DATA-DRIVEN NUMERICAL SIMULATION AND OPTIMIZATION USING MACHINE LEARNING, AND ARTIFICIAL NEURAL NETWORKS METHODS FOR DRILLING DYSFUNCTION IDENTIFICATION AND AUTOMATION

A Dissertation

by

ENRIQUE ZARATE LOSOYA

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Eduardo Gildin
Co-Chair of Committee,	Zenon Medina-Cetina
Committee Members,	Samuel F. Noynaert
	Jerome Schubert
Head of Department,	Jeffrey B. Spath
Chair of Intercollegiate Faculty,	Timothy Jacobs

May 2022

Major Subject: Interdisciplinary Engineering

ABSTRACT

Providing the necessary energy supply to a growing world and market is essential to support human social development in an environmentally friendly. The energy industry is undergoing a digital transformation and rapidly adopting advanced technologies to improve safety and productivity and reduce carbon emissions. Energy companies are convinced that applying data-driven and physics-based technologies is the economical way forward. In drilling engineering, automating components of the drilling process has seen remarkable milestones with considerable efficiency gains. However, more elegant solutions are needed to plan, simulate, and optimize the drilling process for traditional and renewable energy generation.

This work contributes to such efforts, specifically in autonomous drilling optimization, real-time drilling simulation, and data-driven methods by developing: 1) a physics-based and data-driven drilling optimization and control methodologies to aid drilling operators in performing more effective decisions and optimizing the Rate of Penetration (ROP) while reducing drilling dysfunctions. 2) developing an integrated real-time drilling simulator, 3) using data-driven methodologies to identify drilling inefficiencies and improve performance. Initially, a novel drilling control systems algorithm using machine learning methods to maximize the performance of manually controlled drilling while advising was investigated. This study employs feasible non-linear control theory and data analysis to assist in data pre-analysis and evaluation. Further emphasis was spent on developing algorithms based on formation identification and Mechanical Specific Energy (MSE), simulation, and validation. Initial drilling tests were performed in a lab-scale drilling rig with improved ROP and dysfunction identification algorithms to validate the simulated performance. Ultimately, the miniaturized drilling machine was fully automated and improved with several systems to improve performance and study the dynamic behavior while drilling by designing and implementing new control algorithms to mitigate dysfunctions and optimize the rate of penetration (ROP).

Secondly, to overcome some of the current limitations faced by the industry and the need for the integration of drilling simulation models and software, in which cross-domain physics are unified within a single tool through the proposition and publication of an initial common open-source framework for drilling simulation and modeling. An open-source framework and platform that spans across technical drilling disciplines surpass what any single academic or commercial organization can achieve. Subsequently, a complementary filter for downhole orientation estimation was investigated and developed using numerical modeling simulation methods. In addition, the prospective drilling simulator components previously discussed were used to validate, visualize, and benchmark the performance of the dynamic models using prerecorded high-frequency downhole data from horizontal wells.

Lastly, machine-learning techniques were analyzed using open, and proprietary recorded well logs to identify, derive, and train supervised learning algorithms to quickly identify ongoing or incipient vibration and loading patterns that can damage drill bits and slow the drilling process. Followed by the analysis and implementation feasibility of using these trained models into a contained downhole tool for both geothermal and oil drilling operations was analyzed. As such, the primary objectives of this interdisciplinary work build from the milestones mentioned above; incorporating data-driven, probabilistic, and numerical simulation methods for improved drilling dysfunction identification, automation, and optimization.

DEDICATION

To my beloved parents, who raised me through times of hardship but never gave up hopes of a better future. Your selfless actions remain the main source of my conviction and perseverance. I am grateful for having such loving, and caring parents; I hope to return your sacrifices with joy one day and make you proud of the man you raised.

ACKNOWLEDGMENTS

I want to thank my advisors Dr. Eduardo Gildin, and Dr. Zenon Medina-Cetina, for their enthusiasm, guidance, and support over the years.

I am grateful for having excellent advisors. Dr. Eduardo Gildin is one of the most exceptional people I have ever meet. His honest care, dedication, and passion in helping students succeed are reflected in this lengthy record of academic accomplishments. Thank you for entrusting me and being there for us, your students.

Dr. Zenon Medina-Cetina, your outstanding dedication to research, outreach, and detail is inspiring. Thank you for providing me with the vision and opportunity to pursue graduate school, your generous support during unprecedented times, but, more importantly, for teaching me how to be a better researcher.

I would also like to thank Dr. Samuel Noynaert, Dr. Jerome Schubert, and Prof. Fred Dupriest for their excellent technical guidance, interest, and motivation.

I would like to thank all of you for the long advising hours, and discussions. Texas A&M University is very fortunate to have professors of your caliber and passion.

Prof. Fred Dupriest, for his invaluable input on drilling limiters. Fred Florence from RigOps for his resolute support towards drilling automation. Mr. Maldonado for his relentless dedication and support of graduate students. My lab-mates for creating an excellent work environment, Narendra Vishnumolakala, for his unwavering commitment, collaboration, and drive. Francisco Coral for every long conversation while living away from our families, and Bernardo Huesca for his uncompromising passion and encouragement. Lastly, I thank everyone involved in the development of each project.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis (or) dissertation committee consisting of Professor Dr. Eduardo Gildin, Dr. Samuel Noynaert, Dr. Jerome Schubert, of the Harold Vance Department of Petroleum Engineering, Dr. Zenon Medina-Cetina, of the Zachry Department of Civil, and Environmental Engineering.

Funding for research expenditures was provided by the Department of Petroleum Engineering, the Dwight Look College of Engineering in Texas A&M University.

The methods, and processes discussed in Chapter 4, were developed in collaboration with Intuitive Machines. The short-hub hardware component development and impedance modeling calculations discussed in Sec. 3.1 were performed by E-Spectrum.

Partial data analyzed for Section 4.4, and 4.3 was provided by a U.S.-based service company under in-kind collaboration agreement with the Petroleum Engineering Department at Texas A&M University.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

This work and graduate studies were made possible by partial support from the following sources:

- The Mexican National Council of Science and Technology (CONACYT) #470398.
- Foundation CMG through the FCMG Research Chair at Texas A&M University.
- A fellowship from the Reservoir Dynamics and Control Research Group.
- Intuitive Machines through co-ops and internships.

- E-Spectrum Technologies under contract DE-SC001986 from the U.S. Department of Energy.

- The Stochastic Geomechanics Laboratory through the Cross-Border Threat Screening and Supply Chain Defense (CBTS), The U.S. Department of Homeland Security, Science and Technology Directorate (S&T) Office of University Programs (OUP) under collaborative agreement #18STCBT00001.

- The National Science Foundation (NSF) under grant #2100640.

The contents of this dissertation are solely the responsibility of the author and do not necessarily represent the official views any contributor or government agency.

NOMENCLATURE

ROP	Rate of Penetration
RPM	Revelutions per Minute
WOB	Weight on Bit
BHA	Bottom Hole Assembly
MWD	Measurement While Drilling
LWD	Logging While Drilling
MSE	Mechanical Specific Energy
J	Rotary Inertia
m	Mass
e	Eccentricity
k	Stiffness
c	Damping
F_1x	Axial Force of First Disk
F_1y	Lateral Force of First Disk
T_1	Torque of First Disk
F_2x	Axial Force of Second Disk
F_2y	Lateral Force of Second Disk
F_2z	Vertical Force of Second Disk
ϕ_a	Angular Displacement of Drillstring
ϕ_b	Angular Displacement of Drillstring
u_T	Torque Control Signal
T	Torque on Bit Function

$\dot{\phi}_b$	Desired Angular Velocity
m_0	Mass Constant
c_0	Damping Factor Constant
k_0	Spring Constant
U_b	Axial Position of BHA
$\dot{U}_b$	Velocity of BHA
$\ddot{U}_b$	Acceleration of BHA
$\rho(t)$	Rate of Penetration
ρ_0	Constant Nominal Rate of Penetration
R_b	Bit Radius
μ_{bit}	Friction Coefficient at Bit Rock Contact
c_{bit}	Bit Coefficient
ρ_{bit}	Radial Rate of Increase
x_n	Lateral Coordinates
y_n	Longitudinal Coordinates
Φ_n	Torsional Coordinates
Ω	Angular Velocity of Topdrive
T_s	Commanded Torque
${}_{\mathrm{P}}\mathrm{r}_{\mathrm{IMU}}$	IMU Position
${}_{\mathrm{P}}\mathrm{v}_{\mathrm{IMU}}$	IMU Velocity
ϕ_{K}	Bit Attitude
ω_{K}	Bit Angular Velocity
$\delta\theta_{\mathrm{K}}$	Scalar Bit Azimuth about z relative to J
ϕ_{J}	Torsion Attitude Relative to S
ϕ_{S}	Steering Attitude (RSS) or Misalignment

ϕ_{B}	BHA Attitude
ω_{B}	BHA Angular Velocity
ϕ_{M}	Magnetometer Misalignment Relative to K
ϕ_{C}	IMU Misalignment Relative to K
${}_{M}b_{mag}$	Magnetometer Bias in M frame
${}_{B}B_{ds}$	Magnetic Field due to Drillstring in BHA Frame
${}_{C}b_{acc}$	Accelerometer Bias in IMU Frame
${}_{C}b_{gyro}$	Gyroscope Bias in IMU Frame
F_n	Jacobian of Dynamics
H_i	Measurement Sensitivity Matrix
R_i	Measurement Covariance

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TABLE Page

1. INTRODUCTION

In order to extract underground energy resources, such as hidrocarbons and fossil-based reservoirs or a heat source in geothermal reservoirs, a well is drilled to link the underground resource to the surface. It is sometimes necessary to drill several miles underground to access them, e.g., economic sources such as unconventional reservoirs.

The modern drilling process for underground energy extraction consists of using a rotary drilling rig to rotate and assert force into a drillbit to break sedementary rock and create a hole in the ground, see Figure 1.1.

Pipe is feed into the system as needed until the desired target depth and zone is reached. First, fluid is pumped into the pipe and through the drillbit to keep the hole from collapsing from external pressures and circulate drilling cuttings off the borehole. The fluid is then circulated out of the system to the surface, where it goes back into the mud pits for filtering, monitoring and reuse. Finally, the amount of pipe feed and force applied to the drillbit is controlled by the drawworks. As shown in Figure 1.2, a modern rotary drilling rig comprises several electrical and mechanical systems, including, mechanical mechanisms such as the drawwork system, electronics, and pumps to accurately control the drilling process happening below the surface. Lastly, the traditional derrick shape is a lattice structure that provides the structural support needed to hold the top-drive, drillstring, and other mechanisms in place.

The section of the drillstring inmmediatly behind the drill bit is usually referred as the bottom hole assembly (BHA). It consists of necessary components to provide extra weight on bit (WOB), such as the heavy weight drillcollars (HWDC). Measurement tools, such as measurment while drilling (MWD), pressure while drilling (PWD), and logging while drilling (LWD). Additionally, the BHA hosts compoenents for directional control, such as the bent-subs, rotary steerable systems (RSS), or stabilizers.

The circulation fluid also ensures that the pressure inside the hole is sufficient to substain

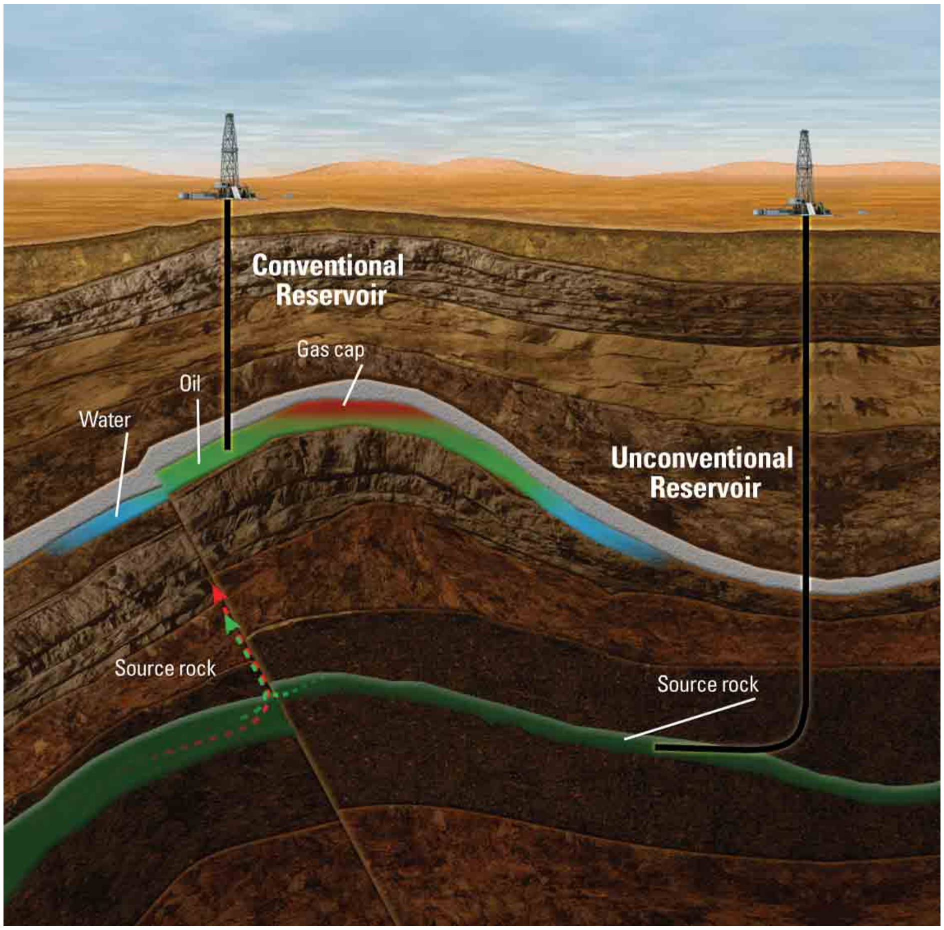


Figure 1.1: Typical setup of a conventional and unconventional energy extration, in this case, from a hidrocarbon reservoirs. A well is drilled from the surface to the extraction geological target zone, *adapted from* [4].

outside pressure, this is, the hydrostatic pressure is larger than the collapse pressure of the rock formation. It also prevents fluid from flowing uncontrolled into the wellbore by maintaining a pressure higher than the porous rock pressure, referred as the pore pressure. Lastly, the asserted hydrostatic pressure must be kept at a lower pressure than the fracturing limit of the rock nearby, referred as the fracturing pressure, to avoid risking uncontrolled losses of drilling fluids into the strata.

The borehole pressure must be higher than the maximum allowed collapse and pore pressures, while being smaller than the fracturing pressure of the rock nearby for any given depth exposed to the rock strata being drilled. The aforementioned relationship rule defined by the lower and upper

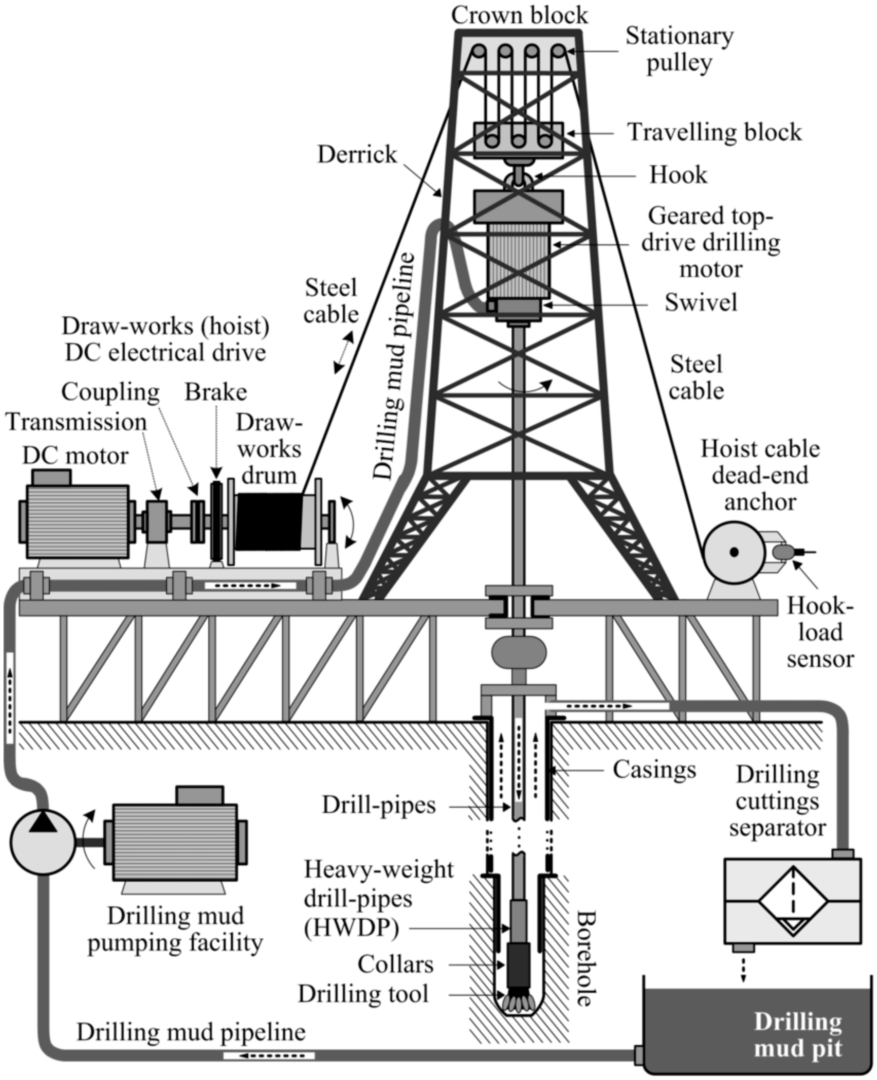


Figure 1.2: Schematic layout of petroleum drilling rig with top-drive and drawworks drives for drill-string rotary and longitudinal motion (Figure after [5]).

bounds of hydrostatic pressures is often referred as the ”drilling window"' [6].

As can be seen in Figure 1.2, the drilling rig is a combination of mechanical and electrical systems. The mechanical systems are the drillstring, the top-drive, and the drillbit. The electrical systems are the pump, the compressor, and the compressor control unit. This combination of systems approach and process automation integration have led to the automation of several components of the traditional drilling rig. Recently, major operators have partnered with service companies to continue evolving the modern drilling rig and automation, for instance, a horizontal well was drilled successfully using a fully automated drilling rig, the PACE-R801, developed by Nabors in coordination with Exxon Mobil [7].

1.1 Drilling Automation

Drilling automation is the use of computer-assisted devices in drilling a wellbore. It involves considerable technology, from accurate and redundant instrumentation, downhole drilling tools to sophisticated control algorithms. Some previous successful examples of automation in today's industry are managed pressure drilling systems and torque oscillation controllers. The economic drivers for automation are increasing complexity, data overwhelm, efficient drilling operations, and safety.

Since 2012, drilling automation has progressed more than many people thought was possible for the industry back then. The next however is more difficult. The integration of all the automated functions to ensure maximum performance. The performance of a highly automated drilling solution is based on how well it coordinates a complex sequence of steps.

More than one hundred experts participated in a workshop to discuss and try to forecast the future of automated drilling. Roughly a third of whom were from outside the oil industry; they wrote a vision statement about what drilling will look like in 2025 [8]. Eventhough some key points have not yet materialized, the overall statement objectives today, are looking every more feasible. Notwithsanding, there are still several drilling System automation challenges that are not commonly addressed in the literature.

Examples of successful automated drilling systems include the addition of the top-drive electrical system to rotate the drillstring – modern systems have anti-stick slip controllers built-in [9, 10, 11, 12]. Another significant development was the inclusion of the top-drive and advanced motion compensation systems for offshore drilling rigs [13]. The use of managed pressure drilling systems used to access previously inaccessible formations offshore [14, 15, 16], and more recently, the creation and evolution of the Rotary Steerable System (RSS) used for downhole trajectory control in combination with Measurement While Drilling (MWD) tools. Rotating while building curve was now possible. With the advent of measuring while drilling tools (MWD) and downhole to surface communication via mud-pulse telemetry, directional drilling became much more affordable, safer, and accessible. A significant increase in horizontal wells activity was seen in 1996. Today, the industry relies on this technology to access previously inaccessible formations. Over the past decade, advances in drilling structures technology have mainly been focused on improved remote control systems and incremental materials, and manufacturing improvements [17, 18, 19].

According to the DNV GL's Technology Outlook 2015 [20] Drilling Automation and smart well technology could decrease CAPEX drilling costs by 30-50% compared with traditional rigs. The BP™ 2015 Technology Outlook estimates that *"advances in technology could reduce industry extraction costs by approximately 25% by 2050 "* [21]. Some operators have heavily invested on drilling automation development projects to commercialize autonomous robotic systems.

Modern automated optimization algorithms have been tested in the field, improvements of 10% to 80% in the rate of penetration (ROP) over a traditionally-controlled rotary drilling rig have been observed. As well as when building and steering wells with an 80% improvement while sliding. The ongoing developments make substantial improvements attainable [22, 23, 24].

Drilling automation has evolved over the previous years, and is becoming a major industry effort, in part, due to declining profit margins and a increasing demand for more efficient and cost effective solutions to provide a sustainable production of energy. Operators and service companies in the oil and gas industry are looking for a solution to automate major components of their operations. Automation has made significant cost, safety, and efficiency improvements in other industries, for instance, in the manufacturing or aviation industry, where automated systems compose a higher stake in daily operations. The concept of automating drilling operations has been in the literature since the 1960s, whereby a combination of electronics and mechanics was the main topic of discussion [25]. Today, the concept of drilling automation has evolved to include the applications of advanced machine learning, artificial intelligence, and real-time modeling within advanced drilling advisory systems [26].

A quick analysis of data obtained from OnePetro [27], an online library of technical literature for the oil and gas exploration and production industry, shows that drilling automation not only continues to be a relevant area of interest in the field, but its relevancy is increasing as more than 400 technical papers per year are being published within "well drilling" in the most relevant SPE conferences and journals (see Figure 1.3).

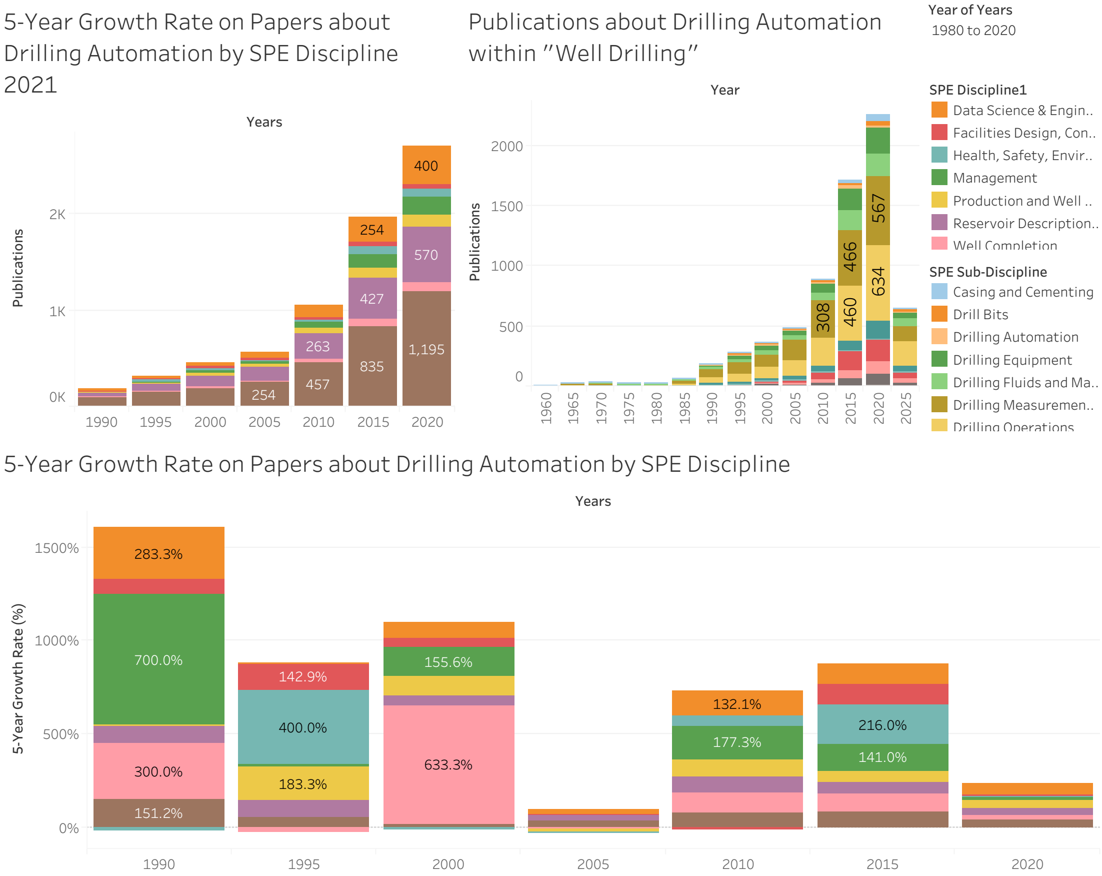


Figure 1.3: SPE publications about drilling automation by discipline since 1990. An interactive version of this dashboard is available online [28].

This research aims to use physics and data-driven methodologies to develop novel real-time automated solutions for drilling optimization and dysfunction identification and drilling processes simulation and control. The objective is to research the current state of the art in drilling dysfunction identification and to develop a library of trained machine learning models using previous laboratory and field measurements reported in the literature.

1.2 Drilling Automation Barriers

Today's oilfield economic conditions demand hydrocarbon production costs, including drilling operations, to be minimized. It is important for drilling engineers planning wells to fully comprehend each dysfunction in order to develop a means to mitigate their impact. As progress is made in this effort, it has become increasingly clear that many issues facing the drilling industry cannot be solved or solutions implemented with traditional drilling technology and platforms. The automation of drilling data collection and control systems response is increasingly becoming a key aspect of advancing many aspects of drilling performance.

Despite the advancement of drilling automation technologies over the last decade, drilling for energy resources and extraction remains a predominately manual operation where major gains in safety and optimization are still to be fullfilled. The automation of drilling operations is a key component of the future of drilling operations, thus:

1.2.1 Why Drilling Automation?

The major drivers for automation where automated systems can effectively mitigate or improve current solutions and the technology needed to address each one of these drivers is within reach [29].

1. ***Increased well complexity and excessive Non-Productive Time (NPT).***

   Multi-Directional extended reach wells are becoming more common especially in the offshore business. Ambitious projects with narrower pressure margins, complex trajectories and uncertainties must rely on robust automated systems in the future to achieve extended distances while mitigating costs, risks while reducing NPT.

2. ***Data overload.***

   Modern computer systems are capable of crushing vast amounts of data; from acquiring, storing, categorizing and interpreting it in a relatively short period. Automated systems will be able to run local optimizations based on history or model-based data that will either take control of drilling parameters or advise the driller accordingly. The system should be able to predict, determine and intervene to avoid any significant failure occurrence.

3. ***Knowledge transfer.***

   Skilled and experienced employees will eventually exit the industry, leaving their roles to younger professionals. This loss in expertise can be reduced using autonomous systems.

4. ***HSE reasons.***

   Safety has always been the top priority for companies where dangerous working and extreme conditions are common, such as the Oil and Gas industry. Also, strict environmental regulations must be followed to avoid serious negative consequences to the ecosystem and the health of collaborators. Automated systems will decrease the number of people working in hazardous zones, as well as improve the HSE monitoring processes currently in place.

5. ***Economic drivers.***

   The automation of drilling operations will also reduce the risk of failure, thereby increasing safety and the efficiency of the company.

Variable	Field Observed Max. Error	Derivative variable error	Small Error	Consequence	Large Error	Consequence	Worst Case Scenario
Torque	*>100%*	MSE / Rig State	*5% - 10%*	Sub-optimal drilling analysis, planning and execution. Inadequate MUT limit	*>20%*	±5, 000$ft * lb$ error for most TDS-11 drives. Bit, Motor, MWD, Tubular or Vibration-Induced Failure.	Loss of Drill String
RPM	*>100%*	MSE / Rig State	*2% - 10%*	Sub-optimal drilling. Improper feedback loop to the TopDrive. Poor Anti-Stick-Slip performance.	*>20%*	Bit or MWD Failure, Vibration dysfunction. Poor Drilling Performance	Loss of Drill String
Hookload	*>100%*	WOB / MSE / Bit Depth / Rig State	*2% - 5%*	Sub-optimal drilling. Improper WOB feedback loop to the draw-works.	*>10%*	± 50, 000lb for most sensors	Loss of Drill String
Block Position	*>25%*	Bit Depth / ROP	*1% - 2%*	Block position error is cumulative. Incorrect MD/TVD/Survey Measurements.	*>5%*	10ft error per joint pipe drilled. Significant survey errors and severely compromised TD.	Wellbore intersection
Flow Out	>100%	Rig State / Kick Detection	*5% - 10%*	Poor well control. Poor stuck pipe/pack off detection.	*>20%*	Inability to detect kicks, packs offs or other problems.	Blowout
Pit Volume	*>5%*	Δ Pit Volume. Kick size/density.	*1% - 2%*	Poor well control. Poor kick detection/performance.	*>5%*	5 bbl could be 100% error in well control calculations.	Blowout
Pump Pressure	>100%	ΔP / MSE Rig State	2% - 5 %	Poor managed pressure performance. Poor well control. Wear/damage to downhole motors/turbines.	>10%	500 - 750 psi error could lead to potential damage to motors/turbines/MWD tools. Potential for kicks/fracturing when operating near pore pressure/narrow fracture gradient.	Blowout
Pump Rate	>100%	MSE / Rig State	2% - 5%	Poor Diagnostic ability with tracers. Poor hole cleaning. Wear/damage to downhole motors/turbines.	>10%	10 Strokes/min error could significantly affect well control or managed pressure scenarios.	Blowout

Table 1.1: Common instrumentation errors found in the field. *After Behounek 2015 [30] & Zenero et al. 2016 [31].*

An experienced human driller could easily detect a significant anomaly in any reading on the fly and act accordingly. Nonetheless, this may not hold true for small but essential instrumentation anomalies. The driller would make decisions based on inaccurate data. Furthermore, reliable instrumentation data is vital for any closed-loop autonomous system, without robust redundancy. A closed-loop system will just follow the difference between the input and feedback output of the controlled plant to compute an appropriate controller gain, which can lead to a significant inaccurate response, that could potentially cause hazardous disruptions[32].

The basic instrumentation data is used to calculate outstanding optimization derivatives variables such as Mechanical Specific Energy (MSE), Weight on Bit (WOB), and Rate of Penetration (ROP); carrying and increasing the error margin. Improved algorithms coupled with instrumentation redundancy and data analysis are being developed to mitigate control or surveillance errors ([33] [34] [35]). The current situation with rig instrumentation data represents an obstacle to drilling automation systems. High speed, reliable and accurate measurements are needed.

Drilling Data

According to the DSATS Data Quality Assurance Subcommittee [36], drilling data is composed of distinct main areas:

1. ***Surface and Downhole Measurements made at a rig***.
   - Measurements from high and low-frequency instrumentation data coming from the rig or below.
2. ***Metadata describing those measurements***.
   - Calibration, accuracy, temperature resolution, drifting, location, manufacturer , timezone, working hours.
3. ***Derived variables from raw measurements.***
   - , Measured Depth (MD), Drill Joint Number, Mechanical Specific Energy (MSE), True

Vertical Depth (TVD, Weight on Bit (WOB), Rate of Penetration (ROP), Stick-Slip Percentage.

4. ***Contextual and situational data that describe the environment within which a measurement is made.***

    - Flow On Annular Pressure, Swabbing, Flow Off annular Pressure, Tripping In/Out, Drilling, Making a Connection, Setting Casing.

5. ***Data generated by real-time applications.***

    - Physics-based Models, Data Driven Models, Recommended WOB for an interval.

6. ***Data about the performance specifications of drilling devices.***

    - Motors, Mud Pumps, Shakers, Pressure Transducers, Top Drives,Drawworks.

7. ***Historical data conceivably used in automation tasks.***

    - Offset wells, Equipment Configuration, Performance Limiters, Historic pump pressure, Dangerous interval observed in offset wells...

Surface-Downhole Data Link

Surface and downhole readings are important for drilling optimization, from the development of dynamic simulation models to the real-time analysis and decision loop. [37, 38]. Unfortunately, the current mud-pulse telemetry system widely used in the industry to communicate downhole data to the surface is extremely bandwidth limited in addition to its inherent lag. Data rates usually vary from 3 bits per second (bps) to 20 bps depending on depth.[39], this limited bandwidth forbids high-rates data transmission from any downhole tool and therefore vanishes any sophisticated real-time drilling optimization autonomous system.

Modern downhole tools are almost entirely autonomous [40]. For example, measurement while drilling and rotary steerable tools. Those tools encapsulate many electronic components that are isolated from surface conditions and feedback which makes much higher logging rates possible; the data is used on their separate closed-loop controllers without any reliance on surface measurements to make the right changes. High-speed logged data is usually stored internally and recovered after the bit has been taken back to the surface then it is analyzed and used for drilling optimization purposes, in preparation for the next well. Downhole-Logging and post-analysis can be very expensive and time-consuming, most operators today do not log and store data on a routine basis.

"Wired" drill-pipe technology has been developed to overcome the bandwidth limitation of current telemetry systems [41]. Transfer rates offered by this solution are orders of magnitude higher than traditional mud pulse telemetry. A wired drill pipe is capable of transferring data at 57,600 bps without significant signal attenuation or noise caused by changes in depth, fluid properties, or formation resistivity. Transmitting every single filtered high-speed measurement from downhole tools in real-time is now possible, albeit cost-prohibitive for most wells. Several fields tested cases highlighting the advantages of wired pipe technology are available in the literature [42, 43, 44, 45, 46, 47]. Unfortunately, such technology costs are still prohibitive for most onshore and not vertically integrated drilling operations.

Levels of Automation

Levels of automation (LOA) has been proposed in the literature to classify an automated process from entirely manual to fully autonomous. Endsley and Kaber[48] produced a ten-level LOA taxonomy to designate functions between human and computer systems. Table. 1.2 shows Endsley's Levels of Automation; Endsley highlighted that that removing the human factor from such high cognitive level processes with limited physical understanding or modeling is detrimental to the performance of any automated system [49, 48]. Parasuraman et al. [50] proposed a four-stage model of human-automation interaction design that defines a recommended level of automation for each of the four basic operations previously discussed; the author empathizes that human involvement in any automated process becomes more significant as higher-risk decision making is required.

Level of Automation	Monitoring	Generating	Selecting	Implementing
1.- Manual Control	Human	Human	Human	Human
2.- Action Support	Human/Computer	Human	Human	Human/Computer
3.- Batch Processing	Human/Computer	Human	Human	Computer
4.- Shared Control	Human/Computer	Human/Computer	Human	Human/Computer
5.- Decision Support	Human/Computer	Human/Computer	Human	Computer
6.- Blended Decision	Human/Computer	Human/Computer	Human/Computer	Computer
7.- Rigid System	Human/Computer	Computer	Human	Computer
8.- Automated Decision	Human/Computer	Human/Computer	Computer	Computer
9.- Supervisory Control	Human/Computer	Computer	Computer	Computer
10.- Full Automation	Computer	Computer	Computer	Computer

Table 1.2: Levels of automation *After Endsley & Kaber 1999 [48].*

1.3 Basic Drilling Principles

1.3.1 Bit Dysfunctions

How fast a well is drilled depends on its drilling speed or most commonly referred as the Rate of Penetration (ROP), ROP directly proportional to the frequency and magnitude of repetition of indentation and sliding into a rock. Indentation is achieved by applying weight on bit (WOB) and sliding is provided through the drill string revolutions per minute (RPM). The amount of indentation that occurs for any given WOB is often referred to as the Depth of Cut (DOC) per revolution. DOC is proportional to WOB applied, this is, the contact area under the cutter increases with force until the load equals the rock strength under overburden. WOB and RPM are independent variables as for any supplied WOB can only achieve a specific DOC regardless of RPM. Alternatively, the WOB being applied has no effect on the distance that each cutter must slide for every revolution of the bit. Thus, WOB and RPM are linearly independent to the Rate of Penetration (ROP).

Bit dysfunctions are inefficiencies in the rock cutting process that impact drilling performance. There are several distinct bit dysfunctions, however, most of them are caused and correlated to excessive downhole vibrations. Whirl is a lateral oscillating sine wave vibration in the bottom hole assembly (BHA). As the drill string is rotated, it begins to move laterally due to mass imbalances, creating a vibration wave that travels through the drill string. When this wave reaches the bit, it tilts the bit and drives the bit into the borehole wall. This movement creates side cutting which in turn drives the creation of borehole patterns. It has been proven in the field that the impact force of the bit hitting the borehole wall is great enough to damage the outside cutters, which in turn negatively impacts ROP [22].

Stick-slip dysfunction, is an alternation in bit speed, relative to the surface RPM, that is driven by oscillations in torque [51, 52, 53]. Full stick-slip can be very damaging to the bit and downhole tools. The typical drilling optimization holds that higher rotary speed can mitigate stick-slip vibrations at the expense of increased whirl vibrations, and higher WOB tends to increase stick-slip and mitigate whirl. The operational challenge is that these tendencies can vary from run to run, and even within an individual section as pipe length, formation, and bit condition; all play a role in the balance between these dysfunctions [40, 54, 55, 56].

Bit dysfunctions can be summarized as:

- Axial or longitudinal vibration caused by the interaction between the rock and the bit. This type of vibration is usually called "bit-bounce".

- Lateral (Bending) vibrations caused by unbalanced centripetal forces during rotation. This type of vibration is usually named whirl. Forward (backward) whirl is the rotation of a deflected section of drill string around the borehole wall in the same (opposite) direction of the drill string rotating axis and is the least (most) damaging type of whirl dysfunction.

- Torsional vibration caused by non-linear interactions between the bit cutters and the rock, coupled with the drill string assembly. Stick-slip is the torsional vibration of the drill string characterized by alternating pauses and intervals of the increased angular velocity of the BHA.

- Hydraulic vibrations caused by the mud circulation system, e.g. pump pulses and mud motors.

- Full stick-slip can be very damaging to the bit and downhole tools. Drilling optimization involves finding the optimal operational strategy that can mitigate damage and increase performance, which usually involves conflicting control decisions. Higher rotary speed can mitigate stick-slip vibrations at the expense of increased whirl vibrations, and higher WOB tends to increase stick-slip and mitigate whirl. The operational challenge is that these tendencies can vary from run to run, and even within an individual section as pipe length, formation, bit condition, etc., change. Balancing these functions is the key to optimal operations [57, 55, 56].

1.3.2 Bit Mechanics

Fundamentally, all bits drill the same, whether it is a six-blade PDC or a three-blade PDC bit; they simply indent and slide. Indentation is achieved by applying weight on bit (WOB) and sliding via the drill string's revolutions per minute (RPM). The amount of indentation that occurs for any given WOB is the Depth of Cut (DOC). As WOB is applied, the DOC increases, meaning the contact area of the cutter buried in the rock increases. The DOC continues to increase until the load over the contact area is equal to the strength of the rock. Torque is a function of the force required to exceed rock strength and therefore increases with DOC. Both WOB and RPM are independent of one another, as a given WOB can only achieve a certain DOC regardless of the RPM at which the string is turning. Alternatively, the WOB being applied has no effect on the distance that each cutter must slide for every revolution of the bit. Therefore, both WOB and RPM are linearly independent with regard to the rate of penetration (ROP).

1.3.3 Mechanical Specific Energy

A pertinent application of data science in the Oil and Gas (O&G) industry is reducing the Mechanical Specific Energy (MSE), a measure of drilling efficiency, which is widely used by Exxon Mobil in its operations [58].

One of the primary means of maximizing drilling performance is through monitoring and responding to Mechanical Specific Energy (MSE), as it can both identify bit dysfunction as well as maximize ROP in easier footage. MSE, being the amount of energy used per volume of rock drilled, is equivalent to rock strength when the bit is efficient (Teale 1965). While in practice it is rare to have MSE equal rock strength due to system inefficiencies such as the control systems, friction, and hole conditions, the bit is considered efficient any time that the ROP increases proportionately with WOB and to be foundering anytime there is a non-linear response [59].

The most crucial parameter in this design is the magnitude of the weight on bit (WOB) that is being transferred into the formation through the drill bit. The difference in magnitude between the top drive assembly weight and the load cell tension times a transfer-ability ratio pre-determined for a given depth is how the weight on bit is calculated.

$$MSE = 0.35 * (\frac{WOB}{A_b} * \frac{120 * \pi * RPM * T}{A_b * ROP}) \tag{1.1}$$

Where, *WOB* is the weight on bit in psi, which is the weight transferred into the formation through the drill bit. A_b is the area of the bit in square inches in^2, *RPM* is the rotational speed of the drill string, *T* is the torque in $ft - lbs$, and *ROP* is the rate of penetration in ft/hr.

The standard approach to control system design is to base the model on physical laws and restrictions to the corresponding physical parameters. This is called the classic or white-box approach. Control systems are required to function based on noisy measurements and inaccurate instruments. Models that are set with adjustable parameters are called gray-box models. In many control applications cases, linear models that do not completely represent a system based on an exact analytical solution to the relationship between the subprocesses involved but approximate them are enough to acquire an acceptable level of controllability. These models are called black boxes models and are derived using system identification techniques ([60]).

Rate of penetration (ROP) is a bilinear function of WOB and rotational speed and that when the linearity is lost, at the so-called founder point, the drilling conditions are suboptimal and the bit can get damaged. To drill as fast as possible, we need to increase the WOB and the rotational speed just below the limit of non-linearity. The boundary between linear and non-linear behavior is unknown and potentially complex as it results from various causes like reaching the maximum depth of cut (DOC), vibrations or bit balling, for instance. To find an optimum working point, the WOB is increased linearly for a small variation and the response of the ROP to the change is analyzed. If the response is linear then we are below the founder point and we can increase the WOB. If it is non-linear from start to end, then we are above the founder point and we need to decrease the WOB. If the first part was linear but then it starts to be non-linear then we know the founder point value. Thereafter a similar sweep is made with the rotational speed (see Figure 4.7). Based on the collected information, a new functioning point is selected and the procedure is repeated. When the limit to the founder level curve is found, a fast-marching cube algorithm is applied to further optimize the ROP.

1.4 Thesis Objectives and Major Tasks

Owning to major improvements in computation, and gaps encountered in the developments of automated workflows applied for drilling optimization, this thesis aims to attain the following objectives laid out below, including major tasks to be followed:

1. To optimize drilling efficiency and performance by automating the selection of important drilling control parameters. More specifically we will:

    (a) Investigate novel drilling control systems algorithms using machine learning and artificial intelligence to maximize the performance of manually controlled drilling rigs while advising, taking control, or removing entirely the driller's input in common drilling operations for energy extraction.

    (b) Validate the theoretically simulated performance of control algorithms by running drilling tests on a previously built lab-scale rig. A fully automated miniaturized drilling machine was enhanced with the addition of several real-time control loops to improve performance and study the dynamic behavior while drilling to design and develop new closed-loop control algorithms in order to mitigate dysfunctions, optimize the rate of penetration (ROP) and perform real-time analysis.

    (c) Develop an automated rock formation identification model using a data-driven modeling approach that will be investigated using real-time drilling data to train and validate a classification model to improve the efficiency of the drilling process by reducing Mechanical Specific Energy (MSE).

    (d) Demonstrate the feasibility of a layer-based determination and change detection of properties of rock formation currently being drilled in an accurate and fast manner. Data for this specific part of the research will be collected from a custom-built lab-scale drilling rig equipped with multiple sensors. The experiment was conducted by drilling through an arrangement of different rock formations of varying rock strength properties. Data was recorded and stored at a frequency of 2 kHz, which was then filtered, processed, and downsampled to extract relevant features.

2. Automate the identification of drilling dysfunctions in real-time which can lead to significant savings for drilling companies as drilling parameters can be computed automatically, and in turn, increase drilling efficiency in real-time. In specific we,

3. Model and simulate coupled drilling processes to optimize horizontal drilling while providing near real-time performance. We will,

   (a) Investigate the current simulation methodologies and models for the main subsystems when drilling a wellbore and capture their assumptions, requirements, and advantages.

   (b) Select and programming different models that approximate dynamics as a module.

   (c) Couple each individual module into a combined dynamic simulator of the drilling process.

   (d) Developing a virtual drilling simulator written in C++ using high-performance libraries to simulate accurate drilling responses

   (e) Leverage the use of parallel computing, Graphics Processing Unit (GPU) acceleration, multi-body, computational fluid dynamics solvers, and other tools.

   (f) Build prospective modules for drill-string dynamics, torque and drag analysis, geologic modeling, wellbore hydraulics, and pressure modeling, including some visualization functionality.

   (g) The simulation will be applied in several applications, including modeling shallow horizontal jetting, rock drilling, and steering, as well as reconstructing spatial data from high fidelity downhole signals for borehole evaluation.

4. To integrate several sections of the developed simulator framework to demonstrate the final capabilities, including modeling shallow horizontal jetting, rock drilling, and steering, as

well as reconstructing spatial data from high fidelity downhole signals for borehole evaluation.

5. To improve downhole estimation algorithms using advanced orientation estimators for navigation and control.

    (a) Implement a Magnetic, Angular Rate, and Gravity (MARG) filter for Downhole orientation estimation.

    (b) Assess its computational performance by applying numerical modeling simulation methods.

    (c) Use components of the prospective drilling simulator previously discussed to validate, visualize, and benchmark the performance of the dynamic models and assumptions using high-frequency downhole data from horizontally drilled Oil & Gas wells.

6. To develop trained machine learning algorithms and extend a real-time advisory system to aid in supporting drilling operators to perform more effective decisions and optimize the Rate of Penetration (ROP) while reducing drilling dysfunctions.

    (a) Initially, machine learning techniques were analyzed using recorded well logs to identify, derive and train the most common supervised machine learning algorithms to quickly identify ongoing or incipient vibration and loading patterns that can damage drill bits and slow the drilling process.

    (b) Followed by the analysis and feasibility and implementation of these trained models into a smarter downhole tool with a short-hub electromagnetic communication system for both geothermal and oil drilling operations was analyzed. Finally, more complex unsupervised and bayesian-based approaches will be analyzed and compared to the current state in the industry.

1.5 Contributions

Our main contributions are as follows:

1. We prove the benefits and feasibility of using high-frequency closed-loop automation and downhole dynamics.

2. Develop a method and device to estimate rock formation types and layer changes being drilled using only downhole vibration data for bit-rock interactions.

3. Provide a solution architecture integrating rarely coupled numerical drilling dynamic models interacting with each other in real-time, such as developing and applying an erosion model with volumetric calculations mimicking borehole propagation.

4. The development of machine learning-based algorithms and initial blocks of a real-time advisory system to aid drilling operators in performing more effective decisions and optimizing the Rate of Penetration (ROP) while reducing drilling dysfunctions.

1.6 Thesis Organization

This thesis is organized as follows. Chapter 1 provides a brief overview of drilling engineering, some successful automated systems, and noteworthy barriers that hinder the development of robust automated solutions.

Chapter 2 describes the application of modeling and simulation-based engineering methods for the development of several physics-based drilling engineering-related simulators; a downhole drillbit motion model, an integrated real-time horizontal directional drilling framework, and the derivation of an improved orientation algorithm for downhole positioning.

Chapter 3 describes the derivation and application of stemming data-driven models for real-time rock formation identification using machine learning techniques and the initial steps towards an optimized drilling advisory system that uses data-driven classifiers and electromagnetic telemetry to detect drilling dysfunction events.

Chapter 4 combines emerging data-driven, physics-based, and numerical simulation methodologies discussed in previous sections to drilling engineering and automation by constructing a completely autonomous lab-scale drilling rig that optimizes drilling performance by identifying rock layers. The use of drilling simulation and virtual models to build, verify, and test a downhole orientation filter and a dynamic borehole shape approximator.

Lastly, Chapter 5 offers a discussion and conclusion of the most relevant results and closing remarks about future work recommendations.

2. PHYSICS-BASED MODELING FOR DRILLING OPTIMIZATION

Computer simulations have been used extensively in industries from aerospace to defense, where simulation-driven design has helped reduce costs while increasing the efficiency and safety of their operations [61, 62]. The oil and gas industry is no outsider to numerical simulations as they have been extensively used, particularly in reservoir engineering in order to model the behavior of multi-phase fluid flow through a porous medium. The focus on drilling automation has led to notable advances in modeling and simulation of drilling over the past decade, especially when dealing with properly defined areas of the drilling process. Modeling and simulating the entire drilling process in high fidelity is still deemed too complex to be achieved with the current technology. However, advances in computational performance and technology will keep sophisticated drilling dynamics modeling an ongoing area of research [63, 64].

This chapter describes the application of modeling and simulation-based engineering for the development of several physics-based drilling engineering-related simulators, a downhole drillbit motion model, an integrated real-time horizontal directional drilling framework, and the derivation of an improved orientation algorithm for downhole positioning.

2.1 Computer Drilling Simulation

The industry has witnessed major developments in drilling system simulators, however these breakthroughs have been rarely adopted [65, 66, 67, 68]. Operators have expressed their need for an integration of models and software, in which cross-domain physics are unified within a single tool. There is no need to rewrite and solve the same problem over and over for every specific tailor-made package. This research proposes to overcome some of the current limitations faced by the industry through the publication of an open-source common framework for drilling simulation and modeling [65, 66, 69, 70, 71, 72].

Nonetheless, further improvements in drilling automation require the integration and verification of simulation models to develop reliable control-systems algorithms. Yet, the oil and gas industry is still considerably behind in the adoption and active development of a pseudo-standardized platform that embodies the most common methods, guidelines, and processes needed to simulate each different aspect of the drilling process. This is despite numerous efforts driven by various groups and technical sections to plan, guide, and promote roadmaps for the adoption of a more standardized and collaborative industry [73, 24, 74]. Real-time drilling simulators and trainers must balance between high complexity and computation time to achieve their goal to provide useful feedback. There is always a trade-off between complexity and computing time in the simulation of complex behaviors [75, 76, 77].

2.1.1 Overview of Drillstring Dynamics Models

Major strides have been made towards the modeling of the complex non-linear behavior of drill string dynamics, bit-cutter interaction, fluids behavior, model reduction, and other interesting dynamic drilling components [51, 78, 76, 79, 52, 80, 81]. This chapter focuses on the implementation of important models of the drilling process. Especially, those that model drillstring vibration. Modeling drillstring dynamics is the basis for system analysis and control of potentially harmful vibrations. Extensive research has been conducted to describe the physical phenomena in wells while drilling mathematically. Most drillstring dynamic models can be classified in the following major categories [82]:

1. **Lumped parameter**. Lumped parameter models are based on the assumption that the entire drillstring is lumped as a mass-spring-damper system and that an ordinary differential equation describes the behavior of the string. This finite-dimensional system representation provides a rough description of the dynamics taking place at different levels of the string; it can be of one to several degrees of freedom (DOF). The basis of these models are also commonly used to model other physical phenomena such as flow in geothermal reservoirs to power transmission, and blood flow [83, 84, 85].

2. **Distributed parameter**. In distributed parameter models, the drillstring is considered a beam subject to axial and torsional forces, which is a system of partial differential equations

that provide the dynamic characterization of the drilling variables in an infinite-dimensional space. The trade-off is the cost of the model accuracy is the complexity involved in its analysis and simulations.

3. **Neural type time-delay**. These models are a subset of the distributed parameter models that assume negligible damping provides an input-output system description. The time delays are related to the speed of the oscillatory waves traveling throughout the rod and the total string length. This type of model provides a good trade-off between system representation accuracy and complexity of the description. A similarity can be drawn from data transmission models accounting for transmission delays in their mathematical representation.

The most common models to simulate drillstring dynamics in the industry are lumped parameters models. The following paragrapghs provide a brief overview of their derivation, major assumptions, and limitations. The reader is referred to [86, 87] for a more detailed description.

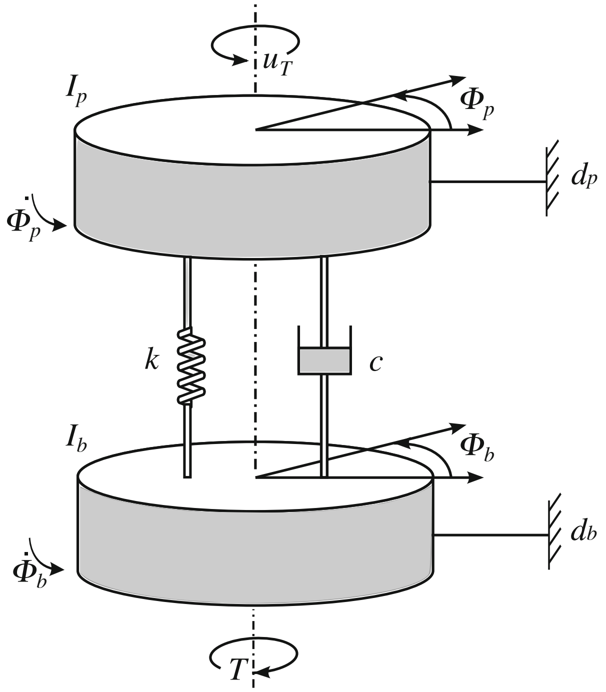


Figure 2.1: Lumped parameter model based on spring-damper-mass dynamic drillstring model.

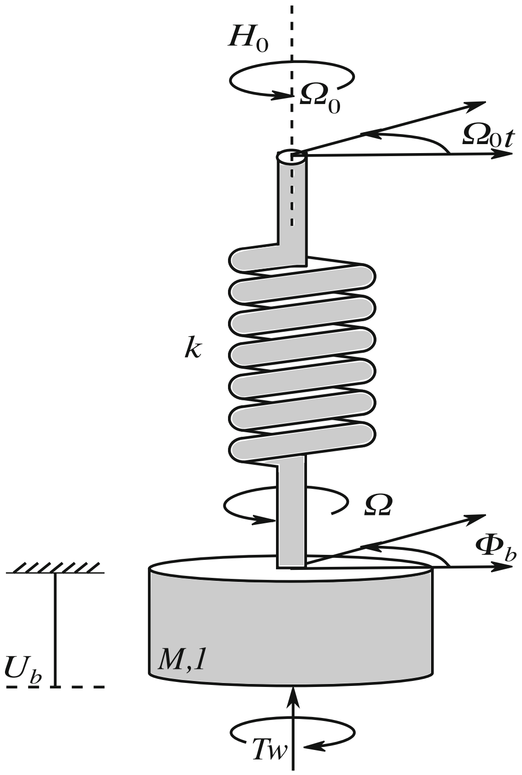


Figure 2.2: Lumped parameter dynamic model of a simplified drillstring.

Torsional dynamics

For torsional dynamics, the use of reduced models for vibration analysis is motivated by the need of defining a simplistic representation of its dynamics, the continuous system, consisting of drillpipes and of the bottom hole assembly, is regarded as a torsional pendulum described by a lumped parameter model with one or multiple degrees-of-freedom (DOF).

Figure 2.1 shows the simplified two-degree-of-freedom torsional model of a conventional vertical drillstring model proposed by [87]. The inertial masses M_I p and I_b, are locally damped by d_b and d_p, which in turn are connected one to each other by a linear spring with torsional stiffness k and torsional damping c. The equations of motion are:

$$I_p\ddot{\phi}_p + c(\dot{\phi}_p - \dot{\phi}_b) + k(\phi_p - \phi_b) + d_p\dot{\phi}_p = u_T \tag{2.1}$$

$$I_b\ddot{\phi}_b - c(\dot{\phi}_p - \dot{\phi}_b) - k(\phi_p - \phi_b) + d_b\dot{\phi}_b = -T(\dot{\phi}_b) \tag{2.2}$$

Where ϕ_b and ϕ_p are the angular displacements of the drillstring and the rotary table. u_T is the torque control signal from the topdrive used to regulate the desired angular velocity $\dot{\phi}_b$. Finally, the torque on bit and other non-linear frictional forces acting along the drillstring are represented by T.

Axial dynamics

A simplified model described by an ordinary differential equation is presented in [88]. The modeling strategy is inspired by the fact that any mass subject to a force in stable equilibrium acts as a harmonic oscillator for small vibrations. This is, the damped harmonic oscillator model describing the longitudinal drillstring motion [88, 89]. The model is a second-order differential equation, on which, $\nu = U_b - \rho_0 t$ with the following form:

$$m\theta\ddot{\nu} + c_0(\dot{(}\nu) + \rho(t)) + k_0\nu = -\mu_1 T(\dot{\Phi}_b) \tag{2.3}$$

Notations m_0, c_0, and k_0 represent the mass, damping, and spring constant; U_b, $\dot{U}_b$,and $\ddot{U}_b$ are the axial position, velocity, and acceleration, of the bottom hole. The system is controlled through the rate of penetration $\rho(t)$, which is an axial speed imposed at the surface, ρ_0 is a constant nominal value. The torque on bit function T depending on the bit angular velocity couples the axial model with the torsional one.

The coefficient μ_1 is given by the following relationship:

$$\mu_1 = 2(R_b\mu_{bit}c_{bit})^{-1} \tag{2.4}$$

Where R_b is the bit radius, μ_{bit} is the friction coefficient at the bit-rock contact, and c_{bit} is the so-called bit coefficient. For a bladed bit, c_{bit} is equal to the dimensionless length of the cutting edge. For a flat bit, c_{bit} is:

$$c_{bit} = \frac{6 + 4\rho_{bit}}{6 + 3\rho_{bit}} \tag{2.5}$$

Where ρ_{bit} is the radial rate of increase of cutter density.

Torsional-Axial-Lateral Dynamics

A discrete system model developed by [1] described in this section and shown in Figure 2.3 of section is usually used for analytical and numerical studies of the coupled axial-torsional-lateral motions of the drillstring.

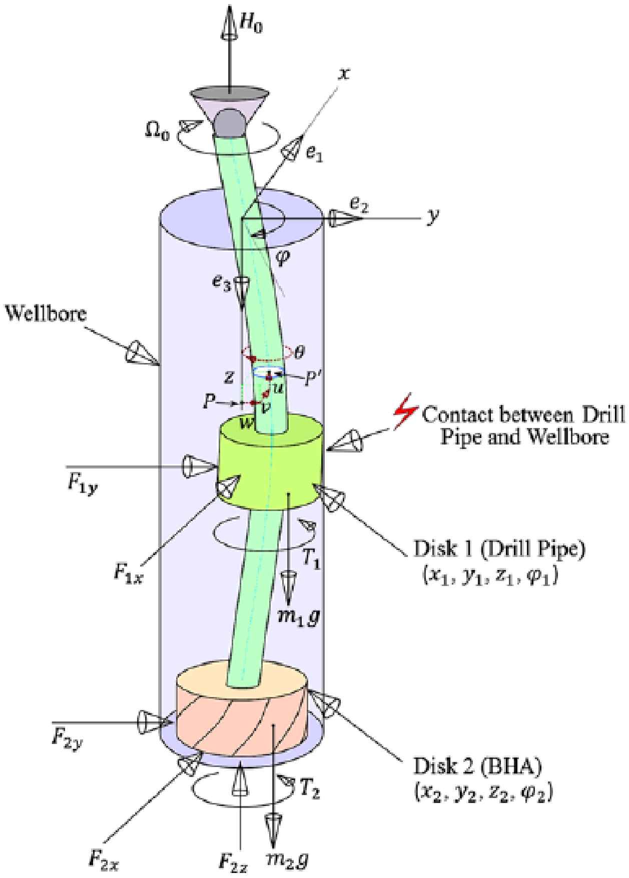


Figure 2.3: Lumped Torsional-Axial-Lateral parameter dynamic model of a drillstring (from [1]).

The model consists of two disks with their corresponding inertia and rotary inertia properties linked together over one constant lenght. The inertia of the pipes are lumped into the first disk, while the inertia properties of the BHA are lumped into the second disk. Both disks's dynamics are described collectively by the goverment equations described in Equation 2.1.1, composed of a total of four DOF; two DOF for lateral motions, one DOF for axial and torsional motions respectively. The system is driven by a rotary motor with a constant angular speed of Ω_0, connected with two identical elastic and massless elements with associated axial, torsional, and bending properties to reflect these of drillstring pipes. The position of each disk is described by the lateral, longitudinal, and torsional coordinates (x_1, y_1, z_1, Φ_1) and (x_2, y_2, z_2, Φ_2). The rotary motor is described by the angular speed Ω and the torque T_s. The system is described by the following governing equations:

$$\begin{cases} m_1\ddot{x}_1 + c_l\dot{x}_1 + 2k_l(x_1 - \frac{x_2}{2}) & = m_1 e\dot{\varphi}_1^2 \cos\varphi_1 + F_{1x} \\ m_1\ddot{y}_1 + c_l\dot{y}_1 + 2k_l(y_1 - \frac{y_2}{2}) & = m_1 e\dot{\varphi}_1^2 \cos\varphi_1 + F_{1y} \\ m_1\ddot{z}_1 + c_a\dot{y}_1 + 2k_a(z_1 - z_2) & = m_1 g - H_0 \\ J_1\ddot{\varphi}_1 + c_1\dot{\varphi}_1 + k_1\varphi_1 & = m_1 \exp\dot{\varphi}_1 2 \sin\varphi_1 + F_{1z} \\ m_2\ddot{x}_2 + c_l\dot{x}_2 + k_l(\frac{x_2}{2} - x_1) & = m_2 e\dot{\varphi}_2^2 \cos\varphi_2 + F_{2x} \\ m_2\ddot{y}_2 + c_l\dot{y}_2 + k_l(\frac{y_2}{2} - y_1) & = m_2 e\dot{\varphi}_2^2 \sin\varphi_2 + F_{2y} \\ m_2\ddot{z}_2 + c_a\dot{z}_2 + k_a(z_2 - z_1) & = m_2 g - F_{2z} \\ J_2\ddot{\varphi}_2 + c_t\dot{\varphi}_2 + k_t(\varphi_2 - \varphi_1) & = -T_{s2} \end{cases} \tag{2.6}$$

Where J and m are the mass and the rotary inertia of each disk, e is the eccentricity which is assumed to be the same, and the influence of torsional acceleration on the lateral centrifugal force has been neglected. The stiffness and damping are denoted by k and c, respectively. The subscripts a, t, and l correspond to axial, torsional, and lateral direction. The quantities F_1x, F_1y, and T_1 are the respective force components and the torque that arises due to interactions between the drillpipe and wellbore. Similarly,the quantities F_2x, F_2y, F_2z, and T_2 are the respective force components and torque that arise due to interactions between the drill bit and rock. The axial, torsional, and lateral vibrations of the discrete system are coupled through these interaction forces and torques.

Dynamic Drillstring Models Discussion

Generally speaking, the dynamics of an oilwell drillstring are influenced by many factors. Most of the existing publications attempt to reveal the dominant vibration mode or the coupling of single modes. Nevertheless, identifying all types of vibrations of drillstring using an analytical or a finite element-based model is much more difficult. However, early models of drillstring dynamics started by using the wave equation to describe its dynamics, as it provided a good approximation reproducing the oscilitory behavior [90]. Simple lumped mass models with field drilling parameters can reproduce complex nonlinear drilling dynamics processes. For instance, [91] developed a single DOF model to study the torsional stability of the drillstring.

A finite-element model of the drillstring by which related dynamic effects, including stick-slip, can be obtained was presented in [92]. In the model, the gyroscopic effect, torsional-bending inertia coupling, inertia-axial stiffening coupling, and gravity were taken into account. The finite-element models were calibrated via a new experimental rig for reproducing major types of drillstring vibrations. Results showed that just a low-dimensional model is enough to predict the behavior of a drillstring in certain configurations.

Model accuracy is related to the involved DOF number. For instance, one DOF model is used to investigate the effects of rotary speed, viscous damping and the natural frequency of the drillstring to investigate the stick-slip phenomenon [91, 93].Examples of two DOF lumped parameter model are used to analyze torsional vibrations resulting when drilling with a polycrystalline diamond compact (PDC) bit. These models are also used to design control strategies to reduce the magnitude of torsional drilling oscillations [94, 95, 12]. Discrete models of a drillstring system of several DOF have been proposed and been used to study non-linear motions. Lastly, an example of coupled axial-torsional dynamics of a drill string are studied through a discrete 32-segment of 128 states model that considers nonlinearities such as dry friction, loss of contact, and state-dependent time delays was proposed in [96]. The normal strain contours of the spatial-temporal system demonstrate the existence of strain wave propagation along the drillstring [97].

Other type of models, including reduced-order models to describe the drilling behavior can be found in [1, 98]. Lastly, a recent review of downhole high frequency data and dynamic models used in discussed in [99].

In this work, we propose the application and formulation of new model based on distributed lumped parameters using idealized conditions to develop an estate estimator filter to study the dynamics of a drillstring system.

2.2 Downhole Dynamic Model Development

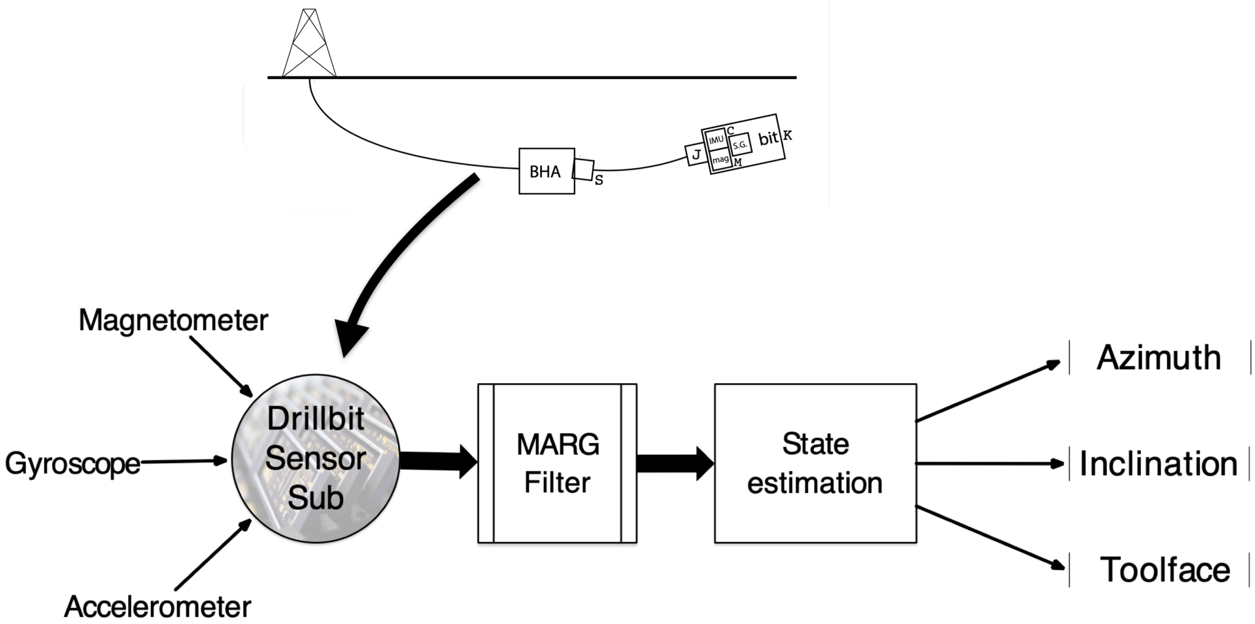


Figure 2.4: Proposed methodology and objective overview for the downhole orientation algorithm and high-frequency validation process.

The following tasks were done as part of this dynamic model development.

1. Developed an initial motion model, parameter estimator, and software architecture (see Sections 2.2.2 and 2.2.5) to model the drilling process.

2. Performed analytic sensitivity (linear covariance analysis) of bit motion (see 2.2.3);

Several parameter estimator and simulation option was evaluated and outlined a in a baseline approach which considered estimation observability, data control volumes, and the flow of data. The baseline approach provides maximum information at the point of processing (i.e., near the BHA) and passes refined information at a lower rate up the drill pipe where it may be combined with information at the BHA and operators' stand. However, for the sake of this section, it is assumed that the information available to the drillbit package is self-contained and does not rely on information from other systems.

2.2.1 Outline of Problem

The problem we set out to address is that of state estimation for a drill bit. The measurements are produced exclusively by a sensor pack located to the end of the drill bit, where the sensors also rotate with the bit. The sensor pack include the typical sensors needed to estimate the orientation of a body, such as an inertial measurement unit, a magnetometer, and three types of strain gauges. The bend measurements are only taken in the bit toolface and bit anti-toolface directions.

Traditionally, sensors are located in the portion of the bottom-hole assembly, which is several feet (or meters) behind the bit (hereafter referred to as the bottom-hole assembly or BHA). In this task, however, while such sensors may exist, the bit is blind to them. Both the bit and the BHA sensor packages may send data up-hole, but the rate of transmission is extremely limited and largely unidirectional.

We set out to formulate the state estimator to work with either a bent sub or a rotary steerable system. Additionally that it workedonly with the information available in the bit's own data control volume (see Figure 2.5). The focus for this phase was the bit data control volume (marked in red in Figure 2.5). Other data volumes included the BHA along with its sensor pack and processoe, additionally, the up-hole data volume, which has information such as, the rate of penetration, the drill string angular rate, this is the desired rotational control speed RPMs, and the desired weight on bit (WOB).

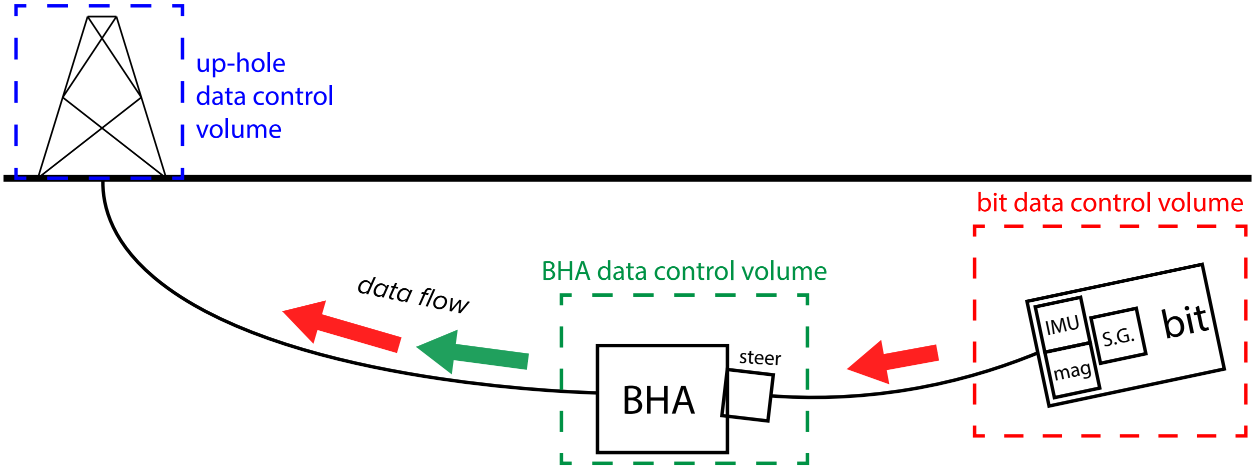


Figure 2.5: The focus for this phase was the bit data control volume (marked in red). Additional data volumes included the BHA along with its sensor pack and processor (green), additionally, the up-hole data volume (blue), which has information such as, the rate of penetration, the drill string angular rate, this is the desired rotational control speed RPMs), and the desired weight on bit (WOB).

In some cases, the drill string and bit may have a mud motor between them, such that the bit-terminus of the string has one rate of rotation and the bit itself has a different rate. A conceptual design was identified (Figure 2.7) which aims to permit either the bent sub or the rotary steerable variants to be estimated, albeit with differences in observability (Figure 2.8).

Additionally, a simulation for testing and tuning of a new filter was designed. The simulation included dynamics which are distinct from the formulation used in the filter.

The system diagrams illustrated in Figures 2.6 and 2.7 show a trade between simplicity of the model and maximal use of available information. Misalignment frames are not shown, but are included for the IMU and magnetometer, as well as for the steering assembly in the bent sub case, PCPF is the planet-centered, planet-fixed frame.

The proposed solution shown in Figure 2.7 requires two additional states to be estimated. Note that the J frame is roughly symbolic of the terminal end of the drill string, which differs from the bit frame K only by the mud motor's induced rotation along the drillbit's z axis; in the absence of a mud motor, these two frames coincide. Additionally, the steering frame S differs from B only by

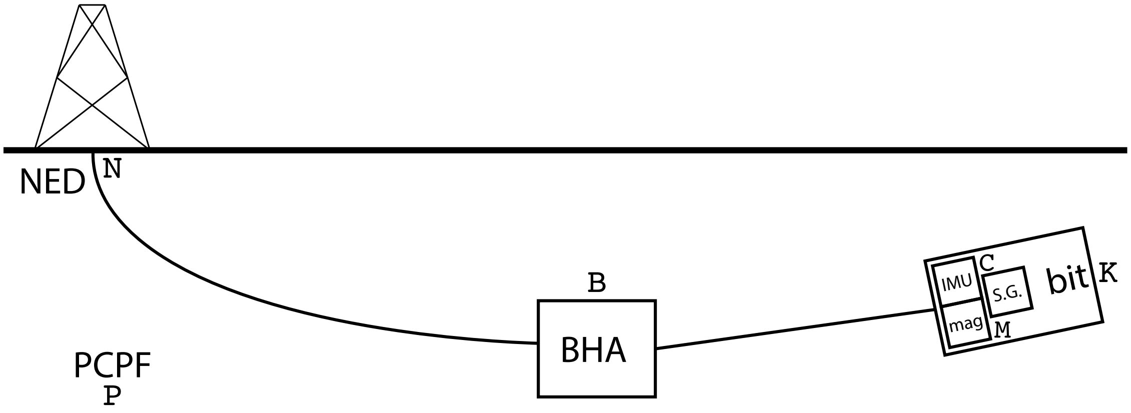


Figure 2.6: An initial design provides observability for the bit attitude as well as that of the BHA. Only two full attitude states need to be estimated. The B frame represents the BHA's center of gravity, and K represents the bit's. The C and m frames are for the IMU and magnetometer, respectively; these are fixed relative to K. The B frame differs from the north–east–down (NED) frame N by an inclination I, azimuth A, and toolface angle T, as well as an estimated distance.

an inclination — a known value when using a bent sub, but otherwise unknown. Frames S and J differ only by the induced torsion and bending in the drillstring along the length between the BHA and the bit; compressive distortion is ignored. The BHA and steering frames tend to differ by an inclination, which is more or less constant for a bent sub but could vary in the presence of a rotary steerable system (RSS).

2.2.2 State Estimation Design Trades

It was decided to use idealized dynamics when possible, relying instead on measurements, so that the proposed estimator works regardless of the drilling medium and specific drill bit characteristics. This strategy is common in aircraft navigation (e.g., in the space shuttle), where the dynamics of the atmosphere are not easily modeled [100].

2.2.3 Observability Analysis

A sensitivity analysis was performed on an early selection of states (for these shown in Figure 2.7). The essence of this test is the observability gramian, which is the matrix of the state-to-state

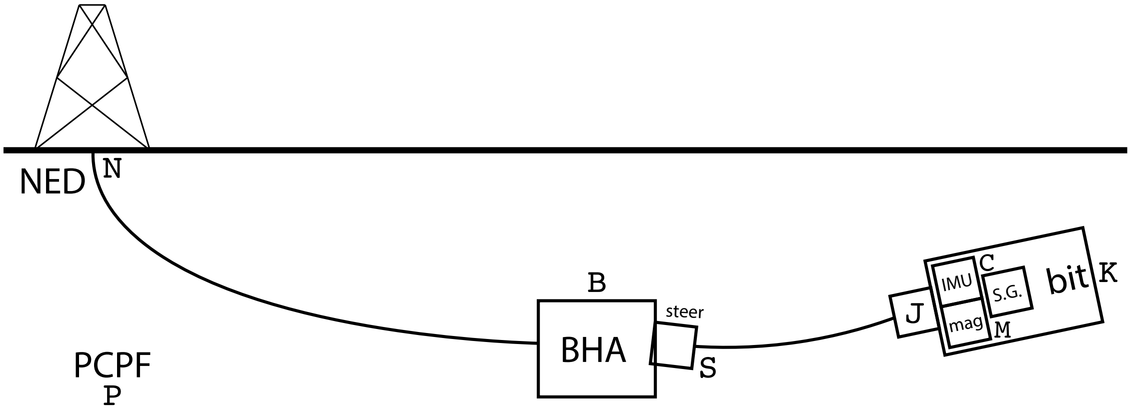


Figure 2.7: The proposed solution requires two additional attitude states to be estimated. Each labeled component represents a reference frame. Note that the J frame is roughly symbolic of the terminal end of the drill string, which differs from the bit frame K only by the mud motor's induced rotation about the bit's z axis; in the absence of a mud motor, these two frames coincide.

product.

$$\mathcal{O} = \sum_{i=k-N}^{k} \Phi(k,i)\,\mathrm{H}_i^{\mathsf{T}}\mathrm{R}_i^{-1}\mathrm{H}_i\,\Phi(i,k)\,, \tag{2.7}$$

Where the state transition model is approximated to a first degreee order as:

$$\Phi(j,i) \approx \mathrm{I} + \mathrm{F}\Delta t\,, \tag{2.8}$$

With $\Delta t = t_j - t_i$. In the equations 2.2.3, and 2.2.3 above, F is the Jacobian dynamics, H_i is the measurement sensitivity matrix for measurement i, and R_i is the measurement covariance. Please note that R_i was defined as, $\mathrm{R}_i = \mathrm{I}$, for the purposes of this test. Lastly, the magnetometer measurement sensitivity matrix H_{mag}, was of the form:

$$H_{\mathrm{mag}}^{\mathsf{T}} = \left[\begin{array}{cc} & \tilde{B} \\ {}_{P}r_{IMU} & \frac{\partial y}{\partial {}_{P}r_{IMU}} \\ {}_{P}v_{IMU} & 0 \\ \phi_{K} & \frac{\partial y}{\partial \phi_{K}} \\ \phi_{B} & \frac{\partial y}{\partial \phi_{B}} \\ \omega_{B} & 0 \\ \phi_{M} & \frac{\partial y}{\partial \phi_{M}} \\ \phi_{C} & 0 \\ {}_{B}B_{ds} & \frac{\partial y}{\partial {}_{B}B_{ds}} \\ {}_{M}b_{mag} & I \\ {}_{C}b_{acc} & 0 \\ {}_{C}b_{gyro} & 0 \end{array}\right] \tag{2.9}$$

However, treated the first term, which is likely negligible (see subsection 2.2.4, and Eq. 2.2.3), as zero. We determined that position observability — and possibly also velocity — was likely to be problematic, even with the inclusion of the IMU measurements (which are of similar form). However, the relative attitude of the bit is observable, as the relative attitude of the BHA. Moreover, the absolute attitude of the bit is highly observable. Moreover, the strain-related relative attitude of the bit is observable, as these measurements are direct.

2.2.4 Selection of Estimated States

It is worth noting that not all states are directly observable; for example, we include an estimate of the position since the last pause, for which there are no direct measurements. We also include velocity in the state, which is only directly observable when it is zeroalso referred to in the literature as a zero-velocity update.

A careful study of available measurements yielded a state vector, shown in Table 2.2.4 which provides as much information as possible about the bit's behavior. It is common to include unobservable Gauss–Markov biases (b_*) to absorb some of the errors. The BHA attitude relative to the

Table 2.1: Definition of the state vector measurements.

State	Description
${}_{\mathrm{P}}\mathrm{r}_{\mathrm{IMU}}$	IMU position
${}_{\mathrm{P}}\mathrm{v}_{\mathrm{IMU}}$	IMU velocity
ϕ_{K}	Bit attitude
ω_{K}	Bit angular velocity
$\delta\theta_{\mathrm{K}}$	Scalar bit attitude about z relative to J
ϕ_{J}	Torsion/bending attitude relative to S
ϕ_{S}	Steering attitude (RSS) or misalignment (bent sub)
ϕ_{B}	BHA attitude
ω_{B}	BHA angular velocity
ϕ_{M}	Magnetometer misalignment relative to K
ϕ_{C}	IMU misalignment relative to K
${}_{\mathrm{B}}\mathrm{B}_{\mathrm{ds}}$	Magnetic field due to drill string in BHA frame
${}_{\mathrm{M}}\mathrm{b}_{\mathrm{mag}}$	Magnetometer bias in M frame
${}_{\mathrm{C}}\mathrm{b}_{\mathrm{acc}}$	Accelerometer bias in IMU frame
${}_{\mathrm{C}}\mathrm{b}_{\mathrm{gyro}}$	Gyroscope bias in IMU frame

bit can most likely be observed, or at least partially observed, using the magnetization of the drill string. A number of states, such as ϕ_{S} and ϕ_{J}, are included to process strain gauge measurements, and though they may be unobservable, we chose to include them so as to attempt to extract as much information as possible about the behavior of the drill string using the strain gauges.

Observability of Position

Flying a spacecraft in the absence of direct position and velocity measurements (e.g., through GPS) is usually referred in the literature as *dead reckoning* [101]. Rather than including a sophisticated atmospheric dynamics model, the position, velocity and attitude estimates are reckoned using the accelerometer and gyroscope. These estimates are treated with suspicion, but are nevertheless often useful. A similar technique was incorporated into the drill bit state estimator described in this section.

Futhermore, position is assumed to also be *indirectly* observable via synthesis of the magnetometer with the accelerometer. That is, the accelerometer is used to accumulate a velocity, which is in turn used to accumulate a rough position estimate. Because the observed earth magnetic field vector is a function of position as:

$$\mathrm{B}(\mathrm{r}) \;=\; \frac{\mu_0}{4\pi}\left(\frac{3}{r^5}\mathrm{r}\left(\mathrm{m}^{\mathsf{T}}\mathrm{r}\right) - \frac{1}{r^3}\mathrm{m}\right), \tag{2.10}$$

$$\tag{2.11}$$

So that the $\frac{\partial \mathrm{B}}{\partial \mathrm{r}}$ term of the measurement sensitivity matrix H will be nonzero:

$$\frac{\partial \mathrm{B}(\mathrm{r})}{\partial \mathrm{r}} \;\propto\; \frac{3\mu_0}{4\pi \mathrm{r}^5}\left(\left(\mathrm{m}^{\mathsf{T}}\mathrm{r}\right)\mathrm{I} + \mathrm{r}\mathrm{m}^{\mathsf{T}} + \mathrm{m}\mathrm{r}^{\mathsf{T}} - \frac{5}{r^2}\left(\mathrm{m}^{\mathsf{T}}\mathrm{r}\right)\mathrm{r}\mathrm{r}^{\mathsf{T}}\right). \tag{2.12}$$

Thereforce, measurements of the magnetic field may also provide position information.

A similar term, $\frac{\partial \mathrm{g}(\mathrm{r})}{\partial \mathrm{r}}$, exists in the gyrocompassing [1] measurement model, because of the magnitude and direction of the gravitational field weakly depend on the position. However, since gyrocompassing is only available when the bit is immobile, e.g., when gravity and gyroscope measurements are both static, it is unlikely that IMU updates can provide position information alone.

In summary, the position is likely only minimally observable, but we determined to include it for the sake of any information about bit behavior that might be gleaned. In the end, the position may be dropped from the filter formulation, and velocity retained only to understand the short-term behavior of the bit. The states correspond to those given in Figure 2.7; the hats indicate that the states are estimates.

The lines represent observability between two frames, with the caveat that the magnetometer only provides a partial attitude estimate. Other lines represent small and large unobservable (or minimally observable) relationships and modeling errors, respectively. The grey line is an unknown, probably depending on the bit's and BHA's attitudes relative to the magnetic field; moreover, this error is also subject to other minimally observable relationships. Note also that strain

[1] Gyrocompassing is defined here as the use of the gyroscope to measure the rate of the earth's rotation while simultaneously determining the direction of gravity using the accelerometer. This measurement model is available only when the bit is immobile.

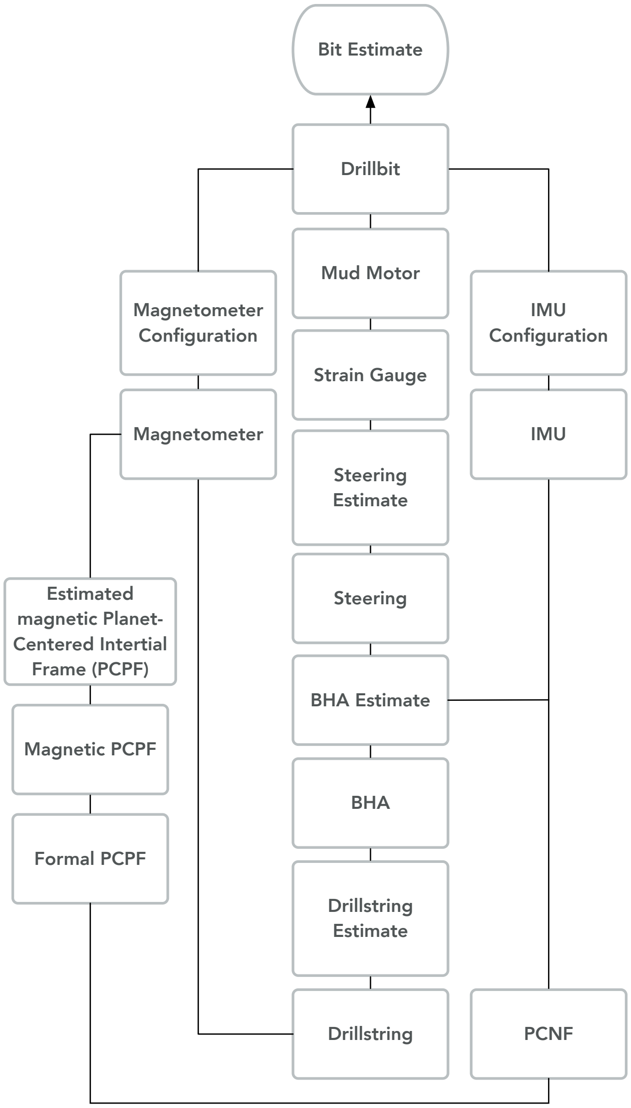


Figure 2.8: Schematic of Observability relationships. The observability of frame orientation relationships by the bit sensors can be visualized as a graph..

gauge measurements are bottlenecked by the minimally observable K–Ĵ attitude. In summary, the bit attitude is easily observable via the right-hand pathway from K (bit) to I (planet-centered inertial), but the estimation of the BHA attitude is more difficult.

Up-hole Processing

In case the BHA attitude is unobservable within the bit data control volume, and it therefore, unlikely that position and velocity will be observable. Much more information can be obtained by an up-hole observer. In such a case, key bit filter states are transmitted up-hole by mud pulse telemetry. A batch or sequential batch processor then incorporates into its estimate information from all three data control volumes (see Figure 2.5):

1. Filter States, especially:
    - Bit position.
    - Angular rate difference between the bit and the drill string.
2. Data available from the BHA's own measurements or filter.
3. Up-hole observations and commands.
    - Desired inclination of the steering tool or the configured inclination if using a bent-sub.
    - Rate of Penetration, v_{rop}.
    - Topdrive's RPMs, ω_z.

Naturally, it should be possible to observe the drillbit's attitude with a much more reduced amount of uncertainty to provide a much better picture of the behavior of its relative motion. Nevertheless, the incorporation of any additional data from the surface was outside the scope of the current study.

2.2.5 Simulation Models Description

It was desirable initially to make the simulation as simple as possible in order to test the filter's ability to determine bit attitude (see Figure 2.9), this is, propagating only the bit's position, attitude, and angular velocity. Input parameters should include the weight on the bit and the $\approx z$ component of the angular velocity; the bit velocity is computed as a function of the attitude and these two input parameters. Simulated measurements include the angular velocity, acceleration, and an idealized earth magnetic field. Futhermore, the bend, torsion, and compression measurements from the strain gauges will be zero.

Noise may be injected into the simulated measurements and the filter's position and velocity accuracy. The next stage was to provide additional inputs — a BHA inclination and an array of drill string angular velocities over time — in order to compute the positions and attitudes of the BHA and steering commands. At this point a drill string magnetization model and bend, compression, and torsion through the strain gauges are added in order to identify the cases where the BHA orientation is fundamentally observable. It is then possible to re-inject noise in order to determine the noise sensitivity and tune the filter. Later, a BHA model can be added, along with strain gauge measurements and a drill string magnetization model.

2.2.6 Drill Plan

The drill plan class defines the trajectory of the well to be drilled. The plan allows the trajectory to be set by distance markers while the propagation takes place in the time domain. The plan is composed of a series of segments where each segment defines a rotate or slide procedure. An additional 'hold' procedure is provided for use when the plan has been achieved but the time of propagation has not run out. The user configures the drill plan prior to executing the simulation. A simple drill plan is described like so:

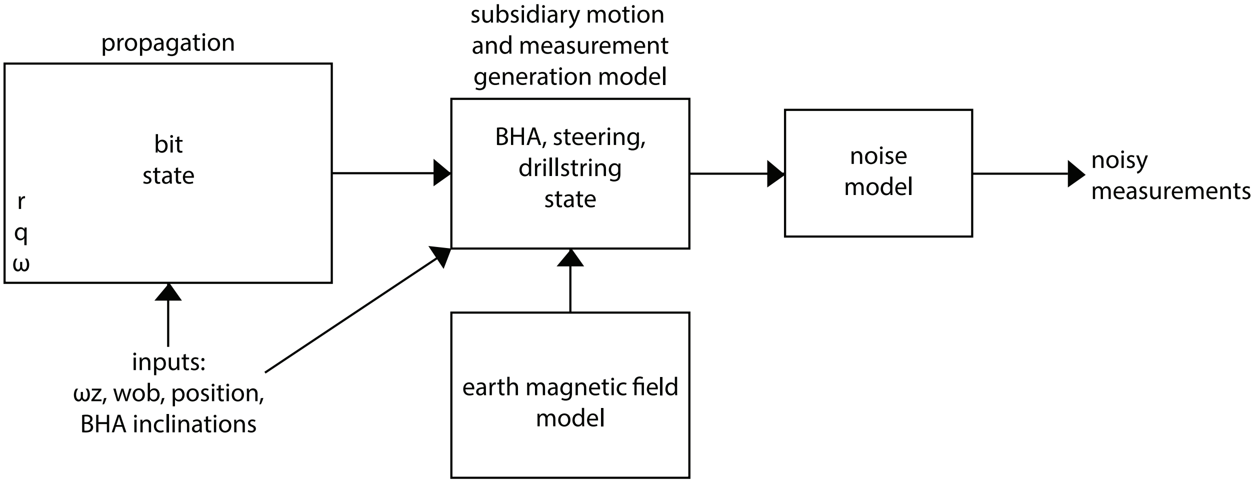


Figure 2.9: Initial simulation architecture used for navigation propagation. Only six states are propagated in the simulation, which is controlled only by the weight-on-bit and the drill bit angular velocity.

```
plan = DrillPlan ()
plan. rotate (90)
plan. slide (15,45)
plan. rotate (90)
plan.hold()
```

This plan (as shown in Figure 2.10) can be interpreted as follows:

1. Initialize a drill plan.
2. Drill in rotation mode for a distance of $\approx 90ft$.
3. Drill in sliding mode until an angle of $\approx 15\,\text{deg}$ has been achieved using a toolface angle of $\approx 45\,\text{deg}$.
4. Drill in rotation mode for a distance of $\approx 90ft$.
5. Hold the current position until the propagation completion time is reached.

2.2.7 State Vector Definition and Propagation

The propagated state vector is given in Table 2.2, and the derived states in Table 2.3. Some of these states are constant in the current version but can be changed in future versions.

The architecture chosen for the ODE solution integrates the equations in time. Time integration was chosen because time histories are desired in order for instrument measurements to be generated from the trajectory, and because the rotation rate of the bit is most easily expressed as an angular rate in time. However, there are several discrete changes in the equations of motion that require special consideration. Importantly, these discrete changes are driven by distance. The equations to be numerically integrated are, effectively, functions of one or more of the dependent variables that arise from the integration (in particular, the downhole penetration distance). During each step of the numerical integration, the current distance is used to determine if any discrete changes are required to the equations.

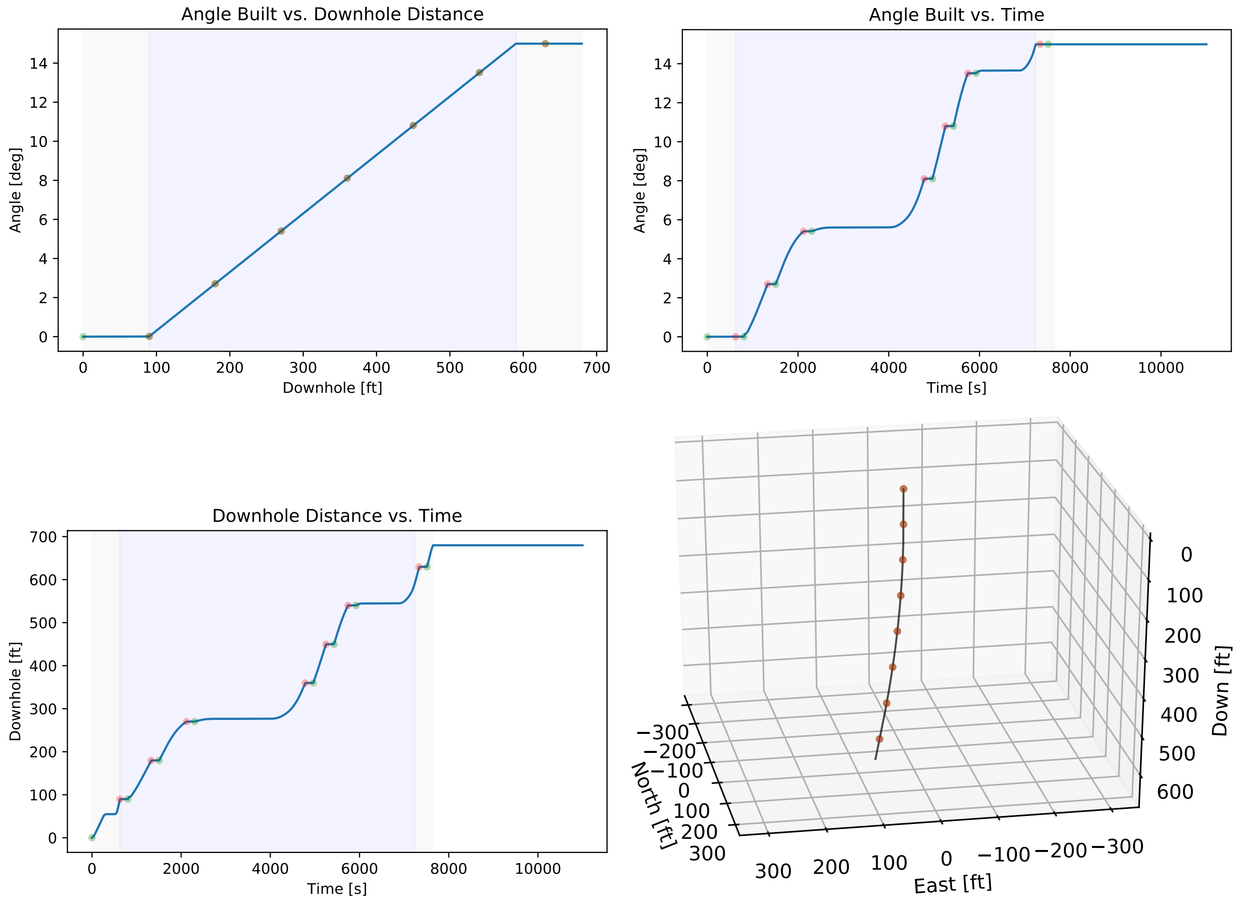


Figure 2.10: The execution of the drill plan is shown under varying weight-on-bit and bit angular velocity. The gray shaded regions represent the rotation mode, and the blue-shaded regions represent the sliding mode. The green and red dots represent the start and stop of the individual stands ($\approx 90 ft$). In the left-hand figure, we see that no angle is built during the first drill stand which is the second step in the given drilling plan (Step 2). After $\approx 90 ft$ of measured depth has been reached, the simulation switches mode to angle-building, which is held until the angle of (≈ 15 deg) is reached (Step 3). At that point, the simulation switches back to rotation mode and drills another $\approx 90 ft$ (Step 4). When the additional $\approx 90 ft$ is reached the simulation switches to 'hold' mode and no further changes are effected (Step 5). The build angle of ≈ 15 deg oriented at ≈ 45 deg from the initial position is visible in the 3D trajectory representation.

State	Description
${}_{\mathrm{N}}\mathrm{r}_{\mathrm{B}}$	BHA position in NED frame
${}_{\mathrm{P}}\mathrm{v}_{\mathrm{B}}$	BHA velocity in NED frame
${}_{\mathrm{B}}\mathrm{r}_{\mathrm{K}}$	bit position in BHA frame
d_{K}	Measured depth
ϕ_{built}	Cumulative angle built during current building phase
${}^{\mathrm{K}}_{\mathrm{B}}\theta$	Angular displacement from BHA frame to bit frame
${}_{\mathrm{B}}\omega_{\mathrm{K}}$	Bit angular velocity with respect to BHA frame
${}_{\mathrm{N}}\omega_{\mathrm{B}}$	BHA Angular velocity with respect to NED frame
${}^{\mathrm{B}}_{\mathrm{K}}\mathbf{q}$	Attitude of bit frame with respect to BHA frame
${}^{\mathrm{N}}_{\mathrm{B}}\mathbf{q}$	Attitude of BHA frame with respect to NED frame

Table 2.2: The sim includes the above states, propagated by numerical integration of ODEs defining the time-rate of change of the state.

State	Description
${}^{\mathrm{N}}_{\mathrm{K}}\mathbf{q}$	Attitude of NED frame with respect to bit frame
${}_{\mathrm{P}}\mathrm{r}_{\mathrm{K}}$	Position of bit relative to PCPF (derived from ${}_{\mathrm{N}}\mathrm{r}_{\mathrm{K}}$ and latitude/longitude of NED origin)
${}_{\mathrm{N}}\mathrm{v}_{\mathrm{K}}$	Velocity of bit relative to NED (derived from numerical differentiation of ${}_{\mathrm{N}}\mathrm{r}_{\mathrm{K}}$)
${}_{\mathrm{N}}\mathrm{a}_{\mathrm{K}}$	Acceleration of bit relative to NED (derived from numerical differentiation of ${}_{\mathrm{N}}\mathrm{v}_{\mathrm{K}}$)

Table 2.3: Several states are derived from the propagated states, but are not themselves directly propagated.

Additionally, an alternate propagation architecture was implemented – derived an equation for the time expired as a function of distance drilled and then integrated all the equations in distance. Then the time histories of the measurements could be reconstructed in post-processing. However, this approach was not pursued because of complications associated with expressing the rotation rate of the bit (which drives the kinematics) as a function of distance.

Stand Connection Events

Shown in Figure 2.2.7 is the temporary hold in downhole rate of penetration (ROP), and distance during the delay. The green dots indicate the start of a stand and the red dots indicate the end of a stand. The figure uses a constant rate of penetration to simplify the interpretation of the downhole progress.

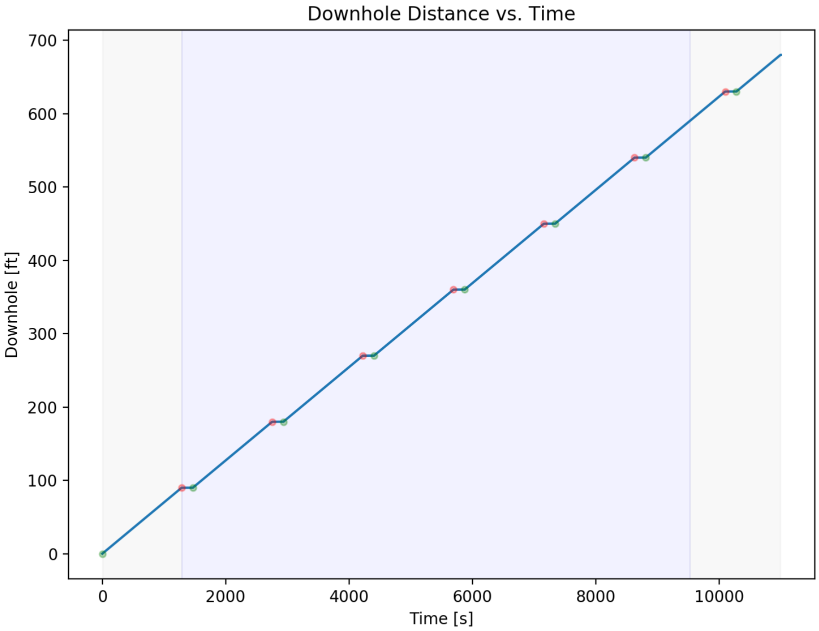


Figure 2.11: Temporary holds may be induced during stand connection events. The green dots indicate the start of a stand and the red dots indicate the end of a stand. The figure uses a constant rate of penetration to simplify the interpretation of the downhole progress.

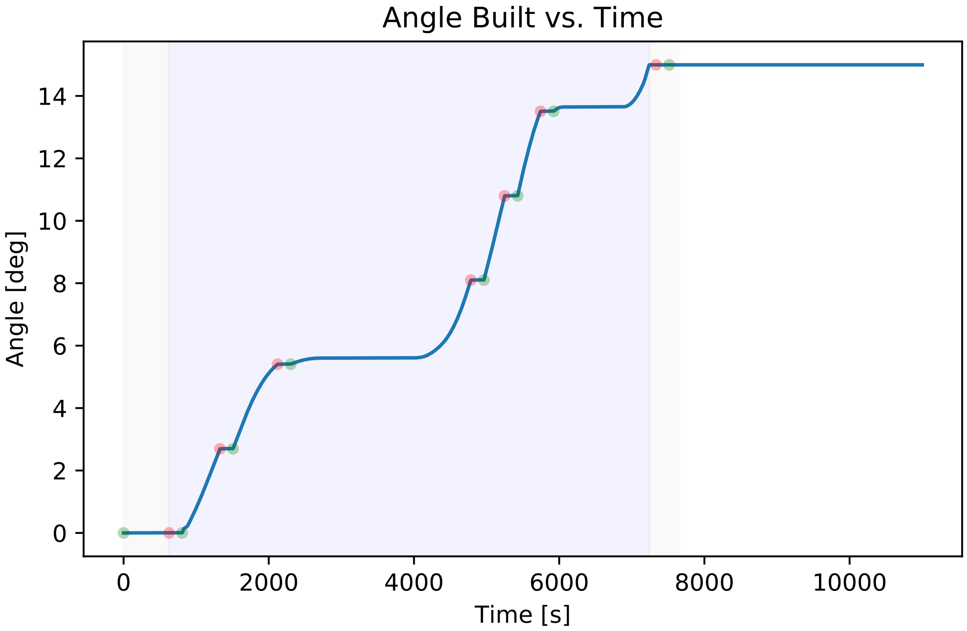


Figure 2.12: The simulation models changing drilling mode between straight bore (rotation) and steering (sliding). The gray shaded region represents the rotation mode, the blue shaded region represents the building (or sliding) mode. The figure uses a constant rate of penetration to simplify the interpretation of the angle building progress.

The simulation models the temporary stop in penetration as pipe stands are added to the drill string (Figure 2.2.7). A configurable input defines the length of pipe stand that can be propagated before a temporary hold in progress is applied, i.e., while a new stand is connected. The delay between stands is user configurable as well.

Steering Modes

A change in drilling mode from a straight bore to a steering mode as well as the reverse (Figure 2.12). A user-defined drill plan determines the distance to proceed in a fixed direction or the angle to be built at a defined toolface angle. The distance advanced and cumulative angle built are monitored during the propagation and mode changes are implemented according to the drill plan.

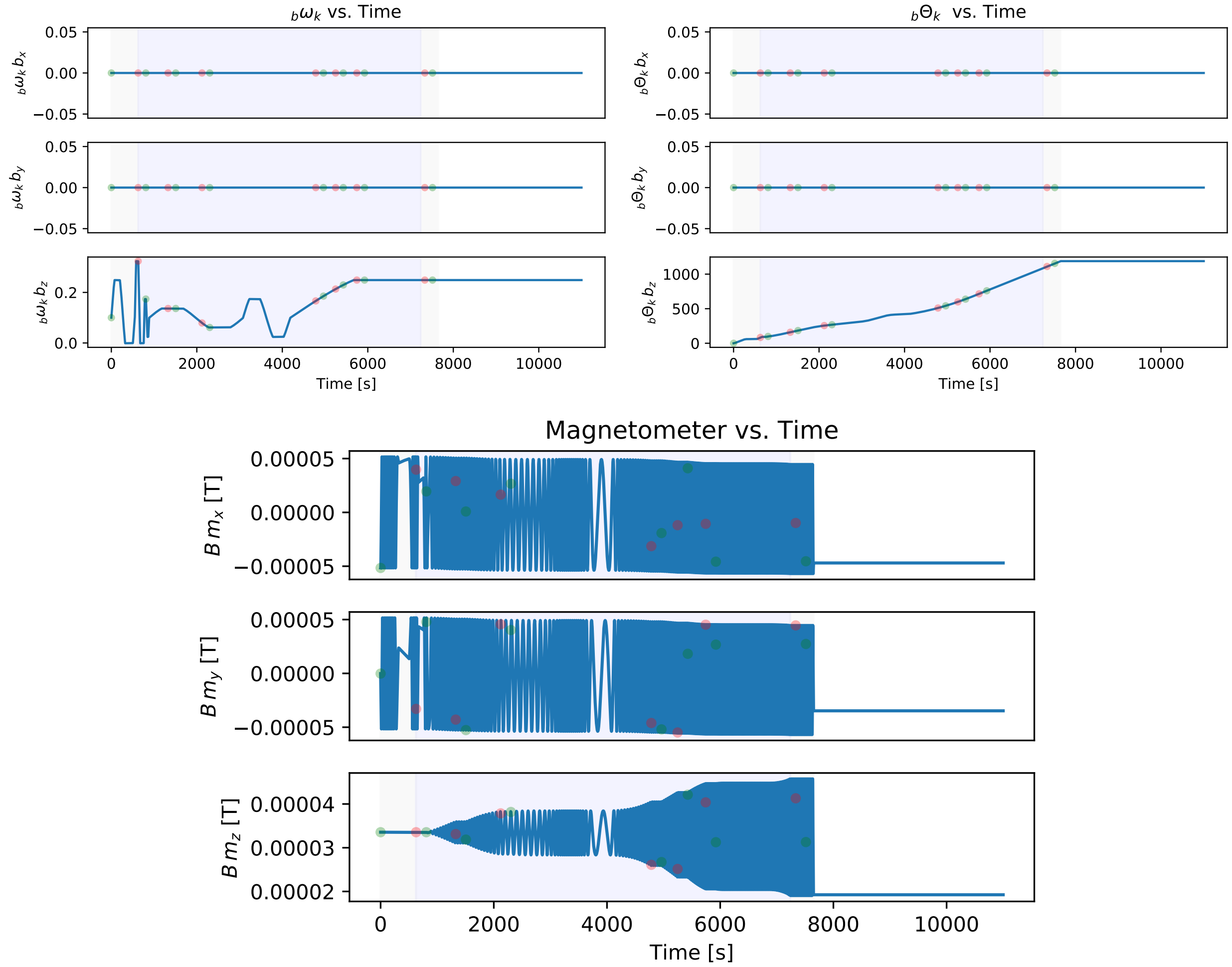


Figure 2.13: IMU and magnetometer measurements are shown, generated from the previously described simulation plan.

2.3 Incorporating Virtual Sensor's Measurements Into the Simulation.

The following sensor models were directly used by the bit motion estimator: the magnetometer, gyroscope, and accelerometer, all shown in Figure 2.13. The vibrometer model is a variant of the accelerometer model and was used to drive environment monitoring and estimation routines.

2.3.1 IMU

The accelerometers and gyroscopes that make up the inclinometer or IMU are common for downhole tools. They provide location relative to the bit's center of mass, and in some cases, an inclination reading relative to the bit. When drilling is stopped, the bit may continue to rotate due to the pumping of mud through the mud motor; however, acceleration due to gravity is clearly detectable, and when the bit ceases rotating, the identification of an east vector (the earth angular velocity vector) should permit 3-DOF attitude determination.

2.3.2 Magnetometer

The magnetometer reports a vector measurement which is a linear combination of the earth's magnetic field (locally surveyed), magnetic interference from the drilling fluid and medium, and the magnetic field originating in the drill string behind the BHA.

The magnetometer measurement model can be written as:

$$ {}_{\mathrm{M}}\tilde{\mathrm{B}}_{\mathrm{mag}} = \delta\mathrm{E}\,{}^{\mathrm{M}}_{\mathrm{K}}\mathrm{T}\left({}^{\mathrm{K}}_{\mathrm{P}}\mathrm{T}\,{}_{\mathrm{P}}\mathrm{B}_{\mathrm{local}} + {}^{\mathrm{K}}_{\mathrm{B}}\mathrm{T}\,{}_{\mathrm{B}}\mathrm{B}_{\mathrm{drillstring}}\right) + {}_{\mathrm{M}}\mathrm{b}_{\mathrm{mag}} + \nu\,, \tag{2.13} $$

Where the magnetic field B terms 'local' and 'drillstring' describe the earth magnetic field and the near-BHA drillstring term, the magnetometer bias term is given by $\mathrm{b}_{\mathrm{mag}}$ in the magnetometer frame, and ν is zero-mean white noise. This expression includes a term $\delta\mathrm{E}$ which represents magnetometer modeling errors such as misalignment, nonorthogonality, and scale factor.

2.3.3 Strain Gauges

The strain gauges report torsion about the drill string axis, axial compression, and bend in the drill string in the toolface or anti-toolface direction. All cases that use the *x* or *y* component of the moment of inertia tensor are described as:

$$J_x = J_y = \frac{\pi}{64}(d_o^4 - d_i^4)\,, \tag{2.14}$$

Where d_o is the outer diameter of the pipe and d_i is the inner diameter. We treat the bit as solid and define its cross-section moment as:

$$J_{\text{bit}} = \frac{\pi}{64}(d^4)\,. \tag{2.15}$$

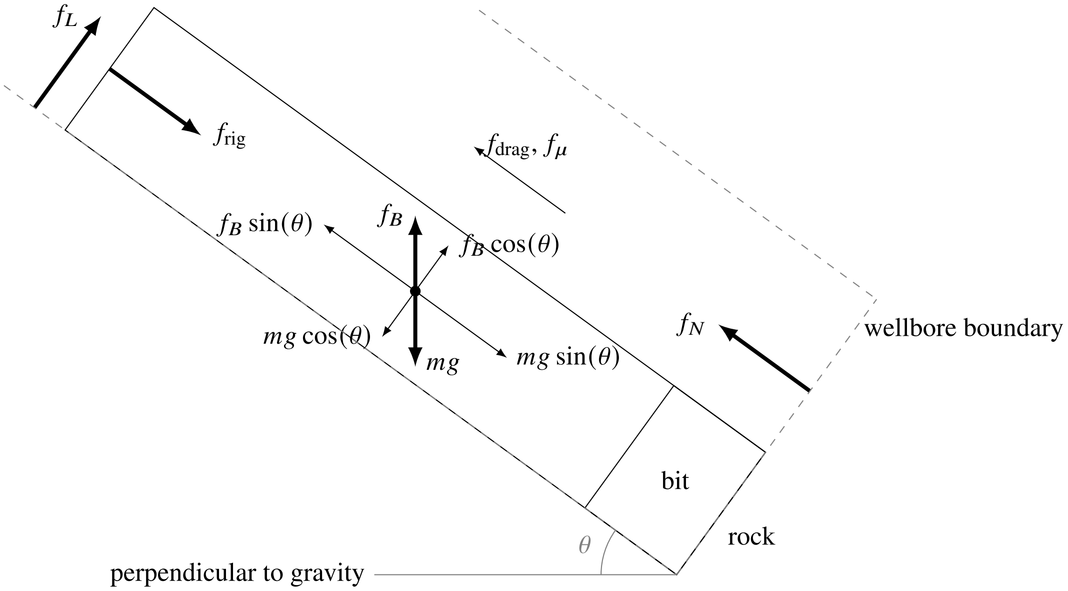


Figure 2.14: Conceptual force diagram used to define the strain gauge measurements.

The force diagram is shown in Figure 2.14. The net forces can then be written as:

$$\sum f_x \;= 0 \;= f_L \sin\gamma + f_B \cos\theta \sin\gamma - mg \cos\theta \sin\gamma \tag{2.16}$$

$$\sum f_y \;= 0 \;= f_L \cos\gamma + f_B \cos\theta \cos\gamma - mg \cos\theta \sin\gamma \tag{2.17}$$

$$\sum f_z \;= 0 \;= f_N - mg \sin\theta - f_{\text{rig}} + f_{\text{drag}} + f_\mu + f_B \sin\theta\,, \tag{2.18}$$

And the equations for the bit alone are similar. We can define f_N as the reaction force at the bit that is is approximately equivalent to the weight on bit, thus, write them as:

$$f_N = f_{\text{wob}} = mg \sin\theta + f_{\text{rig}} - f_{\text{drag}} - f_{\mu} + f_B \sin\theta \tag{2.19}$$

$$\approx mg \sin\theta + f_{\text{rig}} \, , \tag{2.20}$$

Assuming the lateral forces in x and y are much smaller than the axial force, and that the drag, friction, and buoyancy are of much lower magnitude than the weight-on-bit. Additionally, f_{rig} differs from f_{wob} in that the former includes the weight of the drill string, BHA, and other components.

2.3.4 Axial

Using Hooke's law to calculate axial strain, with Young's modulus E as the spring constant and normal strain ϵ_z as the normal strain; the normal stress is:

$$\sigma_z = E\epsilon_z \, . \tag{2.21}$$

Assuming the bit is always under compression:

$$\sigma_z = -\frac{f_{\text{wob}}}{\pi r_0^2} = -\frac{4 f_{\text{wob}}}{\pi d^2} \, , \tag{2.22}$$

Then, substituting back into Hooke's law to obtain the expression:

$$\epsilon_z = -\frac{4 f_{\text{wob}}}{E \pi d^2} \, . \tag{2.23}$$

2.3.5 Torsional

A simple model for the torque acting to induce distortion in the bit was used to simulate the downhole disturbance conditions.

$$
\begin{aligned}
\tau &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_{\text{rig}} - \tau_{\mu_s} - f_{\mu_r} r_0 \end{bmatrix} && (2.24) \\
\tau_{\mu_s} &= \frac{2}{3} \mu_s f_{\text{wob}} r_0 && (2.25) \\
\tau_{\mu_r} &= \mu_r f_{\mu_L} && (2.26)
\end{aligned}
$$

Where τ_{rig} is the torque on the bit from the rig, τ_{μ_s} is torque on the drilling surface of the bit due to the medium, f_{μ_r} is rolling friction on the bit with the borehole contact, and r_0 is the bit radius. Then it can be written as:

$$\tau_{\text{rig}} = \frac{2}{3} \mu_s f_{\text{wob}} r_0 + \mu_r f_L \,, \tag{2.27}$$

Where f_L is rolling friction for the bit. Thus, the shear stress on the bit's surface can be written as:

$$\tau_{\text{max}} = \frac{\tau_{\text{rig}}}{J_{\text{bit}}} \,, \tag{2.28}$$

Using Hooke's law with the modulus of rigidity G as the spring constant and shear strain γ_z as the displacement:

$$\tau_{\text{max}} = G \gamma_z \,. \tag{2.29}$$

Finally, the expression for the torsional displacement is described as:

$$\gamma_z = \frac{1}{G J_{\text{bit}}} \left(\frac{2}{3} \mu_s f_{\text{wob}} r_0 + \mu_r f_L \right) \,. \tag{2.30}$$

2.3.6 Bending

For simplicity, it is assumed that f_{rig} acts parallel to the bit axis. The moment about the bit due to bending therefore is:

$$M = -f_{\text{rig}}e \tag{2.31}$$

Where e is the eccentricity (or bend displacement).

Rewriting Eq. 2.20 in terms of f_{rig} yields:

$$f_{\text{rig}} = f_{\text{wob}} - mg\sin\theta\,, \tag{2.32}$$

noting that $mg\sin\theta$ represents the entire BHA, not just the bit; plugging in in to get the moment,

$$M = -(f_{\text{wob}} - mg\sin\theta)e\,. \tag{2.33}$$

According to the flexure formula, bending stress is defined as:

$$\sigma = -\frac{My}{J_{\text{bit}}}\,, \tag{2.34}$$

Which allows for defining:

$$\sigma_{\max} = \begin{cases} +\frac{Mr_0}{J_{\text{bit}}} & \text{on side of tension} \\ -\frac{Mr_0}{J_{\text{bit}}} & \text{on side of compression}\,. \end{cases} \tag{2.35}$$

Using Hooke's law, it can be written:

$$\epsilon_{\max} = \pm\frac{((f_{\text{wob}} - mg\sin\theta)\,e)\,r_0}{J_{\text{bit}}E}\,. \tag{2.36}$$

The eccentricity term, e can then be made to vary sinusoidally to mimic the drill string motion.

2.4 Integration of a Real-time Drilling Simulator

Simulation of drilling processes involves the seamless integration of strongly coupled physics, numeric, and economic models. A similar overview of mathematical models as that shown in the previous section can easily be done for each major drilling subprocess that influence drilling a wellbore. The models discussed previously usually lack an exact analytical solution, and require iterative numerical methods to compute results. Futhermore, the numerical approximation is not always accurate to reproduce the exact behavior of the drillstring because of the influence trasient behaviours while drilling.

Moreover, non-linear mathematical models of drilling dynamics have been used to support the planning phase of operations, building specialized software suites to support drilling engineering activities. However, more work is needed to move their use from a planning perspective into operations. There is a need for solutions that can be fast enough to adapt to real-time changes in the drilling process; operators desire advanced drilling simulation software that can mimic reality and yet be practical for daily operations.

At the same time, academic researchers are devoting time and effort to develop robust and accurate models of every module of an integrated drilling environment but are challenged with translating their original contributions into the operational setting.

The ongoing need for an integrated training simulation environment of the drilling process was a fundamental challenge for the development of a real-time drilling simulator. This section discusses work related to the integration of a real-time horizntal directional drilling simulator that integrates analytical, linearized, and non-linear drilling models. Including, drill string dynamics, torque and drag analysis, geologic modeling, hydraulics, and wellbore pressure modeling, visualization, and operator interfaces. The simulator was written in C++ and built on top of high-performance open-source libraries that leverages parallel computing, GPU acceleration, multi-body, and computational fluid dynamics solvers as well as other tools. The simulator core was successfully developed and has been used to model horizontal drilling while providing near real-time performance. The framework has been used in several applications successfully, including modeling shallow horizontal jetting, rock drilling, and steering; as well as reconstructing spatial data from high fidelity downhole signals for borehole evaluation, and the development of improved downhole navigation algorithms, also discussed in this work in Chapter 3, and Chapter 4. In the view of the author, publications related to numerical simulations of drillstring are relatively scarce in the literature, as most research has been focused on analytical methods or experimental tests. Futhermore, the author believes that a collaborative development effort between academia and industry will bring solutions to complex modeling needs faster and more economically than any individual endeavor.

The concept started with the development of a Horizontal Directional Drilling (HDD) trainer for a horizontal drilling operator in the summer of 2017. A brief description of it is main components and capabilities are described in the next section.

The main objective of the simulation was to model a 2000' horizontal pilot hole drilling operation while providing closed-loop feedback to unexpected events that might result from bad drilling decisions. Even though the complexity of an HDD operation is lower than what we observe in oil horizontal drilling, the underlying principles and physics are at the same time. Figure 2.15 shows the Bottom Hole Assembly (BHA) configuration used in this scenario, with hardware-in-the-loop human control interfaces with real-time downhole drilling models to simulate the physics of drilling and jetting through.

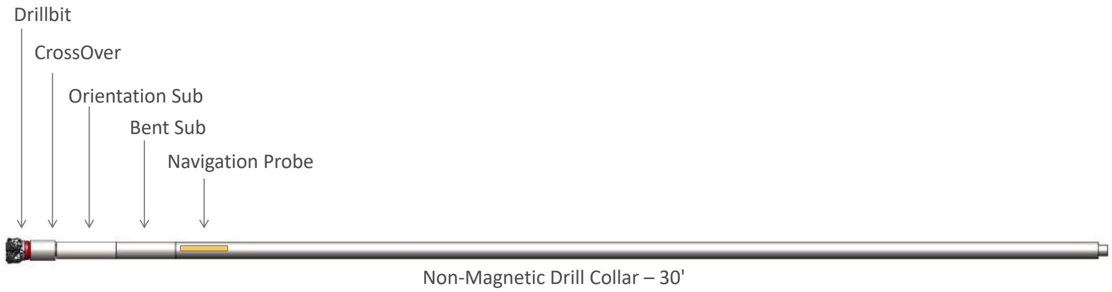


Figure 2.15: Bottom hole assembly configuration used in the development of the drilling simulator.

The underlying objective was to research, model, and implement in real-time the underlying physics to drill through rock and sand. Including deriving the drill bit and probe orientation, rate of penetration, build rate, drillstring dynamics with and without deformation, borehole debris generation and circulation profile, probe measurements. Figure 2.16 shows a snapshot of the simulation of a dynamic drillstring model that was developed initially to simulate deformation dynamics of the drillstring using multi-body simulation.

2.4.1 Overview of Developed Drilling Modules

The simulator was written in C++ taking advantage and leveraging robust open-source tools that have been developed in other industries and academic disciplines over the past years. Two important developments were the use of Chrono, a multi-physics simulation engine to model the rig, drillstring, and other rigid-body dynamics. The main capabilities of the multi-physics engine used are:

- Multi-body Dynamics. Able to run simulations of mechanisms made of rigid bodies. Apply a constraint to parts using a wide set of joints. Add motors, linear actuators, springs, and dampers while applying forces and torque.
- Finite Elements. Create finite elements, and model flexible parts including beams, cables, and shells. Apply local or distributed loads. Perform non-linear analysis with large deformations.
- Large Scale Simulation. Simulate large scenarios such as granular flows, vehicle-soil interaction, and fluid-solid interaction. Perform co-simulation with other CFD and FEA software packages.
- Collision Detection. Define collision shapes using meshes or primitives. Compute frictional contact forces using automated collision detection algorithms. Define surface properties and surfaces [102, 103]).

As well as modeling the dynamics of the jet erosion and consequential borehole propagation through the geological formation properties. This model will be implemented using OpenVDB, an open-source C++ library comprising a novel hierarchical data structure. It is a hierarchical data structure for the efficient representation of sparse, time-varying volumetric data discretized on an efficient 3-D grid system capable of performing several Lagrangian and Eulerian-based methods. Developed by DreamWorks Animation for use in volumetric applications typically encountered in feature film production and is now maintained by the Academy Software Foundation (ASWF) [104]). Their work takes advantage of parallelization techniques such as vectorization or GPU computing using CUDA [105, 106].

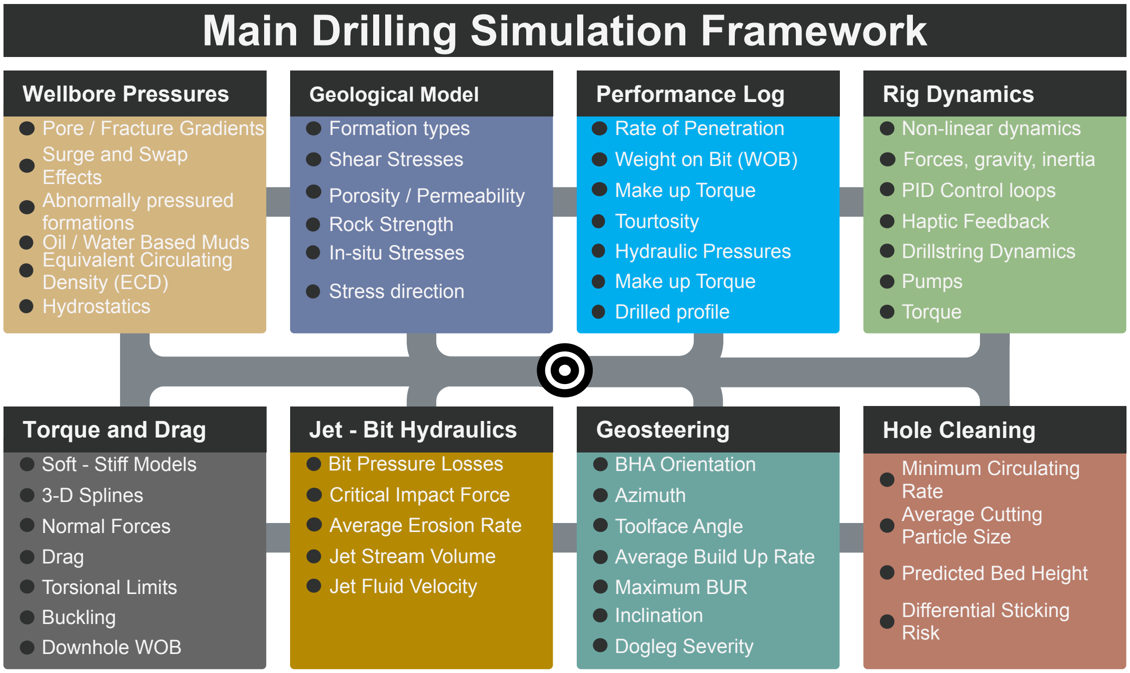


Figure 2.16: Modularized overview of the drilling models developed and implemented with several running concurrently with the main graphics engine and simulation loop.

The framework shown in Figure 2.17 represents a unique case of an integration approach to modeling. The drilling and simulation engine provided was able to:

- Record and log the performance.
  - Records are kept through the entire process. Especially the rate of penetration (ROP), Weight on Bit (WOB), Make up torque (MUT), Hydraulic pressures, jetting forces,

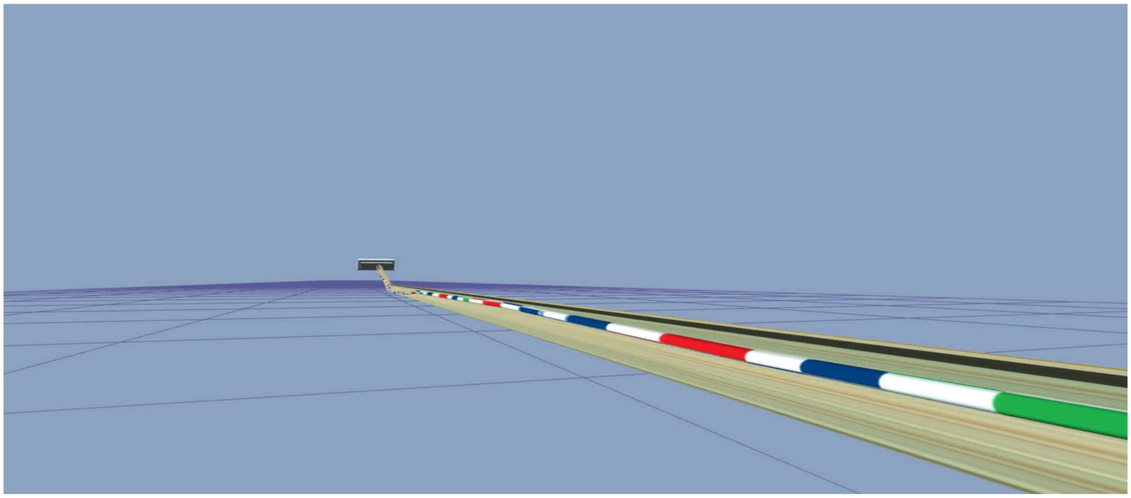


Figure 2.17: Multibody dynamic drillstring model developed initially to simulate deformation dynamics of the drillstring.

and the resultant borehole trajectory.

- Rig dynamics. Capturing the full non-linear response of the dynamics of the whole system was crucial to approximate the outcome. The rig and drill string were modeled and simulated using an open-source library [107]. This allowed providing a more realistic response of makeup torques, mud pump response, and top drive. Motors and actuators are simulated using a proportional-integral-derivative (PID) feedback loop. The benefits and trade-offs of using a multi-body dynamic solver versus a simplified model have been previously discussed [108].

- Drillstring dynamics . Compute torque and drag along the drilled trajectory using soft-string and 3D splines interpolation methods in real-time. Stiff-string models reduce the runtime efficiency as the length of the drillstring increases, although not fully utilized, were also developed and implemented.

- Jet Hydraulics. An important part of the system was modeling erosion velocity as a function of the jet impact force and geological properties of the material.

- Geosteering. The system computes the bottom-hole assembly (BHA) orientation, toolface angles, azimuth, and dogleg severity on a continuous basis using a borehole propagation model based on an eulerian description of the drilled space.

- Borehole propagation. Since drilling is done primarily through the jetting of poorly consolidated formations. Thus, emphasis was given to model the rock surface in contact with the jet stream drill and the corresponding erosion. The geometry of the jet propagation and the amount of rock removed through erosion are tracked over time. Integrating this change provides an approximation of the overall borehole shape. The model solves the intersection and union of removed volumes through time so to define the current borehole shape. Computing and integrating complex geometries per timestep can be highly intensive. Thus, the model was implemented using an open-source library that utilizes a novel volumetric approach to render the drilling operation.

The following previously published drilling models were used in the development of the simulator's components are provided below [109, 67, 110, 52, 64, 111, 112, 113, 6, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130]. Futher work is needed in the documentation, and model validation of some of the implemented models used in the integrated solution to compare performance with other models.

3. DATA-DRIVEN MODELING FOR DRILLING OPTIMIZATION

Data science is a multidisciplinary field of methodologies and processes that have become widespread due to significant strides in computational processing capacity, efficiency, and accessibility, as well as the proliferation of tools to store and process data. The most general objective of these methodologies could be stated as formulating a hypothesis, retrieving, and processing data efficiently to perform analytics for the identification of relationships and subsequent application of data-derived models for the prediction, monitoring, or learning processes of a given phenomenon [131, 132].

This area of knowledge has started changing the current status quo in several industries, including Oil & Gas upstream sector. Service companies began the standardization of protocols and processes over the last decade with several successful interoperability projects, as discussed in the previous section. Now, operators have recognized the need to collaborate on standardizing the storage, management, and transmission of records related to every component of the energy extraction process. A recent byproduct of such an effort is the launch of collaboration agreements with information technology players and funding of open standards to overcome challenges on accessing, processing, and collaborating with data streams [133, 134, 135, 136, 137, 138].

The high-speed analysis of large amounts of data is simply not attainable for human drillers. The oil and gas industry has been actively working on improving the use of real-time drilling data and computer algorithms to provide optimized drilling decisions on-site, for instance, ranging from improving the average rate of penetration, bit lifespan, and reducing the time required to reach the target depth. The implementation of the above developments were met with some roadblocks [139, 140, 141, 142, 143, 144, 145]. A group of service companies has opted for a closed and proprietary approach to drilling automation. Manufacturing and developing their own drilling rigs, wired drill-pipe technology, and operating systems in an attempt to "close the loop" with increased performance on certain wells but with limited impact and at prohibitive cost to most operators [146, 147, 148, 30, 149, 29, 150].

More studies into the utility of identifying and transmitting the type of bit dysfunction for use in a feedback control system for full-scale drilling should be done to considering the latency in determining the recognized pattern type as transmitted to the surface. To this end, a library of bit dysfunction conditions will be developed as part of this research objective. We will focus on creating a physics-driven model-based library using drilling logs from open sources as well as a numerical drill string simulation to derive and improve the performance of the identification algorithm and reduce the effects of latency for a driller to make use of crucial downhole information such as weight-on-bit, torque, and rotary speed by sending the essential downhole parameters in a single moving window information package. Furthermore, all surface drilling measurements, and some downhole drilling parameters, are to be used more effectively to control the drilling process, a technical review of the available literature, including several dysfunction physics-based detection methodologies and other data-driven algorithms. In a summary, we will build a data machine learning (ML) pipeline for each significant dataset type to preserve efficiency as the project progresses. An ML pipeline is fundamentally a mostly automated workflow to acquire, prepare, parse, train, package, validate, and deploy an ML model.

Data from tests similar to these discussed in Chapter 4.1, where a system control technique previously developed for optimizing the drilling performance of an autonomous laboratory drill rig will be examined for its applicability to a full-scale drill rig. In full-scale drilling, however, the slow data transmission rates and bandwidth characteristic of most types of long-range, through-the-earth telemetry result in a time lag between the high-speed measurement of any downhole measurement and its presentation at the surface. In addition, we will use high-frequency real-time drilling data was obtained from the recently open-sourced by Equinor, an International Energy company; the data corresponds to the Volve fields on the Norwegian Continental Shelffield[151] and collected using the Well-site Information Transfer Standard Markup Language (WITSML) – a leading standard used in the drilling industry to transfer data from the rig to the office. The standard has a series of rules and guidelines to follow incorporated into an intrinsically Extensible Markup Language (XML) framework that defines a set of practices for encoding both human and machine-readable documents. WITSML data logged on rig surface and downhole is stored in the proprietary file type. As a result, our dataset consisted of similar data in multiple formats. Initially, a parser will be built in Python for each text-based file type, and all datasets are converted into one CSV file format. Data from different kinds of logs will later be extracted and parsed to a single combined database to identify narrower training opportunity zones.

Drilling data logs from several primary sources with the most required predictors were acquired and parsed to a readable format. An initial exploratory analysis and machine learning training process will take place.

Finally, a library of bit dysfunction conditions and corresponding corrective actions for a whirl, stick-slip, axial vibrations & lateral vibrations were developed using a combination of data-driven and physics-based techniques. This section proposes a technique to overcome two main constraints - speed and bandwidth, faced by most drilling optimization systems. A short-hop EM telemetry system ensures high-speed downhole-surface communication links, and bit dysfunction libraries eliminates or reduces the need for high bandwidth. An initial design of the telemetry system, simulated models, and machine learning algorithms is presented in this chapter to aid in developing a cost-effective intelligent drilling advisory system, which can lead to improvements in safety and non-productive time of drilling operations.

3.1 A Machine Learning-Based Optimization System Prototype

The final primary objective of this research objective is to develop an advanced real-time advisory system to help drillers make more effective decisions and optimize the Rate of Penetration (ROP), thereby improving overall drilling performance. Transformational digital technologies such as distributed processing and machine learning techniques have been utilized in developing the “brains” of the system. This, combined with robust electronics and a high-speed short-hop electromagnetic (EM) telemetry system, enables the development of a downhole drilling optimization system. A finite element model based on the concept of “transfer impedance” was implemented in MATLAB to develop the Earth/Drillstring (E/D) model, which was used for validating the short-hop module prototype before performing field tests. With high-speed telemetry and processor downhole, the system could have the capability to process and analyze raw sensor signals at the bit, improving the data transmission rate, actuation delay, and response time.

Machine Learning models were determined by ensemble learning of multiple classifiers that are trained and tested on real-world drilling data. Four dysfunction types are targeted in this study - Whirl, Stick-slip, Axial vibrations & Lateral vibrations. The limitation of current drilling advisory systems in the industry is that they rely mostly on rig instrumentation data available at the surface due to lack of a reliable, consistent, and, more importantly, cost-effective downhole telemetry system that can offer high-speed and high bandwidth.

Our main contribution to overcome this limitation results from the fact that we are processing the high-frequency data downhole at the bit utilizing advanced pre-trained machine learning models to identify drilling dysfunctions in real-time. The output of the models is transmitted using a state-of-the-art high-speed EM telemetry system, facilitating a real-time advisory system for drillers at the surface. Drilling data from several wells were used to train the models using classification algorithms - Logistic Regression, KNN, Decision Trees, Random Forest, Artificial Neural Networks, and Naive-Bayes classifiers. More than 30 parameters were used in the models with Weight-on-Bit, RPM, Torque, Mechanical Specific Energy playing a major role.

The motivation behind this research can be summarized in three main pillars, the first one is drilling complexity and sensors inaccuracies, followed by a reliance on uphole processing and lastly inefficient downhole-uphole transmission systems. The first one relates to the complexity of drilling operations. Drilling is a complex process on itself that involves the monitorization of several strongly coupled dynamics under an aggressive and harsh environment that the bit endures downhole, the ability to utilize downhole information effectively is elusive; drilling optimization systems rely predominately on rig instrumentation data available at the surface due to lack of a reliable and consistent downhole telemetry system that can offer high-speed and high bandwidth available to communicate downhole conditions cost-effectively. Secondly, sensors are responsible for providing the necessary surface measurement information used for observation, monitoring, and control of any drilling operation; they provide vital readings such as hook load, drill string revolutions per minute (RPM), surface torque, pump pressure, and rate. Nevertheless, cumbersome measurement errors have been observed in the field.

Methodology Summary

Figure 3.1 provides a summary representation of the methodology used in this section. The machine learning feature extraction, parameter selection, model configuration, and creation. The main steps of the pipeline are dataset loading, preprocessing, feature extraction, parameter selection, model configuration, and creation. The open-source programming languages Python, R, and C++ were chosen based on available expertise and the availability of tools. The processes to identify and distinguish each dysfunction will be derived as follows: Drilling parameters obtained from data logs like weight-on-bit, RPM, Torque, among others, are used to obtain derived quantities of interest, for example, Mechanical Specific Energy (MSE). Finding patterns in MSE along with other parameters helps identify each dysfunction type. With this information, algorithms are written to automate the process and identify dysfunction in real-time.

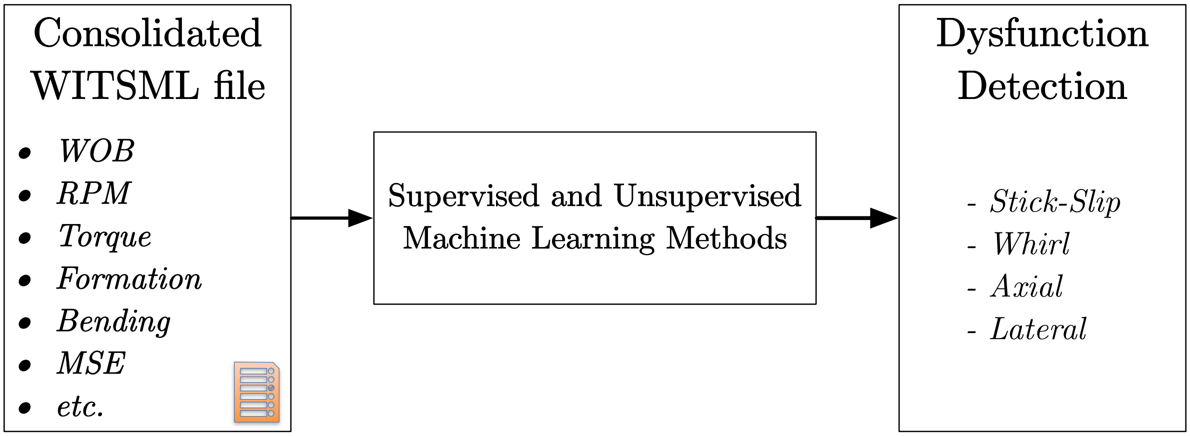


Figure 3.1: The machine learning methodology discussed in this section. Drilling parameters such as weight on bit, RPM, Torque, among others, are used to estimate derived quantities of interest, for instance, Mechanical Specific Energy (MSE), which in turn, aid in identifying and inferring the occurrence of each dysfunction type. Lastly, algorithms based on that inference were written to automate the process and identify dysfunction in real-time.

3.1.1 Optimized Telemetry System Architecture

The partitioning of computing architecture of the proposed system is primarily determined by the throughput limitations of the data communications pathways between the downhole tool components and from the downhole tool components to the surface components. Through-the-earth telemetry (TTE) carrier frequency bandwidths, and therefore, the information transmission rates are severely constrained due to the high-frequency attenuation of the lossy earth media, thus forcing more processing to be performed by the downhole computing elements prior to transmission to the surface. By combining near-bit data acquisition, high-speed telemetry, and preprocessing of downhole data with long-range, albeit slower transmission of results, and effectively greater bandwidth of useful downhole information can be transmitted from the bit to the surface for analysis by the surface data analytic application.

Figure 3.2 shows a schematic depiction of the component-level computing architecture and data communication pathways of the proposed system when deployed in drilling a well. The downhole processing component is modular, consisting of a short-hop Transmitter Sub, located near the bit and below the mud motor, and a short-hop Receiver Sub, located above the mud-motor and attached to the Electromagnetic (EM) Measurement-While-Drilling (MWD) Instrument Collar. The short-hop Transmitter Sub gathers real-time bit dynamics data and transmits this data to the Short-hop Receiver Sub via a high-speed short-hop EM transmission pathway. The Short-hop Receiver Sub receives and preprocesses the data and passes the data to the EM MWD tool via a wired communication tool bus. The EM MWD tool transmits the data to the surface via a long-hop EM uplink to the Drilling Advisory System (DAS), which combines the downhole data with surface data gathered from the rig.

The computing architecture is distributed as follows:

1. Short-hop Transmitter Sub. Signal conditions and digitizes data (defined as data sample rate) and scales data (unit conversion; e.g., pressure, temperature, strain, etc.). Transmits (via short-hop wireless) data to Receiver Sub.

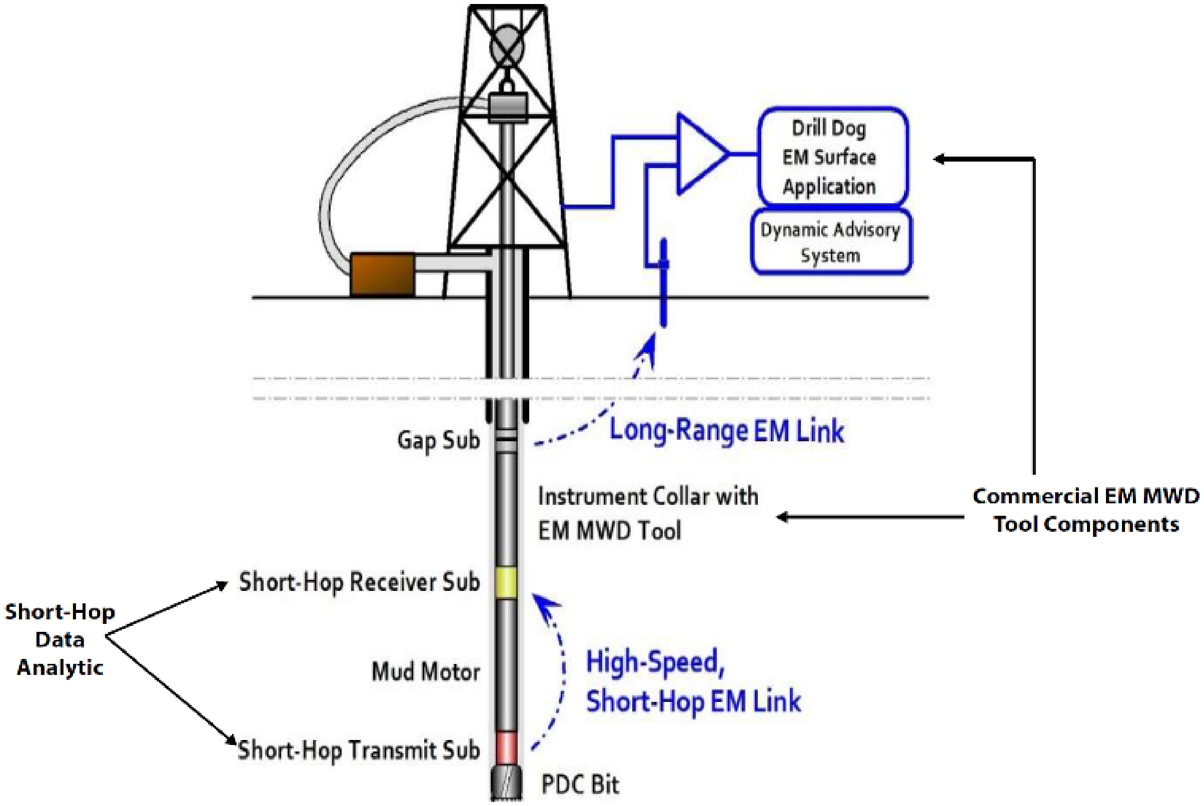


Figure 3.2: Drilling Optimization System Primary Components. The downhole processing component is modular, consisting of a Short-hop Transmitter Sub, located near the bit and below the mud motor, and a Short-hop Receiver Sub, located above the mud-motor and attached to the Electromagnetic (EM) Measurement-While-Drilling (MWD) Instrument Collar. The Short-hop Transmitter Sub gathers real-time bit dynamics data and transmits this data to the Short-hop Receiver Sub via a high-speed short-hop EM transmission pathway. The Short-hop Receiver Sub receives and preprocesses the data and passes the data to the EM MWD tool via a wired communication tool bus. The EM MWD tool transmits the data to the surface via a long-hop EM uplink to the Drilling Advisory System (DAS), which combines the downhole data with surface data gathered from the rig. Image provided by E-Spectrum [152, 153]

.

2. Short-hop Receiver Sub. Receives (via short-hop wireless) data from Transmit Sub. Preprocesses and compresses data as required (e.g. ’binning’ shock or vibration data using severity flags). Transmits (via hardwire tool bus) data to EM MWD Tool.

3. EM MWD Tool. Receives (via hardwire tool bus) data from Receiver Sub. Transmits (via

long-hop wireless) data to EM Surface Application.

4. Dynamic Advisory System. Acquires data from EM Surface Application. Combines downhole data with uphole data gathered from rig (WITS) and performs data analytic advisory function.

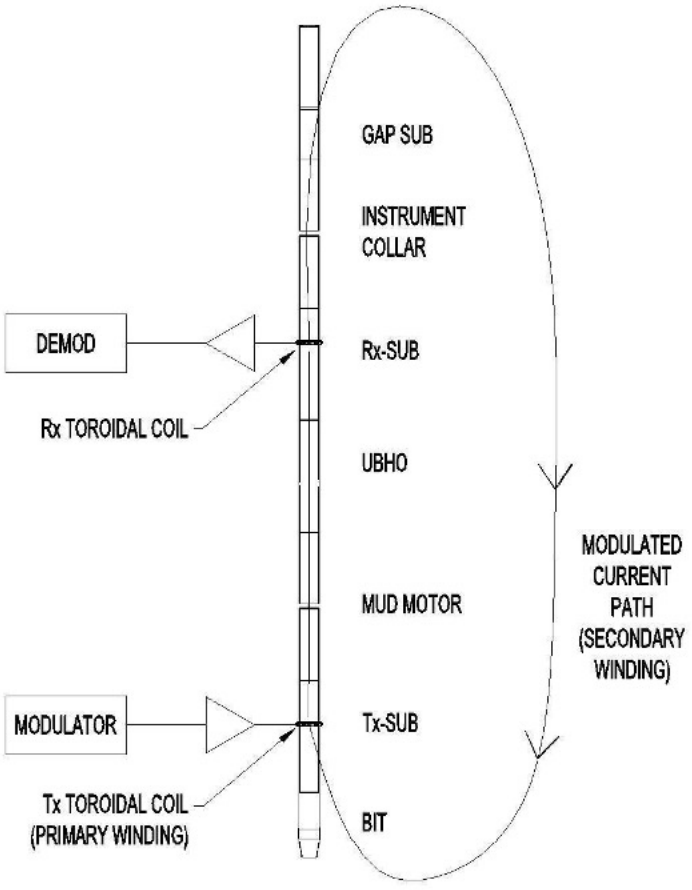


Figure 3.3: Proposed short hop telemetry current path and system to be used in tandem with the developed dysfunction identification library. Adapted from [152].

Data Source and Preprocessing

Depth, acceleration, MSE in the input layer and ROP value as response. The model is trained on 70% of the well data from the field with the rest for validation. Since drilling data is time-series data that is continuously varying, an alternate Recurring Neural Network (RNN) is proposed.

The following three objectives were identified to optimize drilling operations by detecting and mitigating dysfunctions:

- Firstly develop regression models to estimate Rate of Penetration (ROP) from the available data without dysfunctions. And then detect for anomalies in the ROP value while testing. These anomalies are result of dysfunctions downhole while drilling Apply classification models to classify and predict presence of dysfunction as a boolean.

- Using unsupervised, clustering techniques to find patterns in unlabeled raw drilling data with able variables and group the results into dysfunction binary.

Several well-known regression Machine Learning techniques can be used to estimate ROP. We use different type of Machine Learning models to estimate the value of ROP. First, the database was splitted our data into train, validation and test sets. The validation set is set aside to help fine tune our hyperparameters for different ML models. The goal is to further improve the accuracy of ROP prediction. The results can then be compared to determine the best models for estimating the value of ROP.

Validation and Testing of Models

Upon validation by numerical modeling, the proposed telemetry system was successfully tested on a prototype in a laboratory setting. The proof of concept is carried out in two stages.

1. Bench-top Testing: Validated the earth/drill string mathematical model by using the model to design a pair of scaled transmit/receive coils, and performed testing of these coils, in situ, using a Transmitter Assembly prototype board featuring a high-temperature micro processing unit.

2. In-Situ Testing: Once the bench-top prototype was completed and functionally tested, a simplified 1:48 scale model of the proposed BHA and an in-situ test container using a saltwater medium to simulate variable earth resistivities were constructed. Tests were conducted using the proposed BHA scale model to confirm that the model predictions matched the empirical test results for various coil spacings and simulated formation load resistivities.

3.1.2 Machine Learning Based Dysfunction Library

The system control technique previously described in Chapter 2 for optimizing the drilling performance of an autonomous laboratory drill rig was examined for its applicability to a full-scale drill rig. This control technique employs weight-on-bit, torque, and rotary speed measurements in a feedback loop that allows an optimal weight-on-bit and rotary speed to be used to maximize bit performance and survival under rapidly changing and diverse drilling conditions.

In full-scale drilling, however, the slow data transmission rates typical of most types of long-range, through-the-earth telemetry result in a time lag, or latency, between the high-speed measurement of any downhole data and its presentation at the surface. Consequently, we assisted on the determination of the effects of this latency on a driller's ability to make use of downhole information such as weight-on-bit, torque, and rotary speed (in addition to surface measurements of the same parameters) to more effectively control the drilling process.

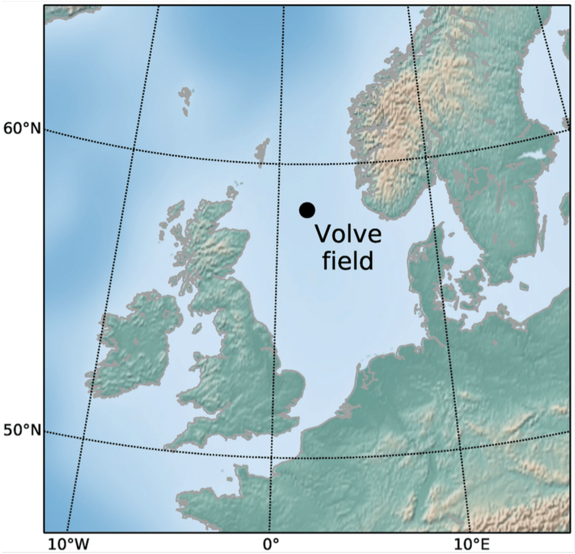
The objective is to research and develop a library of trained machine learning models using previous laboratory and field measurements reported in the literature to recognize one or more of the following bit dysfunctions: full stick-slip, bit whirl, and interfacial severity. The utility of identifying and transmitting the type of bit dysfunction for use in a feedback control system for full-scale drilling will also be assessed, once again taking into account the latency in identification of the recognized pattern type as it is transmitted to the surface.

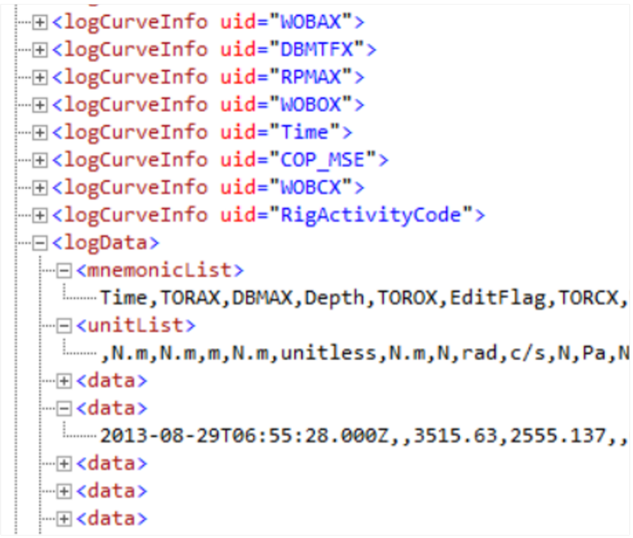
Overview

As part of this thesis, we developed a bit-dysfunction "library" was developed using drilling logs from open sources as well as numerical drill string simulations.

Initially we have determined the following measurements as the minimum input parameters to provide a robust drilling pattern identification algorithm: Downhole Weight on Bit (WOB), Downhole Torque, Weight on Bit (WOB), Downhole Vibrations, RPM, Torque, Depth, Drilling rig status, Equivalent Circulating Density (ECD), Annulus pressure. As well as other derived measurements such as MSE, ROP, or Torsional Severity Estimate (TSE).

The data is an aggregate from multiple service companies, which makes it inconsistent and needs preprocessing as shown in Figure 3.4.





```
<logCurveInfo uid="WOBAX">
<logCurveInfo uid="DBMTFX">
<logCurveInfo uid="RPMAX">
<logCurveInfo uid="WOBOX">
<logCurveInfo uid="Time">
<logCurveInfo uid="COP_MSE">
<logCurveInfo uid="WOBCX">
<logCurveInfo uid="RigActivityCode">
<logData>
  <mnemonicList>
    Time,TORAX,DBMAX,Depth,TOROX,EditFlag,TORCX,
  <unitList>
    ,N.m,N.m,m,N.m,unitless,N.m,N,rad,c/s,N,Pa,N
  <data>
  <data>
    2013-08-29T06:55:28.000Z,,3515.63,2555.137,,
  <data>
  <data>
  <data>
```

Figure 3.4: Location of the Volve field and an extract of the WITSML-based data showing an overview of the XML-formatted variables. The WITSML data was parsed to generate a single file of measurements constituting WOB, RPM, Torque, MSE, Bending Moment, Depth, and responses - ROP, Whirl, Stick-slip, Axial, Lateral Vibrations.

The WITSML data is parsed to generate a single stream of data constituting all the input variables concerned - WOB, RPM, Torque, MSE, Bending Moment, Depth, and responses - ROP, Whirl, Stick-slip, Axial, Lateral Vibrations. The data used for modeling consisted of 7 variables and four outputs, each with two labels in binary. Not all variables, not outputs, are from a common data source. Therefore, combining all of them into one dataset required a significant amount of data cleaning. We chose the measured depth of drilling to be our criteria, and data points are matched

#	Name	Unit	Description	Sampling frequency
1	DEPT	M	Bit Depth	1 ft
2	ROP5	M/HR	Avg. ROP per 5ft	2 Hz
3	GRMA ECO RT	GAPI	Gamma Ray	1 Hz
4	SWOB	KKGF	Surface Weight on Bit	2 Hz
5	Stick RT	RPM	MWD PKtoPK RPM	2 Hz
6	TRPM RT	RPM	MWD TUR RPM	2 Hz
7	SPPA	BAR	SRF PUMPRS	2 Hz
8	DHAT	° C	Downhole Annulus Temperature	2 Hz
9	DHAP	BAR	Downhole Annulus Pressure	2 Hz
10	ECD	G/C3	Equivalent Circulating Density	2 Hz
11	TQA	KNM	SRF STOR	2 Hz
12	TFLO	LPM	HSPM Total Pump Flow	2 Hz
13	SHKRSK RT		MWD SHKRSK	2 Hz
14	RPM	RPM	SRF SRPM	2 Hz

Table 3.1: Most important variables used in the preliminary training phase plot of the variables used as the training set.

based using depth as the main index. After performing this step, a single dataset containing labeled data shown in Table 3.1 was generated.

3.1.3 Exploratory Analysis

An exploratory analysis was performed on a sample subset of data for each well and in aggregate to observe characteristical patterns of each dysfunction. For instance, Figure 3.6 shows the result of a well response for (a) Stick-slip and (b) whirl severity with WOB and RPM as input variables.

3.1.4 Machine Learning Models Methodology

Three statistical methods were used to detect drilling dysfunction in order to optimize drilling operations by detecting and mitigating dysfunctions.

1. First, linear regression models were used to estimate Rate of Penetration (ROP) from the available data without dysfunctions, as well as to detect anomalies in the ROP values while testing. It is assumed that the majority of these anomalies are directly correlated to having

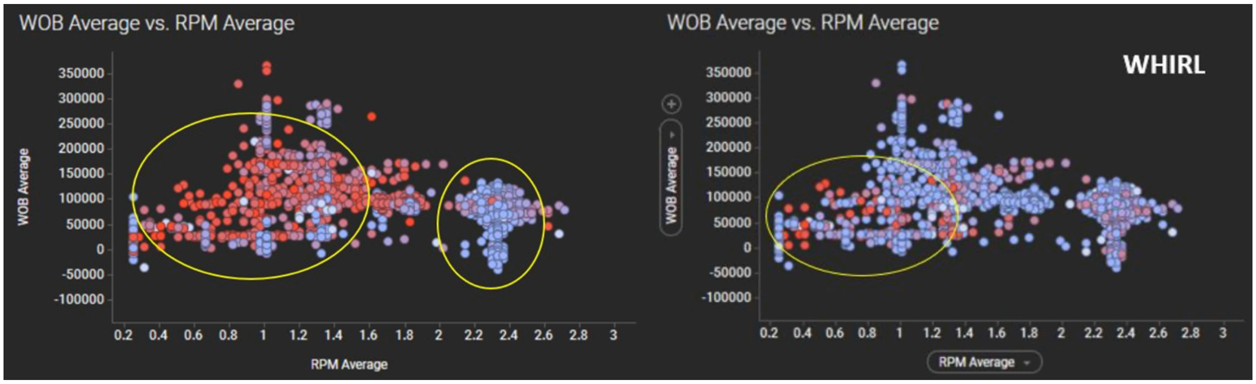


Figure 3.5: Data clustering done as part of the exploratory analysis shows dystinct patterns in the extracted real-time drilling data.

downhole dysfunctions while drilling.

2. Classification models to classify and predict the presence of dysfunction using mutually exclusive binary classification, the details of this classification is discussed later in this section.

3. Lastly, patterns in the unlabeled raw drilling data were identified using unsupervised clustering techniques, and the results were grouped into dysfunction boolean flag – true or false–. We used distinct types of Machine Learning models to estimate the value of ROP. Initially, the dataset was separated into training, validation, and testing sets. The validation set was set aside to help fine-tune our hyperparameters for different ML models. The goal was to improve the accuracy of the average ROP prediction further. These results were later compared to determine the best models for estimating the value of ROP.

Only one out of the three approaches discussed above revealed promising results. Moreover, estimating both ROP and performing anomaly detection using the same dataset proved to be non-feasible with the available data. Data clustering methods yielded unsatisfactory results due to the lack of a large enough and consistent dataset, as several companies and tools were involved in collecting and logging the whole set. Results along with improvements recommendations for future work are detailed below.

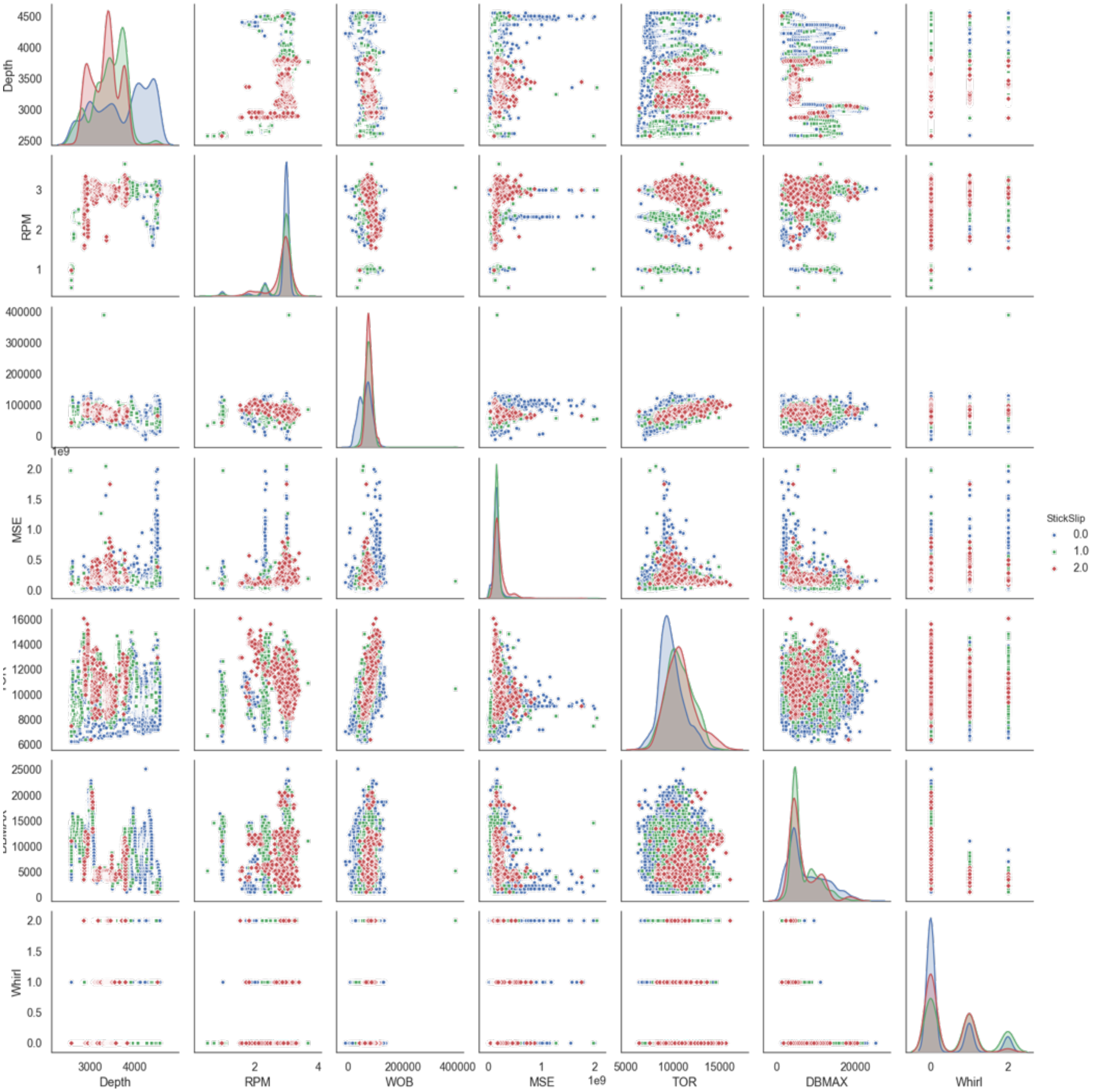


Figure 3.6: Pair plot of the variables used in the aggregated training set. The result of a well response for (a) Stick-slip and (b) whirl severity – 0.0, 1.0, and 2.0) with WOB and RPM as input variables.

3.1.5 Accuracy of Machine Learning Models

The accuracy of machine learning classification and regression models can be used as a metric to compare their performance, as applied in this section, the accuracy to predict the response variables. The accuracy of the model is measured by the confusion matrix. The confusion matrix is a table that shows the number of correct and incorrect predictions. The diagonal of the matrix is the number of correct predictions. The off-diagonal elements are the number of incorrect predictions. The accuracy of the model is the ratio of the diagonal to the sum of the off-diagonal elements.

The formal definition of accuracy is given by the following formula [154]:

$$Accuracy = \frac{\text{Correct predictions}}{\text{Total predictions}} \tag{3.1}$$

Additionally for binary classification accuracy is calculated in terms of negatives and positives as:

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{3.2}$$

Where T_P is the true positives, T_N are true negatives, and F_P are the dalse positives, lastly, F_N are false negatives.

3.1.6 Discussion and Results

Results from each of the classifiers were compared in terms of accuracy and complexity defined in Section 3.1.5. The highest accuracy achieved in successfully identifying dysfunctions downhole when tested on real-world data was $\approx 89\%$.

Multiple statistical classification models were trained on the ’cleaned’ dataset with different levels of accuracy, classifying the correct value when presented with a new set of data. The most successful classifiers by accuracy as listed below, based on test accuracy and complexity of the models. Classifiers, namely Logistic Regression, K-Nearest Neighbors, Decision Trees, Random Forests, and Neural Networks, were used to classify the presence of dysfunction on Stick-slip. First, data was divided into the train set and test set. We only train our experimental models on a subset of our dataset (one well) for training. Since our features have different scales, we carry out data normalization before feeding it into any machine learning models. The most promising models from our experiments are then trained on the complete training set (consists of multiple wells) and finally validate using our test set to make sure the model performs well on unseen data. After the cleaning state, our final data set consists of 36,415 data points. This is not ideal for training complex models such as a neural network since we risk overfitting the data set and perform poorly on unseen samples. Another aspect of the data is that the distribution for different classes is different, with some classes occurring more often than the other. This further complicates our training and testing phrase since some adjustments are expected to accommodate this attribute of our dataset. Figure 3.7 shows the correlation matrix of the most important variables of the training dataset used to derive the results discussed in this section.

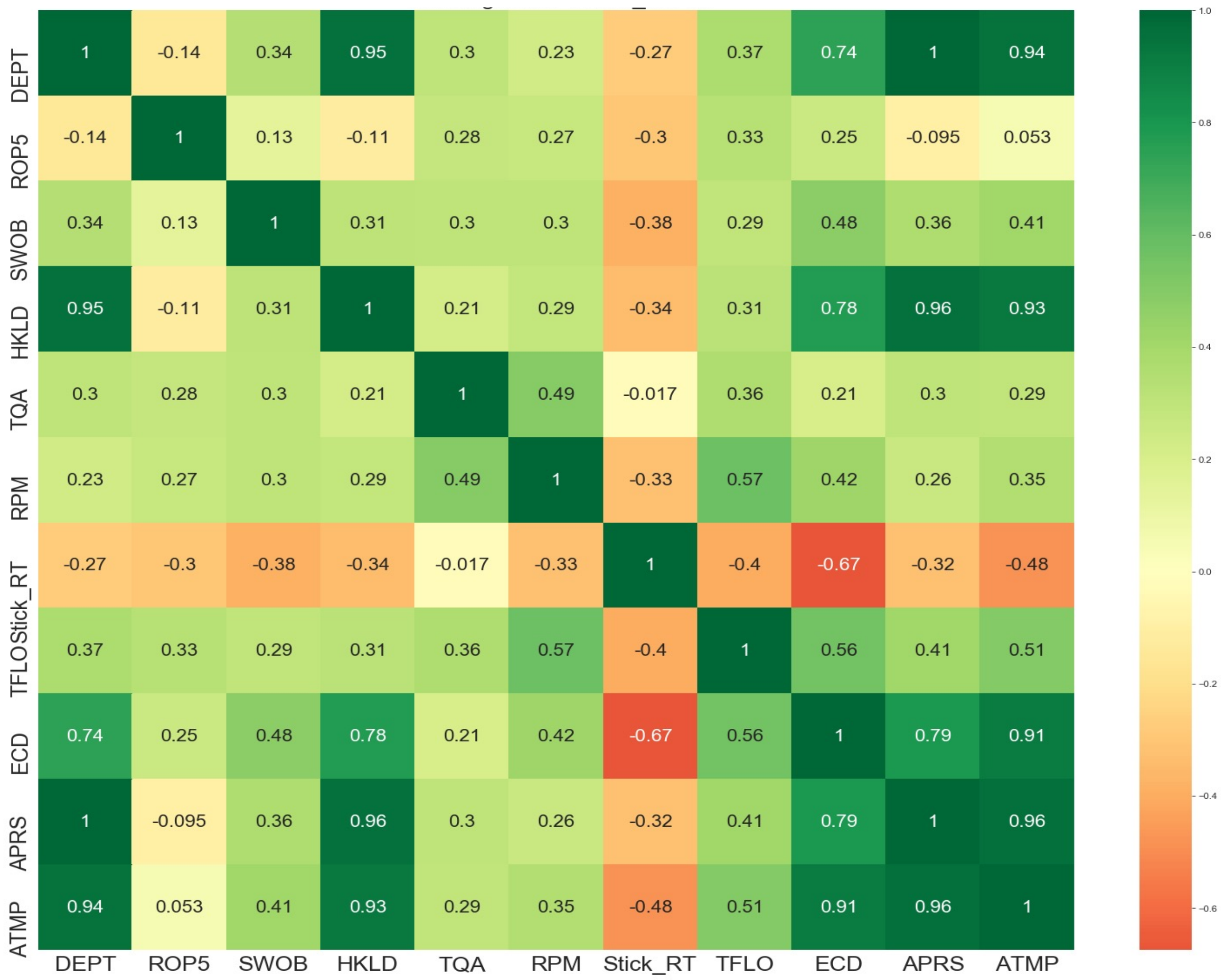


Figure 3.7: Correlation matrix plot showing the distribution and correlation between drilling data variables used in the training phase of the preliminary machine learning models.

Random Forests model achieved the highest overall accuracy of 89% or least test error. Tree-based methods, in general, are less complex to implement and have good interpretability. Decision Trees suffer from high variance, and in order to decrease variance at the expense of slightly increased bias, bootstrap aggregating or bagging of trees is suggested. In the Random Forests model, bagged trees are built with sampled parameters instead of a full set.

Classification results of Random Forests model are as shown in Figure 3.9. As can be seen in Figure 3.7 the data is skewed towards label '0' or 'no stick slip' condition. Therefore, instead of overall accuracy, a better metric such as the f1-score will provide more insights into predictions. Recall of class '0' is 0.95 which means true positives are 95% of all actual labels. Recall of class '1' is 0.70 which means true positives are 70% of all predicted labels. In other words, it can be explained for this case of stick slip detection in this fashion- The model predicted stick slip 5% of the time when it was normal, this is, no stick slip condition. Similarly, the model predicted that there was no stick slip 30% of the time when there was stick slip happening in fact. In other words, it can be explained for this case of stick slip detection in this fashion. The model predicted stick slip 5% of the time when it was normal, no stickslip condition.

Various Neural Network configurations were examined by changing the number of hidden layers e.g., *(2,3)* with different numbers of hidden units for each layer. It was observed that the more complex the network became, this is, a neural network with more inputs, and hidden layers, did not give better performance compared to the simpler models and in some cases perform worse than their simple counterparts. To this end, it was decided that a a fully connected architecture which consists of one input layer (6 input nodes), two hidden layers, each with 128 hidden units, and an output layer with two output nodes, 1 for each possible classes provided the best result for this type of dataset. The architecture and training process is shown in Figure 3.8. Lastly, results of the neural network model model and other classifiers are summarized in Table 3.9.

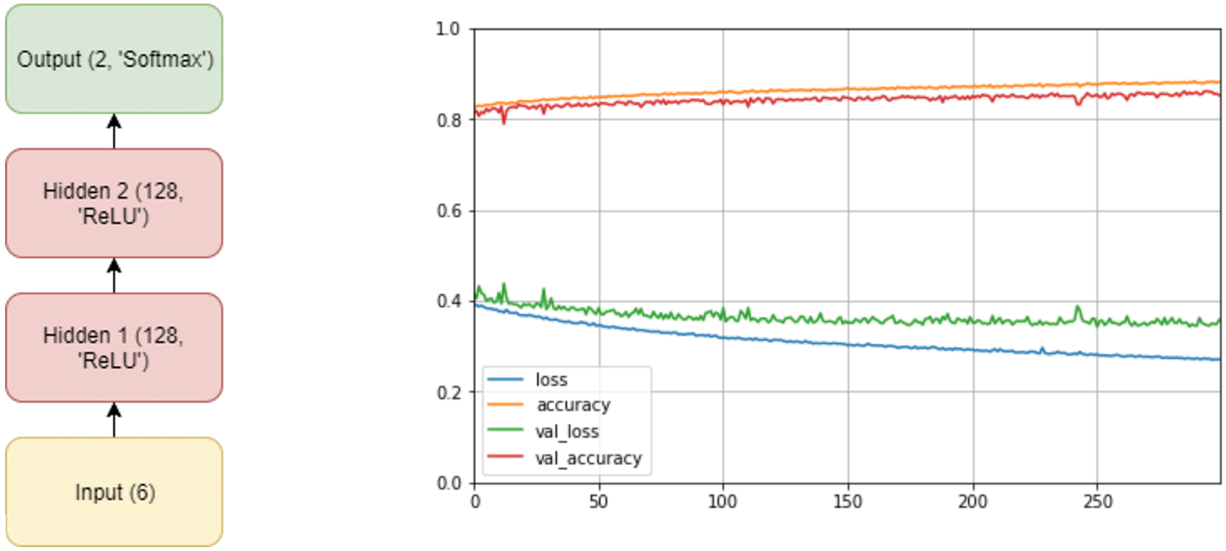


Figure 3.8: A fully connected architecture of one input layer (6 input nodes), two hidden layer, each with 128 hidden units, and an output layer with two output nodes (1 for each possible classes).

Model	Accuracy
Logistic Regression	77%
KNN-10	75%
Decision Trees	85%
Random Forests	89%
Neural Networks	86%

Figure 3.9: Accuracy results from the several supervised machine learning models on the initial drilling dataset identifying stick slip.

3.2 Formation Identification Algorithm Development

The objective of this section is to present a novel rock formation identification model using a data-driven modeling approach. This study explores the use of real-time drilling data to train and validate a classification model to improve the efficiency of the drilling process by reducing Mechanical Specific Energy (MSE). In this study, the feasibility of a layer-based determination and change detection of properties of rock formation currently being drilled in near real-time is demonstrated.

As described in section 4.1, the data for this study was collected from the custom-built lab-scale drilling rig equipped with multiple sensors. The experiment was conducted by drilling through an arrangement of different rock formations of varying rock strength properties. Data was recorded and stored at a frequency of 2 kHz, then filtered, processed, and downsampled to extract relevant features.

This dataset was used to train an Artificial Neural Network and other machine learning classification algorithms. Feature selection was made first with the ten most notable features found by Random Forest, and the second set with derived measurements and down-sampled dynamic features from the sensors. The classification analysis was divided into two steps: the best predictors/features extraction and classification model building. The models were trained using multiple classification algorithms, namely logistic regression, linear discriminant analysis (LDA), Support Vector Machines (SVM), Random Forest (RF), and Artificial Neural Networks (ANN). The overall project approach is visualized in Figure 3.10.

The rest of this chapter is organized as follows. A general description of the testing rig is presented in Section 3.2.1. A description of the most important data elements and the source of the data, preprocessing techniques, and the methodology used in this paper are discussed in Section 3.3. Lastly, the results and findings from multiple classification algorithms: Random Forest, Artificial Neural Networks, Linear Discriminant Analysis (LDA), Linear Regression, and Support Vector Machines (SVM), are briefly described in Subsection 3.3.1.

3.2.1 Data Source

All methods discussed in this section used the same training set and testing set where the whole dataset was divided into 80% training set and 20% testing set. A similar approach could be used in the industry to train algorithms to adjust their drilling parameters to the optimal based material hardness and maximize efficiency by minimizing the Mechanical Specific Energy (MSE) of the drilling process for a given formation or rock strength under overburden stress.

Initially, raw data were sampled at a frequency of 2000Hz, which contained high levels of noise given the high sampling rate of the electrical instrumentation; the data was filtered and downsam-

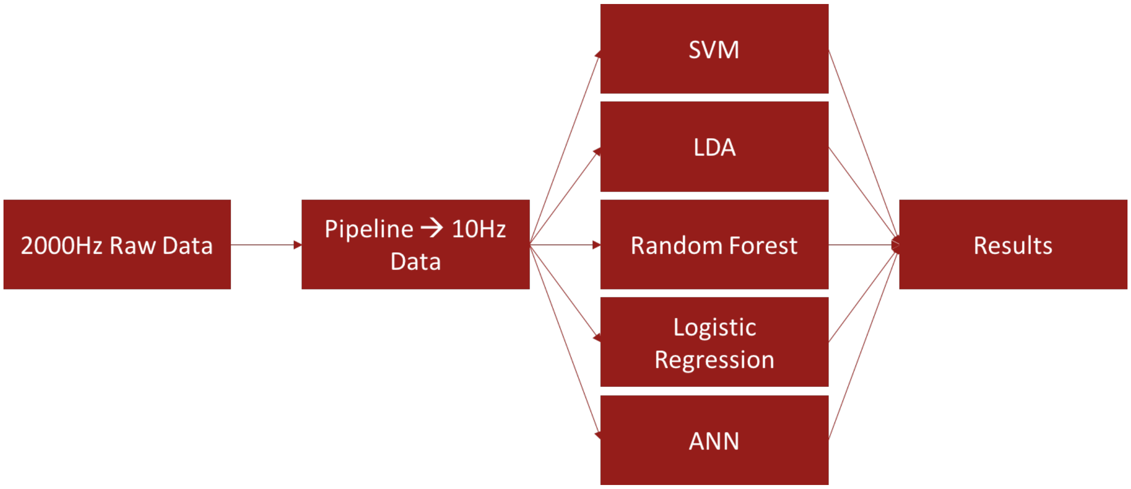


Figure 3.10: Proposed high-level methodology overview of the real-time formation identification algorithm development, described in this section.

pled to an effective rate of 10Hz. Once the information was processed. The following machine learning classification methods used applied: Random Forest, Artificial Neural Networks, LDA, Linear Regression, and SVM. Based on these, the implementation of the algorithms is described in Section 3.2.2. Lastly, the results and findings are described in Section **??**. The overall project approach is visualized in Figure 3.10.

A lab-scale autonomous rig was built with an objective addressing several issues in drilling operations with automation for the SPE Drillbotics 2017 competition, an international student competition designed to accelerate the uptake of automated drilling systems [2]. Drilling data generated from the last set of successfully drilled experiments with the lab-scale rig was used for this project. As shown in Figure [**?**], the rig acquires data from eight different analog and digital sensors. A laser-based sensor measures the traveling block distance and its position related to the top of the structure. This allowed us to measure with accuracy what the drilled depth is. A Wheatstone bridge load-cell measures the tension on the draw-works line that is generated by the total weight of the top drive motor and the overall assembly. The difference between the hook load tension and the original weight is used to get an approximation of the weight on the bit (WOB).

High-frequency data contained raw voltages and recordings from the different sensors, from which 101 other observations were derived; in total, our table contained 125 predictors, with one output. Each dataset contains time and frequency domain observations, including downhole vibration data. An example of the labels of the data studied is seen in Table 3.2.1.

Input Parameter	Response
1. Depth (in)	
2. Program Run time (estimate in millis)	Rock Formation Type
3. WOB (bf)	
4. RPM	
5. Torque (in-lb)	
6. Avg ROP (over dx data points) (ft/hr)	
7. DOC (mm/cutter)	
8. Z vibrations (G's)	
9. X Vibrations (G's)	
10. Y vibrations (G's)	
11. Rig displacement (mm)	
12. Hookload (lb)	
13. UP Vib Z (G's)	
14. Instantaneous MSE (psi)	
15. MSE (psi)	
16. Desired WOB setpoint	
17. Desired RPM setpoint	
18. Z ACC mean	
19. Z ACC RMS	
20. Z ACC Max	
21. Z ACC Min	
22. Z ACC STD	
23. Z ACC Kutosis	
24. Z ACC Skewness	
25. Z ACC Variance	
26. Z ACC MAD	
27. Z ACC IQR	
28. Z ACC Energy	
29. Z ACC Entropy	
30. Z ACC Dominant Frequency	
31. Z ACC Magnitude	
32-45 X ACC	
46-59 Y ACC	
60-73 Z Jerk	
74-87 X Jerk	
88-101 Up _Y Jerk	
102-115 Gyro/Magnetometer data	

Table 3.2: Input predictor parameters used for training.

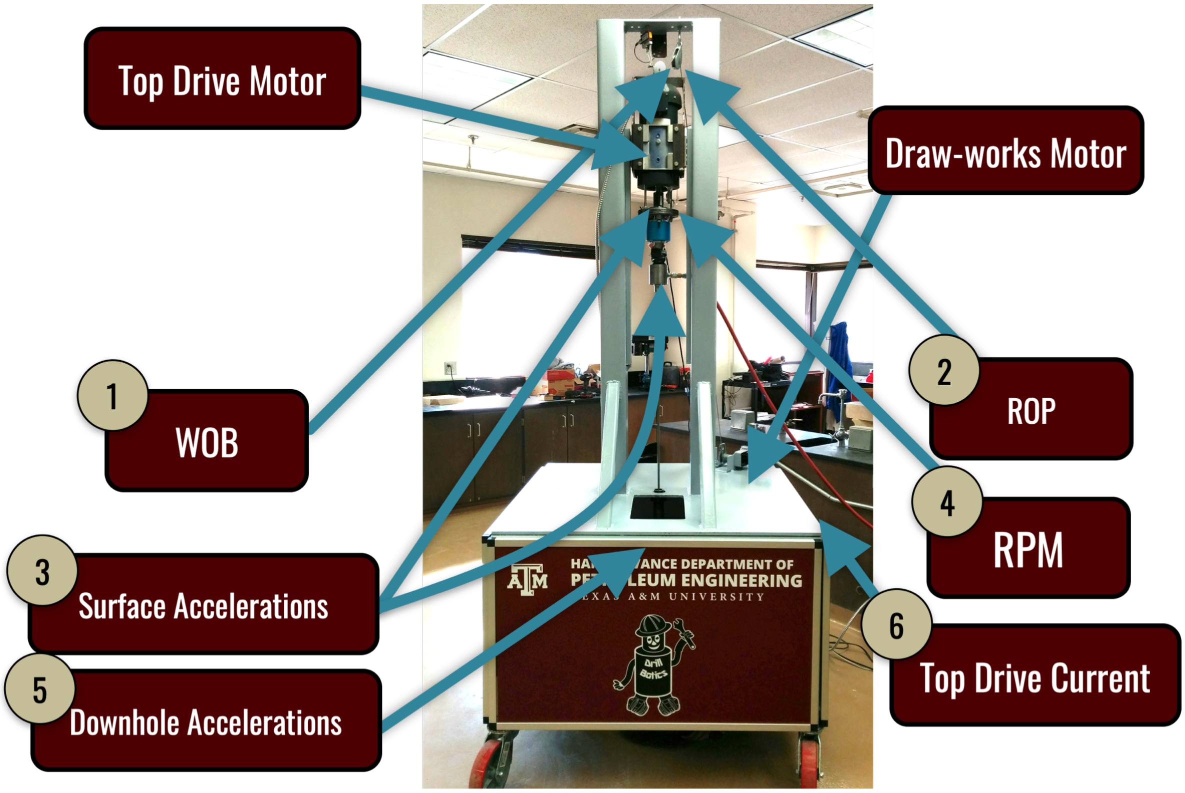


Figure 3.11: Data was acquired using the lab-scale setup pictured above. The lab rig was instrumented with important sensors, both uphole and downhole [2].

The rig acquired depth data using a laser-based sensor to measures the traveling block distance, and its position relates to the top of the structure. This allows us to measure with accuracy what the drilled depth is. A Wheatstone bridge load-cell measures the tension on the draw-works line generated by the total weight of the top drive motor and the overall assembly. The difference between the hook load tension and the original weight to get an approximation of the actual weight on the bit (WOB) was used.

The following steps were taken in order to process our high frequency sampled data to data to train the classification algorithms to reduce the amount of noise in each signal and large file size of the data [155, 156, 51]. Due to limited processing capabilities of the personal computer used to run the models it was not possible to fit the whole dataset into its random access memory (RAM).

1. A Matlab script was written to automatically perform the following steps needed to generate

our training or validation dataset

2. Exclude the initial and final data points of the sensors' raw voltages data

3. Apply a fifth-order median filter to the raw dataset

4. Apply third-order butter-worth low-pass filter vibrations with their most important dominant frequencies responses as a function of revolutions per minute. Therefore, a cutoff frequency of 300 Hz was decided, which for data sampled at 2000 Hz being the cutoff frequency corresponds to 0.2 radians.

5. Convert raw voltages to meaningful physical values

6. Extract the revolutions per minute (RPM) ticks frequency into PRM values

7. Compute linear interpolation of the RPM signal and add it to our columns matrix

8. Extract and compute the approximate weight on bit correction factor function, in order to determine the axial force that is transmitted to the bit.

9. Use the hook-load axial tension to compute an accurate approximation of the real weight on bit

10. Extract all vectors and merge them into a single matrix

11. Compute the correct time vector based on a given sampling frequency

12. Identify the correct measurements when a given rock sample was currently being drilled

13. Add the *formationType* identifier to a vector of the same size; this is, add the response vector for every observation

14. Parse the correct arguments and extract statistical measurements on the high frequency sampled data

15. Perform a fast Fourier transform of high-frequency vibration data

Formation type	1	2	3	5	6	8
Number of observations	13217	805	899	45	28	5214

Table 3.3: The distribution of formations types after removing formation types 4, 7, and 9. The total number of observations was 20,208 downsampled to 10 Hz.

16. Down-sample the data using a linear interpolation function

17. Prepare and sort the column matrix and print it to a csv file. This process led to our final dataset structure

3.2.2 Test-run Scenario

The lab-scale rig shown in Figure 3.11 was used to drill through layers of multiple rock samples with varying formation types, layers, and inclinations in late 2018. However, the most complete and interesting dataset was obtained from drilling into the rock shown in Figure 3.12. The formations encountered were soft cement, sandstone, granular cement, limestone tiles, and asphalt from top to bottom. A total of 9 unique formation types were embedded in the "rock sample" Data from the drilling test was used to train the machine learning model

High-frequency data recorded at 2000Hz was filtered down to 10Hz and used as a training set for all the algorithms described in this section. In this case, the response, y, was coded in all implementations as the formation type, `FormationType`. The remaining data is distributed across the formations, as seen in Table 3.3 It is worth mentioning that data points with FormationType 4, 7, and 9 were removed as there was only seen 8, 3, and 1 data points out of a total of 20,220 data points available

3.3 Machine Learning Models Implementation Description and Results

The following paragraphs describe the implementation of the algorithms using R including which packages, predictors, parameters, and methods are used.

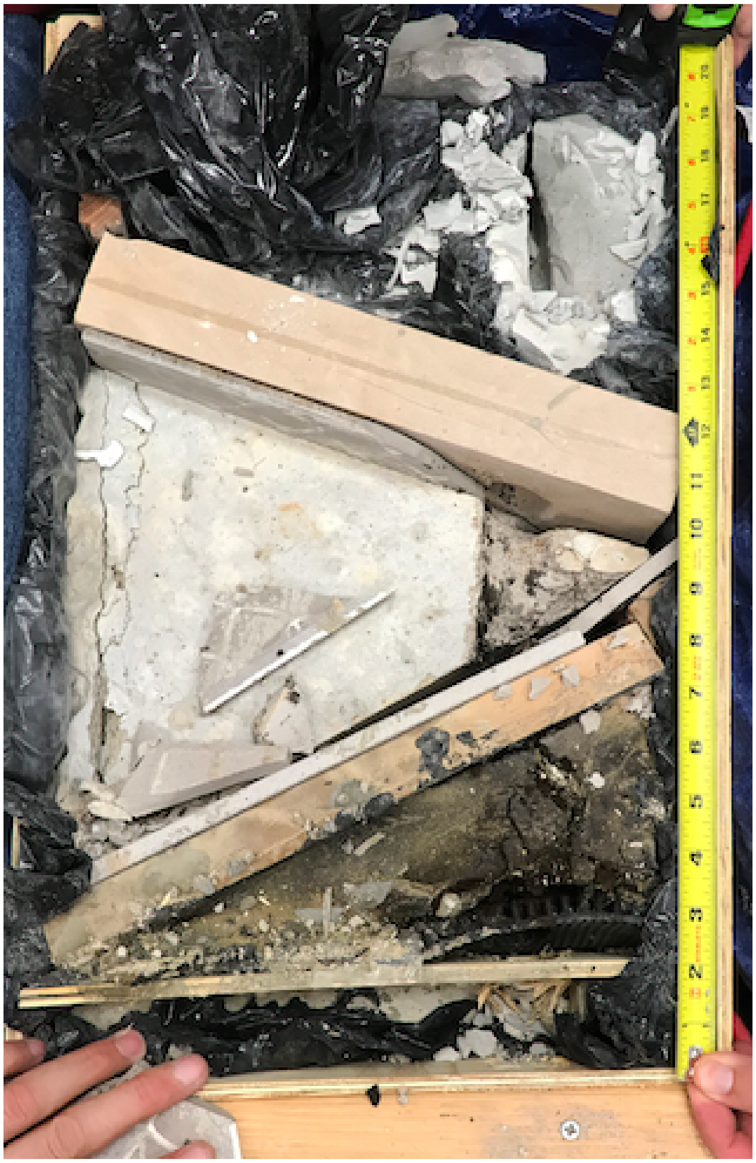


Figure 3.12: The rock sample after having been drilled. From top to bottom the formations encountered are soft cement (7 in) sandstone (2 in), granular cement (2.5 in), limestone tiles (1.5 in), asphalt (4in).

Random Forest

Random Forest (RF) was the first method to be applied in order to understand the importance of each predictor. The *randomForest* package was installed and used in R to administer and implement this type of algorithm. Two measurements of variable importance were used as a criterion to determine the most important predictors. The first was Mean Decrease Accuracy (MDA), also known as permutation importance. This factor shows the decrease of accuracy for each predictor terms on the out-of-bag cross validated errors when a given variable is excluded from the model during training but before prediction. The second one is the mean Gini index, which illustrates a decrease in node impurity as each variable is excluded from the training set, which measures the gain of purity by splits of a given variable to determine its usefulness. Moreover, "the GINI Index is computed by The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature" [157]. As shown in Figure 3.13, The most important three predictors for this model were found to be *LSR_mean*, *Torque_mean*, and *tRPM*.

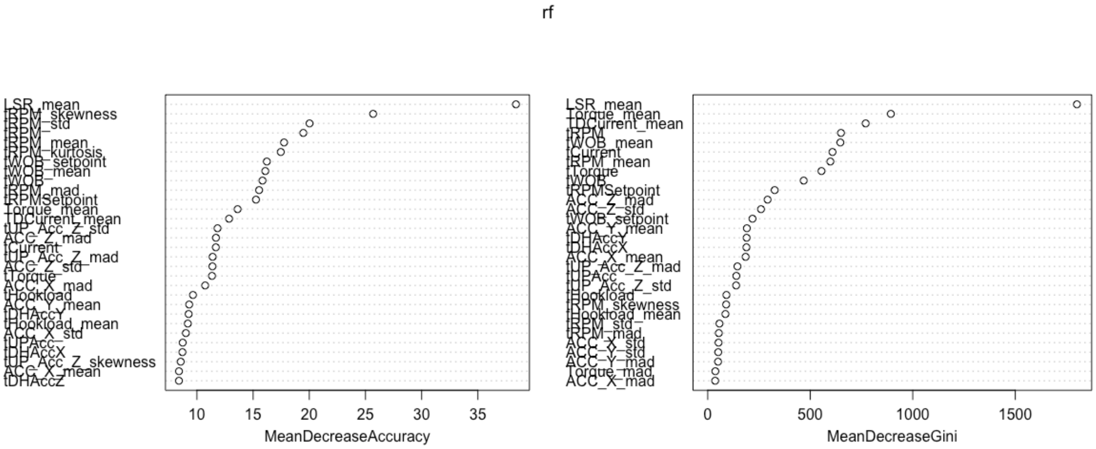


Figure 3.13: Global variable importance of predictors according to a Mean Decrease Accuracy (MDA) and mean GINI metrics. The most important three predictors for this model were found to be *LSR_mean*, *Torque_mean*, and *tRPM*.

The error rate for an increasing number of trees as the hyper-parameter in the random forest method is shown in Figure 3.14, it was clear that error rates become more stable with at least 100 tree divisions. The out-of-bag error rate was found to be 0.17% for 500 tree divisions, which is, to a certain extent, an overfitted model. Its confusion matrix for this model found in Table 3.4.

A “best” subset of predictors based on the GINI index was used instead of the whole input parameters to avoid over fitting models while training. Additionally, cross-validation was

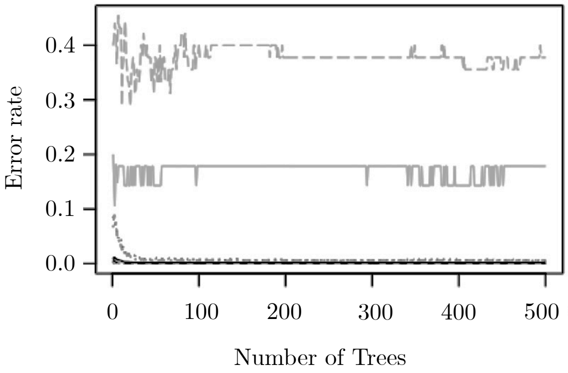


Figure 3.14: Error rate as a function of the number of tree branches for random forest using all predictors.

also used for the training process. In this case the best subset of the ten most important predictors were selected as:*"tRPM_skewness"*, *"tRPM_std","tRPM","tRPM_mean","tRPM_kurtosis, "tWOB_setpoint","tWOB_mean","tWOB," "tRPM_mad","tRPMSetpoint"*. Results are shown in Figures 3.4 and 3.15.

0	1	2	3	5	6	8	Class error
1	13215	2	0	0	0	0	0.00015
2	2	801	2	0	0	0	0.00496
3	0		894	0	0	1	0.00556
5	0	0	13	28	0	4	0.3778
6	0	1	0	3	23	1	0.17857
8	0	0	0	1	0	5213	0.00019

Table 3.4: Confusion matrix results for the top twenty most important predictors by mean GINI index or impurity using random forest.

On this model, tDHAccX, tDHAccY, tDHAccZ, tCurrent, tTorque were used as predictors

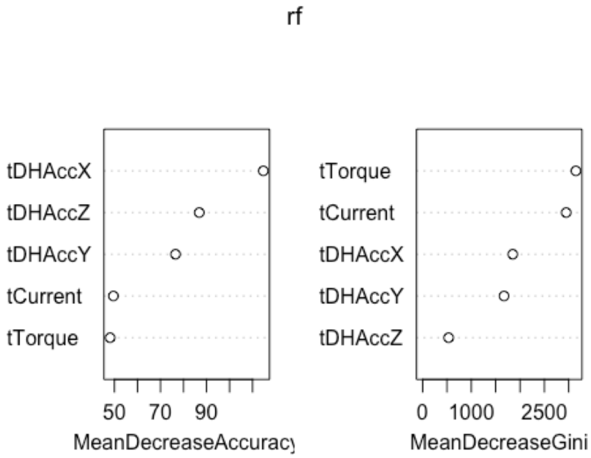


Figure 3.15: Confusion matrix results for the top twenty most important predictors by mean GINI value using random forest.

variables to predict the response, FormationType. The importance of each predictor is shown in Table 3.4. Moreover, error rate change for tree number of Random Forest model is illustrated in Fig 3.15. The estimated error rate was 4.4% for $\approx$ 500 trees.

After analyzing the importance of variables and error behavior, the data was divided into training and testing data. With validation set approach accuracy of model was found as $\approx$ 95%. The confusion matrix for the model can be found in Table 3.4. Cross Validation was also implemented to Random Forest model and accuracy for classification model was 96%.

Artificial Neural Networks

An Artificial Neural Network (ANN) is an interconnected group of nodes which is inspired by the how our own brain works. Each input is connected to all nodes in the next layer, also called a neuron, by a synapse. Each synapse has an assigned weight, w^l_{jk}, which is the weight from the k neuron in layer $l-1$ to i neuron in the layer l. Each neuron's value, z^l_j is computed by the sum of the weighed input plus a bias b^l_j. A schematic illustration for one neuron is seen in Figure **??**.

The packages used for the implementation of ANN were *neuralnet*. Additionally, the package *nnet* was used to encode the categorical response, y, into a "one-hot" vector using *class.ind(as.factor(y))*. Assuming that a neuron has M synapses the value of the neuron is computed using the summing function:

$$z_j^l = b_j^l + \sum_{k \in M} w_{jk}^l a_k^{l-1} \tag{3.3}$$

Once the value z_j^l has been calculated, it goes through an activation function σ, often the sigmoid function $\sigma(z) = \frac{1}{1-\exp^{-t}}$, before sending the activated value $a_j^l = \sigma(z_j^l)$ onto the next layer.

The architecture of an ANN is highly flexible and is depending on the number of layers and neurons in each layer. In Figure 3.16 is seen an ANN with 4 inputs, 1 hidden layer with 3 neurons and one output. In the following notation the output layer will be denoted L and the layer before the output layer will be denoted $L-1$ until the first layer is reached.

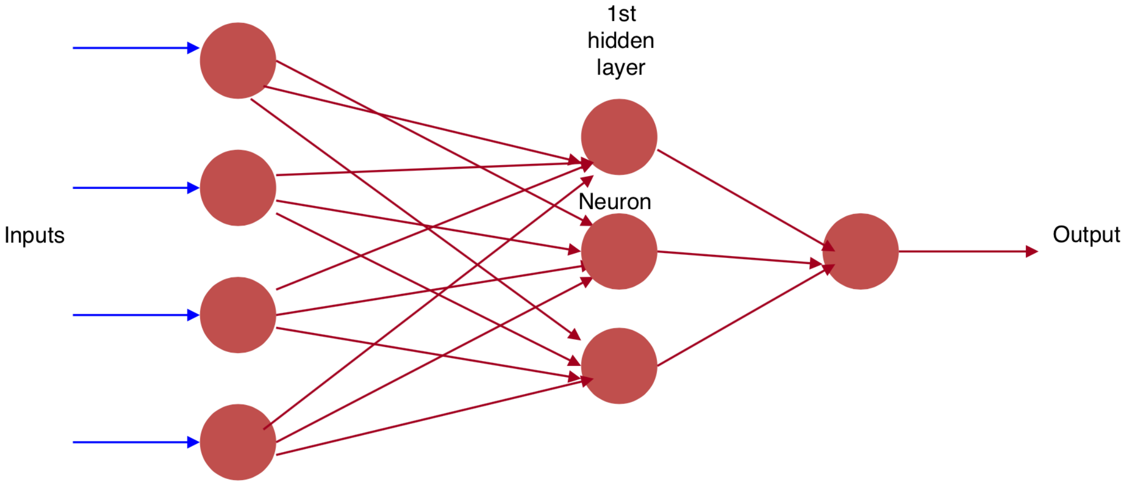


Figure 3.16: An example of an ANN with 4 inputs, 1 hidden layers with 3 neurons and one output.

The training of all the weights within an ANN is usually done using back propagation [158]. The idea is to figure out how much each weight w_{jk}^l and bias b_j^l influence the cost function C. It is there for useful to find $\frac{\partial C}{\partial w_{jk}^l}$ and $\frac{\partial C}{\partial b_j^l}$. In order to do this the error of the output function is defined as $\delta_j^l = \frac{\partial C}{\partial z_j^l}$.

Formation Type (FT)		FT 1	FT 2	FT 3	FT 4	FT 5	FT 6
1		1	0	0	0	0	0
3		0	0	1	0	0	0
5	$\longrightarrow$	0	0	0	0	1	0
2		0	1	0	0	0	0
4		0	0	0	1	0	0
6		0	0	0	0	0	1

Table 3.5: Binary table was used to classify rock formation responses.

The back propagation algorithm have two assumptions for the cost function. The first assumption is that the cost function C can be written as the average of the cost function of all the inputs $C = \frac{1}{n}\sum_x C_x$, where C_x is the cost function for one input. The second assumption is that the cost function can be written as a function of the output, $C = \sum_j C(a_j^L)$.

The back propagation algorithm works in four main steps:

1. **Feed-forward** – Compute all local values $z^l = w^l a^{l-1} + b^l$ and the activations $a^l = \sigma(z^l)$

2. **Output Error** – Compute the error vector from the output layer $\delta^L = \Delta_a C \cdot \sigma'(z^L)$, where C is the cost function, which can be the sum of squared errors, and the index L denotes the output layer

3. **Back propagate the error** – For each layer, $l = L-1, L-2, \ldots, 2$ we compute the back propagated error $\delta^l = ((w^{l+1})^T)\delta^{l+1} \cdot \sigma'(z^l)$

4. **Gradient Decent** – For each layer l we update the weights as $w^l = w^l - \eta\delta^l(a^{l-1})^T$ and $b^l = b^l - \eta\delta^l$, where η is an arbitrary number known as the learning rate

This means that each categorical response gets its own binary vector determining if the response is included in this row or not. An example is seen below:

The data was then centered and standardized. The formula to pass to the neuralnet is all response variable equal to all prediction terms. In the case with 6 response variables and 5 prediction terms it could be. Once the one-hot vector has been created the total training data frame is created which includes all prediction terms and the one-hot response vectors. The data is then centered and standardized using *R* standard *scale* function. The formula to pass to the neuralnet is all response variable equal to all prediction terms (Eq. 3.3). In the case with 6 response variables and 5 prediction terms it could be:

$$FT1+FT2+FT3+FT5+FT6+FT8 = tDHAccX+tDHAccY+tDHAccZ+tCurr+tT \quad (3.4)$$

The setup and training of the neural network was performed using `neuralnet`. The parameter `hidden` determines the number and size of the the hidden layers. It is given a vector as an input where each row determines a new layer, and the value of the row is the number of neurons in the layer. The activation function is set to logistic using `act.fct = 'logistic'`. The output is set to categorical using `linear.output = FALSE`. `err.fuc = 'sse'` and the maximum number of iterations is set to 10,000 using `stepmax = 1e4`. The activation function is set to logistic using `act.fct = 'logistic'`. The learning algorithm is set by default to resilient back propagation and the learning rate factor is set using `learningrate.factor = list(minus = 0.5, plus = 1.2)`.

With these minimum parameters set the ANN can be trained. Once the network is trained the results are computed and are stored in a matrix *pr.nn*. This is done using the function `pr.nn`. This is done using the function `pr.nn = compute(nn, train(X))`, `nn` being the trained network, `train` being our data frame and `X` being the prediction terms. Results from the computation are then extracted to a one-hot vector, which needed to be transposed back to a single vector for comparison with the original formation. This was done using`max.col(pr.nn`. Once this is done the prediction rate was calculated as `mean(mean(results$Prediction == results$Original_Value)`.

A 5-time hold-out validation is created where 20% of the data is randomly sampled as test data and the remaining 80% remains training data. The parameters are set in the same way as stated earlier.

ANN Results

The results based on the dynamic data, *tDHAccX, tDHAccY, tDHAccZ, tCurrent, tTorque*, are seen in Table 3.6, and 3.7. From the table it is seen that the more complex the network the better cross-validated results is obtained.

1 Layer	5	6	7	8	9	10
Cross Validated Value	94.62%	94.43%	94.93%	94.74%	94.74%	95.15%
2 Layer	(10,5)	(10,6)	(10,7)	(10,8)	(10,9)	(10,10)
Cross Validated Value	95.42%	95.31%	95.40%	**95.61%**	95.37%	95.59%

Table 3.6: The results are based on the dynamic data. As the complexity of the network increases, the cross-validated accuracy increases to the mean.

The results based on the 10 most prominent predictors found from Random Forest are seen in Table 3.6. From the table, it is seen that the most complex network performs the best in the cross-validated results.

Linear Discriminant Analysis

The `MASS` package was installed and loaded in order to implement the Linear Discriminant Analysis (LDA) method [159, 131]. The data, then, undergoes a cleaning process where the empty

1 Layer	5	6	7	8	9	10
Cross Validated Value	94.62%	94.43%	94.93%	94.74%	94.744%	95.15%
2 Layer	(10,5)	(10,6)	(10,7)	(10,8)	(10,9)	(10,10)
Cross Validated Value	95.42%	95.31%	95.40%	**95.61%**	95.37%	95.59%

Table 3.7: The results based on the dynamic data. As the complexity of the network increases, the cross-validated accuracy increases to the mean.

rows are eliminated using the omit.na command. This step eliminates potential data compatibility issues that might be encountered along with the data analysis procedure. As previously mentioned, datapoints with an observation of formation type index of 4, 7 & 9 were eliminated as well since these formations are not part of the experimental data but are a result of downsampling the sensors data from 2khz to 10Hz. Subsequently, the data frame structure is portioned into a training set and testing set by randomly sampling 80% of the total data for the former and the rest for the latter. To avoid inconsistent results when splitting the data, we set the seed for the random number generator to 1 using the `set.seed(1).`

The LDA model was then implemented using the ten most prominent predictors previously found using the random forest method. The MASS library implementation of the LDA algorithm takes the response variable, the predictors, and the training data set as its inputs and generates model parameters as its output. The LDA routine is a model that was used later to make predictions while using the training data. The predictive capability is summarized in a confusion matrix that tallies the true positive, true negative, false positive, and false-negative results. To better understand these metrics, the information forms the confusion matrix was transformed to show the percentages of correct and false predictions from the trained model. However, to ensure the consistency and repeatability of the LDA model results, a Kfold cross-validation function was developed. This function called cv. lda takes the data frame, the model, the name of the response variable, the number of K folds, and a seed number as inputs. The function splits the data into a training set, and a testing set applies the LDA model and then calculated the prediction error for each fold. Finally, the cross-validation errors are plotted to aid in determining the necessary value of K for which the predictions are consistent.

Table 3.8 summarizes accuracy results using the live predictors as input parameters while Table 3.9 shows results based on the ten best predictors. The LDA classifier yielded a prediction accuracy of 85.81% using K=15 for cross-validation in the live predictors case and an accuracy of 96.78% while using the ten best predictors with K = 15 folds for cross-validation.

The prediction outputs are given in terms of the probability that each data point belongs to each class type instead of the type labels for each one. Hence, we assigned the type of label for each data point with the probability of the highest one. The average error of this method was 3.43%. This model was also trained using only five predictors, tDHAccX, tDHAccY, tDHAccZ, tCurrent, tTorque, to predict response FormationType with an estimated error of 12.5%.

Formation Type	Layer 1	Layer 2	Layer 3	Layer 5	Layer 6	Layer 8
Accuracy	91%	44%	32%	0%	0%	88%

Table 3.8: Linear discriminant analysis (LDA) model classifier accuracy for the different rock layers using the "live predictor" approach based on a moving average window of the data. Layer 5 was excluded due to having too few samples.

Formation Type	Layer 1	Layer 2	Layer 3	Layer 5	Layer 6	Layer 8
Accuracy	99%	77%	68%	0%	100%	100%

Table 3.9: Linear discriminant analysis (LDA) model classifier accuracy for different rock layers using the top ten most important variables by their GINI index. Layer 5 was excluded due to having too few samples.

Bacause the scaling of the inputs determines the effective weights on the output layer, it can have a large effect on the final solutions. It is considered good practice to standardize all inputs vectors so to have zero mean and standard deviation. This ensures that all inputs are treated equally and allows to select a proper range for the randomly distributed starting weight [160].

Logistic Regression

A multinomial distribution function is used to train the multi-classification model, then the function predict is used to predict. Noticeably, the prediction outputs are the probabilities that each data point belongs to each type instead of the type labels for each one. Hence, for each data point, assign to each type label with the probability of the highest one. The results are shown in

Table 9. From it, we can calculate the error of the method as 3.44%. The model was also trained using only five predictors *tDHAccX, tDHAccY, tDHAccZ, tCurrent, tTorque* to predict response FormationType with an estimated error of 12.5%.

The response in all implementation includes 6 types after removing 3 types 4, 7 and 9 due to their small numbers of data points, which means that the classification problem is multi classification problem. Therefore, instead of using the package in class, we use `nnet` package. Here we use the 10 most important predictors by using randomForest. The function `multinom` is used to train the multi-classification model, then the function `predict` is used to predict. Noticeably, the prediction outputs are the probabilities that each data point belongs to each type instead of the type labels for each one. Hence, for each data point, we assign it the type label whose probability is the highest one.

This method resulted in an estimated error of 3.44% and results are listed in Table 3.10.

	1	2	3	4	5	6
1	2581	7	10	0	0	35
2	5	147	11	0	0	0
3	2	5	147	11	0	0
5	3	14	12	180	2	1
6	5	0	1	0	0	1

Table 3.10: Logistic regression model accuracy using the 10 most important prediction terms by their GINI index.

Support Vector Machines

The package `e1071` was used in this section. We defined a linear kernel resulting in a 54.1% error rate which is almost random, thus a radial kernel was set. First, we tune the parameters cost and gamma using the function tune - we get the best results to train the SVM model. Here the best parameters are *cost=1000* and *gamma=0.5*. However, the prediction outputs are double instead of integers that correspond to 6 types. So, for each data point, we assign it the type that is integer closest to its prediction output [161].

The results are shown in Table 3.11. From it we can calculate the error of the method is 1.06% as in different previous sections the model including five predictors to predict response FormationType gave an estimated error of 7.74%.

	1	2	3	4	5	6
1	2618	1	2	0	0	0
2	12	161	0	0	0	0
3	4	5	197	1	0	3
4	0	0	2	1	2	1
5	0	0	0	1	5	8
6	0	0	0	1	0	1017

Table 3.11: Support Vector Machines (SVM) regression model accuracy using the 10 most important prediction terms by their GINI index.

3.3.1 Test Case Scenario Results and Discussion

All methods discussed in this section used the same training set and testing set where the whole dataset was divided into 80% training set and 20% testing set. A similar approach could be used in the industry to train algorithms to adjust their drilling parameters to the optimal based material hardness and maximize efficiency by minimizing the Mechanical Specific Energy (MSE) of the drilling process for a given formation or rock strength under overburden stress.

Drilling data generated from the last set of successfully drilled experiments with a lab-scale rig was used. The rig acquires data from eight different analog and digital sensors. A laser-based sensor measures the traveling block distance and its position related to the top of the structure. This allowed us to measure with accuracy what the drilled depth is. A Wheatstone bridge load-cell measures the tension on the draw-works line that is generated by the total weight of the top drive motor and the overall assembly. We use the difference between the hook load tension and the original weight to get an approximation of the actual weight on the bit force being applied.

	Algorithm accuracy				
Predictors	**LR**	**LDA**	**SVM**	**RF**	**ANN**
Prominent	96.6%	96.8%	98.9%	99.5%	99.6%
Dynamic	87.5%	85.8%	92.3%	96.3%	95.6%

Table 3.12: Accuracy result summary of Linear Regression (LR), Linear Discriminant Analysis(LDA), Support Vector Machines (SVM), Forest (RF) and Artificial Neural Networks (ANN) classifiers. Random Forest (RF) and Artificial Neural Networks (ANN) performed the best for both the most prominent and dynamic predictors in terms of overall accuracy. Lastly, it can be computationally intensive to compute the derived statistics of the predictors used in this analyis in real-time or some measurements may not even be available in a downhole environment, however, an alternative approach would be to use a subset of the dynamic predictors, which are only filtered and downsampled to 10Hz data, and therefore able to be processed in real-time. Further emphasis should be given to using them as the basis for more complex methods to perform real-time inferences while drilling.

Results shown in the previous sections found that using the most prominent predictors provided better results compared to using the dynamic predictors only Table 3.12. From the results, it was observed that Random Forest and ANN yielded the best performance in terms of accuracy when using both, the most prominent and dynamic time-based predictors.

It was found that random forest and ANN performed the best with prediction accuracy of 99.48% and 99.58%, respectively, for the data set with the ten most prominent features. The high prediction rate accuracy for the most prominent predictors suggests that if the high-frequency data can be processed in real-time, predicting what formation is being drilled in is possible to achieve in near real-time as shown in Figure 3.17.

It can be computationally intensive to compute the derived statistics of the predictors used in this analyis in real-time or some measurements may not even be available in a downhole environment, however, an alternative approach would be to use a subset of the dynamic predictors, which are only filtered and downsampled to 10Hz data, and therefore able to be processed in real-time.

These predictors give less accurate results but are still useful with an accuracy of 95.61% for the ANN model will provide enough information to determine the type of rock being drilled. However, the skewed data set partly explains the high accuracy because, initially, the same formation was drilled for a much longer period of time. However, during critical ranges between distinct formations, the prediction accuracy falls to 69.2%. Nonetheless, the model can still provide enough information to determine the prevailing type of rock being drilled even during transition periods. This result is illustrated in Figure 3.17 where the classification of the samples was stable.

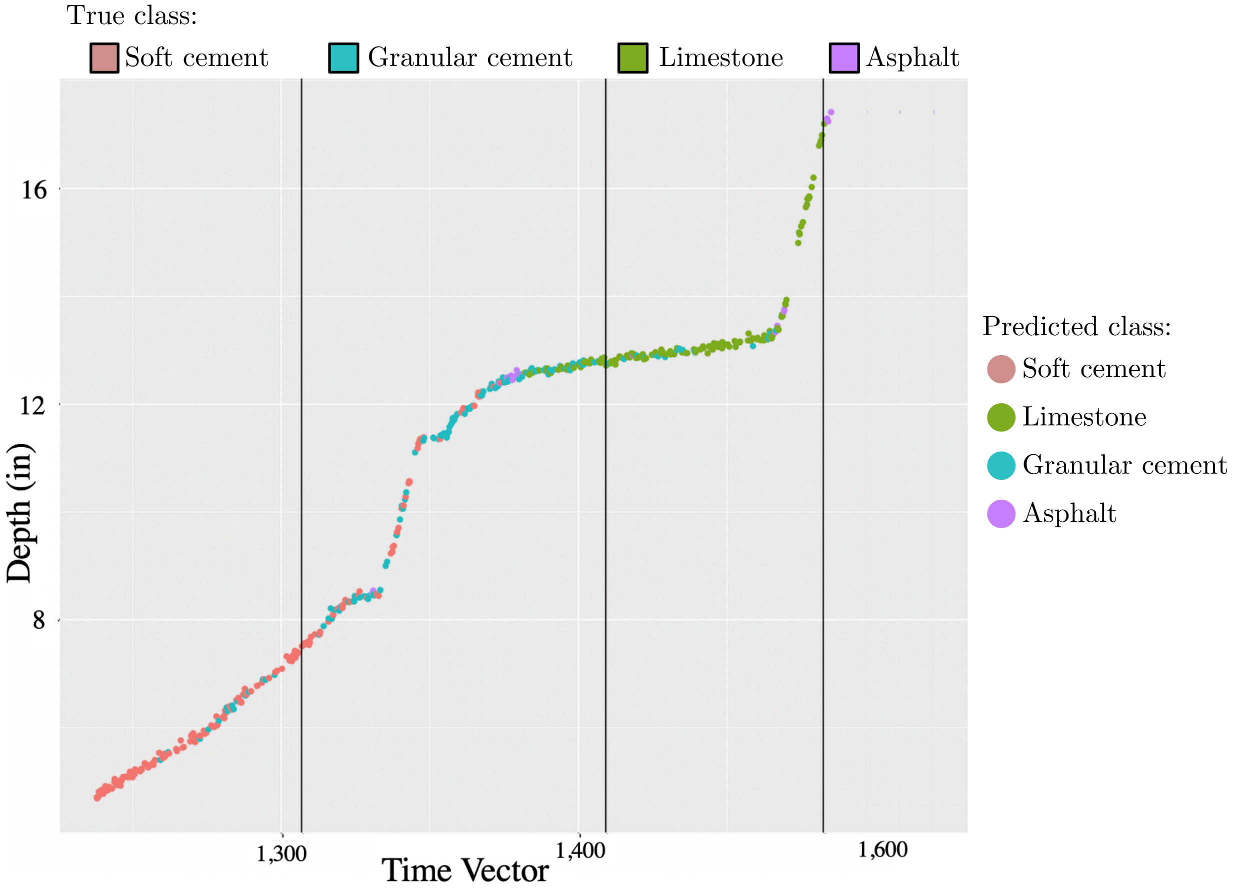


Figure 3.17: The results showed that artificial neural networks yielded the best performance in terms of accuracy when using both the most prominent and dynamic time-based predictors. Shown here are dynamic predictors with an overall accuracy of 95.61%. However, the skewed data set partly explains the high accuracy because, initially, the same formation was drilled for a much longer period of time. However, during critical ranges between distinct formations, the prediction accuracy falls to 69.2%. Nonetheless, the model can still provide enough information to determine the prevailing type of rock being drilled even during transition periods.

Additionally, results shown in the previous sections found that using the most prominent predictors provided better results compared to using a moving window subset of the data as with dynamic predictors.

The results, however, can be improved with further investigations and more experimental data. Potentially, it is possible to train an algorithm that could predict what formation type we are drilling in real-time. This could be done using a simple moving average of the past *n* classifications, which could provide more stability in predicting. Finally, it could be possible to predict what formation type is being drilled in within a few seconds after hitting the formation with further work.

4. APPLIED HYBRID-MODELING AND MACHINE LEARNING FOR DRILLING OPTIMIZATION

Oilfield economic conditions emphasize the need to quickly recognize and respond to drilling dysfunctions to maximize performance and minimize well costs. It is crucial for drilling engineers planning wells to fully comprehend each dysfunction to develop a means to mitigate their impact. As progress was made in this effort, it became increasingly clear that many issues facing the drilling industry cannot be solved or solutions implemented with traditional drilling technology and platforms. The automation of drilling data collection and control systems response is becoming an essential aspect of advancing many aspects of drilling performance for operators. Developing and applying hybrid modeling techniques can help to address some of these issues.

As expanded below, hybrid modeling techniques combine mechanistic, physics-trained machine learning or data-driven models with substantial benefits. This chapter will discuss the application of the hybrid approach to the drilling optimization problem by merging and implementing some of the developed models and frameworks discussed in previous Chapters.

Modeling plays an essential role in the field of science and engineering processes. At its very essence, modeling a process is the translation of knowledge into a mathematical representation using abstracted concepts. Moreover, modeling approaches can be categorized based on the nature of a priori knowledge about a process. For instance, generally, first-principles, mechanistic or phenomenological models represent a class of more transparent or ”white box” models compared to data-driven modeling, which is usually based on a less clear modeling framework based exclusively on process data and its analysis, this is the so-called “black-box” approach [162]. A mathematical classification based on the models’ parameterization similarities can be used to organize engineering models into parametric and non-parametric models broadly.

Parametric models rely on prior understanding of the process; the number of parameters is fixed and might have a physical or empirical interpretation depending on the level of knowledge sophistication. “White-box” models fall in the category of parametric models, while on the other side, non-parametric models are determined exclusively from data. In non-parametric models, the number and nature of the parameters are not fixed in advance by prior knowledge. Hybrid modeling approaches are in between both modeling techniques.

The development of hybrid semi-parametric or “grey-box” models has seen an uptick in adoption and implementation due to the advent, and accesibility of computational power and perhaps more importantly, vast amounts of process data availble [163]. Futhermore, hybrid models can offer key advantages over traditional mechanistic or stand-alone data-driven modeling, a recent review of the terminology used in the literature is shown in [163]. These types of models have been widely recognized, both in academia and the industry, as they can provide significant benefits such as having more transparency when it comes to the description of the modeling approach, the use of a more comprehensive knowledge base, and their cost-effective development compared to traditional parametric models [164, 165, 166].

In summary, the hybrid modeling approach aims to capture mechanistic information along with data-driven surrogate models. The essence is to combine a priori knowledge such as static, and kinetic laws with nonparametric models built using processed data [167].

The following items summarize the main research objectives investigated in the development of the dysfunction identification algorithms and models discussed in this Chapter:

- The development of a dysfunction identification library used in conjunction with the improved short-hop telemetry system to communicate analyzed conditional flags instead of raw data to the surface.

- Preliminary investigation and use of Bayesian modeling to improve dysfunction identification models.

- The use of non-linear dynamic simulator to generate synthetic training databases for physics-based data for machine learning methodologies.

- The contruction and development of a lab-scale drilling rig to implement and test automation algorithms.

4.1 Development of Autonomous Machine to Optimize Drilling Models

This section discusses the benefits of creating a fully autonomous drilling machine and explores how data can enhance drilling operations and performance. To this end, a miniaturized autonomous drilling machine was built to perform operations in terms of rate of penetration and energy efficiency. The miniaturized rig uses commonly available industrial sensors that lead to a large amount of data to be analyzed in real-time. Instrumentation data was collected at a rate of 65 kHz, totaling over 130 million data points for each on average. A PID controller was designed and implemented in a micro-controller to accurately adjust the weight on the bit (WOB) and avoid disturbances. Raw data was collected, filtered, and averaged over to mitigate sensor noise and drift. The information was then passed to MATLAB, where the automated algorithm analysis was performed. The analysis results are used in a closed-loop control algorithm to optimize the rate of penetration, energy efficiency and mitigate drilling failures. The algorithm uses real-time instrumentation data to implement an automated step-test and optimize drilling parameters “on the fly".

Therefore, in order to test our hypothesis we designed and built a lab-scale drilling machine and a downhole sensor integrated into a control scheme. We have applied several sensor types and telemetry techniques in custom-machined bottom-hole assemblies (BHA) that follow a 1.125 in (28.5mm) bit. The collected data is being used for vibration mitigation in real-time without human intervention.

The key limitation regarding the physical construction of the rig was the maximum allowable weight on bit (WOB) due to the selection of a thin-walled tube for the drillstring. The drill pipe used was a 0.375-in (9.5mm) aluminum round tube, with a wall thickness of only 0.035 in (0.9mm). Limiting the rig’s top-drive motor to 2.5 horsepower (HP). A two-blade, 1.125-in (28.5mm) polycrystalline diamond (PDC) micro-bit was provided. It should be noted that certain drilling processes, such as tripping and making connections for additional pipe, are not included as a part of the development of this lab drilling rig.

Measuring and controlling aspects of drilling are as important as the equipment design, so they must model the drilling states to determine appropriate response algorithms. Drilling parameters adjusted before dysfunctions severely limit performancea series of structured tests in various rock types to pre-tune their drilling algorithms. The drilling parameters that maximize performance are immediately calculated and implemented with improved state-detection algorithms and new optimization techniques. A thin-walled pipe intentionally limits the ability to apply weight on the bit (WOB), ultimately generating an environment that promotes bit whirl.

Focusing on MSE rather than ROP directly provides several key advantages. Primarily, the approach is a systemic and not random approach. The algorithm does not test arbitrary parameters in hopes of stumbling onto the most suitable set of parameters, nor is there the risk of not identifying the optimum set of parameters. Secondly, once the ideal parameters have been identified, a variation in MSE provides useful information of downhole conditions, whether that is a new formation, the onset of dysfunction, or mechanical failure. The most common control logic that all modern drilling rigs use today is generally a very simplistic ”cruise-control” often referred to as an "auto-driller“ [168]. Rigs are equipped with many sensors and instrumentation to measure and control several states of the drilling process. The most important controlled parameters are WOB, drill string RPM, and mud flow rate. We propose the implementation of a vendor-independent, more intelligent, semi-automated method of selecting the parameters based on Mechanical Specific Energy (MSE) surveillance and optimization coupled with downhole pattern recognition algorithms that may result in a more efficient drilling process.

MSE is affected by the nature and location of the rock currently being drilled. This project aims to improve the robustness of this proved model by applying data science methods to real drilling data from a controlled experiment.

Bit dysfunctions are prevalent in the lab-scale rig, as the dominant dysfunction in the drilled footage is bit whirl. Whirl is the lateral oscillating sine wave vibration in the BHA. As the string is rotated, it begins to move laterally due to mass imbalances, creating a vibration that travels through the string. When this vibration reaches the bit, it tilts the bit and drives the bit into the borehole wall. This movement creates side cutting which in turn drives the creation of borehole patterns.

In field operations, the impact force of the bit hitting the borehole wall is great enough to damage the outside cutters, which in turn negatively impacts ROP. Whirl is the only dysfunction that gets better with additional WOB, as the additional weight increases the DOC and limits the bits ability to move laterally [59, 39].

With the limited torsional strength of the 0.375 in. (9.5mm) aluminum pipe, the operation was restricted by the amount of torque that can be applied to the string without failure. This limits the Depth of Cut (DOC) and WOB that can be achieved in all sections of the hole, creating an environment that promotes an increase in the magnitude of bit whirl. Throughout the testing process, signs that bit whirl was limiting performance were common, such as borehole patterns seen in Figure 4.1 and impact damage to the outer cutter as shown in Figure 4.2. Thus, it was critical to have a control system that recognizes, responds and minimizes whirl.

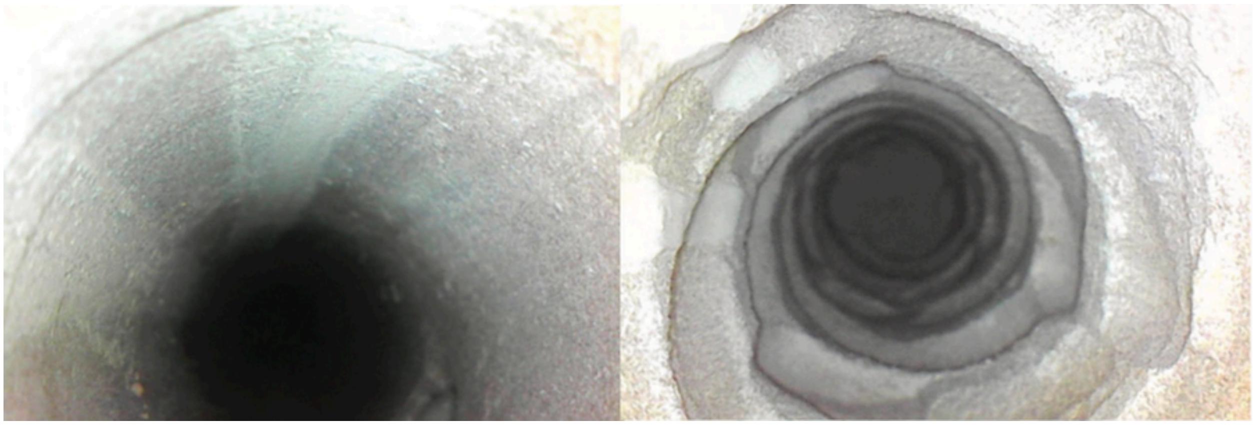


Figure 4.1: Bit whirl-induced spiral borehole patterns lead to, at best, slower drilling performance, and, at worst, damaged drill bits or downhole equipment, after [3].

Stick-slip, in the simplest sense, is an alternation in bit speed, relative to the surface RPM, that is driven by oscillations in torque [87]. Given that the rig is so limited by torque, one would assume that stick-slip is also a dominant factor that limits the drilling performance in practice and in our miniaturized drilling rig. However, in the panti since the drill pipe has a relatively low ratio of the length (36 in. [914.4mm]) to the wall thickness (0.035 in. [0.9mm]), the drill pipe simply

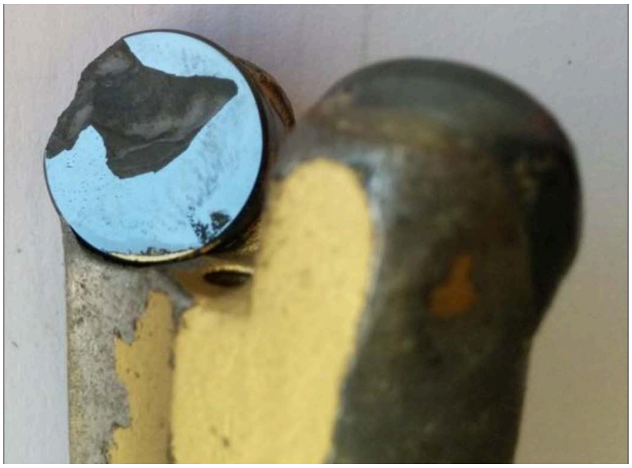


Figure 4.2: Severe bit whirl caused bit damage that would in turn reduce ROP for the duration of the drilling interval, and increase the risk of pipe failure, after [2].

cannot achieve torque oscillations great enough that will significantly affect performance without the pipe undergoing torsional failure. This can be calculated by first calculating the polar moment of inertia, and then using this value to determine the maximum angle of twist for a given torque. For example, with torsional failure occurring at 32 in-lbf (3.6 N-m), the pipe will only exhibit a maximum of 1.21° of twist. Since the magnitude of twist is so small at its peak, the string length will only shorten or elongate a negligible amount, meaning the DOC will not oscillate enough for stick slip to occur before the pipe shears.

4.1.1 Bit Selection

Bits were designed primarily to address the most likely bit dysfunction: whirl. The most crucial change that was made was extending the gauge length of the bit, as the extra length limits the severity of side cutting (lateral displacement) that the bit exhibits before it drills any further [169].

The bit configurations tested were 1) two-blade PDC, 2) two-blade PDC with an extended gauge, and 3) a three-blade PDC with an extended gauge. For the testing and data discussed in this section the three-blade PDC with extended gauge drill bit (3) was used, as the ROP gains found in testing were substantial, as shown in Figure 4.3, due in part to the bit's ability to transition between rock layers without high levels of bit whirl occurring. This was most noticeable when transitioning between lower strength to higher strength rock types.

From a mechanics standpoint, the two-blade PDC with extended gauge should drill 30% faster than the three-blade, as the two-blade PDC will achieve a greater DOC compared to the three-blade PDC when the same WOB is applied. This two-blade requires more contact area per cutter for the load per area to equal rock strength. The most critical difference between the two bits is that they both generate the same torque at the same WOB, as the number of blades does not affect the overall aggressiveness of the bit. Bit aggressiveness, or more accurately called the coefficient of sliding fricction (μ), is the amount of torque generated for a given WOB [170, 171]. Being that the system is limited by both torque and bit whirl, increasing the DOC without increasing the torque would significantly increase performance.

Unfortunately, a manufacturing problem resulted in the two-blade and three-blade PDC cutters having different gauge lengths, with the three-blade being the longer of the two. When compared, the three-blade outperformed the two-blade in every test, strictly due to the extra gauge length constraining lateral movement of the bit as shown in Figure 4.3.

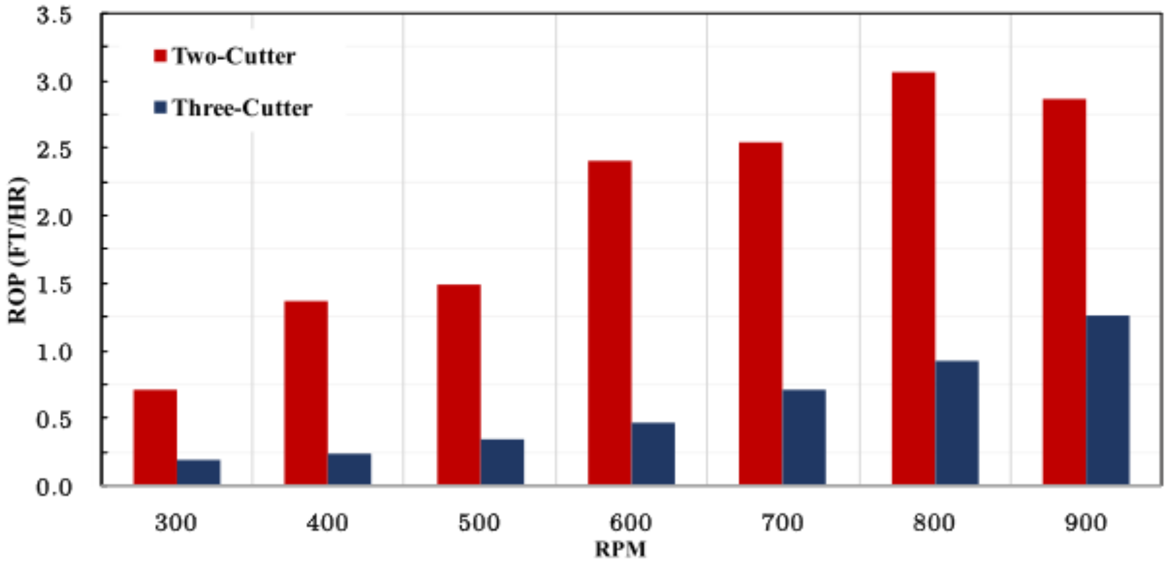


Figure 4.3: Primarily due to the longer gauge, the three-blade PDC bit has significantly higher ROP due to suppressed whirl. Data was retrieved using 40 lbf of WOB in sandstone.

4.1.2 Rig Instrumentation

The most important instrumentation sensors used in the development of this project will be discussed in this section. Some measurements are derived variables of fundamental readings such as torque, WOB, vibration, and temperature are discussed. Instrumentation manufacturers specify the range, sensitivity, precision, resolution, linearity, etc. of a device. Sensors must be calibrated and corrected periodically to assess their performance.

The manufacturer specifications were followed to quantify the precision of each sensor used in the control algorithm. Noise, temperature drift, and other non-linearities were not considered in this table. Calibration procedures were carried out to verify the linearity of transducers and the amount of error for a given reading. Sensor locations can be seen in Figure 4.4.

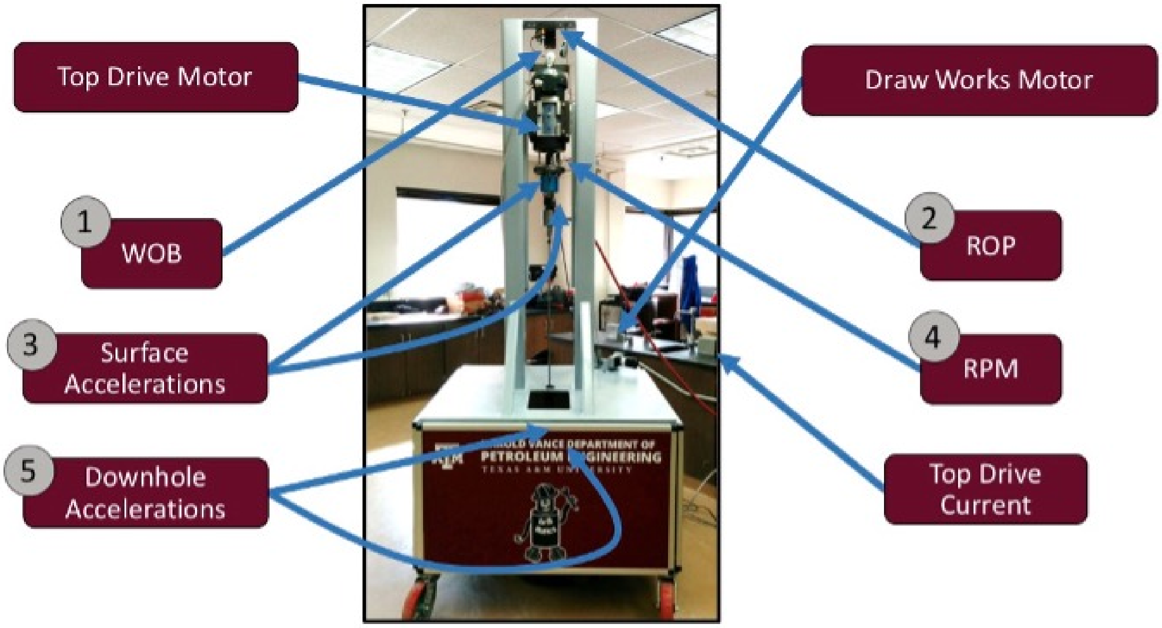


Figure 4.4: The lab setup is equipped with several key sensors both surface and downhole.

Revolutions per minute

To measure RPM, an optical tachometer made with an optoelectronic circuit composed of an infrared emitter (IR) diode coupled with a photo-resistor and a transistor, is used to generate pulses at each revolution. The IR diode shoots light to the motor shaft where a light-reflecting tape is placed; then the photoresistor excites a transistor base to generate a voltage pulse. The signal is discretized based on a voltage cutoff, and it is used to evaluate its frequency. The frequency corresponds to the shaft rotational speed. The motor rotational speed is controlled by adjusting the voltage supplied to the coils. An industrial motor driver is used to control it accurately.

4.1.3 Safety

All the electrical circuits, power supplies, and analog signal filters are located inside a separate junction box to ensure safety and continued performance under a variety of environmental conditions. The junction boxes isolate sensitive electronics from ambiance moisture and unwanted external factors.

A redundant emergency shutdown system is used in the form of a mechanical switch located on top of the junction boxes to shut down the drawworks motor, top drive motor, and all systems in a single action. As with industrial standards, this operation is independent of any software computation; it directly cuts the power supplied to the system. Alternatively, a digital emergency shutoff is located on the output display screen to immediately turn off the drawworks and the top drive motors.

4.1.4 Close-loop Controller

A scheduled-gain PID Controller was designed and implemented in a micro-controller to accurately adjust the amount of weight on the bit (WOB) and avoid disturbances. High-frequency data was acquired using LabVIEW and analyzed in real-time through the MATLAB programming environment. The data was then passed to MATLAB, where the automated algorithm analysis is performed. The results of the analysis are used in a closed-loop control algorithm to optimize the rate of penetration, energy efficiency and mitigate drilling failures. The algorithm uses real-time instrumentation data to implement an automated step-test and optimize drilling parameters on the fly. Increasing the average rate of penetration and reducing vibration-induced borehole irregularities.

A system identification approach was used to derive the transfer function for each configuration of the rig, using the following procedure:

1. A calibration procedure was taken before, and after each run.
2. High frequency sampled raw data was acquired and saved using LabView cDAQ at a high enough sampling frequency.
3. The excitation loop was fed directly into the basic proportional controller and ran continuously until each test was finished.
4. Instrumentation data were re-sampled and analyzed using the MATLAB System Identification Toolbox.
5. A transfer function was extracted from the linear response observed under static conditions.

The PID controller was implemented in a microcontroller running at 84MHz. Using a microcontroller reduces the size, costs, and energy consumption compared to a design based on a microprocessor or a full-sized computer. Modern consumer-level micro-controllers are capable of archiving remarkable high operating speeds and efficiencies [172]. An example of an excitation run and the signal acquired to derive the transfer function is observed in Figure 4.5.

More details of the performance and implementation of the scheduled-gain PID controller are available in [3]. Further refinements to the transfer function of the WOB and mechanism were implemented, for instance, a depth-based correction function was used to better estimate the transferred WOB force to the bit as shown in Figure 4.6.

Mechanical Specific Energy

Mechanical Specific Energy (MSE) was first used to model the drilling process of mines excavation as a crushing process. More significant volumes of rock are fragmented into much smaller

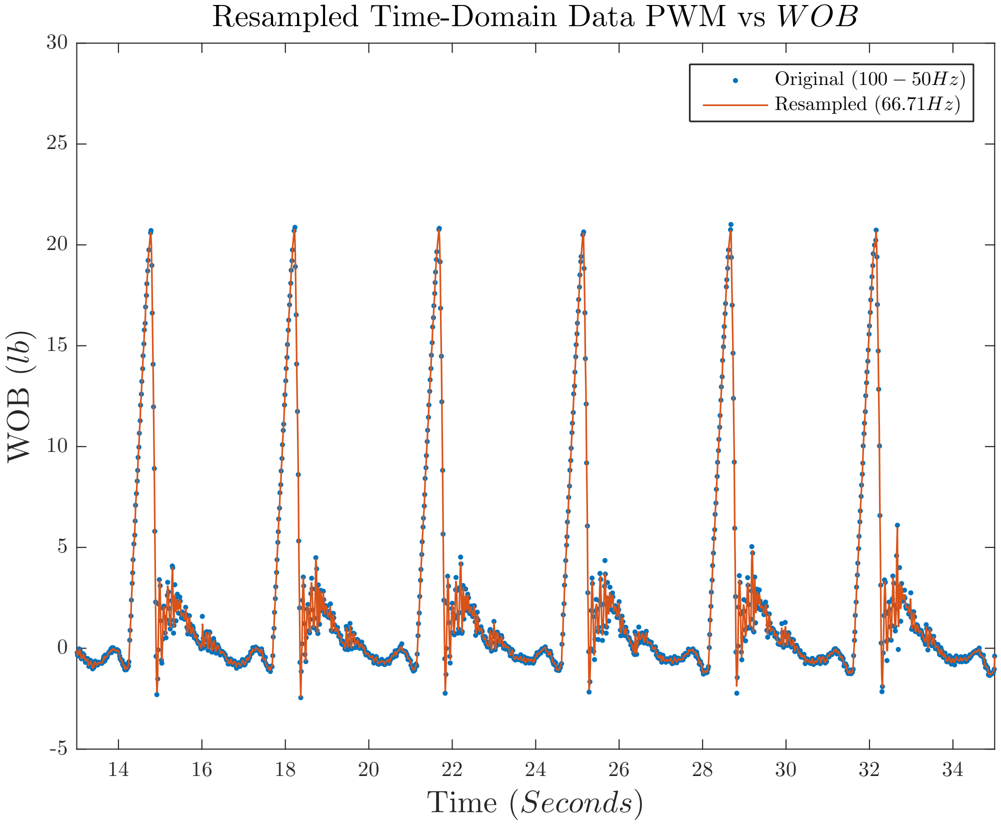


Figure 4.5: Drawworks's WOB control excitation response under steady state conditions [3].

pieces that allow recovering the material economically. Teale [173] proposed the MSE model, which relates the amount of work applied in rotary drilling from its relating trust and rotating components to a given cross-sectional area from which rock is being removed [173].

This concept has been ported to the drilling industry and used to monitor drilling performance, using MSE, and to measure the efficiency of the drilling operation. It is defined as the energy used per volume of rock drilling.

Although it is clear that no single value of MSE can be used as an indication of the efficiency of every drilling operation at exactly a given time due to variations in rock homogeneity, percussion frequency, and other complex drilling dynamics variables, an average trend over a homogeneous section of rock was determined to be enough to calculate, model and predict performance. It was later discovered that drilling specific dysfunctions are also identifiable using mechanical specific energy. As seen in the experiment shown in Figure 4.7, mean MSE values were obtained for

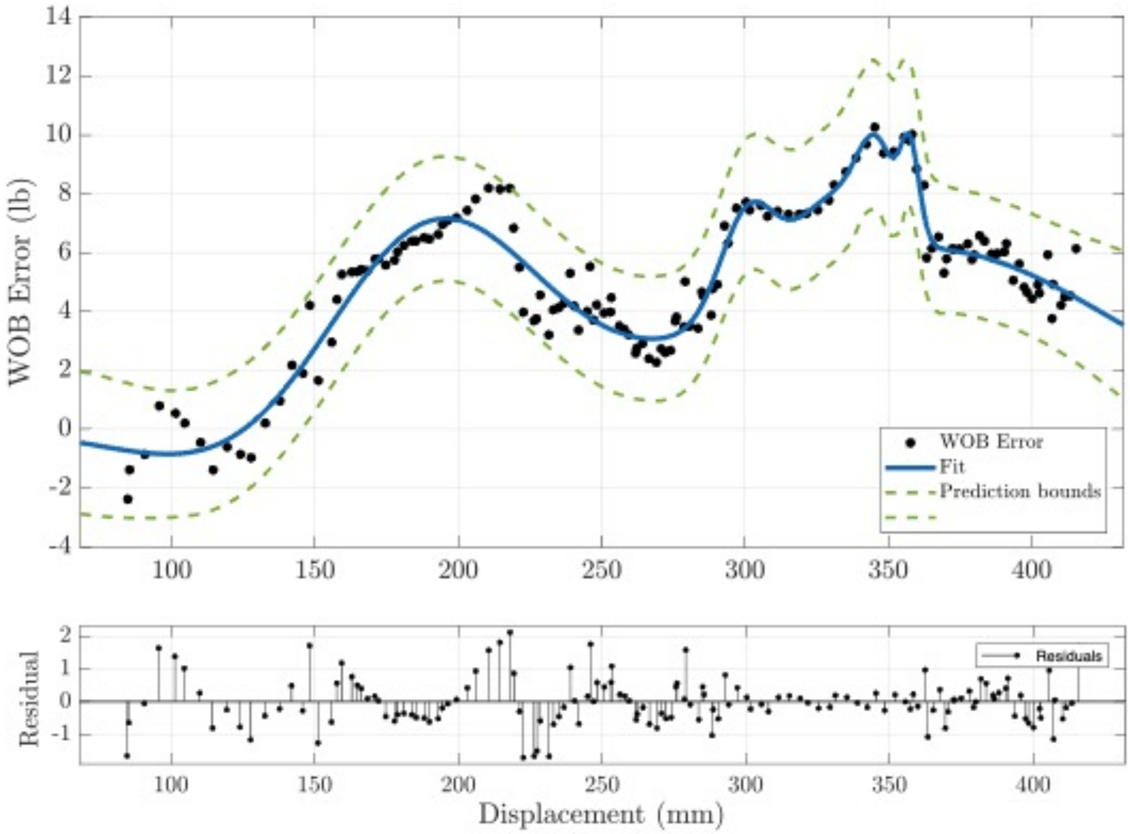


Figure 4.6: Due to the importance of accurate WOB measurements, correlations composed of a piecewise function made of Gaussian splices is used to correct the WOB measurements for depth. WOB correction factor can get as high as 16.0 lb after [2].

different drilling operations configurations for a given bit and formation type.

A linear relationship between weight on bit (WOB) and revolutions per minute (RPM) is observed partly because drilling bits indent until the vertical contact area is such that the force per area is less than the rock strength. If the rock strength is higher the amount of force needed to offset the loss of force per area will increase, therefore the rate of penetration (ROP) will decrease. If ROP increases with RPM because the sliding distance per period for each PDC cutter increases the amount of drilled rock. Lastly, measured torque will increase linearly with depth of cut (DOC), this is the amount of rock that the bit indents into, due to its natural relationship with force per unit area required to fracture the rock exposed to the cutter face.

The MSE model estimates the rotational and axial work as a function of the volume of rock

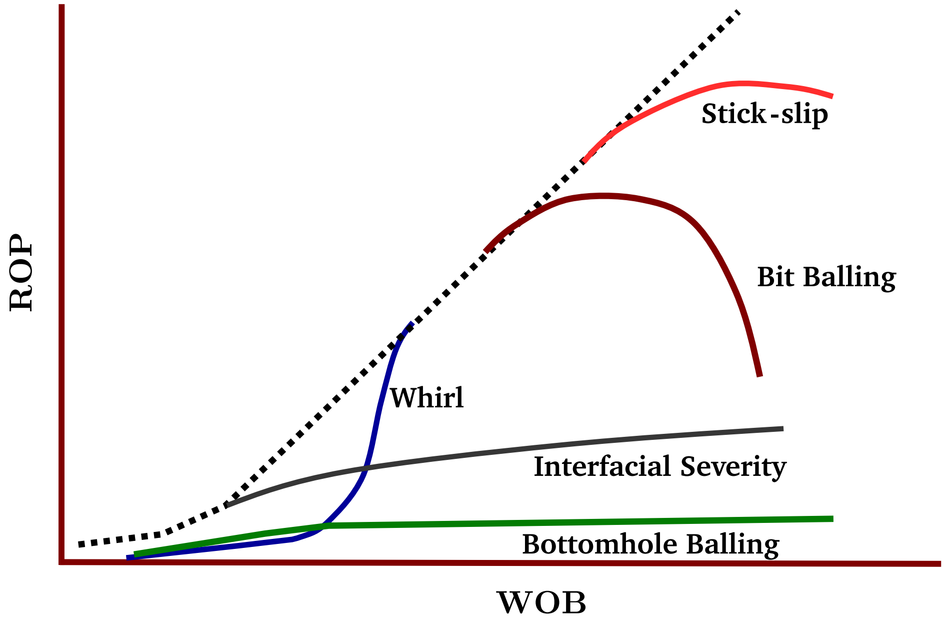


Figure 4.7: Mechanical specific energy and its relationship to common drilling dysfunctions, adapted from [59].

drilled. Other parameters such as rock compressive strength, overburden, bit type are intrinsically correlated. MSE is primarily used as an efficiency index which states that if the bit is 100% efficient, the Mechanical Specific Energy will equal the rock compressive strength. However, that is never the case.

The MSE equation without a correction factor is:

$$MSE = \frac{480 \cdot T \cdot RPM}{OD^2 \cdot ROP} + \frac{4 \cdot WOB}{\pi \cdot OD^2} \tag{4.1}$$

The development of both the concept and idealization of Mechanical Specific Energy (MSE) has become a valuable parameter for drilling engineers to improve drilling performance and efficiency. MSE tends to increase, decrease or remain constant depending on bit dysfunctions and the overall drilling efficiency. Different types of dysfunctions are encountered in the wellbore, and each dysfunction has a particular control response necessary to mitigate as shown in Figure 4.7.

The advantages of using real-time tracking of Mechanical Specific Energy have been reported to archive remarkable improvements in drilling efficiency. Drilling performance is difficult to be equitably measured and compare from well to well, because many performance indexes that are used as correlation factors are based on data from offsets wells and different rig configurations, or wellbore trajectories. Therefore, a new approach was taken by an operator, and a pilot program in 2003 was established to verify the usefulness of on-site MSE tracking first displayed as a rig site surveillance tool in 2003 [58].

The rock formation identification model discussed in Chapter 3 was combined with a common drilling optimization methodology based on reducing MSE over a period of time called, a "step test" and programmed to be excuted autonomosly. Knowing the expected lithology would allow the system to better optimize its automation algorithm to achieve the optimum parameters as quickly as possible. For example, more aggressive parameters can be selected when drilling through an easy to drill formation, such as sandstone, while a more conservative operating envelope can be used for harder to drill lithologies such as granite. The system uses of live drilling data to estimate the current lithology in real time using one of the pre-trained models developed in the previous chapter.

4.1.5 Automated Step Test and Rock Formation Identification Test Run Results

The automation algorithm was based entirely on monitoring MSE rather than ROP. While it is true that a low MSE usually correlates with high ROP, this is not always the case. In real drilling applications, the highest achievable ROP can correlate to bit damage that will ultimately be counterproductive. Even in this small-scale application, there were observable instances of bit damage due to high whirl as shown in Figure 4.2 this damage would reduce the overall ROP for the run and should be avoided. Thus, the goal was to minimize MSE to drill as quickly and efficiently as possible while avoiding damage and minimizing dysfunction. Therefore, even though ROP is calculated and displayed on the control panel, it is not actively used in the automation algorithm. This approach is counter to the majority of the auto driller systems in use today where the main set points used to maximize performance are WOB and ROP.

An simplified overview of the automation algorithm is shown in Figure 4.8, it starts by utilizing the WOB and RPM parameters at completion of the pilot hole. It then changes either increases or decreases one parameter after a period of time and estimates the magnitude change in MSE, reacting based on if MSE decreased, stayed the same, or increased. If the MSE for that interval decreases, then that change has made the drilling more efficient and the algorithm changes the parameter again in the same direction of the first change. If the MSE remains constant, then the drilling is just as efficient, and the algorithm selects the larger of the two parameters, preferring to maximize ROP. If MSE increases, then the change has made drilling less efficient and the system will reverse the change and continue changing the parameter in the opposite direction.

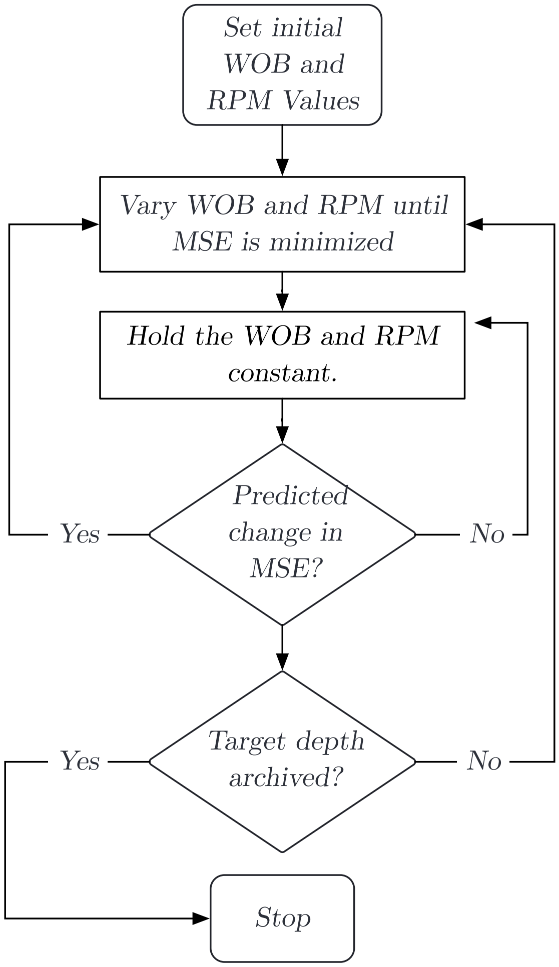

Set initial WOB and RPM Values
Vary WOB and RPM until MSE is minimized
Hold the WOB and RPM constant.
Predicted change in MSE?
Yes
No
Target depth archived?
Yes
No
Stop

Figure 4.8: Simplified MSE automation algorithm flowchart used in the lab-scale drilling rig; it was based entirely on monitoring MSE rather than ROP.

Once the parameters that result in the lowest MSE have been identified, the system holds the parameters constant until a change in MSE is predicted. A linear regression algorithm was utilized to predict the upcoming MSE within a confidence interval. If the subsequent MSE falls outside of the interval, the system concludes that there has been a change in the downhole conditions and begins the optimization process again. Figure 4.9 shows the selected WOB magnitudes and rotation speeds over the drilling time.

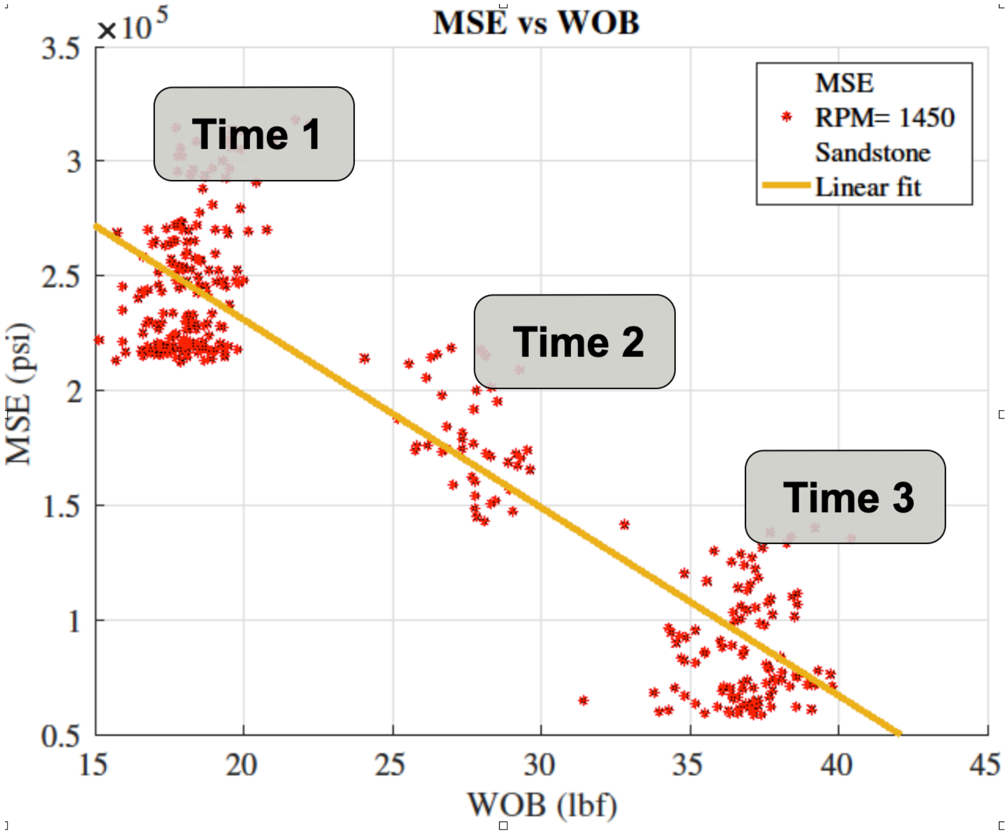


Figure 4.9: Automatically selected WOB magnitudes and rotation speeds over the drilling time, the algorithm minimizes MSE by varying WOB in this case.

In addition, focusing on MSE rather than ROP provides several key advantages. Primarily, the approach is a systemic not random approach. The algorithm does not test arbitrary parameters in hopes of stumbling on to the most suitable set of parameters, nor is there the risk of not identifying the optimum set of parameters. Secondly, once the ideal parameters have been identified, a change in MSE informs us of a change in the downhole conditions, whether that is a new formation, the onset of dysfunction, or mechanical failure. Until MSE changes the system saves time by not searching for new parameters. Based on Figure 4.10 and Figure 3.17 the automation algorithm proved successful in that it was able to quickly identify the maximum operating conditions as the ideal parameters in a short time after completing the pilot hole.

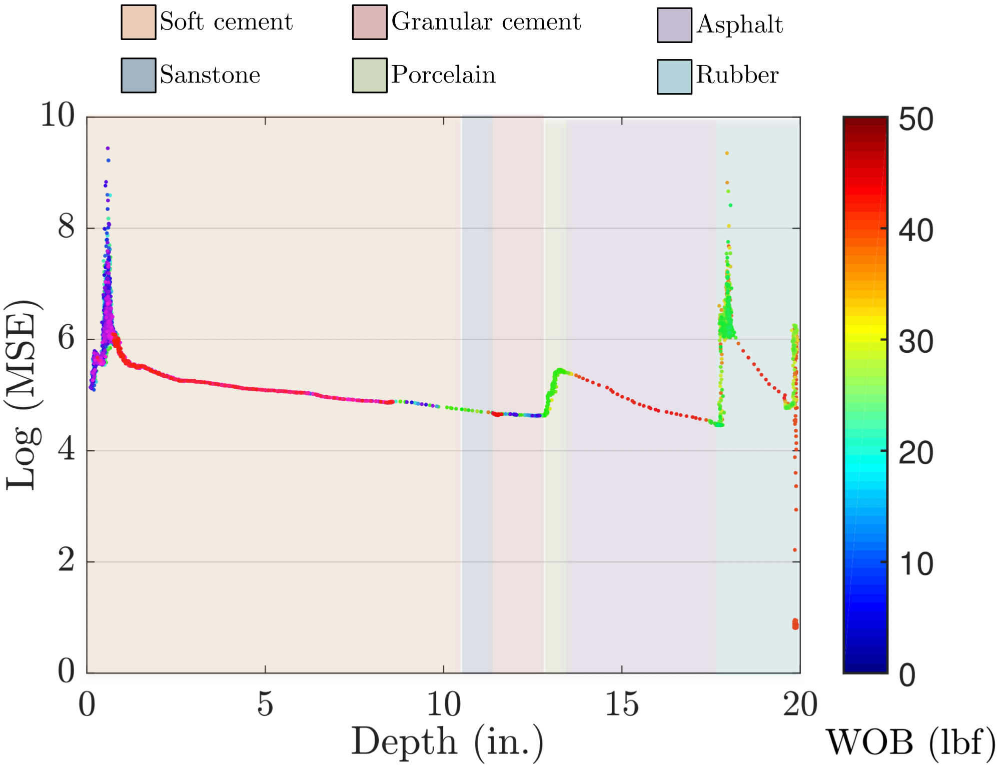


Figure 4.10: The main objective of the optimization algorithm was to drill efficiently, which was clearly achieved as the overall MSE trend is shown decreasing, even at different WOB set points, figure from [2].

4.2 Development of an Orientation Algorithm for a Downhole Drill-bit and Testing

This section follows up on work developed and discussed in Chapters 2 and 3, especially the creation, integration, and use of drilling simulations to simulate drilling conditions, measurements, and estimate downhole orientation as discussed in Sections 2.1.1, 2.3, and 2.4. The following paragraphs provide a summary of the research work and tasks performed as part of the development, validation, and testing of a Drill Bit Motion Model (DBMM) for an onshore horizontally drilled wellbore. In addition to testing using virtual sensors discussed in Section 2.3, one of the main objectives of this research work was to validate and test the filter using actual prerecorded sensor data from a drilled wellbore. To archive this goal the following tasks were performed:

1. Converted written Python prototype software to a C++ Dynamic link library (DLL).
2. Validated C++ version of the DBMM model against the Python code.
3. Updated the DBMM architecture to support processing variable input data rate, and multiple binary data files (DBS) with reset capabilities.
4. Researched, implemented, and updated the numerical algorithms for improved performance and estimation.
5. Evaluated DBMM filter performance against a representative phone's IMU sensor suite.
6. Evaluated DBMM filter performance against a service logged DBS binary-based data.

Profile Driven Inputs

A previously developed kinematic simulator discussed in Section 2.2.6 that relies on user provided inputs to drive the velocities that affect the dynamics of the motion was used as a generic profile generation routine to ease the scripting of a given configuration input. In particular, the time history of the weight on bit (WOB) and the bit angular velocity are read from a profile table as the simulation propagates the states.

These inputs directly affect the rate of penetration in the downhole direction according to:

$$\begin{aligned} v_{\text{rop}} &\propto f_{\text{wob}} && (4.2) \\ &\propto \omega_z\,, && (4.3) \end{aligned}$$

where f_{wob} is the commanded downward weight-on-bit topside and ω_z is the commanded angular velocity of the bit about the drill string.

The direction of the rate of penetration is defined to be the bit's z axis, which depends on the full 3-DOF bit angular velocity $\mathrm{\omega}_{\mathrm{K}}$. Thus, we define the bit reference trajectory using:

$$ {}_{\mathrm{P}}\mathrm{v}_{\mathrm{K}}{}^{*} = {}^{\mathrm{P}}_{\mathrm{K}}\mathrm{T}^{*} \begin{bmatrix} 0 \\ 0 \\ v_{\text{rop}} \end{bmatrix}, \tag{4.4}$$

With,

$$v_{\text{rop}} = k\omega_z f_{\text{wob}}\,, \tag{4.5}$$

And where the asterisk denotes the reference trajectory. As always, the attitude is propagated as:

$$\frac{d}{dt}\,{}^{\mathrm{K}}_{\mathrm{P}}\mathbf{q}^{*} = \frac{1}{2}\Omega\left(\mathrm{\omega}^{*}_{\mathrm{K}}\right)\,{}^{\mathrm{K}}_{\mathrm{P}}\mathbf{q}^{*} \tag{4.6}$$

$$\Omega(\mathrm{\omega}) = \begin{bmatrix} 0 & -\mathrm{\omega}^{\mathsf{T}} \\ \mathrm{\omega} & -[\mathrm{\omega}\times] \end{bmatrix} = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix}. \tag{4.7}$$

The drilling profiles were generated by selecting a 'baseline' profile and then scaling it by the initial value and timeframe required. Baseline profiles can be added by defining a set of frequencies and amplitudes as well as a clipping constant. Thus, the simulation was able to generate a profile that appears to increase or decrease to a specified value and then hold that value for some amount of time, this is, actively drilling, then holding position. Futhermore, these profiles can be automatically generated for Monte Carlo sensitivity analysis purposes.

Definitions

This work builds upon the complementary filter MARG described in [174]. Using q_{acc} and q_{mag} to indicate the case-to-z-down and z-down-to-neD transformations. In the case frames, we treat $\hat{z}$ as pointing downhole; and neD is the magnetic north version of the NED frame. In the absence of magnetometer data, the complementary filter stores attitude in a z-down frame which may or may not be identical to the neD frame, but at most only differs by an about-z rotation.

We will define:

$$
\begin{aligned}
{}^{\mathrm{RD}}_{\mathrm{R}}\mathbf{q} &= \mathrm{q}_{\mathrm{acc}_{\mathrm{R}}} && (4.8)\\
{}^{\mathrm{UD}}_{\mathrm{RD}}\mathbf{q} &= \mathrm{q}_{\mathrm{mag}_{\mathrm{R}}} && (4.9)
\end{aligned}
$$

For each $\mathrm{R} \in \{\mathrm{J}, \mathrm{K}\}$ (mud motor and bit case frame, respectively). Here, ${}^{\mathrm{J}}_{\mathrm{K}}\mathbf{q}$ is the known transformation from bit case frame to mud motor frame; note that this formulation requires a fixed bend between the sub and bit.

We begin by computing $\mathrm{q}_{\mathrm{acc}}$ for the bit and sub (case frame B):

$$
\begin{aligned}
{}^{\mathrm{KD}}_{\mathrm{K}}\mathrm{T}\,{}_{\mathrm{K}}\tilde{\mathrm{a}} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}} && (4.10)\\
{}^{\mathrm{JD}}_{\mathrm{J}}\mathrm{T}\,{}^{\mathrm{J}}_{\mathrm{B}}\mathrm{T}\,{}_{\mathrm{B}}\tilde{\mathrm{a}} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}}. && (4.11)
\end{aligned}
$$

It is also assumed that the K and J frames have their z axes aligned — that is, pointed in the same direction ($\hat{\mathrm{z}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}}$), allowing us to write:

$$
\begin{aligned}
{}^{\mathrm{KD}}_{\mathrm{K}}\mathrm{T}\,\hat{\mathrm{z}} &= {}_{\mathrm{KD}}\begin{bmatrix} z_x \\ z_y \\ z_z \end{bmatrix} = {}_{\mathrm{KD}}\mathrm{Z} && (4.12) \\
{}^{\mathrm{JD}}_{\mathrm{J}}\mathrm{T}\,\hat{\mathrm{z}} &= {}_{\mathrm{JD}}\begin{bmatrix} z_x \\ z_y \\ z_z \end{bmatrix} = {}_{\mathrm{JD}}\mathrm{Z}\,. && (4.13)
\end{aligned}
$$

Now, we use the definition of $\mathrm{q_{mag}}$ to compute an intermediate z-down frame which we call simply D. However, instead of passing our $\mathrm{q_{mag}}$ function ${}^{\mathrm{RD}}_{\mathrm{R}}\mathrm{T}\,{}_{\mathrm{R}}\tilde{\mathrm{b}}$, we pass it the $\hat{/}$., .z vector transformed to the respective RD frame. $\mathrm{q_{mag}}$ then computes the about-z rotation from the RD frame to the 'common' z-down frame UD. Which now have a path to compute:

$$
{}^{\mathrm{K}}_{\mathrm{B}}\mathbf{q} = {}^{\mathrm{K}}_{\mathrm{KD}}\mathbf{q} \otimes {}^{\mathrm{KD}}_{\mathrm{UD}}\mathbf{q} \otimes {}^{\mathrm{UD}}_{\mathrm{JD}}\mathbf{q} \otimes {}^{\mathrm{JD}}_{\mathrm{J}}\mathbf{q} \otimes {}^{\mathrm{J}}_{\mathrm{B}}\mathbf{q}\,. \tag{4.14}
$$

Moreover, we can easily compute $\mathrm{q_{mag}}$ from UD to neD using either magnetometer.

4.2.1 Filter Algorithm Description and Implementation

The filter processes accelerometer, gyro, and magnetometer measurements to produce estimates of inclination, azimuth, and toolface vs. time. The magnetometer and accelerometer are used to detect the bit orientation with respect to magnetic North and Down, and the gyro is used to propagate the local frame rotation and resolve the toolface rotation state. The MARG filter architecture is designed to provide improved performance in the presence of magnetic interference. By decoupling how the individual sensor measurements are processed, the MARG allows an accurate estimate of inclination and toolface, constraining magnetic field disturbances primarily to a degradation of performance in azimuth. The algorithm is predominantly based on prior work published [175], but adapted and tested for downhole dynamic conditions. The quaternion-based formulation allows for increased computation efficiency and a decrease in orientation errors, as operations can be represented in matrix and tensor forms. All of which, have been previously been optimized in most general processing hardware.

Unit quaternions can be used for representing a rotation of angle θ and the axis $\hat{u}$.

$$q = \begin{Bmatrix} cos(\frac{\theta}{2}) \\ \hat{u}sin(\frac{\theta}{2}) \end{Bmatrix} = \begin{Bmatrix} q_w \\ \bar{q} \end{Bmatrix} \tag{4.15}$$

Applying the quaternion, the rotation of a vector by an angle θ, around unit vector $\hat{u}$, can be computed as:

$$\bar{v}_2 = q \otimes \bar{v}_1 \otimes q^* \tag{4.16}$$

Where q^* is the inverse rotation or conjugate quaternion given by:

$$q^* = \begin{Bmatrix} cos(\frac{\theta}{2}) \\ -\hat{u}sin(\frac{\theta}{2}) \end{Bmatrix} = \begin{Bmatrix} q_w \\ -\bar{q} \end{Bmatrix} \tag{4.17}$$

The MARG filter state is given in terms of a quaternion, q global to local (${}^G_L q$), where the local frame (L) is defined as the sensor frame which is fixed with respect to the bit or sub and the global frame (G) is the North-East-Down (NED) frame; "North" in this context is Magnetic North uncorrected for the local variation from Geographic True North. For the sake of clarity, a rigorous description of the rotation is such that: (${}^G_L q$) represents a rotation from the NED frame to the local frame, that gives to the orientation of the local frame with respect to NED. The q global to local quaternion can be used to map a vector expressed in the local frame to the global frame:

$${}^G\bar{v} = {}^G_L q \otimes {}^L\bar{v} \otimes {}^G_L q^* \tag{4.18}$$

Similarly by converting the quaternion to a rotation matrix:

$$R({}^G_L q) * {}^L\bar{v} = {}^G\bar{v} \tag{4.19}$$

Under quiescent conditions when the bit is steady or stationary and the external forces acting on the bit are small, the accelerometer can be used to give an estimate of the local gravity vector:

$$R({}^G_L q) * {}^L\bar{a} = -{}^G\bar{g} \tag{4.20}$$

Similarly, the magnetometer gives a measure of the magnetic north vector when local magnetic field disturbances are small:

$$R({}^G_L q) * {}^L\bar{m} = {}^G\bar{b} \tag{4.21}$$

where

${}^L\bar{a}$ is the measured acceleration in the local sensor frame.

${}^G\bar{g}$ is the gravity vector in the NED frame, i.e. [0 0 1].

${}^L\bar{m}$ is the measured magnetic field in the local sensor frame.

${}^G\bar{b}$ is the direction of magnetic North in the global NED frame.

These accelerometer and magnetometer measurements are processed in a way that they contribute to separately components of the $({}^G_L q)$ quaternion. The magnetometer is first rotated to an intermediate representation L, using the accelerometer, so that the L z-axis is aligned with the global z-axis (down), isolating the horizontal magnetic field contribution into the x/y components of L. Then the azimuthal rotation is computed that rotates $L = [l_x, l_y, l_z]$ to a north-down $[l_{xy}, 0, l_z]$.

The angular rate measurements from the gyro $({}^L w)$ are then used to propagate the dynamic rotation of the local frame with respect to NED in-between times when the accelerometer and magnetometer can be used to give clean reliable estimates of the current rotation to the global reference North and Down axes directions.

$$q(t_k) = q(t_{k-1}) + \frac{dq}{dt} * \Delta t \tag{4.22}$$

$$q(t_k) = q(t_{k-1}) - 1/2 * {}^{L}w \otimes q(t_{k-1}) \tag{4.23}$$

$$q(t_k) = q(t_{k-1}) - \frac{1}{2} * \begin{vmatrix} 0 & w_x & w_y & w_z \\ -w_x & 0 & w_z & w_y \\ -w_y & w_z & 0 & -w_x \\ -w_z & -w_y & w_x & 0 \end{vmatrix} * \begin{vmatrix} q_w \\ q_x \\ q_y \\ q_z \end{vmatrix} \tag{4.24}$$

To convert the $({}^{G}_{L}q)$ rotation to inclination, azimuth, and toolface angle a 3-2-3 Euler angle sequence is applied:

$$R({}^{G}_{L}q) = R_3(toolface) * R_2(inclination) * R_3(azimuth) \tag{4.25}$$

Assuming the nominal orientation ($\mathbf{R}({}^{G}_{L}q) = [I]_{3x3}$) aligns the bit vertically down with toolface zero pointing North, a Z-axis rotation by the azimuth angle, followed by a rotation about the intermediate Y-axis by the inclination angle, and finally a rotation about the local Z-axis by the toolface “roll" or “rotation" angle gives the orientation of the bit with respect to the global NED fame.

Note that internal to the MARG filter, much of the quaternion computation is implemented in terms of the conjugate rotation q local to global ($q^* = {}^{L}_{G}q$). As a result, it is necessary the MARG q^*, is propagated as a function of $q^*(t_{k-1})$ and the negative of the measured gyro observation, -(${}^{L}w$). At the completion of the *ProcessMeasurements()* step, ${}^{G}_{L}q$ is returned by computing the inverse of the conjugate rotation: $q = (q^*)^*$. Also note that Equation 4.20 and Equation 4.24 above include corrections to the reference documentation used as the primary basis for this work [175]. In Equation 4.20, the vector direction observed by a stationary accelerometer is Up not Down. As a result, it is necessary for the DBMM filter to process the negative of the measured accelerometer to properly estimate inclination given the formulation of the equations derived from the reference.

Therefore, Equation 4.24 includes a missing negative (${}^{1}/_{2}$) term in the expansion of the quaternion derivative.

Sensor Data Inputs

The MARG DBMM filter sensor data inputs are passed into the filter as a C++ standard queue, std::queue, of the DBMMSensorInput class. Each sample of recorded data (accelerometer, gyro, magnetometer, and time) is stored using units of meters per second squared, radians per second, teslas, and seconds respectively. Then these individual samples are stacked in a queue for all samples over the interval of interest. The DBMM filter is passed a pointer to the entire queue of data and is only ever operating on ten doubles at a time (3x accelerometers, 3x gyros, 3x magnetometers, and time). The vibrometer is included in the sample set but is not currently used in the MARG DBMM filter.

```
class DBMMSensorInput
{
    double time;                              ///< Time of the measurement set (seconds)
    std::array<double, 3> magnetometer;       ///< Magnetometer sensor input (teslas)
    std::array<double, 3> gyro;               ///< Gyro sensor input (radians per second)
    std::array<double, 3> accelerometer;      ///< Acceleration sensor input (meters per second squared).
    std::array<double, 3> vibration;          ///< Vibration sensor input (meters per second squared).
};
```

Typical Usage

Once the sensor data input queue is specified, the user call run the *ProcessMeasurements()* function to run through the entire list of data in the queue. The DBMM will return a textsc-std::vector of DBMMOutput results, one for every input time sample.

```
class DBMMFilterOutput
{
public:
    double azimuth;      //< Azimuth angle (radians)
    double inclination;  //< Inclination angle (radians)
    double toolface;     //< Toolface angle (radians)
};
```

```
class DBMMOutput
{
public:
    double met_time;                              //< Timestamp from beginning of file
    int n_measurements;                           //< Number of measurements processed.
    std::array<double, 4> q_global_to_local;      //< Quaternion from NED to Sensor Frame (scalar first )
    DBMMFilterOutput filterOutput;                //< Filter output expressed in azim/incl/toolface angles
    DBMMFilterDebugOutput dbg;                    //< Debug output
};
```

Error Messages

The MARG DBMM filter has optional inputs for handling messages reported by the filter. The user can create a std::queue mapped to a std:string for error messages and log messages to capture insight into issues and status being reported. If a message handler queue is not created, the filter will process the input measurement data without storing its corresponding error or status messages.

```
DBMMMain myDBMMMain;


 testerror_ptr  = std :: make_shared<std::queue<std :: string >>();
myDBMMMain.SetErrorMessages(test_error_ptr); // Set the error message queue


 test_log_ptr = std :: make_shared<std::queue<std :: string >>();
myDBMMMain.SetLogMessages(test_log_ptr);   // Set the log message queue
```

Error Reporting

If the MARG DBMM filter encounters an error, a corresponding error code values in Table 4.1 will be returned:

Error Code	Error Description
-100	Sensor data input pointer not set
-101	DBMM output data pointer not set
-102	Sensor data queue contains no data
-200	Negative timestep
-201	Large timestep (default: dt>0.01s)

Table 4.1: Error codes returned by the MARG DBMM filter testing routine.

Reset Functionality

The MARG DBMM Filter can likewise be called in a recurring loop, processing and returning results on individual samples or small blocks of samples at a time. In this mode, the user can review the results after each step and re-initialize the filter if necessary by calling the *ResetFilter()* function. This forces a re-initialization of the local frame orientation with respect to NED using the accelerometer and magnetometer measurements provided at the beginning of the next *ProcessMeasurements()* step.

Configurable Options

Configurable settings for the MARG DBMM filter can be assigned using an input pointer to the DBMMSettings class. Descriptions for the individual configuration parameters and their default values are given below:

```
class DBMMSettings
{
    float big_dt_tolerance = 0.01f;        ///< Largest allowable timestep between samples. Default 0.01
    float lerp_threshold = 0.9f;           ///< Spherical interpolation threshold. Defaults to 0.9
    bool use_magnetometer = true;          ///< Flag to disable processing of magnetometer data. Default True
    bool use_adaptive_gain = true;         ///< Accelerometer LPF adaptive gain control. Defaults to True
    float adaptive_gain_error1 = 0.1f;     ///< Lower gain for accelerometer gravity detection. Default 0.1
    float adaptive_gain_error2 = 0.2f;     ///< Upper gain for accelerometer gravity detection. Default 0.2
    float a_cutoff = 0.001f;               ///< Accelerometer gain if use_adaptive_gain=false. Default 0.001
    float b_cutoff = 0.01f;                ///< Fixed cutoff gain for magnetometer LPF. Defaults to 0.01
    std::array<float, 3> r_sensor = { 0, 0, 0 };   ///< Position of sensor (meters). Defaults to (0,0,0)
};
```

4.2.2 Discussion of Testing Using Prerecorded Downhole Dynamic Data.

Deployment of the DBMM filter as a C++ DLL was completed. The C++ DLL was bench tested for runtime performance on a Windows laptop; it was determined that the filter is able to load and process an hour of recorded data sampled at $1kHz$ in approximately $4-6$ seconds of run time. The filter can process any amount of recorded data from the gyro, accelerometer, and magnetometer samples recorded at user-specified sampling frequency while maintaining a constant memory footprint. The memory is required to load and store the variable-sized input; the output data are handled by the user external to the DBMM filter DLL.

The DBMM filter was validated against the Python version of the prototype and then updated for performance running against prerecorded data. During quiescent periods, the filter was able to estimate valid inclination, azimuth, and toolface histories. However, it was determined that either significant magnetic field interference was present while drilling or the magnetometer used to record the data wasn't properly calibrated, which degraded the filter's ability to reliably estimate azimuth. Nevertheless, this demonstrates a key property of the filter; isolating the effect of magnetic disturbance to the azimuth channel allowed an estimate of inclination and toolface to be reliably generated.

In order to characterize the filter performance under ideal conditions, the raw sensor data was collected from accelerometer, gyro, and magnetometer measurements from the phone's IMU and run through the filter. Dynamic test cases rolling through the full range of motion individually in inclination, azimuth, and toolface were evaluated to demonstrate the capability of rotating through any orientation.

4.2.3 Testing MARG Filter Using a Mobile IMU Sensor.

The objective of the following tests was to verify the performance of the MARG filter with real sensor data under ideal conditions in an environment with negligible magnetic interference and predefined operational conditions.

IMU Test Description

Gyroscope, magnetometer, and accelerometer measurement data were collected from a phone's IMU at a sampling frequency of 100Hz. The accelerometer and gyro sensors used in this model are a variant of Bosh's 6-DOF BMI160 [176]. The exact magnetometer specifications are unfortunately difficult to pinpoint since all motion sensors have been built into Apple's system on a chip. Several cases were performed to isolate rotations independently in each rotational axis and the unfiltered sensor data was processed using the MARG C++ filter for each case.

A summary of the test cases performed in this experiment is listed in Table 4.2. The phone's Z-axis was pointing to the magnetic North while its Y-axis pointed to the East as initial conditions for all tests shown in this memo.

Test	Azimuth	Change	From	To	Figure
1	North	Static			Fig. 4.11
2	North	Inclination	90	0	Fig. 4.12
3	North	Azimuth	360	0	Fig. 4.13
4	North	Toolface	360	0 -> 360 -> 0	Fig. 4.14
5	North	Rotating	North	South-west	Fig. 4.15
6	North	Rotating and Azimuth	North	North-east	Fig. 4.16

Table 4.2: Summary of Mobile IMU testing cases performed.

4.2.4 Inclination Estimation Results

Case 1: Static Conditions

As a first test case, the IMU was held fixed in a known orientation for one minute, to verify that the alignment of the sensors matched our expectations and that a steady and consistent attitude was estimated by the MARG filter. The phone IMU's Z-axis is out of the screen, Y is out to the top, and so X is to the user's right when looking at the screen. The phone is held initially screen down, X-North, and Y-East to align the local phone's sensor X-Y-Z axis with the global NED frame. The sensor was then rotated so that X turned to point South-east, Z tilts up 125 degrees and rotates 180 degrees to achieve a known non-zero orientation in each axis. Figure 4.11 shows the MARG output for this case giving the expected result in each parameter: inclination=125°, azimuth=325°, and toolface=360°.

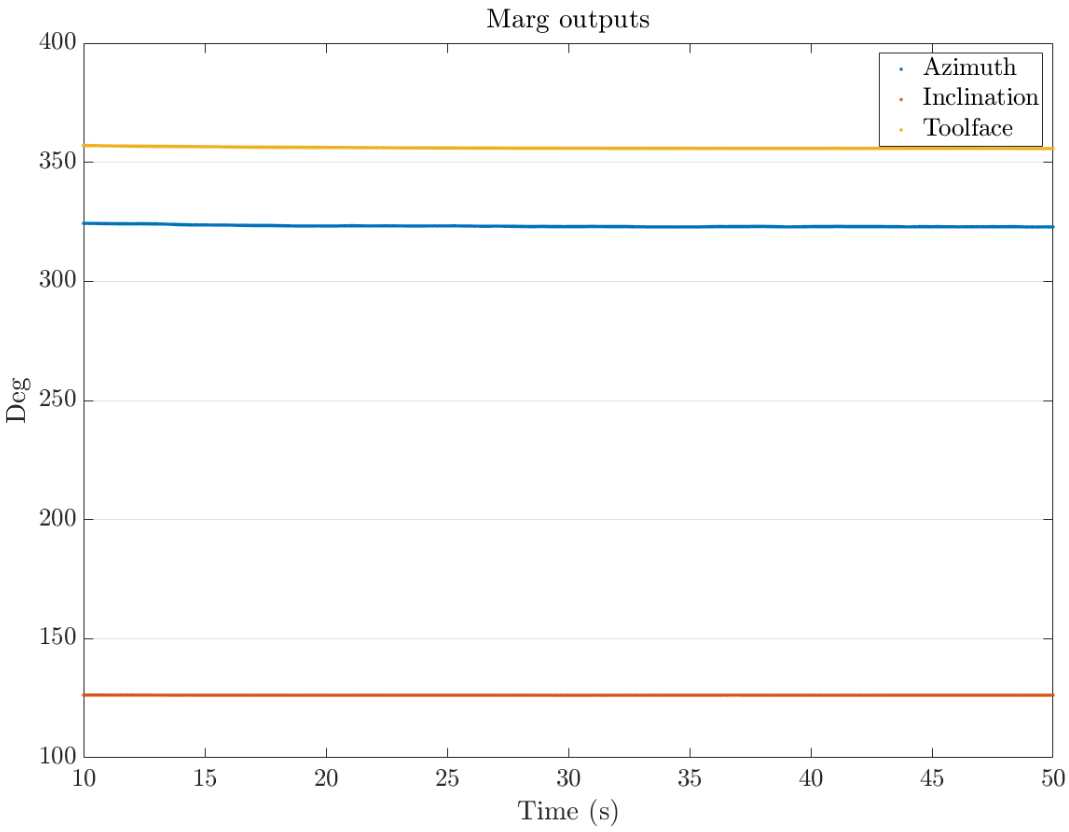


Figure 4.11: MARG filter outputs using a stationary sensor, with inclination=125°, azimuth=325°, and toolface=360°.

Case 2: Inclination Change

For the second case, the phone's IMU has rotated North-west and tilted back and forth ±25° degrees around horizontal. Figure 4.12 shows the azimuth and toolface is steady while the inclination oscillates around 90 degrees.

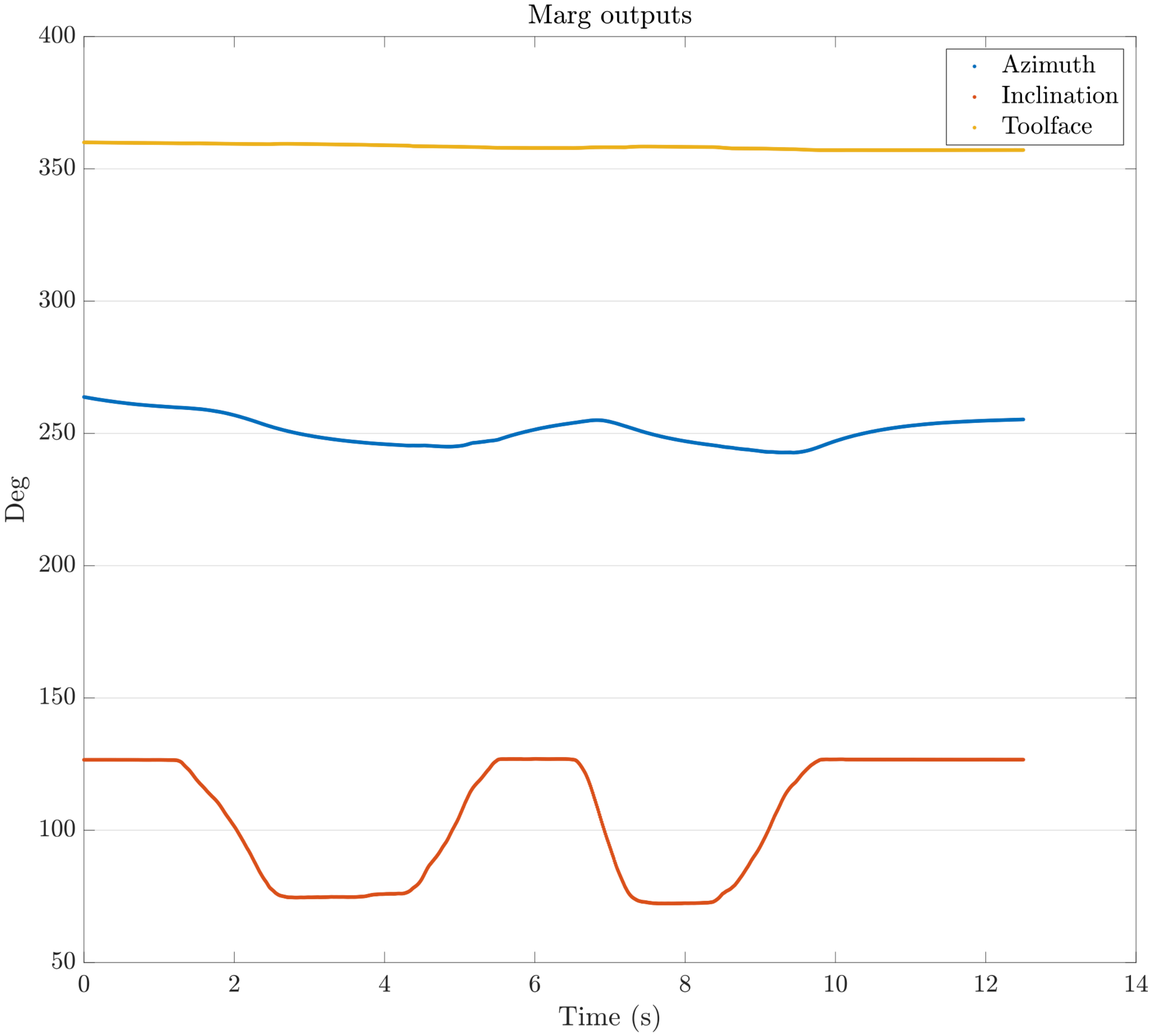


Figure 4.12: MARG outputs changing inclination from 125 degrees to 75 deg (85 to 30 deg from horizontal). The small change observed in azimuth is due to the relative location of the sensor inside the chassis to the axis of rotation. We can accurately track inclination changes even with small variations on the azimuth outputs.

Case 3: Azimuth Change

For the third case, the IMU was initialized in 90 deg inclination at zero azimuth and toolface. The phone was then turned so that the heading made a full 360 deg sweep turning East from North back to the original orientation. Figure 4.13 shows the expected variation in azimuth while the inclination and toolface are relatively fixed.

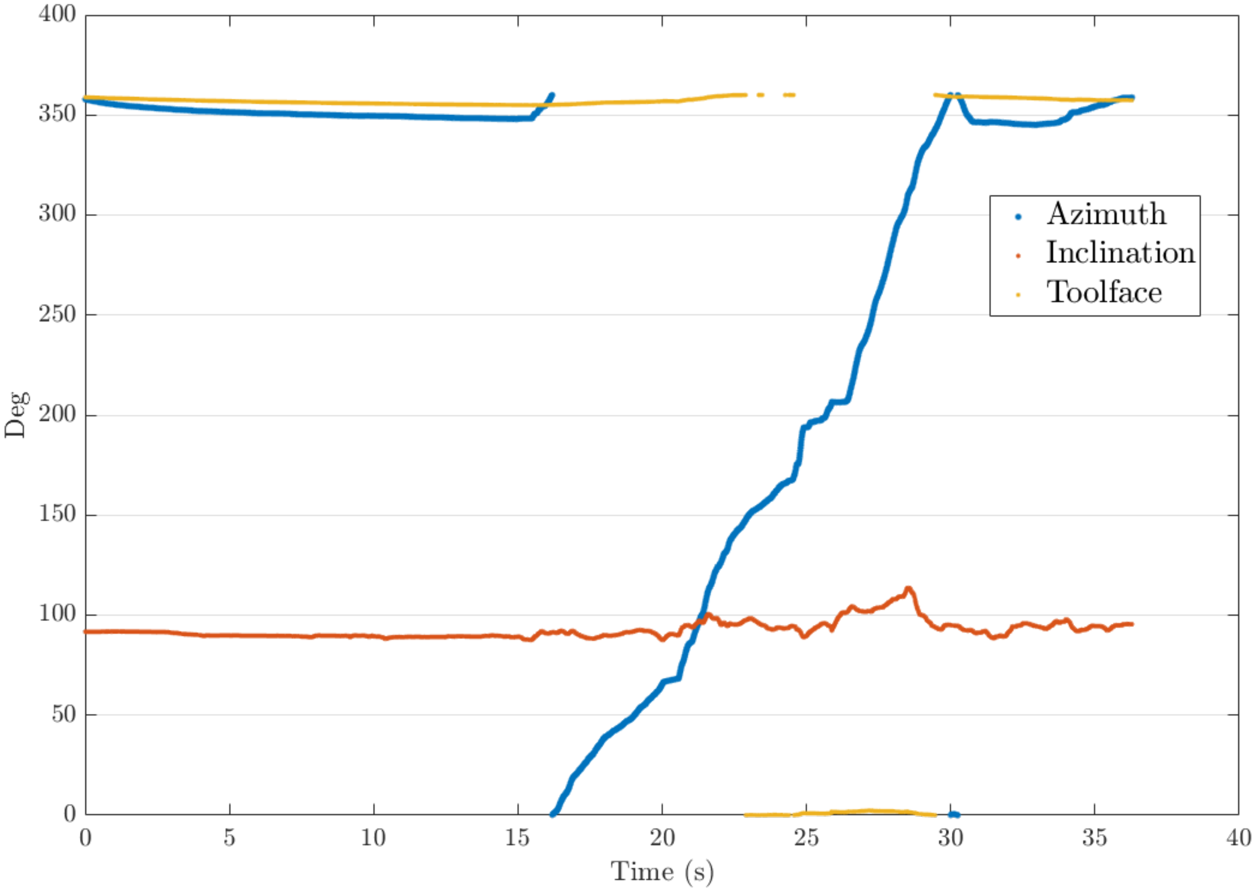


Figure 4.13: MARG outputs changing azimuth from North to East

Case 4: Toolface Change

For the forth case, the toolface rotation was isolated. The phone was initially oriented in a 90 degree inclination, heading North while three full revolutions in toolface were executed (two clockwise followed by one counter-clockwise). Figure 4.14 shows the expected variation in toolface while inclination and azimuth are held relatively constant.

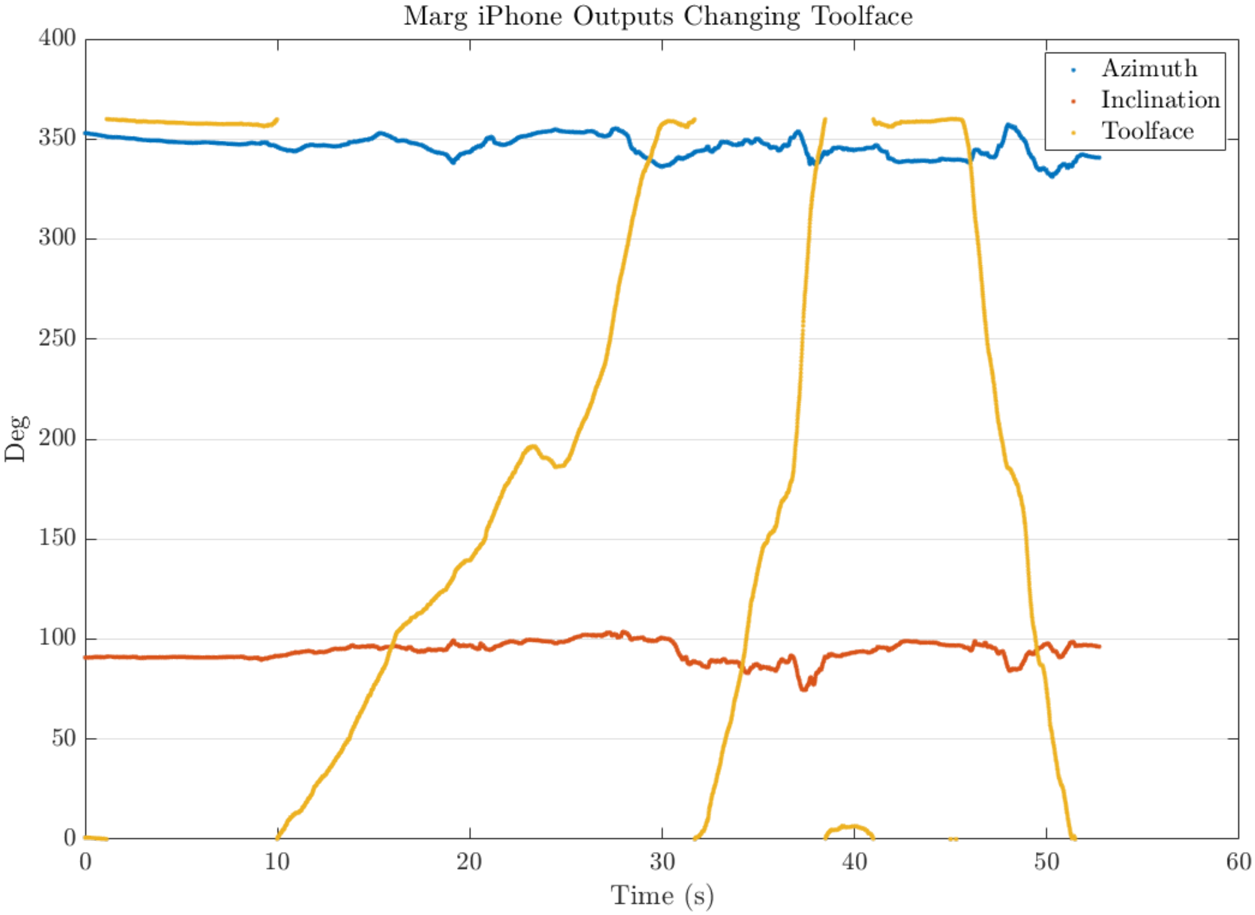


Figure 4.14: MARG outputs changing toolface two revolutions in the clockwise direction followed by one full revolution anticlockwise

Case 5: Rotating Toolface at High RPM

For the fifth case, the phone's IMU was mounted on an electric drill pointing North-west in order to evaluate the MARG performance while spinning the toolface at high RPM. The phone was held at a 90-degree inclination, and the toolface was spun through numerous revolutions. There were three intervals of spinning executed followed by a pause. Figure 4.15 shows that the inclination and azimuth are steady through the entire test, and the toolface stabilizes to a stable value after the phone stops spinning after each interval.

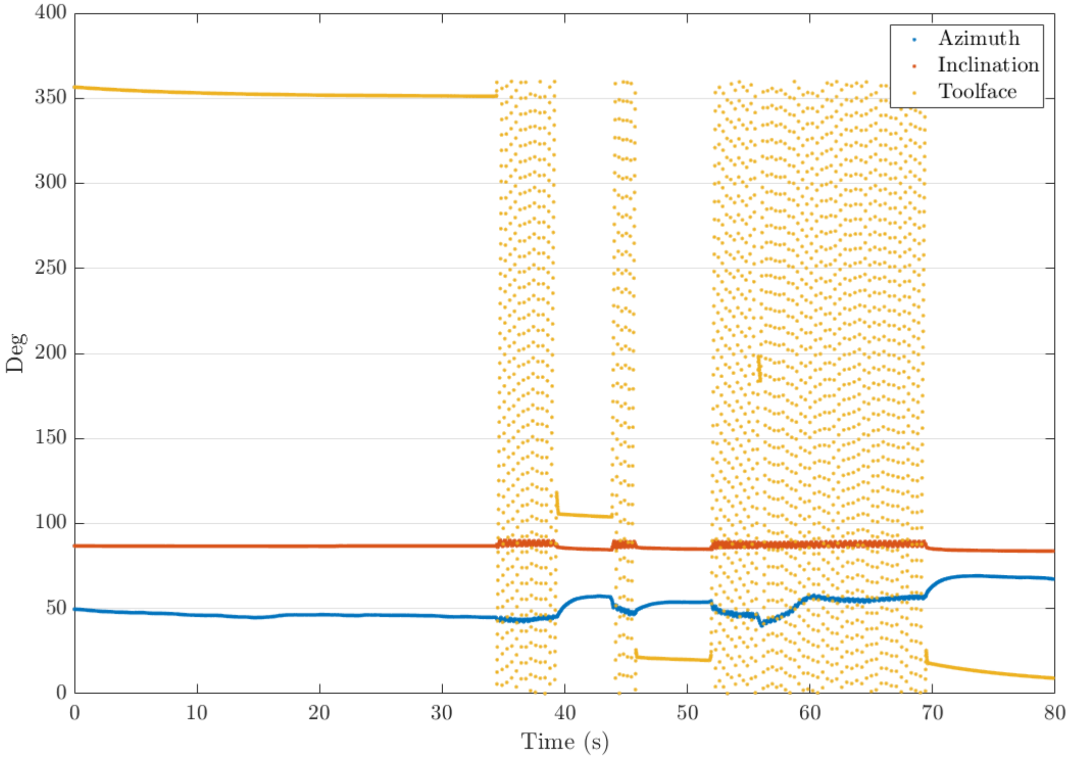


Figure 4.15: Rotating sensor at about 130 RPM and slightly varying azimuth. The MARG filter can accurately compute toolface, inclination, and azimuth while rotating at high RPM. Note that the change observed in the azimuth is real as the base used for this test wasn't fixed to the table. Also, the effect of the cutoff frequency used internally is observed since increasing the rotation speed reduces the variance on the data.

Case 6: Rotating at High RPM and Changing Azimuth

For the sixth case, the azimuth was varied while the phone was spinning at high RPM. Figure 4.16 shows that the MARG filter is able to track the azimuth change with some variance that appears to vary slightly with the toolface spin rate.

4.2.5 IMU Sensor Test Conclusions

Altogether, the phone's IMU experiments show that the MARG C++ filter is working as intended even when rotating at 140 RPM. Most disturbances are rejected under complex movement and the solution stabilizes at a much faster rate than other conventional methods [175]. These tests were conducted to provide a quick validation that the MARG DBMM filter works under ideal

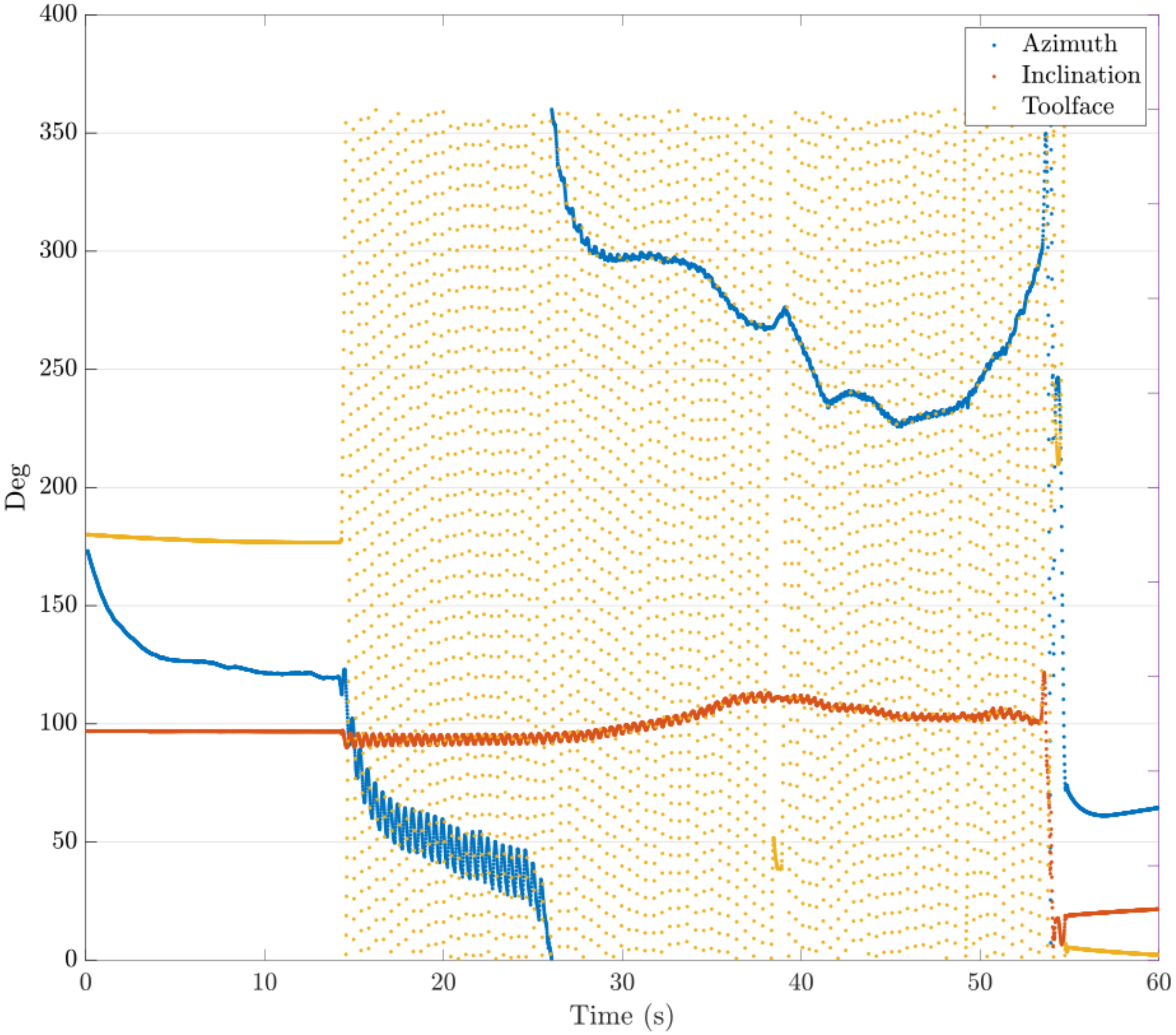


Figure 4.16: MARG Outputs, spinning from 120 RPM to 140 RPM, and moving azimuth for one full revolution. From North to East direction. Azimuth, inclination, and toolface are accurately computed while rotating. The test bench base failed around 55 seconds into the test after which the module fell about a meter to the ground.

conditions under known dynamic situations. The sensors are not the same as what is used in the original DBMM sensor suite, but they do produce the same fundamental measurements in a similar configuration which is sufficient to confirm the MARG filter functionality. The true state is not precisely recorded with the currently working test bench, so the validation is limited to a qualitative comparison vs. expected outcomes. A quantitative assessment of inclination, azimuth, and toolface error could be attained with a more sophisticated test bench in the future.

4.3 Testing Filter on Downhole Data

4.3.1 Performance Metrics

Table 4.3.1 show the performance of the C++ MARG filter running using its DBS DataReader class. The tests were run on a desktop computer using an Intel i7 8700K 6C/12T processor running at 5.1 GHz with 32GB of 3200 MHz - DDR4 memory. The filter runs at an average ratio of 1000 times faster than real-time. For instance, an hour of recorded data would take about 4 seconds to process. This ratio doesn't take into consideration writing the results to a file as such operations require the use of much slower Input/Output instructions.

A significant performance drop is observed when running the filter on 50k blocks of data because the machine runs out of physical memory and starts using the virtual page, slowing down the overall computing time. It is also worth mentioning that the current version of the DataReader class doesn't take advantage of multithreaded computing. Therefore single-core performance and frequency are the most important parameters when it comes to overall performance.

Blocks	**Compute Time - C_t (ms)**	**C_t/block (ms)**	**Realtime R_t (s)**	**R_t/C_t Ratio**
1,000	1068	1.068	1,094	1,024x
2,000	2188	1.094	2,188	1,000x
4,000	4418	1.105	4,376	990x
10,000	10843	1.084	10,940	1,008x
20,000	22369	1.118	21,880	978x
30,000	31798	1.060	32,820	1,032x
40,000	43885	1.097	43,760	997x
50,000	88936	1.779	54,700	615x

Table 4.3: MARG C++ filter performance metrics.

4.3.2 Downhole Inclination Results

In order to test the MARG DBMM filter on sensor data recorded downhole, a data reader was developed to read high frequency logged DBS files and isolate unfiltered accelerometer, gyro, and magnetometer data for processing in the DBMM DLL. For the purposes of this analysis, a specific non-disclosed wellbore was selected because it appears to be drilling for most of the file and contains several brief pauses that can be used to evaluate the transition to and from active drilling. Figure 4.17 shows a segment of data where the drill bit is stationary and a steady inclination, azimuth, and the toolface estimate is obtained, initially estimating a 90-degree inclination with a West-north-westerly heading. At time t≈4060 seconds in the file, the drill bit begins rotating, and variation of the inclination estimate is observed as the toolface starts to spin. Unfortunately, the filter, however, is not to be able to reliably estimate azimuth while drilling over long periods of time.

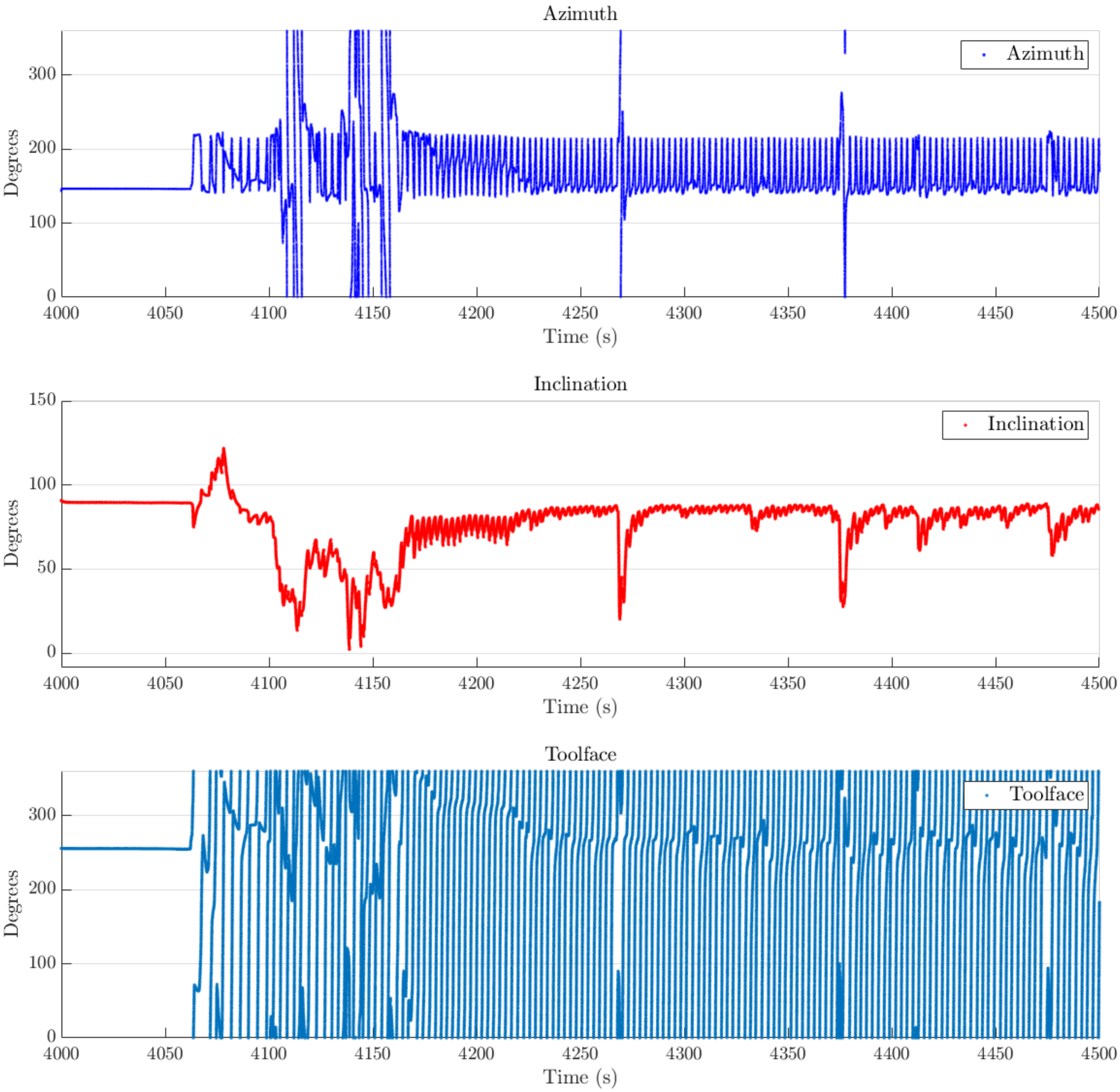


Figure 4.17: Filter outputs tested using real high-frequency downhole data from a horizontal well.

Figure 4.18 shows the unfiltered sensor data inputs used to obtain the results shown in Figure 4.17. Some potential issues were apparent with the raw sensor data calibration or conversion looking at the data in this chart. In the accelerometer, a measure of gravity can be detected in the Y-axis near 9.8 m/s^2 for the first 20 seconds while stationary; the X and Z-axes are both near zero. The vibrometer is shown for comparison against the accelerometer, but was not used in the filter;

it is reporting ~80 m/s^2 in the Y-axis, ~65 m/s^2 in the X-axis and ~40 m/s^2 in the Z-axis with significant noise of ~ ±10 m/s^2 in each channel, including measurements taken while the bit was stationary. That observation indicates that calibrating large biases and/or converting the raw data to engineering units with this sensor was not done properly. Therefore, the data from this sensor was not usable in the current form.

There was a discrepancy with the raw magnetometer readings. Measuring the Earth's magnetic field strength, a measurement on the order of 25-65 μT is expected. However, the magnetometer downhole is reporting a mean observation on the order of 1000 μT. Thus, there was either an issue converting the recorded data to engineering units or there was a local magnetic field disturbance that was more than ten times stronger than the Earth's magnetic field. The orientation of the magnetometer also appeared to be inconsistent with the available documentation provided. A subset of Figure 4.18 is shown in Figure 4.19 that shows this phenomena.

The magnetometer X-axis is aligned with the drill rotational vector, rather than the local Z-axis, which is expected. In addition, with a near-horizontal borehole, it is expected that and the alternating signal will appear as a phase shift between the other two axes orthogonal to the spin direction (the Y and Z-axes in this case). The two channels do both include an oscillating signal when drilling, but they do not have similar minimum and maximum amplitudes and are not always well correlated. Note a recommended magnetometer correction (to swap the Y and Z magnetometer axes and negate X) was also tested but did not resolve the azimuth estimation performance over longer periods of time. Further research was required to determine how to properly process the recorded magnetometer readings. Inspection of the raw gyro data provided a good indicator for the drill revolutions per minute downhole. The bit's RPM frequency is not fully stable until about t≈ 4200 seconds. The bit spins and stops periodically while ramping up speed that appears to indicate non-full stick-slip occurring downhole while the string stores and releases energy which is commonly seen during drilling.

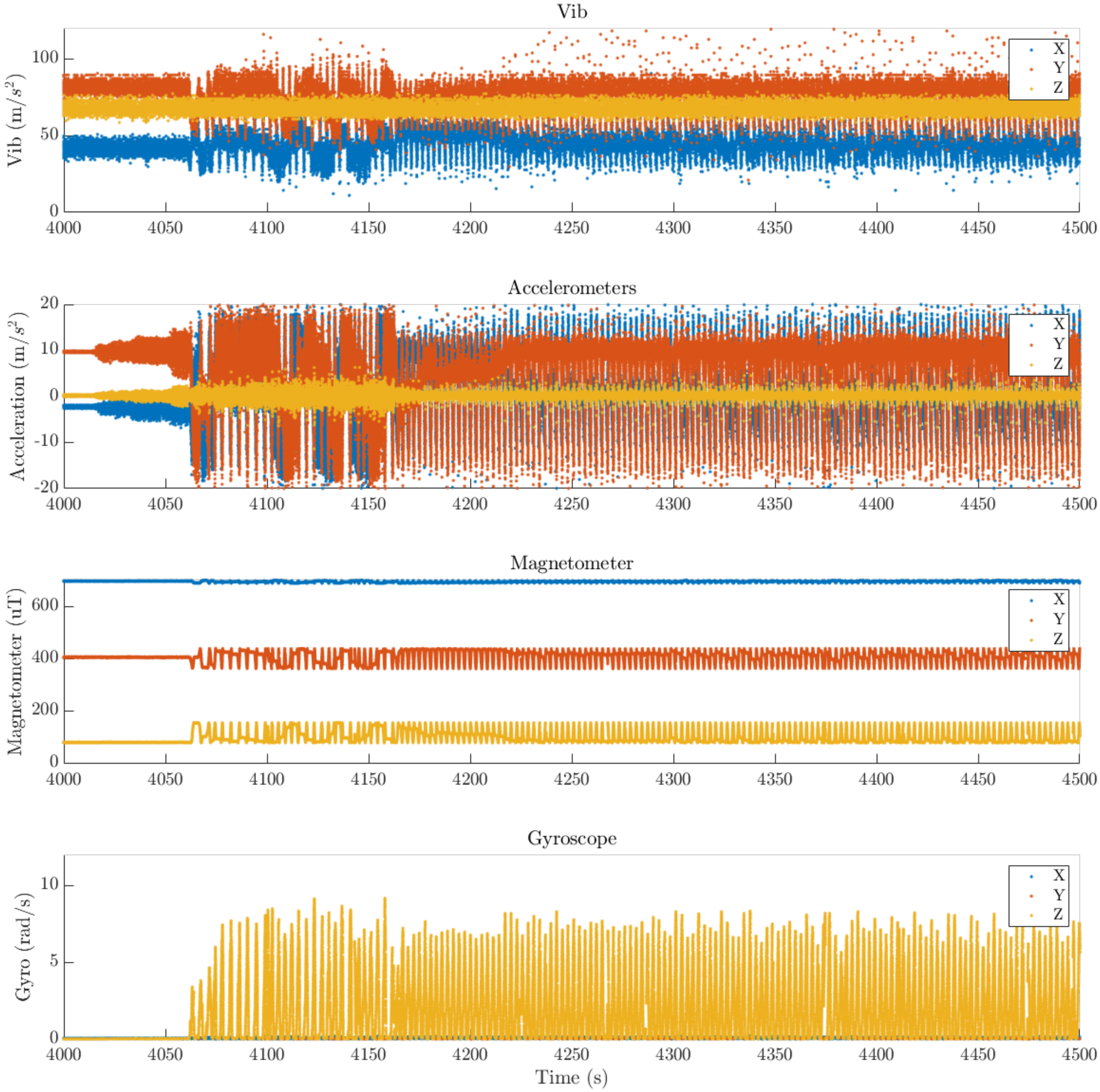


Figure 4.18: MARG filter outputs tested using high-frequency downhole data from an oil producing horizontal well – using the inverted magnetometer sensor' input definition

4.3.3 Magnetic Interference Effects

The MARG DBMM filter was able to track toolface and inclination under nominal conditions downhole. Azimuth, on the other hand, is only tracked while the bit is not rotating and there's no significant variance in the readings from the sensors. Azimuth has been proven a difficult mea-

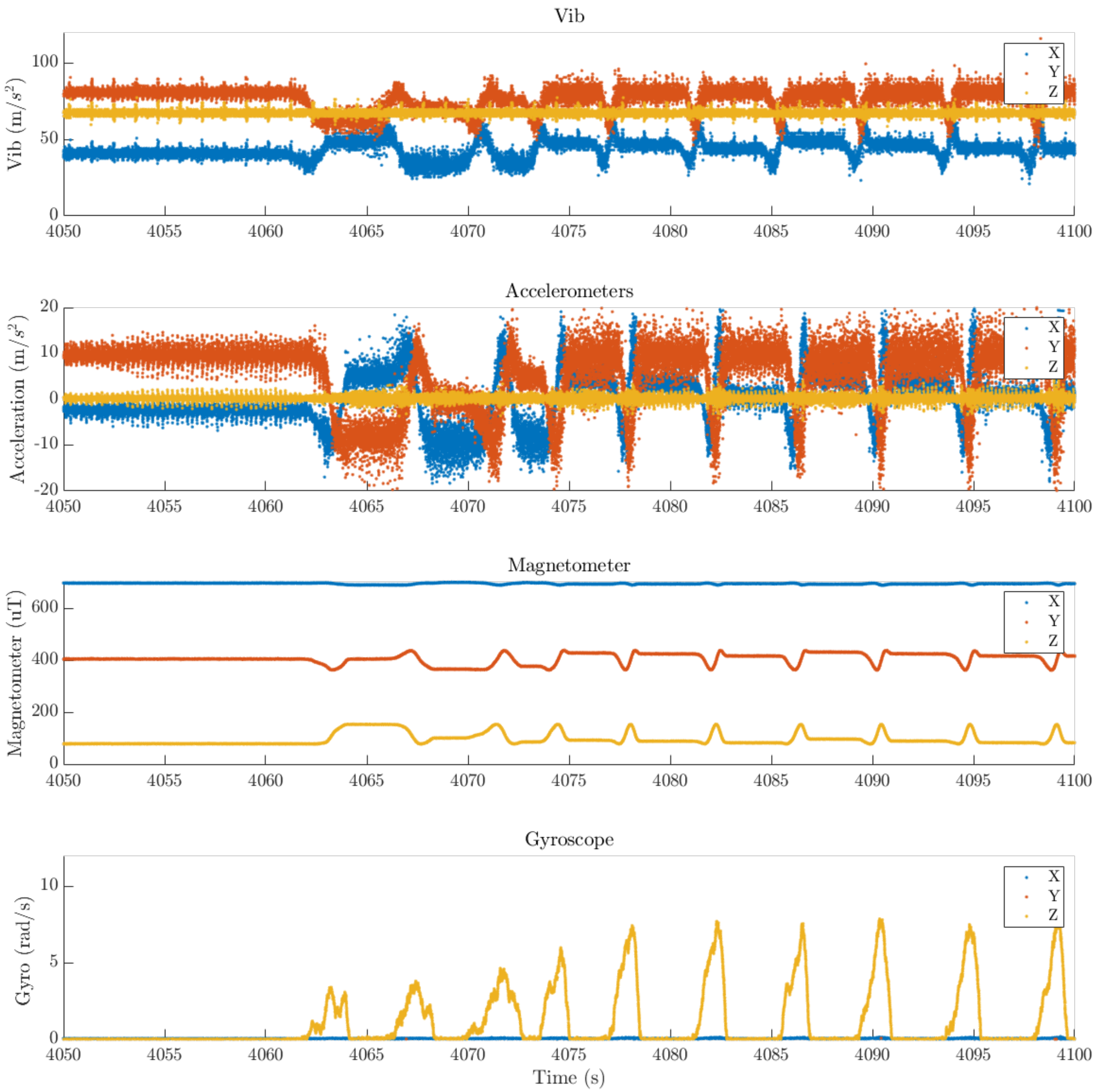


Figure 4.19: Smaller timeframe of the MARG inputs shown in Figure 4.17.

surement to derive while drilling horizontally; azimuth error increases at high inclination angles and deeper depths [177, 178, 179]. Results shown in the previous section indicate that either the magnetometer sensor used downhole is not accurately measuring the magnitude of the Earth's magnetic field or there was significant magnetic interference downhole. The large discrepancies in the magnetic field readings prevented the filter from converging on a consistent azimuth estimate while the bit is rotating, especially if run for extended periods of time.

Magnetic interference could occur downhole due to nearby magnetized steel sections in the drill assembly, an adjacent casing string, or even geological sources. The error is observed as a large variance of the azimuth outputs. The average magnetic field of the earth is about 25-65 μT. However, field observations in the literature have estimated a BHA magnetic field strength of about 8-10 μT [180]. The strength of which depends primarily on the distance between the magnetic source to the sensor and the average inclination with respect to Earth's magnetic field. A change of just 3 μT results in a 2-3 degrees error in azimuth reading under static conditions [181] while a magnetic interference of around 8 μT results in up to 20 degrees of error [182]. Azimuth error increases at high inclination angles, increasing linearly as inclination increases [183, 184].

Some baseline magnetic interference tests were conducted in order to verify the impact that magnetic disturbance has on the azimuth outputs from the MARG filter. Figure 4.3.3 shows the effect in Azimuth output caused by nearby pulsed magnetic interference. Artificial disturbances were induced using an electric drill located around 8" (20 cm) away from the magnetometer. This first test was to characterize the magnitude of disturbances that could have been induced by the phone's IMU test rig.

The MARG Outputs are plotted on the left Y-axis while raw magnetometer measurements are shown in purple and plotted against the right y-axis (μT). An error of ± 2-3 degrees is observed, the error is proportional to the ratio of the shock magnitude and the apparent Earth's magnetic field.

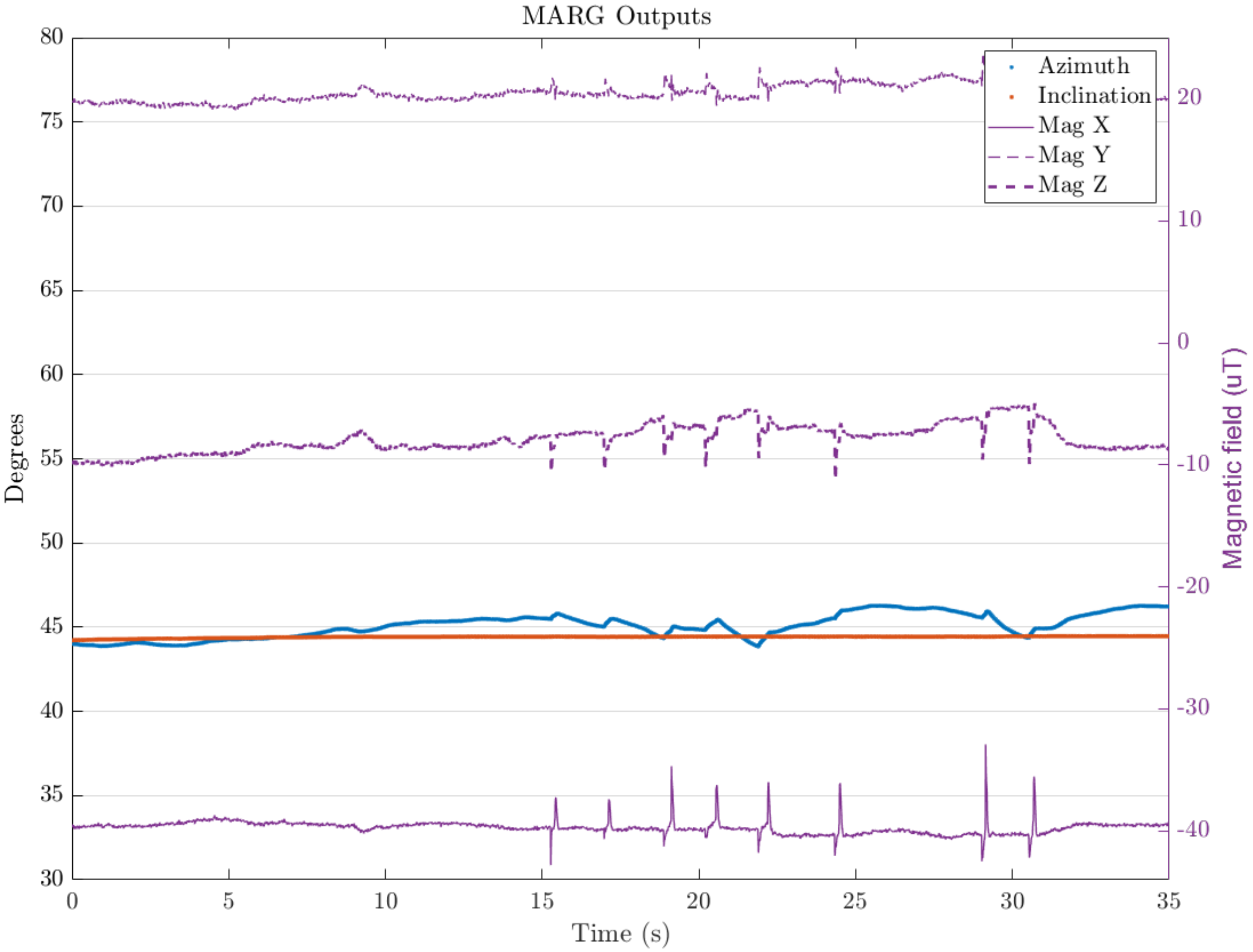


Figure 4.20: Effect of a nearby magnetic interference caused by electromagnetic pulses from an electric drill on the azimuth outputs. MARG Outputs are plotted on the left $Y - axis$ while raw magnetometer measurements are shown in purple and plotted against the right $y - axis$ (μT). This plot demonstrates a key property of the filter; isolating the effect of magnetic disturbance to the azimuth channel allowed an estimate of inclination and toolface to be maintained throughout the disturbance.

For the next test, a small magnet with a magnetic field strength of around 1800 μT was used to induce magnetic interference while the phone was resting at a ~ 90 deg inclination on a table. The objective of this test was to determine the distance at which we begin to sense the magnet's field. As seen in Figure 4.21, the sensor picks up the magnetic disturbance when the magnet is around 12.5" (32 cm) away from the sensor. The gives an indication of the relative magnitude disturbance then needs to be within a foot of the magnetometer to produce the field strength recorded downhole.

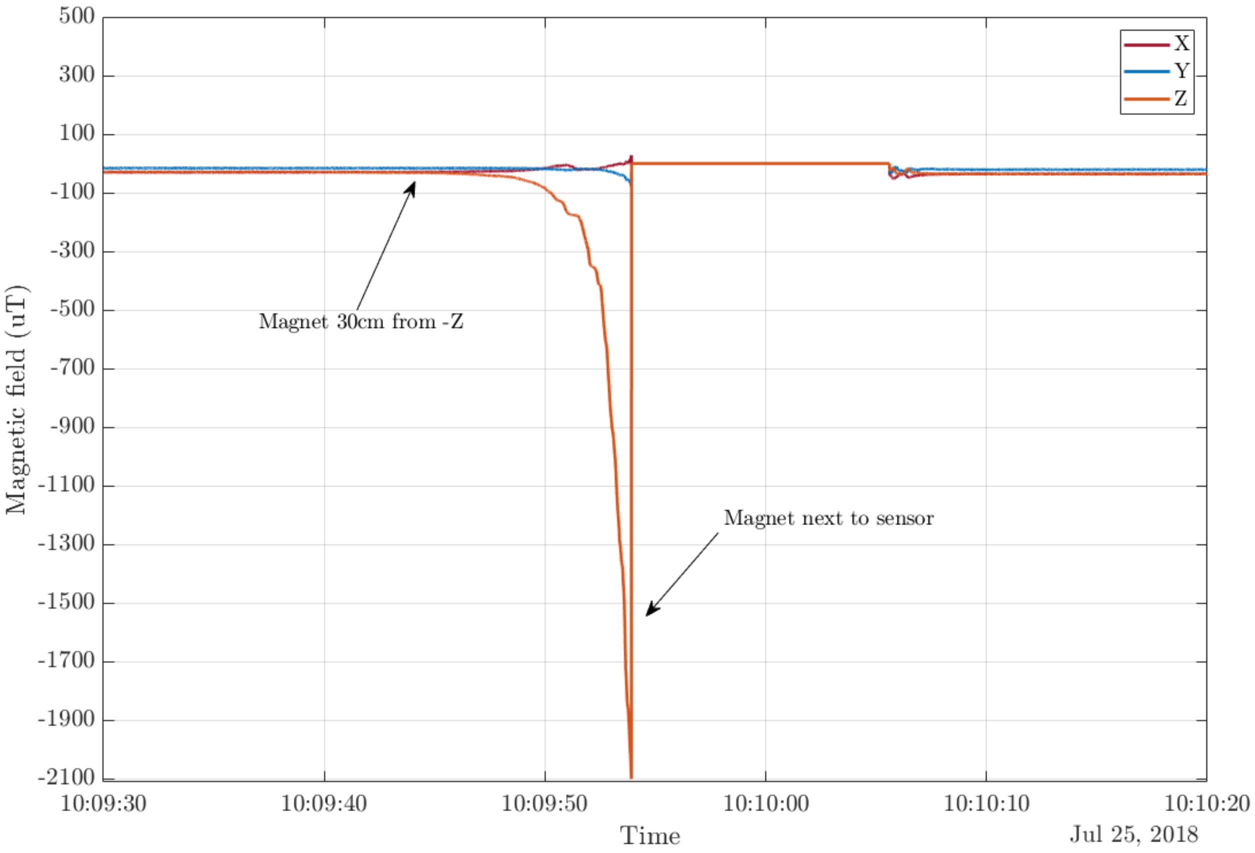


Figure 4.21: Effect of a nearby magnet on magnetometer readings at 12.5" (32 cm). The magnet had an average magnetic field strength of 1400-2200 μT.

Figure 4.22 shows the nominal magnetometer measurements while rotating. The drill briefly induces magnetic interference just before rotating. However, it does not appear to have a significant impact on the MARG outputs or the amplitude of the sine wave observed. Looking closely at the raw magnetometer measured by the phone's IMU, the phase-shifting oscillatory signal can be seen between the X and Y channels while drilling. There is a slight discrepancy between the peak field strength between these channels (within 5%), as compared to the observation in Figure 4.19 where the Y-axis data is 400% times that of the Z-axis.

Further analysis is needed to reliably estimate the azimuth downhole in the presence of substantial magnetic interference. The magnetometer variance while rotating is heavily correlated to the frequency of rotation suggesting that major disturbances could be filtered in the MARG, much as is done the accelerometer to allow sampling only when a clean measure of the local gravity field (or magnetic field in this case) is observable.

4.4 Volumetric Borehole Shape Reconstruction from High-Frequency Downhole Dynamics Measurements

An accurate indication of the drilled borehole condition and shape is essential to several decision-makers when drilling a wellbore. For instance, completion engineers favor a much less tortuous well path to reduce stuck-pipe incidents. In contrast, driller engineers tend to prefer speed over tortuosity, which can cause damage to expensive downhole equipment and even the drill bit. Nevertheless, estimating the borehole shape requires specialized measuring equipment such as measurement while drilling tools to get a caliper log that captures the diameter of the hole for that interval, which is not always available or economically feasible to conduct.

Consequently, to overcome this problem, we hypothesized that accurately reconstructing the borehole shape from downhole dynamic data should be possible if having access to it. Thus, we proposed using high frequency downhole dynamic data and the orientation estimation filter developed in Section 4.2 to estimate the bit orientation, approximate the bit-rock interaction erosion, and integrate downhole dynamics at every filter output to create a virtual 3D representation of the resultant borehole shape.

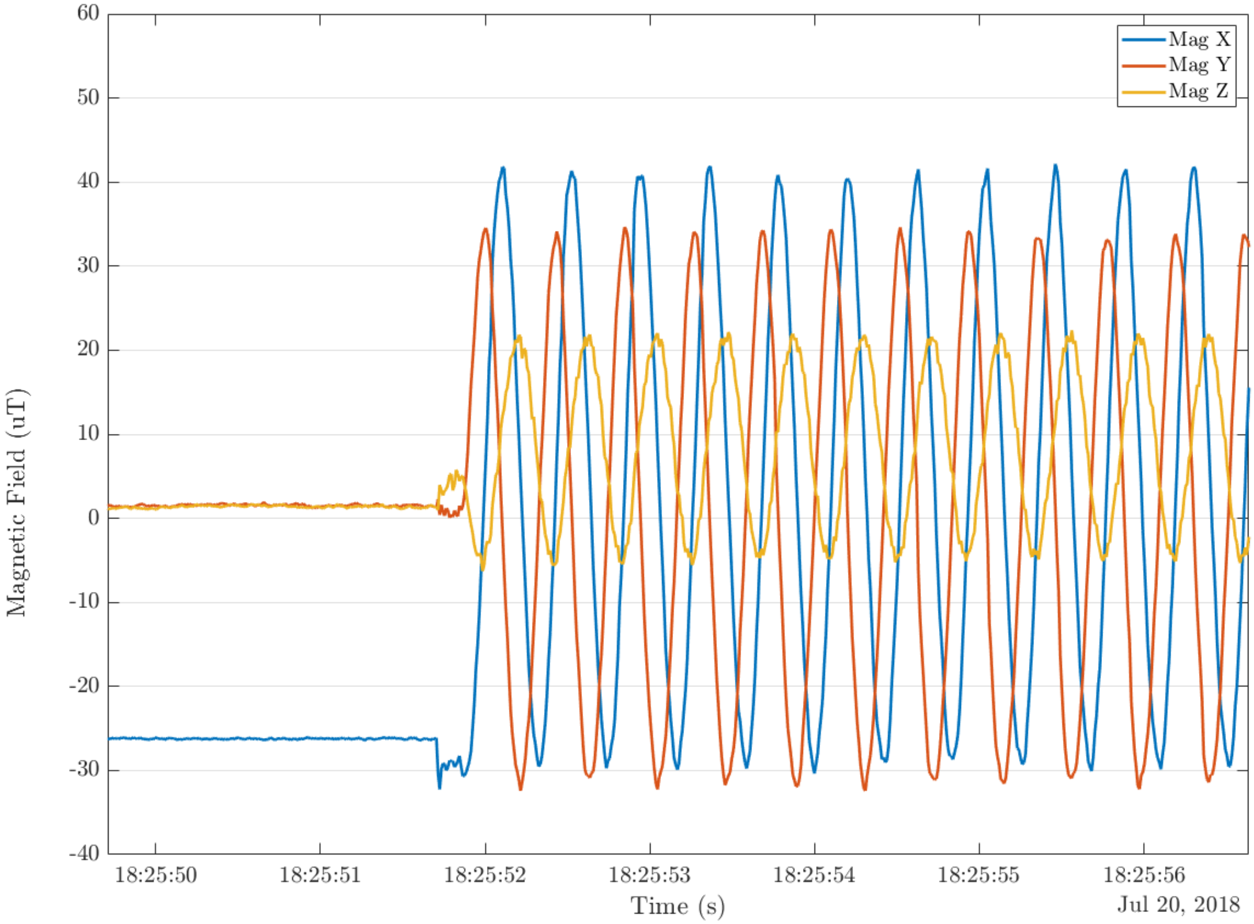


Figure 4.22: Raw magnetometer readings while rotating. Magnetometer measurements taken from Case 5 (Sec. 4.2.4) shown in Figure 4.16.

This chapter provides a summary of the engineering work outlining tasks completed as part of the development of a *Generation of Bit Hole Shape Data*, as a use case derived from the development of the previous filter discussed in Section 4.2. During this component of this work, the following tasks were performed:

1. Developed the DBMM Hole Shape Processor algorithm and software

    - Applied the MARG DBMM filter results with provided time and depth displacement logs and a provided bit cutting shape to generate a hole shape history.

    - Extracted and integrated downhole dynamics data from the MARG filter output leveraging a sparse volumetric framework (4.4.1).

- Quantized the resultant hole shape data into 36 sectors at by time or measured depth.
- Integrated the hole shape processor functionality into the Drill Bit Shape Processor (DBSP) DLL.

2. Evaluated the DBMM Hole Shape Processor performance against a representative sensor suite.

4.4.1 Algorithm Summary

Deployment of the DBMM Hole Shape Processor as a C++ DLL was completed. The algorithm leverages an open-source C++ software package called Open-VDB [104] that was designed to enable fast and efficient storage and manipulation of sparse volumetric data discretized into three-dimensional space. A sparse 3D point space is created that is used to model the rock that is being drilled.

A customizable 3D model of the drill bit shape is instantiated in the virtual 3D grid space, and its aggregate motion is evaluated to capture the resultant hole shape that is produced from its motion. The points in the 3D grid which the bit overlaps are eliminated, and as the bit spins and moves the outer most excursions of the bit are used to build a history of the drilled borehole shape. The boundary of remaining (non-overlapping) points are rasterized into a solid surface representing the borehole wall that is then is then subsampled into 36 sectors and reported out as quantized hole shape data once a second .

Figure 4.23 shows an internal view from the 3D volumetric drilling space in Open-VDB. The red points represent the resulting drilled borehole that is generated from essentially extruding the aggregate motion of the drill bit shape through this space and level setting or eliminating any points that it intersects while drilling. As the bit moves and rotates, the algorithm expands the size of the 3D space with a density of points in close vicinity of the bit, and proportional to the simulated flow rate. This process allowed for an efficient number of points to be used to represent and store the drilled borehole shape, based solely on high frequency downhole data.

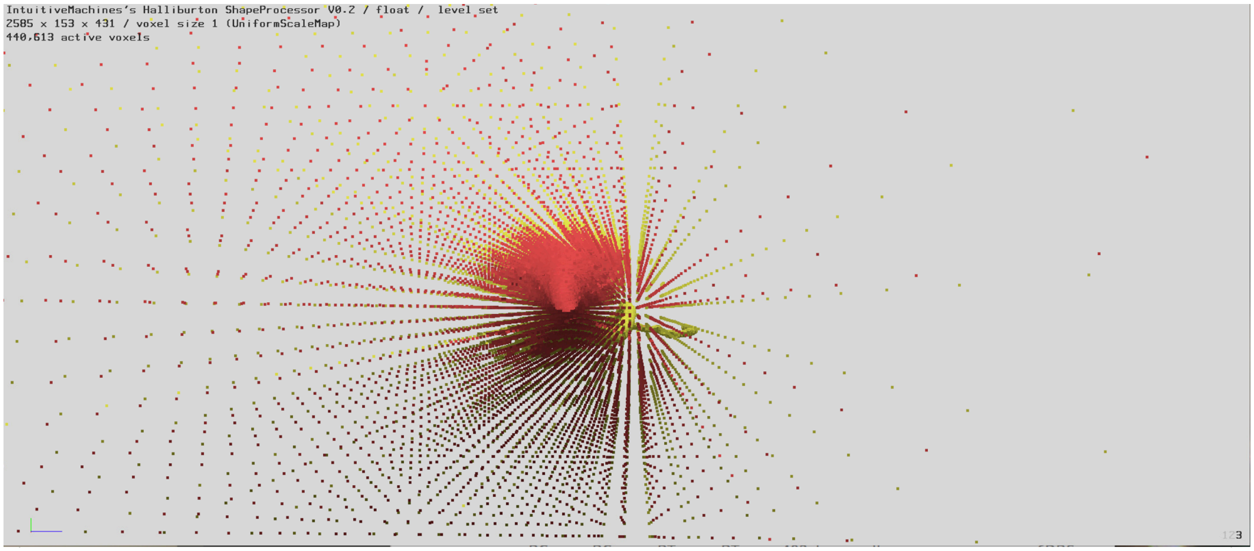

IntuitiveMachines's Halliburton ShapeProcessor V0.2 / float / level_set
2585 x 153 x 431 / voxel size 1 (UniformScaleMap)
440,613 active voxels

Figure 4.23: Level set points within the 3D volumetric drilling space modeled in Open-VDB.

Each point is the center of a unit 'voxel' of space in the 3D grid space and represents the lowest hierarchical level of a grid. Each voxel for this application is being used to store a binary state representing a state of either drilled borehole or solid rock. However, a voxel can store several other sparse attributes in space. For instance, rock density, magnetic or gamma properties of the rock, forces, etc.

DBMM Hole Shape Processor Inputs

The DBMM Hole Shape Processor data inputs are passed into the shape processor as a *std* :: *queue* of the DBSPInput class. Each sample of recorded data (inclination, azimuth, toolface, ROP, and time) are stored in units of radians, feet per hour, and seconds respectively. Then these individual samples are stacked in a queue for all samples over the interval of interest.

```
class DBSPInput
{
  double time;                      ///< Time of the measurement set (seconds)
  double rop;                       ///< Rate of penetration (ft/hr)
  DBMMFilterOutput margOutput;      ///< Marg filter output expressed in az/inc/toolface angles
};

class DBMMFilterOutput
{
    double azimuth;           ///< Azimuth angle (radians)
    double inclination;       ///< Inclination angle (radians)
    double toolface;          ///< Toolface angle (radians)
};
```

Typical Usage

Once the input data queue is specified, the user call run the shape_processor() function to run through the entire list of data in the queue. The DBMM Hole Shape Processor will return a std::vector of DBSPOutput results at one second time slices. On a standard PC laptop, the algorithm can process 100 samples per second of processing time. If the MARG inputs are provided at ≈ 1000 Hz sample rate, the algorithm runs about 10-20 times slower than in real-time. The aggregate bit motion over this interval will be used to determine an average borehole shape which is sampled at borehole cross-sections once every second. The primary result in the output struct is the quantized hole shape data sampled as 36 sectors of a cross-section every time tag. Data and statistics for the average measured depth and borehole radius are supplementary information that is also provided.

```
class DBSPOutput
{
  double MD;          ///< Current measured depth (ft)
  double MD_Average;      ///< Average block measured depth (ft)
  double radius;        ///< Instantaneous radius
  double radius_average;  ///< Radius moving average
  double radius_std;      ///< Radius moving standard deviation
  double time;              ///< Measurement epoch time (s)
  std :: array<double, 36> SPSegments; //< Quantized hole shape segments
};
```

Error Messages

Additional optional inputs for handling messages can be reported by the filter. The user can create a std::queue mapped to a std::string for error messages and log messages to capture insight into issues and status being reported. If a message handler queue is not created, the filter will continue to process the input measurement data (if possible) without storing its corresponding error or status messages.

```
    DBSP::DBSPMain myDBSPMain;

     test_error_ptr   = std :: make_shared<std::queue<std :: string >>();
    myDBSPMain.SetErrorMessages(test_error_ptr);

     test_log_ptr   = std :: make_shared<std::queue<std :: string >>();
    myDBSPMain.SetLogMessages(test_log_ptr);
```

Error Reporting

If an error is encountered, a corresponding error code values in Table 4.4 is returned:

Error Code	Error Description
-100	Input data pointer not set
-101	Output data pointer not set
-102	Input data queue contains no data

Table 4.4: Error reporting codes

Configurable Options

The configurable settings for the DBMM Hole Shape Processor can be assigned using an input pointer to the DBSPSettings class. Descriptions for the individual configuration parameters and their default values are given below:

```
class DBSPSettings
{
  float L1 = 2.0f;     ///< Length of sensor module : Default 2.0 in
  float L2 = 5.00f;    ///< Length of bit cutter from the end sensor sub : Default 5.0 in
  float R1 = 1.00f;      ///< Radius of sensor module, Default 1.0 in
  float R2 = 4.25f;    ///< Radius of cutter profile , Default 4.25 in

  int inputs_moving_average = 1;   ///< Moving average window size (# of samples). Default=1 (off)
};
```

The bit shape can be configured and is defined by the L_1, L_2, R_1, and R_2 parameters. These specify the lengths and radii of the sub and drill bit cutter. By default, the drill bit radius is set to 4.25 inches. The inputs_moving_average parameter is provided to allow the user to configure the size of a moving average window that helps to filter out high-frequency noise. A window size averaging over 20 samples seemed to provide improved performance while evaluating the shape processor performance against available DBS data. If the input data is filtered prior to running the shape processor, setting the window size to one will effectively disable the data smoothing capability.

It is also possible to increase the grid density in Open-VDB effectively improving the resolution of information that is represented in each voxel. This increase in resolution will have a polylogarithmic impact on performance and overall memory requirements of $\approx O((\log n)^3)$. Refinement of the unit voxel size for performance may be required once DBS and associated ROP log data from the field is available for evaluation.

4.5 Wellbore Shape Processor Testing

In order to characterize the hole shape processor performance, it was necessary to generate gyroscope, magnetometer, and accelerometer data under known conditions in the lab; these were collected from an iPhone X at a sampling frequency of $100Hz$. The raw sensor data was run through the MARG DBMM complementary filter to produce estimates for the inclination, azimuth, and toolface history of the bit, and this output was then coupled with representative ROP rates expected while drilling [185] [175].

The iPhone was moved and oriented to emulate drilling through a horizontal, slightly tortuous bore. The ROP data for this case was created to represent expected rates while drilling through rock in order to calibrate the size of a VDB voxel that was necessary to smoothly and efficiently capture the boundary of the borehole shape. Figure 4.24 shows the level set view from outside the drilled borehole path. The outermost drilled points can be seen in red and the borehole surface boundary is shown in yellow. All points in the grid have a yellow 'solid rock' state at the beginning of the simulation. As the bit shape is advanced through the space any points it intersects are given a red drilled state. The majority of the drilled red points in the interior of the borehole are not shown in this figure; they are filtered to allow for the outermost points to be isolated and used for the calculation of statistics on the running borehole center, average radius, and borehole wall shape.

Rasterized drilled borehole using a sparse volumetric environment.

Figure 4.25 shows the interval view of the drilled borehole after the points are rasterized into a solid surface. This is the boundary between the outermost red points and the nearest yellow points meshes into a surface model. The tool provides the capability to export this surface as OBJ or other solid part file format that can be used in analysis and simulation beyond the scope of this shape characterization effort.

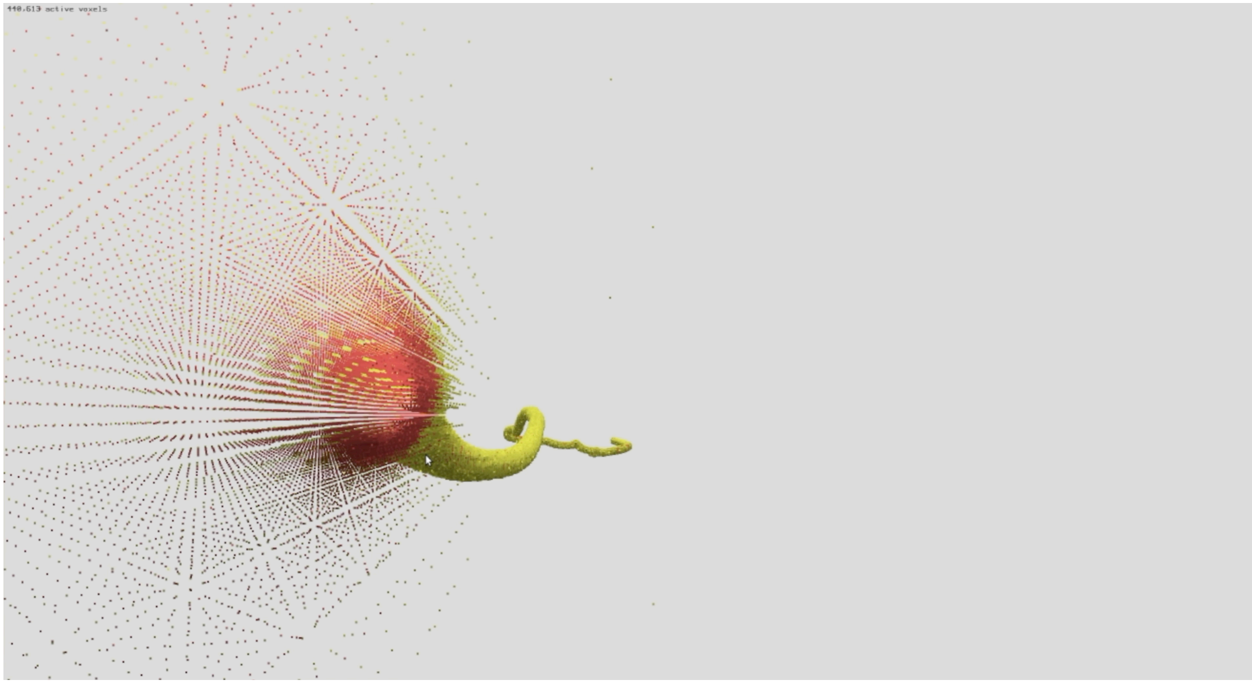


Figure 4.24: Level set view from outside the drilled borehole path view from below the surface.

Rasterized drilled borehole subsurface view

Figure 4.26 shows the variation in the shape data output as 36 sector samples at various measured depths. The center of the borehole in this view is defined by the running window over the prior two sections of the drilled borehole. This provides a reference in which deviations from the center can be captured which provides a way to define a borehole center beyond simply tracking the centroid of the wall shape at every cross-section. If the borehole has a brief section in which one side is more deeply eroded, this will be shown as a larger deviation from the center rather than offsetting the center for that sample cross-section.

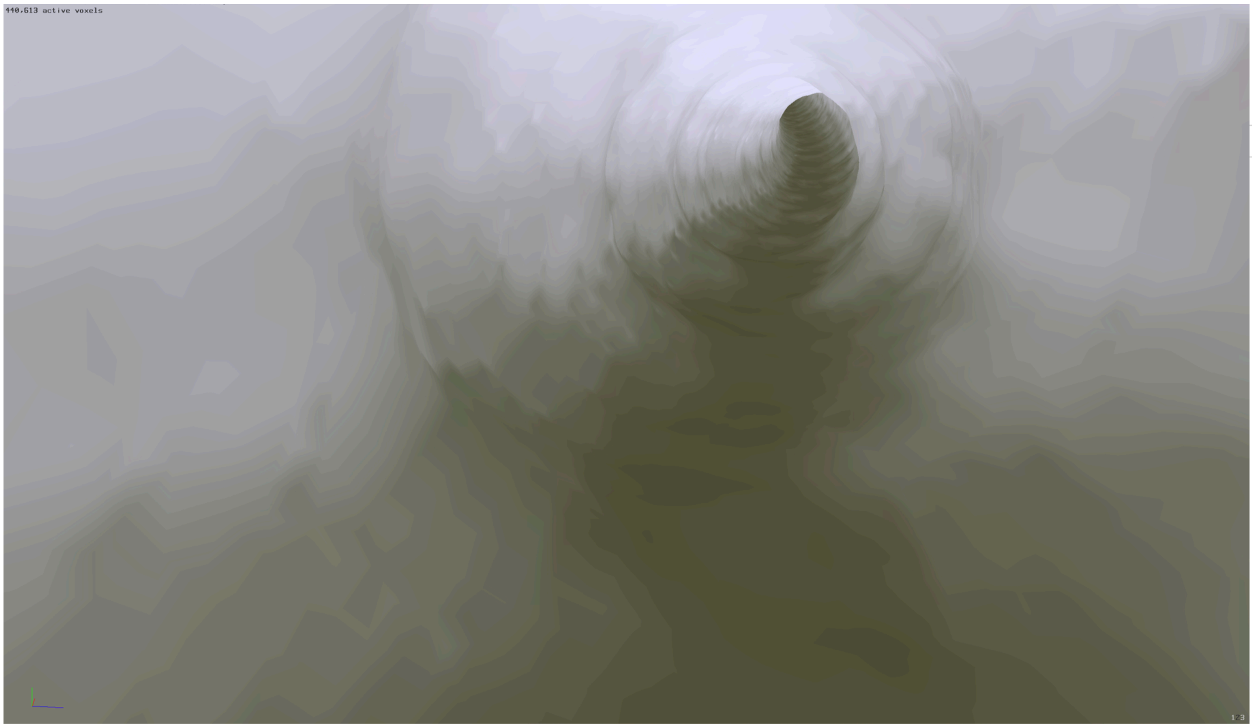


Figure 4.25: Rasterized view of the solid surface representing the interior of the drilled borehole path. This is the boundary between the outermost red points and the nearest yellow points meshes into a surface model. The developed software provides the capability to export this volume for simulation beyond the scope of this effort.

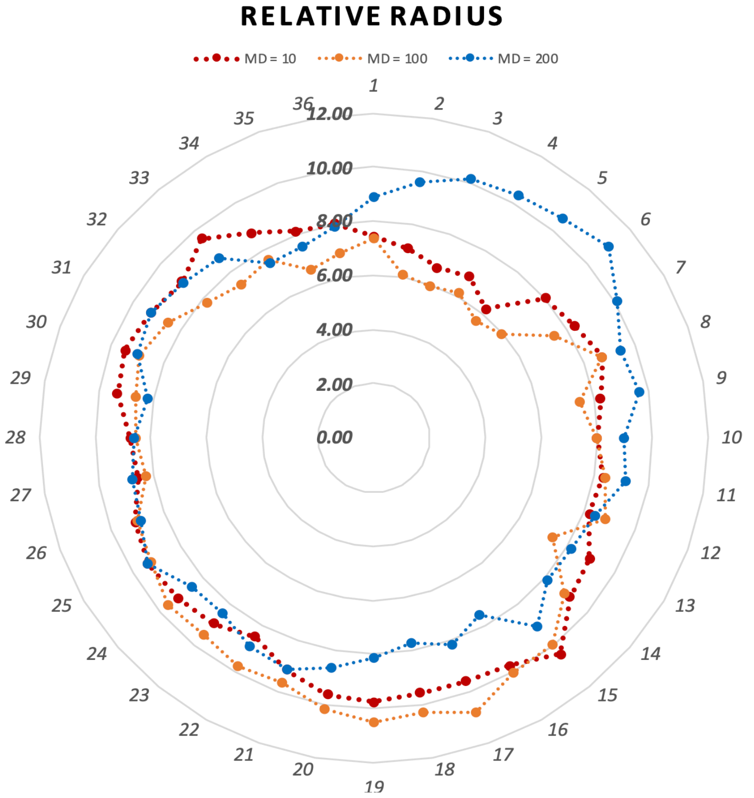


Figure 4.26: Quantized hole shape data for borehole cross-sections at varied stands volumetrically calculated from logged high frequency downhole data near the bit. This figure shows the variation in the shape data output as 36 sector samples at various measured depths. The center of the borehole in this view is defined by the running window over the prior two sections of the drilled borehole. This provides a reference in which deviations from the center can be captured which provides a way to define a borehole center beyond simply tracking the centroid of the wall shape at every cross-section.

5. CONCLUSIONS

First, the first objective of this study was to investigate automated drilling control algorithms using data-driven and physics-based methodologies to improve the performance of manually controlled drilling rigs while advising, taking control, or removing the driller's input in common oil drilling operations entirely. This study employed control theory, data engineering, and machine learning to assist in data exploration and evaluation. Further emphasis was spent on the development of control algorithms, numerical simulations, and validation after the initial phase of building the miniaturized rig was completed. Drilling tests were performed in a lab-scale drilling rig to validate the theoretically simulated performance. Moreover, a fully automated miniaturized drilling machine was then enhanced with the addition of several systems to improve performance and study its dynamic behavior while drilling to design, develop, and apply control algorithms in order to mitigate dysfunctions, optimize the rate of penetration (ROP), and perform estimations in real-time.

An automated rock formation identification model using a data-driven modeling approach was investigated using real-time drilling data obtained from these tests to train and validate a machine learning classification model to further improve the efficiency of the drilling process when combined with a commonly employed technique based on reducing Mechanical Specific Energy (MSE). Results demonstrate the feasibility of a layer-based determination and change detection of properties of rock formation currently being drilled as accurately and near real-time as possible. The experiment was conducted by drilling through an arrangement of different rock formations of varying rock strength properties. Data was recorded and stored at a frequency of 2 kHz, which was then filtered, processed, and downsampled to extract relevant features. An Artificial Neural Network and other machine learning classifiers were used to classify the rock formation currently being drilled. This could lead to savings for energy companies as optimal drilling parameters for a specific type of rock and overburden can be computed and used to improve its overall drilling efficiency. The hypothesis tested that accurately identifying rock formation type in real-time while drilling is feasible with lesser computational cost and complexity. This research may provide the building blocks for the development of a completely autonomous downhole device that reduces the need for highly sophisticated sensors or data transmission processes downhole.

Secondly, simulation of drilling processes involves the seamless integration of strongly coupled physics, numeric, and economic models. Operators desire advanced drilling simulation software that can mimic reality and yet be practical for daily operations. At the same time, academic researchers devote time and effort to develop robust and accurate models of every module of an integrated drilling environment. Still, they are challenged with translating their original contributions into the operational setting.

This research contributed to solving some of those issues through the development of several drilling simulation modules based on the open-source paradigm. The prospective modules developed and to publish include drill-string dynamics, torque and drag, geologic erosion modeling, wellbore hydraulics, and pressure modeling. The original simulator was written in C++ and built using high-performance libraries that leverage parallel computing, GPU acceleration, multi-body, computational fluid dynamics solvers, and other tools. The objective is for interested parties to further develop the framework into proprietary and protected business software via custom functionality. The simulator core was initially developed to model horizontal drilling while providing near real-time performance. The framework was applied in several applications discussed earlier in Chapter 2 and Chapter 4, including modeling shallow horizontal jetting, rock drilling, and steering, as well as reconstructing spatial data from high fidelity downhole signals for borehole evaluation. Additionally, components were used in the development of downhole navigation algorithms.

A more open collaborative development effort between academia and the energy industry will undoubtedly bring solutions to complex modeling needs faster and more economically than any individual endeavor. Subsequently, a complementary MARGS-Based filter for downhole orientation estimation was investigated and developed using numerical modeling simulation methods. In addition, components of the prospective drilling simulator previously discussed were used to validate, visualize, and benchmark the performance of the dynamic models and assumptions using high-frequency data logged while drilling a traditional oil horizontal well.

Thirdly, the initial development of trained machine learning algorithms and extending a real-time advisory system was informed in this thesis. The objective is that this work will aid in supporting drilling operators to perform more effective decisions and optimize the rate of penetration (ROP) while reducing drilling dysfunctions. Initially, machine learning techniques were analyzed using open and proprietary recorded well logs to identify, derive and train the most common supervised machine learning algorithms in order to identify ongoing or incipient vibration and loading patterns that can damage drill bits and slow the drilling process. We then analyzed the feasibility and implementation of these trained models into a smarter downhole tool with a short-hub electromagnetic communication system for geothermal and oil drilling operations.

Lastly, the main take aways and contributions of this work were:

- This study applied control theory and data analysis to assist in data pre-analysis, optimization, and evaluation of drilling operation. Further emphasis has been placed on the development and integration of complex algorithms for drilling simulation.

- Drilling efficiency and performance were optimized by identifying drilling dysfunctions and formation type in real-time.

- The performance of a lab-scale drilling machine was improved by automating step-steps and combining downhole instrumentation.

- Demonstrated the combination of multidisciplinary modeling procedures and computational tools meant for other industries, such as the use of movie animation studios to model borehole shapes from dynamic data.

- Proved that accurately identifying rock formation type in real-time while drilling is feasible with lesser computational cost and complexity.

- Provided a solution architecture for the integration of rarely coupled numerical drilling dynamic models interacting with each other in real-time, e.g., an erosion model with volumetric calculations mimicking borehole propagation.

- Tested and validated methodology on lab-scale drilling tests to train machine learning models to classify drilling events and dysfunctions.
- This research work contributed with a Magnetic, Angular Rate, and Gravity (MARG) filter for Downhole Orientation Estimation.
- Proved that high-frequency dynamics measurements combined with an orientation estimation filter and parallel computing tools can compute an approximate representation of the borehole without having to run a caliper log.
- The publication of the uncoupled written modules in C++ of drill-string dynamics, torque and drag, geologic, wellbore pressure, and hydraulics modeling.

5.1 Future Work Discussion

Drilling Simulation and Automation will advance much faster over the next decade. More capable surface and downhole devices will open the door for "hybrid systems" that use real-time mechanistic or numerical models, on-site measurements, and historical field data to increase the reliability and accuracy of model predictions. We highlight some of the open issues to be developed further:

- The main source of drilling logs was acquired using the typical WITSML standard for logging in the Oil & Gas industry. The main steps of the pipeline are dataset loading, preprocessing, feature extraction, parameter selection, model configuration, and creation. The open-source programming languages Python, R, and C++ were chosen based on available expertise and the availability of tools. An early prototype of the ML pipeline was completed and has been utilized to apply more complex feature engineering and models. However, most feature engineering, model validation, packaging, and deployment are being done manually. Even though an initial prototype is was developed internally, more effort is needed on reducing the necessary time to use WITSML and data-driven models in an academic setting.

- Results suggest that predicting drilling dysfunctions in real-time is feasible. In-situ testing of the bit dysfunction library to validate the trained data-driven models while drilling, however, was not covered in the scope of this project and will be a subject for the future.

- More extended testing and validation of the modularized drilling models is needed to decouple from the hardware in the loop and visualization components.

- Refinement and modularization of written drilling models into an easier programming language such as Python with proper documentation for the future user is needed to ease its use and application. Including the creation of established "testing" cases to benchmark, and verify the computation results of drilling models, something that an interested industry group is actively working on.

- The bit state estimator and drilling models can be integrated. A partial validation of the processing of simulated magnetometer, gyroscope, and accelerometer measurements with the existing error and formation interaction statistical models can be performed to further increase the accuracy of the filter describer in the previous sections. Once the integration is complete, strain gauge calculations can also be added and complexity added to formation interaction and sensor error models. This step-wise introduction of complexity will assure that the most observable measurements are successfully integrated in simulation first and increasing fidelity added as soon as acceptable estimation results are validated by a simulation.

- A deeper investigation and use of Bayesian modeling to improve the prediction of dysfunction identification models that include surface measurements should be done to not only detect the onset of the dysfunction but potentially predict it from happening.

- The use of the non-linear dynamic simulator to generate artificial physics-based data for artificial intelligence and reinforcement learning techniques.

- Apply sparse sensing and model-order reduction techniques to estimate minimum downhole sensing requirements.

REFERENCES

[1] X. Liu, N. Vlajic, X. Long, G. Meng, and B. Balachandran, "Nonlinear motions of a flexible rotor with a drill bit: stick-slip and delay effects," *Nonlinear Dynamics*, vol. 72, no. 1, pp. 61–77, 2013.

[2] E. Zarate-Losoya, T. Cunningham, I. El-Sayed, S. Noynaert, and F. Florence, "Lab-scale drilling rig autonomously mitigates downhole dysfunctions and geohazards through bit design, control system and machine learning," *IADC/SPE Drilling Conference and Exhibition*, 2018.

[3] E. Z. Losoya, E. Gildin, and S. F. Noynaert, "Real-time rate of penetration optimization of an autonomous lab-scale rig using a scheduled-gain pid controller and mechanical specific energy," *IFAC-PapersOnLine*, vol. 51, no. 8, pp. 56–61, 2018.

[4] schlumberger Well Testing Services, "T4-ts-0014: Petroleum geochemistry, understanding petroleum origin and processes controlling fluid behavior.." `https://www.slb.com/-/media/files/testing-services/product-sheet/fluid-lab-services-petroleum-geochemistry-ps.ashx`, 2014.

[5] P. prljan, D. Pavkovic, M. Cipek, T. Staroveski, M. Klaic, and D. Kolar, "Automation systems design and laboratory prototyping aimed at mature petroleum drilling rig retrofitting," *Tehnicki Vjesnik*, vol. 27, pp. 229–236, 02 2020.

[6] R. Mitchell, S. Miska, *et al.*, *Fundamentals of drilling engineering*. Society of Petroleum Engineers, 2011.

[7] S. Rassenfoss, "Exxonmobil testing the next step: Total drilling automation on land," Oct 2021.

[8] *Drilling Systems Automation - Preparing for the Big Jump Forward*, vol. All Days of *SPE/IADC Drilling Conference and Exhibition*, 03 2013. SPE-163422-MS.

[9] NOV, "Top drive solutions." Online, 2015.

[10] A. Kyllingstad and P. J. Nessjøen, "A new stick-slip prevention system," 2009.

[11] P. NESSJOEN and Å. KYLLINGSTAD, "Method and apparatus for reducing stick-slip," Jan. 23 2013. EP Patent App. EP20,120,188,975.

[12] A. Kyllingstad and P. J. Nessjoen, "Hardware-in-the-loop simulations used as a cost-efficient tool for developing an advanced stick-slip prevention system," *IADC / SPE Drilling Conference and Exhibition*, 2010.

[13] P. Watt, "Offshore drilling techniques," *Inst. Gas Eng., Commun.;(United Kingdom)*, vol. 985, 1976.

[14] B. Rehm, J. Schubert, A. Haghshenas, A. S. Paknejad, and J. Hughes, *Managed pressure drilling*. Elsevier, 2013.

[15] *Making the Impossible Possible with MPD Application on 0.4 ppg Mud Weight Window A Case History Successful Automated Managed Pressure Drilling, Managed Pressure Run (Tieback Liner), and Managed Pressure Cementing in a Semisubmersible Offshore Sarawak Malaysia*, vol. Day 1 Mon, September 28, 2015 of *SPE Annual Technical Conference and Exhibition*, 09 2015. D011S005R005.

[16] M. Mæland, "Managed pressure drilling: The solaris prospect-hpht exploration well," *Thesis*, 2013.

[17] M. K. Bak, "Model based design of electro-hydraulic motion control systems for offshore pipe handling equipment," 2014.

[18] C. Carpenter *et al.*, "Real-time analysis for remote operations centers," *Journal of Petroleum Technology*, vol. 65, no. 09, pp. 160–163, 2013.

[19] C. Bossi *et al.*, "Succeeding in the shale business with a lean well manufacturing management system," *Journal of Petroleum Technology*, vol. 69, no. 03, pp. 42–44, 2017.

[20] G. LV, "Technology outlook 2025: Energy." http://to2025.dnvgl.com/energy/, April 2016.

[21] BP, "Bp technology outlook." Article, 2015.

[22] J. Dunlop, R. Isangulov, W. D. Aldred, H. A. Sanchez, J. L. S. Flores, J. A. Herdoiza, J. Belaskie, and C. Luppensenti, "Increased rate of penetration through automation," 2011.

[23] W. R. Nelson, "Improving safety of deepwater drilling through advanced instrumentation, diagnostics, and automation for BOP control systems," 2016.

[24] J. P. de Wardt, M. Behounek, C. Chapman, and D. Putra, "Drilling Systems Automation - Preparing for the Big Jump Forward," 2013.

[25] R. Bromell, "Automation in drilling," *Journal of Petroleum Technology*, 1967.

[26] J. D'Angelo, M. S. Khaled, P. Ashok, and E. van Oort, "Pareto optimal directional drilling advisory for improved real-time decision making," *Journal of Petroleum Science and Engineering*, vol. 210, 2022.

[27] S. SPE, "OnePetro.org," Oct. 2021.

[28] E. Zarate-Losoya, "Interactive dashboard of SPE publications about drilling automation by discipline." Available: "`https://public.tableau.com/app/profile/enrique.losoya/viz/DrillngAutomation/Dashboard1`", Mar. 2021.

[29] J. D. Macpherson, J. P. de Wardt, F. Florence, C. Chapman, M. Zamora, M. Laing, and F. Iversen, *Drilling-Systems Automation: Current State, Initiatives, and Potential Impact*, vol. 28. Society of Petroleum Engineers (SPE), dec 2013.

[30] M. Behounek, "Operators group on drilling data quality (ogddq)," *Sensors to Solutions SPE Workshop*, 1-3 September. 2015.

[31] N. Zenero, S. Koneti, and W. Schnieder, "Iron roughneck make up torque–its not what you think!," 2016.

[32] K. Ogata, *Modern Control Engineering*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 4th ed., 2001.

[33] P. Pournazari, P. Ashok, E. van Oort, S. Unrau, and S. Lai, "Enhanced kick detection with low-cost rig sensors through automated pattern recognition and real-time sensor calibration," 2015.

[34] H. P. Lohne, J. E. Gravdal, E. W. Dvergsnes, G. H. Nygaard, and E. H. Vefring, "Automatic calibration of real-time computer models in intelligent drilling control systems - results from a north sea field trial," 2008.

[35] E. Cayeux, B. Daireaux, E. W. Dvergsnes, and F. Florence, "Toward drilling automation: On the necessity of using sensors that relate to physical models," *SPE Drilling & Completion*, vol. 29, pp. 236–255, jun 2014.

[36] SPE, "Spe drilling systems automation technical section site," 2016.

[37] Ø. Andersen, "Using RealTime data for well design optimization, not just drilling optimization," 2016.

[38] W. G. Lesso, M. Ignova, F. Zeineddine, J. M. Burks, and J. B. Welch, "Testing the combination of high frequency surface and downhole drilling mechanics and dynamics data under a variety of drilling conditions.," 2011.

[39] C. Klotz, P. R. Bond, I. Wassermann, and S. Priegnitz, "A new mud pulse telemetry system for enhanced MWD/LWD applications," vol. l, 2008.

[40] J. R. Bailey, C. C. Elsborg, R. W. James, P. E. Pastusek, M. T. Prim, and W. W. Watson, "Design evolution of drilling tools to mitigate vibrations," 2013.

[41] J. Fay, H. Fay, and A. Couturier, "Wired pipes for a high-data-rate MWD system," 1992.

[42] C. McCartney, S. Allen, M. Hernandez, D. MacFarlane, A. Baksh, and M. E. Reeves, "Step-change improvements with wired-pipe telemetry," 2009.

[43] A. Wilson, "Wired-drillpipe field trials reveal potential benefits over traditional pipe," *Journal of Petroleum Technology*, vol. 65, pp. 105–107, jun 2013.

[44] M. J. Manning, J. D. Macpherson, D. E. Taylor, N. Baksh, C. Peveto, L. Farnsworth, and S. R. Lemke, “Wired pipe-enabled logging while drilling applications,” vol. d, 2008.

[45] A. K. Vajargah, G. Sullivan, and E. van Oort, “Automated fluid rheology and ECD management,” 2016.

[46] S. T. Edwards, C. J. Coley, N. A. Whitley, R. G. Keck, V. Ramnath, T. Foster, K. Coghill, and M. Honey, “A summary of wired drill pipe (IntelliServ networked drillstring) field trials and deployment in bp,” 2013.

[47] R. A. Shishavan, C. Hubbell, H. D. Perez, J. D. Hedengren, D. S. Pixton, and A. P. Pink, “Multivariate control for managed-pressure-drilling systems by use of high-speed telemetry,” *SPE Journal*, vol. 21, pp. 459–470, apr 2016.

[48] M. R. ENDSLEY, “Level of automation effects on performance, situation awareness and workload in a dynamic control task,” *Ergonomics*, vol. 42, pp. 462–492, mar 1999.

[49] A. Vera, “Integration of human and machine intelligence: toward performance in non-deterministic environments.” Presentation at the SPE DSATS/IADC ART Symposium: Human Integration with Automation and Machine Intelligence, 29 February 2016.

[50] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, “A model for types and levels of human interaction with automation,” *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, vol. 30, no. 3, pp. 286–297, 2000.

[51] A. Bowler, R. Harmer, L. Logesparan, J. Sugiura, B. Jeffryes, and M. Ignova, “Continuous high-frequency measurements of the drilling process provide new insights into drilling-system response and transitions between vibration modes,” *SPE Drilling & Completion*, vol. 31, pp. 106–118, may 2016.

[52] T. Feng, M. Vadali, and D. Chen, “Modeling and analysis of directional drilling dynamics,” vol. 58295, p. V003T43A006, 2017.

[53] R. I. Leine, D. H. van Campen, and W. J. G. Keultjes, “Stick-slip whirl interaction in drill-string dynamics,” *Journal of Vibration and Acoustics*, vol. 124, no. 2, pp. 209–220, 2002.

[54] J. Bailey, G. Payette, M. Prim, J. Molster, A. Al Mheiri, P. McCormack, and K. LeRoy, “Mitigating drilling vibrations in a lateral section using a real-time advisory system,” in *Abu Dhabi International Petroleum Exhibition & Conference*, OnePetro, 2017.

[55] A. Wilson, “Drilling modeling and simulation: Current state and future goals,” *Journal of Petroleum Technology*, vol. 67, no. 09, pp. 140–142, 2015.

[56] S. F. Noynaert, E. Gildin, *et al.*, “Going beyond the tally book: A novel method for analysis, prediction and control of directionally drilled wellbores using mechanical specific energy,” in *SPE Annual Technical Conference and Exhibition*, Society of Petroleum Engineers, 2014.

[57] V. A. Bavadiya, M. J. Aljubran, J. M. Kibe, S. M. Christy, H. N. Le, R. Ahmed, and F. Florence, “Design, construction and operation of an automated drilling rig for the DSATS university competition,” 2015.

[58] F. E. Dupriest, W. C. Elks, S. Ottesen, P. E. Pastusek, J. R. Zook, and C. R. Aphale, “Borehole-quality design and practices to maximize drill-rate performance,” *SPE Drilling & Completion*, vol. 26, pp. 303–316, jun 2011.

[59] F. E. Dupriest and W. L. Koederitz, “Maximizing drill rates with real-time surveillance of mechanical specific energy,” 2005.

[60] L. Liung, *System identification-Theory for the user 2nd ed.* 1999.

[61] U. Sellgren, *Simulation-driven design : Motives, Means, and Opportunities*. Doctoral thesis, comprehensive summary, 1999.

[62] N. R. Council, *Statistics, Testing, and Defense Acquisition: New approaches and methodological improvements*. National Academies Press, 1998.

[63] G. Downton, “Challenges of modeling drilling systems for the purposes of automation and control,” *IFAC Proceedings Volumes*, vol. 45, no. 8, pp. 201–210, 2012.

[64] L. Perneder, E. Detournay, and G. Downton, “Bit/rock interface laws in directional drilling,” *International Journal of Rock Mechanics and Mining Sciences*, vol. 51, pp. 81–90, 2012.

[65] E. Z. Losoya, E. Gildin, S. F. Noynaert, Z. Medina-Zetina, T. Crain, S. Stewart, J. Hicks, *et al.*, “An open-source enabled drilling simulation consortium for academic and commercial applications,” in *SPE Latin American and Caribbean Petroleum Engineering Conference*, Society of Petroleum Engineers, 2020.

[66] P. Pastusek, G. Payette, R. Shor, E. Cayeux, U. J. Aarsnes, J. Hedengren, S. Menand, J. Macpherson, R. Gandikota, M. Behounek, *et al.*, “Creating open source models, test cases, and data for oilfield drilling challenges,” in *SPE/IADC International Drilling Conference and Exhibition*, Society of Petroleum Engineers, 2019.

[67] V. Tikhonov, K. Valiullin, A. Nurgaleev, L. Ring, R. Gandikota, P. Chaguine, and C. Cheatham, “Dynamic model for stiff-string torque and drag (see associated supplementary discussion),” *SPE Drilling & Completion*, vol. 29, no. 03, pp. 279–294, 2014.

[68] N. Vishnumolakala, E. Gildin, and S. Noynaert, “A simulation environment for automatic managed pressure drilling control,” *IFAC-PapersOnLine*, vol. 48, no. 6, pp. 128–133, 2015.

[69] S. I. Ødegård, B. T. Risvik, K. S. Bjørkevoll, Ø. Mehus, R. Rommetveit, and M. Svendssen, “Advanced dynamic training simulator for drilling as well as related experience from training of drilling teams with focus on realistic downhole feedback,” in *SPE/IADC Drilling Conference*, OnePetro, 2013.

[70] J. Petersen, R. Rommetveit, K. S. Bjorkevoll, and J. Froyen, “A general dynamic model for single and multi-phase flow operations during drilling, completion, well control and intervention,” in *IADC/SPE Asia Pacific Drilling Technology Conference and Exhibition*, OnePetro, 2008.

[71] V. Tikhonov, O. Bukashkina, V. Liapidevskii, and L. Ring, “Development of model and software for simulation of surge-swab process at drilling,” in *SPE Russian Petroleum Technology Conference and Exhibition*, OnePetro, 2016.

[72] J. Booth, I. D. R. Bradford, J. M. Cook, J. D. Dowell, G. Ritchie, and I. Tuddenham, “Meeting future drilling planning and decision support requirements: A new drilling simulator,”

p. 13, 2001/1/1/ 2001.

[73] F. Iversen, L. J. Gressgård, J. Thorogood, M. K. Balov, and V. Hepso, "Drilling automation: Potential for human error," *SPE Drilling & Completion*, vol. 28, no. 01, pp. 45–59, 2013.

[74] F. Florence, C. Chapman, J. Macpherson, and M. Cavanaugh, "Implementation of drilling systems automation-halifax workshop summary: Industry standards, business models and next steps," in *SPE Annual Technical Conference and Exhibition*, Society of Petroleum Engineers.

[75] A. Maria, "Introduction to modeling and simulation," in *Proceedings of the 29th conference on Winter simulation*, pp. 7–13, 1997.

[76] K. T. Gon, B. P. Zeigler, and H. Prashofer, "Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems," 2000.

[77] B. P. Zeigler, A. Muzy, and E. Kofman, *Theory of modeling and simulation: discrete event & iterative system computational foundations*. Academic press, 2018.

[78] B. Zeigler, H. Prähofer, and T. G. Kim, "Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems," vol. 2, 01 2000.

[79] A. Kueck, B. Recke, and G. Ostermeyer, "Reduction of nonlinear drill string models using proper orthogonal decomposition," *PAMM*, vol. 14, no. 1, pp. 205–206, 2014.

[80] T. Feng, M. Vadali, Z. Ma, D. Chen, and J. Dykstra, "A finite element method with full bit-force modeling to analyze drillstring vibration," *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 9, p. 091016, 2017.

[81] A. Busch, A. Islam, D. W. Martins, F. P. Iversen, M. Khatibi, S. T. Johansen, R. W. Time, and E. A. Meese, "Cuttings-transport modelingpart 1: Specification of benchmark parameters with a norwegian-continental-shelf perspective," *SPE Drilling & Completion*, vol. 33, no. 02, pp. 130–148, 2018.

[82] M. B. S. Márquez, I. Boussaada, H. Mounier, and S.-I. Niculescu, “An overview of drill-string models,” *Analysis and Control of Oilwell Drilling Vibrations*, pp. 9–23, 2015.

[83] H. Sarak, M. Onur, and A. Satman, “Lumped-parameter models for low-temperature geothermal fields and their application,” *Geothermics*, vol. 34, no. 6, pp. 728–755, 2005.

[84] D. Matko, G. Geiger, and W. Gregoritza, “Pipeline simulation techniques,” *Mathematics and computers in simulation*, vol. 52, no. 3-4, pp. 211–230, 2000.

[85] X. Lin, H. E. Perez, S. Mohan, J. B. Siegel, A. G. Stefanopoulou, Y. Ding, and M. P. Castanier, “A lumped-parameter electro-thermal model for cylindrical batteries,” *Journal of Power Sources*, vol. 257, pp. 1–11, 2014.

[86] M. B. S. Márquez, I. Boussaada, H. Mounier, and S.-I. Niculescu, “Analysis and control of oilwell drilling vibrations,” *A Time-Delay Systems Approach, ser. Advances in Industrial Control. Springer*, 2015.

[87] T. Richard, C. Germay, and E. Detournay, “A simplified model to explore the root cause of stickslip vibrations in drilling systems with drag bits,” *Journal of Sound and Vibration*, vol. 305, no. 3, pp. 432–456, 2007.

[88] N. Challamel, “Rock destruction effect on the stability of a drilling structure,” *Journal of sound and vibration*, vol. 233, no. 2, pp. 235–254, 2000.

[89] E. Detournay and P. Defourny, “A phenomenological model for the drilling action of drag bits,” in *International journal of rock mechanics and mining sciences & geomechanics abstracts*, vol. 29, pp. 13–23, Elsevier, 1992.

[90] J. Bailey and I. Finnie, “An analytical study of drill-string vibration,” 1960.

[91] G. Halsey, A. Kyllingstad, and A. Kylling, “Torque feedback used to cure slip-stick motion,” in *SPE Annual Technical Conference and Exhibition*, OnePetro, 1988.

[92] Y. Khulief and H. Al-Naser, “Finite element dynamic analysis of drillstrings,” *Finite elements in analysis and design*, vol. 41, no. 13, pp. 1270–1288, 2005.

[93] Y.-Q. Lin and Y.-H. Wang, "Stick-slip vibration of drill strings," 1991.

[94] J. F. Brett, "The genesis of bit-induced torsional drillstring vibrations," *SPE Drilling Engineering*, vol. 7, no. 03, pp. 168–174, 1992.

[95] J. D. Jansen and L. Van Den Steen, "Active damping of self-excited torsional vibrations in oil well drillstrings," *Journal of sound and vibration*, vol. 179, no. 4, pp. 647–668, 1995.

[96] X. Liu, N. Vlajic, X. Long, G. Meng, and B. Balachandran, "State-dependent delay influenced drill-string oscillations and stability analysis," *Journal of Vibration and Acoustics*, vol. 136, no. 5, 2014.

[97] X. Zhu, L. Tang, and Q. Yang, "A literature review of approaches for stick-slip vibration suppression in oilwell drillstring," *Advances in Mechanical Engineering*, vol. 6, p. 967952, 2014.

[98] C.-M. Liao, B. Balachandran, M. Karkoub, and Y. L. Abdel-Magid, "Drill-string dynamics: reduced-order models and experimental studies," *Journal of Vibration and Acoustics*, vol. 133, no. 4, 2011.

[99] L. Li, J. Wang, Y. Yu, Y. Xing, F. Zhang, Y. Zhang, and Y. Li, "A review of the research and development of high-frequency measurement technologies used for nonlinear dynamics of drillstring," *Shock and Vibration*, vol. 2021, 2021.

[100] M. Z. Jacobson and M. Z. Jacobson, *Fundamentals of atmospheric modeling*. Cambridge university press, 1999.

[101] L. Ojeda and J. Borenstein, "Personal dead-reckoning system for gps-denied environments," in *2007 IEEE International Workshop on Safety, Security and Rescue Robotics*, pp. 1–6, IEEE, 2007.

[102] D. Negrut, R. Serban, and A. Tasora, "Posing multibody dynamics with friction and contact as a differential complementarity problem," *Journal of Computational and Nonlinear Dynamics*, vol. 13, no. 1, 2018.

[103] H. Mazhar, T. Heyn, A. Pazouki, D. Melanz, A. Seidl, A. Bartholomew, A. Tasora, and D. Negrut, “Chrono: a parallel multi-physics library for rigid-body, flexible-body, and fluid dynamics,” *Mechanical Sciences*, vol. 4, no. 1, pp. 49–64, 2013.

[104] K. Museth, “Vdb: High-resolution sparse volumes with dynamic topology trans. graph,” vol. 32, no. 3, pp. 1–22, 2013.

[105] J. Beyer, M. Hadwiger, and H. Pfister, “State-of-the-art in gpu-based large-scale volume visualization,” in *Computer Graphics Forum*, vol. 34, pp. 13–37, Wiley Online Library, 2015.

[106] J. Nickolls, I. Buck, M. Garland, and K. Skadron, “Scalable parallel programming with cuda,” in *ACM SIGGRAPH 2008 classes*, p. 16, ACM, 2008.

[107] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. Negrut, “Chrono: An open source multi-physics dynamics engine,” pp. 19–49, 2015.

[108] W. Schiehlen, “Multibody system dynamics: roots and perspectives,” *Multibody system dynamics*, vol. 1, no. 2, pp. 149–188, 1997.

[109] M. F. Abughaban, A. W. Eustes, J. P. de Wardt, and M. Willerth, “Improving torque and drag prediction using the advanced spline curves borehole trajectory,” in *SPE/IADC Drilling Conference and Exhibition*, OnePetro, 2017.

[110] V. Tikhonov, O. Bukashkina, V. Liapidevskii, and L. Ring, “Development of model and software for simulation of surge-swab process at drilling,” in *SPE Russian Petroleum Technology Conference and Exhibition*, vol. All Days, (SPE-181933-MS).

[111] A. Nikoofard, U. J. F. Aarsnes, T. A. Johansen, and G. Kaasa, “State and parameter estimation of a drift-flux model for underbalanced drilling operations,” *IEEE Transactions on Control Systems Technology*, vol. 25, no. 6, pp. 2000–2009, 2017.

[112] K. S. Bjørkevoll, B. Daireaux, and P. C. Berg, “Possibilities, limitations and pitfalls in using real-time well flow models during drilling operations,” in *SPE Bergen One Day Seminar*, OnePetro, 2015.

[113] B. S. Aadnoy, I. Cooper, S. Miska, R. F. Mitchell, and M. L. Payne, *Advanced drilling and well technology*. SPE Houston, TX, 2009.

[114] H. I. Bilgesu, N. Mishra, and S. Ameri, “Understanding the effect of drilling parameters on hole cleaning in horizontal and deviated wellbores using computational fluid dynamics,” in *Eastern Regional Meeting*, vol. All Days, (SPE-111208-MS), 2007.

[115] D. Nguyen and S. Rahman, “A three-layer hydraulic program for effective cuttings transport and hole cleaning in highly deviated and horizontal wells,” in *SPE/IADC Asia Pacific Drilling Technology*, Society of Petroleum Engineers.

[116] J. Li and S. Walker, “Sensitivity analysis of hole cleaning parameters in directional wells,” *SPE Journal*, vol. 6, no. 04, pp. 356–363, 2001.

[117] R. B. Adari, S. Miska, E. Kuru, P. Bern, and A. Saasen, “Selecting drilling fluid properties and flow rates for effective hole cleaning in high-angle and horizontal wells,” in *SPE Annual Technical Conference and Exhibition*, OnePetro, 2001.

[118] T. Sifferman and T. Becker, “Hole cleaning in full-scale inclined wellbores,” *SPE Drilling Engineering*, vol. 7, no. 02, pp. 115–120, 1992.

[119] R. A. Sanchez, J. Azar, A. Bassal, and A. Martins, “The effect of drillpipe rotation on hole cleaning during directional well drilling,” in *SPE/IADC drilling conference*, Society of Petroleum Engineers, 1997.

[120] A. Saasen and G. Laklingholm, “The effect of drilling fluid rheological properties on hole cleaning,” in *IADC/SPE Drilling Conference*, Society of Petroleum Engineers, 2002.

[121] S. Akhshik, M. Behzad, and M. Rajabi, “Cfd-dem simulation of the hole cleaning process in a deviated well drilling: the effects of particle shape,” *Particuology*, vol. 25, pp. 72–82, 2016.

[122] S. K. Dewangan and S. L. Sinha, "Exploring the hole cleaning parameters of horizontal wellbore using two-phase eulerian cfd approach," *The Journal of Computational Multiphase Flows*, vol. 8, no. 1, pp. 15–39, 2016.

[123] T. V. Aarrestad, "Torque and drag-two factors in extended-reach drilling," *Journal of Petroleum Technology*, vol. 46, no. 09, pp. 800–803, 1994.

[124] C. Johancsik, D. Friesen, and R. Dawson, "Torque and drag in directional wells-prediction and measurement," *Journal of Petroleum technology*, vol. 36, no. 06, pp. 987–992, 1984.

[125] H. Ho-Cheng, *An analysis of drilling of composite materials*. PhD thesis, University of California, Berkeley, 1988.

[126] M. Payne and F. Abbassian, "Advanced torque and drag considerations in extended-reach wells," in *IADC/SPE drilling conference*, OnePetro, 1996.

[127] R. Clark and K. Bickham, "A mechanistic model for cuttings transport," in *SPE Annual Technical Conference and Exhibition*, Society of Petroleum Engineers, 1994.

[128] T. Feng, M. Vadali, Z. Ma, D. Chen, and J. Dykstra, "A finite element method with full bit-force modeling to analyze drillstring vibration," *Journal of Dynamic Systems, Measurement, and Control*, vol. 139, no. 9, pp. 091016–091016–10, 2017. 10.1115/1.4036083.

[129] R. Samuel, "A new well-path design using clothoid spiral curvature bridging for ultra-extended-reach drilling," *Spe Drilling & Completion*, vol. 25, no. 03, pp. 363–371, 2010.

[130] *New Aspects of Torque-and-Drag Modeling in Extended-Reach Wells*, vol. All Days of *SPE Annual Technical Conference and Exhibition*, 09 2010. SPE-135719-MS.

[131] J. Friedman, T. Hastie, R. Tibshirani, *et al.*, *The elements of statistical learning*, vol. 1. Springer series in statistics New York, 2001.

[132] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[133] K. G. Maver, N. M. Hernandez, F. Baillard, and R. Cooper, “Processing of unstructured geoscience and engineering information for instant access and extraction of new knowledge,” *First Break*, vol. 38, no. 6, pp. 59–64, 2020.

[134] P. Jong and J. Krebbers, “Transforming the subsurface data and application landscape with the open subsurface data universe forum,” in *SEG Technical Program Expanded Abstracts 2019*, pp. 4987–4989, Society of Exploration Geophysicists, 2019.

[135] W. Mckenzie, F. Morandini, P. Verney, L. Deny, J.-F. Rainaud, R. Eastick, G. Masters, D. Mack Endres, *et al.*, “Integrating witsml, prodml, and resqml standards for cross-domain workflows,” in *SPE Intelligent Energy International*, Society of Petroleum Engineers, 2012.

[136] J. Pickering, N. Whiteley, J. Roye, H. Turnbull, and D. R. Cavanagh, “A standard real-time information architecture for drilling and completions,” in *SPE Annual Technical Conference and Exhibition*, OnePetro, 2007.

[137] *WITSML Changing the Face of Real-Time*, vol. All Days of *SPE Intelligent Energy International Conference and Exhibition*, 02 2008. SPE-112016-MS.

[138] E. Gurina, K. Antipova, N. Klyuchnikov, and D. Koroteev, “Machine learning microservice for identification of accident predecessors,” in *SPE Middle East Oil & Gas Show and Conference*, OnePetro, 2021.

[139] S. W. Lai, J. Ng, A. Eddy, S. Khromov, D. Paslawski, R. van Beurden, L. Olesen, G. S. Payette, and B. J. Spivey, “Large-Scale Deployment of a Closed-Loop Drilling Optimization System: Implementation and Field Results,” *SPE Drilling & Completion*, vol. 36, pp. 47–62, 03 2021.

[140] T. A. Olukoga and Y. Feng, “Practical Machine-Learning Applications in Well-Drilling Operations,” *SPE Drilling & Completion*, pp. 1–19, 04 2021.

[141] D. T. Phan, C. Liu, M. J. AlTammar, Y. Han, and Y. N. Abousleiman, “Application of Artificial Intelligence To Predict Time-DependentMud-Weight Windows in Real Time,” *SPE Journal*, pp. 1–21, 10 2021.

[142] T. Hoeink and C. Zambrano, "Shale discrimination with machine learning methods," in *51st US Rock Mechanics/Geomechanics Symposium*, OnePetro, 2017.

[143] I. Simon, V. Koryabkin, A. Semenikhin, and A. Gruzdev, "Determination of lithologic difference at the bottom of wells using cognitive technologies," in *SPE Russian Petroleum Technology Conference*, OnePetro, 2018.

[144] *Case Study: Innovative Application Of Automated Drilling System With Hybrid Coiled Tubing Drilling Rig Excels In Cost-Sensitive Environment*, vol. All Days of *IADC/SPE Asia Pacific Drilling Technology Conference and Exhibition*, 08 2008. SPE-114597-MS.

[145] D. A. Blankenship, J. A. Henfling, A. J. Mansure, S. D. Knudsen, D. Chavira, and R. D. Jacobson, "High-temperature diagnostics-while drilling system.," tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2006.

[146] C. W. De Silva, *Sensors and Actuators: Engineering System Instrumentation*. CRC Press, 2015.

[147] J. P. de Wardt, J. D. Macpherson, M. Zamora, B. Dow, S. Hbaieb, R. A. Macmillan, M. L. Laing, A. M. DiFiore, C. E. Inabinett, and M. W. Anderson, "Drilling systems automation roadmap - the means to accelerate adoption," 2015.

[148] C. D. Chapman, J. L. Sanchez, R. D. L. Perez, and H. Yu, "Automated closed-loop drilling with ROP optimization algorithm significantly reduces drilling time and improves downhole tool reliability," 2012.

[149] F. Florence and F. P. Iversen, "Real-time models for drilling process automation: Equations and applications," 2010.

[150] J. Macpherson, "Technology Focus: Drilling Systems Automation and Management (February 2020)," *Journal of Petroleum Technology*, vol. 72, pp. 60–60, 02 2020.

[151] Equinor, "Disclosing all volve data," 2018.

[152] ESpectrum, "Espectrum technologies oil & gas llc," 01 2020.

[153] E. Z Losoya, N. Vishnumolakala, E. Gildin, S. Noynaert, Z. Medina-Cetina, J. Gabelmann, D. Glowka, and R. Houston, "Machine learning based intelligent downhole drilling optimization system using an electromagnetic short hop bit dynamic measurements," in *SPE Annual Technical Conference and Exhibition*, OnePetro, 2020.

[154] S. B. Kotsiantis, I. Zaharakis, P. Pintelas, *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[155] S. M. Kuo, B. H. Lee, and W. Tian, *Real-time digital signal processing: fundamentals, implementations and applications*. John Wiley & Sons, 2013.

[156] N. Pastorek, K. R. Young, and A. Eustes, "Downhole sensors in drilling operations," pp. 11–13, 2019.

[157] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," in *Ensemble machine learning*, pp. 157–175, Springer, 2012.

[158] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.

[159] W. N. Venables and B. D. Ripley, *Modern applied statistics with S-PLUS*. Springer Science & Business Media, 2013.

[160] F. Hastie, Tibshirani, *The Elements of Statistical Learning*. Springer-Verlag New York, 2 ed., 2009.

[161] D. Meyer, "Support vector machines: The interface to libsvm in package e1071," 2004.

[162] K. Hangos and I. Cameron, "Process modelling and model analysis," 2001.

[163] J. Sansana, M. N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L. H. Chiang, and M. S. Reis, "Recent trends on hybrid modeling for industry 4.0," *Computers & Chemical Engineering*, vol. 151, p. 107365, 2021.

[164] M. L. Thompson and M. A. Kramer, "Modeling chemical processes using prior knowledge and neural networks," *AIChE Journal*, vol. 40, no. 8, pp. 1328–1340, 1994.

[165] A. P. Teixeira, N. Carinhas, J. M. Dias, P. Cruz, P. M. Alves, M. J. Carrondo, and R. Oliveira, “Hybrid semi-parametric mathematical systems: Bridging the gap between systems biology and process engineering,” *Journal of Biotechnology*, vol. 132, no. 4, pp. 418–425, 2007. In Memoriam W.-D. Deckwer: Merging Process Engineering and Systems Biology.

[166] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, “Integrating physics-based modeling with machine learning: A survey,” *arXiv preprint arXiv:2003.04919*, vol. 1, no. 1, pp. 1–34, 2020.

[167] M. Von Stosch, R. Oliveira, J. Peres, and S. F. de Azevedo, “Hybrid semi-parametric modeling in process systems engineering: Past, present and future,” *Computers & Chemical Engineering*, vol. 60, pp. 86–101, 2014.

[168] G. Payette, B. Spivey, L. Wang, J. Bailey, D. Sanderson, R. Kong, M. Pawson, and A. Eddy, “A real-time well-site based surveillance and optimization platform for drilling: Technology, basic workflows and field results,” in *SPE/IADC Drilling Conference and Exhibition*, OnePetro, 2017.

[169] C. B. Mann, F. E. Dupriest, and S. F. Noynaert, “Successful design and operational practices to mitigate common bit damage mechanisms in hard laminated formations,” in *IADC/SPE Drilling Conference and Exhibition*, OnePetro, 2016.

[170] R. Pessier and M. Fear, “Quantifying common drilling problems with mechanical specific energy and a bit-specific coefficient of sliding friction,” in *SPE annual technical conference and exhibition*, OnePetro, 1992.

[171] A. Bajpai, M. Ghasemi, and X. Song, “Design and simulation of a lab-scale down-hole drilling system,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 233, no. 8, pp. 2591–2598, 2019.

[172] J. Koomey, S. Berard, M. Sanchez, and H. Wong, “Implications of historical trends in the electrical efficiency of computing,” *IEEE Annals Hist. Comput.*, vol. 33, pp. 46–54, mar 2011.

[173] R. Teale, "The concept of specific energy in rock drilling," in *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, vol. 2, pp. 57–73, Elsevier, 1965.

[174] R. G. Valenti, I. Dryanovski, and J. Xiao, "A linear kalman filter for marg orientation estimation using the algebraic quaternion algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, pp. 467–481, Feb 2016.

[175] R. G. Valenti, I. Dryanovski, and J. Xiao, "Keeping a good attitude: A quaternion-based orientation filter for imus and margs," *Sensors*, vol. 15, no. 8, pp. 19302–19330, 2015.

[176] B. Sensortec, "Bst-bmi160-ds000-07 small, low power inertial measurement unit." `https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMI160-DS000-07.pdf` (Accessed Jun, 2018)., Feb. 2015.

[177] P. A. Amundsen, S. Ding, B. K. Datta, T. Torkildsen, and A. Saasen, "Magnetic shielding during MWD azimuth measurements and wellbore positioning," vol. 25, pp. 219–222, Society of Petroleum Engineers (SPE), jun 2010.

[178] T. Torkildsen, I. Edvardsen, A. Fjogstad, A. Saasen, P. A. Amundsen, T. H. Omland, *et al.*, "Drilling fluid affects mwd magnetic azimuth and wellbore position," in *IADC/SPE Drilling Conference*, Society of Petroleum Engineers, 2004.

[179] T. I. Waag, T. Torkildsen, P. A. Amundsen, E. Nyrnes, and A. Saasen, "The design of BHA and the placement of magnetometer sensors influence how magnetic azimuth is distorted by the magnetic properties of drilling fluids," vol. 27, pp. 393–406, Society of Petroleum Engineers (SPE), sep 2012.

[180] A. Scott, "Determining downhole magnetic interference on directional surveys," in *Middle East Technical Conference and Exhibition*, Society of Petroleum Engineers, 1979.

[181] D. Zijsling and R. Wilson, "Improved magnetic surveying techniques: Field experience," in *Offshore Europe*, Society of Petroleum Engineers, 1989.

[182] C. Cheatham, S. Shih, D. Churchwell, J. Woody, and P. Rodney, "Effects of magnetic interference on directional surveys in horizontal wells," in *SPE/IADC Drilling Conference*, Society of Petroleum Engineers, 1992.

[183] A. Russell and R. Roesler, "Reduction of nonmagnetic drill collar length through magnetic azimuth correction technique," in *SPE/IADC Drilling Conference*, Society of Petroleum Engineers, 1985.

[184] P. G. A.G. Brooks, "Practical application of a multiple-survey magnetic-correction algorithm," vol. 51, pp. 36–37, Society of Petroleum Engineers (SPE), feb 1999.

[185] S. Stewart and E. Z. Losoya, "User guide and final technical notes for the MARG DBMM complimentary filter." IM-2018, 2018.