot(rop_new_bb, mdrkb)
    hold on
    plot(rop_new_vibration, mdrkb)
    plot(rop_new_wear, mdrkb)
    plot(rop_new_ucs, mdrkb)
    plot(rop, mdrkb)
    hold off
    set(gca, 'YDir','reverse')
    title(' New ROP')
    xlabel('ROP [m/hr]')
    ylabel('Depth [mMD]')
    legend('New ROP Bit balling', 'New ROP Vibration','New ROP Wear', 'New ROP Hard Rock', 'Original ROP')
    grid on
    box on
    % Calculate time difference between original and new ROP

    k=0;
    for i=1:length(mdrkb)
        k=k+1;
        time_orig(k)= 1/rop(i); % 1/rop(i);
        time_n(k) = 1/rop_new(i); % 1/rop_new(i);
    end

    time_orig(isinf(time_orig)) = []; % Remove inf values
    time_n(isinf(time_n)) = []; % Remove inf values

    time_original = sum(time_orig)
    time_new = sum(time_n)

    % Calculate time difference between original and new ROP for each drilling
    % event

    k=0;
    for i=1:length(mdrkb)
        k=k+1;
        time_bb(k) = 1/rop_new_bb(i); % 1/rop_new_bb(i);
        time_vibration(k) = 1/rop_new_vibration(i);%1/rop_new_vibration(i);
        time_wear(k) = 1/rop_new_wear(i); %1/rop_new_wear(i);
        time_ucs(k) = 1/rop_new_ucs(i); %1/rop_new_ucs(i);
    end

    time_bb(isinf(time_bb)) = []; % Remove inf values
    time_vibration(isinf(time_vibration)) = []; % Remove inf values
    time_wear(isinf(time_wear)) = []; % Remove inf values
    time_ucs(isinf(time_ucs)) = []; % Remove inf values
```

```
time_bb_1 = sum(time_bb)
time_vibration_1 = sum(time_vibration)
time_wear_1 = sum(time_wear)
time_ucs_1 = sum(time_ucs)
figure(35)
subplot(1,2,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

subplot(1,2,2)
plot(rop_new, mdrkb, 'linewidt', 0.5)
hold on
plot(rop, mdrkb, 'linewidt', 0.5)

num1 = num2str(round(time_original));
num2 = num2str(round(time_new));

   txt1 = ['Original:  '];
   txt2 = ['New: '];
   txt3 = [' hrs'];
   txt4 = [' '];

txt32 = strcat(txt1, txt4, num1, txt4, txt3);
text(110, 570, txt32, 'fontsize', 10,'Interpreter','latex')

txt42 = strcat(txt2, txt4, num2, txt3);
text(110, 770, txt42, 'fontsize', 10,'Interpreter','latex')

hold off

set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylim([0, TD+200])
xlim([0, 150])
legend('New ROP', 'Original ROP')
grid on
box on

nameofplot = 'time_efficient_drilling_';
fig1 = figure(35);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Bit runs and NPT

```
% Give variable names to bit runs
bit_nr = bit_data(:,1);                                 % Bit number
bit_in = bit_data(:,2);                                 % Depth in bit, [mMD]
bit_out = bit_data(:,3);                                % Depth out bit, [mMD]
bit_size = bit_data(:,4);                               % Bit size, [in]
bit_core = bit_data(:,7);                               % Coring or not
total_bits = length(bit_nr);                            % Total number of bits

k=0;
for i = 1:length(bit_nr)
    k=k+1;
    if bit_core(k) == 2
        bit_in_core(k) = bit_in(i);
        bit_out_core(k) = bit_out(i);
    else
        bit_in_core(k) = NaN;
        bit_out_core(k) = NaN;
    end
end

% Plot bit-runs and drilling events/NPT
```

```
figure(36)
subplot(1,3,1)
scatter(bit_nr, bit_in, 30, 'filled')
hold on
scatter(bit_nr, bit_out, 30, 'filled')
scatter(bit_nr, bit_in_core, 30, 'filled')
scatter(bit_nr, bit_out_core, 30, 'filled')
hold off
set(gca, 'YDir','reverse')
xlim([0 length(bit_nr)+1])
title('Bit Runs')
xlabel('Bit number')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bit In', 'Bit Out', 'Core run in', 'Core run Out')
grid on
box on

subplot(1,3,2)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
xlim([0 40])
title('Bit Size')
xlabel('Bit size [in]')
ylim([0, TD+200])
grid on
box on

subplot(1,3,3)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

nameofplot = 'core_bit_drillingevents_NPT_';
fig1 = figure(36);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% Display number of bit runs
disp(['Number of bit-runs: ' num2str(total_bits) ]);
% Calculate time for bit change
trip_speed = 500; % Tripping speed assuming 500 m/hr

% Look at figure 36 and study which bits are changed due to drilling
% events. Relate bit changes to bit size and drilling events plot.

% Manually insert the bit number to calculate cost of bit change. Other bits are entered as zero.
% 34/4-15 S
%change_bit = [0,0,0,0,0,0,0];

% 34/4-15 A
%change_bit=[0,0,0,0,0,0,0,0,0];

% 34/4-16 S
%change_bit=[0,0,0,0,0];

% 33/6-3 S
%change_bit = [0,0,0,0,0,6,7,0];

% 35/3-6
%change_bit = [0,0,0,0,5,6,0,0,0];

%35/8-4
%change_bit = [0,0,0,4,0,0,0,0];

%35/8-5
%change_bit = [0,0,0,0,0,6,0,0,0,0,0,0,0,0,0,0];

%35/9-8
%change_bit = [0,0,0,0,0,0,0,0,0,0,0,0,0];

%35/9-11 S
```

```
% change_bit = [0,0,0,0,0,0];

% 15/12-18 S
% change_bit = [0,0,0,0,0,0,7,0,0];

% 15/12-21
 %change_bit = [0,0,0,0,0,0,0,0,0,0,0];


 % 15/12-23
% change_bit = [0,0,0,0,0,0,0,0];


% Norwegian Sea

% 6406/1-1
%change_bit = [0,0,0,0,5,0,0,0,0,0,0,0,0,0,0];

% 6406/2-3
%change_bit = [0,0,0,0,0,6,0,8,0,0,0,0,0,0,15,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

% 6406/2-5
%change_bit = [0,0,0,0,0];

% 6406/2-7
%change_bit = [0,0,0,0,0];

% 6506/11-6
%change_bit = [0,0,0,0,0,6,0,8,9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

% 6407/8-4 S
%change_bit = [0,0,3,0,0,0,0,0,0,0,0];

% 6407/8-5 S
%change_bit = [0,0,0,0];

% 6407/8-6
%change_bit = [0,0,0,0,0,0,0,8,0];

% 6407/8-7
%change_bit = [0,0,0];

% 6407/9-9
%change_bit = [0,0,0,0];


if length(change_bit) ~= length(bit_nr)
    disp(['Error!'])
    disp(['The change bit vector must contain: ' num2str(total_bits) ' numbers'])
    disp(['Current change bit vector contains: ' num2str(length(change_bit)) ' numbers'])
else
    disp('Correct bit change inputs.')
end

k=0;
for i = 1:length(bit_nr)
    k = k+1;
    if change_bit (k) == bit_nr(k);
        depth_bit_change(k) = bit_out(k);
        depth_bit_change_1(k) = bit_out(k);
    else
        depth_bit_change(k) = NaN;
        depth_bit_change_1(k) = NaN;
    end
end

depth_bit_change_1(isnan(depth_bit_change_1)) = []; % Remove NaN values
disp(['Depth of bit change: ' num2str(depth_bit_change_1) ' mMD'])

k = 0;
for i = 1:length(bit_nr)
    k=k+1;
    if depth_bit_change(k) == bit_out(k)
        trip_time_lost(k) = (bit_out(k)*2)/trip_speed;
    else
        trip_time_lost(k) = 0;
    end
end

k = 0;
for i = 1:length(bit_nr)
    k=k+1;
    total_trip_time_1(k) = (bit_out(i)*2)/trip_speed;
end

disp(['Trip times: ' num2str(round(trip_time_lost,1))])
total_trip_time_lost = sum(trip_time_lost);
disp(['Total tripping time lost: ' num2str(round(total_trip_time_lost,1)) ' hours'])
```

```
total_trip_time = sum(total_trip_time_1);
disp(['Total tripping time: ' num2str(round(total_trip_time,1)) ' hours'])
```

Calculate total lost time due to inefficient drilling and bit changes

```
% Lost time due to inefficient drilling
rop_lost_time = time_original - time_new;
disp(['Lost time due to inefficient drilling: ' num2str(round(rop_lost_time,1)) ' hours'])

% Lost time due to bit change
disp(['Lost time due to bit change: ' num2str(round(total_trip_time_lost,1)) ' hours'])

% Total lost time
total_lost_time = rop_lost_time + total_trip_time_lost;
disp(['Total lost time: ' num2str(round(total_lost_time,0)) ' hours'])
% Lost time due to each drilling event
rop_lost_time_bb = time_original - time_bb_1;
rop_lost_time_vibration = time_original - time_vibration_1;
rop_lost_time_wear = time_original - time_wear_1;
rop_lost_time_ucs = time_original - time_ucs_1;

disp(['Lost time due to Bit Balling: ' num2str(round(rop_lost_time_bb,0)) ' hours'])
disp(['Lost time due to Vibrations: ' num2str(round(rop_lost_time_vibration,0)) ' hours'])
disp(['Lost time due to Bit wear: ' num2str(round(rop_lost_time_wear,0)) ' hours'])
disp(['Lost time due to Hard Rock: ' num2str(round(rop_lost_time_ucs,0)) ' hours'])
% Plot bit runs, bit size, new rop, drilling events and NPT and lost time

depth_text_1 =  (TD+200)/5;
depth_text_2 = depth_text_1 + 150;

figure(37)
subplot(1,4,1)
scatter(bit_nr, bit_in, 30, 'filled')
hold on
scatter(bit_nr, bit_out, 30, 'filled')
scatter(bit_nr, bit_in_core, 30, 'filled')
scatter(bit_nr, bit_out_core, 30, 'filled')
set(gca, 'YDir','reverse')
xlim([0 length(bit_nr)+1])
title('Bit Runs')
xlabel('Bit number')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bit In', 'Bit Out', 'Core run in', 'Core run Out')
num1 = num2str(round(total_trip_time_lost,0));
txt1 = ['Lost time:  '];
txt3 = [' hrs'];
txt4 = [' '];
txt32 = strcat(txt1, txt4);
text((length(bit_nr)*(2/3)), depth_text_1, txt32, 'fontsize', 10,'Interpreter','latex')
txt33 = strcat(num1, txt4, txt3);
text((length(bit_nr)*(2/3)), depth_text_2, txt33, 'fontsize', 10,'Interpreter','latex')
hold off
box on
grid on

subplot(1,4,2)
plot(bdia, mdrkb)
xlim([0 40])
title('Bit Size')
xlabel('Bit Size [in]')
ylim([0, TD+200])
set(gca, 'YDir','reverse')
grid on
box on

subplot(1,4,3)
plot(rop_new, mdrkb, 'linewidt', 0.5)
hold on
plot(rop, mdrkb, 'linewidt', 0.5)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylim([0, TD+200])
xlim([0, rop_lim])
legend('New ROP', 'Original ROP')
num1 = num2str(round(rop_lost_time,0));
txt1 = ['Lost time:  '];
txt3 = [' hrs'];
txt4 = [' '];
txt32 = strcat(txt1, txt4);
text(100, depth_text_1, txt32, 'fontsize', 10,'Interpreter','latex')
txt33 = strcat(num1, txt4, txt3);
text(100, depth_text_2, txt33, 'fontsize', 10,'Interpreter','latex')
hold off
grid on
box on
```

```
subplot(1,4,4)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
num1 = num2str(round(total_lost_time,0));
txt1 = ['Total lost time:  '];
txt3 = [' hrs'];
txt4 = [' '];
txt32 = strcat(txt1, txt4);
text(2.2, depth_text_1, txt32, 'fontsize', 10,'Interpreter','latex')
txt33 = strcat(num1, txt4, txt3);
text(2.2, depth_text_2, txt33, 'fontsize', 10,'Interpreter','latex')
hold off
grid on
box on

nameofplot = 'total_lost_time_';
fig1 = figure(37);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Export to Excel

```
% Transpose vectors before exporting
Bitballing = transpose(bitballing);
Vibration = transpose(vibration);
Bit_Wear = transpose(wear);
Hard_Rock = transpose(ucs_event);

Bitballing(Bitballing == -1) = 0; % Set non flagged events equal to zero.
Vibration(Vibration == -1) = 0; % Set non flagged events equal to zero.
Bit_Wear(Bit_Wear == -1) = 0; % Set non flagged events equal to zero.
Hard_Rock(Hard_Rock == -1) = 0; % Set non flagged events equal to zero.
T = table(mdrkb, tvdrkb, mse, mse_corr, ds, mse_ds, Bitballing, Vibration, Bit_Wear, Hard_Rock, rop, wob, tqa, bdia,
ecd, sppa, sonic);
t = table(total_lost_time, total_trip_time,total_trip_time_lost, time_original, rop_lost_time, rop_lost_time_bb,
rop_lost_time_vibration, rop_lost_time_wear, rop_lost_time_ucs, cut_off, ucs_cutoff_1);
name = '_NPT_Drilling_Events_Detection';
file = '.xlsx';
filename = strcat(wellname, name, file);
writetable(T,filename,'Sheet',1)
writetable(t, filename, 'sheet',2)
```

### *9.3 Updated MATLAB Workflow Torque-Method*

# Detection of NPT and Drilling Events on the NCS from Depth-based Drilling Mechanics Data

# Updated Matlab Workflow Torque-Method

```
clear all
clc
```

Import Well Data

```
% North Sea

%{
% 34/4-15 S
wellname = '34_4-15 S';
```

```
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\34_4-15 S\34_4-15 S Ready');
 drilling_data = xlsread('Ready_34_4-15 S  DRILLING DATA - MUDLOG.xlsx'); % Drilling data
 composite_data =xlsread('Ready_34_4-15 S COMPOSITE_LOG.xlsx'); % Composite data
 bit_data = xlsread('Ready_34_4-15 S  BIT_CHANGES.xlsx'); % Bit data
 %}

 %%{
 % 34/4-15 A
 wellname = '34_4-15 A';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\34_4-15 A\34_4-15 A Ready');
 drilling_data = xlsread('Ready_34_4-15 A  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('34_4-15 A COMPOSITE_LOG.xlsx');
 bit_data = xlsread('Ready_34_4-15 A  BIT_CHANGES.xlsx'); % Bit data
 %}

 %{
 % 34/4-16 S
 wellname = '34_4-16 S';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\34_4-16 S');
 drilling_data = xlsread('Ready_34_4-16 S  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_34_4-16 S COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_34_4-16 S  BIT_CHANGES.xlsx'); % Bit data
 %}


 %{
 % 33/6-3 S
 wellname = '33_6-3 S';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\33_6-3 S\33_6-3 S Ready');
 drilling_data = xlsread('Ready_33_6-3 S  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_33_6-3 S COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_33_6-3 S  BIT_CHANGES.xlsx'); % Bit data
 %}

 %{
 % 35/3-6
 wellname = '35_3-6';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_3-6\35_3-6 Ready');
 drilling_data = xlsread('Ready_35_3-6  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_35_3-6 COMPOSITE_LOG.xlsx');
 bit_data = xlsread('Ready_35_3-6  BIT_CHANGES.xlsx'); % Bit data
 %}

 %{
 % 35/3-4
 wellname = '35_3-4';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_3-4\35_3-4 Ready');
 drilling_data = xlsread('Ready_35_3-4  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_35_3-4 COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_35_3-4  BIT_CHANGES.xlsx'); % Bit data
 %}

 %{
 % 35/8-4
 wellname = '35_8-4';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_8-4\35_8-4 Ready');
 drilling_data = xlsread('Ready_35_8-4  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_35_8-4 COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_35_8-4  BIT_CHANGES.xlsx'); % Bit data
 %}

 %{
 % 35/8-5 S
 wellname = '35_8-5 S';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_8-5 S\35_8-5 S Ready');
 drilling_data = xlsread('Ready_35_8-5 S  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_35_8-5 S COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_35_8-5 S  BIT_CHANGES.xlsx'); % Bit data
 %}
```

```
%{
% 35/9-8
wellname = '35_9-8';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_9-8\35_9-
8 Ready');
drilling_data = xlsread('Ready_35_9-8  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_35_9-8 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_35_9-8  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 35/9-11 S
wellname = '35_9-11 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\35_9-11
S\35_9-11 S Ready');
drilling_data = xlsread('Ready_35_9-11 S  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_35_9-11 S COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_35_9-11 S  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 15/12-18 S
wellname = '15_12-18 S';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\15_12-18 S
Ready');
drilling_data = xlsread('Ready_15_12-18 S DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_15_12-18 S COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_15_12-18 S BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 15/12-21
wellname = '15_12-21';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\15_12-21
Ready');
drilling_data = xlsread('Ready_15_12-21  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_15_12-21 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_15_12-21  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 15/12-23
wellname = '15_12-23';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\15_12-23
Ready');
drilling_data = xlsread('Ready_15_12-23 DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_15_12-23 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_15_12-23 BIT_CHANGES.xlsx'); % Bit data
%}



% Norwegian Sea

%{
% 6406/1-1
wellname = '6406_1-1';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6406_1-1
Ready');
drilling_data = xlsread('Ready_6406_1-1  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('READY_6406_1-1 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_6406_1-1  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 6406/2-3
wellname = '6406_2-3';
path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6406_2-3
Ready');
drilling_data = xlsread('Ready_6406_2-3  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_6406_2-3 Composite.xlsx');
bit_data = xlsread('Ready_6406_2-3  BIT_CHANGES.xlsx'); % Bit data
%}

%{
% 6406/2-5
```

```
 wellname = '6406_2-5';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6406_2-5
Ready');
 drilling_data = xlsread('Ready_6406_2-5  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_6406_2-5 COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_6406_2-5  BIT_CHANGES.xlsx'); % Bit data
 %}


 %{
 % 6406/2-7
 wellname = '6406_2-7';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6406_2-7
Ready');
 drilling_data = xlsread('Ready_6406_2-7  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_6406_2-7 COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_6406_2-7  BIT_CHANGES.xlsx'); % Bit data
 %}

 %{
 % 6506/11-6
 wellname = '6506_11-6';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6506_11-6
Ready');
 drilling_data = xlsread('Ready_6506_11-6  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_6506_11-6 COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_6506_11-6  BIT_CHANGES.xlsx'); % Bit data
 %}


 %{
 % 6407/8-4 S
 wellname = '6407_8-4 S';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6407_8-4 S
Ready');
 drilling_data = xlsread('Ready_6407_8-4 S  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_6407_8-4 S COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_6407_8-4 S  BIT_CHANGES.xlsx'); % Bit data
 %}

 %{
 % 6407/8-5 S
 wellname = '6407_8-5 S';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6407_8-5 S
Ready');
 drilling_data = xlsread('Ready_6407_8-5 S  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_6407_8-5 S COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_6407_8-5 S  BIT_CHANGES.xlsx'); % Bit data
 %}

 %{
 % 6407/8-6
 wellname = '6407_8-6';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6407_8-6
Ready');
 drilling_data = xlsread('Ready_6407_8-6 DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_6407_8-6 COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_6407_8-6  BIT_CHANGES.xlsx'); % Bit data
 %}



 %{
 % 6407/8-7
 wellname = '6407_8-7';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6407_8-7
Ready');
 drilling_data = xlsread('Ready_6407_8-7  DRILLING DATA - MUDLOG.xlsx');
 composite_data =xlsread('Ready_6407_8-7 COMPOSITE_LOG_.xlsx');
 bit_data = xlsread('Ready_6407_8-7  BIT_CHANGES.xlsx'); % Bit data
 %}

 %{
 % 6407/9-9
 wellname = '6407_9-9';
 path(path, 'C:\Users\Håvard\Documents\skole\Master_UiS\4. Master-oppgave\well data\well data\6407_9-9
Ready');
```

```
drilling_data = xlsread('Ready_6407_9-9  DRILLING DATA - MUDLOG.xlsx');
composite_data =xlsread('Ready_6407_8-7 COMPOSITE_LOG_.xlsx');
bit_data = xlsread('Ready_6407_9-9  BIT_CHANGES.xlsx'); % Bit data
%}
```

Name Drilling Data and Composite logs variables

```
% Name drilling data
tvdrkb = drilling_data(:,1); % True Vertical Depth, meter RKB
mdrkb = drilling_data(:,2);  % Measured Depth, meter RKB
bdia = drilling_data(:,3);   % Bit Size, inches
rop = drilling_data(:,4);    % Rate of Penetration, m/hr
hkla = drilling_data(:,5);   % Hook-load average, ton
hklx = drilling_data(:,6);   % Hook-load max, ton
wob = drilling_data(:,7);    % Weight on Bit, ton
tqa = drilling_data(:,8);    % Torque average, KNm
tqx = drilling_data(:,9);    % Torque max, KNm
rpmb = drilling_data(:,10);  % Revolutions per minute bit, 1/min
sppa = drilling_data(:,11);  % StandPipe Pressure, bar
tva = drilling_data(:,12);   % Active Tank Volume, m^3
mfoa = drilling_data(:,13);  % Mud-flow out average, liter/min
mfia = drilling_data(:,14);  % Mud-flow in average, liter/min
mwout = drilling_data(:,15); % Mud-weight out, sg
mwin = drilling_data(:,16);  % Mud-weight in, sg
mtoa = drilling_data(:,17);  % Mud Temperature out average, °C
mtia = drilling_data(:,18);  % Mud Temperature in average, °C
ecd = drilling_data(:,19);   % Equivalent Circulating Density, sg
ppore = drilling_data(:,20); % Pore Pressure, sg
lith = drilling_data(:,21);  % Lithology
dexp = drilling_data(:,22);  % Drilling exponent
TD = max(mdrkb);             % Target Depth, meter MD RKB
% Name Composite Log Data
gammaray_raw = composite_data(:, 5);        % Gamma-ray, raw data, gAPI
gammaray_depth_raw = composite_data(:, 4);  % Gamma-ray, depth meter MD RKB
sonic_raw = composite_data(:, 7);           % Sonic log, raw data, µs/ft
sonic_depth_raw = composite_data(:, 6);     % Sonic log, depth meter MD RKB

% Create equal length arrays for composite logs and drilling data
gammaray_raw(isnan(gammaray_raw)) = []; % Remove NaN values
gammaray_depth_raw(isnan(gammaray_depth_raw)) = []; % Remove NaN values
sonic_raw(isnan(sonic_raw)) = []; % Remove NaN values
sonic_depth_raw(isnan(sonic_depth_raw)) = []; % Remove NaN values

gammaray = interp1(gammaray_depth_raw, gammaray_raw, mdrkb); % Gamma-ray, gAPI
sonic = interp1(sonic_depth_raw, sonic_raw, mdrkb);          % Sonic log, µs/ft
```

Plot the Variables

```
figure(1) % Plot drilling parameters: ROP, WOB, RPM, Torque, Mud flow in, SPPA and ECD, Bit-size

subplot(1,8,1)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
grid on
box on

subplot(1,8,2)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
grid on
box on

subplot(1,8,3)
plot(rpmb, mdrkb)
set(gca, 'YDir','reverse')
title('RPM')
xlabel('RPM')
```

```
grid on
box on

subplot(1,8,4)
plot(tqa, mdrkb)
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
grid on
box on

subplot(1,8,5)
plot(mfia, mdrkb)
set(gca, 'YDir','reverse')
title('Mud Flow In')
xlabel('Flow [liter/min]')
grid on
box on

subplot(1,8,6)
plot(sppa, mdrkb)
set(gca, 'YDir','reverse')
title('SPPA')
xlabel('SPPA [bar]')
grid on
box on

subplot(1,8,7)
plot(ecd, mdrkb)
set(gca, 'YDir','reverse')
title('ECD')
xlabel('ECD [s.g.]')
xlim ([0 2.5])
grid on
box on

subplot(1,8,8)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [inch]')
xlim([0 40])
grid on
box on

nameofplot = 'Drilling_Data_figure_';
fig1 = figure(1);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% Check if required drilling data is available
if sum(tqa)>0 && sum(rpmb)>0 && sum(rop)>0 && sum(wob)>0 && sum(bdia)>0
    disp('All required Drilling Data is available')
else
    disp('Data missing')
end
figure(2) % Plot Gamma-ray, Sonic log and Lithology
subplot(1,3,1)
plot(gammaray, mdrkb)
set(gca, 'YDir','reverse')
title('Gamma-Ray')
xlabel('Gamma-ray [gAPI]')
ylabel('Depth [mMD]')
ylim([0 TD+200])
grid on
box on

subplot(1,3,2)
plot(sonic, mdrkb)
set(gca, 'YDir','reverse')
title('Sonic Log')
xlabel('Interval velocity [µs/ft]')
```

```
ylim([0 TD+200])
grid on
box on

L = zeros(length(mdrkb), 1);
k = 0;
for i = 1:length(mdrkb);
    k=k+1;
    if lith(k) >= 500 && lith(k) < 700 % Claystone
        claystone(k) = L(i)+1;
    else
        claystone(k) = L(i) - 1;
    end
end

k = 0;
for i = 1:length(mdrkb);
    k=k+1;
    if lith(k) >= 300 && lith(k) < 400 % Sandstone
        sandstone(k) = L(i)+2;
    else
        sandstone(k) = L(i) - 1;
    end
end

k = 0;
for i = 1:length(mdrkb);
    k=k+1;
    if lith(k) >= 700 && lith(k) < 800 % Carbonates
        carbonates(k) = L(i)+3;
    else
        carbonates(k) = L(i) - 1;
    end
end

subplot(1,3,3)
scatter(claystone, mdrkb, 20, 'filled', 'g')
hold on
scatter(sandstone, mdrkb, 20, 'filled', 'y')
scatter(carbonates, mdrkb, 20, 'filled', 'b')
set(gca, 'YDir','reverse')
title('Lithology')
xlabel('Lithology')
hold off
xlim([0, 4])
legend('Claystone', 'Sandstone', 'Carbonates')
ylim([0 TD+200])
grid on
box on

nameofplot = 'Gamma_sonic_lithology_';
fig1 = figure(2);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

## Calculate Mechanical Specific Energy

```
wob_lbs = wob*2204.62; % Convert from tonnes to lbs
tqa_ft_lbs = tqa*737.56; % Convert from knm to ft-lbs
rop_ft_hr = rop*3.28; % Convert from m/hr to ft/hr
factor = 0.35; % Efficiency factor

mse_psi = factor * (wob_lbs./((pi/4).*(bdia.^2)) +
(120*pi.*rpmb.*tqa_ft_lbs)./(((pi/4).*(bdia.^2)).*rop_ft_hr)); % Mechanical specific energy, psi
mse = mse_psi./145; % Convert MSE from PSI to MPa
```

Plot MSE and Drilling Data

```
figure(3) % Plot drilling parameters: MSE, ROP, WOB, Torque and bit size
subplot(1,5,1)
```

```
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('MSE')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
xlim([0 500]) % xlim([0 1500])
grid on
box on

subplot(1,5,2)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
grid on
box on

subplot(1,5,3)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
grid on
box on

subplot(1,5,4)
plot(tqa, mdrkb)
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
grid on
box on

subplot(1,5,5)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [Inch]')
xlim([0 40])
grid on
box on

nameofplot = 'drillingparameters_';
fig1 = figure(3);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
figure(4) % Plot drilling parameters: MSE, ROP, WOB, RPM, Torque, Mud flow in, SPPA, ECD and Bit-size
subplot(1,9,1)
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('MSE')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
xlim([0 500]) % xlim([0 1500])
grid on
box on

subplot(1,9,2)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
xlim([0 200])
grid on
box on

subplot(1,9,3)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
```

```
grid on
box on

subplot(1,9,4)
plot(rpmb, mdrkb)
set(gca, 'YDir','reverse')
title('RPM')
xlabel('RPM')
grid on
box on

subplot(1,9,5)
plot(tqa, mdrkb)
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
grid on
box on

subplot(1,9,6)
plot(mfia, mdrkb)
set(gca, 'YDir','reverse')
title('Mud Flow In')
xlabel('Flow [liter/min]')
grid on
box on

subplot(1,9,7)
plot(sppa, mdrkb)
set(gca, 'YDir','reverse')
title('SPPA')
xlabel('SPPA [bar]')
grid on
box on

subplot(1,9,8)
plot(ecd, mdrkb)
set(gca, 'YDir','reverse')
title('ECD')
xlabel('ECD [s.g.]')
grid on
box on

subplot(1,9,9)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [inch]')
xlim([0 40])
grid on
box on

nameofplot = 'drillingparameters2_';
fig1 = figure(4);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

### Plot MSE and Create MSE Corrected

```
figure(5) % Plot Mechanical specific energy
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('Mechanical specific energy')
xlabel('Mechanical specific energy [MPa]')
ylabel('Depth [mMD]')
xlim([0 1500])
grid on
box on

nameofplot = 'Mechanical_specific_energy_';
fig1 = figure(5);
```

```
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Create Baseline for MSE*

```
% Manually insert the baselines

depth_interval = 1; % Interval for each measurement in the drilling data excel file. Default is 1 meter.

% North Sea
%{
% 34/4-15 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2001]; % Depth Start Baseline
baseline_mse_max = [2000, max(mdrkb)]; % Depth Stop Baseline
mse_min = [8, 8];   % MSE Value Start Baseline
mse_max = [8, 140]; % MSE Value Stop Baseline
%}

%%{
% 34/4-15 A
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2001];
baseline_mse_max = [2000, max(mdrkb)];
mse_min = [8, 8];
mse_max = [8, 130];
%}

%{
% 34/4-16 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2300];
baseline_mse_max = [2300, max(mdrkb)];
mse_min = [8, 8];
mse_max = [8, 100];
%}

%{
% 33/6-3 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1601];
baseline_mse_max = [1600, max(mdrkb)];
mse_min = [10, 12];
mse_max = [12, 110];
%}

%{
% 35/3-6
baselines = 3; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1130, 3081];
baseline_mse_max = [1129, 3080, max(mdrkb)];
mse_min = [2, 2, 200];
mse_max = [2, 200, 170];
%}

%{
% 35/3-4
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2001];
baseline_mse_max = [2000, max(mdrkb)];
mse_min = [2, 2];
mse_max = [2, 70];
%}

%{
% 35/8-4
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2501];
baseline_mse_max = [2500, max(mdrkb)];
mse_min = [5, 20];
```

```
mse_max = [20, 160];
%}

%{
% 35/8-5 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2001];
baseline_mse_max = [2000, max(mdrkb)];
mse_min = [5, 5];
mse_max = [5, 140];
%}

%{
% 35/9-8
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1001];
baseline_mse_max = [1000, max(mdrkb)];
mse_min = [5, 5];
mse_max = [5, 230];
%}

%{
% 35/9-11 S
baselines = 3; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1700.25, 3200.25];
baseline_mse_max = [1700, 3200, max(mdrkb)];
mse_min = [5, 5, 190];
mse_max = [5, 190, 550];
%}

%{
% 15/12-18 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2501];
baseline_mse_max = [2500, max(mdrkb)];
mse_min = [10, 15];
mse_max = [15, 550];
%}

%{
% 15/12-21
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2301];
baseline_mse_max = [2300, max(mdrkb)];
mse_min = [10, 35];
mse_max = [35, 140];
%}

%{
% 15/12-23
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 2301];
baseline_mse_max = [2300, max(mdrkb)];
mse_min = [1, 15];
mse_max = [15, 60];
%}


% Norwegian Sea

%{
% 6406/1-1
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 3601];
baseline_mse_max = [3600, max(mdrkb)];
mse_min = [1, 50];
mse_max = [50, 500];
%}

%{
% 6406/2-3
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 3251];
baseline_mse_max = [3250, max(mdrkb)];
mse_min = [5, 5];
mse_max = [5, 500];
%}
```

```
%{
% 6406/2-5
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 3201];
baseline_mse_max = [3200, max(mdrkb)];
mse_min = [5, 40];
mse_max = [40, 400];
%}

%{
% 6406/2-7
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 3001];
baseline_mse_max = [3000, max(mdrkb)];
mse_min = [5, 25];
mse_max = [25, 400];
%}

%{
% 6506/11-6
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 3101];
baseline_mse_max = [3100, max(mdrkb)];
mse_min = [5, 40];
mse_max = [40, 600];
%}

%{
% 6407/8-4 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1901];
baseline_mse_max = [1900, max(mdrkb)];
mse_min = [5, 15];
mse_max = [15, 200];
%}

%{
% 6407/8-5 S
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1801];
baseline_mse_max = [1800, max(mdrkb)];
mse_min = [5, 20];
mse_max = [20, 150];
%}

%{
% 6407/8-6
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1901];
baseline_mse_max = [1900, max(mdrkb)];
mse_min = [5, 20];
mse_max = [20, 80];
%}

%{
% 6407/8-7
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1901];
baseline_mse_max = [1900, max(mdrkb)];
mse_min = [5, 15];
mse_max = [15, 120];
%}

%{
% 6407/9-9
baselines = 2; % Minimum 1 and maximum 3 baselines
baseline_mse_min = [min(mdrkb), 1619];
baseline_mse_max = [1618, max(mdrkb)];
mse_min = [5, 15];
mse_max = [15, 70];
%}


% Create baselines
if baselines >= 1
```

```
    baseline_1 = baseline_mse_min(1):depth_interval:baseline_mse_max(1); % Default:
baseline_mse_min(1):1:baseline_mse_max(1);
    baseline1 = [baseline_mse_min(1), baseline_mse_max(1)];
    mse_1 = [mse_min(1), mse_max(1)];
    baseline_mse_1 = (interp1(baseline1, (mse_1), baseline_1));
end

if baselines >= 2
    baseline_2 = baseline_mse_min(2):depth_interval:baseline_mse_max(2); % Default:
baseline_mse_min(2):1:baseline_mse_max(2);
    baseline2 = [baseline_mse_min(2), baseline_mse_max(2)];
    mse_2 = [mse_min(2), mse_max(2)];
    baseline_mse_2 = (interp1(baseline2, (mse_2), baseline_2));
end

if baselines >= 3
    baseline_3 = baseline_mse_min(3):depth_interval:baseline_mse_max(3); % Default:
baseline_mse_min(3):1:baseline_mse_max(3);
    baseline3 = [baseline_mse_min(3), baseline_mse_max(3)];
    mse_3 = [mse_min(3), mse_max(3)];
    baseline_mse_3 = (interp1(baseline3, (mse_3), baseline_3));
end

if baselines == 1
   baseline_mse = [baseline_mse_1];
elseif baselines == 2
    baseline_mse = [baseline_mse_1, baseline_mse_2];
elseif baselines == 3
   baseline_mse = [baseline_mse_1, baseline_mse_2, baseline_mse_3];
end
```

*Plot Baseline and MSE*

```
figure(6)
subplot(1,2,1)
plot(mse, mdrkb)
hold on
plot(baseline_mse, mdrkb, 'linewidth', 1.5)
hold off
set(gca, 'YDir','reverse')
title('Mechanical specific energy')
xlabel('Mechanical specific energy [MPa]')
ylabel('Depth [mMD]')
legend('MSE', 'Baseline MSE')
xlim([0 1500])
grid on
box on

subplot(1,2,2)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit Size [in]')
xlim([0 40])
grid on
box on

nameofplot = 'Mechanical_specific_energy_baseline_';
fig1 = figure(6);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Create and Plot MSE Corrected*

```
baseline_mse_transposed = transpose(baseline_mse); % Transpose the baseline vector
mse_corr = mse - baseline_mse_transposed; % Create MSE corrected
mse_corr(mse_corr < 0) = 0; % Remove negative MSE values

figure(7)
subplot(1,2,1)
```

```
plot(mse, mdrkb)
hold on
plot(baseline_mse, mdrkb, 'linewidth', 1.5)
hold off
set(gca, 'YDir','reverse')
title('Mechanical Specific Energy')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
legend('MSE', 'Baseline MSE')
xlim([0 1500])
grid on
box on

subplot(1,2,2)
plot(mse_corr, mdrkb)
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('MSE [MPa]')
xlim([0 1500])
grid on
box on

nameofplot = 'MSE_Corrected_';
fig1 = figure(7);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

### Calculate Drilling Strength and MSE/DS ratio

*Calculate Drilling Strength and MSE/DS ratio*

```
doc = rop_ft_hr./(rpmb.*5); % Depth of cut, [inch]
ds_psi = wob_lbs./(bdia.*doc); % Drilling strength, [psi]
ds = ds_psi /145; % Convert from psi to MPa
mse_ds = mse./ds; % Ratio between Mechanical spesific energy and drilling strength
```

*Plot MSE, MSE Corrected, DS and MSE/DS*

```
figure(8)
subplot(1,4,1)
plot(mse, mdrkb)
set(gca, 'YDir','reverse')
title('MSE')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
xlim([0 1500])
grid on
box on

subplot(1,4,2)
plot(mse_corr, mdrkb)
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('MSE [MPa]')
xlim([0 1500])
grid on
box on

subplot(1,4,3)
plot(ds, mdrkb)
set(gca, 'YDir','reverse')
title('DS')
xlabel('DS [MPa]')
xlim ([0 3000])  %([0 1500])
grid on
box on

subplot(1,4,4)
plot(mse_ds, mdrkb)
set(gca, 'YDir','reverse')
title('MSE/DS')
```

```
xlabel('MSE/DS')
xlim([0, 10])
grid on
box on

nameofplot = 'MSE_MSE_Corr_DS_MSEDS_';
fig1 = figure(8);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Estimate UCS

```
% Estimate UCS according to lithology

for i = 1:length(mdrkb)
    if claystone (i) == 1
        UCS(i) = 1.35*(304.8/sonic(i))^2.6; % Globally Shale
    elseif carbonates(i) == 3
        UCS (i) = ((7682/sonic(i))^1.82)/145; % MilitzerStoll Lime
    elseif sandstone(i) == 2
        if tvdrkb(i) < 2000
            UCS(i) = 1.4138*10^7*(sonic(i)^-3); % GulfCoast WeakSand
        elseif 2000 <= tvdrkb(i)
            UCS(i) = 1200*exp(-0.036*sonic(i)); % BowenBasinAustralia_Sand
        end
    else
        UCS(i) = NaN;
    end
end

for i = 1:length(mdrkb)
    if UCS(i) >= 1000;
        UCS(i) = NaN;
    end
end

figure(18)
subplot(1,3,1)
plot(UCS, mdrkb)
title('UCS')
set(gca, 'YDir','reverse')
xlabel('UCS [MPa]')
ylabel('Depth [mMD]')
ylim([0, TD+200])
grid on
box on

subplot(1,3,2)
scatter(claystone, mdrkb, 20, 'filled', 'g')
hold on
scatter(sandstone, mdrkb, 20, 'filled', 'y')
scatter(carbonates, mdrkb, 20, 'filled', 'b')
set(gca, 'YDir','reverse')
title('Lithology')
xlabel('Lithology')
hold off
xlim([0, 4])
ylim([0 TD+200])
legend('Claystone', 'Sandstone', 'carbonates')
grid on
box on

subplot(1,3,3)
plot(sonic, mdrkb)
set(gca, 'YDir','reverse')
title('Sonic')
xlabel('Interval velocity [µs/ft]')
ylim([0 TD+200])
grid on
box on
```

```
nameofplot = 'UCS_lithology_sonic_';
fig1 = figure(18);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% Sort UCS estimations according to Hardness Category
ucs = transpose(UCS);
for i = 1:length(mdrkb)
    if ucs(i) <= 0.6 && ucs(i) > 0
        hardness(i) = 1;                    % Soil < 0.6 MPa
    elseif ucs(i) <= 1.25 && ucs(i) > 0.6
        hardness(i) = 2;                    % Very Soft Rock: 0.6 - 1.25 MPa
    elseif ucs(i) <= 5 && ucs(i) > 1.25
        hardness(i) = 3;                    % Soft Rock = 1.25 – 5.0 MPa
    elseif ucs(i) <= 12.5 && ucs(i) > 5
        hardness(i) = 4;                    % Moderately Soft Rock: 5 – 12.5 MPa
    elseif ucs(i) <= 50 && ucs(i) > 12.5
        hardness(i) = 5;                    % Moderately Hard Rock: 12.5 - 50 MPa
    elseif ucs(i) <= 100 && ucs(i) > 50
        hardness(i) = 6;                    % Hard Rock: 50-100 MPa
    elseif ucs(i) <= 250 && ucs(i) > 100
        hardness(i) = 7;                    % Very Hard Rock: 100 - 250 MPa
    elseif ucs(i) < 1000 && ucs(i) > 250
        hardness(i) = 8;                    % Extremely Hard Rock > 250 MPa
    else
        hardness(i) = NaN;
    end
end
figure(23)
subplot(1,2,1)
scatter(ucs, mdrkb, 20, hardness, 'filled')
title('Hardness Rock')
xlabel('UCS [MPa]')
ylabel('Depth [mMD]')
set(gca, 'YDir','reverse')
colorbar
hard = colorbar;
ylabel(hard, 'Hardness Category');
ylim([0 TD+100])
colorbar('Ticks',[1,2,3,4,5,6,7,8], 'TickLabels', {'','','','','','','',''})
colorbar off
grid on
box on

subplot(1,2,2)
scatter(mse, mdrkb, 20, hardness, 'filled')
title('Hardness Rock')
xlabel('MSE [MPa]')
set(gca, 'YDir','reverse')
colorbar
hard = colorbar;
ylabel(hard, 'Hardness Category');
ylim([0 TD+100])
xlim([0 2000])
colorbar('Ticks',[1,2,3,4,5,6,7,8], 'TickLabels',{'Soil','Very Soft Rock','Soft Rock','Moderately Soft
Rock','Moderately Hard Rock', 'Hard Rock', 'Very Hard Rock', 'Extremely Hard Rock'})
grid on
box on

nameofplot = 'Hardness_UCS_MSE_';
fig1 = figure(23);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Create Cut-offs*

```
cut_off = 150; % Cut-off value for bitballing, bit wear and Vibrations [MPa]
ucs_cutoff_1 = 100; % Cut-off value for Very Hard Rock [MPa]
ineff_mse_ds = 1.5; % Cut off values for inefficient drilling mse/ds ratio

a = ones(length(mdrkb), 1); % Array of ones the length of the well
mse_cutoff = cut_off .* a; % Array of MSE cut-off [MPa]
ucs_cutoff = ucs_cutoff_1.*a; % Array of UCS cut-off
ineff_mse_ds_cutoff = ineff_mse_ds.*a; % Array of mse/ds cut-off
```

*Plot MSE corrected with cut-off values, DS and MSE/DS*

```
figure(9)
subplot(1,3,1)
plot(mse_corr, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
plot(ucs, mdrkb, 'linewidt',1)
plot(ucs_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
legend('MSE corrected', 'Cut-Off MSE', 'UCS', 'Cut-Off UCS')
xlim([0 1500])
grid on
box on

subplot(1,3,2)
plot(ds, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('DS')
xlabel('DS [MPa]')
xlim([0 1500])
legend('DS', 'Cut-Off')
grid on
box on

subplot(1,3,3)
plot(mse_ds, mdrkb)
hold on
plot(ineff_mse_ds_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE/DS')
xlabel('MSE/DS')
legend('DS', 'Cut-Off')
xlim([0, 10])
grid on
box on

nameofplot = 'Cutoff_values_mse_ds_ucs_';
fig1 = figure(9);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Detecting Bit Balling, Vibration, Wear or UCS*

```
b = zeros(length(mdrkb), 1); % Array of zeros
b0 = b - 1; % Not an event
b1 = b+1; % Bit balling
b2 = b + 2; % Vibration
b3 = b + 3; % Wear
b4 = b + 4; % Hard Rock
lim_ucs = 1000.*a; % Limit UCS array
```

```
k = 0;
    for i =1:length(mdrkb);
        k = k+1;
        ucs_event(k) = b0(i);
        bitballing (k) = b0 (i);
        wear(k) = b0(i);
        vibration (k) = b0(i);
    end

k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if ucs(i) >= ucs_cutoff(i) && ucs(i) <= lim_ucs(i)
        ucs_event(k) = b4(i);
    end
end

k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if mse_corr(k) >= mse_cutoff(k) && ds(k) >= mse_cutoff(k) && mse_ds(k) <= ineff_mse_ds_cutoff(k) &&
claystone(k) == 1
            bitballing (k) = b1 (i);
    end
end

k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if mse_corr(k) >= mse_cutoff(k) && ds(k) >= mse_cutoff(k) && mse_ds(k) <= ineff_mse_ds_cutoff(k) &&
bitballing(k) == -1 %claystone(k) ~= 1
            wear (k) = b3 (i);
    end
end

k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if mse_corr(k) >= mse_cutoff(k) && ds(k) >= mse_cutoff(k) && mse_ds(k) > ineff_mse_ds_cutoff(k)
            vibration (k) = b2 (i);
    end
end
```

*Plot MSE Corrected, DS, MSE/DS and Drilling Events*

```
figure(10)
subplot(1,4,1)
plot(mse_corr, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
plot(ucs, mdrkb, 'linewidt',1)
plot(ucs_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE Corrected')
xlabel('MSE [MPa]')
ylabel('Depth [mMD]')
legend('MSE corrected', 'Cut-Off MSE', 'UCS', 'UCS Cut-Off')
xlim([0 1500])
ylim([0, TD+200])
grid on
box on

subplot(1,4,2)
plot(ds, mdrkb)
hold on
plot(mse_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('DS')
xlabel('DS [Mpa]')
xlim([0 1500])
legend('DS', 'Cut-Off')
ylim([0, TD+200])
grid on
box on
```

```
subplot(1,4,3)
plot(mse_ds, mdrkb)
hold on
plot(ineff_mse_ds_cutoff, mdrkb, 'linewidt', 2)
hold off
set(gca, 'YDir','reverse')
title('MSE/DS')
xlabel('MSE/DS')
legend('DS', 'Cut-off Value')
xlim([0, 10])
ylim([0, TD+200])
grid on
box on

subplot(1,4,4)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
grid on
box on
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])

nameofplot = 'Drilling events & NPT detection_';
fig1 = figure(10);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

*Plot Drilling Events and NPT Detection and Drilling Parameters*

```
figure(11) % Plot Drilling events & NPT, ROP, WOB, RPM, Torque, Mud flow in, SPPA and ECD, Bit-size
subplot(1,9,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 6])
grid on
box on

subplot(1,9,2)
plot(rop, mdrkb)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylim([0, TD+200])
grid on
box on


subplot(1,9,3)
plot(wob, mdrkb)
set(gca, 'YDir','reverse')
title('WOB')
xlabel('WOB [ton]')
```

```
ylim([0, TD+200])
grid on
box on

subplot(1,9,4)
plot(rpmb, mdrkb)
set(gca, 'YDir','reverse')
title('RPM')
xlabel('RPM')
ylim([0, TD+200])
grid on
box on

subplot(1,9,5)
plot(tqa, mdrkb)
set(gca, 'YDir','reverse')
title('Torque')
xlabel('Torque [kNm]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,6)
plot(mfia, mdrkb)
set(gca, 'YDir','reverse')
title('Mud Flow In')
xlabel('Flow [liter/min]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,7)
plot(sppa, mdrkb)
set(gca, 'YDir','reverse')
title('SPPA')
xlabel('SPPA [bar]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,8)
plot(ecd, mdrkb)
set(gca, 'YDir','reverse')
title('ECD')
xlabel('ECD [s.g.]')
ylim([0, TD+200])
grid on
box on

subplot(1,9,9)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
title('Bit Size')
xlabel('Bit size [inch]')
xlim([0 40])
ylim([0, TD+200])
grid on
box on

nameofplot = 'Detection NPT & events and drillingparameters_';
fig1 = figure(11);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

## Investigate NPT and drilling events for each hole section

*Create new variables for each hole section*

```
% For each possible hole section diameter, new create variables for wob, rop,
% mdrkb, torque, rpm and bit-size. If hole section is not present, set
% equal to NaN.
```

```
k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 6
        wob_6(k) = wob(i);
        rop_6(k) = rop(i) ;
        mdrkb_6 (k) = mdrkb(i);
        tqa_6 (k) = tqa(i);
        rpmb_6(k) = rpmb(i);
        bitsize_6(k) = 6;
    else
        wob_6(k) = NaN;
        rop_6(k) = NaN ;
        mdrkb_6 (k) = NaN;
        tqa_6 (k) = NaN;
        rpmb_6(k) = NaN;
        bitsize_6(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 8.5
        wob_8_5(k) = wob(i);
        rop_8_5(k) = rop(i) ;
        mdrkb_8_5 (k) = mdrkb(i);
        tqa_8_5 (k) = tqa(i);
        rpmb_8_5(k) = rpmb(i);
        bitsize_8_5(k) = 8.5;
    else
        wob_8_5(k) = NaN;
        rop_8_5(k) = NaN ;
        mdrkb_8_5 (k) = NaN;
        tqa_8_5 (k) = NaN;
        rpmb_8_5(k) = NaN;
        bitsize_8_5(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 9.88
        wob_9_88(k) = wob(i);
        rop_9_88(k) = rop(i) ;
        mdrkb_9_88 (k) = mdrkb(i);
        tqa_9_88 (k) = tqa(i);
        rpmb_9_88(k) = rpmb(i);
        bitsize_9_88(k) = 9.88;
    else
        wob_9_88(k) = NaN;
        rop_9_88(k) = NaN ;
        mdrkb_9_88 (k) = NaN;
        tqa_9_88 (k) = NaN;
        rpmb_9_88(k) = NaN;
        bitsize_9_88(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 12.25
        wob_12_25(k) = wob(i);
        rop_12_25(k) = rop(i) ;
        mdrkb_12_25 (k) = mdrkb(i);
        tqa_12_25 (k) = tqa(i);
        rpmb_12_25(k) = rpmb(i);
        bitsize_12_25(k) = 12.25;
    else
        wob_12_25(k) = NaN;
        rop_12_25(k) = NaN ;
        mdrkb_12_25 (k) = NaN;
        tqa_12_25 (k) = NaN;
        rpmb_12_25(k) = NaN;
```

```
        bitsize_12_25(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 17.5
        wob_17_5(k) = wob(i);
        rop_17_5(k) = rop(i) ;
        mdrkb_17_5 (k) = mdrkb(i);
        tqa_17_5 (k) = tqa(i);
        rpmb_17_5(k) = rpmb(i);
        bitsize_17_5(k) = 17.5;
    else
        wob_17_5(k) = NaN;
        rop_17_5(k) = NaN ;
        mdrkb_17_5 (k) = NaN;
        tqa_17_5 (k) = NaN;
        rpmb_17_5(k) = NaN;
        bitsize_17_5(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 26
        wob_26(k) = wob(i);
        rop_26(k) = rop(i) ;
        mdrkb_26 (k) = mdrkb(i);
        tqa_26 (k) = tqa(i);
        rpmb_26(k) = rpmb(i);
        bitsize_26(k) = 26;
    else
        wob_26(k) = NaN;
        rop_26(k) = NaN ;
        mdrkb_26 (k) = NaN;
        tqa_26 (k) = NaN;
        rpmb_26(k) = NaN;
        bitsize_26(k) = 0;
    end
end

k = 0;
for i= 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 36
        wob_36(k) = wob(i);
        rop_36(k) = rop(i) ;
        mdrkb_36 (k) = mdrkb(i);
        tqa_36 (k) = tqa(i);
        rpmb_36(k) = rpmb(i);
        bitsize_36(k) = 36;
    else
        wob_36(k) = NaN;
        rop_36(k) = NaN ;
        mdrkb_36 (k) = NaN;
        tqa_36 (k) = NaN;
        rpmb_36(k) = NaN;
        bitsize_36(k) = 0;
    end
end
```

*Plot Drilling events & NPT with WOB-ROP-Depth, WOB-Torque-Depth and WOB-Torque-ROP Crossplots for Each Bit Size*

```
% Find out which bit-sizes are in use in the drilling data excel file
bits=0;
disp('The bit-sizes used for this well are:')
if sum(bitsize_9_88)>0
    disp('9 7/8"')
    bits=bits+1;
end
if sum(bitsize_36)>0
    disp('36"')
    bits=bits+1;
end
```

```
if sum(bitsize_26)>0
    disp('26"')
    bits=bits+1;
end
if sum(bitsize_17_5)>0
    disp('17 1/2"')
    bits=bits+1;
end
if sum(bitsize_12_25)>0
    disp('12 1/4"')
    bits=bits+1;
end
if sum(bitsize_8_5)>0
    disp('8 1/2"')
    bits=bits+1;
end
if sum(bitsize_6)>0
    disp('6"')
    bits=bits+1;
end

disp(['The drilling data contains ' num2str(bits) ' different bit-sizes'])
rop_lim = 120; % Limit on y-axis for ROP. Default 120 m/hr.
wob_lim = 25; % Limit on x-axis for WOB. Default 25 ton.
tqa_lim = 35; % Limit on Y-axis for Torque. Default 35 KNm.
% Plot drilling events and NPT and WOB/ROP plot for each hole section
figure(12)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on


p=1; % Subplot number
if sum(bitsize_9_88)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_9_88, rop_9_88, 20, mdrkb, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_36)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_36, rop_36, 20, mdrkb, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
```

```
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_26)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_26, rop_26, 20, mdrkb, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_17_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_17_5, rop_17_5, 20, mdrkb, 'filled')
    title('17.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_12_25)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_12_25, rop_12_25, 20, mdrkb, 'filled')
    title('12.25" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_8_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_8_5, rop_8_5, 20, mdrkb, 'filled')
    title('8.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end
```

```
if sum(bitsize_6)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_6, rop_6, 20, mdrkb, 'filled')
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

nameofplot = 'wob_rop_depth_bitsize_events_NPT_';
fig1 = figure(12);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% WOB-Torque-Depth Crossplot
figure(6354534)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

tqa_lim = 35; % Limit torque yaxis. Default 35 knm

p=1; % Subplot number
if sum(bitsize_9_88)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_9_88, tqa_9_88, 20, mdrkb, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_36)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_36, tqa_36, 20, mdrkb, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
```

```
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_26)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_26, tqa_26, 20, mdrkb, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_17_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_17_5, tqa_17_5, 20, mdrkb, 'filled')
    title('17.5" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_12_25)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_12_25, tqa_12_25, 20, mdrkb, 'filled')
    title('12.25" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_8_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_8_5, tqa_8_5, 20, mdrkb, 'filled')
    title('8.5" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
```

```
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_6)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_6, tqa_6, 20, mdrkb, 'filled')
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

nameofplot = 'wob_torque_depth_bitsize_events_NPT_';
fig1 = figure(6354534);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% WOB-Torque-ROP Crossplot
figure(65325223)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

tqa_lim = 35; % Limit torque yaxis. Default 35 knm

p=1; % Subplot number
if sum(bitsize_9_88)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_9_88, tqa_9_88, 20, rop, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_36)>0
```

```
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_36, tqa_36, 20, rop, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_26)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_26, tqa_26, 20, rop, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_17_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_17_5, tqa_17_5, 20, rop, 'filled')
    title('17.5" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_12_25)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_12_25, tqa_12_25, 20, rop, 'filled')
    title('12.25" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_8_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_8_5, tqa_8_5, 20, rop, 'filled')
    title('8.5" Section')
    xlabel('WOB [ton]')
```

```
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'ROP [m/hr]');
    %set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_6)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_6, tqa_6, 20, rop, 'filled')
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'ROP [m/hr]');
    set(depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

nameofplot = 'wob_torque_rop_bitsize_events_NPT_';
fig1 = figure(65325223);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Look at each hole section separately

6" Hole Section

```
% If 6" hole section exist, calculate efficient bit line and plot
if sum(bitsize_6)>0
    k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            efficient_rop_6(k) = rop_6(i);
            efficient_wob_6(k) = wob_6(i);
            efficient_tqa_6(k) = tqa_6(i);
            eff_rop_6 (k) = rop_6(i);
            eff_wob_6 (k) = wob_6(i);
            eff_tqa_6(k) = tqa_6(i);
        else
            ineff_rop_6(k) = rop_6(i);
            ineff_wob_6(k) = wob_6(i);
            ineff_tqa_6(k) = tqa_6(i);
        end
    end

    efficient_rop_6(isnan(efficient_rop_6)) = []; % Remove NaN values
    efficient_wob_6(isnan(efficient_wob_6)) = []; % Remove NaN values
    efficient_tqa_6(isnan(efficient_tqa_6)) = []; % Remove NaN Values

    c_6 = polyfit(efficient_wob_6, efficient_rop_6, 1); % Create trendline
    disp(['ROP = ' num2str(c_6(1)) '*WOB + ' num2str(c_6(2))]) % % Display equation ROP = a*WOB + b
    d_6 = polyfit(efficient_wob_6, efficient_tqa_6,1); % Create trendline WOB-Torque
    disp(['Torque = ' num2str(d_6(1)) '*WOB + ' num2str(d_6(2))]); % % Display equation Torque = a*WOB +
b
```

```
x= linspace(0,wob_lim); % WOB
y_6 = c_6(1)*x + c_6(2); % Trendline
y2_6 = d_6(1)*x + d_6(2); % Trendline WOB-Torque

figure(133)
subplot(1,3,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 6" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_6), max(mdrkb_6)])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

subplot(1,3,2)
scatter(wob_6, rop_6, 20, mdrkb_6, 'filled')
title('6" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth [mMD]');
set( depth, 'YDir', 'reverse' );
axis([0 wob_lim 0 rop_lim])
hold on
h_6 = plot(x,y_6, '--r','linewidth', 2);
legend(h_6, 'Efficient Bit')
hold off
grid on
box on

subplot(1,3,3)
scatter(eff_wob_6, eff_rop_6, 20, 'filled')
hold on
scatter(ineff_wob_6, ineff_rop_6, 20, 'filled')
plot(x,y_6, '--r','linewidth', 2);
hold off
title('6" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
legend('Efficient', 'Inefficient', 'Efficient Bit Line')
axis([0 wob_lim 0 rop_lim])
grid on
box on

nameofplot = 'wob_rop_depth_bitsize_6_events_NPT';
fig1 = figure(133);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);
h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);


figure(1354543)
subplot(1,3,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 6" Section')
xlabel('')
ylabel('Depth [mMD]')
```

```
    ylim([min(mdrkb_6), max(mdrkb_6)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_6, tqa_6, 20, mdrkb_6, 'filled')
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_6 = plot(x,y2_6, '--r','linewidth', 2);
    legend(h_6, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_6, eff_tqa_6, 20, 'filled')
    hold on
    scatter(ineff_wob_6, ineff_tqa_6, 20, 'filled')
    plot(x,y2_6, '--r','linewidth', 2);
    hold off
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on

    nameofplot = 'wob_torque_depth_bitsize_6_events_NPT';
    fig1 = figure(1354543);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 6" hole section do not exist');
end
```

*8 1/2" Hole Section*

```
% If 8 1/2" hole section exist, calculate efficient bit line and plot
if sum(bitsize_8_5)>0
    k = 0;
    for i = 1:length(mdrkb)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1 % No drilling
events flagged
       efficient_rop_8_5(k) = rop_8_5(i);
       efficient_wob_8_5(k) = wob_8_5(i);
       efficient_tqa_8_5(k) = tqa_8_5(i);
       eff_rop_8_5 (k) = rop_8_5(i);
       eff_wob_8_5 (k) = wob_8_5(i);
       eff_tqa_8_5(k)  = tqa_8_5(i);
    else
        ineff_rop_8_5(k) = rop_8_5(i);
        ineff_wob_8_5(k) = wob_8_5(i);
        ineff_tqa_8_5(k) = tqa_8_5(i);
    end
    end

    efficient_rop_8_5(isnan(efficient_rop_8_5)) = []; % Remove NaN values
    efficient_wob_8_5(isnan(efficient_wob_8_5)) = []; % Remove NaN values
    efficient_tqa_8_5(isnan(efficient_tqa_8_5)) = []; % Remove NaN Values

    c_8_5 = polyfit(efficient_wob_8_5, efficient_rop_8_5, 1); % Create trendline WOB - ROP
```

```
    disp(['ROP = ' num2str(c_8_5(1)) '*WOB + ' num2str(c_8_5(2))]); % % Display equation ROP = a*WOB + b
    d_8_5 = polyfit(efficient_wob_8_5, efficient_tqa_8_5,1); % Create trendline WOB-Torque
    disp(['Torque = ' num2str(d_8_5(1)) '*WOB + ' num2str(d_8_5(2))]); % % Display equation Torque =
a*WOB + b


    x = linspace(0,wob_lim); % WOB
    y_8_5 = c_8_5(1)*x + c_8_5(2); % Trendline WOB-ROP
    y2_8_5 = d_8_5(1)*x + d_8_5(2); % Trendline WOB-Torque


    figure(13)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 8.5" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_8_5), max(mdrkb_8_5)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_8_5, rop_8_5, 20, mdrkb_8_5, 'filled')
    title('8.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_8_5 = plot(x,y_8_5, '--r','linewidth', 2);
    legend(h_8_5, 'Efficient Bit Line')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_8_5, eff_rop_8_5, 20, 'filled')
    hold on
    scatter(ineff_wob_8_5, ineff_rop_8_5, 20, 'filled')
    plot(x,y_8_5, '--r','linewidth', 2);
    hold off
    title('8.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on

    nameofplot = 'wob_rop_depth_bitsize_8_5_events_NPT';
    fig1 = figure(13);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);


    figure(135454)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
```

```
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 8.5" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_8_5), max(mdrkb_8_5)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_8_5, tqa_8_5, 20, mdrkb_8_5, 'filled')
    title('8.5" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h2_8_5 = plot(x,y2_8_5, '--r','linewidth', 2);
    legend(h2_8_5, 'Efficient Bit Line')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_8_5, eff_tqa_8_5, 20, 'filled')
    hold on
    scatter(ineff_wob_8_5, ineff_tqa_8_5, 20, 'filled')
    plot(x,y2_8_5, '--r','linewidth', 2);
    hold off
    title('8.5" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on

    nameofplot = 'wob_Torque_depth_bitsize_8_5_events_NPT';
    fig1 = figure(135454);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

else
    disp('The 8 1/2" hole section do not exist');
end
```

*12 1/4" Hole Section*

```
% If 12 1/4" hole section exist, calculate efficient bit line and plot
if sum(bitsize_12_25)>0
    k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
           efficient_rop_12_25(k) = rop_12_25(i);
           efficient_wob_12_25(k) = wob_12_25(i);
           efficient_tqa_12_25(k) = tqa_12_25(i);
           eff_rop_12_25 (k) = rop_12_25(i);
           eff_wob_12_25 (k) = wob_12_25(i);
           eff_tqa_12_25(k) = tqa_12_25(i);
        else
            ineff_rop_12_25(k) = rop_12_25(i);
            ineff_wob_12_25(k) = wob_12_25(i);
            ineff_tqa_12_25(k) = tqa_12_25(i);
        end
    end
```

```
    efficient_rop_12_25(isnan(efficient_rop_12_25)) = []; % Remove NaN values
    efficient_wob_12_25(isnan(efficient_wob_12_25)) = []; % Remove NaN values
    efficient_tqa_12_25(isnan(efficient_tqa_12_25)) = []; % Remove NaN Values
    c_12_25 = polyfit(efficient_wob_12_25, efficient_rop_12_25, 1); % Create trendline
    disp(['ROP = ' num2str(c_12_25(1)) '*WOB + ' num2str(c_12_25(2))]) % % Display equation ROP = a*WOB +
b
    d_12_25 = polyfit(efficient_wob_12_25, efficient_tqa_12_25,1); % Create trendline WOB-Torque
    disp(['Torque = ' num2str(d_12_25(1)) '*WOB + ' num2str(d_12_25(2))]); % % Display equation Torque =
a*WOB + b

    x= linspace(0,wob_lim); % WOB
    y_12_25 = c_12_25(1)*x + c_12_25(2); % Trendline WOB-ROP
    y2_12_25 = d_12_25(1)*x + d_12_25(2); % Trendline WOB-Torque

    figure(14)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 12.25" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_12_25), max(mdrkb_12_25)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_12_25, rop_12_25, 20, mdrkb_12_25, 'filled')
    title('12.25" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_12_25 = plot(x,y_12_25, '--r','linewidth', 2);
    legend(h_12_25, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_12_25, eff_rop_12_25, 20, 'filled')
    hold on
    scatter(ineff_wob_12_25, ineff_rop_12_25, 20, 'filled')
    plot(x,y_12_25, '--r','linewidth', 2);
    hold off
    title('12.25" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on

    nameofplot = 'wob_rop_depth_bitsize_12_25_events_NPT';
    fig1 = figure(14);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);


    figure(14543)
    subplot(1,3,1)
```

```
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 12.25" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_12_25), max(mdrkb_12_25)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_12_25, tqa_12_25, 20, mdrkb_12_25, 'filled')
    title('12.25" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_12_25 = plot(x,y2_12_25, '--r','linewidth', 2);
    legend(h_12_25, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_12_25, eff_tqa_12_25, 20, 'filled')
    hold on
    scatter(ineff_wob_12_25, ineff_tqa_12_25, 20, 'filled')
    plot(x,y2_12_25, '--r','linewidth', 2);
    hold off
    title('12.25" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on

    nameofplot = 'wob_torque_depth_bitsize_12_25_events_NPT';
    fig1 = figure(14543);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 12 1/4" hole section do not exist');
end
```

*17 1/2" Hole Section*

```
% If 17 1/2" hole section exist, calculate efficient bit line and plot
if sum(bitsize_17_5)>0
    k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
           efficient_rop_17_5(k) = rop_17_5(i);
           efficient_wob_17_5(k) = wob_17_5(i);
           efficient_tqa_17_5(k) = tqa_17_5(i);
           eff_rop_17_5 (k) = rop_17_5(i);
           eff_wob_17_5 (k) = wob_17_5(i);
           eff_tqa_17_5(k) = tqa_17_5(i);
        else
            ineff_rop_17_5(k) = rop_17_5(i);
            ineff_wob_17_5(k) = wob_17_5(i);
```

```
            ineff_tqa_17_5(k) = tqa_17_5(i);
        end
    end

    efficient_rop_17_5(isnan(efficient_rop_17_5)) = []; % Remove NaN values
    efficient_wob_17_5(isnan(efficient_wob_17_5)) = []; % Remove NaN values
    efficient_tqa_17_5(isnan(efficient_tqa_17_5)) = []; % Remove NaN Values

    c_17_5 = polyfit(efficient_wob_17_5, efficient_rop_17_5, 1); % Create trendline
    disp(['ROP = ' num2str(c_17_5(1)) '*WOB + ' num2str(c_17_5(2))]) % % Display equation in the form y =
a*x + b
    d_17_5 = polyfit(efficient_wob_17_5, efficient_tqa_17_5,1); % Create trendline WOB-Torque
    disp(['Torque = ' num2str(d_17_5(1)) '*WOB + ' num2str(d_17_5(2))]); % % Display equation Torque =
a*WOB + b

    x= linspace(0,25); % WOB
    y_17_5 = c_17_5(1)*x + c_17_5(2); % Trendline
    y2_17_5 = d_17_5(1)*x + d_17_5(2); % Trendline WOB-Torque


    figure(15)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 17.5" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_17_5), max(mdrkb_17_5)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_17_5, rop_17_5, 20, mdrkb_17_5, 'filled')
    title('17.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_17_5 = plot(x,y_17_5, '--r','linewidth', 2);
    legend(h_17_5, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_17_5, eff_rop_17_5, 20, 'filled')
    hold on
    scatter(ineff_wob_17_5, ineff_rop_17_5, 20, 'filled')
    plot(x,y_17_5, '--r','linewidth', 2);
    hold off
    title('17.5" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on

    nameofplot = 'wob_rop_depth_bitsize_17_5_events_NPT_';
    fig1 = figure(15);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
```

```
    print(gcf, '-dpdf', '-fillpage', txtconcat);

    figure(155432)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 17.5" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_17_5), max(mdrkb_17_5)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_17_5, tqa_17_5, 20, mdrkb_17_5, 'filled')
    title('17.5" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_17_5 = plot(x,y2_17_5, '--r','linewidth', 2);
    legend(h_17_5, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_17_5, eff_tqa_17_5, 20, 'filled')
    hold on
    scatter(ineff_wob_17_5, ineff_tqa_17_5, 20, 'filled')
    plot(x,y2_17_5, '--r','linewidth', 2);
    hold off
    title('17.5" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on

    nameofplot = 'wob_tqa_depth_bitsize_17_5_events_NPT_';
    fig1 = figure(155432);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 17 1/2" hole section do not exist');
end
```

*26" Hole Section*

```
% If 26" hole section exist, calculate efficient bit line and plot
if sum(bitsize_26)>0;
    k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
           efficient_rop_26(k) = rop_26(i);
           efficient_wob_26(k) = wob_26(i);
           efficient_tqa_26(k) = tqa_26(i);
           eff_rop_26 (k) = rop_26(i);
           eff_wob_26 (k) = wob_26(i);
```

```
            eff_tqa_26(k) = tqa_26(i);
        else
            ineff_rop_26(k) = rop_26(i);
            ineff_wob_26(k) = wob_26(i);
            ineff_tqa_26(k) = tqa_26(i);
        end
    end

    efficient_rop_26(isnan(efficient_rop_26)) = []; % Remove NaN values
    efficient_wob_26(isnan(efficient_wob_26)) = []; % Remove NaN values
    efficient_tqa_26(isnan(efficient_tqa_26)) = []; % Remove NaN Values

    c_26 = polyfit(efficient_wob_26, efficient_rop_26, 1); % Create trendline
    disp(['ROP = ' num2str(c_26(1)) '*WOB + ' num2str(c_26(2))]) % % Display equation ROP=a*WOB + b
    d_26 = polyfit(efficient_wob_26, efficient_tqa_26,1); % Create trendline WOB-Torque
    disp(['Torque = ' num2str(d_26(1)) '*WOB + ' num2str(d_26(2))]); % % Display equation Torque = a*WOB
+ b

    x= linspace(0,25); % WOB
    y_26 = c_26(1)*x + c_26(2); % Trendline
    y2_26 = d_26(1)*x + d_26(2); % Trendline WOB-Torque


    figure(16)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 26" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_26) max(mdrkb_26)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_26, rop_26, 20, mdrkb_26, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_26 = plot(x,y_26, '--r','linewidth', 2);
    legend(h_26, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_26, eff_rop_26, 20, 'filled')
    hold on
    scatter(ineff_wob_26, ineff_rop_26, 20, 'filled')
    plot(x,y_26, '--r','linewidth', 2);
    hold off
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on

    nameofplot = 'wob_rop_depth_bitsize_26_events_NPT_';
    fig1 = figure(16);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
```

```
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);


    figure(165435)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 26" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_26) max(mdrkb_26)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_26, tqa_26, 20, mdrkb_26, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_26 = plot(x,y2_26, '--r','linewidth', 2);
    legend(h_26, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_26, eff_tqa_26, 20, 'filled')
    hold on
    scatter(ineff_wob_26, ineff_tqa_26, 20, 'filled')
    plot(x,y2_26, '--r','linewidth', 2);
    hold off
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on

    nameofplot = 'wob_torque_depth_bitsize_26_events_NPT_';
    fig1 = figure(165435);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 26" hole section do not exist');
end
%
```

36" Hole Section

```
% If 36" hole section exist, calculate efficient bit line and plot
if sum(bitsize_36)>0;
    k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
```

```
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
           efficient_rop_36(k) = rop_36(i);
           efficient_wob_36(k) = wob_36(i);
           efficient_tqa_36(k) = tqa_36(i);
           eff_rop_36 (k) = rop_36(i);
           eff_wob_36 (k) = wob_36(i);
           eff_tqa_36(k) = tqa_36(i);
        else
            ineff_rop_36(k) = rop_36(i);
            ineff_wob_36(k) = wob_36(i);
            ineff_tqa_36(k) = tqa_36(i);
        end
    end

    efficient_rop_36(isnan(efficient_rop_36)) = []; % Remove NaN values
    efficient_wob_36(isnan(efficient_wob_36)) = []; % Remove NaN values
    efficient_tqa_36(isnan(efficient_tqa_36)) = []; % Remove NaN Values

    c_36 = polyfit(efficient_wob_36, efficient_rop_36, 1); % Create trendline
    disp(['ROP = ' num2str(c_36(1)) '*WOB + ' num2str(c_36(2))]) % % Display equation ROP = a*WOB + b
    d_36 = polyfit(efficient_wob_36, efficient_tqa_36,1); % Create trendline WOB-Torque
    disp(['Torque = ' num2str(d_36(1)) '*WOB + ' num2str(d_36(2))]); % % Display equation Torque = a*WOB
+ b

    x= linspace(0,25); % WOB
    y_36 = c_36(1)*x + c_36(2); % Trendline
    y2_36 = d_36(1)*x + d_36(2); % Trendline WOB-Torque



    figure(16)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 36" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_36) max(mdrkb_36)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_36, rop_36, 20, mdrkb_36, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_36 = plot(x,y_36, '--r','linewidth', 2);
    legend(h_36, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_36, eff_rop_36, 20, 'filled')
    hold on
    scatter(ineff_wob_36, ineff_rop_36, 20, 'filled')
    plot(x,y_36, '--r','linewidth', 2);
    hold off
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
```

```
    box on

    nameofplot = 'wob_rop_depth_bitsize_36_events_NPT';
    fig1 = figure(99);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);

    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);



    figure(165432)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 36" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_36) max(mdrkb_36)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_36, tqa_36, 20, mdrkb_36, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_36 = plot(x,y2_36, '--r','linewidth', 2);
    legend(h_36, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_36, eff_tqa_36, 20, 'filled')
    hold on
    scatter(ineff_wob_36, ineff_tqa_36, 20, 'filled')
    plot(x,y2_36, '--r','linewidth', 2);
    hold off
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 tqa_lim])
    grid on
    box on

    nameofplot = 'wob_tqa_depth_bitsize_36_events_NPT';
    fig1 = figure(165432);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);

    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 36" hole section do not exist');
end
```

9 7/8" Section

```
% If 9 7/8" hole section exist, calculate efficient bit line and plot
if sum(bitsize_9_88)>0;
    k = 0;
    for i = 1:length(mdrkb)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
           efficient_rop_9_88(k) = rop_9_88(i);
           efficient_wob_9_88(k) = wob_9_88(i);
           efficient_tqa_9_88(k) = tqa_9_88(i);
           eff_rop_9_88 (k) = rop_9_88(i);
           eff_wob_9_88 (k) = wob_9_88(i);
           eff_tqa_9_88(k) = tqa_9_88(i);
        else
            ineff_rop_9_88(k) = rop_9_88(i);
            ineff_wob_9_88(k) = wob_9_88(i);
            ineff_tqa_9_88(k) = tqa_9_88(i);
        end
    end

    efficient_rop_9_88(isnan(efficient_rop_9_88)) = []; % Remove NaN values
    efficient_wob_9_88(isnan(efficient_wob_9_88)) = []; % Remove NaN values
    efficient_tqa_9_88(isnan(efficient_tqa_9_88)) = []; % Remove NaN Values

    c_9_88 = polyfit(efficient_wob_9_88, efficient_rop_9_88, 1); % Create trendline for WOB-ROP
    disp(['ROP = ' num2str(c_9_88(1)) '*WOB + ' num2str(c_9_88(2))]) % % Display equation ROP = a*WOB + b
    d_9_88 = polyfit(efficient_wob_9_88, efficient_tqa_9_88,1); % Create trendline WOB-Torque
    disp(['Torque = ' num2str(d_9_88(1)) '*WOB + ' num2str(d_9_88(2))]); % % Display equation Torque =
a*WOB + b

    x= linspace(0, wob_lim); % WOB
    y_9_88 = c_9_88(1)*x + c_9_88(2); % Trendline
    y2_9_88 = d_17_5(1)*x + d_9_88(2); % Trendline WOB-Torque


    figure(1654)
    subplot(1,3,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 9 7/8" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_9_88) max(mdrkb_9_88)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    subplot(1,3,2)
    scatter(wob_9_88, rop_9_88, 20, mdrkb_9_88, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_9_88 = plot(x,y_9_88, '--r','linewidth', 2);
    legend(h_9_88, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,3,3)
    scatter(eff_wob_9_88, eff_rop_9_88, 20, 'filled')
    hold on
    scatter(ineff_wob_9_88, ineff_rop_9_88, 20, 'filled')
    plot(x,y_9_88, '--r','linewidth', 2);
    hold off
```

```
title('9 7/8" Section')
xlabel('WOB [ton]')
ylabel('ROP [m/hr]')
legend('Efficient', 'Inefficient', 'Efficient Bit Line')
axis([0 wob_lim 0 rop_lim])
grid on
box on

nameofplot = 'wob_rop_depth_bitsize_9_88_events_NPT';
fig1 = figure(1654);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);
h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);



figure(16554)
subplot(1,3,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 9 7/8" Section')
xlabel('')
ylabel('Depth [mMD]')
ylim([min(mdrkb_9_88) max(mdrkb_9_88)])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

subplot(1,3,2)
scatter(wob_9_88, tqa_9_88, 20, mdrkb_9_88, 'filled')
title('9 7/8" Section')
xlabel('WOB [ton]')
ylabel('Torque [KNm]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth meter MD');
set( depth, 'YDir', 'reverse' );
axis([0 wob_lim 0 tqa_lim])
hold on
h_9_88 = plot(x,y2_9_88, '--r','linewidth', 2);
legend(h_9_88, 'Efficient Bit')
hold off
grid on
box on

subplot(1,3,3)
scatter(eff_wob_9_88, eff_tqa_9_88, 20, 'filled')
hold on
scatter(ineff_wob_9_88, ineff_tqa_9_88, 20, 'filled')
plot(x,y2_9_88, '--r','linewidth', 2);
hold off
title('9 7/8" Section')
xlabel('WOB [ton]')
ylabel('Torque [KNm]')
legend('Efficient', 'Inefficient', 'Efficient Bit Line')
axis([0 wob_lim 0 tqa_lim])
grid on
box on

nameofplot = 'wob_tqa_depth_bitsize_9_88_events_NPT';
fig1 = figure(16554);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);
h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
```

```
    print(gcf, '-dpdf', '-fillpage', txtconcat);
else
    disp('The 9 7/8" hole section do not exist');
end
```

Plot hole sections together

```
% Plot drilling events and NPT and WOB/ROP plot for each hole section in
% one plot including efficient bit line
figure(8876)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

p=1; % Subplot number
if sum(bitsize_9_88)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_9_88, rop_9_88, 20, mdrkb, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_9_88 = plot(x,y_9_88, '--r','linewidth', 2);
    %legend(h_9_88, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_36)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_36, rop_36, 20, mdrkb, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_36 = plot(x,y_36, '--r','linewidth', 2);
    %legend(h_36, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_26)>0
    p=p+1;
```

```
    subplot(1,(bits+1),p)
    scatter(wob_26, rop_26, 20, mdrkb, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_26 = plot(x,y_26, '--r','linewidth', 2);
    %legend(h_26, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_17_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_17_5, rop_17_5, 20, mdrkb, 'filled')
    title('17 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_17_5 = plot(x,y_17_5, '--r','linewidth', 2);
    %legend(h_17_5, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_12_25)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_12_25, rop_12_25, 20, mdrkb, 'filled')
    title('12 1/4" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_12_25 = plot(x,y_12_25, '--r','linewidth', 2);
    %legend(h_12_25, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_8_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_8_5, rop_8_5, 20, mdrkb, 'filled')
    title('8 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
```

```
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_8_5 = plot(x,y_8_5, '--r','linewidth', 2);
    %legend(h_8_5, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_6)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_6, rop_6, 20, mdrkb, 'filled')
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_6 = plot(x,y_6, '--r','linewidth', 2);
    %legend(h_6, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

nameofplot = 'eff_wob_rop_depth_bitsize_events_NPT_';
fig1 = figure(8876);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% Plot drilling events and NPT and WOB/ROP plot for each hole section in
% one plot including efficient bit line showing efficient and inefficient
% depths.
figure(54363)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

p=1; % Subplot number
if sum(bitsize_9_88)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_9_88, eff_rop_9_88, 20, 'filled')
    hold on
    scatter(ineff_wob_9_88, ineff_rop_9_88, 20, 'filled')
    plot(x,y_9_88, '--r','linewidth', 2);
    hold off
    title('9 7/8" Section')
```

```
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    %legend('Efficient', 'Inefficient', 'Efficient Bit Line')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_36)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_36, eff_rop_36, 20, 'filled')
    hold on
    scatter(ineff_wob_36, ineff_rop_36, 20, 'filled')
    plot(x,y_36, '--r','linewidth', 2);
    hold off
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_26)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_26, eff_rop_26, 20, 'filled')
    hold on
    scatter(ineff_wob_26, ineff_rop_26, 20, 'filled')
    plot(x,y_26, '--r','linewidth', 2);
    hold off
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_17_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_17_5, eff_rop_17_5, 20, 'filled')
    hold on
    scatter(ineff_wob_17_5, ineff_rop_17_5, 20, 'filled')
    plot(x,y_17_5, '--r','linewidth', 2);
    hold off
    title('17 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_12_25)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_12_25, eff_rop_12_25, 20, 'filled')
    hold on
    scatter(ineff_wob_12_25, ineff_rop_12_25, 20, 'filled')
    plot(x,y_12_25, '--r','linewidth', 2);
    hold off
    title('12 1/4" Section')
```

```
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_8_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_8_5, eff_rop_8_5, 20, 'filled')
    hold on
    scatter(ineff_wob_8_5, ineff_rop_8_5, 20, 'filled')
    plot(x,y_8_5, '--r','linewidth', 2);
    hold off
    title('8 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_6)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(eff_wob_6, eff_rop_6, 20, 'filled')
    hold on
    scatter(ineff_wob_6, ineff_rop_6, 20, 'filled')
    plot(x,y_6, '--r','linewidth', 2);
    hold off
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('ROP [m/hr]')
    axis([0 wob_lim 0 rop_lim])
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

nameofplot = 'eff_noneff_wob_rop_depth_bitsize_events_NPT_';
fig1 = figure(54363);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% Plot drilling events and NPT and WOB-Torque plot for each hole section in
% one plot including efficient bit line
figure(8876)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on
```

```
p=1; % Subplot number
if sum(bitsize_9_88)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_9_88, tqa_9_88, 20, mdrkb, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_9_88 = plot(x,y2_9_88, '--r','linewidth', 2);
    %legend(h_9_88, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_36)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_36, tqa_36, 20, mdrkb, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_36 = plot(x,y2_36, '--r','linewidth', 2);
    %legend(h_36, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_26)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_26, tqa_26, 20, mdrkb, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_26 = plot(x,y2_26, '--r','linewidth', 2);
    %legend(h_26, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_17_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_17_5, tqa_17_5, 20, mdrkb, 'filled')
    title('17 1/2" Section')
    xlabel('WOB [ton]')
```

```
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_17_5 = plot(x,y2_17_5, '--r','linewidth', 2);
    %legend(h_17_5, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_12_25)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_12_25, tqa_12_25, 20, mdrkb, 'filled')
    title('12 1/4" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_12_25 = plot(x,y2_12_25, '--r','linewidth', 2);
    %legend(h_12_25, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_8_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_8_5, tqa_8_5, 20, mdrkb, 'filled')
    title('8 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_8_5 = plot(x,y2_8_5, '--r','linewidth', 2);
    %legend(h_8_5, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_6)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_6, tqa_6, 20, mdrkb, 'filled')
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_6 = plot(x,y2_6, '--r','linewidth', 2);
```

```
    %legend(h_6, 'Efficient Bit')
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

nameofplot = 'eff_wob_torque_depth_bitsize_events_NPT_';
fig1 = figure(8876);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% Plot drilling events and NPT and WOB-Torque plot for each hole section in
% one plot including efficient bit line showing efficient and inefficient
% depths
figure(8876)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

p=1; % Subplot number
if sum(bitsize_9_88)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_9_88, tqa_9_88, 20, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    axis([0 wob_lim 0 tqa_lim])
    hold on
    scatter(ineff_wob_9_88, ineff_tqa_9_88, 20, 'filled')
    h_9_88 = plot(x,y2_9_88, '--r','linewidth', 2);
    hold off
    grid on
    box on
end

if sum(bitsize_36)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_36, tqa_36, 20, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    axis([0 wob_lim 0 tqa_lim])
    hold on
    scatter(ineff_wob_36, ineff_tqa_36, 20, 'filled')
    h_36 = plot(x,y2_36, '--r','linewidth', 2);
    hold off
    grid on
    box on
end

if sum(bitsize_26)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_26, tqa_26, 20, 'filled')
```

```
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    axis([0 wob_lim 0 tqa_lim])
    hold on
    scatter(ineff_wob_26, ineff_tqa_26, 20, 'filled')
    h_26 = plot(x,y2_26, '--r','linewidth', 2);
    hold off
    grid on
    box on
end

if sum(bitsize_17_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_17_5, tqa_17_5, 20, 'filled')
    title('17 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    axis([0 wob_lim 0 tqa_lim])
    hold on
    scatter(ineff_wob_17_5, ineff_tqa_17_5, 20, 'filled')
    h_17_5 = plot(x,y2_17_5, '--r','linewidth', 2);
    hold off
    grid on
    box on
end

if sum(bitsize_12_25)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_12_25, tqa_12_25, 20, 'filled')
    title('12 1/4" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    axis([0 wob_lim 0 tqa_lim])
    hold on
    scatter(ineff_wob_12_25, ineff_tqa_12_25, 20, 'filled')
    h_12_25 = plot(x,y2_12_25, '--r','linewidth', 2);
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_8_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_8_5, tqa_8_5, 20, 'filled')
    title('8 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    axis([0 wob_lim 0 tqa_lim])
    hold on
    scatter(ineff_wob_8_5, ineff_tqa_8_5, 20, 'filled')
    h_8_5 = plot(x,y2_8_5, '--r','linewidth', 2);
    hold off
    grid on
    box on
end

if sum(bitsize_6)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_6, tqa_6, 20, 'filled')
    title('6" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    axis([0 wob_lim 0 tqa_lim])
    hold on
    scatter(ineff_wob_6, ineff_tqa_6, 20, 'filled')
    h_6 = plot(x,y2_6, '--r','linewidth', 2);
    hold off
    grid on
    box on
```

```
end

nameofplot = 'eff_noneff_wob_torque_depth_bitsize_events_NPT_';
fig1 = figure(8876);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% Plot drilling events and NPT and WOB-Torque with ROP plot for each hole section in
% one plot including efficient bit line
figure(8876)
subplot(1,(bits+1),1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

p=1; % Subplot number
if sum(bitsize_9_88)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_9_88, tqa_9_88, 20, rop, 'filled')
    title('9 7/8" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_9_88 = plot(x,y2_9_88, '--r','linewidth', 2);
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_36)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_36, tqa_36, 20, rop, 'filled')
    title('36" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'ROP [m/hr]');
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_36 = plot(x,y2_36, '--r','linewidth', 2);
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_26)>0
    p=p+1;
```

```
    subplot(1,(bits+1),p)
    scatter(wob_26, tqa_26, 20, rop, 'filled')
    title('26" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'ROP [m/hr]');
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_26 = plot(x,y2_26, '--r','linewidth', 2);
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_17_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_17_5, tqa_17_5, 20, rop, 'filled')
    title('17 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'ROP [m/hr]');
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_17_5 = plot(x,y2_17_5, '--r','linewidth', 2);
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_12_25)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_12_25, tqa_12_25, 20, rop, 'filled')
    title('12 1/4" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'ROP [m/hr]');
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_12_25 = plot(x,y2_12_25, '--r','linewidth', 2);
    hold off
    grid on
    box on
    if bits + 1 ~= p
        colorbar off
    end
end

if sum(bitsize_8_5)>0
    p=p+1;
    subplot(1,(bits+1),p)
    scatter(wob_8_5, tqa_8_5, 20, rop, 'filled')
    title('8 1/2" Section')
    xlabel('WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'ROP [m/hr]');
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_8_5 = plot(x,y2_8_5, '--r','linewidth', 2);
    hold off
    grid on
    box on
```

```
        if bits + 1 ~= p
            colorbar off
        end
    end

    if sum(bitsize_6)>0
        p=p+1;
        subplot(1,(bits+1),p)
        scatter(wob_6, tqa_6, 20, rop, 'filled')
        title('6" Section')
        xlabel('WOB [ton]')
        ylabel('Torque [KNm]')
        colorbar
        depth = colorbar;
        ylabel(depth, 'ROP [m/hr]');
        axis([0 wob_lim 0 tqa_lim])
        hold on
        h_6 = plot(x,y2_6, '--r','linewidth', 2);
        hold off
        grid on
        box on
        if bits + 1 ~= p
            colorbar off
        end
    end

    nameofplot = 'eff_wob_torque_rop_bitsize_events_NPT_';
    fig1 = figure(8876);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);

    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);
```

## Calculate Lost Time

6" Hole Section

```
    % If 6" hole section exist, calculate new ROP from efficient bit-line
    % based on drilling events.

    if sum(bitsize_6)>0
        figure(266)
        subplot(1,4,1)
        scatter(bitballing, mdrkb, 20, 'filled', 'r')
        hold on
        scatter(vibration, mdrkb, 20, 'filled', 'b')
        scatter(wear, mdrkb, 20, 'filled', 'c')
        scatter(ucs_event, mdrkb, 20, 'filled', 'm')
        hold off
        set(gca, 'YDir','reverse')
        title('Drilling Events & NPT 6" Section')
        xlabel('')
        ylabel('Depth [mMD]')
        ylim([min(mdrkb_6), max(mdrkb_6)])
        legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
        xlim([0 5])
        grid on
        box on

        % Calculate new WOB from efficient bit-line from WOB-Torque plot for 6" hole section
        k = 0;
        for i = 1:length(mdrkb_6)
            k = k+1;
            if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
                wob_new_6(k) = wob_6(i); % No events flagged, use original ROP
            else
                wob_new_6(k) = (tqa_6(i) - d_6(2))/d_6(1); % Flagged event, calculate new WOB from Efficient
  bit-line WOB-Torque
            end
        end

        % Calculate new ROP from efficient bit-line for 6" hole section
```

```
      k = 0;
      for i = 1:length(mdrkb_6)
          k = k+1;
          if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
              rop_new_6(k) = rop_6(i); % No events flagged, use original ROP
          else
              rop_new_6(k) = c_6(1)*wob_new_6(i) + c_6(2); % Flagged event, calculate ROP from Efficient
bit-line
          end
      end

      % Making sure New Rop is always larger than original ROP
      k=0;
      for i=1:length(mdrkb_6)
         k=k+1;
         if rop_new_6(i) < rop_6(i)
             rop_new_6(k) = rop_6(i);
         end
      end

      subplot(1,4,2)
      plot(rop_new_6, mdrkb)
      hold on
      plot(rop_6, mdrkb)
      hold off
      set(gca, 'YDir','reverse')
      title('6" Section New ROP')
      xlabel('ROP [m/hr]')
      legend('New ROP', 'Original ROP')
      xlim([0 rop_lim])
      ylim([min(mdrkb_6), max(mdrkb_6)])
      ylabel('Depth [mMD]')
      box on
      grid on

      subplot(1,4,3)
      scatter(wob_6, rop_6, 20, mdrkb_6, 'filled')
      title('6" Section')
      xlabel('WOB [ton]')
      ylabel('ROP [m/hr]')
      colorbar
      depth = colorbar;
      ylabel(depth, 'Depth [mMD]');
      set( depth, 'YDir', 'reverse' );
      axis([0 wob_lim 0 rop_lim])
      hold on
      h_6 = plot(x,y_6, '--r','linewidth', 2);
      legend(h_6, 'Efficient Bit')
      hold off
      grid on
      box on

      subplot(1,4,4)
      scatter(wob_new_6, tqa_6, 20, rop_new_6, 'filled')
      title('8.5" Section')
      xlabel('New WOB [ton]')
      ylabel('Torque [KNm]')
      colorbar
      depth = colorbar;
      ylabel(depth, 'New ROP [m/hr]');
      axis([0 wob_lim 0 tqa_lim])
      hold on
      h_6 = plot(x,y2_6, '--r','linewidth', 2);
      legend(h_6, 'Efficient Bit')
      hold off
      grid on
      box on

      nameofplot = 'New_ROP_6_';
      fig1 = figure(266);
      fig1.Renderer = 'Painters';
      filetype = '.pdf';
      txtconcat = strcat(nameofplot, wellname, filetype);
      h=gcf;
      set(h, 'PaperOrientation', 'landscape');
      set(h, 'Paperposition', [0 0 13 8]);
      print(gcf, '-dpdf', '-fillpage', txtconcat);
```

```
       % Calculate which ROP change is due to which drilling event
       k=0;
       for i = 1:length(mdrkb_6)
          k = k+1;
          if bitballing(k) == 1
              rop_bb_new_6(k) = c_6(1)*wob_new_6(i) + c_6(2);
          else
              rop_bb_new_6(k) = rop_6(i);
          end
          if vibration(k) == 2
              rop_vibration_new_6(k) = c_6(1)*wob_new_6(i) + c_6(2);
          else
              rop_vibration_new_6(k) = rop_6(i);
          end
          if wear(k) == 3
              rop_wear_new_6(k) = c_6(1)*wob_new_6(i) + c_6(2);
          else
              rop_wear_new_6(k) = rop_6(i);
          end
          if ucs_event(k) == 4
              rop_ucs_new_6(k) = c_6(1)*wob_new_6(i) + c_6(2);
          else
              rop_ucs_new_6(k) = rop_6(i);
          end
       end
  else
      disp('The 6" hole section do not exist');
  end
```

8 1/2" Hole Section

```
  % If 8 1/2" hole section exist, calculate new ROP from efficient bit-line
  % based on drilling events.

  if sum(bitsize_8_5)>0
      figure(26)
      subplot(1,4,1)
      scatter(bitballing, mdrkb, 20, 'filled', 'r')
      hold on
      scatter(vibration, mdrkb, 20, 'filled', 'b')
      scatter(wear, mdrkb, 20, 'filled', 'c')
      scatter(ucs_event, mdrkb, 20, 'filled', 'm')
      hold off
      set(gca, 'YDir','reverse')
      title('Drilling Events & NPT 8.5" Section')
      xlabel('')
      ylabel('Depth [mMD]')
      ylim([min(mdrkb_8_5), max(mdrkb_8_5)])
      legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
      xlim([0 5])
      grid on
      box on

      % Calculate new WOB from efficient bit-line from WOB-Torque plot for 8.5" hole section
      k = 0;
      for i = 1:length(mdrkb_8_5)
          k = k+1;
          if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
              wob_new_8_5(k) = wob_8_5(i); % No events flagged, use original ROP
          else
              wob_new_8_5(k) = (tqa_8_5(i) - d_8_5(2))/d_8_5(1); % Flagged event, calculate new WOB from
 Efficient bit-line WOB-Torque
          end
      end

      % Calculate new ROP from new WOB using efficient bit-line for 8.5" hole section
      k = 0;
      for i = 1:length(mdrkb_8_5)
          k = k+1;
          if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
              rop_new_8_5(k) = rop_8_5(i); % No events flagged, use original ROP
          else
              rop_new_8_5(k) = c_8_5(1)*wob_new_8_5(i) + c_8_5(2); % Flagged event, calculate new ROP from
 Efficient bit-line
          end
```

```
    end

    % Making sure New Rop is always larger than original ROP
    k=0;
    for i=1:length(mdrkb_8_5)
       k=k+1;
       if rop_new_8_5(i) < rop_8_5(i)
           rop_new_8_5(k) = rop_8_5(i);
       end
    end

    subplot(1,4,2)
    plot(rop_new_8_5, mdrkb)
    hold on
    plot(rop_8_5, mdrkb)
    hold off
    set(gca, 'YDir','reverse')
    title('8.5" Section New ROP')
    xlabel('ROP [m/hr]')
    legend('New ROP', 'Original ROP')
    xlim([0 rop_lim])
    ylim([min(mdrkb_8_5), max(mdrkb_8_5)])
    ylabel('Depth [mMD]')
    box on
    grid on

    subplot(1,4,3)
    scatter(wob_new_8_5, rop_new_8_5, 20, mdrkb_8_5, 'filled')
    title('8.5" Section')
    xlabel('New WOB [ton]')
    ylabel('New ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_8_5 = plot(x,y_8_5, '--r','linewidth', 2);
    legend(h_8_5, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,4,4)
    scatter(wob_new_8_5, tqa_8_5, 20, rop_new_8_5, 'filled')
    title('8.5" Section')
    xlabel('New WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'New ROP [m/hr]');
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_8_5 = plot(x,y2_8_5, '--r','linewidth', 2);
    legend(h_8_5, 'Efficient Bit')
    hold off
    grid on
    box on


    nameofplot = 'New_ROP_8_5_';
    fig1 = figure(26);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

    % Calculate which ROP change is due to which drilling event
    k=0;
     for i = 1:length(mdrkb_8_5)
        k = k+1;
        if bitballing(k) == 1
            rop_bb_new_8_5(k) = c_8_5(1)*wob_new_8_5(i) + c_8_5(2);
        else
```

```
            rop_bb_new_8_5(k) = rop_8_5(i);
        end
        if vibration(k) == 2
            rop_vibration_new_8_5(k) = c_8_5(1)*wob_new_8_5(i) + c_8_5(2);
        else
            rop_vibration_new_8_5(k) = rop_8_5(i);
        end
        if wear(k) == 3
            rop_wear_new_8_5(k) = c_8_5(1)*wob_new_8_5(i) + c_8_5(2);
        else
            rop_wear_new_8_5(k) = rop_8_5(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_8_5(k) = c_8_5(1)*wob_new_8_5(i) + c_8_5(2);
        else
            rop_ucs_new_8_5(k) = rop_8_5(i);
        end
     end
else
    disp('The 8 1/2" hole section do not exist');
end
```

12 1/4" Hole section

```
% If 12 1/4" hole section exist, calculate new ROP from efficient bit-line
% based on drilling events.

if sum(bitsize_12_25)>0
    figure(28)
    subplot(1,4,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 12.25" Section')
    xlabel('')
    ylabel('Depth [mMD]')
    ylim([min(mdrkb_12_25), max(mdrkb_12_25)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    % Calculate new WOB from efficient bit-line from WOB-Torque plot for 12.25" hole section
    k = 0;
    for i = 1:length(mdrkb_12_25)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            wob_new_12_25(k) = wob_12_25(i); % No events flagged, use original ROP
        else
            wob_new_12_25(k) = (tqa_12_25(i) - d_12_25(2))/d_12_25(1); % Flagged event, calculate new WOB
from Efficient bit-line WOB-Torque
        end
    end

    % Calculate new ROP from efficient bit-line for 12.25" hole section
    k = 0;
    for i = 1:length(mdrkb_12_25)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            rop_new_12_25(k) = rop_12_25(i); % No events flagged, use original ROP
        else
            rop_new_12_25(k) = c_12_25(1)*wob_new_12_25(i) + c_12_25(2); % Flagged event, calculate ROP
from Efficient bit-line
        end
    end

    % Making sure New Rop is always larger than original ROP
    k=0;
    for i=1:length(mdrkb_12_25)
       k=k+1;
       if rop_new_12_25(i) < rop_12_25(i);
```

```
            rop_new_12_25(k) = rop_12_25(i);
        end
    end

    subplot(1,4,2)
    plot(rop_new_12_25, mdrkb)
    hold on
    plot(rop_12_25, mdrkb)
    hold off
    set(gca, 'YDir','reverse')
    title('12.25" Section New ROP')
    xlabel('ROP [m/hr]')
    ylim([min(mdrkb_12_25), max(mdrkb_12_25)])
    xlim([0 rop_lim])
    ylabel('Depth [mMD]')
    legend('New ROP', 'Original ROP')
    grid on
    box on

    subplot(1,4,3)
    scatter(wob_new_12_25, rop_new_12_25, 20, mdrkb_12_25, 'filled')
    title('12.25" Section')
    xlabel('New WOB [ton]')
    ylabel('New ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth [mMD]');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_12_25 = plot(x,y_12_25, '--r','linewidth', 2);
    legend(h_12_25, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,4,4)
    scatter(wob_new_12_25, tqa_12_25, 20, rop_new_12_25, 'filled')
    title('12.25" Section')
    xlabel('New WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'New ROP [m/hr]');
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_12_25 = plot(x,y2_12_25, '--r','linewidth', 2);
    legend(h_12_25, 'Efficient Bit')
    hold off
    grid on
    box on

    nameofplot = 'New_ROP_12_25_';
    fig1 = figure(28);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

     % Calculate which ROP change is due to which drilling event
   k=0;
     for i = 1:length(mdrkb_12_25)
        k = k+1;
        if bitballing(k) == 1
            rop_bb_new_12_25(k) = c_12_25(1)*wob_new_12_25(i) + c_12_25(2);
        else
            rop_bb_new_12_25(k) = rop_12_25(i);
        end
        if vibration(k) == 2
            rop_vibration_new_12_25(k) = c_12_25(1)*wob_new_12_25(i) + c_12_25(2);
        else
            rop_vibration_new_12_25(k) = rop_12_25(i);
        end
        if wear(k) == 3
```

```
            rop_wear_new_12_25(k) = c_12_25(1)*wob_new_12_25(i) + c_12_25(2);
        else
            rop_wear_new_12_25(k) = rop_12_25(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_12_25(k) = c_12_25(1)*wob_new_12_25(i) + c_12_25(2);
        else
            rop_ucs_new_12_25(k) = rop_12_25(i);
        end
     end
else
    disp('The 12 1/4" hole section do not exist');
end
```

17 1/2" Hole Section

```
% If 17 1/2" hole section exist, calculate new ROP from efficient bit-line
% based on drilling events.

if sum(bitsize_17_5)>0
    figure(30)
    subplot(1,4,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 17.5" Section')
    xlabel('')
    ylabel('Depth [meter MD]')
    ylim([min(mdrkb_17_5), max(mdrkb_17_5)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    % Calculate new WOB from efficient bit-line from WOB-Torque plot for 17.5" hole section
    k = 0;
    for i = 1:length(mdrkb_17_5)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            wob_new_17_5(k) = wob_17_5(i); % No events flagged, use original ROP
        else
            wob_new_17_5(k) = (tqa_17_5(i) - d_17_5(2))/d_17_5(1); % Flagged event, calculate new WOB
from Efficient bit-line WOB-Torque
        end
    end

    % Calculate new ROP from efficient bit-line for 17.5" hole section
    k = 0;
    for i = 1:length(mdrkb_17_5)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            rop_new_17_5(k) = rop_17_5(i); % No events flagged, use original ROP
        else
            rop_new_17_5(k) = c_17_5(1)*wob_new_17_5(i) + c_17_5(2); % Flagged event, calculate ROP from
Efficient bit-line
        end
    end

    % Making sure New Rop is always larger than original ROP
    k=0;
    for i=1:length(mdrkb_17_5)
       k=k+1;
       if rop_new_17_5(i) < rop_17_5(i);
          rop_new_17_5(k) = rop_17_5(i);
       end
    end

    subplot(1,4,2)
    plot(rop_new_17_5, mdrkb)
    hold on
    plot(rop_17_5, mdrkb)
    hold off
```

```
set(gca, 'YDir','reverse')
title('17.5" Section New ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
legend('New ROP', 'Original ROP')
ylim([min(mdrkb_17_5), max(mdrkb_17_5)])
xlim([0 rop_lim])
grid on
box on

subplot(1,4,3)
scatter(wob_new_17_5, rop_new_17_5, 20, mdrkb_17_5, 'filled')
title('17.5" Section')
xlabel('New WOB [ton]')
ylabel('New ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth meter MD');
set( depth, 'YDir', 'reverse' );
axis([0 wob_lim 0 rop_lim])
hold on
h_17_5 = plot(x,y_17_5, '--r','linewidth', 2);
legend(h_17_5, 'Efficient Bit')
hold off
grid on
box on

subplot(1,4,4)
scatter(wob_new_17_5, tqa_17_5, 20, rop_new_17_5, 'filled')
title('17.5" Section')
xlabel('New WOB [ton]')
ylabel('Torque [KNm]')
colorbar
depth = colorbar;
ylabel(depth, 'New ROP [m/hr]');
axis([0 wob_lim 0 tqa_lim])
hold on
h_17_5 = plot(x,y2_17_5, '--r','linewidth', 2);
legend(h_17_5, 'Efficient Bit')
hold off
grid on
box on

nameofplot = 'New_ROP_17_5_';
fig1 = figure(30);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);
h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);

 % Calculate which ROP change is due to which drilling event
k=0;
 for i = 1:length(mdrkb_17_5)
    k = k+1;
    if bitballing(k) == 1
        rop_bb_new_17_5(k) = c_17_5(1)*wob_new_17_5(i) + c_17_5(2);
    else
        rop_bb_new_17_5(k) = rop_17_5(i);
    end
    if vibration(k) == 2
        rop_vibration_new_17_5(k) = c_17_5(1)*wob_new_17_5(i) + c_17_5(2);
    else
        rop_vibration_new_17_5(k) = rop_17_5(i);
    end
    if wear(k) == 3
        rop_wear_new_17_5(k) = c_17_5(1)*wob_new_17_5(i) + c_17_5(2);
    else
        rop_wear_new_17_5(k) = rop_17_5(i);
    end
    if ucs_event(k) == 4
        rop_ucs_new_17_5(k) = c_17_5(1)*wob_new_17_5(i) + c_17_5(2);
    else
        rop_ucs_new_17_5(k) = rop_17_5(i);
    end
```

```
        end
  else
      disp('The 17 1/2" hole section do not exist');
  end
```

26" Hole Section

```
  % If 26" hole section exist, calculate new ROP from efficient bit-line
  % based on drilling events.

  if sum(bitsize_26)>0
      figure(32)
      subplot(1,4,1)
      scatter(bitballing, mdrkb, 20, 'filled', 'r')
      hold on
      scatter(vibration, mdrkb, 20, 'filled', 'b')
      scatter(wear, mdrkb, 20, 'filled', 'c')
      scatter(ucs_event, mdrkb, 20, 'filled', 'm')
      hold off
      set(gca, 'YDir','reverse')
      title('Drilling Events & NPT 26" Section')
      xlabel('')
      ylabel('Depth [meter MD]')
      ylim([min(mdrkb_26), max(mdrkb_26)])
      legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
      xlim([0 5])
      grid on
      box on

      % Calculate new WOB from efficient bit-line from WOB-Torque plot for 26" hole section
      k = 0;
      for i = 1:length(mdrkb_26)
          k = k+1;
          if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
              wob_new_26(k) = wob_26(i); % No events flagged, use original ROP
          else
              wob_new_26(k) = (tqa_26(i) - d_26(2))/d_26(1); % Flagged event, calculate new WOB from
 Efficient bit-line WOB-Torque
          end
      end

      % Calculate new ROP from efficient bit-line for 26" hole section

      k = 0;
      for i = 1:length(mdrkb_26)
          k = k+1;
          if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
              rop_new_26(k) = rop_26(i); % No events flagged, use original ROP
          else
              rop_new_26(k) = c_26(1)*wob_new_26(i) + c_26(2); % Flagged event, calculate ROP from
 Efficient bit-line
          end
      end

      % Making sure New Rop is always larger than original ROP
      k=0;
      for i=1:length(mdrkb_26)
         k=k+1;
         if rop_new_26(i) < rop_26(i)
             rop_new_26(k) = rop_26(i);
         end
      end

      subplot(1,4,2)
      plot(rop_new_26, mdrkb)
      hold on
      plot(rop_26, mdrkb)
      hold off
      set(gca, 'YDir','reverse')
      title('26" Section New ROP')
      xlabel('ROP [m/hr]')
      ylabel('Depth [mMD]')
      legend('New ROP', 'Original ROP')
      xlim([0 rop_lim])
      ylim([min(mdrkb_26), max(mdrkb_26)])
      grid on
```

```
    box on

    subplot(1,4,3)
    scatter(wob_new_26, rop_new_26, 20, mdrkb_26, 'filled')
    title('26" Section')
    xlabel('New WOB [ton]')
    ylabel('New ROP [m/hr]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_26 = plot(x,y_26, '--r','linewidth', 2);
    legend(h_26, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,4,4)
    scatter(wob_new_26, tqa_26, 20, rop_new_26, 'filled')
    title('26" Section')
    xlabel('New WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'New ROP [m/hr]');
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_26 = plot(x,y2_26, '--r','linewidth', 2);
    legend(h_26, 'Efficient Bit')
    hold off
    grid on
    box on

    nameofplot = 'New_ROP_26_';
    fig1 = figure(32);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

     % Calculate which ROP change is due to which drilling event
   k=0;
     for i = 1:length(mdrkb_26)
        k = k+1;
        if bitballing(k) == 1
            rop_bb_new_26(k) = c_26(1)*wob_new_26(i) + c_26(2);
        else
            rop_bb_new_26(k) = rop_26(i);
        end
        if vibration(k) == 2
            rop_vibration_new_26(k) = c_26(1)*wob_new_26(i) + c_26(2);
        else
            rop_vibration_new_26(k) = rop_26(i);
        end
        if wear(k) == 3
            rop_wear_new_26(k) = c_26(1)*wob_new_26(i) + c_26(2);
        else
            rop_wear_new_26(k) = rop_26(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_26(k) = c_26(1)*wob_new_26(i) + c_26(2);
        else
            rop_ucs_new_26(k) = rop_26(i);
        end
     end
else
    disp('The 26" hole section do not exist');
end
```

36" Hole Section

```
% If 36" hole section exist, calculate new ROP from efficient bit-line
% based on drilling events.

if sum(bitsize_36)>0
    figure(33)
    subplot(1,4,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
    hold on
    scatter(vibration, mdrkb, 20, 'filled', 'b')
    scatter(wear, mdrkb, 20, 'filled', 'c')
    scatter(ucs_event, mdrkb, 20, 'filled', 'm')
    hold off
    set(gca, 'YDir','reverse')
    title('Drilling Events & NPT 36" Section')
    xlabel('')
    ylabel('Depth [meter MD]')
    ylim([min(mdrkb_36), max(mdrkb_36)])
    legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
    xlim([0 5])
    grid on
    box on

    % Calculate new WOB from efficient bit-line from WOB-Torque plot for 36" hole section
    k = 0;
    for i = 1:length(mdrkb_36)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            wob_new_36(k) = wob_36(i); % No events flagged, use original ROP
        else
            wob_new_36(k) = (tqa_36(i) - d_36(2))/d_36(1); % Flagged event, calculate new WOB from
Efficient bit-line WOB-Torque
        end
    end

    % Calculate new ROP from efficient bit-line for 36" hole section
    k = 0;
    for i = 1:length(mdrkb_36)
        k = k+1;
        if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
            rop_new_36(k) = rop_36(i); % No events flagged, use original ROP
        else
            rop_new_36(k) = c_36(1)*wob_new_36(i) + c_36(2); % Flagged event, calculate ROP from
Efficient bit-line
        end
    end

    % Making sure New Rop is always larger than original ROP
    k=0;
    for i=1:length(mdrkb_36)
       k=k+1;
       if rop_new_36(i) < rop_36(i)
           rop_new_36(k) = rop_36(i);
       end
    end

    subplot(1,4,2)
    plot(rop_new_36, mdrkb)
    hold on
    plot(rop_36, mdrkb)
    hold off
    set(gca, 'YDir','reverse')
    title('36" Section New ROP')
    xlabel('ROP [m/hr]')
    ylabel('Depth [mMD]')
    legend('New ROP', 'Original ROP')
    ylim([min(mdrkb_36), max(mdrkb_36)])
    xlim([0 rop_lim])
    grid on
    box on

    subplot(1,4,3)
    scatter(wob_new_36, rop_new_36, 20, mdrkb_36, 'filled')
    title('36" Section')
    xlabel('New WOB [ton]')
    ylabel('New ROP [m/hr]')
```

```
    colorbar
    depth = colorbar;
    ylabel(depth, 'Depth meter MD');
    set( depth, 'YDir', 'reverse' );
    axis([0 wob_lim 0 rop_lim])
    hold on
    h_36 = plot(x,y_36, '--r','linewidth', 2);
    legend(h_36, 'Efficient Bit')
    hold off
    grid on
    box on

    subplot(1,4,4)
    scatter(wob_new_36, tqa_36, 20, rop_new_36, 'filled')
    title('36" Section')
    xlabel('New WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'New ROP [m/hr]');
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_36 = plot(x,y2_36, '--r','linewidth', 2);
    legend(h_36, 'Efficient Bit')
    hold off
    grid on
    box on

    nameofplot = 'New_ROP_36_';
    fig1 = figure(33);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

     % Calculate which ROP change is due to which drilling event
    k=0;
     for i = 1:length(mdrkb_36)
        k = k+1;
        if bitballing(k) == 1
            rop_bb_new_36(k) = c_36(1)*wob_new_36(i) + c_36(2);
        else
            rop_bb_new_36(k) = rop_36(i);
        end
        if vibration(k) == 2
            rop_vibration_new_36(k) = c_36(1)*wob_new_36(i) + c_36(2);
        else
            rop_vibration_new_36(k) = rop_36(i);
        end
        if wear(k) == 3
            rop_wear_new_36(k) = c_36(1)*wob_new_36(i) + c_36(2);
        else
            rop_wear_new_36(k) = rop_36(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_36(k) = c_36(1)*wob_new_36(i) + c_36(2);
        else
            rop_ucs_new_36(k) = rop_36(i);
        end
     end
else
    disp('The 36" hole section do not exist');
end
```

9 7/8" Hole Section

```
% If 9 7/8" hole section exist, calculate new ROP from efficient bit-line
% based on drilling events.

if sum(bitsize_9_88)>0
    figure(333)
    subplot(1,4,1)
    scatter(bitballing, mdrkb, 20, 'filled', 'r')
```

```
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT 9 7/8" Section')
xlabel('')
ylabel('Depth [meter MD]')
ylim([min(mdrkb_9_88), max(mdrkb_9_88)])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

% Calculate new WOB from efficient bit-line from WOB-Torque plot for 9 7/8" hole section
k = 0;
for i = 1:length(mdrkb_9_88)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
        wob_new_9_88(k) = wob_9_88(i); % No events flagged, use original ROP
    else
        wob_new_9_88(k) = (tqa_9_88(i) - d_9_88(2))/d_9_88(1); % Flagged event, calculate new WOB
from Efficient bit-line WOB-Torque
    end
end

% Calculate new ROP from efficient bit-line for 9 7/8" hole section
k = 0;
for i = 1:length(mdrkb_9_88)
    k = k+1;
    if bitballing(k) == -1 && vibration(k) == -1 && wear(k) == -1 && ucs_event(k) == -1
        rop_new_9_88(k) = rop_9_88(i); % No events flagged, use original ROP
    else
        rop_new_9_88(k) = c_9_88(1)*wob_new_9_88(i) + c_9_88(2); % Flagged event, calculate ROP from
Efficient bit-line
    end
end

% Making sure New Rop is always larger than original ROP
k=0;
for i=1:length(mdrkb_9_88)
   k=k+1;
   if rop_new_9_88(i) < rop_9_88(i)
       rop_new_9_88(k) = rop_9_88(i);
   end
end

subplot(1,4,2)
plot(rop_new_9_88, mdrkb)
hold on
plot(rop_9_88, mdrkb)
hold off
set(gca, 'YDir','reverse')
title('9 7/8" Section New ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
legend('New ROP', 'Original ROP')
ylim([min(mdrkb_9_88), max(mdrkb_9_88)])
xlim([0 rop_lim])
grid on
box on

subplot(1,4,3)
scatter(wob_new_9_88, rop_new_9_88, 20, mdrkb_9_88, 'filled')
title('9 7/8" Section')
xlabel('New WOB [ton]')
ylabel('New ROP [m/hr]')
colorbar
depth = colorbar;
ylabel(depth, 'Depth meter MD');
set( depth, 'YDir', 'reverse' );
axis([0 wob_lim 0 rop_lim])
hold on
h_9_88 = plot(x,y_9_88, '--r','linewidth', 2);
legend(h_9_88, 'Efficient Bit')
hold off
```

```
    grid on
    box on

    subplot(1,4,4)
    scatter(wob_new_9_88, tqa_9_88, 20, rop_new_9_88, 'filled')
    title('9 7/8" Section')
    xlabel('New WOB [ton]')
    ylabel('Torque [KNm]')
    colorbar
    depth = colorbar;
    ylabel(depth, 'New ROP [m/hr]');
    axis([0 wob_lim 0 tqa_lim])
    hold on
    h_9_88 = plot(x,y2_9_88, '--r','linewidth', 2);
    legend(h_9_88, 'Efficient Bit')
    hold off
    grid on
    box on

    nameofplot = 'New_ROP_9_88_';
    fig1 = figure(333);
    fig1.Renderer = 'Painters';
    filetype = '.pdf';
    txtconcat = strcat(nameofplot, wellname, filetype);
    h=gcf;
    set(h, 'PaperOrientation', 'landscape');
    set(h, 'Paperposition', [0 0 13 8]);
    print(gcf, '-dpdf', '-fillpage', txtconcat);

     % Calculate which ROP change is due to which drilling event
    k=0;
     for i = 1:length(mdrkb_9_88)
        k = k+1;
        if bitballing(k) == 1
            rop_bb_new_9_88(k) = c_9_88(1)*wob_new_9_88(i) + c_9_88(2);
        else
            rop_bb_new_9_88(k) = rop_9_88(i);
        end
        if vibration(k) == 2
            rop_vibration_new_9_88(k) = c_9_88(1)*wob_new_9_88(i) + c_9_88(2);
        else
            rop_vibration_new_9_88(k) = rop_9_88(i);
        end
        if wear(k) == 3
            rop_wear_new_9_88(k) = c_9_88(1)*wob_new_9_88(i) + c_9_88(2);
        else
            rop_wear_new_9_88(k) = rop_9_88(i);
        end
        if ucs_event(k) == 4
            rop_ucs_new_9_88(k) = c_9_88(1)*wob_new_9_88(i) + c_9_88(2);
        else
            rop_ucs_new_9_88(k) = rop_9_88(i);
        end
     end
else
    disp('The 9 7/8" hole section do not exist');
end
```

Concatinate ROPs, Calculate new time and plot new ROP

```
% Concatinate the new ROPs
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 9.88
        rop_new(k) = rop_new_9_88(i);
    elseif bdia(k) == 36
        rop_new(k) = rop_new_36(i);
    elseif bdia(k) == 26
        rop_new(k) = rop_new_26(i);
    elseif bdia(k) == 17.5
        rop_new(k) = rop_new_17_5(i);
    elseif bdia(k) == 12.25
        rop_new(k) = rop_new_12_25(i);
    elseif bdia(k) == 8.5
        rop_new(k) = rop_new_8_5(i);
```

```
    elseif bdia(k) == 6
        rop_new(k) = rop_new_6(i);
    else
        rop_new(k) = rop(i);
    end
end

figure(34)
plot(rop_new, mdrkb)
hold on
plot(rop, mdrkb)
hold off
set(gca, 'YDir','reverse')
title(' New ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
legend('New ROP', 'Original ROP')
grid on
box on


% Concatinate the new ROPs for each drilling event
k = 0;
for i = 1:length(mdrkb)
    k = k+1;
    if bdia(k) == 9.88
        rop_new_bb(k) = rop_bb_new_9_88(i);
        rop_new_vibration(k) = rop_vibration_new_9_88(i);
        rop_new_wear(k) = rop_wear_new_9_88(i);
        rop_new_ucs(k) = rop_ucs_new_9_88(i);
    elseif bdia(k) == 36
        rop_new_bb(k) = rop_bb_new_36(i);
        rop_new_vibration(k) = rop_vibration_new_36(i);
        rop_new_wear(k) = rop_wear_new_36(i);
        rop_new_ucs(k) = rop_ucs_new_36(i);
    elseif bdia(k) == 26
        rop_new_bb(k) = rop_bb_new_26(i);
        rop_new_vibration(k) = rop_vibration_new_26(i);
        rop_new_wear(k) = rop_wear_new_26(i);
        rop_new_ucs(k) = rop_ucs_new_26(i);
    elseif bdia(k) == 17.5
        rop_new_bb(k) = rop_bb_new_17_5(i);
        rop_new_vibration(k) = rop_vibration_new_17_5(i);
        rop_new_wear(k) = rop_wear_new_17_5(i);
        rop_new_ucs(k) = rop_ucs_new_17_5(i);
    elseif bdia(k) == 12.25
        rop_new_bb(k) = rop_bb_new_12_25(i);
        rop_new_vibration(k) = rop_vibration_new_12_25(i);
        rop_new_wear(k) = rop_wear_new_12_25(i);
        rop_new_ucs(k) = rop_ucs_new_12_25(i);
    elseif bdia(k) == 8.5
        rop_new_bb(k) = rop_bb_new_8_5(i);
        rop_new_vibration(k) = rop_vibration_new_8_5(i);
        rop_new_wear(k) = rop_wear_new_8_5(i);
        rop_new_ucs(k) = rop_ucs_new_8_5(i);
    elseif bdia(k) == 6
        rop_new_bb(k) = rop_bb_new_6(i);
        rop_new_vibration(k) = rop_vibration_new_6(i);
        rop_new_wear(k) = rop_wear_new_6(i);
        rop_new_ucs(k) = rop_ucs_new_6(i);
    else
        rop_new_bb(k) = rop(i);
        rop_new_vibration(k) = rop(i);
        rop_new_wear(k) = rop(i);
        rop_new_ucs(k) = rop(i);
    end
end

figure(343334)
plot(rop_new_bb, mdrkb)
hold on
plot(rop_new_vibration, mdrkb)
plot(rop_new_wear, mdrkb)
plot(rop_new_ucs, mdrkb)
plot(rop, mdrkb)
hold off
set(gca, 'YDir','reverse')
```

```
title(' New ROP')
xlabel('ROP [m/hr]')
ylabel('Depth [mMD]')
legend('New ROP Bit balling', 'New ROP Vibration','New ROP Wear', 'New ROP Hard Rock', 'Original ROP')
grid on
box on
% Calculate time difference between original and new ROP

k=0;
for i=1:length(mdrkb)
    k=k+1;
    time_orig(k)= depth_interval/rop(i);
    time_n(k) = depth_interval/rop_new(i);
end

time_orig(isinf(time_orig)) = []; % Remove inf values
time_n(isinf(time_n)) = []; % Remove inf values

time_original = sum(time_orig)
time_new = sum(time_n)

% Calculate time difference between original and new ROP for each drilling
% event

k=0;
for i=1:length(mdrkb)
    k=k+1;
    time_bb(k) = depth_interval/rop_new_bb(i);
    time_vibration(k) = depth_interval/rop_new_vibration(i);
    time_wear(k) = depth_interval/rop_new_wear(i);
    time_ucs(k) = depth_interval/rop_new_ucs(i);
end

time_bb(isinf(time_bb)) = []; % Remove inf values
time_vibration(isinf(time_vibration)) = []; % Remove inf values
time_wear(isinf(time_wear)) = []; % Remove inf values
time_ucs(isinf(time_ucs)) = []; % Remove inf values

time_bb_1 = sum(time_bb)
time_vibration_1 = sum(time_vibration)
time_wear_1 = sum(time_wear)
time_ucs_1 = sum(time_ucs)
figure(35)
subplot(1,2,1)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

subplot(1,2,2)
plot(rop_new, mdrkb, 'linewidt', 0.5)
hold on
plot(rop, mdrkb, 'linewidt', 0.5)

num1 = num2str(round(time_original));
num2 = num2str(round(time_new));

   txt1 = ['Original:  '];
   txt2 = ['New: '];
   txt3 = [' hrs'];
   txt4 = [' '];

txt32 = strcat(txt1, txt4, num1, txt4, txt3);
text(110, 570, txt32, 'fontsize', 10,'Interpreter','latex')

txt42 = strcat(txt2, txt4, num2, txt3);
```

```
text(110, 770, txt42, 'fontsize', 10,'Interpreter','latex')

hold off

set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylim([0, TD+200])
xlim([0, 150])
legend('New ROP', 'Original ROP')
grid on
box on

nameofplot = 'time_efficient_drilling_';
fig1 = figure(35);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Bit runs and NPT

```
% Give variable names to bit runs
bit_nr = bit_data(:,1);                      % Bit number
bit_in = bit_data(:,2);                      % Depth in bit, [mMD]
bit_out = bit_data(:,3);                     % Depth out bit, [mMD]
bit_size = bit_data(:,4);                    % Bit size, [in]
bit_core = bit_data(:,7);                    % Coring. Core = 1: No Core. Core = 2: Core.
total_bits = length(bit_nr);                 % Total number of bits

k=0;
for i = 1:length(bit_nr)
    k=k+1;
    if bit_core(k) == 2
        bit_in_core(k) = bit_in(i);
        bit_out_core(k) = bit_out(i);
    else
        bit_in_core(k) = NaN;
        bit_out_core(k) = NaN;
    end
end

% Plot bit-runs and drilling events/NPT
figure(36)
subplot(1,3,1)
scatter(bit_nr, bit_in, 30, 'filled')
hold on
scatter(bit_nr, bit_out, 30, 'filled')
scatter(bit_nr, bit_in_core, 30, 'filled')
scatter(bit_nr, bit_out_core, 30, 'filled')
hold off
set(gca, 'YDir','reverse')
xlim([0 length(bit_nr)+1])
title('Bit Runs')
xlabel('Bit number')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bit In', 'Bit Out', 'Core run in', 'Core run Out')
grid on
box on

subplot(1,3,2)
plot(bdia, mdrkb)
set(gca, 'YDir','reverse')
xlim([0 40])
title('Bit Size')
xlabel('Bit size [in]')
ylim([0, TD+200])
grid on
box on

subplot(1,3,3)
```

```
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
hold off
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
grid on
box on

nameofplot = 'core_bit_drillingevents_NPT_';
fig1 = figure(36);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
% Display number of bit runs
disp(['Number of bit-runs: ' num2str(total_bits) ]);
% Calculate time for bit change
trip_speed = 500; % Tripping speed assuming 500 m/hr

% Look at figure 36 and study which bits are changed due to drilling
% events. Relate bit changes to bit size and drilling events plot.

% Manually insert the bit number to calculate cost of bit change. Other bits are entered as zero.

% 34/4-15 S
%change_bit = [0,0,0,0,0,0,0];

% 34/4-15 A
change_bit=[0,0,0,0,0,0,0,0,0];

% 34/4-16 S
%change_bit=[0,0,0,0,0];

% 33/6-3 S
%change_bit = [0,0,0,0,0,6,7,0];

% 35/3-6
%change_bit = [0,0,0,0,5,6,0,0,0];

%35/8-4
%change_bit = [0,0,0,4,0,0,0,0];

%35/8-5 S
%change_bit = [0,0,0,0,0,6,0,0,0,0,0,0,0,0,0,0];

%35/9-8
%change_bit = [0,0,0,0,0,0,0,0,0,0,0,0,0,0];

%35/9-11 S
%change_bit = [0,0,0,0,0,0];

% 15/12-18 S
%change_bit = [0,0,0,0,0,0,7,0,0];

% 15/12-21
%change_bit = [0,0,0,0,0,0,0,0,0,0,0,0];


 % 15/12-23
% change_bit = [0,0,0,0,0,0,0,0,0];


% Norwegian Sea

% 6406/1-1
%change_bit = [0,0,0,0,5,0,0,0,0,0,0,0,0,0,0,0];
```

```
% 6406/2-3
%change_bit = [0,0,0,0,0,6,0,8,0,0,0,0,0,0,15,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

% 6406/2-5
%change_bit = [0,0,0,0,0];

% 6406/2-7
%change_bit = [0,0,0,0,0];

% 6506/11-6
%change_bit = [0,0,0,0,0,6,0,8,9,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];

% 6407/8-4 S
%change_bit = [0,0,3,0,0,0,0,0,0,0,0];

% 6407/8-5 S
%change_bit = [0,0,0,0];

% 6407/8-6
%change_bit = [0,0,0,0,0,0,0,8,0];

% 6407/8-7
%change_bit = [0,0,0];

% 6407/9-9
%change_bit = [0,0,0,0];



if length(change_bit) ~= length(bit_nr)
    disp(['Error!'])
    disp(['The change bit vector must contain: ' num2str(total_bits) ' numbers'])
    disp(['Current change bit vector contains: ' num2str(length(change_bit)) ' numbers'])
else
    disp('Correct bit change inputs.')
end

k=0;
for i = 1:length(bit_nr)
    k = k+1;
    if change_bit (k) == bit_nr(k);
        depth_bit_change(k) = bit_out(k);
        depth_bit_change_1(k) = bit_out(k);
    else
        depth_bit_change(k) = NaN;
        depth_bit_change_1(k) = NaN;
    end
end

depth_bit_change_1(isnan(depth_bit_change_1)) = []; % Remove NaN values
disp(['Depth of bit change: ' num2str(depth_bit_change_1) ' mMD'])

k = 0;
for i = 1:length(bit_nr)
    k=k+1;
    if depth_bit_change(k) == bit_out(k)
        trip_time_lost(k) = (bit_out(k)*2)/trip_speed;
    else
        trip_time_lost(k) = 0;
    end
end

k = 0;
for i = 1:length(bit_nr)
    k=k+1;
    total_trip_time_1(k) = (bit_out(i)*2)/trip_speed;
end

disp(['Trip times: ' num2str(round(trip_time_lost,1))])
total_trip_time_lost = sum(trip_time_lost);
disp(['Total tripping time lost: ' num2str(round(total_trip_time_lost,1)) ' hours'])
total_trip_time = sum(total_trip_time_1);
disp(['Total tripping time: ' num2str(round(total_trip_time,1)) ' hours'])
```

Calculate total lost time due to inefficient drilling and bit changes

```
% Lost time due to inefficient drilling
rop_lost_time = time_original - time_new;
disp(['Lost time due to inefficient drilling: ' num2str(round(rop_lost_time,1)) ' hours'])

% Lost time due to bit change
disp(['Lost time due to bit change: ' num2str(round(total_trip_time_lost,1)) ' hours'])

% Total lost time
total_lost_time = rop_lost_time + total_trip_time_lost;
disp(['Total lost time: ' num2str(round(total_lost_time,0)) ' hours'])
% Lost time due to each drilling event
rop_lost_time_bb = time_original - time_bb_1;
rop_lost_time_vibration = time_original - time_vibration_1;
rop_lost_time_wear = time_original - time_wear_1;
rop_lost_time_ucs = time_original - time_ucs_1;

disp(['Lost time due to Bit Balling: ' num2str(round(rop_lost_time_bb,0)) ' hours'])
disp(['Lost time due to Vibrations: ' num2str(round(rop_lost_time_vibration,0)) ' hours'])
disp(['Lost time due to Bit wear: ' num2str(round(rop_lost_time_wear,0)) ' hours'])
disp(['Lost time due to Hard Rock: ' num2str(round(rop_lost_time_ucs,0)) ' hours'])
% Plot bit runs, bit size, new rop, drilling events and NPT and lost time

depth_text_1 =  (TD+200)/5;
depth_text_2 = depth_text_1 + 150;

figure(37)
subplot(1,4,1)
scatter(bit_nr, bit_in, 30, 'filled')
hold on
scatter(bit_nr, bit_out, 30, 'filled')
scatter(bit_nr, bit_in_core, 30, 'filled')
scatter(bit_nr, bit_out_core, 30, 'filled')
set(gca, 'YDir','reverse')
xlim([0 length(bit_nr)+1])
title('Bit Runs')
xlabel('Bit number')
ylabel('Depth [mMD]')
ylim([0, TD+200])
legend('Bit In', 'Bit Out', 'Core run in', 'Core run Out')
num1 = num2str(round(total_trip_time_lost,0));
txt1 = ['Lost time:  '];
txt3 = [' hrs'];
txt4 = [' '];
txt32 = strcat(txt1, txt4);
text((length(bit_nr)*(2/3)), depth_text_1, txt32, 'fontsize', 10,'Interpreter','latex')
txt33 = strcat(num1, txt4, txt3);
text((length(bit_nr)*(2/3)), depth_text_2, txt33, 'fontsize', 10,'Interpreter','latex')
hold off
box on
grid on

subplot(1,4,2)
plot(bdia, mdrkb)
xlim([0 40])
title('Bit Size')
xlabel('Bit Size [in]')
ylim([0, TD+200])
set(gca, 'YDir','reverse')
grid on
box on

subplot(1,4,3)
plot(rop_new, mdrkb, 'linewidt', 0.5)
hold on
plot(rop, mdrkb, 'linewidt', 0.5)
set(gca, 'YDir','reverse')
title('ROP')
xlabel('ROP [m/hr]')
ylim([0, TD+200])
xlim([0, 150])
legend('New ROP', 'Original ROP')
num1 = num2str(round(rop_lost_time,0));
txt1 = ['Lost time:  '];
txt3 = [' hrs'];
txt4 = [' '];
```

```
txt32 = strcat(txt1, txt4);
text(95, depth_text_1, txt32, 'fontsize', 10,'Interpreter','latex')
txt33 = strcat(num1, txt4, txt3);
text(95, depth_text_2, txt33, 'fontsize', 10,'Interpreter','latex')
hold off
grid on
box on

subplot(1,4,4)
scatter(bitballing, mdrkb, 20, 'filled', 'r')
hold on
scatter(vibration, mdrkb, 20, 'filled', 'b')
scatter(wear, mdrkb, 20, 'filled', 'c')
scatter(ucs_event, mdrkb, 20, 'filled', 'm')
set(gca, 'YDir','reverse')
title('Drilling Events & NPT')
xlabel('')
ylim([0, TD+200])
legend('Bitballing', 'Vibration', 'Wear', 'Hard Rock')
xlim([0 5])
num1 = num2str(round(total_lost_time,0));
txt1 = ['Total lost time:  '];
txt3 = [' hrs'];
txt4 = [' '];
txt32 = strcat(txt1, txt4);
text(2.2, depth_text_1, txt32, 'fontsize', 10,'Interpreter','latex')
txt33 = strcat(num1, txt4, txt3);
text(2.2, depth_text_2, txt33, 'fontsize', 10,'Interpreter','latex')
hold off
grid on
box on

nameofplot = 'total_lost_time_';
fig1 = figure(37);
fig1.Renderer = 'Painters';
filetype = '.pdf';
txtconcat = strcat(nameofplot, wellname, filetype);

h=gcf;
set(h, 'PaperOrientation', 'landscape');
set(h, 'Paperposition', [0 0 13 8]);
print(gcf, '-dpdf', '-fillpage', txtconcat);
```

Export to Excel

```
% Transpose vectors before exporting
Bitballing = transpose(bitballing);
Vibration = transpose(vibration);
Bit_Wear = transpose(wear);
Hard_Rock = transpose(ucs_event);

Bitballing(Bitballing == -1) = 0; % Set non flagged events equal to zero.
Vibration(Vibration == -1) = 0; % Set non flagged events equal to zero.
Bit_Wear(Bit_Wear == -1) = 0; % Set non flagged events equal to zero.
Hard_Rock(Hard_Rock == -1) = 0; % Set non flagged events equal to zero.
T = table(mdrkb, tvdrkb, mse, mse_corr, ds, mse_ds, Bitballing, Vibration, Bit_Wear, Hard_Rock, rop, wob,
tqa, bdia, ecd, sppa, sonic);
t = table(total_lost_time, total_trip_time,total_trip_time_lost, time_original, rop_lost_time,
rop_lost_time_bb, rop_lost_time_vibration, rop_lost_time_wear, rop_lost_time_ucs, cut_off, ucs_cutoff_1);
name = '_NPT_Drilling_Events_Detection';
file = '.xlsx';
filename = strcat(wellname, name, file);
writetable(T,filename,'Sheet',1)
writetable(t, filename, 'sheet',2)
```