Preface

This report is written as a master thesis in the subject of *TTK4900 Technical cybernetics, master thesis (No; Teknisk kybernetikk, master oppgave)* for the two year master program in industrial cybernetics at the Norwegian University of Science and Technology (NTNU) in Trondheim. The thesis is written for the Department of Engineering Cybernetics, in cooperation with the Department of Geoscience and Petroleum. The work is centered around the design, building and implementation of instrumentation setup and control system for NTNU's contribution to the annual international Drillbotics competition, where universities competes in building an autonomous miniature drilling rig to solve an engineering challenge stated by DSATS. The NTNU team qualified for this competition based on a design report delivered to the competition committee last semester.

The NTNU Drillbotics 2019 team consist of 3 student from the petroleum department and myself from the cybernetics department. My team members from the petroleum's department are writing a separate master thesis as a group on the same subject, with focus on mechanical design, directional drilling technology and the overall system implementation, with the Drillbotics problem statement in center. My team members are; Hilde Eiken Helle, Juan Manuel Montoza and Muhammad Umer Azam.

The Drillbotics project has been great experience, working in an interdisciplinary team to solve a highly relevant problem statement with autonomous drilling and autonomous process performance control. The major part of my contribution to the Drillbotics project is centered around the design, building and implementation of an inertial measurement unit (IMU) used to track the position of the drillbit in real time. The printed circuit board (PCB) has been designed together with chief engineer Steffen Wærnes Moen, who is a part of the NTNU Drillbotics support team as an electrical engineer. Steffen has also provided with sufficient training in how to make a PCB by the photoresist method in the PTS electrical lab, and help with setting up the I^2C communication on the microcontroller and modbus communication for all motors. Soldering, debugging and programming of the sensor card has been a time consuming but important part of the project, where using downhole data in the closed loop control of the rig is a requirement. Besides the sensor card, my work has revolved around programming of the control system in Labview. As the only cybernetics student, I have been the overall responsible for the programming part of the project. However, due to the time consuming work with the downhole sensor card, programming of individual motor control and individual sensor readings have been a collaborative work between Hilde Eiken Helle, Muhammad Umer Azam and me. All closed loop controls, PID tuning, signal filtering, sensor calibration and magnetic distortion handling is programmed by me. The state machine have been designed by the team as a whole and programmed by me, Hilde and Umer.

First, I would like to thank Steffen Wærnes Moen and Noralf Vedvik for helping with setting up communication protocols for actuators and sensors, and mechanical fixes on the rig, respectively. They have both been very helpful with their insight and expertise, and great discussion partners in the day to day problem solving of practical and theoretical issues. Steffen and Noralf also joined the team as technical support when the team competed in the Drillbotics competition in Celle, Germany. Without Steffen and Noralf this project would not have been possible.

I would also like to thank all supervisors related to the project; Lars Struen Imsland, Sivge Hovda, Tor Berge Gjersvik, Alexey Pavlov and Sigbjørn Sangesland. Your guidance and constructive feedback have been very helpful during biweekly status meetings, where your expertise and experience as engineers from industrial or scientific projects have helped the team see challenges from a different perspective. John-Morten Godhavn from Equinor, the initiator of NTNU Drillbotics when he worked at NTNU, have also frequently been attending the status meetings and helped with his knowledge from the industry. An interdisciplinary support team have been a vital part of this project as Drillbotics requires knowledge within a variety of disciplines such as mechanical-, electrical-, petroleum- and cybernetics engineering. Biweekly status meeting have also helped the team keep up progression and to prioritize critical parts of the project.

Finally, I would like to thank our sponsors. The Department of Geoscience and Petroleum at NTNU have provided the team with funding to buy mechanical parts, electrical components and for rig-/team travel expenses and accommodation to attend the Drillbotics competition in Celle, Germany. Lyng Drilling have

helped with both guidance during design and funding of the manufacturing of custom drillbits. Thank you all for your contribution and for supporting innovative research projects like Drillbotics.

Abstract

This thesis describes the work related to the design and implementation of instrumentation setup and control system for the NTNU Drillbotics' contribution to the 2019 Drillbotics competition. Drillbotics is an annual international competition for universities to design and build a miniature drilling rig that uses sensors and control algorithms to autonomously drill a rock sample. The 2019 problem statement is defined as follows:

Design a rig and related equipment to autonomously drill a well, using downhole sensors, that obtains as much horizontal displacement from surface as possible along the rock's "north" direction, as quickly as possible while maintaining borehole quality and integrity of the drilling rig and drillstring.

The NTNU Drillbotics team consist of three students from the petroleum department and one student from the cybernetics department. The team will solve the problem statement by performing a slide while drilling operation by using a bend sub, with a downhole motor for bit rotation and a top drive servo motor for pipe rotation and steering. A downhole IMU device will be used to measure the orientation of the directional tool, and further used for steering and position tracking.

The scope for the thesis is to design, build and implement a downhole sensor card, a magnetic ranging setup, topside actuation- and instrumentation setup and an autonomous control system to control the drilling operation.

A downhole sensor card containing an inertial measurement unit (IMU) have been successfully designed, built and implemented along with an active magnetic ranging setup. Results from practical experiments show that the IMU can measure the orientation of the BHA toolface, and track the position of the drillbit in 6 DOF. The results also show that active magnetic ranging can be used to improve the accuracy of position tracking and position control in a miniature drilling rig with high internal magnetic distortion. Further testing is needed to confirm accuracy and robustness of the position controller during drilling.

Topside actuation- and instrumentation setup have been implemented and successfully tested, including a new top drive, a solenoid valve and a tank level sensor.

A PID WOB controller and PI position controller have been designed and implemented in the control system to control the WOB and drillbit position respectively. The result shows a stable WOB control response using Kalman filtered feedback. The position controller is not tested due to a technical issue with an electrical swivel, but the internal PI toolface controller have been successfully tested by performing practically experiment using active magnetic ranging.

A control system has been designed and implemented in Labview to autonomously control the miniature drilling rig to solve the Drillbotics Competition 2019 problem statement; e.g. WOB-, toolface- and position control, with two-layered safety logic and setpoint optimization of the weakest actuator in a 9-state state machine. In addition, a simulation script has been implemented to simulate the boolean logic in the state machine. The simulation script is an un-modeled copy of the control system, where all motor and sensor inputs/outputs are replaced by manually controlled variables. The script can potentially lay the basis for a digital twin for future project groups.

The European final for the Drillbotics competition was held in Celle, Germany, from 11-14. June. Unfortunately, the NTNU rig did not arrive in time for the competition due to problems with German customs control. However, the NTNU team got to present the design and proposed solution, and to demonstrate the working principle of the control system with a simulation script.

The 2018/19 problem statement by DSATS has proven to be a challenging task for all teams that entered the Drillbotics competition. The increased complexity with limitations for outer dimensions, time and budget have forced all teams to think out of the box. This clearly emerged during the European final in Celle, Germany, where all teams that entered the A-final presented their own solutions on how the task should be solved with fundamental differences in terms of steering, actuation and control algorithm. None of the teams presented a perfect solution without a vital trade-off. This shows how challenging the problem statement is,

and why the Drillbotics competition can be a drive for innovation among students and within the oil- and gas industry.

Sammendrag

Denne avhandlingen beskriver arbeidet med design og implementering av instrumentering og kontrollsystem for NTNU Drillbotics' bidrag til Drillbotics-konkurransen 2019. Drillbotics er en årlig internasjonal konkurranse for universiteter til å designe og bygge en miniatyr borerigg som ved bruk av sensorer og kontrollalgoritmer skal borre en steinprøve helt autonomt. Problemstillingen for 2019 konkurransen er definert som følger:

Design en rigg med relatert utstyr som ved bruk av nedi hulls sensorer, autonomt skal borre en brønn som borrer mest mulig horisontalt mot nord, så fort som mulig, mens borehullskvalitet, rig- og borestrengsintegritet opprettholdes.

NTNU Drillbotics består av tre studenter fra institutt for petroleumsteknologi og en student fra institutt for teknisk kybernetikk. Teamet vil løse problemstillingen ved å gjennomføre en slide-while-drilling operasjon, ved å bruke en nedi hulls motor til å rotere borekronen og en servo motor til å rotere og styre borestrengen. Boreverktøyet vil ha en liten vinkel, eller en såkalt "bend sub". En nedi hulls sensor vil bli brukt til å måle orienteringen til boreverktøyet og bruke dette til å spore posisjonen.

Omfanget av denne avhandlingen innebærer design, bygging og implementering av et nedi hulls sensorkort, et magnetisk sporingsoppsett, topside motor- og instrumenteringsoppsett, og et autonomt kontrollsystem.

Et nedi hulls sensorkort, inkludert en IMU måleenhet har blitt designet, bygget og implementert sammen med et oppsett for aktiv magnetisk sporing. Resultater gjennom praktiske eksperiment viser at IMU enheten kan måle orienteringen til boreverktøyet, og spore posisjonen til borekronen i 6 frihetsgrader. Resultatet viser også at aktiv magnetisk sporing kan bli brukt til å forbedre nøyaktigheten til posisjonssporing og posisjonskontroll for en miniatyr borerigg som er utsatt for høy intern magnetisk forstyrrelse. Videre testing er nødvendig for å teste robustheten og nøyaktigheten til posisjonskontrolleren under boring.

Topside motor- og instrumenteringsoppsett har blitt implementert og testet, inkludert en ny top drive servomotor, elektronisk ventil og en nivåmåler.

En PID-kontroller og en PI-kontroller har blitt designet og implementert i kontrollsystemet for å kontrollere henholdsvis vekt på borekronen og borekroneposisjon. Resultater viser en stabil vekt-på-borekrone responsen hvor et Kalman filter er brukt på tilbakekoblingen. Posisjonskontrolleren er ikke testet i riggen grunnet tekniske problemer med den elektriske slipp-ringen, men den interne toolface PI kontrolleren er testet med et vellykket resultat ved å gjennomføre praktiske eksperiment ved bruk av aktiv magnetisk sporing.

Et kontrollsystem har blitt designet og implementert i Labview for å autonomt kontrollere boreriggen til å løse 2019-problemstillingen. Dette innebærer Vekt-på-borekrone- og posisjonskontroll, to nivåer med sikkerhetslogikk, en automatisk settpunktvelger som tilpasser seg til svakeste motor og en tilstandsmaskin med 9 nivåer. I tillegg er et simuleringsprogram laget for å kunne simulerer logikken i tilstandsmaskinen. Simuleringsprogrammet er en u-modellert kopi av kontrollsystemet hvor alle motor og sensor innganger og utganger er byttet ut med manuelt kontrollerte variabler. Simuleringsprogrammet kan ligge grunnlaget for videre utvikling av en digital tvilling ved seinere prosjekt.

Europa-finalen for Drillbotics konkurransen ble holdt i Celle i Tyskland 11.-14. Juni. Uheldigvis satt riggen til NTNU fast i tollen og rakk aldri frem til konkurransen. Laget fikk uansett presentert frem designet og løsningen, og på konkurransedagen viste laget fram simuleringsprogrammet for å demonstrere logikken i kontrollsystemet.

Problemstillingen for Drillbotics 2019 har vist seg å være en svært utfordrende oppgave for alle lag som deltok i konkurransen. Den økte vanskelighetsgraden i kombinasjon med begrensninger for ytre dimensjoner, tid, og budsjett har tvunget alle lag til å tenke utenfor boksen. Dette ble tydeliggjort under europa-finalen hvor alle lag som deltok i A-finalen presenterte løsninger med fundamentale forskjeller på hvordan man skulle løse utfordringen med hensyn på styring, motordesign og kontrollalgoritme. Ingen av lagene presenterte en løsning hvor man ikke fjernet eller svekket en vital del av den løsningen. Dette viser hvor utfordrende 2019-problemstillingen er, og hvorfor Drillbotics konkurransen kan være en driver for innovasjon blant studenter ut mot olje og gass industrien.

List of Figures

List of Tables

Nomenclature

Abbreviations

AMR	Active magnetic ranging
BHA	Bottom hole assembly
DH	Downhole
DP	Drill pipe
EMM	Mini electric motor
HM	Hoisting motor
I^2C	Inter-Integrated Circuit
IMU	Inertial measurement unit
MWD	Measure while drilling
PCB	Printed circuit board
PDM	Positive displacement motor
PM	Pump motor
PMR	Passive magnetic ranging
ROP	Rate of penetration
RPM	Rotations per minute
SP	Set point
SWD	Slide while drilling
TD	Top drive
TVD	True vertical depth
$UART$	Universal Asynchronous Receiver/Transmitter
WOB	Weight on bit
DOF	Degrees of freedom

Constants

g	gravitational acceleration
k_d	Derivative gain [PID]
k_i	Integral gain [PID]
k_p	Proportional gain [PID]
r	reference [PID]

Variables

$\hat{x}$	Estimated state variable [kalman]
$\hat{y}$	Estimated measurement output [kalman]
e	Error
u	System input

x	State variable
y	Measurement output

Table of Contents

1 Introduction

1.1 Background

In January 2016, the oil price reached its lowest point in 13 years with a price of 27,67 USD per barrel [1]. This marked the rock bottom of the 2015 oil crisis that lasted out 2017. The crisis led to decreased investments in the oil marked and many companies had to lay off employees. However, the crisis also forced the industry to start think differently. The last couple of years, it can be observed an increased focus on business sustainability, digitization and automation in the oil and gas industry. Better Resource Utilization in the 21st century (BRU21) is a research and innovation program, launched by NTNU in collaboration with industrial partners, in digital and automation solutions for the oil and gas industry. The BRU21 report suggests a way forward with increased research and development on automated solutions, e.g. the drilling process (among other focus areas) [2].

Equinor and MHWirth are among the companies that have made big investments within automated drilling the last decade [3]. With an average cost of 450000 USD per day to hire a drilling rig, and between 30 and 70 days to complete a drilling operation, there is a huge potential for cost reductions. A drilling operation is among the most high risk and complex operations in any industry today. Equinor sees autonomous drilling as a necessity in the future, where an autonomous drilling operation will lead to increased efficiency in terms of cost per barrel, and increased material- and environmental safety. The company is working on a plug-and-play solution that can be used at multiple locations. Development of autonomous drilling technology will not only benefit the oil and gas industry, but the whole energy production sector in the long run. The idea is that if you solve autonomy for high complex, high risk operations, autonomy for low risk, low complex operations will become easy in comparison. Complexity drives innovation. This is essential if lower efficiency renewable energy sources is to be economically sustainable compared to oil and gas, when the industry is transiting more and more to renewables over the next decades.

One of the initiatives to accelerate the development of innovative technology in the oil and gas industry, is DSATS [4]. Drilling Systems Automation Technical Section (DSATS) is a part of the Society of Petroleum Engineers (SPE). DSATS work actively to promote, standardize and support all initiatives within drilling automation systems. One of these initiatives is Drillbotics, an annual international competition for universities to design and build a miniature drilling rig that uses sensors and control algorithms to autonomously drill a rock sample provided by DSATS [5]. This is the third year NTNU enters the competition, with prior finishes at 2nd (2017) and 1st (2018) place [6][7]. The rigs from previous years are allowed to be re-used and modified. Hence, the 2018 rig design can be defined as the 2019 teams starting point [8].

This thesis is centered around the work related to the design and building of the instrumentation setup and autonomous control system for the NTNU Drillbotics' contribution to the Drillbotics 2019 competition. The report is based on the design proposal the team delivered to DSATS in December 2018, that qualified NTNU as the top ranked team for the competition [9][10]. The report is also building on the work done by Mikkel Leite Arnø, responsible for the control system for the NTNU Drillbotics 2018 team [11].

1.2 Drillbotics Competition 2019

This year marks the fifth annual international Drillbotics competition for universities to design and build an autonomous miniature drilling rig to solve a problem statement defined by DSATS. During previous competitions, the main focus has been on autonomously dealing with drillstring mechanics in vertical drilling through a multilayered rock sample. The main focus for the 2019 competition is on autonomous directional drilling, where drillstring mechanics will still be a prioritised concern. The competition final is divided in two locations; Houston (Texas) for US teams and Celle (Germany) for non-US teams. The European final is held from 11.-14. of June 2019. Below, the 2019 problem statement and summarized competition objective is presented. Summarized rig requirements is presented in Table 1. For more details, see the complete *Drillbotics Guidelines 2019* in Appendix A.

Problem statement for the 2018/2019 competition:
Design a rig and related equipment to autonomously drill a well, using downhole sensors, that obtains as much horizontal displacement from surface as possible along the rock's "north" direction, as quickly as possible while maintaining borehole quality and integrity of the drilling rig and drillstring.

Summarized competition objective:

- Drill a 4 inch vertical pilot hole followed by a deviated well along the rock's "north" centerline. A horizontal displacement of 2 3/8 inch along the "north" centerline will result in max score. Any deviation from the "north" centerline will result minus points subtracted from the final score. A score chart for drilling accuracy can be found in Appendix A.
- Closed loop control of the rig based on downhole data is mandatory.
- No lateral forces may be applied above the rock.
- No instrumentation of the rock sample.
- The control system run must autonomously through the whole operation with only a start and stop button. Team interaction is allowed if necessary, but will result in minus points.
- The teams will be graded on mechanical-, electrical and instrumentation design, control algorithm and innovative ways to present the operation data in continuous time. The team will also be graded on how fast and accurate they drill and borehole quality.
- Competition time limit is 3 hours.
- Electrical power consumption supplied from the grid is limited to 25hp or 18,6 kW.
- Budget is limited to 10.000 USD.
- Rigs from previous years team can be re-used and modified for the new problem statement.
- Teams are limited to 5 students per team.

Table 1: Summarized Rig Requirements

Part:	**Dimensions [in]:**	**Material:**	**Comment:**
Drillpipe	0.375 (OD)	Aluminium	Drillpipes will be provided by DSATS.
Drillbit	<1.5 (OD), <2.0(L)	Optional	Two 1.25" drillbits will be provided by DSATS. Optional to use own design.
Bit sub	<10% of drillbit OD	Optional	Own design is required.
Rock Sample	12x24x24 (WxLxH)	Sandstone	Rock samples will be provided for the competition.
Riser	<8.0(L)	Optional	Optional but useful for return flow handling.
BHA	<1.5 (OD)	Optional	Own design is required.
DH sensors	Optional	Optional	Required to be included in the closed loop control.

In Figure 1, the rock sample and a suggested well path is presented. The well path illustrated is not a requirement, but illustrates a perfect path with the minimum requirements for max score. Any horizontal displacement that exceed 2 3/8 inch along the rock's "north" centerline will still result in the max score.

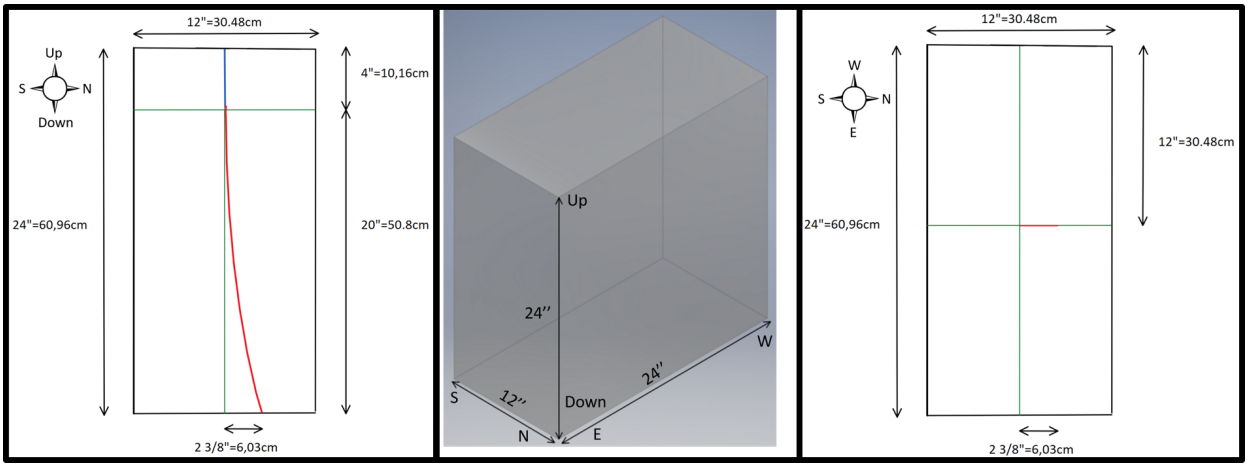


Figure 1: Rock sample and minimum drill path requirements for max score.

1.2.1 European Final

The European final was held in Celle, Germany, from 11.-14. June 2019. The final was divided into two parts; (1) a 30 min presentation of the design and proposed solution followed by a 60 min QA, and (2) a practical demonstration of the drilling rig during the competition day. Unfortunately, the NTNU rig was stuck in the German customs service and did not arrive in time for the competition. Only 2 out of 5 rigs made it for the competition due to difficulties with the customs service, and only 1 out of 5 drilled on the competition day (due to technical problems).

However, the NTNU team got to present its design and solution and to demonstrate the working principle of the control system with a simulation script. Competition results will be posted on the official Drillbotics blog when results from the European and American final is compared [12].

1.3 Scope

The scope for this master thesis is centered around the design, building and implementation of instrumentation setup and control system for the NTNU Drillbotics' contribution to the Drillbotics 2019 competition. With the introduction of directional drilling for this years competition, the competition scope and complexity have increased compared to earlier years. It is expected that the problem statement will remain similar next year with minor changes. Due to this, the main objective for this thesis is focused on the design and implementation of a working solution to solve the Drillbotics problem statement, where advanced modeling and optimization of the drilling process will not be a part of this thesis' scope. The scope for this thesis implies:

- Design, building and implementation of a downhole sensor card for an inertial measurement unit (IMU) capable of tracking the position of the drillbit in 6 DOF in real time.
- Design and implementation of a magnetic ranging setup that can be used for the drillbit to navigate in a highly magnetic distorted environment.
- Design and implementation of the topside actuation- and instrumentation setup for the miniature drilling rig.
- Design and implementation of a control system in Labview to autonomously control a miniature drilling rig to solve the Drillbotics Competition 2019 problem statement; e.g. WOB-, Torque-, ROP-, toolface- and position control, with boolean safety logic and setpoint optimization in a 9 state state machine.

- Design and implementation of a drilling simulator (un-modeled) for early implementation and testing of code to simulate the boolean logic in the state machine.
- Collaborate in the overall system design to ensure controllability and observability from a control perspective point of view; e.g. mechanical steerability of the downhole directional tool.

1.4 The team

The NTNU Drillbotics team 2019 consist of four students; three from the petroleum department and one from the cybernetics department. The team is supported by three supervisors from the petroleum department and one from the cybernetics department. Further, the team is supported by one electrical engineer and one mechanical engineer at the NTNU PTS electrical lab and mechanical workshop. The supervisors role is to guide and give feedback in biweekly status meetings. The lab engineers role is to support the day to day work in the test hall. A project team- and support team organizational chart is presented in Figure 2 and Figure 3 respectively.

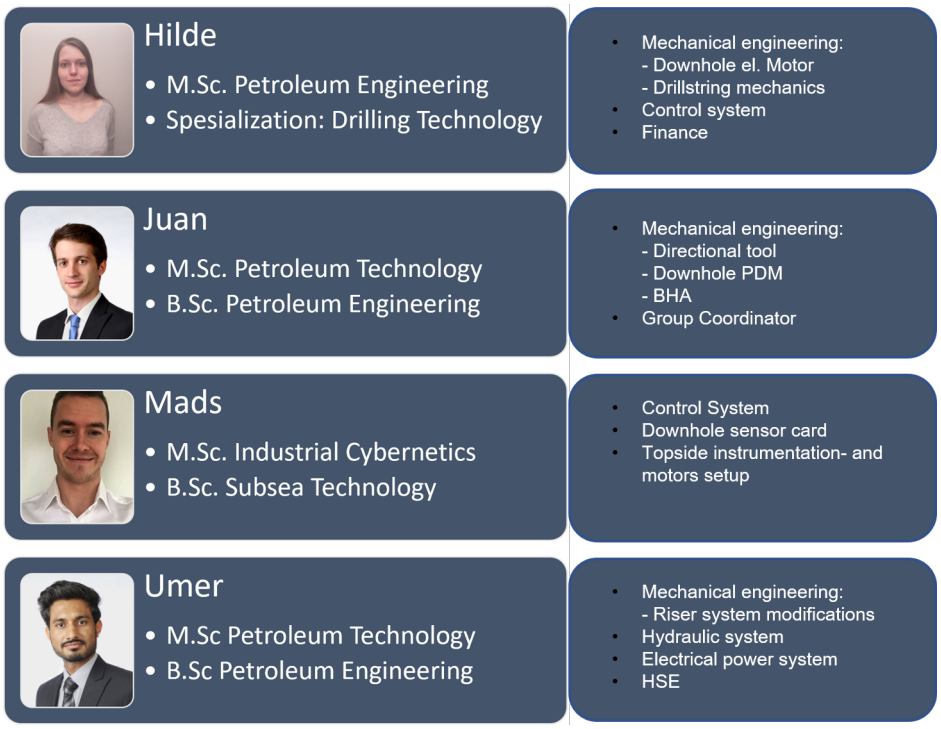


Figure 2: Project team organization chart

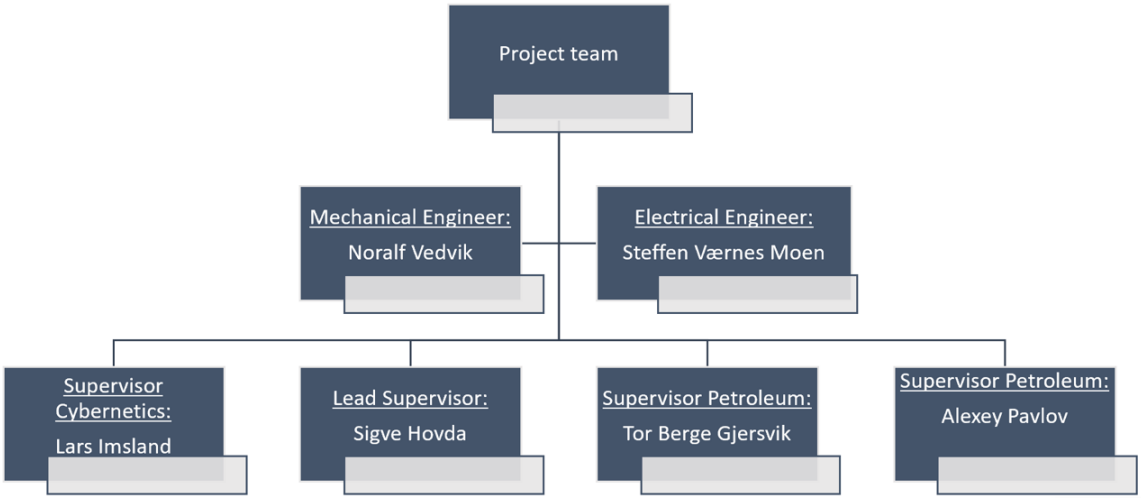


Figure 3: Support team organization chart

2 Theoretical Background

In this section, relevant theory within process and performance control, MEMS magnetometer technology and magnetic ranging will be described. The theoretical background functions as a support chapter, where the objective is to create a theoretical foundation the report can base its decisions and assumptions on, and to create an understanding of the challenges the project team is facing regarding position tracking and position control in a highly distorted magnetic environment. The report frequently refers back to the theory section when appropriate.

2.1 Control Theory

Relevant theory within process- performance control was first introduced in the *Drillbotics design report* [9]. This theory is highly relevant for the implementation stage of the project, and will therefore be rendered in its full form. However, some constants, variables and sentences will be assigned to new letters or reformulated. All text in subsection 2.1 in taken from *4.3 Control Theory*[9].

2.1.1 PID Controller

A PID controller, or a proportional-integral-derivative controller, is a highly used closed loop feedback controller within process/performance control in many industries. The team will use this controller to control the rig response during the drilling operation. Figure 4 shows an overview of the PID control scheme.

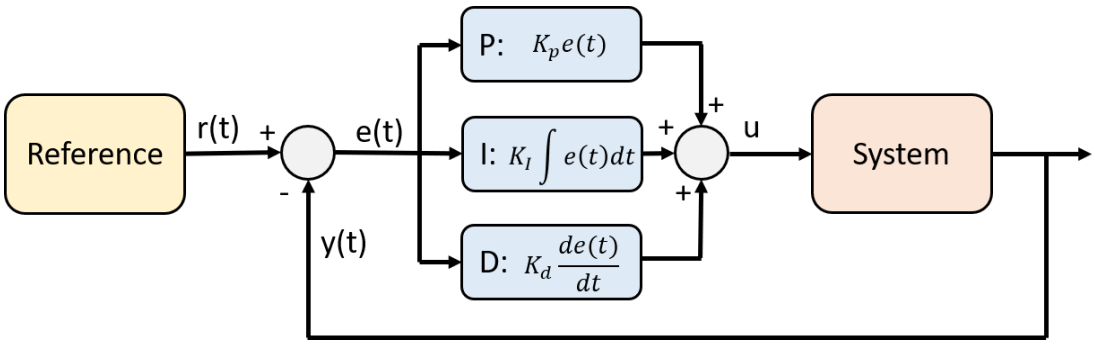


Figure 4: Overview of PID control scheme

The working principle of the PID controller starts with measuring the control state $y(t)$. This is further used in feedback to the controller where an state error $e(t)$ is calculated compared to a chosen set point. The error is then fed into the controller where proportional, integral and derivative effect is applied to calculate the new input to the system. The main goal by using a PID controller is to minimize the state error and to achieve a steady state response. A mathematical expression of the controlled input u can for a PID controller be defined as

$$u(t) = [K_p e + K_I \int e(t)dt + K_d \frac{de(t)}{dt}] \tag{1a}$$

where

$$K_I = \frac{K_p}{T_i} \tag{1b}$$

$$K_d = K_p T_d \tag{1c}$$

and the equation in the frequency domain becomes

$$u(s) = e(s)[K_p + \frac{K_I}{s} + sK_d] \tag{1d}$$

To better explain how the PID controller effects the system response, a graphical illustration is presented in Figure 5. Along with the figure, the effects can be explained accordingly:

- **P:** The proportional effect determines how fast the output response is going to the reference. That is, a large proportional gain K_p results in a small rise time. The rise time is the time it takes for the system to reach a set percentage of the reference value after an impulse response. The rise time is shown in the figure.
- **I:** The integral effect eliminates the error between the set point and the measurement, often referred to as the steady state error. That is, large integral gain K_I results in a response eliminating the steady state error, moving the response up to the desired reference. The steady state error is shown in the figure.
- **D:** The derivative effect, also known as the dampening effect, dampens out oscillations on the response. By adjusting the derivative gain K_d, the controller can determine how fast oscillations should be dampened out. That is, the derivative gain determines the settling time which is the time it takes for the system to find a steady state after a impulse response.

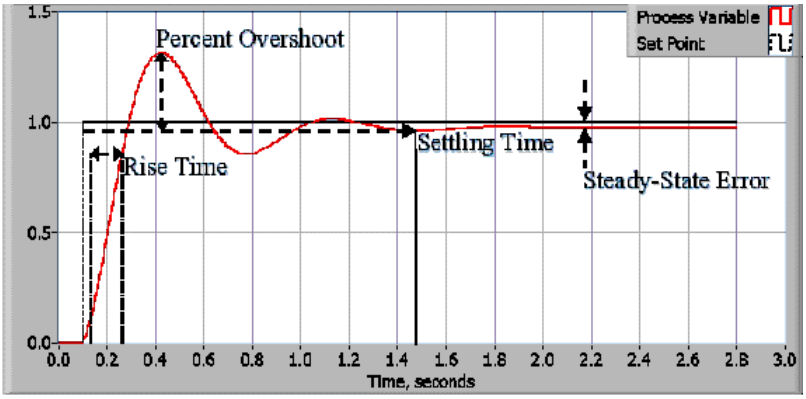


Figure 5: Overview of PID control response [13]

The autonomous control system will during the drilling operation perform impulse responses when switching between different set points for the control states. Hence, a proper tuning of these effects will be important to achieve an appropriate response. All theory used for this short description of the working principles of a PID controller is based on literature written by Jens G. Balchen et al.[14].

2.1.2 Kalman Filter

A Kalman filter is an optimal estimation algorithm that predicts a parameter of interests, such as location, speed and direction in the presence of noise in the measurement. It is often used to predict variables of interest that can only be measured indirectly. In this project the problem case is to measure and control the position of the drill bit in a directional drilled well. Hence, the position cannot be measured directly, so it needs to be estimated by using the measured ROP, the measured arc length of drill pipe and by measuring the orientation of the bit using down hole sensors. With the expectations of high-frequent noise on the down hole measurements, a Kalman filter is highly suited. To explain how a Kalman filter works, it's useful to start by explaining how a state observer works.

State Observer
A state observer is primarily a mathematical model that estimates the measured output and compares it with the real measurement. A state observer is shown in Figure 6. The model takes in the system input u and calculates an estimated output $\hat{y}$. The estimated output $\hat{y}$ is then compared with the measured y and an error e_y is calculated. The loop is closed by feeding the error e_y multiplied with the observer gain L back into the system. The overall goal is to eliminate the estimator error by finding to best value for the observer gain L. This is where a Kalman filter comes in.

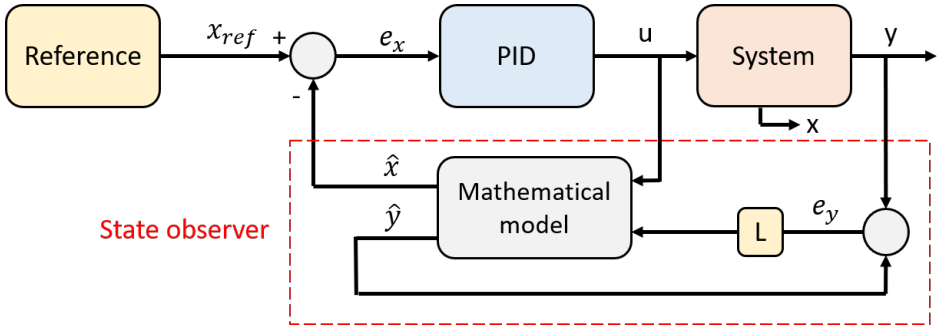


Figure 6: Overview of state estimator scheme

Measurement Noise Co-variance
When taking a measurement, as the angle between the orientation of our drill bit and the magnetic north, it can be expected that the measurement value will vary around the true measurement. If the variance is small and the power spectral density is constant, this noise can be defined as white noise. In Figure 7 the covariance of a signal is shown, where μ is the zero mean, σ is the variance and σ^2 is the covariance.

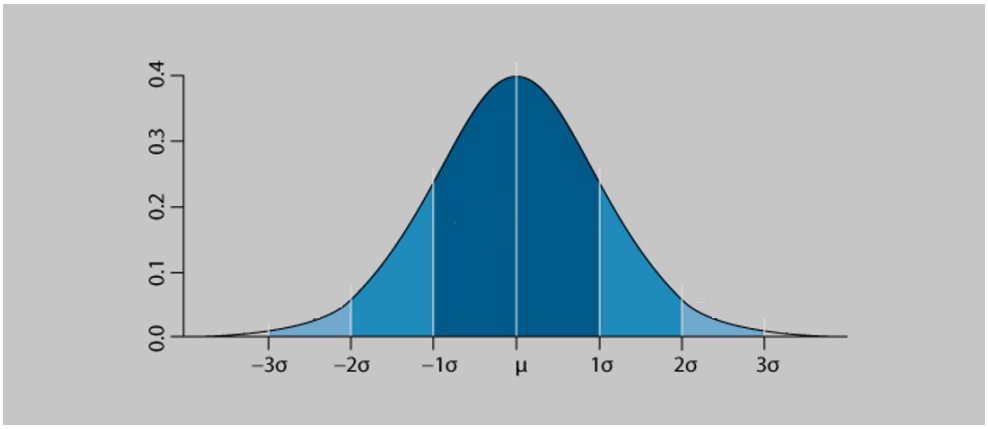


Figure 7: Graphical illustration of the covariance σ^2, showing the variance σ relative to the mean value μ

The working principle of a Kalman filter
Using the above and simple probability, it can be defined that our estimated state $\hat{x}$ is most likely the mean of the covariance of the measurement (state). In Figure 8, an illustration of the working principle of the Kalman filter is presented as a probability density function. First, an initial estimated state $\hat{x}_{k-1}$ is made. Comparing this to the state $\hat{x}_k$ in the next time step, it can be seen that the variance increases and the probability for estimating the correct state value decreases. This is the point where our true measurements come in. By multiplying the covariance of $\hat{x}_k$ and y_k, the optimal estimate can be found by taking the mean of the newly combined covariance. The Kalman filter finds these optimal estimates by iterating through an algorithm using the principles explained above.

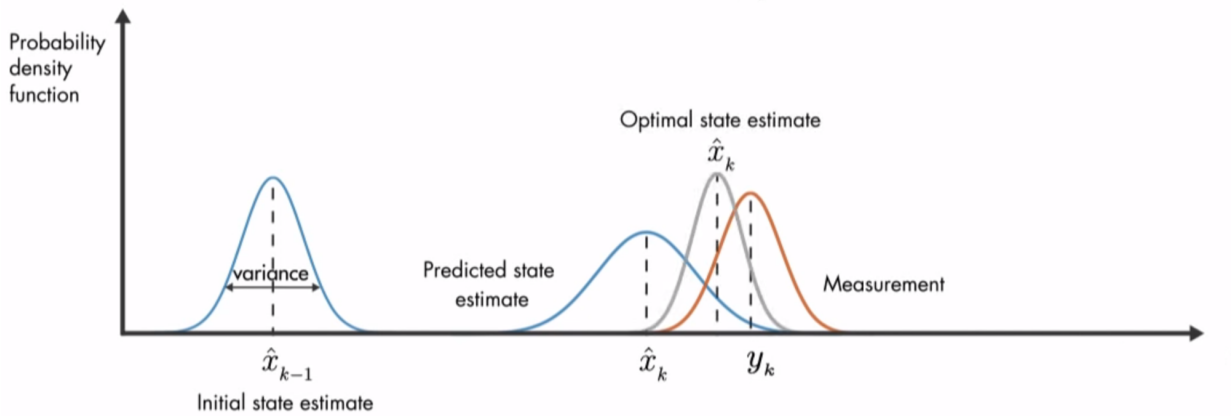


Figure 8: Illustration on how a Kalman filter finds the optimal state estimation [15].

Algorithm
The discrete time Kalman filter algorithm iterates through 4 steps. For a general process model we have:

$$x[k+1] = A_d x[k] + B_d u[k] + w[k],\ y[k] = Cx[k] + v[k] \tag{2}$$

By assuming the noise and disturbance are unbiased and white, we have:

$$E[v[k]] = 0,\ E[w[k]] = 0,\ E[v[k]v[l]^T] = \delta[k,l]R_d,\ E[w[k]w[l]^T] = \delta[k,l]Q_d \tag{3}$$

The algorithm initializes at

$$\hat{x}^-[0] = E[x(0)],\ and\ P^-[0] = E[(x[0] - \hat{x}^-[0])(x[0] - \hat{x}^-[0])^T] \tag{4}$$

and iterates through the following steps:

$$1.\ L[k] = P^-[k]C^T(CP^-[k]C^T + R_d)^{-1} \tag{5a}$$
$$2.\ \hat{x}[k] = \hat{x}^-[k] + L[k](y[k] - C\hat{x}^-[k]) \tag{5b}$$
$$3.\ P[k] = (I - L[k]C)P^-[k](I - L[k]C)^T + L[k]R_d L[k]^T \tag{5c}$$
$$4.\ \hat{x}^-[k+1] = A_d\hat{x}[k] + B_d u[k],\ P^-[k+1] = A_d P[k]A_d^T + Q_d \tag{5d}$$

At each iteration, the Kalman algorithm uses the prior estimated state $\hat{x}^-[k]$ and the prior error covariance $P^-[k]$, to calculates a new Kalman gain L[k]. The Kalman gain is used to calculate the new estimated state $\hat{x}[k]$ and the new error covariance $P[k]$. Finally, the algorithm calculates next iteration's prior estimated state $\hat{x}^-[k+1]$ and prior error covariance $P^-[k+1]$ that is used to initialize the next iteration. The algorithm returns to start and continues the same way.

2.1.3 Cohen-Coon tuning method

The Cohen-Coon tuning method is a well used PID tuning method that has proven to work well on most self-stabilization processes, such as a drilling operation. The method is based on open loop testing, where the goal is to find a transfer function for the system which can be used to identify the appropriate PID gains for an ideal system response. The method was successfully used to tune the WOB- and torque controllers

in Drillbotics 2018 [11]. Hence, the team will use this as inspiration for tuning the new controllers in 2019. A commonly used transfer function is the first-order plus dead time (FOPDT), and can be defined in the frequency domain as follows [16]:

$$H(s) = \frac{Kw^{-t_d s}}{\tau s + 1} \tag{6}$$

where t_d is the dead time, τ is the time constant and K is the process gain defined by

$$K = \frac{\Delta PV\%}{\Delta CO\%} \tag{7}$$

where PV is the process variable input and CO is the controller output.

The Cohen-Coon tuning method can be described through several steps where we will perform a step response to go from one steady state to another. The steps can be described as follows including a graphical illustration in Figure 9 [17]:

1. Choose a constant CO value and wait for the systems PV to settle at a steady state.
2. Perform a step response by changing OV a few percents and want for the system to settle at a new steady state.
3. Calculate K by using Equation 7
4. Find the maximum slope on the PV curve (inflection point) and draw a tangential line that intersects with the original level of PV. This is shown in Figure 9.
5. Calculate t_d which is the time between the beginning of the step response and the tangential intersection with original PV.
6. Calculate τ which is the time between after the dead time t_d to the time where PV has reached 63% of its total change.
7. Redo step 1-6 until you can produce similar values with small variance on the calculated constants, and use the average values in the next steps.
8. Use Cohen-Coon tuning rules in Table 2 to calculate the new controller settings. Note that these rules uses quarter-amplitude damping response, which is a rather fast response that quickly eliminates steady state error but slightly overshoots and oscillates a few times around the set point before it settles at a steady state.

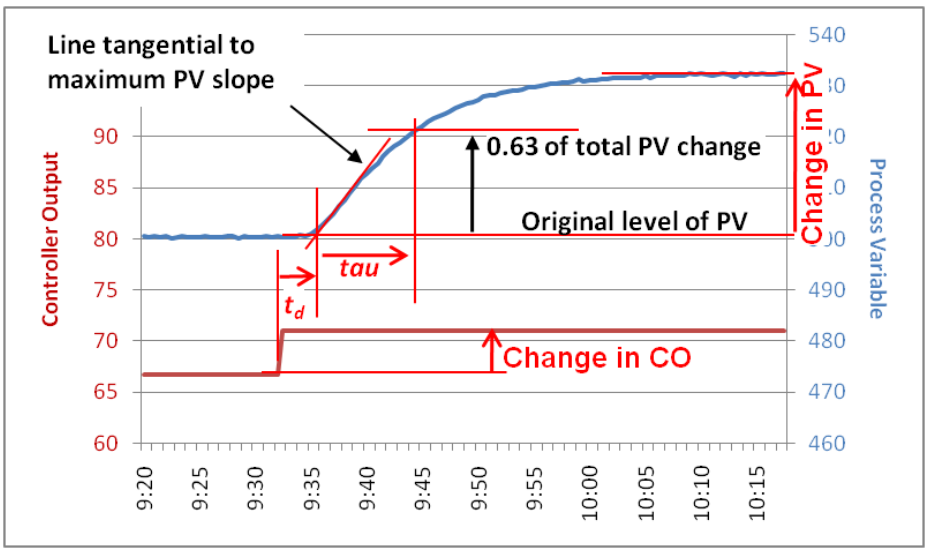


Figure 9: Cohen-Coon tuning with step response [17]

Table 2: Cohen-Coon tuning rules

	Controller Gain	**Integral Time**	**Derivative Time**
P	$K_c = \frac{1.03}{K}(\frac{\tau}{t_d} + 0.34)$		
PI	$K_c = \frac{0.9}{K}(\frac{\tau}{t_d} + 0.092)$	$T_I = 3.33t_d\frac{\tau + 0.092t_d}{\tau + 2.22t_d}$	
PD	$K_c = \frac{1.24}{K}(\frac{\tau}{t_d} + 0.129)$		$T_D = 0.27t_d\frac{\tau + 0.324t_d}{\tau + 0.129t_d}$
PID	$K_c = \frac{1.35}{K}(\frac{\tau}{t_d} + 0.185)$	$T_I = 2.5t_d\frac{\tau + 0.185t_d}{\tau + 0.611t_d}$	$T_I = 0.37t_d\frac{\tau}{\tau + 0.185t_d}$

The team will use this method as a starting point for tuning the WOB and torque controllers, but further fine tuning will most likely be performed. Since a drilling operation can be described as a "slow" process, a quick controller that overshoots will not be necessary. The tuning process using the Cohen-Coon tuning method will be further described with the implementation in phase 2.

2.1.4 First order Explicit Runge-Kutta

The first order explicit Runge-Kutta method (ERK1), also known as the Euler's method, is a simple but important integration scheme. The method uses last known function value and calculates a new value based on the function derivative multiplied by a step length. This derivation is based on literature written by Olav Egeland and Tommy Gravdahl [18]. A numerical expression of the solution can be computed using:

$$y_{n+1} = y_n + hf(y_n, t_n) \tag{8}$$

where h is the step length and $f(y_n, t_n)$ is the time-derivative of the function $y(t)$. The method is step-wise linear and highly suitable for calculating the discrete time position of a first order differential equation, or for a system with relatively slow dynamics like a directional drilling operation. The stability function for

the method can be defined as

$$R(h\lambda) = 1 + h\lambda \tag{9a}$$

where stability is ensured whenever

$$|R(h\lambda)| \leq 1 \tag{9b}$$

That is, the method has a stability region inside a unity circle around $Re = -1$. This is shown in Figure 10. For real eigenvalues λ, stability is ensured whenever

$$h \leq -\frac{2}{\lambda} \tag{9c}$$

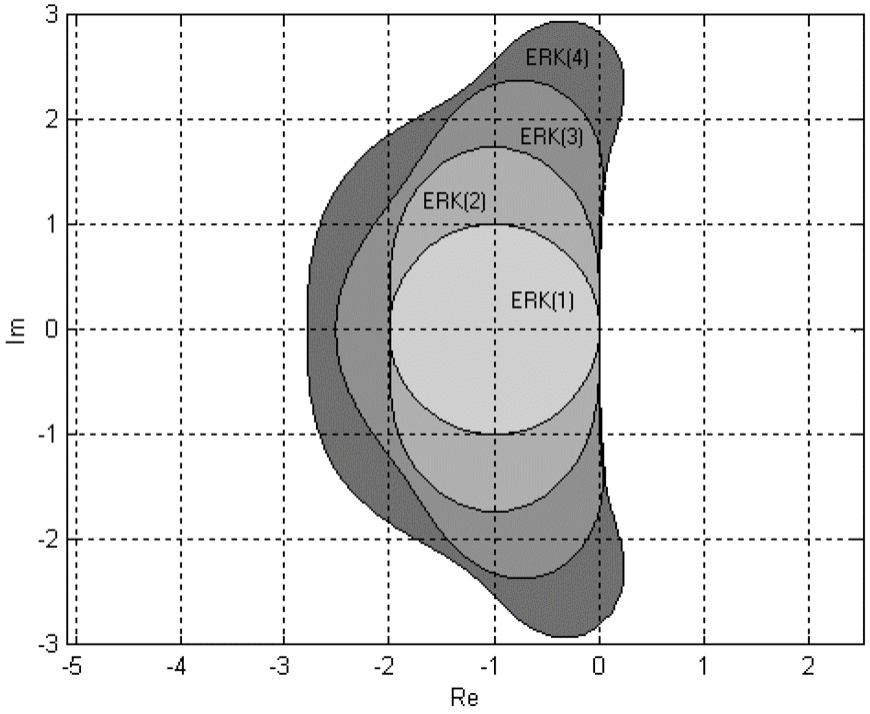


Figure 10: Explicit Runge-Kutta stability regions [18]

The Euler's method will be used to calculate the position of the drill bit in reference to a world frame located on top of the rock sample. This will be further described in subsection 5.3.

2.2 MEMS Sensor Technology

The main sensors that will be used to tracking and control the position of the bit, is a microelectromechanical system (MEMS) magnetometer and accelerometer sensor. For a miniature drilling rig build in a steel frame (magnetic material), with internal rotating parts and electrical equipment within close proximity to the magnetic sensor, it can be expected that the drilling operation will happen in a highly non-constant magnetic distorted environment. The drilling process itself will also produce vibrations. Therefore, it is important to understand how the two sensors work and how distortion will affect the measurements. Below, the working principle of the magnetometer, accelerometer and magnetic distortion will be described.

2.2.1 MEMS Accelerometer

Working Principle:
The MEMS accelerometer used for this project uses changes in capacitance together with a known spring constant to measure linear acceleration in 3 dimensions. The working principle is illustrated in Figure 11. Separate proof masses for each axis is suspended with springs in a base structure. Movable plate in the proof masses is placed in between fixed plate in the base structure, which represent capacitors. Deflection of proof masses is measured by using the capacitance difference. Multiple pairs of movable and fixed plates is added to amplify the capacitance difference. By using the capacitance difference together with the known spring constant, the linear acceleration can be measured [19].

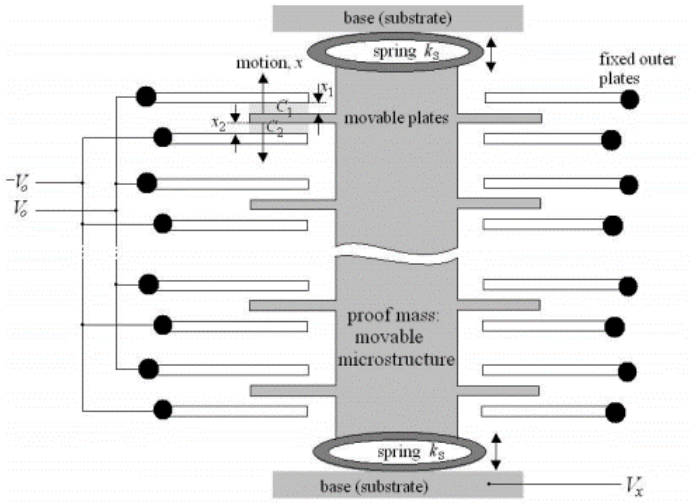


Figure 11: Capacitive accelerometer with proof masses [19]

Earth's Gravitation:
The earth's gravitational acceleration have a constant direction and magnitude of approximate $9,81 m/s^2$ or 1g perpendicular the the earth's surface (with small variations dependent on location). For a stationary sensor, the earth's gravitation can be used as a reference direction for up/down with great precision. However, MEMS accelerometers highly sensitive and noise can be expected even for a stationary sensor.

Measurement noise:
Measurement noise can for a MEMS accelerometer be defines as high frequent oscillations on the measurement signal caused by electrical disturbance from other electrical equipment or electromagnetic interference. This noise was measured an analyzed in the the *Drillbotics Design Report* and can be assumed to be white. A Kalman filter will be implemented to deal with accelerometer measurement noise.

Process noise:
The dynamic movement of the drillbit digging in to the rock sample will generate vibrations, which is transmitted from the bit and up the drillstring to the rest of the rig. This can for the accelerometer be defined as process noise. While drilling, the team will run with constant weight on bit (WOB) and constant rotations per minute (rpm). For a restricted BHA in a tight borehole, this will result in oscillating noise on the process state around the true state value. A Kalman filter will be implemented to deal with vibration process noise.

2.2.2 MEMS Magnetometer

Working Principle:
The MEMS magnetometer used for this project uses the hall effect to measure the magnetic field strength.

The working principle of a hall effect sensor is illustrated in Figure 12. A hall effect sensor takes advantage of two important magnetic characteristics; flux density (B) and polarity (north, south). A rectangular p-type semiconductor is connected to a DC power supply in a closed loop circuit. Consequently, low current is passing through the semiconductor. When the semi conductor is placed in a magnetic field, the magnetic flux lines exert a force on the passing electrons, diverting them to out to the sides. If the electron density is higher on one side, a potential difference is produced across the semiconductor. This potential difference can be defined as the hall voltage and is used to determine the strength and direction of a magnetic field [20].

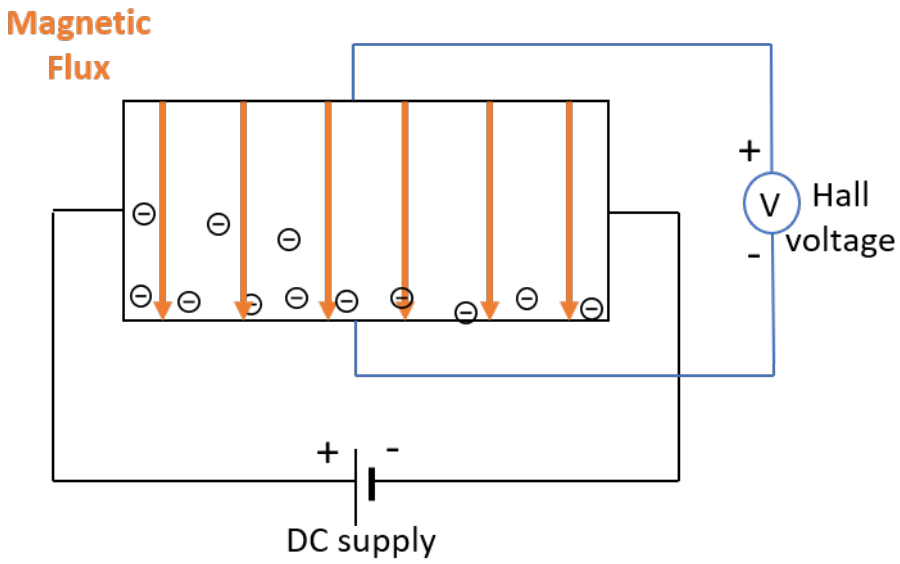


Figure 12: Hall effect working principle

Earth's Magnetic Field:
The earth's magnetic field strength varies between 25-65 µT dependent on location relative the the poles [21]. Even if the magnetic poles are wandering widely, it is sufficiently slow for ordinary compass navigation with relative high precision. A graphical illustration of the earth's magnetic field is illustrated in Figure 13 [22]. For a non-distorted local environment, the earth's magnetic flux lines can be treated as straight lines with constant direction and magnitude. Further, the magnetic flux is independent on position and orientation of the measuring tool, and can be used as a reference direction for north/south the same way the earth's gravitation can be used as a reference direction for up/down.

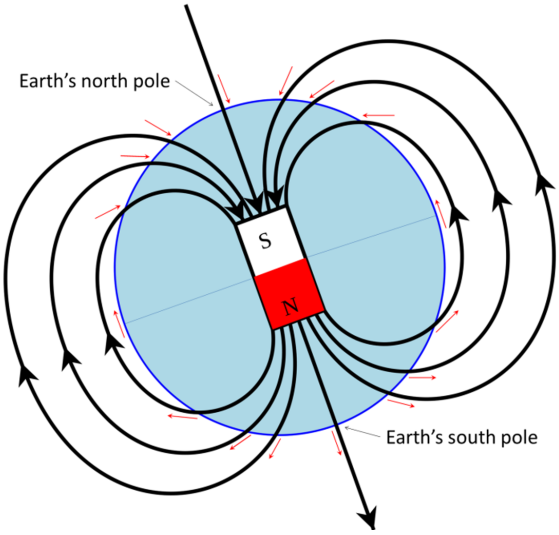


Figure 13: Earth's magnetic field [22].

Measurement Noise:
The theoretical principle for the measurement noise on the magnetometer is the same as for the accelerometer. A Kalman filter will be implemented to deal with this.

Process Noise:
As described in the intro, a miniature drilling rig build in a steel frame, with electrical equipment and rotating parts within close proximity to the magnetic sensor, will create distortion. Magnetic distortion can be divided into two categories; hard iron- and soft iron distortion. Hard iron distortion is created by objects that produces a magnetic field, which results in a offset in the measurement. An electromagnet or electrical equipment is examples of hard iron distortion sources. Soft iron distortion is created by objects that deflects or alternates the existing magnetic field, resulting in a biased stretch of the measurement that is unevenly distributed between the three measurement axis'. A magnetic material such as iron is an example on a soft iron distortion source [23].

For an undistorted magnetic sensor rotated 360 degrees around all axis, a magnitude plot of two axis, let's say x and y, will result in a circle centered around the origin (0,0). Hard iron distortion will generate a constant additive value to all sensor axis', resulting in a offset of the origin of the circle from (0,0). Soft iron distortion is dependent on position and orientation relative to the magnetic sensor, and is therefore not additive. It will affect each sensor axis differently and result in ellipsoidal shape of the xy magnitude plot. An illustration of an undistorted, hard iron- and soft iron distorted magnetic magnitude plot is presented in Figure 14.

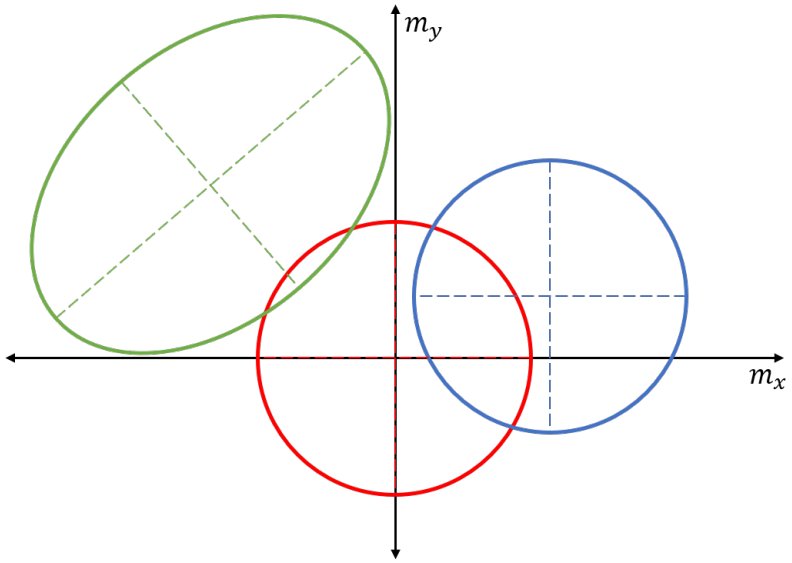


Figure 14: Magnetic distortion; red circle is undistorted, blue circle is hard iron distorted with an offset to the origin and green circle is both hard- and soft iron distorted with offset to the origin and an ellipsoidal stretch of the axis scale.

For a local magnetic distortion source, the magnetic field strength decreases cubicly by the distance r from the source [24]. That means if a magnetic distortion source is out of reach, it will not affect the magnetic sensor.

All distortion sources with constant position and orientation relative to the magnetic sensor can effectively be calibrated for. However, this is not the case in this project. While drilling, the magnetometer will move relative to several distortion sources with internally rotating distortion sources. In other words, the magnetometer will move through a magnetic environment where the flux lines is constantly changing based on the position of the sensor and based on the position of the internally rotating parts. The ideal case where the earth's magnetic field exert straight flux lines in a local non-distorted environment is far from the reality in the miniature drilling rig with the current design. To achieve the ideal case, all parts of magnetic materials and electrical equipment with close proximity to the sensor, must be replaced by a non-magnetic material like stainless steel or moved.

In the drilling industry, the main concern is to deal with internal distortion sources, where the sensor sub is usually placed 3-5 meters above the magnetic hot zones. For a miniature drilling rig, it will be quite difficult or even infeasible to move the sensor sub away from all internal distortion in the BHA. Longer BHA will result in lower directional drilling capabilities. Only a major design change can remove all internal distortion. In a already challenging problem statement delivered by DSATS, the team have not prioritized this due to budget concerns and not to limit possible design options and material selections. However, all systems that needed upgrading, like the riser system, have been replaced with non-magnetic material parts. An overview of the remaining magnetic distortion sources and the removed sources is summarized below:

- Electrical mini motor (EMM)
- PDM rotor
- Universal joints
- Drillbit
- Rig frame
- Riser frame bars (replaced)

- Bell nipple (replaced)

The magnetic process noise will be handled with the use of magnetic ranging, sensor fusion with gyroscope and magnetic distortion mapping of the drilling area in the rig.

2.3 Magnetic Ranging

In this section, the working principle of magnetic ranging will be explained.

2.3.1 Passive Ranging

Passive magnetic ranging (PMR) is the analysis of distance and direction to a static magnetic field signal source (defined by Haliburton) [25]. In the drilling industry, PMR is used to track a target well relative to a ongoing drilling well, where a metal pipe or casing sub in the target well exerts a passive magnetic field which can be tracked by a magnetic sensor. An illustration of PMR is presented in Figure 15, where the drilling operation uses a casing stub of magnetic material to carefully navigate relative and into the target well.

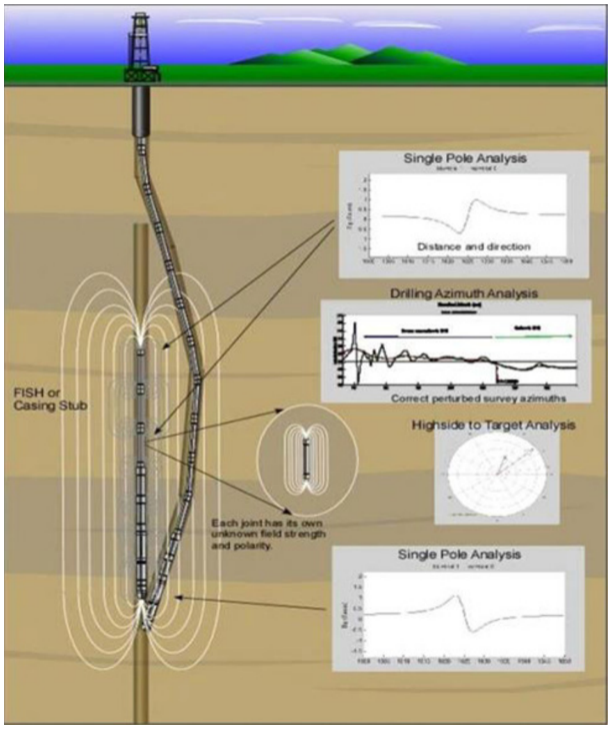


Figure 15: Passive magnetic ranging [25]

2.3.2 Active Ranging

Active magnetic ranging (AMR) is as for PMR the analysis of distance and direction to a local magnetic field, but now an active magnetic field is used. In the drilling industry, current is injected into the formation nearby a target well casing, where the resulting short circuit let current flow along the casing structure [25]. If this current is alternating, it will produce a magnetic field which can be detected by a magnetic sensor. This way a drilled well can navigate relative to a target well with relative high precision. An example is presented in Figure 16.

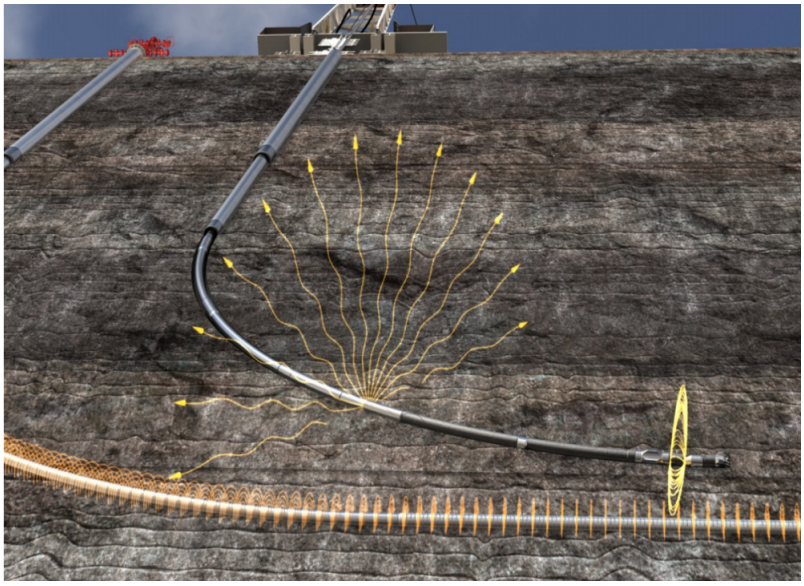


Figure 16: Active magnetic ranging [25]

2.3.3 Magnetic ranging in a miniature drilling rig

In a miniature drilling rig, all magnetic material creates a passive magnetic field and is a potential source of passive ranging. If the magnetometer is moving relative to a passive source and the source is within range of the sensor, the passive distortion will affect the measurement. If the magnitude of the passive source is much larger then the earth's magnetic field, the magnetometer will "lock in" on the passive source and treat it as a local "north".

The same principle works for an active source like a magnet or an electromagnet. Placed in a distance greater than 10cm away from the sensor, an electromagnet will have similar properties like a drain or a point source, where the flux lines is moving along the sides of the magnet, into the north pole (drain) and out from the south pole (source). Placed in an existing magnetic field consisting of the earth's magnetic field and multiple distortion sources, an electromagnet can be used to alternate the direction the existing flux lines. If aligned with the earth poles, an electromagnet can help amplify the north-component of the 3-dimensional magnetic flux vector. And if the electromagnet is strong, it can be used to drown external distortion sources. However, an electromagnet will still work as a drain/source where the flux lines will be generated from or go into the center of the magnet. This means that by using a magnet, the flux lines will not be equally straight lines in a restricted local environment, but rather lines pointing towards/away from the center of the magnet (dependent on the orientation). A magnetic sensor within the range will treat the center of the magnet as a local "north".

The main idea behind using magnetic ranging in a miniature drilling rig is to use active and passive distortion sources to help navigate the drillbit in a highly distorted magnetic environment. If the sources is aligned with the direction the team wants to drill in, they can be used to create an "artificial north" that the sensor will lock on to and navigate relative to. The magnetic ranging setup will be described in subsection 3.7.

3 Rig Setup

In this section, the rig setup for the NTNU Drillbotics 2019 rig will be reviewed. The objective is to give the reader a complete understanding of the working principle of the rig, and an overview over all actuators and sensors that will be included in the control system. This includes a review of the main mechanical parts, actuators and instrumentation units used within each subsystem. For a more detailed description of the mechanical design, check out section *6 Mechanical Design* in *Design Report NTNU - Drillbotics 2018/19* [9].

3.1 System Overview

3.1.1 Proposed Solution to Problem Statement

The proposed solution on how the team will solve the problem statement defined in subsection 1.2 can be summarized as follows:

- **Bend sub and SWD:** Directional drilling will be performed by using a bend sub in a slide while drilling (SWD) operation. A small bend is designed in the last joint of the drilling tool, giving the drillbit a small offset angle relative to the the rest of the tool stack. By rotating only the bit and at the same time apply weight on bit (WOB), the drilled well will slowly start to build angle in the direction the bend in pointing in, relative to the vertical centerline. By orienting the toolface of the bend sub, the direction the tool builds angle can be controlled. Two actuators is needed for a bend sub SWD solution to provide (1) bit rotation and (2) pipe rotation. Both is explained in the next two points.
- **DH power section:** A downhole (DH) power section is placed in the bottom hole assembly (BHA) to provide bit rotation. The team have two possible configuration of the DH power section; (1) a positive displacement motor (PDM) that runs on the hydraulic power provided by the drilling fluid (water), and (2) an electrical mini motor (EMM) that requires top side power supply (wires through drillpipe).
- **Top drive servo motor:** A servo motor is used as top drive to provide both speed and position control. When the top drive is in velocity mode, the rig can perform vertical drilling where the drillpipe (with the attached BHA stack) gets rotated. This will result in a slightly over-gauged drilled hole due to the bend sub angle. When in position mode, the top drive can be used to steer the toolface of the bend sub and thereby control what direction the tool builds angle.
- **DH sensor card:** An inertial measurement unit (IMU) is placed in the BHA to track the orientation of the drillbit relative to a world frame in real time. By using these data in a feedback loop, the position of the bit can be tracked and further controlled to follow a precalculated reference path.

3.1.2 Subsystems

The rig setup can be divided into 6 subsystems; (1) rotary system, (2) hoisting system, (3) hydraulic system, (4) riser system, (5) electrical power system and (6) control system. Location for each subsystem is presented in Figure 17. The main functionality of each subsystems can be summarized as follows:

1. **Rotary System:** Actuate rotation of drillpipe and position of directional tool.
2. **Hoisting System:** Hoist the rotary system up/down long the derrick (drill tower) to provide rate of penetration (ROP) and weight on bit (WOB).
3. **Hydraulic System:** Pump water from supply tank, through rotary system for bit rotation (dependent on configuration) and out the drillbit nozzles for hole cleaning and bit cooling.
4. **Riser system:** Transport water and return cuttings to drain and provide radial stabilization.
5. **Electrical Power System:** Distribute power to all drives, motors and sensors in the system.

6. **Control System:** Control the drilling operation autonomously based on sensor feedback and a pre-programmed operation plan. Will be reviewed separately in section 6.

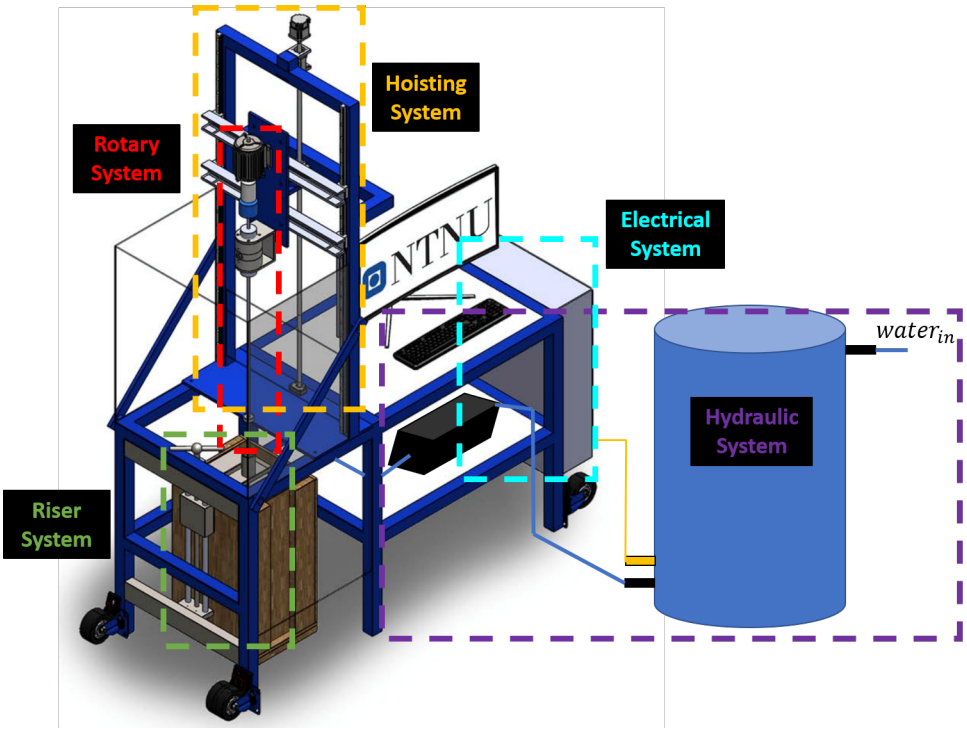


Figure 17: Rig sub systems

3.1.3 Rig Structure and Required Test Space

The rig frame is made in painted steel and have outer measurements of $1650x700x2850$ millimeters (LxWxH) with the derrick installed in upright position. The derrick however, can be folded down for easier transportation. This will give the rig frame outer dimensions of $2530x700x1240$ millimeters (LxWxH), which is convenient since the rig is being shipped to Celle, Germany. With the electrical cabinet, top drive and water tank system installed, the required test space for the rig is approximately $2000x2000x2850$ millimeters (LxWxH) with maneuverable space included. The rig frame is reused with its full design from previous years setup [8].

3.2 Rotary System

As explained above, the main function of the rotary system is to provide torque and rpm to rotate the drillpipe and directional tool. The main components in the rotary stack is presented in Figure 18. For details and function for each components, see Table 3. In the next subsections, all parts within the rotary system that directly relates to the control and steering of the drilling operation will be described in detail. This includes TD servo motor, electrical swivel and the bottom hole assembly (BHA) and drillbit.

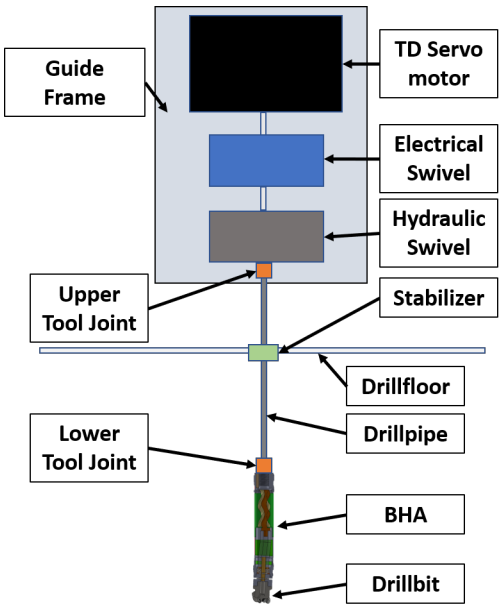


Figure 18: Rotary stack

Table 3: Rotary system main parts and function.

Part name:	**Product name:**	**Supplier:**	**Function:**
Top Drive	BCH2MM1523CA6C	Schneider Electric	Actuate torque and rpm to drillpipe.
Electrical Swivel	SNG012-12	Senring	Connects wires inside/outside the rotary stack by a slip ring to prevent wind up.
Hydraulic Swivel	Custom hydraulic swivel NTNU Drillbotics 2017	PTS workshop	Provides an upper hydraulic seal and connection point for water in.
Tool Joints	3/8"ID toll joint fitting, 2018	Vertex	Connects drillpipe to drive shaft at the hydraulic swivel and to the BHA.
Stabilizer	Custom roller bearing 3/8"ID	Abra kulelager	Support drillpipe to prevent buckling.
Drillpipe	6061 T6 aluminium 0,277"ID, 3/8"OD	ZORO	Transmit water and torque to drilling tool and communication wires up to el. swivel.
BHA	NTNU BHA 2019	PTS workshop 3D Metprint	Host downhole sensor card and downhole power section.
Drillbit	NTNU Drillbit 2019	Lyng Drilling	Drill and dig into rock sample.

3.2.1 Top Drive Servo Motor

The top drive used for this project is a Schneider Electric BCH2MM1523CA6C 1,6 kW Servo motor. The motor can support a stall torque of 7,8 Nm, a peak torque of 19,7 Nm and have a nominal speed of 2000 rpm. Based on previous tests conducted without a bend sub, the expected nominal torque while drilling is 0,3 Nm [9]. By including the increased arm a bend sub will provide, the nominal torque can be expected to be higher than this. Still, the servo motor is more than strong enough for this application. More info about the Schneider Electric Servo motor can be found in the online data sheet [26].

Previous year, the team used a regular AC motor as top drive. Since the new problem statement includes directional drilling, position control of the top drive is necessary with the current design. That means the top drive will have two main functions; (1) rotate pipe in velocity mode to drill vertically and (2) rotate pipe in position mode to steer while performing directional drilling.

The motor drive is a Lexium28 LXM28AU15M3X, which supports modbus communication protocol. The accuracy of the internal position controller is stated to be 0,1 percent (or 0,36 deg). This is more than accurate enough for this application. More info on the servo motor drive can be found in the online data sheet [27]. The Schneider Electric Servo motor and Lexium28 drive is presented in Figure 19.

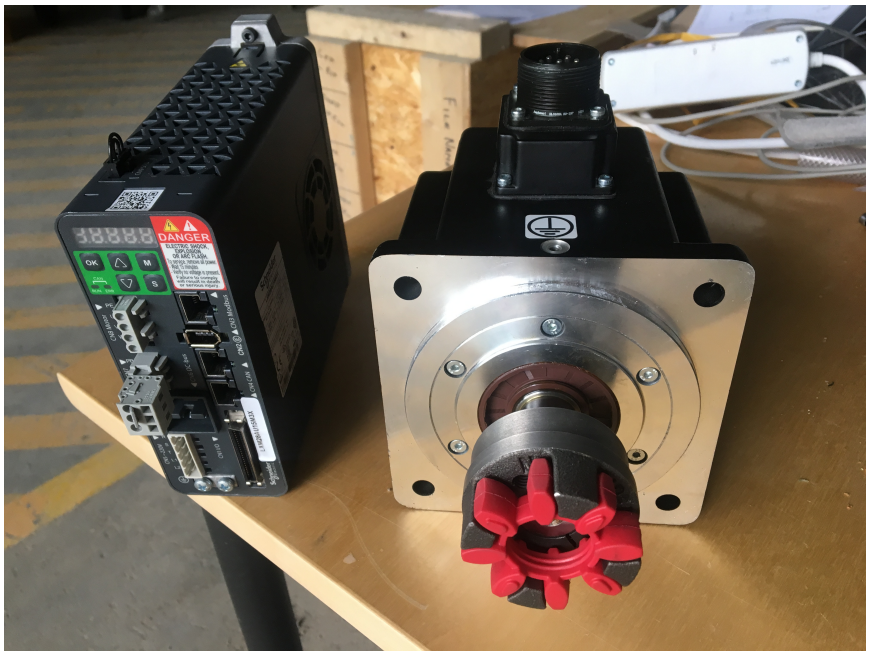


Figure 19: Schneider Electric Servo motor and Lexium motor drive.

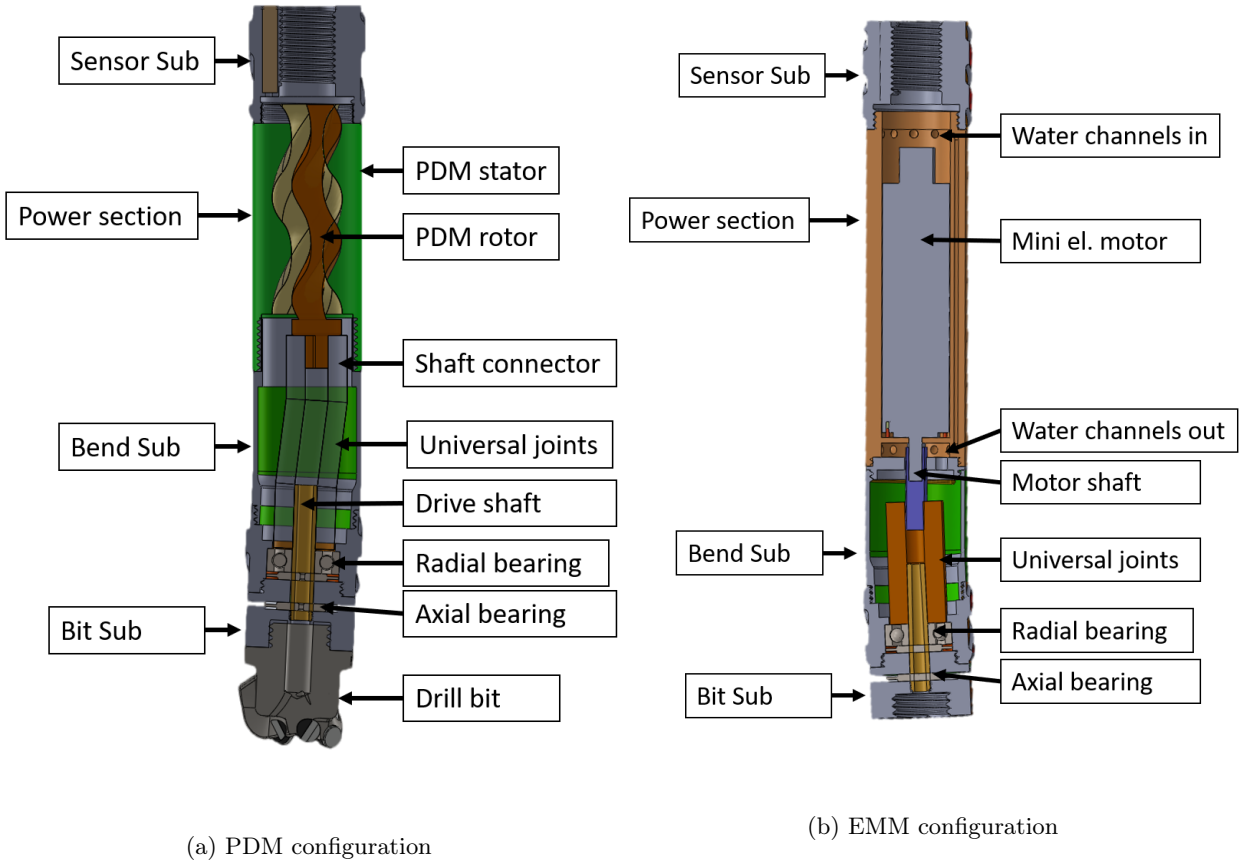
3.2.2 Electrical Swivel

The team uses a SNG012-12 series super high speed slip ring by Senring as electrical swivel. SNG012-12 is presented in Figure 20. This swivel supports 12 circuit 5A signal or power lines, and have a max speed of 5000 rpm. The slip ring is used to connect signal and power lines from the downhole electrical mini motor and sensor card within the rotary system, to power supply and computer outside the rotary system. The slip ring prevents the wires from winding up. More info on the SNG012-12 can be found in the online data sheet [28].

Figure 20: Electrical swivel - SNG012-12 slip ring by Senring [28].

3.2.3 Bottom Hole Assembly

The bottom hole assembly (BHA) is completely redesigned for this project to support directional drilling, with a downhole power section, bend sub, sensor sub and downhole sensor card. The downhole sensor card and sensor sub will described separately in section 4. The BHA comes in two configurations; (1) PDM and (2) EMM. The PDM configuration uses a positive displacement motor as power section to rotate the drillbit, while the EMM configuration uses an electrical DC motor. Both configurations is presented in Figure 21.



(a) PDM configuration

(b) EMM configuration

Figure 21: BHA configuration chart

PDM Configuration:
The PDM consist of two main components; (1) a rotor and (2) a stator. The rotor is designed with one less tooth than the stator so that it can move relative to the stator. When water under high pressure (10-30 Bar) is pumped into the top of the PDM, it forces the rotor to rotate relative to the stator to let the water pass through. The rotor is further connected to two universal joints to handle the off-center rotation from the PDM and the offset angle in the bend sub. The universal joints is connected to the bit by a drive shaft in the bend sub. A radial bearing takes up the radial force and helps stabilize the bit. An axial bearing takes up the axial force and transmit it to the BHA housing to prevent axial force on the PDM rotor.

The team have tested several configurations of the PDM throughout the semester. Both the rotor and stator have been 3D printed in stainless steel by a company that specializes on custom 3D printing, 3D Metprint AB [29]. The m2m solution have been successfully tested with bit velocities over 500 rpm. However, due to problems with leakage and wear inside the PDM where water can pass through without having to

push away the rotor, the m2m solution turned out to be unreliable and with way too low torque. The team have also tested several configurations with metal rotor and a 3D printed rubber stator in flexible plastic, locked into a steel sleeve. The idea behind this design is to machine/print the parts with negative tolerance to create a vacuum between the rotor and the stator, and by this reduce the water leakage. This configuration have also not resulted in a reliable performance. The PDM is still an important part of the NTNU Drillbotics' proposed solution to the problem statement, where the team is convinced that with the right material combination, tolerance, testing time and budget, a miniature PDM will be the best option in a SWD solution with a downhole power section. Results from the PDM performance limit testing can be found in subsubsection 7.1.1.

EMM Configuration:
The power section in the EMM configuration consist of a motor housing sub and an electrical brushless DC motor. The motor housing is 3D printed in stainless steel by 3D Metprint AB [29]. The housing features water channels in the housing walls so water can bypass the motor and reenter below the motor. A set screw prevent the motor from vertical displacement and anti-torque pins in the bottom prevents the motor from being rotated by the reaction torque while drilling. The motor shaft is (as for the PDM) connected to a universal joint and a drive shaft before its connected to the bit. A radial bearing and an axial bearing take up the radial and axial forces and transmit them to the housing structure to prevent forces on the motor shaft. The motor is glued into the housing on the top to provide a seal, and filled with high viscous grease from the bottom side to prevent water from entering the gearbox.

The electrical DC motor is a DCX19S GB 48V with GPX19HP 138:1 planetary gearhead by Maxon Motor [30]. The motor have an OD of 19mm and can deliver 0,8 Nm at 83 rpm. Based on the torque tests last semester the performance by the EMM is good enough for slow drilling, but if the WOB is increased up to a given point it will stall out. However, the motor comes with sensors to measure rpm and torque which makes it possible to lower the WOB if the motor stalls under drilling. The DC19S is controlled with an EPOS4 50/5 digital position controller drive through serial communication. The drive supports 5A with an internal voltage regulator of 10-50 VDC. More info on the DC19S and the EPOS4 dirve can be found in their respective online data sheets [30][31]. The DCX19S and the Epos drive is presented in Figure 22.

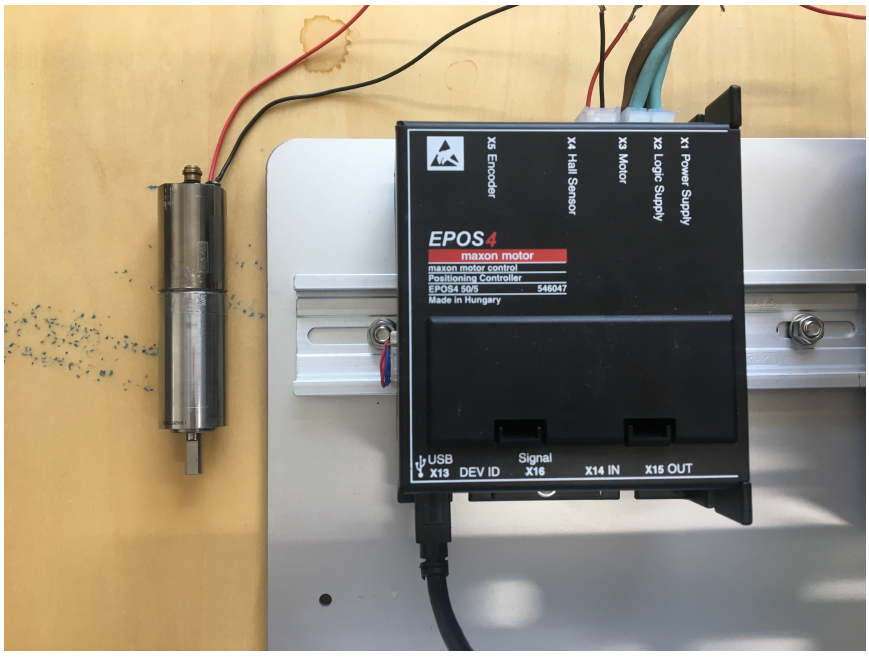


Figure 22: Maxon DCX19S motor and EPOS4 motor drive.

3.2.4 Drillbit

The team have in total 4 different drillbits; (1) NTNU 2018 bit, (2) Alibaba bit, (3) DSATS bit and (4) NTNU 2019 bit. The 2018 team designed a custom drillbit together with Lyng Drilling. This bit was designed to maximize performance in vertical drilling. With the introduction of directional drilling, a new bit design is required. The Alibaba bit was ordered (on alibaba.com) for the team to have have a cheap drillbit with the correct outer dimensions for early testing. The design of this bit is not ideal for a directional drilling operation and have only been used as a placeholder for custom bits to arrive. As stated in the competition guidelines in Appendix A, it is not required to design a custom drillbit. DSATS will provide all teams with a drillbit suitable for directional drilling. As this is an engineering competition where the mechanical design is being evaluated, and to have the upper edge on the other teams, the team have designed and build a custom drillbit together with Lyng Drilling. Lyng Drilling is NTNU Drillbotics' main sponsor, and have provided the team with feedback and guidance during the design face, and facilitation and funding of the manufacturing of the drillbit. The 2019 drillbit, is designed specifically to maximize performance in directional drilling, with more rough side cutters compared to last years bit. All bits is presented in Figure 23.

Figure 23: Drillbits. From left to right: NTNU 2018 custom bit, Alibaba bit, DSATS bit and NTNU 2019 custom bit.

3.3 Hoisting System

The main function of the hoisting system is to hoist the rotary system up and down the derrick to provide ROP and WOB. An AC motor is connected to a linear ball screw which is linearly fixed at rig floor and at the top of the derrick to allow rotation. A nut bracket provides a threaded connection with the ball screw and is fixed to a cylindrical load cell. The load cell is further connected to the guide frame which the rotary system is installed on. To allow a frictionless vertical hoisting movement, vertical roller guides is installed along the derrick beam structure. Slides on each side of the roller guides is fixed on to the guide frame. By running the hoisting motor, the guide frame with the rotary stack can be hoisted up and down along the derrick. The main components of the hosting stack is presented in Figure 24. Details and function for each component can be found in Table 4. The hoisting system have one actuator (with internal sensors) and one standalone sensor; the hoisting motor (HM) and the load cell. Both are highly relevant for the control system and will be described separately in the next subsections.

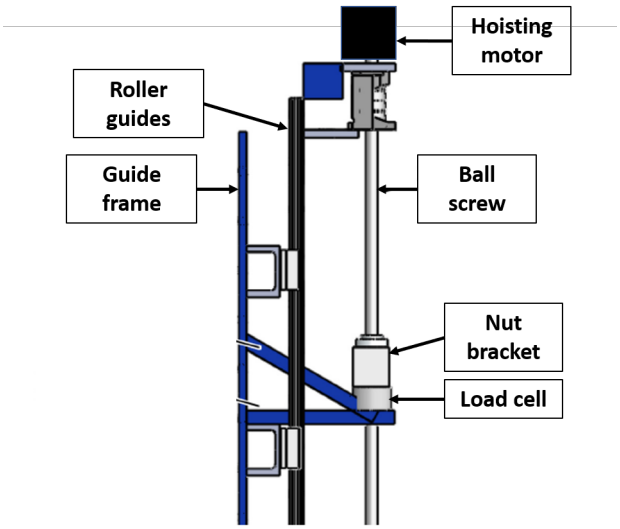


Figure 24: Hoisting stack [32]

Table 4: Hoisting system main parts and function

Part name:	**Product name:**	**Supplier:**	**Function:**
Hoisting motor	GST03-2M VBR 063C42	Lenze	Actuate vertical hoisting movement to provide ROP and WOB.
Ball screw	KGT16x5 FGR	Eichenberger	Mechanism that drives the vertical movement of the rotatry stack.
Load cell	TC4-AMP	AEP Tranducers	Measure weight on bit (WOB).
Roller guides	HG series (custom config)	HIWIN	Provide vertical guidance and low friction movement of the rotary system.

3.3.1 Hoisting Motor

The motor used to hoist the rotary system up and down along the derrick, is a IE3 m550-P three-phase AC motor geared with a GST03-2M VCK 063C32 gearbox by Lenze. The motor including gears have an maximum output torque of 45 Nm with a pre-gear rpm of 3400. The gear ration is approximately 1:9 (8.934). The hoisting motor is reused from last years setup and supports modbus communication protocol. Internal sensors can measure RPM, torque and position with high precision. The motor is further controlled with a 8400 Topline C motor drive by Lenze (E84AVTCE7512SX0). An internal PID controller in the drive controls a user defined rpm setpoint with high precision. More info on the AC motor, gearbox and motor drive can be found in their respective online data sheets [33][34][35]. The IE3 m550-p AC motor and the 8400 Topline C motor drive is presented in Figure 25.

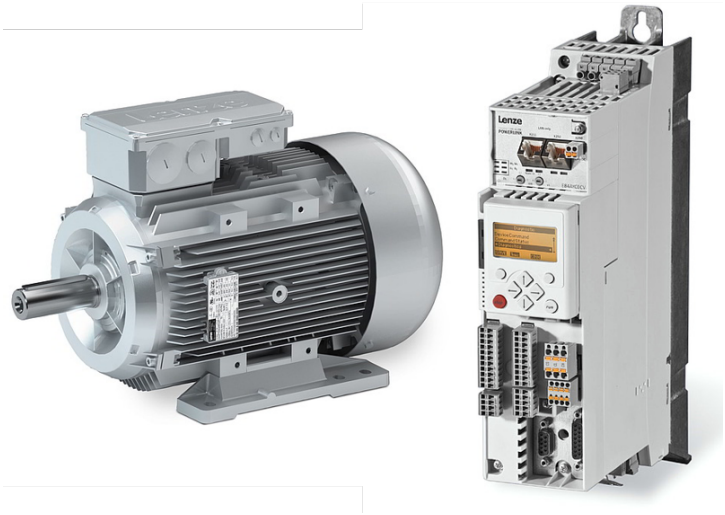


Figure 25: IE3 m550-p AC motor with GST03-2M gearbox (not illustrated) and 8400 Topline C motor drive by Lenze [33][35]

3.3.2 Load Cell

The load cell used to measure the WOB is a TC4-AMP transducer by AEP Transducer [36]. The TC4-AMP transducer was first introduced by the 2018 team, and will be reused this year. The load cell is installed on the linear ball screw on one side and to the guide frame (rotary system) on the other side. When the rotary stack tags the rock, the reacting force will be transmitted through the rotary system and up to the load cell. The force is measured by the transducer and converted into an output voltage of +-10V. The TC4-AMP can also be configured to output 4-20 mA current if the measurement signal needs to be transmitted over a long distance. Since the transmission distance in the rig is maximum 2,5 meters, the output voltage configuration will be used. The maximum load range is +-2500N. The WOB can be calculated using linear interpolation. For the measured WOB in [kg], we have:

$$WOB = V\frac{F_2 - F_1}{g(V2 - V_1)} - m_{offset} \tag{10}$$

where V is the measured voltage, F_2 and F_1 is the max and min force in [N] in range (-2500N, +2500N), g is the gravitational acceleration in [m/s^2] to convert to from [N] to [kg] and m_{offset} is a constant subtracted to cancel the measured weight of the equipment in the rotary system. The TC4-AMP is presented in Figure 26.

To read the analog measurement signal, National Instruments USB-6212 multifunction Input/Output (I/O) device is used. This device supports 16 analog I/O and 32 digital I/O, and is used on all topside instrumentation. More info on the device can be found in the online data sheets [37].

In the Drillbotics design report, several tests where conducted to analyse the load cell measurement signal. The results showed that the signal was effected by both white measurement noise and by process noise. The team have divided the process noise into into two possible sources; (1) vibrations from the drilling process itself and (2) a linear measurement offset caused by the hoisting system switches from hoisting up and down. For a PID controlled WOB state, the process noise offset is dependent on the switching frequency of the PID caused by the derivative effect in the controller. PID tuning and measurement filtering will therefore be important for a stable and safe response. A PID controller and a Kalman filter will be implemented to control the WOB.

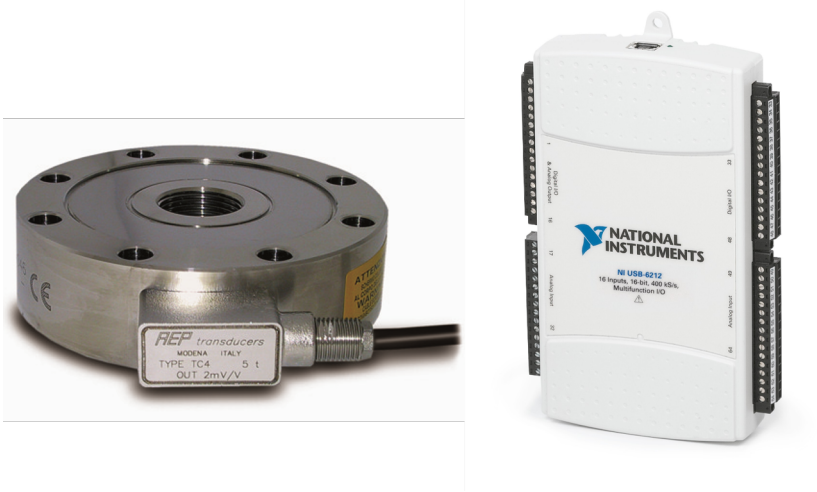


Figure 26: TC4-AMP cylindrical load cell by AEP Transducers and NI USB-6212 multifunction I/O device by National Instruments.

3.4 Hydraulic System

The main function of the hydraulic system is to circulate water trough the rotary stack to provide bit cooling, hole cleaning and drive the PDM while in PDM configuration. A supply tank is connected to a water inlet controlled by a solenoid valve at the top, and a water outlet controlled by a manual ball valve at the bottom. A tank is included to ensure sufficient water supply due to the required flow rate when in PDM mode. The tank level is measured by a pressure gauge located on the bottom of the tank. A pump and a AC motor is located in the center of the rig structure, pumping water from the supply tank and up through the rotary stack. The pump outlet pressure is measured by a pressure gauge. The water exist the rotary stack through the drillbit. An illustration of the hydraulic system is presented in Figure 27. Details and function for each component can be found in Table 5. In the next subsection, the solenoid valve, pressure gauge and the pump motor will be described in more detail.

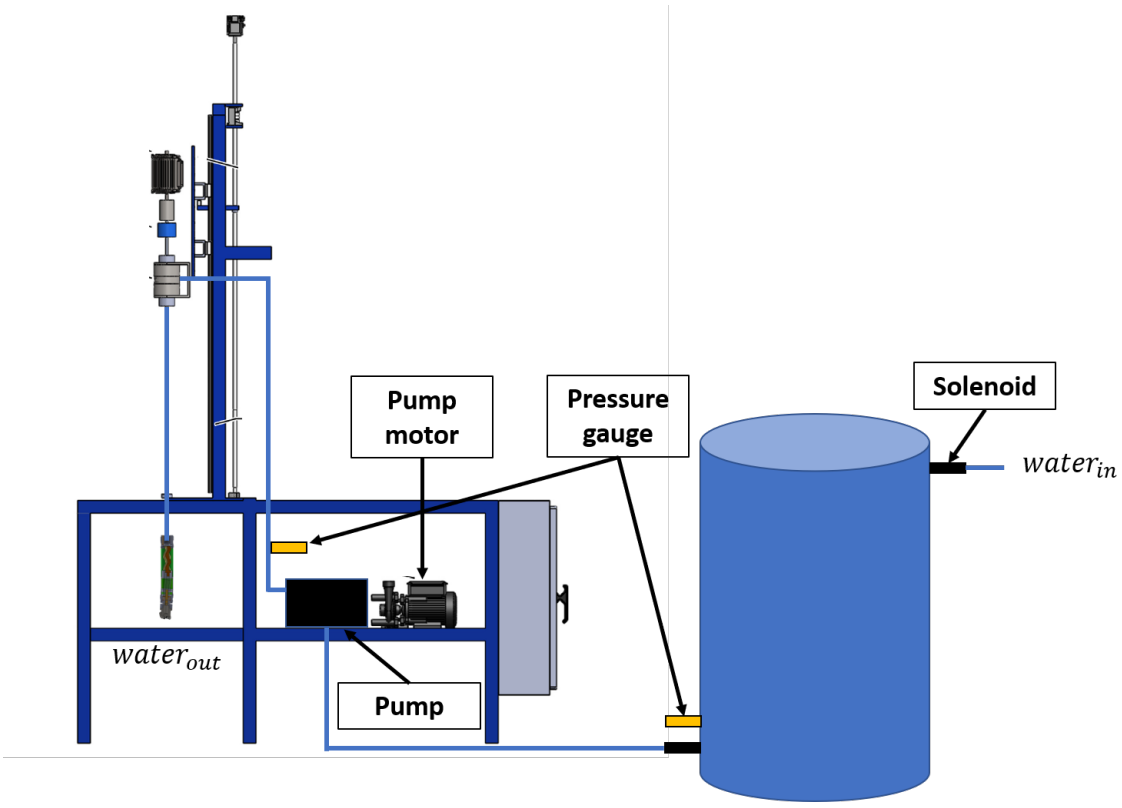


Figure 27: Hydraulic system overview

Table 5: Hydraulic system main parts and function.

Part name:	**Product name:**	**Supplier:**	**Function:**
Tank	200L water tank	Ahlsell	Supply pump with the required flow rate to run the PDM.
Solenoid valve	24V dc NC	RS Pro	Open/close the water inlet to prevent tank overflow.
Pressure gauge	PCE-28	Aplicens S.A.	Measure tank level and pump outlet pressure.
Pump	HC980A	Hawk	Pump water
Pump motor	Thurm GMBH K21R 112 M-6	VEM Motors	Actuate pumping of water

3.4.1 Solenoid Valve

The solenoid valve is a 24V NC (normally closed) valve by RS Pro [38]. The valve is controlled by a P2RF-05-S solid state relay (SSR) by Omron [39]. The relay is further controlled by a 5V control signal to switch high/low to open/close the valve. The solenoid valve combined with the tank pressure gauge is used to implement an automatic tank filling system. This will be further described in section 6. The solenoid valve and SSR is presented in Figure 28.

Figure 28: Solenoid valve from RS Pro and SSR from Omron [38][39].

3.4.2 Pressure Gauge

The pressure gauge used to measure the tank bottom pressure and pump outlet pressure, is a PCE-28 pressure transmitter by Aplicens A.S. The PCE-28 can measure 0-100 bar, converted to a 4-20 mA or 0-10 V signal. The team have configured the device in current mode for both the tank pressure and pump outlet pressure. The pressure can be calculated using linear interpolation. For the measured output pressure P in [bar], we have:

$$P = \frac{(I - I_1)(P_2 - P_1)}{I_2 - I_1} - P_{offset} \tag{11}$$

where I is the measured current in [mA], I_2 and I_1 is the min and max current (4-20mA) and P_2 and P_1 is the max and min pressure (0-100 bar). P_{offset} is subtracted if calibration is needed. More info on the PCE-28 can be found in the online datasheet [40]. The PCE-28 is presented in Figure 29.

Figure 29: PCE-28 pressure transmitter by Aplicens A.S. [40].

3.4.3 Pump Motor

The pump used is a HC980A triplex pump from HAWK runned by a Thurm GMBH K21R three-phase AC motor by VEM Motors that supports modbus communication protocol. The pump setup is reused from the 2018 setup [32].

3.5 Riser System

The main function of the riser system is to transport water in drill cuttings out from the bore hole to a drain. Water is pumped out of the drillbit with high pressure and flow rate. Water and cuttings will flow along the outer side of the BHA and up to the rock surface. The riser is installed pre-drilling on top of the rock of by using a manual hoisting system at the front of the rig. A rubber seal on the bottom of the riser provides the necessary seal between the riser and the rock for the water to continue flowing up into the riser. A bell nipple is installed in the middle of the riser where an exit flow line is installed. Water and cutting flows trough the exit line and out to a drain. An illustration of the riser system is presented in Figure 30.

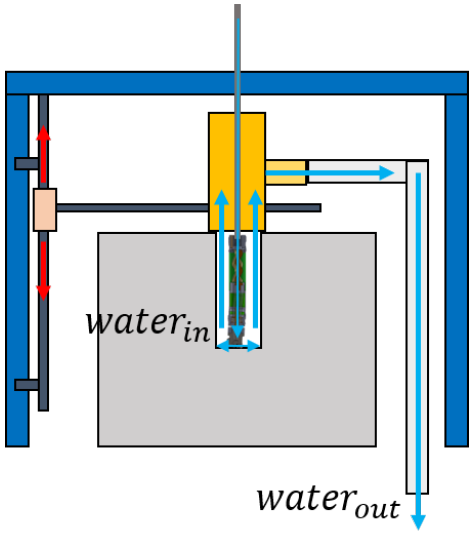


Figure 30: Riser system overview.

A natural continuation on this system is to implement a closed loop circulation system by filtering the water outlet from the riser system and feed it back to the supply tank in the hydraulic system. Due to time and physical space concerns, this has not been prioritized for this years competition. Most parts used in the riser system have been designed and build at the PTS workshop, and is reused from previous year. The riser itself however, is redesigned to fit the new BHA. As described in subsubsection 2.2.2, all parts of magnetic materials close to the drilling operation is replaced with parts in stainless steel. For the riser system, this includes the bell nipple and riser frame bars.

There are no actuators, sensors or any part with direct relevance for the control system in the riser system. Therefore, the system will not be described any further.

3.6 Electrical Power System

The main function of the electrical power system is to distribute power to all drives, motors and sensors in the system. The electrical cabinet is placed on the backside of the rig with a 3 phase 400V voltage input. From here the power is converted and distributed to all electrical equipment in the rig. All drives and and

DAQ's is placed inside the cabinet and is powered directly from the converted power input. Separate power lines for each motor and sensor is pulled from the cabinet to its designated location. An emergency safety switch is located on the side of the cabinet if the power needs to be cut manually if the control system malfunctions. Upper and lower safety switches is also installed on the guide rails in the hoisting system to prevent the system from hosting to high/low. Chief engineer Steffen Wærnes Moen has been responsible for the implementation of all electrical power distribution in the system. This is due to safety concerns where the team have non sufficient experience with electrical power systems or the correct certificates to install such systems in an official building.

3.6.1 Power Distribution

The electrical power distribution is presented in Figure 31.

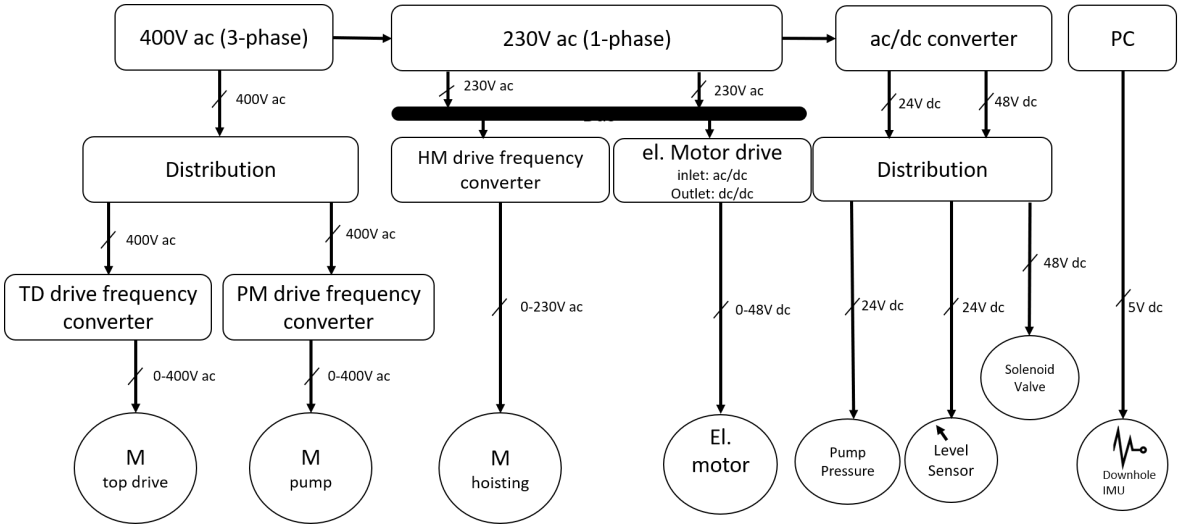


Figure 31: Power distribution chart.

3.7 Magnetic Ranging Setup

The main objective with the magnetic ranging setup is to help tracking the position of the drillbit in a highly magnetic distorted environment. As explained in subsection 2.2 and subsection 2.3, an ideal magnetic environment is not the case for a miniature drilling rig. Initial tests shows that magnetic flux lines bends towards the east direction as you move forward in the rock formation. This is due to the rig steel frame that works as a passive magnetic distortion source. By using a passive or active magnetic ranging source, the team can create an artificial "north" for a restricted local environment. The team will use an electromagnet for this purpose that will be aligned with the magnetic north flux lines to amplify the north direction. However, by using an electromagnet the properties of the local magnetic field will change where an electromagnet will more or less function as a sink. The magnetic flux will travel along the side of the magnet and into or out from the poles. The magnetic ranging setup including an illustration of the magnetic flux lines in the ideal-, true- and magnetic ranging case is presented in Figure 32.

The electromagnet used is a 7 Ohm magnet supplied with a 12V dc power supply. Based on Ohm's law, this will draw a max current of approximate 1,71A. The number of turns is unknown due to the use of a in-house magnet found in the PTS electrical equipment storage unit. This magnet have been used to lift a payload of 200Kg in its previous application. Based on initial testing, the magnetic strength is more than strong enough for this application.

The team also tried to make a homemade electromagnet with an isolated copper wire turned around a steel core. The resistance in the copper wire turned out to be too high, resulting in a weak magnetic field. With the results from the in-house magnet, the work with creating a custom electromagnet was not proceeded with. The custom and in-house electromagnet is presented in Figure 33.

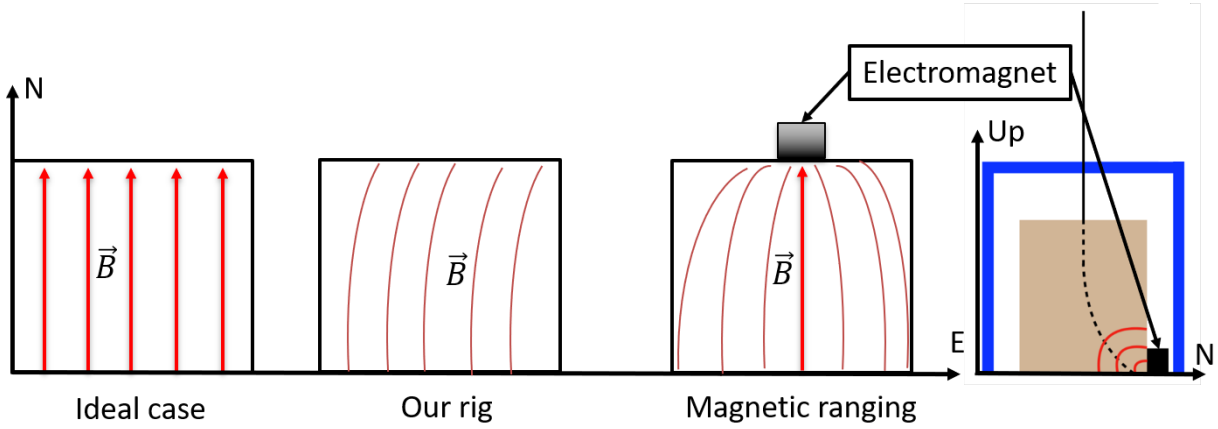


Figure 32: Active magnetic ranging setup using an electromagnet aligned with the magnetic north flux lines, and illustration of the magnetic flux lines for the ideal-, true- and magnetic ranging case.

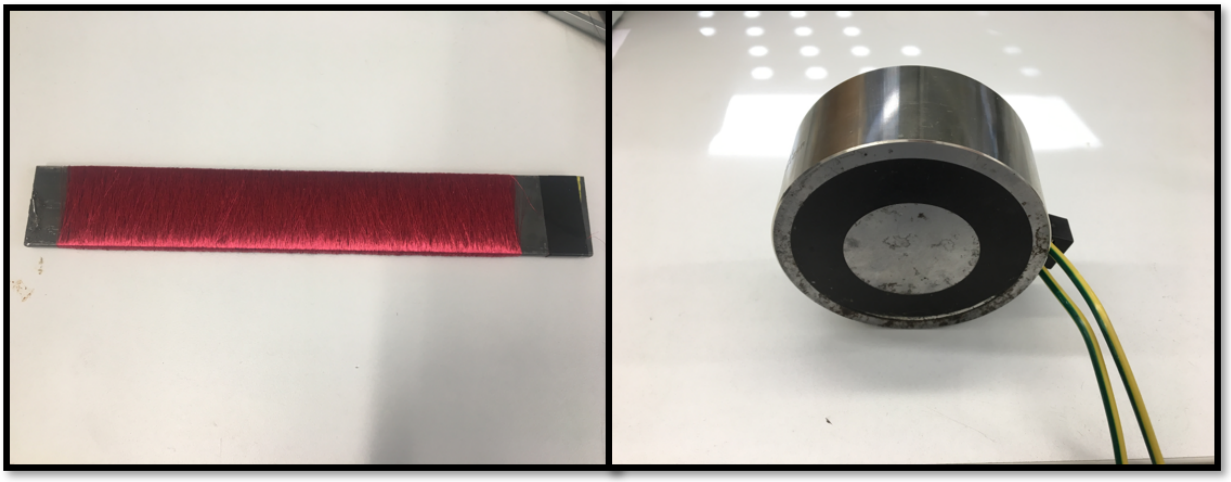


Figure 33: Custom (left) and in-house (right) electromagnet, where the in-house is used in the active magnet ranging setup.

Sink properties of a local magnetic filed can be taken advantage of in terms of steering options. This will be described further when the position/steering controllers is introduced in subsection 5.3.

4 Downhole Sensor Card

In this section, the down hole sensor design and programming will be presented. This includes design consideration, PCB-design, prototype and final product, top-side-down-hole communication flow, sensor housing in the BHA, wiring and connections, programming in simplicity studio and sensor calibration.

4.1 General Overview

4.1.1 Design Considerations

The down hole sensor design considerations can be defined as follows:

1. Ability to measure azimuth and inclination
2. Ability to measure vibrations
3. Ability to measure angular velocity
4. Preferred dimensions less then 15x25x3mm (w*l*t)
5. Waterproof housing and connectors
6. Discrete time communication with top side equipment using wires or wireless communication.

4.1.2 9-axis Motion Tracking

The team will use the ICM-20948 9-axis motion tracking device from TDK InvenSense shown in Figure 34 [41]. This inertial measurement unit (IMU) includes a 3-axis gyroscope, 3-axis accelerometer and a 3-axis compass (magnetometer). With this IMU, the team will be able to measure angular velocity, linear acceleration and magnetic field strength as 3-dimmensional vectors. By combining these vectors, the team will be able to calculate azimuth, inclination and also vibrations. Specifications for the ICM-20948 can be found in the online data sheet [42].

Figure 34: TDK InvenSense ICM-20948 9-axis IMU[41]

4.1.3 Downhole Sensor Sub

The downhole sensor sub is designed in Solidworks by the mechanical team to provide a housing for the downhole sensor card outside the main hydraulic flow line. The sub features drillpipe connection on the top, power section connection on the bottom and a sensor slot on the side. The sensor slot is 14x22x3,7mm (w,l,t) and machined with a CNC milling machine on the PTS mechanical workshop. The main design principle is

to keep the sensor card away from the main flow line due to the potential risk of leakage, but also to reduce vibrations turbulent water will induce on the card. Sensor wires will exit through a small opening at the bottom of the slot and enter the main flow line. This will be described further in the next subsection. The downhole sensor sub is presented in Figure 35.

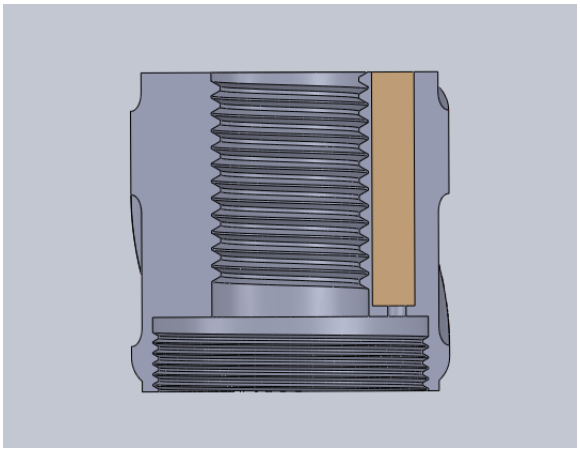


Figure 35: Downhole sensor sub.

4.1.4 Topside-Downhole-Communication

There are two main options on how a topside computer can communicate with a downhole sensor card; (1) wireless or (2) wired. In the *NTNU Drillbotics design report* it was argued that for the current mechanical design proposal, a wireless solution would be of higher risk and maybe also infeasible in terms of limited space, budget, data quality insurance and early test capabilities [9]. A wireless solution would have required a battery pack and a data transceiver (RF etc.) in an already crowded BHA. Also, there is a risk involved with sending data through steel, water, rock and air, and this would have had to be tested and proven before such a design can be proceeded with. Therefore, the team have designed the topside-downhole-communication as a wired solution.

Wires
The team will use 11 pin circular plastic shell connectors from Omnetics to connect output wires from EMM en downhole sensor card. The EMM require 2 wires for power and 4 wires for sensors. The downhole sensor card requires 2 wires for power and 2 for data communication. Hence, a total of 10 wires i required. Last year, the team used 6 pin circular plastic shell connectors which is both lack of pins and has larger OD then the new nano connectors. The connector will be sealed with a glued shrinking tube at the competition to prevent shorting due to water leakage. A comparison between new and old connectors can be found in Table 6 and Figure 36.

Table 6: Connector specification comparison.

Name:	**OD [mm]:**	**Amp rating [A]:**	**OD (wires) [mm]:**	**Pins [nr]**
2018 connector:	5,6	1,0	1,0	6
2019 connector	3,89	1,0	0,64	11

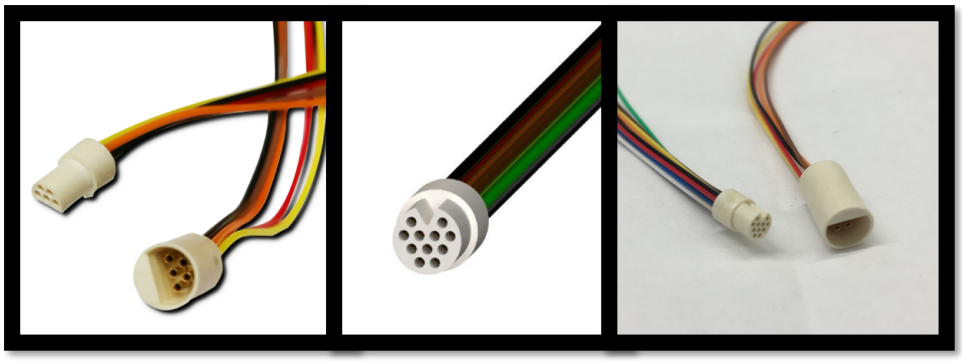


Figure 36: Comparison between this years 11 pin nano connector and last years 6 pin connector.

Wired setup

The wired setup consists of five main connection-points. An illustration of the setup is given in Figure 37. Specifications for wires and connectors can be found in Table 7.

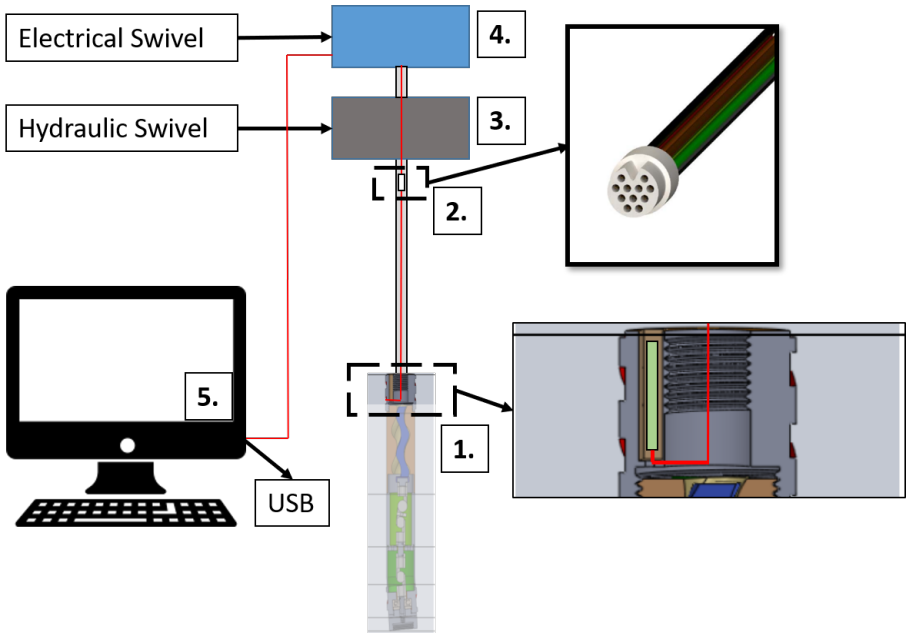


Figure 37: Wired setup for topside-downhole-communication.

1. **Sensor card:** Wires for USB is soldered directly on the sensor card and brought out through the opening at the bottom of the sensor sub. The sensor card with the soldered wires will be sealed with a layer of epoxy.
2. **Drill pipe:** An 11 pin circular plastic-shell quick connector is located at the top of the drill pipe, connecting the sensor card and EMM with the bottom of the hydraulic swivel. The quick connector was first introduced by the NTNU Drillbotics 2018 team. The idea by using a quick connector is to avoid damage to the sensor card in case of a twist off. This year, the team will drill with much lower WOB and torque, so the risk of twist-off is lower. Based on that, the team will focus on a more robust solution to ensure topside-downhole-communication uptime. The quick connectors will be sealed with a heat shrink tube. This will prevent water from entering the connector, and the seal is easy to cut off for disassembly or replacement of parts such as a new sensor card.

3. **Hydraulic swivel:** The wires passes through a bearing in the hydraulic swivel that provides both seal and ability for to rotate. The top of the wires are now on the dry side, but still inside the rotating part of the system.
4. **Electrical swivel:** To exit the rotating part of the system, the wires are connected to the electrical swivel. The slip ring within the swivel connects the wires inside with the wires outside. This way the wires will not be wound up while rotating the pipe.
5. **Computer:** Last connection point is the USB-in on the computer, connecting the computer with the output-wires from the electrical swivel.

Table 7: Wire- and connector specifications:

Wires:	
OD:	$< 1,0$mm
Amp rating:	1A
Quick connectors:	
OD:	$5,6$mm
OD (with seal):	$< 6,0$mm
Amp rating:	1A
Pins	5

4.2 PCB Design

The team has designed a printed circuit board (PCB) with good help from chief engineer Steffen Wærnes Moen. The sensor card will be built with different dimensions and configurations in terms of testing and prototyping. Prototypes will generally be larger to make space for output headers for easy programming and debugging. The final PCB design will be compressed to fit in the BHA sensor sub, and all wires for communication and power will be soldered directly on to the card. A detailed description on how the team went from prototype to final design, is presented in subsection 4.3.

4.2.1 PCB Schematics and 2D Model

The DH sensor card includes the EFM32 Gecko microcontroller (EFM32G210F128-QFN32), ICM-20948 9-axis IMU and CP2104 USB-to-UART bridge as main components. Specifications for the EFM32 Gecko and the CP2104 can be found in their online data sheets respectively [43] [44]. In Figure 38 the final PCB schematics is presented. The PCB is designed in Altium Designer. A full sized version of the schematics can be found in Appendix B. The PCB is wired with respect to USB-to-UART Input/Output (IO) communication and power in, and I2C configuration of the microcontroller and sensor chip. Details around wiring is described in the online data sheets as referred to above.

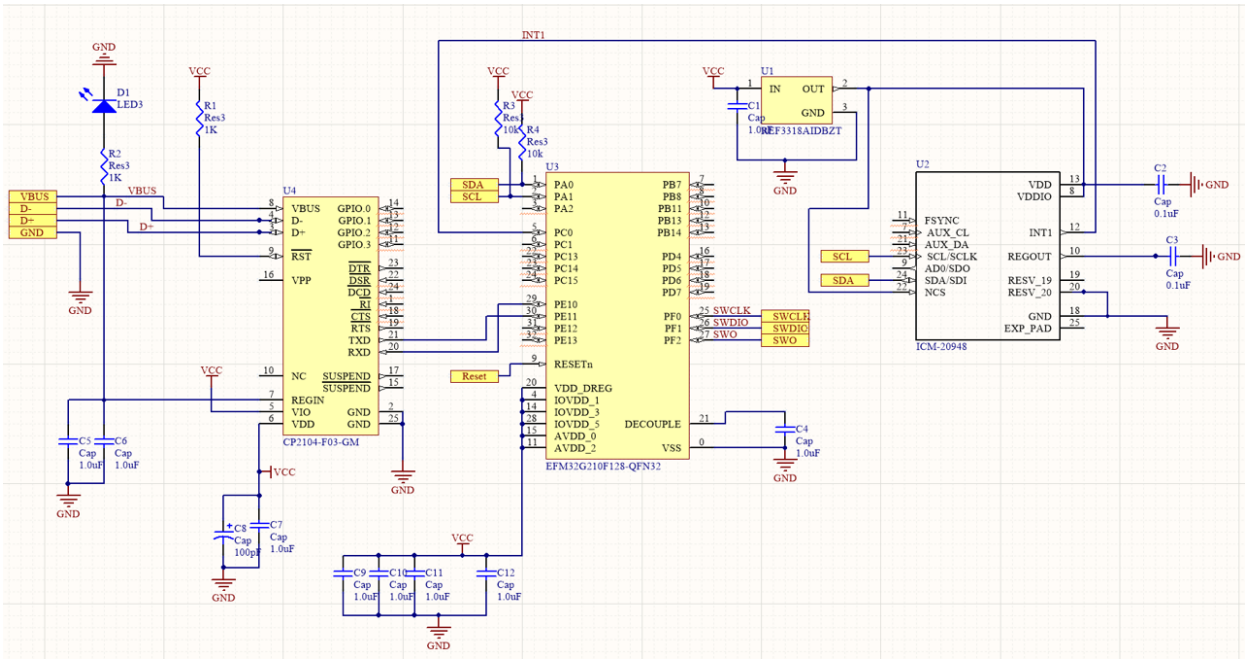


Figure 38: PCB schematics designed in Altium Designer.

In Figure 39, a 2D model of the final PCB is presented to illustrate placement of each component, USB input/output, programming pins and outer dimensions. Thickness of the PCB varies between different versions and will therefore be presented for each version in subsection 4.3. Components and specifications can be found in Table 8.

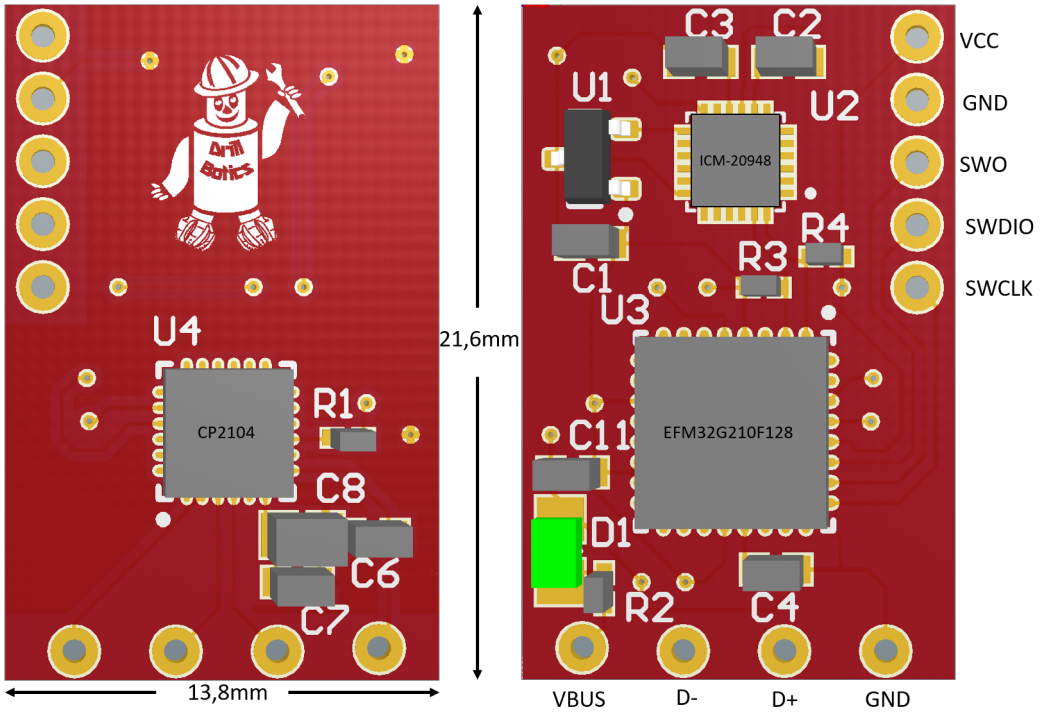


Figure 39: PCB 2D model with component placement and input/output pins and outer dimensions. Pins on bottom is for USB and pins on top/side is for programming of the microcontroller.

Table 8: PCB components and specifications

Name:	**Productname:**	**Details:**	**Supplier:**
U1	REF3318AIDBZT	3,3V-1,8V DC/DC converter	Texas Instruments
U2	ICM-20948	9-axis MotionTracking device	TDK Invensense
U3	EFM32G210F128	Microcontroller	Silicon Labs
U4	CP2104	USB-to-UART-bridge	Silicon Labs
R1-2	1kΩ	Resistor	RS components
R3-4	10kΩ	Resistor (later changed to 1kΩ)	RS components
D1	X	LED diode	RS components
C1+C4-7+C9-12	C0603C105K9RACAUTO	1,0µF Capacitor	RS components
C2-3	C0603C105K9RACAUTO	0,1µF Capacitor	RS components
C8	C0603C105K9RACAUTO	100pF Capacitor	RS components

4.2.2 Power Distribution

The PCB is power by 5V DC by a computer over a USB cable. A LED is installed to indicate voltage in. The CP2104 takes in 5V and outputs 3,3V by using an internal voltage regulator. The EFM32 microcontroller is powered by 3,3V. ICM-20948 is power by 1,8V that is converted from the 3,3V CP2104 output. A power distribution chart can be found in Figure 40.

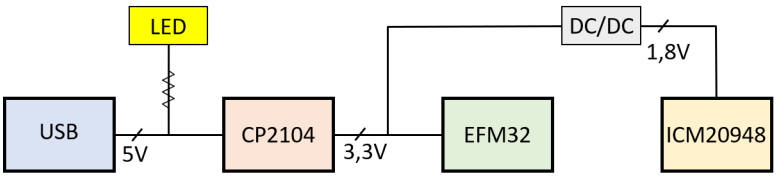


Figure 40: PCB power distribution chart.

4.2.3 Communication Flow

As explained in subsubsection 4.1.4, top-side-down-hole communication will happen over the serial communication protocol. The EFM32 microcontroller is programmed to read data from the ICM-20948 sensor chip over I2C communication and send this data over serial communication to a top-side computer in a while loop. More details around the communication between the components will be described in subsection 4.4. In Figure 41, a rough outline of the communication flow is presented.

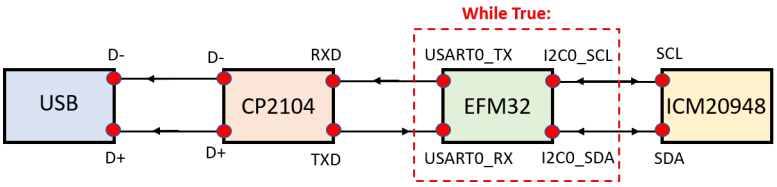


Figure 41: PCB communication flow chart.

4.3 From Prototype to Final Design

Below, the process of going from a prototype to the final sensor card design will be described. It must be stated that together with programming of the card itself, the process of designing, soldering, debugging and re-designing the card has been a time consuming part of the project. All prototypes are build by in the electronics lab at NTNU PTS using the photoresist method. The final PCB design was ordered from a custom PCB specialist, isolated, tested and ready to be soldered by the team at the electronics lab.

4.3.1 Work Flow

The work flow of designing and building the sensor card can be divided into 8 steps. The steps is presented in Figure 42 as a linear sequence, but it also includes looping back and forth between the steps as problems occur.

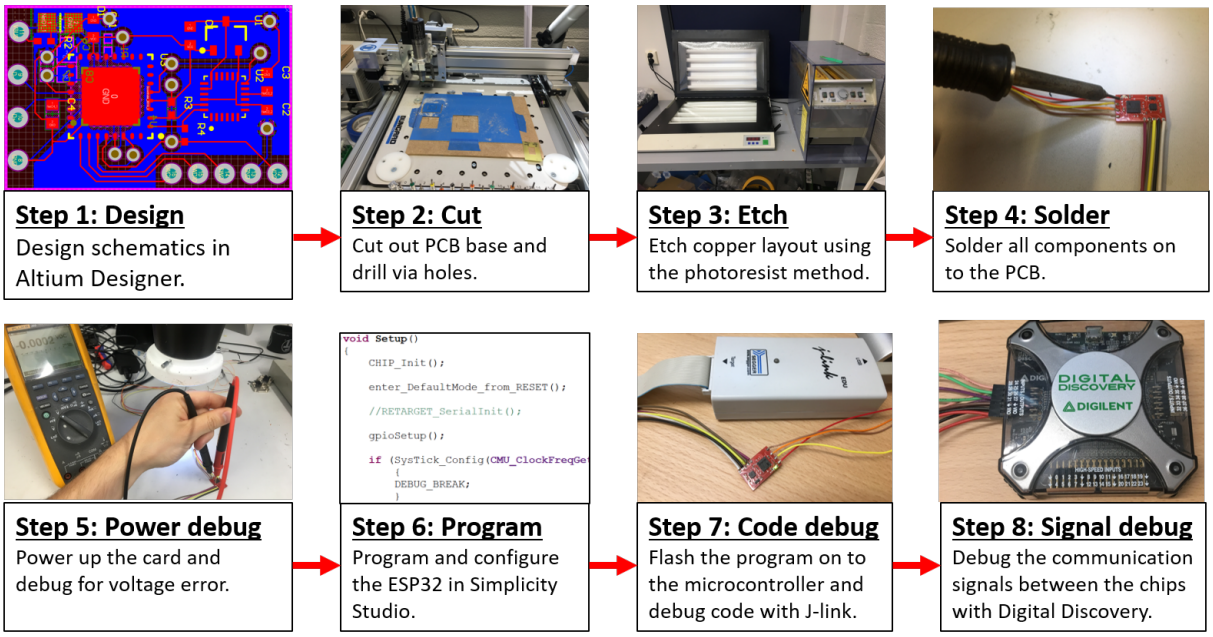


Figure 42: Work flow of designing, building, programming and debugging the sensor card.

Step 1 - Design:
The first step is to design the PCB layout in Altium Designer. This is a great software for PCB design were 2D schematics and 3D modeling are well integrated. The software supports most components were you can download the footprint from a online library. The design step includes frequent use of the datasheets for each component. Not all pins are needed for the setup used for this project. The team returns to the design step whenever a new prototype needs to be build due to results after debugging, new outer dimensions or new configuration of the chips and I/O pins.

Step 2 - Cut:
The second step is to cut out the outer dimensions for the PCB and and drill via holes using a milling/drilling machine. A file from Altium is exported with the dimensions for the cutout and coordinates for via holes. The team uses two-layer PCB sheets made of plastic with a thin copper-layer on each side.

Step 3 - Etch:
The third step is to etch the wire layout between all the components using the photoresist method [45]. The two-layer PCB sheets is covered with a light sensitive material that after exposed to strong UV light can be etched away with a acid mixture. By printing the PCB layout on to a transparent plastic sheet and attaching the sheet the PCB cutout, the light sensitive material below the print will not be exposed to light, and further be harder to etch away. This way the team can etch away all unwanted copper, leaving the copper wiring between all the components. This is a well known method used in the industry today and it's great for prototyping. However, all copper wires between the components are exposed and therefore easy to short. Because of this, the team is only using this method for prototyping were the final PCB design will be ordered from a company that specializes on custom PCB manufacturing. This will be further described in subsubsection 4.3.3.

Step 4 - Solder:
The fourth step is to solder all the components on to the card and wires through the via holes. The EFM32 and ICM-20949 needs a strong ground below the chip. For these two, hot air have been used to melt the thin bed blow the chip. Since the components are on the micro scale, a thin soldering iron have been used along with a microscope to solder the pins. The work flow process returns to this step whenever a voltage-,

ground-bug, short or a component failure is found during the debug phase.

Step 5 - Power Debug:
The fifth step revolves around debugging for voltage error, shorts and broken components. This is done in 4 steps; (1) the connectivity for each pin is tested with a millimeter. (2) the ground is debugged for shorts. (3) the sensor card is powered up with a 5V input and hopefully no components gets fried instantly. (4) the voltage is measured and compared with the datasheet for each component. If an error is found, the team will return to step 1-4 and fix the problem. Usually, the error originates from either the etching or soldering step. If the error is severe and cannot be fixed with either soldering or hot air, a new PCB must be made.

Step 6 - Program:
The sixth step is to program and configure the ESP32 in Simplicity Studio. The software is delivered by Silicon Labs and will be further described in subsubsection 4.4.1. The programming step mostly revolves around writing and reading 32-bit signals to/from the EFM32 registers. More details around programming will be described in subsection 4.4.

Step 7 - Flash/debug code:
The seventh step is to flash the code on to the microcontroller and debug the code using the J-link debugger. The debug tool is delivered by SEGGER Microcontroller and support most microcontrollers from Silicon Labs [46], including EFM32. J-link is connected to a computer by USB and to the programming pins on the EFM32, as explained in subsubsection 4.2.1. While the code is being flashed on to the microcontroller, J-link will give popup errors in simplicity studio to indicate if an error has occurred. After the code is successfully flashed on to the chip, the tool lets you insert breakpoints into the code to easily debug the code. J-link is also used to update the code on if changes is needed after testing.

Step 8 - Signal Debug:
The last step revolves around debugging the logic communication signals between the components, as presented back in the communication flow chart in Figure 41, by using the Digital Discovery debug tool from Digilent [47]. The tool is a logic signal analyze tool that has been frequently used to analyze the I2C communication between the EFM32 and the ICM-20948. By using the included software Waveforms, the team is able to analyze high/low logic voltage signal sent from the SDA-pin on EFM32 and read the response from the ICM-20948.

4.3.2 Prototypes

It is been build in total 3 prototypes, namely P1, P2 and P3 (P for prototype). In Figure 43, all prototypes is presented with both front and back side. Below, the idea behind each prototype design and results from debugging which leads to the next design, will be explain.

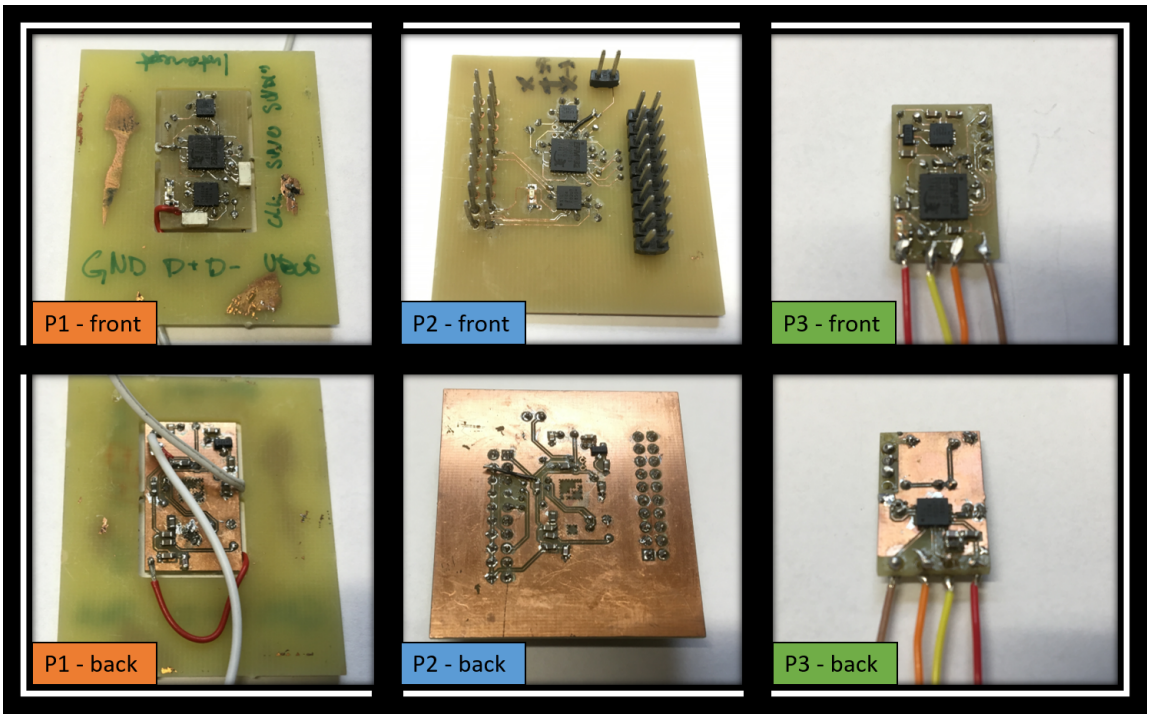


Figure 43: PCB prototypes

Prototype 1 - first try:

- **Design:** The idea behind the P1 design is to have all chip-components on the top to reduce thickness, micro-connectors for USB- and programming pins for quick connect/disconnect and outer dimensions of 16x25mm (w,l) for easy soldering.
- **Debug:** The first card has a lot of voltage errors, burning through at least three CP2104 due to shorts and voltage bridges as a result of bad etching. A total of three cards is to be started from scratch due to these problems. One ground-island was found on the backside of the card. This is temporarily fixed with a wire, but need to be fixed for next design (red wire in P1 picture). The team is able to flash the program on to the microcontroller, but the card is unreliable and hard to debug. In general, the design is lacking of possibilities to analyze the circuit, making it hard to identify errors to be a electrical or programming error. The two white wires shows an attempt two analyse the I2C communication between the EFM32 and ICM-20948. Due to this, it is decided to make next prototype larger with pin-headers for power, programming and signal analysis. The micro-connectors is trashed at this stage due to thin and fragile wires that is easy to snap.

Prototype 2 - easy analysis:

- **Design:** The P2 design is mostly the same for all main components, but pin-headers is included for easy connect/disconnect of power, programming and signal analysis. The pins on the left is for USB IO and UART signal analysis, the pins on the right is for programming and J-link program debug tool and the two pins on top is for I2C signal analysis.
- **Debug:** All voltage errors is fixed with P2 after some debugging. A new ground-island is found and temporarily fixed (black wire on P2-back picture). The program is successfully flashed on to the microcontroller, but the EFM32 have problems communicating with the ICM-20948. By using the I2C signal analysis pins, it can be seen that the the signal is constant low. The EFM32 is configured to send

logic signals over low voltage pulses. In the ICM-20948 datasheet, it is specified that the SDA/SCL pins must be configured with a 10kΩ pull-up filter to pull the voltage high after a logic low. It turns out that this resistor is too high and is reduced to 1kΩ. With this fix, the EFM32 is able to communicate with the ICM-20948 and the card can be programmed. Most of the programming is done by using the P2 due to easy debug analysis and quick connections.

Prototype 3 - compressed outer dimensions:

- **Design:** The P3 is designed to match the outer dimensions needed to fit in the BHA sensor-sub. All ground-islands is fixed and both USB and programming IO is designed to be soldered directly on the card. As explained in subsubsection 4.1.4, this is considered to be the most robust solution with the current mechanical design and the planned exit from the sensor-sub.
- **Debug:** P3 is tested in installed in the sensor-sub to confirm that the outer dimensions fit. The card has an unidentified problem with connecting to the ICM-20948 on power-up, but works consistently after some reboots. The P3 design is approved and ready to be ordered.

4.3.3 Final Design

The final PCB design is based on the P3 design and is ordered from a custom PCB specialist, JLCPCB [48]. JLCPCB offer high quality custom PCB's with isolated internal wiring, pre-connected via-holes and tin-coated footprints for each component. This makes it easier to solder and harder to short circuit. The team have ordered two versions of the final design.

Version 1 - error found:
In Figure 44, the V1 is presented with the un-soldered PCB's from JLCPCB, final sensor card with all components and wires soldered on, and finally V1 sealed in a heat-shrink-tube ready for testing.

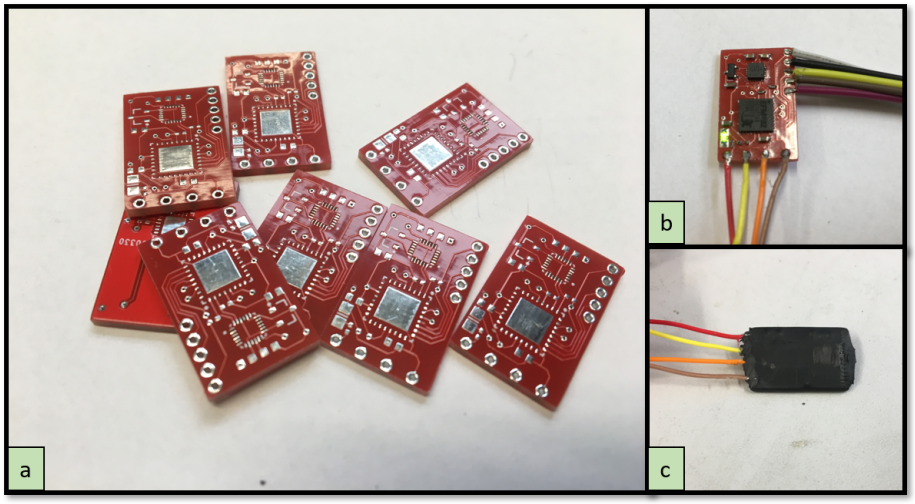


Figure 44: DH sensor card V1. a: Custom PCB's ordered from JLCPCB. b: V1 powered up with both USB and programming wires soldered on. c: V1 sealed with a heat-shrink-tube ready for testing.

V1 is used for most of the sensor calibration programming. While testing, it is discovered the reason for the communication problems at power-up. The ADO pin on ICM-20948 determines the chip address and is initially floating. This means it can either be high or low giving the ICM-20948 to potential addresses; 68 (low) or 69 (high). This pin is not connected on V1, which means that on power-up it can go either high or low. Further, on power-up a init function toggles the reset pin. This leads to a total of two power-ups and a

25% chance of guessing the right chip address. Based on this, it is decided to order a version 2 of the final design were ADO is connected to ground. This will give the ICM-20948 a constant address of 68. Also, the heat shrink tube turns out to leak at the wire inlet side. It is decided to replace the heat shrink seal with a thin epoxy layer that covers all components.

Version 2 - final design:
In Figure 45, the V2 is presented with front- and back side and with epoxy layer that provides a sufficient water seal. In this version, the Drillbotics logo and component name is printed on to the PCB surface. V2 placed in the sensor sub is presented in Figure 46.

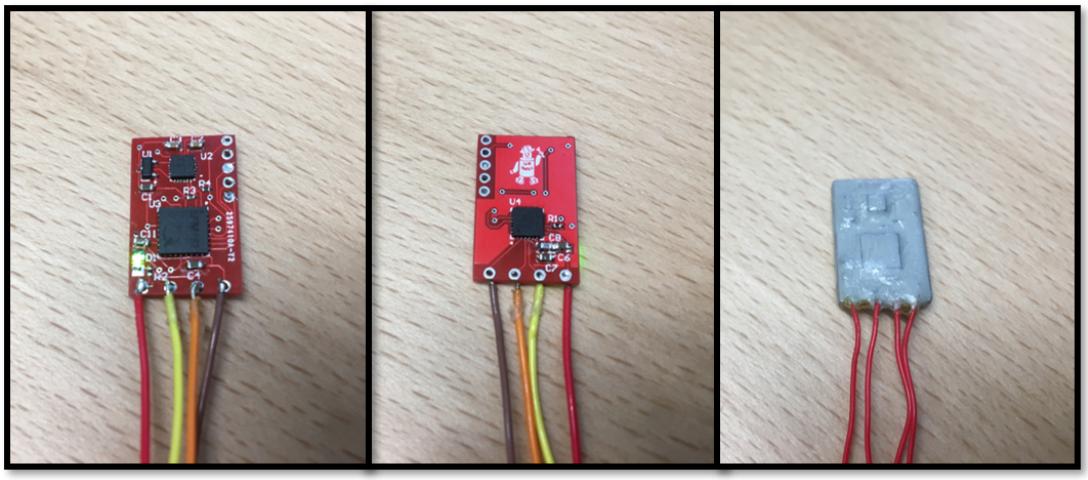


Figure 45: DH sensor card v2. Front-, back side and epoxy seal.

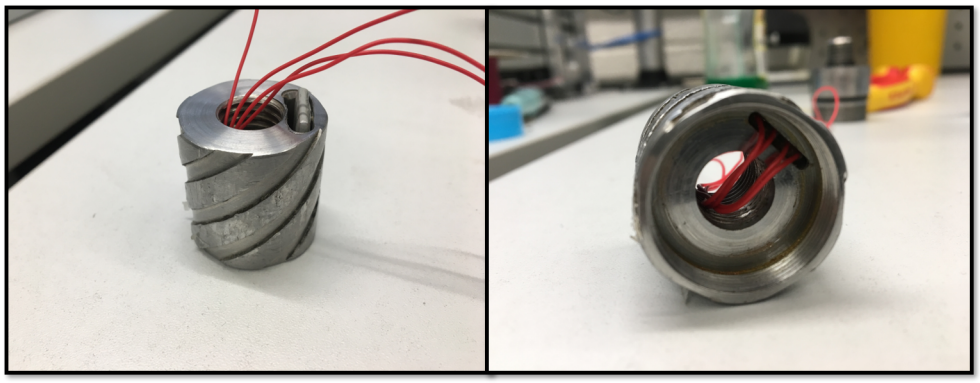


Figure 46: DH sensor card v2 placed in the DH sensor sub.

4.4 Programming of Microcontroller

4.4.1 Simplicity Studio

The software used to program and configure the microcontroller is Simplicity Studio 4 by Silicon Labs [49]. This software a great integrated development environment (IDE) made especially for Silicon Labs' 8-bit or 32-bit microcontrollers (MCU). The main functions used in Simplicity Studio for this project is coding using the IDE, MCU configurator and the debug function. The code is written in C++ and the IDE analyze the code as you write, indicating if there is a syntax error with a good help function to correct the error. The

integrated chip configurator makes it easy to create header files that support custom setup of the MCU. Simplicity Studio support debug tools like J-link by Segger that lets you flash and debug the code on the MCU. In Figure 47, the MCU-configuration used for the DH sensor card with I2C communication on pin 1-2, serial UART on pin 29-30 and flash/debug on pin 25-27 is presented.

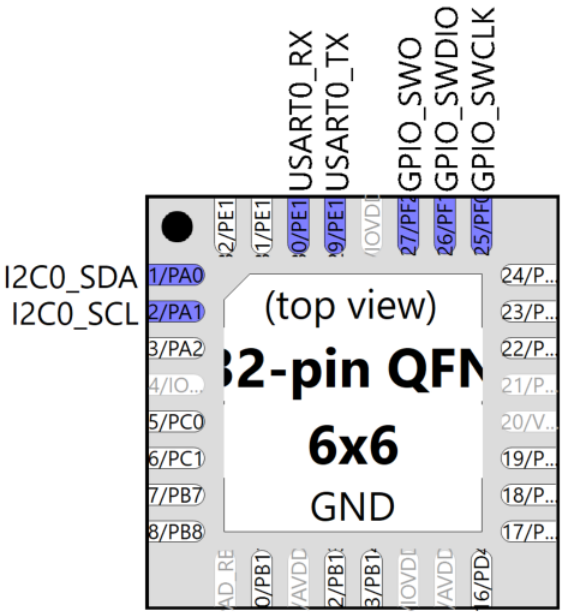


Figure 47: Built in MCU-configurator in Simplicity Studio (screenshot).

4.4.2 Communication Protocols

Before the main program is presented, the communication protocols must be introduced. The main program uses two different communication protocols for the serial communication between the chips and from the MCU to the topside computer; I2C and UART. Below, the protocols will be explained including examples of the main communication functions used in the program. These functions lays the basis for all internal and external communication that is happening in the code.

I2C:
Inter-Integrated Circuit (I^2C) is a master-slave communication protocol that uses two bidirectional open drain lines; serial data (SDA) and serial clock (SCL) (Pin 1 and 2 on EFM32) [50]. The team uses three main functions for the I2C communication between the MCU and the IMU (ICM-20948); (1) i2c-write-register, (2) i2c-read-register and i2c-read-registers (plural). These functions let you write 8-bit to one register, read 8-bit from one register and read 8-bits from a series of registers respectively. The ICM-20948 is a 8-bit chip that stores the sensor measurements in 16-bit series divided on two register; high and low. This way it is practical to use the i2c-read-registers to read both the high- and low-bits of the measurements and later combine them to a 16-bit signed integer. In Figure 48, the I2C communication functions is presented.

UART:
Universal Asynchronous Receiver/Transmitter (UART) is a communication protocol to transmit (Tx) and receive (Rx) serial data [51]. Unlike I2C, the UART transmit data asynchronously, which means there is no clock signal to sync the the transmitter with the receiver. Instead UART uses start and stop bits to indicate when the receiver shall start and stop to read a data packet. The team uses UART to send serial data from the MCU to the topside computer through the USB-to-UART bridge on the sensor card. These data is sent by a simple print function presented in Figure 49. Hence, all print functions in the code is sent over serial UART to the topside computer.

(a) I2C write function that takes chip-address, register-address and the data you want to write as input. Used to select register banks and to configure ICM-20948 on power-up.

```
void i2c_write_register(__uint8_t address, __uint8_t reg_offset, __uint8_t write_data)
{
      cmd_array[0] = reg_offset;
      data_array[0] = write_data;
      i2c_transfer(address, cmd_array, data_array, 1, 1, I2C_FLAG_WRITE_WRITE);
      return;
}
```

(b) I2C read function that takes chip-address and register-address as input, and stores output in a data-array. Used to read singular registers on the ICM-20948.

```
void i2c_read_register(uint8_t address, uint8_t reg_offset)
{
      cmd_array[0] = reg_offset;
      i2c_transfer(address, cmd_array, data_array, 1, 1, I2C_FLAG_WRITE_READ);
      return data_array[0];
}
```

(c) I2C read function that takes chip-address, register-address and N number of register that shall be read after the register-address, and stores the output in a data-array. Used to read multiple registers like accelerometer- and gyroscope high/low at the same time.

```
void i2c_read_registers(uint8_t address, uint8_t reg_offset, uint8_t count)
{
      cmd_array[0] = reg_offset;
      i2c_transfer(address, cmd_array, data_array, 1, count, I2C_FLAG_WRITE_READ);
      return;
}
```

Figure 48: I2C communication functions. See full code to review external functions imported as includes.

```
void print(const char* format, ...)
{
    char       msg[130];
    va_list    args;

    va_start(args, format);
    vsnprintf(msg, sizeof(msg), format, args); // do check return value
    va_end(args);

    while (*msg != 0)
      {
            if (*msg == '\n')
            {
                  USART_Tx(USART0, '\r');
            }
            USART_Tx(USART0, *msg++);
      }
}
```

Figure 49: UART print function. See full code to review external functions imported as includes.

4.4.3 Main Program

The main program is a C++ code flashed on the MCU that runs automatically on power-up. Below, an overview of the high level part of the main program will be presented. To review the full code, see Appendix C. The code is well commented to make it easy to understand for everyone with basic programming experience. However, it is recommended to use the datasheet for the ICM-20948 while reviewing as a reference [42].

The main program consists of two stages; (1) the setup-stage and (2) the while-loop-stage. The setup-stage includes a set of functions that runs sequentially after power-up to setup and configure the EFM32 and ICM-20948. It also includes a optional sensor calibration sequence. The while-loop-stage includes reads and prints of sensor measurement data while true. The while loop runs at a update frequency of approximate 60 Hz. A screen caption of the serial print of the IMU measurement data in Putty (a SSH terminal client) is presented in Figure 50. The following main functions runs within each stage.

Setup():

- **CHIP-Init()** - init of EFM32.
- **gpioSetup()** - configure GPIO-pins on EFM32 using the header files generated by the MCU-configurator in Simplicity Studio.
- **I2C-Init()** - init of I2C communication.
- **i2c-read-register(ICM-address, WHO-AM-I** - This function reads the WHO-AM-I-register of the ICM-20948 to confirm that there is a connection.
- **calibrateAccGyro()** - calibration function for accelerometer and gyroscope.
- **initICM20948()** - setting for ICM-20948 is written on to the chip.
- **calibrateMagnetometer()** - calibration function for magnetometer.

While(1):

- **readAccelData(accelCount)** - reads accelerometer data and stores the 3-dimensional vector it in an array called accelCount. This array is overwritten for every loop.
- **readGyroData(gyroCount)** - reads gyroscope data and stores the 3-dimensional vector in an array called gyroCount. This array is overwritten for every loop.
- **readMagData(magCount)** - reads magnetometer data and stores the 3-dimensional vector in an array called magCount. This array is overwritten for every loop.
- **print(...)** - all vector components are printed over serial for further analysis in Labview.
- **print**("c", $0x04$) - a control-word is printed to stop the transmission.

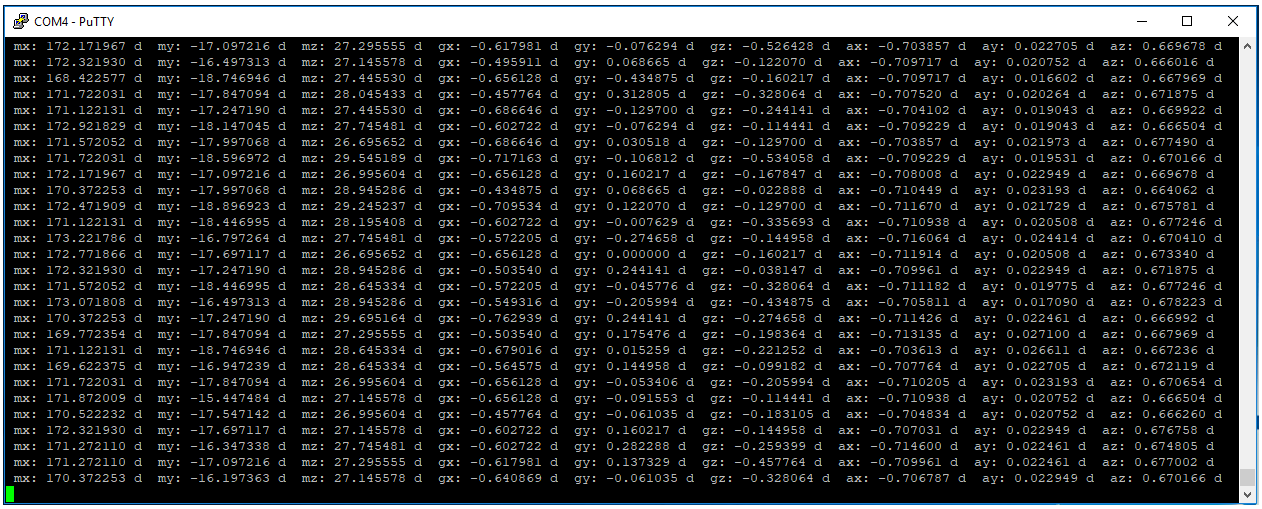


Figure 50: Serial print of IMU measurement data in Putty, a windows based SSH terminal client.

4.4.4 Sensor Calibration

In this subsection, the equations used for sensor calibration will be introduced. The calibration can be divided into equations for accelerometer, gyroscope and magnetometer. All sensors are precalibrated by the manufacturer, but thermal stresses during soldering of the components on to the PCB and local distortion requires recalibration of the sensors. The calibration principle is equal for all calibration algorithms, were a six-element linear model comprising a gain and offset in each of the three axis, will be used. For the magnetometer, an additional distortion orientation correction must be included to cope with soft iron distortion. The sensitivity scale and sample frequency is configurable for each sensor. The sensor configurations is presented in Table 9.

Table 9: Sensor configurations

	Accelerometer:	**Gyroscope:**	**Magnetometer:**
Sensitivity:	+- 2g	+- 250 dps	+- 4900 µT
Scale factor:	16384 LSB/g	131 LSB/dps	0,15 µT/LSB
Sampling rate:	1,1 kHz	1,1 kHz	100 Hz (max)
Output resolution:	16 bits	16 bits	16 bits
Low-pass filter	None	188 Hz	None

Accelerometer:
The accelerometer can measure linear acceleration in 3-dimensions and can be calibrated for relative or absolute accelerations (with or without gravity). The team will use the gravity to determine the tilt of the BHA and will therefore use an absolute acceleration calibration. The gravitational acceleration is defined as +-1g with a vertical perpendicular orientation to the tangent of the earths surface. n is defined as units of bit counts per g and a_i is the accelerometer output measurement in g for $i \in [x, y, z]$. It is assumed that the measurement noise is white with zero-mean offset. The linear calibration model becomes [52]:

$$n\mathbf{a} = n\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \mathbf{P}_a\tilde{\mathbf{a}} + \mathbf{q}_a + \epsilon_a \tag{12a}$$

$$= \begin{bmatrix} p_{ax} & 0 & 0 \\ 0 & p_{ay} & 0 \\ 0 & 0 & p_{az} \end{bmatrix} \begin{bmatrix} \tilde{a}_{xj} \\ \tilde{a}_{yj} \\ \tilde{a}_{zj} \end{bmatrix} + \begin{bmatrix} q_{ax} \\ q_{ay} \\ q_{az} \end{bmatrix} + \begin{bmatrix} \epsilon_{ax} \\ \epsilon_{ay} \\ \epsilon_{az} \end{bmatrix} \tag{12b}$$

where p_{ai} is channel gains, q_{ai} is zero-g offsets and $\tilde{a}_{ij}$ is the accelerometer state variable for axis $i \in [x, y, z]$ and orientation $j \in [+g, -g]$. ϵ_a is the white measurement noise and will be neglected for the next part. Expanding the equation into its x, y and z components gives:

$$na_x = p_{ax}\tilde{a}_{xj} + q_{ax} \tag{12c}$$

$$na_y = p_{ay}\tilde{a}_{yj} + q_{ay} \tag{12d}$$

$$na_z = p_{az}\tilde{a}_{zj} + q_{az} \tag{12e}$$

Two measurement orientations is required for each axis, and $A_i = +-g/\sqrt{3}$ is defined to give an equal positive gravitational field for each orientation and an absolute value of the 3-dim gravitational vector of 1g. $\hat{n}$ is defined as units of bit counts. $\bar{a}_{ij} = \sum \tilde{a}_{ij}[k]/k$ is defined to average the acceleration for a given axis and direction over k-samples. For j-orientation and i-axis this gives:

$$p_{ai}\bar{a}_{i+} + q_{ai} = \frac{\hat{n}}{\sqrt{3}}, \qquad p_{ai}\bar{a}_{i-} + q_{ai} = \frac{-\hat{n}}{\sqrt{3}} \tag{12f}$$

Solved for gains p_{ai} and offsets q_{ai} gives

$$p_{ax} = \frac{2\hat{n}}{\sqrt{3}(\bar{A}_{x+} - \bar{A}_{x-})}, \qquad q_{ax} = \frac{-(\bar{A}_{x+} + \bar{A}_{x-})}{\sqrt{3}(\bar{A}_{x+} - \bar{A}_{x-})} \tag{12g}$$

$$p_{ay} = \frac{2\hat{n}}{\sqrt{3}(\bar{A}_{y+} - \bar{A}_{y-})}, \qquad q_{ay} = \frac{-(\bar{A}_{y+} + \bar{A}_{y-})}{\sqrt{3}(\bar{A}_{y+} - \bar{A}_{y-})} \tag{12h}$$

$$p_{az} = \frac{2\hat{n}}{\sqrt{3}(\bar{A}_{z+} - \bar{A}_{z-})}, \qquad q_{az} = \frac{-(\bar{A}_{z+} + \bar{A}_{z-})}{\sqrt{3}(\bar{A}_{z+} - \bar{A}_{z-})} \tag{12i}$$

The accelerometer calibration will use an average of $k = 1000$ samples per orientation. The calibration will be conducted prior to installment in the BHA and saved as constants in the program. The accelerometer calibration procedure and results is presented in subsubsection 7.2.1.

Gyroscope:
The gyroscope can measure angular velocity around 3-axis. In the directional drilling operation, the team will use a slide-while-drilling method, which means the drill pipe will only be rotated to steer the toolface to correct for east/west deviation. That is, the gyroscope will mostly measure noise. Due to the problem with drift, a gyroscope is not a reliable sensor for this application. However, the gyroscope will be used as a reference to indicate twisting of the pipe while steering and as a backup for the magnetometer in high magnetic distorted areas. It is assumed that the the factory calibration for gain/scale is within an acceptable accuracy, and will therefore not be corrected for.The noise is assumed to be white with zero-mean offset. n is defined as units of bit counts per deg/s The linear calibration model becomes:

$$n\mathbf{g} = n\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \mathbf{P}_g\tilde{\mathbf{g}} + \mathbf{q}_g + \epsilon_g \tag{13a}$$

$$= \begin{bmatrix} p_{gx} & 0 & 0 \\ 0 & p_{gy} & 0 \\ 0 & 0 & p_{gz} \end{bmatrix} \begin{bmatrix} \tilde{g}_x \\ \tilde{g}_y \\ \tilde{g}_z \end{bmatrix} + \begin{bmatrix} q_{gx} \\ q_{gy} \\ q_{gz} \end{bmatrix} + \begin{bmatrix} \epsilon_{gx} \\ \epsilon_{gy} \\ \epsilon_{gz} \end{bmatrix} \tag{13b}$$

where p_{gi} is gain, q_{gi} is offset, ϵ_{gi} is white measurement noise and $\tilde{g}_i$ is the state variable. The precalibrated gain is set to $p_{gi} = 1$. For a stationary non-rotating gyro, we have:

$$n\mathbf{g} = \mathbf{q}_g + \epsilon_g \tag{13c}$$

The bias offset can be calculated by taking the average of the stationary measurement. Solved for x, y and z, the bias offset becomes:

$$q_{gx} = \frac{-\sum_{k=1}^{k} g_x[k]}{kn} \tag{13d}$$

$$q_{gy} = \frac{-\sum_{k=1}^{k} g_y[k]}{kn} \tag{13e}$$

$$q_{gz} = \frac{-\sum_{k=1}^{k} g_z[k]}{kn} \tag{13f}$$

for $k \in [0, 1000]$ samples.

(13g)

The gyroscope calibration will be conducted prior to installment in the BHA and saved as constants in the program. The calibration process and results is presented in subsubsection 7.2.2.

Magnetometer:
The magnetometer can measure magnetic field strength in 3-dimensions. The sensor uses the hall effect principle were a voltage difference is produced across an an electrical semiconductor, transverse to the current in the conductor, perpendicular the the magnetic field [53]. As explained in subsection 2.2, a MEMS magnetometer is highly sensitive to distortion. All distortion sources with constant orientation and position relative to the magnetometers reference frame can be calibrated for. This is not the case in this application, where the BHA will move relative to a steel rig with internal rotating parts and electrical equipment within close proximity to the magnetometer. However, stationary parts in the BHA can still be calibrated for. n is defined as units of bit counts per µT. Below, a linear calibration model for the magnetometer is presented.

The linear model becomes:

$$n\mathbf{m} = n\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \mathbf{R}^T\mathbf{P}_m\mathbf{R}\tilde{\mathbf{m}} + \mathbf{q}_m + \epsilon_m \tag{14a}$$

$$= \begin{bmatrix} r11 & r21 & r31 \\ r12 & r22 & r32 \\ r13 & r23 & r33 \end{bmatrix} \begin{bmatrix} p_{mx} & 0 & 0 \\ 0 & p_{my} & 0 \\ 0 & 0 & p_{mz} \end{bmatrix} \begin{bmatrix} r11 & r12 & r13 \\ r21 & r22 & r23 \\ r31 & r32 & r33 \end{bmatrix} \begin{bmatrix} \tilde{m}_x \\ \tilde{m}_y \\ \tilde{m}_z \end{bmatrix} \tag{14b}$$

$$+ \begin{bmatrix} q_{mx} \\ q_{my} \\ q_{mz} \end{bmatrix} + \begin{bmatrix} \epsilon_{mx} \\ \epsilon_{my} \\ \epsilon_{mz} \end{bmatrix} \tag{14c}$$

where p_{mi} is scale gain, q_{mi} is offset, ϵ_{mi} is white measurement noise and $\tilde{m}_i$ is the magnetometer state variable. Bias offset corrects for hard iron distortion, where the measurement area for each axis is moved and centered in the origin. A rotation matrix R is multiplied to orient and align the 3-dimensional magnitude plot with the sensor axis'. This is done so a linear scale gain can be calculated to correct for soft iron distortion. All axis are scaled to output an equal measurement range and then rotated back to its original orientation. The overall goal is for the 3-dimensional magnitude plot to go from an ellipsoidal shape, to a perfect sphere. To find the gain and offset, the sensor be rotated around all axis in a figure eight movement to find the max's and min's for each axis. The hard iron offset correction becomes:

$$q_{mx} = \frac{m_x[max] + m_x[min]}{2} \tag{14d}$$

$$q_{my} = \frac{m_y[max] + m_y[min]}{2} \tag{14e}$$

$$q_{mz} = \frac{m_z[max] + m_z[min]}{2} \tag{14f}$$

The Euler angles for the rotation around each axis is found by calculating the 2-dimensional maximum point for each axis pair corresponding the the axis that it's being rotated around. In other words. Find the max vector length of m(x,y) to calculate the rotation around z-axis. For the Euler angle (X-Y-Z, roll-pitch-yaw) using the atan2 function, we have:

$$\psi_x = atan2(\sqrt{m_y^2 + m_z^2}[max], m_y) \tag{14g}$$

$$\phi_y = atan2(\sqrt{m_z^2 + m_x^2}[max], m_z) \tag{14h}$$

$$\theta_z = atan2(\sqrt{m_x^2 + m_y^2}[max], m_x) \tag{14i}$$

For the soft iron correction, the delta for each axis and average delta for all axis combined must be found for the rotated ellipsoid. We have:

$$\Delta m_x = \frac{m_x[max] - m_x[min]}{2} \tag{14j}$$

$$\Delta m_y = \frac{m_y[max] - m_y[min]}{2} \tag{14k}$$

$$\Delta m_z = \frac{m_z[max] - m_z[min]}{2} \tag{14l}$$

$$\Delta m = \frac{\Delta m_x + \Delta m_y + \Delta m_z}{3} \tag{14m}$$

The soft iron scale gain becomes:

$$p_{mx} = \frac{\Delta m}{\Delta m_x} \tag{14n}$$

$$p_{my} = \frac{\Delta m}{\Delta m_y} \tag{14o}$$

$$p_{mz} = \frac{\Delta m}{\Delta m_z} \tag{14p}$$

The magnetometer calibration will conducted after installment in the BHA to encounter the local distortion from magnetic material parts and electrical equipment (EMM). The calibration process and results is described in subsubsection 7.2.3.

5 System and Control Design

For a drilling rig to be able to run autonomously in a safe and efficient operation, ideally, it needs to be able to observe and control the state of its process variables. Observability and controllability can be achieved by instrumenting the rig with sensors and actuators respectively. In most cases, not all process variables can be observed and controlled. However, if you can control some variables and observe some more, it is still possible to run a stable operation by using closed loop control.

In this section, the closed loop controllers will be presented. This includes a system description, simple modeling of the underlying physics and dynamics of the process, and finally the control design. The section is based on the proposed controller design presented in *Design Report - Drillbotics 2018/19* [9], were some subsections will be rendered partly or completely. However, this is needed for a complete understanding of the final controller design. All rendered subsections is stated and updated to support the final mechanical design.

5.1 System Description

First, the drilling rig system will be described from a control-perspective point of view. The system description is based on the final rig setup described in section 3.

5.1.1 System inputs and outputs

The rig has primarily 5 functions.

Rig Main Functions:

1. Hoist up/down
2. Rotate drill string
3. Pump water out of drill bit for cut removal
4. Rotate only the drill bit - new function 2019
5. Orient tool face - new function 2019

To perform the main functions listed above, the rig can control 2 servo motors, 1 AC motor and additional 1 DC motor dependent on the configuration. The servo motors can control both the position and the rpm, but not at the same time. The hoisting servo motor will only be controlled in rpm-mode, so the position input for this motor will be neglected in this review. The TD servo will switch between position- and rpm-mode. As explained in section 3, the rig has two configurations on how to rotate the bit, namely by using a positive displacement motor (PDM) or an electrical mini motor (EMM) in the down hole power section. When in PDM configuration, the rig can manipulate 4 inputs and measure 23 outputs. When in EMM configuration, the rig can manipulate additional 1 input and measure additional 2 outputs. All system inputs and outputs are given in Table 10 and Table 11.

Table 10: System inputs

Name:	**Variable:**	**Unit:**	**Mode/Config:**
Top Drive RPM	$\dot{\omega}_{td}$	rounds/min	TD RPM-Mode
Top Drive Position	ω_{td}	steps	TD Position-Mode
Hoisting Motor RPM	$\dot{\omega}_{hm}$	rounds/min	All configs
Pump Motor RPM	$\dot{\omega}_{pm}$	rounds/min	All configs
Mini El. Motor RPM	$\dot{\omega}_{mm}$	rounds/min	EMM-Config

Table 11: System outputs

Name:	**Variable:**	**Unit:**
DH 3-axis Magnetometer	$\vec{m} = [mx, my, mz]$	µT
DH 3-axis Gyroscope	$\vec{g} = [gx, gy, gz]$	deg/s
DH 3-axis Accelerometer	$\vec{a} = [ax, ay, az]$	g
Top drive torque	T_{td}	Nm
Top drive RPM	$\dot{\omega}_{td}$	rounds/min
Top drive position	ω_{td}	steps
Hoisting motor torque	T_{hm}	Nm
Hoisting motor RPM	$\dot{\omega}_{hm}$	rounds/min
Hoisting motor position	ω_{hm}	steps
Pump motor torque	T_{pm}	Nm
Pump motor RPM	$\dot{\omega}_{pm}$	rounds/min
Pump outlet pressure	P_{pm}	Bar
Water Tank pressure	P_{wt}	Bar
WOB	P_{wob}	Kg
El. mini motor torque	T_{emm}	Nm
El. mini motor rpm	$\dot{\omega}_{emm}$	rounds/min

By using the listed inputs and outputs, the team will design a control system based on competition guidelines described in subsection 1.2. First, the control objective needs to be defined.

5.1.2 Control Objective

The control objective for the Drillbotics 2019 competition is to drill a directional well, kicking off from a 4" vertically drilled pilot hole, drill minimum 2 3/8" horizontally along the magnetic north center line and 20" vertically (if you don't exit before the rock bottom). Both topside and down hole sensors must be included in the main control loop and the rig must be controlled autonomously. Also, the competition time limit is set to 3 hours. That is, the system needs to track and control the position of the tool face in 3-dimensions, control torque and WOB for safety consideration, and maintain the needed ROP-level to finish within the time limit or as fast as possible. The control objective can be summarized as follows:

Summarized Control Objective:

- Control WOB
- Control Torque
- Control ROP
- Control position

Since torque, WOB and ROP are closely related, they can in theory all be used as control parameter to control the progress of the drilling operation. However, the team will only control the WOB and limit the torque and ROP during the drilling process. This way way the team will be able to control three important parameters by only controlling one. For this to work, the team will have to design a WOB setpoint selector. The WOB setpoint selector will be further described in subsubsection 5.2.4. A comparison of the different control parameter options is described in subsubsection 5.2.1.

For position control, the position of the drill bit will be used as the control parameter.

Hence, the team will use the the following 2 controllers.

- WOB controller
- Position controller

The controllers will be further described in subsection 5.2 and subsection 5.3.

5.1.3 System Vectors

Based on the control objective and by using Table 10 and Table 11, the system states, inputs, outputs and set points can be written on vector form: We have

$$\mathbf{x} = \begin{bmatrix} x_b^I \\ y_b^I \\ z_b^I \\ P_{wob} \end{bmatrix}, \mathbf{u} = \begin{bmatrix} \dot{\omega}_{td} \\ \omega_{td} \\ \dot{\omega}_{hm} \\ \dot{\omega}_{pm} \\ \dot{\omega}_{emm} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \vec{m} \\ \vec{g} \\ \vec{a} \\ T_{td} \\ \dot{\omega}_{td} \\ \omega_{td} \\ T_{hm} \\ \dot{\omega}_{hm} \\ \omega_{hm} \\ T_{pm} \\ \dot{\omega}_{pm} \\ P_{pm} \\ P_{wt} \\ P_{wob} \\ T_{emm} \\ \dot{\omega}_{emm} \end{bmatrix}, \mathbf{r} = \begin{bmatrix} x_{ref} \\ y_{ref} \\ zref \\ P_{wob,sp} \end{bmatrix} \tag{15}$$

where a description for each variable or measurement can be found in Table 10 and Table 11. The state vector $\mathbf{x}$ consists of the variables the team wants to control in a closed loop, where $[x_b^I, y_b^I, z_b^I]$ is the drill bit position in the inertial frame, and P_{wob} is the WOB measured by the load cell. The input vector $\mathbf{u}$ will be the only variables our controllers can manipulate to change output measurements or to eliminate setpoint errors. The output vector $\mathbf{y}$ includes all measured states in the system. Some of the states will be used in a closed loop, others will have boolean logic's making the control system switch WOB setpoint, operation state or terminate if special events occur. The working principle of the state machine will be further described in subsection 6.6, were the implementation of the control system in Labview is explained.

5.2 WOB Control Design

5.2.1 Design Considerations

As explained before, the team can choose different control parameters to control the speed and safety of the drilling operation. The design considerations can be summarized as follows:

Design Consideration:

- Drill pipe twist off limit [Nm]
- Drill pipe buckling limit [Kg]
- BHA hydraulic pressure limit [Bar]
- PDM torque limit [Nm]
- EMM torque limit [Nm]
- HM torque limit [Nm]
- TD torque limit [Nm]

- Competition time limit [hours]

Below is a comparison between the three option the team have considered to use as the main control mode.

Option 1 - ROP Mode
The first option is to drill with constant ROP. The advantage with this is that the team will know when the operation is finished and it will be easier to track position. The input to the system will be the HM rpm, $\dot{\omega}_{hm}$, were the rotation of the bit/pipe will constant. However, this is not a safe solution because the controller will not care about the WOB and torque, risking the drill pipe to buckle or twist off in order to reach the ROP setpoint. You can always implement a setpoint selector and limit WOB and torque, but this will not be efficient a most probably cause in high frequent setpoint shifting. This option was never really considered.

Option 2 - Torque Mode
The second option is to use the drill bit torque or drill pipe torque as main control parameter. This is the safest option for motor-safety were the set point can be set based on the specs of all the motors that provide torque to either the bit or the drill pipe (TD, PDM, EMM), or to the twist of limit of the drill pipe. The input to the system will be HM rpm. This option does not directly control the WOB in terms of the buckling limit, but since WOB and torque are closely related it can be indirectly controlled as long as the bit/pipe is rotating. Last year, the NTNU team designed a torque controller to be used if a given torque limit was exceeded [11]. However, based on several tests the ROP fell significantly when in torque mode and the controller was basically discarded during the competition run by choosing a high torque limit that would never be reached. Based on these results and the limitations of measuring the PDM torque in directional drilling mode, torque will not be used as the main control mode.

Option 3 - WOB Mode
The third option is to use WOB as the main control parameter to control the speed and safety of the drilling process. The option will be using HM rpm as input and bit/pipe rpm will be set to a constant value. The advantage with this option is that the system can directly control the WOB in terms of buckling, and indirectly control torque which is also dependent on bit/pipe rpm and rock friction. Since the rock sample will be of homogeneous sandstone with known hardness, the team will have good control of the theoretical torque for a given WOB and bit/pipe rpm SP. Last year, the NTNU team used a WOB controller as the main control mode during the drilling phase, and in terms of ROP this was the best controller. If one consider controlling torque for mechanical safety reasons and ROP for the competition time limit, the team has two options for this to work; (1) Design a setpoint selector or (2) switch between different controllers like last years team. Due to this years changes in the control objective with both vertical and directional drilling, and the mechanical and instrumentation changes done on the rig, the team will use a WOB setpoint selector. The team believes the limited possibilities of measuring the drill bit torque will make it difficult and inefficient to switch between different control modes, and that a WOB setpoint selector will be both safer and simpler. As explained before, the details around the WOB setpoint selector will be explained in subsubsection 5.2.4. To conclude, the team will use WOB as the main control mode during both the vertical and directional drilling phase.

5.2.2 WOB Model

Before the WOB controller can be designed, the physics behind the WOB measurement needs to be modeled. As stated in subsection 1.3, complex system modeling is not a part of the scope for this thesis and the team mainly will use the cohen-coon tuning method and step response tuning to find the controller gains. However, a simple model will still help to find a starting point and to get a understanding what reaction time the controller needs to have.

The WOB must be modeled in two different cases; (1) before the bit tags the rock and (2) when the bit tags the rock.

WOB model 1 - before rock tag

Assume the BHA is being hoisted down by the HM and the tool have reached terminal speed. HM rpm and the ball screw pitch is known. The load cell is calibrated to be zero when in hoist-down-mode. The bit is not touching the rock and the sum of forces from the bit perspective is equal to zero. A graphical illustration can be found in Figure 51.

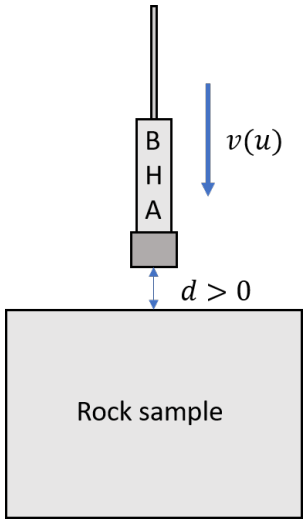


Figure 51: WOB model 1: The bit is not touching the rock and the WOB is therefore equal to zero.

The linear velocity of the bit can be expressed as follows:

$$\dot{x}(t) = pk_r k_t u(t) \tag{16a}$$

where $\dot{x}$ is the bit linear velocity, u is the HM rpm, p is the ball screw pitch, k_r is the gear ratio between the HM and the ball screw and k_t is a simple constant to convert from [rpm] to [m/s].

(16b)

WOB model 2 - after rock tag

After the bit has tagged, the WOB can be modeled in three stages; (1) deceleration, (2) equilibrium and (3) acceleration. At the first stage, the rotary system has a velocity in the vertical down direction and is being decelerated by the normal force from the rock. The second stage can be divided in two; (2.1) when the velocity is zero and the gravity force is divided between the normal force and hoisting force, and (2.2) where the velocity is zero and the bit rests on the rock. That means the WOB is equal to the gravity force acting on the equipment in the rotary system. In the third stages, the hosting force goes from being a reaction force to the gravity, to actively push down the rotary system in the downward direction. The sum of forces will still be equal to zero.

All the three stages can be written out mathematically as follows:

$$(1.) \sum F = G - F_{normal} - F_{hoist} = -ma \tag{17a}$$

$$(2.1) \sum F = G - F_{normal} - F_{hoist} = 0 \tag{17b}$$

$$(2.2) \sum F = G - F_{normal} = 0 \tag{17c}$$

$$(3.) \sum F = G - F_{normal} + F_{hoist} = 0 \tag{17d}$$

where

$$F_{wob} = F_{normal} \tag{17e}$$

The team is planning to use a WOB SP in the range of $WOB_{SP} = [10, 30]kg$. With a rotary-system-weight of X kg means that the rig will only operate in stage 1 and 2.1 in the WOB development model. After the system has settled it will manly be operating in stage 2.1, loading and unloading the gravity force of from the rotary system on to the bit.

Even if the theoretical model seems simple, there are a couple of difficulties that can cause trouble. The time the hoisting force goes from being a reaction force to the gravity to actively push the bit down, happens over a distance of millimeters. Also, the rotary system goes from having kinetic energy to being decelerated over the same small unknown distance. It would be simpler to model this distance relative to the hoisting position towards a static surface, but this is not the case in our system where the distance changes as drill cuttings gets removed. At the risk of over-complicating the model, the team will move forward with a more practical approach.

However, by creating a simple model of the acceleration, the team can get an indication of how fast the WOB development grows. A graphical illustration of of the model can be found in Figure 52.

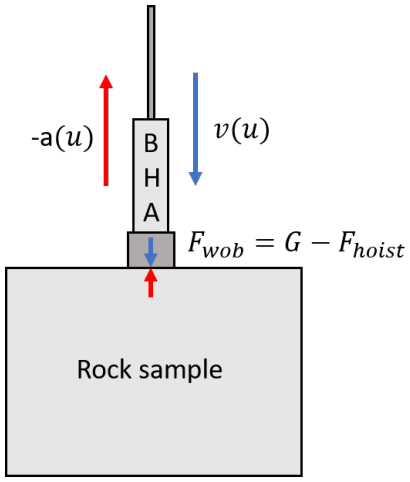


Figure 52: WOB model 2: The bit has tagged the rock and the WOB is increasing as a function of the input.

For a given rpm input to the HM, the rotary system will at the moment it touches the rock have a velocity and deceleration. We have:

$$-a\ddot{x} + b\dot{x} = cu \tag{18a}$$

where a, b, c are constants. To simplify the model, we chose $a = b = c = 1$, and get

$$-\ddot{x} + \dot{x} = u \tag{18b}$$

Since the acceleration is directly related to the force, we set this as our state variable. The input u will be modeled as a step response in the frequency domain. We get

$$\hat{x} = \ddot{x} \tag{18c}$$

$$\frac{\hat{x}}{s} = \dot{x} \tag{18d}$$

Insert into equation

$$-\hat{x} + \frac{\hat{x}}{s} = \frac{u}{s} \tag{18e}$$

Sorting for $\hat{x}$

$$\hat{x} = -u\frac{1}{s-1} \tag{18f}$$

Finally we take the inverse Laplace transformation and get the step response in the time domain:

$$a(t) = x(t) = -ue^t \tag{18g}$$

Even if the model is very simple, it still shows that the WOB will grow exponentially after the bit tags the rock. This means the WOB controller will have to perform quickly to prevent buckling. Due to the complexity of the system and the rapidly changing model conditions, the WOB will not be modeled any further. The team will do practical experiments on the rig and model the WOB development by performing step responses.

5.2.3 WOB Controller

The working principle of the WOB controller is to eliminate the error between the chosen WOB SP and the measured WOB from the load cell. This is accomplished by feeding the error into a PID controller which calculates a new SP for the hoisting motor RPM. The internal PID in the hoisting motor drive controls the HM rpm SP. The closed loop is completed with the load cell taking a new measurement, feeding a filtered WOB back to the error calculation and into the controller. An overview of the WOB control scheme is presented in Figure 53.

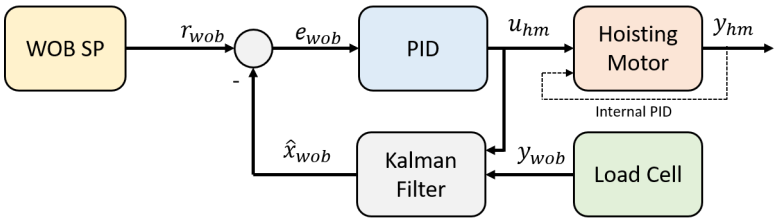


Figure 53: WOB control scheme

PID:
By following the theory explained in subsubsection 2.1.1, the WOB PID controller can by written out mathematically. For a general PID controller we have

$$u(t) = e(t)[k_p + k_d\frac{d}{dt} + k_i\int dt] \tag{19a}$$

where

$$k_d = k_p T_d \tag{19b}$$

$$k_i = \frac{k_p}{T_i} \tag{19c}$$

For this system the PID output equal to the HM rpm input becomes

$$u_{hm}(t) = k_p e_{wob}(t) + k_i\frac{d(e_{wob}(t)}{dt} + k_d\int e_{wob}(t)dt \tag{19d}$$

where

$$e_{wob}(t) = r_{wob} - \hat{x}_{wob}(t) \tag{19e}$$

where r_{wob} is the WOB setpoint and $\hat{x}_{wob}$ is the Kalman filtered WOB state.

Kalman filter:
The team will use a Kalman filter on the measurement feedback to filter out unwanted measurement- and process noise. The theory behind the Kalman filter is explained subsubsection 2.1.2. Due to the complexity of the WOB model and inaccuracies in the HM rpm readings, the WOB will be modeled as a stationary process. For a "un-modeled" system, the Kalman filter response will resemble to the response from a regular low-pass filter, where the tuning of Q_k and R_k will indirectly determine the cut off frequency.

For a general discrete time system with white noise disturbance, we have

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{20a}$$
$$y_k = Cx_k + Du_k + v_k \tag{20b}$$

with the process- and measurement noise covariance matrices

$$Q_k = E[w_k w_k^T] \tag{20c}$$
$$R_k = E[v_k v_k^T] \tag{20d}$$

With one state, the system matrices for the stationary system will be defined as:

$$A = C = 1 \qquad B = D = 0 \tag{21}$$

This means the Kalman filter will fully rely on the measurement signal to estimate the next state, resulting in a time-delay between the true and filtered state which dependent on how "fast" the filter is tuned. To avoid a critical overshoot of the true WOB state, the PID controller must be tuned accordingly "slow" with a over-dampened response to cope with this time-delay. The measurement noise variance will be calculated over 1000 samples. The process noise will be tuned to achieve the desired response. Results and tuning process is presented in subsection 7.4.

5.2.4 WOB Setpoint Selector

As described in subsubsection 5.1.2, WOB, ROP and torque are closely related and dependent on each other and the physical properties of the rock formation. Based on the arguments presented in subsubsection 5.2.1, a single WOB controller can be used to control all 3 states (to some extent) by implementing a WOB setpoint selector. The main objective with the SP selector is to maximize performance of the weakest actuator in the system within a defined working area. For this system, the downhole power section (EMM configuration) will be the weakest actuator and will stall out long before any of the other actuators. Before the algorithm is presented, WOB control mode settings and physical assumptions can be summarized as follows:

- WOB is PID controlled with HM rpm as system input
- Pipe-, bit- and pump rpm is set to a constant SP based on testing.
- If WOB SP is increased, pipe- and bit torque and rop is increased due to the physical relation.
- If WOB SP is decreased, pipe- and bit torque and rop is decreased due to the physical relation.

The team will use a simple discrete time boolean optimization algorithm to choose the WOB SP. The algorithm evaluates different parameters (mainly the DH power section torque) every 10 second, and have the opportunity to perform an incremental SP change of +-1Kg, or to stay at the current SP. The SP selector

can only perform an incremental change is the current SP is reached or is within a given limit. The WOB SP will be limited with a max value to protect drill pipe from buckling. The TD and HM torque will be limited by a max value to protect the motors from wear and the drill pipe from the risk of twist off. If the torque limit gets exceeded, a new and lower WOB setpoint will be set to lower the torque. The EMM nominal torque will be will be divided into 3 levels; (1) too low, (2) optimal and (3) too high. The levels will perform a incremental SP change of +1, +0 and -1 Kg respectively, where the goal is to stay in the EMM optimal level. The level limits must be chosen based on testing to prevent a limbo effect, where the EMM torque jumps in and out of the optimal level zone. The ROP will be limited by a minimum value to ensure that the operation will finish in time. If the ROP is too low, a new and higher WOB setpoint will be set to increase the ROP. If a state named the "safety sequence" is activated in the state machine in the control system, the WOB SP will be decreased by one increment. The safety sequence will be described along with the rest of the control system implementation in section 6.

A pseudo code to illustrate the working principle of the WOB SP selector is presented in Figure 54 below:

Figure 54: WOB SP Selector algorithm.

```
while True:
    if t = 10 True & WOB = WOB SP:

        #EMM
        if EMM torque < EMM low:
            WOB SP =+ 1
        if EMM torque > EMM low & < EMM high:
            WOB SP = WOB SP
        if EMM toque > EMM high:
            WOB SP =- 1

        #ROP
        if ROP < ROP min:
            WOB SP =+ 1

        #HM
        if HM torque > HM max:
            WOB SP =- 1

        #TD
        if TD torque > TD max:
            WOB SP =- 1

        #reset time
        t =- 10
    else:
        WOB SP = WOB SP

    if safety sequence = True:
        WOB SP =- 1
```

5.3 Position Controll Design

5.3.1 Design considerations

Based on the competition objective presented in subsection 1.2, the team have the following design consideration for the position controller:

- Drill a 4" vertical pilot hole.
- Drill a 2 3/8" horizontal and 20" vertical deviated well path along the north center line.
- Down hole measurements must be included in the closed loop control.
- High process noise in terms of vibrations while drilling.
- High magnetic distortion from parts installed on the rig and internally in the BHA.

Hence, the team needs to observe and control the relative position in 6 degrees of freedom (DOF), inclination and azimuth.

5.3.2 Steering- and Position Tracking options

Based on the rig setup presented in section 3, the team have the following steering- and position tracking options:

- Drill straight by rotating drill pipe using the TD servo in vertical-drilling mode.
- Build inclination by rotating only the drill bit using the PDM/EMM in slide-drilling mode.
- Control azimuth by orienting the BHA tool face using the TD servo in slide-drilling mode.
- Manipulate the inclination build rate by adjusting setpoints for WOB.
- Measure relative orientation of the BHA by using the DH IMU.
- Observe and control ROP by using the WOB controller.

5.3.3 Coordinate Frames

In order to track and control the position of the drill bit in 6 DOf, a set of frames needs to be defined. There are in total 5 main frames in our system. All frames are described in the list below and can be found in Figure 55.

Frames:

1. **World frame** - position free and fixed orientation with z-axis in gravity-up direction and x-axis north.
2. **Inertial frame** - located on top of the rock in the center, fixed to the rock with z-axis up and x-axis along the 12" side.
3. **Rig frame** - located on the drill floor, fixed to the rig with z-axis up and x-axis towards the riser system.
4. **TD frame** - located on the TD servo motor, fixed with z-axis up and x-axis to the internal motor position. TD servo motor is further fixed to the rig with a linear translation offset to the rig frame.
5. **IMU body frame** - located on the DH IMU, fixed to the sensor card in the sensor housing sub in the BHA. The IMU body frame axis representation will be derived in subsubsection 5.3.4.

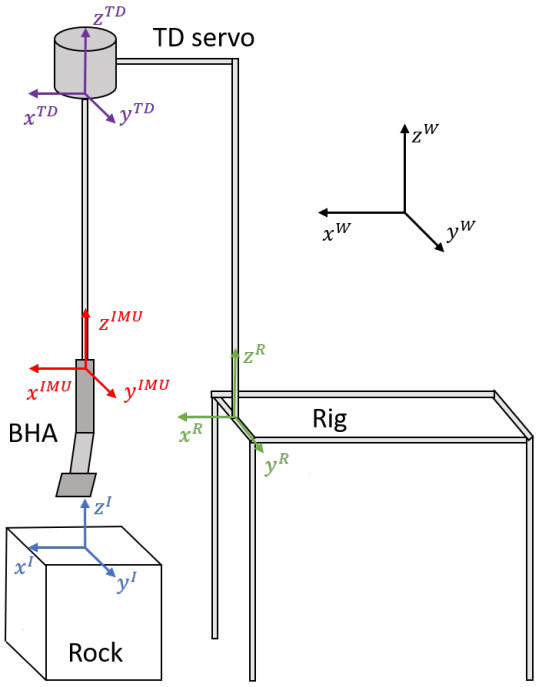


Figure 55: System frames

It is important to be aware of all system frames and how they relate to each other. The world frame is the overall reference frame fixed to the up- and north-direction, were the system frames can be rotated relative to the world frame. The rock sample holds the inertial frame, from were the team will track the position of the drill bit. The rock sample will be placed on the floor and therefore needs to be leveled and aligned with the world frame. The rotary system is through the hoisting system fixed to the rig frame. The rig frame, which is mechanically connected to the TD- and IMU body frame, also needs to be leveled with the world frame. Since the TD can rotate relative to the rig frame, rig alignment to the world frame's north axis is not necessary. However, the TD servo will be used to orient the BHA toolface, and by that steer in which direction the tool builds angle. Leveling and alignment of the rock and the rig will be done manually before the drilling starts.

5.3.4 IMU Body Frame

Up until now, the IMU body frame has just been defined as a frame located on the DH sensor card. The frame itself needs to be defined and represented as a rotation matrix relative to the world frame by using the accelerometer- and the magnetometer vectors. First, the body frame orientation can be defined. The internal accelerometer- and magnetometer frames and the body frame relative to the ICM-20948 chip is presented in Figure 56. The frame vectors can be defined as

$$\mathbf{a}^a = \begin{bmatrix} a_x^a \\ a_y^a \\ a_z^a \end{bmatrix} \qquad \mathbf{m}^m = \begin{bmatrix} m_x^m \\ m_y^m \\ m_z^m \end{bmatrix} \tag{22}$$

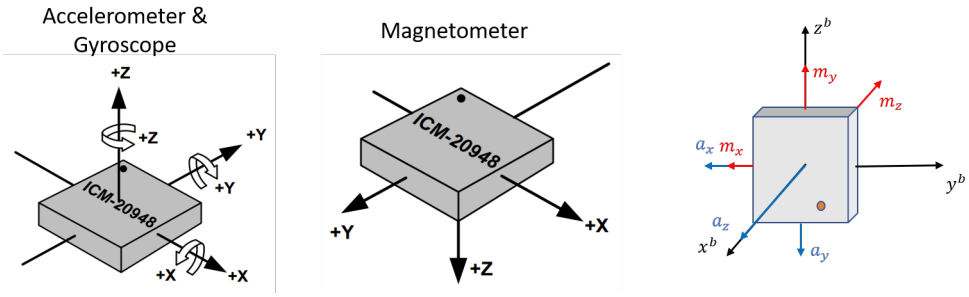


Figure 56: Accelerometer- and magnetometer frame relative orientation to the ICM-20948 sensor chip.

As explained in subsubsection 4.1.3, the DH sensor card will be placed in the sensor sub with the dot-marking pointing in down direction and in towards the center line. Let's define the body frame with z^b up, x^b in towards the center of the BHA and y^b as the orthogonal vector to both z^b and x^b, as shown in Figure 56. The accelerometer- and magnetometer frames needs to be rotated into the body frame. Let b, a and m denote the body frame and the original frames for each of the sensors.

To represent the accelerometer i the b-frame, the b-frame must be rotated -90deg around the x^b axis and $+90$deg around y^b axis. We have:

$$\mathbf{a}^b = \mathbf{R}_a^b \mathbf{a}^a \tag{23a}$$

$$\begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} a_x^a \\ a_y^a \\ a_z^a \end{bmatrix} \tag{23b}$$

$$= \begin{bmatrix} a_z^a \\ -a_x^a \\ -a_y^a \end{bmatrix} \tag{23c}$$

$$\tag{23d}$$

For the magnetometer, the b-frame must be rotated $+90$deg around the x^b axis and -90deg around y^b axis. We have:

$$\mathbf{m}^b = \mathbf{R}_m^b \mathbf{m}^m \tag{24a}$$

$$\begin{bmatrix} m_x^b \\ m_y^b \\ m_z^b \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} m_x^m \\ m_y^m \\ m_z^m \end{bmatrix} \tag{24b}$$

$$= \begin{bmatrix} -m_z^m \\ -m_x^m \\ m_y^m \end{bmatrix} \tag{24c}$$

The resulting orientation of the frames is presented in Figure 57.

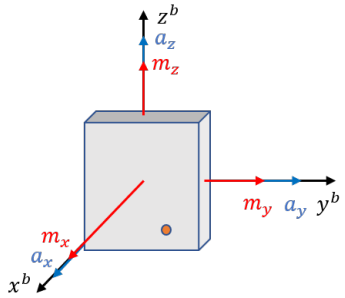


Figure 57: Accelerometer and magnetometer frames rotated into the body frame.

Body- and world frame relation

With all frames aligned, the relationship between the body frame and the world frame can be derived. In order to describe the orientation of the IMU, two reference points is needed. For this project, the earth's gravity and magnetic field will be used to describe the relative orientation of the card. That is, the world frame will defined with x^w north, y^w west and z^w up. The accelerometer vector a^b is a direct representation of the relation between z^w and z^b. The projection of the magnetometer vector m^b on to the earth horizontal plane represents the relation between x^w and x^b. For illustration purpose, let the z^b axis be aligned with the earth's gravitation and the x^b have a random orientation in the xy-plane. The accelerometer- and magnetometer vector in the body frame is presented in Figure 58.

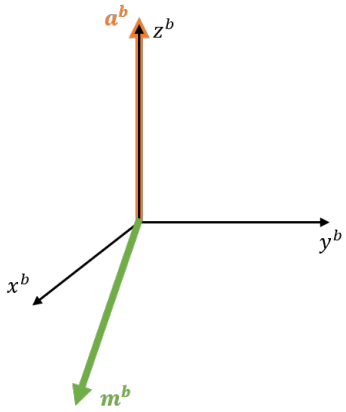


Figure 58: Accelerometer vector when IMU is aligned with the earth gravity, and random magnetometer vector.

The cross product between a^b and m^b will result in a vector perpendicular to both a^b and m^b. When a^b is aligned with up-direction, this will result in a vector in the horizontal earth-plane. Let's define this vector as y_w^b (west direction in world frame seen in body frame). We have:

$$\mathbf{y}_w^b = \mathbf{a}^b \times \mathbf{m}^b \tag{25a}$$

$$\begin{bmatrix} y_{wx}^b \\ y_{wy}^b \\ y_{wz}^b \end{bmatrix} = \begin{bmatrix} a_y^b m_z^b - a_z^b m_y^b \\ a_z^b m_x^b - a_x^b m_z^b \\ a_x^b m_y^b - a_y^b m_x^b \end{bmatrix} \tag{25b}$$

The cross product between y_w^b and a^b will result in a vector perpendicular to both y_w^b and a^b. Let's define

this vector as x_w^b (north direction in world frame seen from body frame). We have:

$$\mathbf{x}_w^b = \mathbf{y}_w^b \times \mathbf{a}^b \tag{26a}$$

$$\begin{bmatrix} x_{wx}^b \\ x_{wy}^b \\ x_{wz}^b \end{bmatrix} = \begin{bmatrix} a_y^b m_z^b - a_z^b m_y^b \\ a_z^b m_x^b - a_x^b m_z^b \\ a_x^b m_y^b - a_y^b m_x^b \end{bmatrix} \tag{26b}$$

The accelerometer vector a^b is a direct representation of the relation between the up direction in the world frame, and the body frame. Let's define this vector as z_w^b (up direction in world frame seen from body frame). We have:

$$\mathbf{z}_w^b = \mathbf{a}^b \tag{27a}$$

$$\begin{bmatrix} z_{wx}^b \\ z_{wy}^b \\ z_{wz}^b \end{bmatrix} = \begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} \tag{27b}$$

Before the rotation matrix can be defined, the orientation vectors needs to be normalized. We have

$$\hat{\mathbf{x}}_w^b = \frac{\mathbf{x}_w^b}{\sqrt{(x_{wx}^b)^2 + (x_{wy}^b)^2 + (x_{wz}^b)^2}} \tag{28a}$$

$$\hat{\mathbf{y}}_w^b = \frac{\mathbf{y}_w^b}{\sqrt{(y_{wx}^b)^2 + (y_{wy}^b)^2 + (y_{wz}^b)^2}} \tag{28b}$$

$$\hat{\mathbf{z}}_w^b = \frac{\mathbf{z}_w^b}{\sqrt{(z_{wx}^b)^2 + (z_{wy}^b)^2 + (z_{wz}^b)^2}} \tag{28c}$$

Finally, the rotation matrix that describes the orientation of the body frame relative to the world frame becomes:

$$\mathbf{R}_w^b = \begin{bmatrix} \hat{\mathbf{x}}_w^b & \hat{\mathbf{y}}_w^b & \hat{\mathbf{z}}_w^b \end{bmatrix} \tag{29a}$$

$$= \begin{bmatrix} \hat{x}_{wx}^b & \hat{y}_{wx}^b & \hat{z}_{wx}^b \\ \hat{x}_{wy}^b & \hat{y}_{wy}^b & \hat{z}_{wy}^b \\ \hat{x}_{wz}^b & \hat{y}_{wz}^b & \hat{z}_{wz}^b \end{bmatrix} \tag{29b}$$

Inertial and body frame relation

The main frame the team needs to track while drilling, is the IMU body frame relative to the inertial frame. In reality, the main objective to track is the drill bit, which is located with a linear offset to the IMU body frame. Just to simplify the model, the IMU body frame will represent the position of the whole BHA for now. The bit position relative to the IMU body frame is described in subsubsection 5.3.5. Below, the angular relation between the inertial frame and the body frame is defined. With the rock aligned and leveled with the world frame, the rotation matrix from body to inertial frame will be equal to rotation matrix from body to world frame. That is, for an ideal situation we have:

$$\mathbf{R}_I^b = \mathbf{R}_w^b \tag{30}$$

The inertial frame is centered on the top of the rock sample, with the z-axis pointing up and the x-axis along the 12" side of the rock in the north center-line (after manual alignment). The body frame is representing the position and orientation of the BHA, starting with the same orientation and position as the inertial frame before drilling. The x^b axis represent the toolface in which way the directional tool builds angle. The body frame can rotate around all axis, represented with the rotation angles; roll θ_z, pitch ϕ_x and yaw γ_y. This is shown in Figure 59. ROP will build in the negative z^b direction. Roll, θ_z, is the only rotation the team can manipulate to steer by using the TD servo motor.

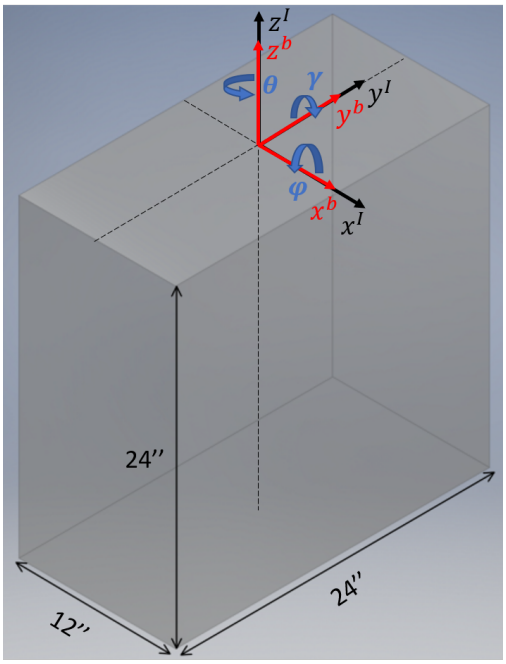


Figure 59: Inertial frame and body frame illustrated on the rock sample.

The rotation angles presented in Figure 59 can more easily be visualized in the 2-dimensional plane. Figure 60 shows the body frame relative to the inertial frame while following a drill path. The angles θ_z, ϕ_y and γ_x represent the rotation around its respective normal vector relative to the inertial frame.

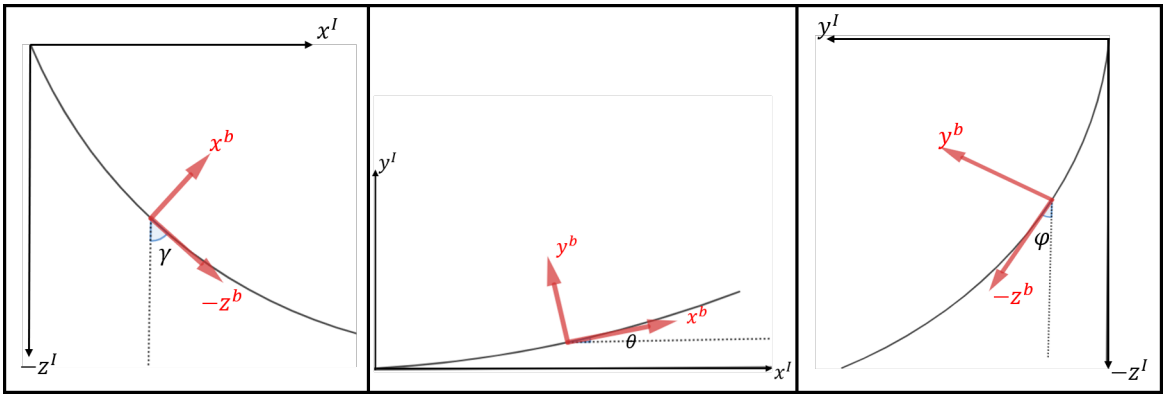


Figure 60: Angular relation between the inertial- and body frame.

The Euler angles can be calculated by using the components in the rotation matrix. To take care of the trigonometrical quadrant sign-error, where a value of sine or cosine can represent two possible angles in the

2-dimensional unit circle, the $atan2$ function will be used. For a rotation matrix between the body- and inertial frame, we have:

$$\mathbf{R}_I^b = \mathbf{R}_w^b = \begin{bmatrix} q11 & q12 & q13 \\ q21 & q22 & q23 \\ q31 & q32 & q33 \end{bmatrix} \tag{31a}$$

$$= R_x(\phi)R_y(\gamma)R_z(\theta) \tag{31b}$$

where

$$R_z = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{31c}$$

$$R_y = \begin{bmatrix} cos\phi & 0 & sin\phi \\ 0 & 1 & 0 \\ -sin\phi & 0 & cos\phi \end{bmatrix} \tag{31d}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\gamma & -sin\gamma \\ 0 & sin\gamma & cos\gamma \end{bmatrix} \tag{31e}$$

The Euler angles becomes:

(31f)

Roll angle

$$\theta = atan2(q21, q11) \tag{31g}$$

Pitch angle

$$\gamma = atan2(-q31, \sqrt{(q32)^2 + (q33)^2}) \tag{31h}$$

Yaw angle

$$\phi = atan2(q32, q33) \tag{31i}$$

Three important parameters used in the drilling industry to track the drilling process is inclination, azimuth and toolface angle. Inclination is the angle between the vertical up axis in the world frame and the z-axis along the BHA and is used to describe how much tilt the directional tool have built. Azimuth is the angle between the magnetic north axis in the world frame and the projected BHA z-axis on to the world xy-plane, and is used to describe the magnetic heading. Toolface is the angle between the magnetic north axis in the world frame and the projected BHA x-axis on to the world xy-plane, and is used to describe the orientation of the BHA bend. Inclination and azimuth

5.3.5 BHA Bit Position

The drill bit is located with a linear offset to the DH sensor card. An illustration is presented in subsubsection 5.3.5. It is assumed that the BHA remains rigid during the drilling operation, where the bending moment will be taken up in the flexible drill pipe. The bend sub has a fixed angle offset γ_{bit} with respect to the vertical center line. The bend sub will also point in a random direction in the xy-plane with respect to the body frame, dependent on how much you torque it up during installation. The team will use a torque wrench to ensure equal starting conditions, but manual measurement of the offset angle θ_{bit} between the x^b and x^{bit} will have to be done.

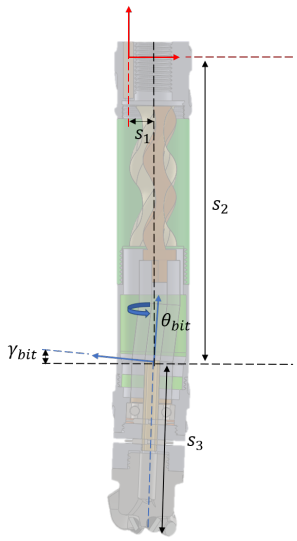


Figure 61: BHA bit position

The position of the drill bit relative to the sensor card can be described with a 4x4 transformation matrix. This matrix can hold both relative orientation and position. In subsubsection 5.3.5, it can be seen that the bit position can be described with two linear translations along x^b and $-z^b$, two rotations around z^b and the rotated y^b, and finally a linear translation along the rotated $-z^b$. We have:

$$\mathbf{T}^b_{b,bit} = \mathbf{T}^b_1\mathbf{T}^1_2\mathbf{T}^2_3\mathbf{T}^3_4\mathbf{T}^4_{bit} \tag{32a}$$

$$= \begin{bmatrix} 1 & 0 & 0 & s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta_b & -sin\theta_b & 0 & 0 \\ sin\theta_b & cos\theta_b & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{32b}$$

$$\begin{bmatrix} cos\gamma_b & 0 & sin\gamma_b & 0 \\ 0 & 1 & 0 & 0 \\ -sin\gamma_b & 0 & cos\gamma_b & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{32c}$$

$$= \begin{bmatrix} cos\theta_b sin\gamma_b & -sin\theta_b & cos\theta_b sin\gamma_b & -cos\theta_b sin\gamma_b s_3 + s_1 \\ sin\theta_b cos\gamma_b & cos\theta_b & sin\theta_b sin\gamma_b & -sin\theta_b sin\gamma_b s_3 \\ -sin\gamma_b & 0 & cos\gamma_b & -cos\gamma_b s_3 - s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{32d}$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & x^b_{bit} \\ r_{21} & r_{22} & r_{23} & y^b_{bit} \\ r_{31} & r_{32} & r_{33} & z^b_{bit} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{32e}$$

where the the orientation is stored in the upper left 3x3 matrix and the position in the upper right 3x1 vector. The team is mainly interested in the position of the bit relative to the sensor card. Hence, the position from body to bit seen from the body frame can be defines as

$$\mathbf{x}^b_{b,bit} = \begin{bmatrix} x^b_{b,bit} \\ y^b_{b,bit} \\ z^b_{b,bit} \end{bmatrix} = \begin{bmatrix} -cos\theta sin\gamma s_3 + s_1 \\ -sin\theta sin\gamma s_3 \\ -cos\gamma s_3 - s_2 \end{bmatrix} \tag{32f}$$

5.3.6 Optimal Trajectory

The position controller will be programmed to follow a precalculated reference path. This path was derived in the Drillbotics design report and will therefore be rendered in its full form for a complete understanding of the position controller [9]. However, some constants, variables and sentences will be assigned to new letters or reformulated. All text in subsubsection 5.3.6 is taken from *9.3.1 Derivation of Optimal Trajectory* [9]:

Before the position equations can be derived, a reference trajectory needs to be defined. The optimal trajectory will, in this case, be the trajectory that minimizes the stress on the drill string, and at the same time is able to build enough angle to exit the rock sample at least x_{min} =2 3/8" from a vertical center line through the rock bottom. As described in *6 Mechanical Design*[9], calculations show that the horizontal displacement is limited to approximately $x_{max} = 4.5$" to stay within the elastic limit of the drill string. With no further calculations, it can be argued that the optimal trajectory will be with a constant angle build rate, and a horizontal displacement of 2 3/8"$\leq x < 4.5$". Due to multiple uncertainties in the mathematical model and by using the fact that the calculated values are only valid for a perfect scenario, the team has decided to keep the horizontal displacement as close to the competition limit as possible, plus a safety factor. Below, the team will derive the optimal reference path and the needed angle build rate, based on a horizontal displacement of $3.15" = 8cm$ as a starting point.

After the initializing hole of 4" is drilled, the directional drilling starts. Assume we kick off in a 0° angle in reference to the vertical z-axis, and exit the rock sample at an unknown angle α through the rock bottom. An illustration is shown in Figure 62 where $z = 20"$, $x_1 = 3.15"$ and the curved line EG is the optimal path.

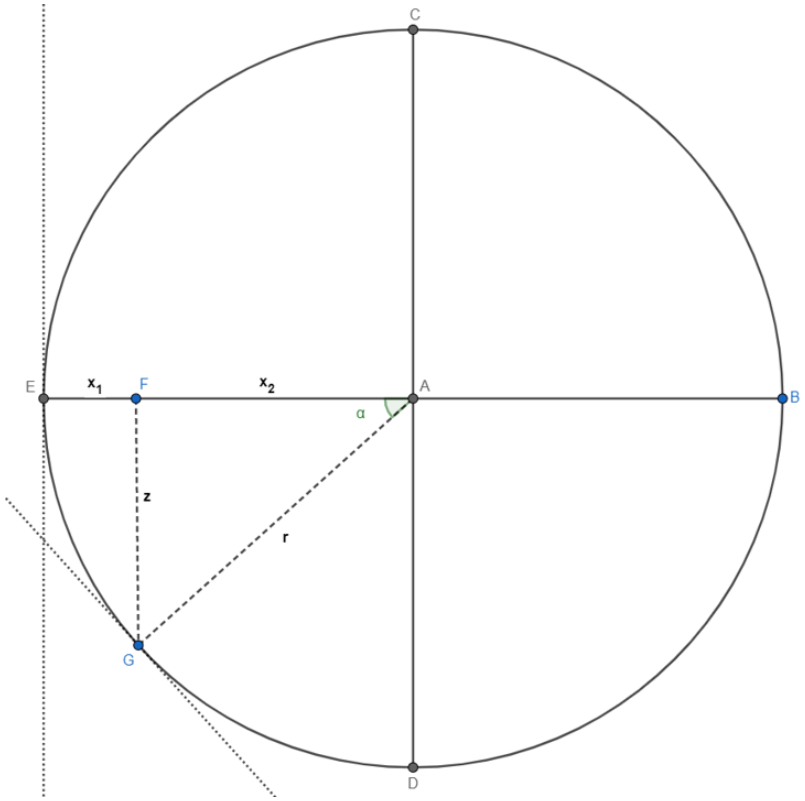


Figure 62: Optimal drilling trajectory derivation

Using the geometric dependencies, we have:

$$r = x_1 + x_2 \tag{33a}$$
$$r^2 = x_2^2 + z^2 \tag{33b}$$
$$\sin\alpha = \frac{z}{r} \tag{33c}$$

Hence, the drill path radius r and the exit angle α becomes:

$$r = \frac{x_1^2 + z^2}{2x_1} \tag{33d}$$
$$\alpha = \arcsin\frac{z}{r} \tag{33e}$$

The calculated constants can be found in Table 12. With r and α, the needed angle build rate $\dot{\alpha}$ can be calculated with respect to drilled depth. Using

$$z_t(t) = r\alpha(t) \tag{34a}$$

where z_t is the arc length of the drilled trajectory (denoted "t") along the optimal reference path, the angle build rate in unit [deg/cm] for both the arc length z_t and the depth z respectively then becomes:

$$\dot{\alpha}_{z_t} = \frac{\alpha}{z_t} \tag{34b}$$
$$\dot{\alpha}_z = \frac{\alpha}{z} \tag{34c}$$

Finally, the reference path for the optimal trajectory can be expressed in 3 dimensions with respect to α_r:

$$\mathbf{r}^I = \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} x_0 + r(1 - \cos\alpha_r) \\ y_0 \\ z_0 - r\sin\alpha_r \end{bmatrix} \tag{35}$$

where $[x_0\ y_0\ z_0]$ is the staring point of our body frame relative to the inertial frame, and $\alpha_r \in [0\ 17.9°]$. A graphical illustration of the optimal trajectory and the different trajectory zones, can be found in Figure 63.

Table 12: Constants for optimal trajectory calculations

α	17.9°	Exit angle relative to the vertical z-axis
z	20.00" = 50.80 cm	Vertically drilled depth along the z-axis
z_t	20.33" = 51.64 cm	Trajectory arc length
x_1	3.15" = 8.00 cm	Horizontally drilled length along the x-axis
x_2	61.93" = 157.29 cm	Geometrical dependency
x_{min}	2 3/8"= 6.03 cm	Min horizontal displacement along North axis to achieve max points
x_{max}	4.5"= 11.43 cm	Max horizontal displacement along North axis to stay within elastic limit
r	65.07" = 165.29 cm	Drill path radius
$\dot{\alpha}_{z_t}$	0.347 deg/cm	Angle build rate with respect to trajectory length
$\dot{\alpha}_z$	0.352 deg/cm	Angle build rate with respect to vertically drilled depth

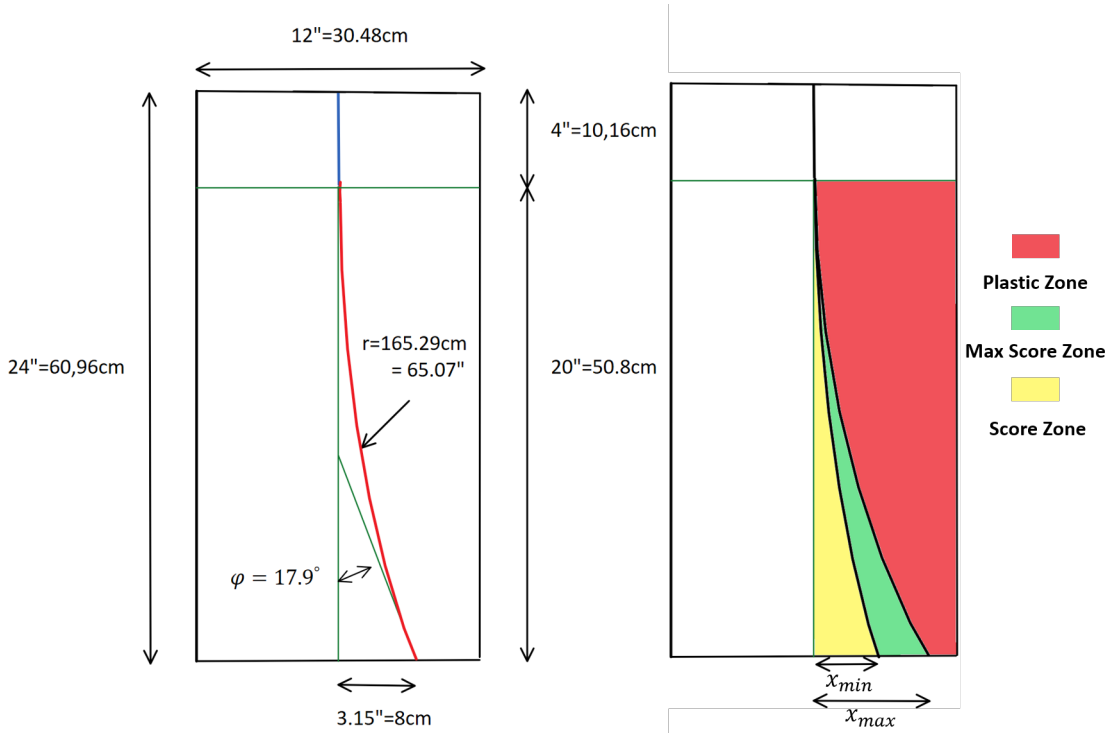


Figure 63: Left: The optimal drilling trajectory illustrated in the xz-plane centered in $y = 0$, with a horizontal displacement of 8cm. Right: Trajectory zones where the goal is to stay within the max score zone.

5.3.7 Position Equation

In order to calculate the position of the drill bit in continuous time, the team will use the first order Explicit Runge Kutta equation (ERK1), also known as the Euler equation. This method calculates the position in discrete time based on last location, a chosen step-length and the rate of change. The method is step-wise linear and is described in more detail in subsection 2.1. By using the fact that the drilling operation can defined as a slow process and with an IMU update frequency of 60 Hz, the Euler equation will make a good approximation of the position in continuous time. The general expression for the Euler equation can be defined as:

$$y_{k+1} = y_k + h\dot{y}_k \tag{36}$$

where k is the iteration number and h is the step-length.

In order to use the Euler equation to calculate the position of the bit in the inertial frame for each time step, the rotation matrix R_I^b that describes relation between the inertial- and body frame must be used. For an orthogonal rotation matrix $R \in R^{nxn}$ the following dependencies yields:

$$\mathbf{R}_b^I = (\mathbf{R}_I^b)^{-1} = (\mathbf{R}_I^b)^T \tag{37}$$

There are two vectors that needs to be rotated back to the inertial frame; (1) the velocity vector (ROP) and (2) the bit position vector relative to the body frame. The ROP is actuated by the linear movement of the HM along the z^I axis. This movement is transmitted from the HM system trough the drill pipe and down to the BHA. It can be assumed that this movement is parallel to the length of the BHA, hence also parallel

to the z^b axis. This means that the ROP will always move along z^b and the velocity vector can be defined as:

$$\dot{\mathbf{x}}_b^b = \begin{bmatrix} 0 \\ 0 \\ \dot{z}^b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ ROP \end{bmatrix} \tag{38}$$

By using the Euler equation, the rotation matrix from body- to inertial frame and the bit position in the body frame, the discrete time bit position in the inertial frame can be derived as:

$$\mathbf{x}_{bit,k+1}^I = \mathbf{x}_k + \mathbf{R}_b^I \left[h\dot{\mathbf{x}}_b^b + \mathbf{x}_{b,bit}^b \right] \tag{39a}$$

$$= \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} + \begin{bmatrix} q_{11} & q_{21} & q_{31} \\ q_{12} & q_{22} & q_{32} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \left[h \begin{bmatrix} 0 \\ 0 \\ \dot{z}^b \end{bmatrix} + \begin{bmatrix} x_{b,bit}^b \\ y_{b,bit}^b \\ z_{b,bit}^b \end{bmatrix} \right] \tag{39b}$$

$$= \begin{bmatrix} x_k + q_{31} h \dot{z}_{hm} + \hat{x}_{b,bit}^I \\ y_k + q_{32} h \dot{z}_{hm} + \hat{y}_{b,bit}^I \\ z_k + q_{33} \dot{z}_{hm} + \hat{z}_{b,bit}^I \end{bmatrix} \tag{39c}$$

where the bit position is described with the the position of last iteration, the position change based on the step length, orientation and hoisting speed and a constant bit offset relative to the body frame seen from the inertial frame.

An alternative to using the ERK is to model the kinematics of the system by using the IMU data. That is, the linear velocity can be found from the integral of the acceleration, orientation from magnetometer- and accelerometer data and orientation rate from the gyroscope. This was never really considered due to expected process noise from the drilling operation, drift on gyroscope and the fact that the topside hoisting motor will provide an accurate velocity and vertical position. That is, it is considered that it will more accurate to only use the downhole sensor data to calculate the orientation of the IMU, and rather use topside references for velocity. The acceleration is neglected from the model due to the fact that this mainly consists of process noise (vibrations).

5.3.8 Toolface Controller

The main objective of the toolface controller is to orient the BHA toolface in the direction of where to build angle. In Figure 64, an overview of the block diagram for the toolface controller is presented. As for the WOB controller, a Kalman filter is implemented to output a filtered estimation of the IMU orientation to deal with the expected high-frequency noise on the measurement signal. The toolface angle $\hat{x}_{tf}$ is further calculated and compared with a chosen setpoint. The toolface error is fed into a PI controller that calculates a new SP for the TD position. The loop is closed with the IMU taking a new measurement.

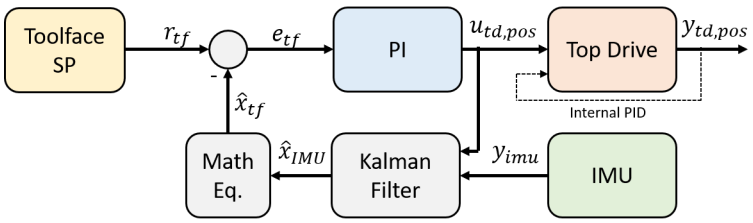


Figure 64: Toolface controller scheme.

PID: The team considers a PI controller to be sufficient to control the toolface angle. The derivative effect is removed to avoid unnecessary toolface correction due to process noise from vibrations and internal magnetic

distortion in the BHA. These corrections is not necessary and will only lead to torque peaks on the DH power section. Further, the team needs a controller that does not overshoot and corrects for steady state error. For a slow drilling process, a slow tuned PI controller where position and not velocity is system input will in theory work well for this application. The mathematical derivation of the PI controller is similar to the PID controller presented in subsection 5.2 only without the derivative effect and will therefore not be written out any further.

Kalman filter: As for the WOB controller, the IMU will be modeled as a stationary system. The raw IMU measurements will be filtered directly, where the control state will be calculated after. The measurement noise variance will be calculated for each sensor axis by analysing the stationary measurements. The process noise covariance matrix will be tuned to get the desired response. A time delay due to the "un-modeled" parts of the system will still be present where the state estimation will fully rely on the measurements, but for a slow drilling process this will not be as critical as for the WOB controller. For the WOB, the state grows exponentially for a constant velocity input. For the IMU data, the time constants will be up to several minutes due to the relative slow ROP and even slower turn rate. This will be taken advantage of in the tuning of the filter. For the 9 state system, the system matrices for a stationary IMU consisting of magnetometer, gyroscope and accelerometer becomes:

$$A = C = diag[1] \in \mathbf{R}^{9x9} \tag{40a}$$

$$B = D = 0 \tag{40b}$$

and the measurement- and process noise covariance matrices becomes:

$$R = \begin{bmatrix} Var(mag) & 0 & 0 \\ 0 & Var(gyro) & 0 \\ 0 & 0 & Var(accel) \end{bmatrix} \tag{40c}$$

$$Q = \begin{bmatrix} Q(mag) & 0 & 0 \\ 0 & Q(gyro) & 0 \\ 0 & 0 & Q(accel) \end{bmatrix} \tag{40d}$$

where

$$Var(mag) \ \& \ Var(gyro) \ \& \ Var(accel) \ \& \ Q(mag) \ \& \ Q(gyro) \ \& \ Q(accel) \in \mathbf{R}^{3x3} \tag{40e}$$

5.3.9 Position Controller

The main objective of the position controller is to correct for deviation in azimuth. In Figure 65, an overview of the block diagram for the position controller is presented. The position controller is design around the toolface PI controller, where the accumulated azimuth from the north centerline is calculated and used as toolface SP. The estimated position is calculated by using the Euler equation with the estimated IMU orientation and hoisting velocity as input. The estimated position is compared with the desired reference path presented in Equation 35 and azimuth from the rock center is calculated. The toolface PI controller will by adjusting the toolface steer back to the desired reference path. Based on the steering capabilities of the directional tool, only correction for deviation in azimuth will be corrected for. That is, inclination build rate can be manipulated to some extent but will not be corrected for to simplify the controller. This is also not a requirement in the competition problem statement, and the team have designed the bend angle for the tool to drill within the max score zone.

Both PI controller and Kalman filter is the same as for the toolface controller and will not be written out any further.

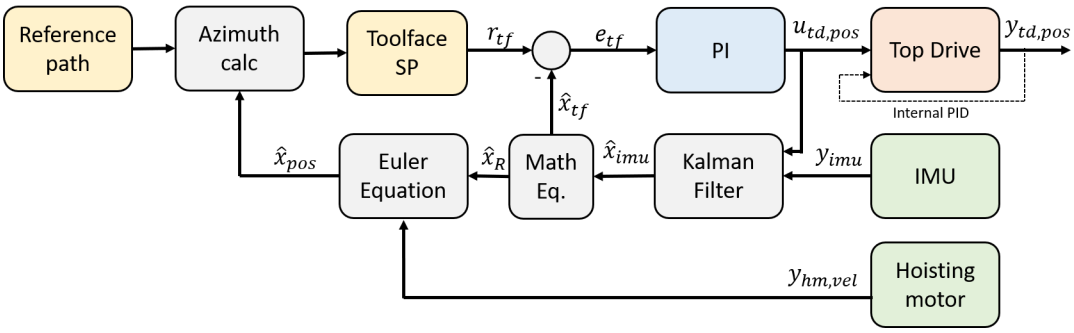


Figure 65: Position controller scheme.

6 Implementation of Control System

In this section, the implementation of the control system in Labview will be explained. This includes a review of the instrumentation- and actuator communication setup, implementation of state controllers, implementation of a state machine and simulation script, and implementation of the graphical user interface. The implementation of the control system in Labview is one of the main parts of the scope for this project, and can be read as a part of the project result. Chief engineer, Steffen Værnes Moen, have helped the team to setup external modbus and CANopen communication protocol with the respective drives for the topside actuators. The internal programming in Labview have been done by the team alone. The section frequently refers to figures with block diagrams to exemplify the code. These block diagrams is generally too large to present in a report figure, and is mainly used as a visual reference. The reader is referred to the appendix for full size block diagrams.

6.1 Introduction to the Autonomous Control System

6.1.1 Design Considerations

The team have decided on the following design consideration and criteria for the 2019 control system.

- Design a state machine to solve 2018/19 problem statement introduced in subsection 1.2.
- Design a control system for the 2019 mechanical design with considerations on the weakest actuator in the system.
- Clean and well organized code for easy handover to next years team.
- User friendly graphical user interface.
- Systematic data logging for later analysis.
- Improved system safety logic to ensure material- and human safety, and operational uptime.
- Program a state machine simulation script for early code testing and to build the foundation for a digital twin for later projects.

6.1.2 Labview 2018

The team will use Labview 2018 to program the autonomous control system. Laboratory Virtual Instrument Engineering Workbench (Labview), is a development environment for graphical programming by National Instruments (NI). As for any other programming language, the environment supports while- and for-loops, if statements and variables, inputs and outputs. The only difference is the syntax, where the code is connected through block diagrams. A main program in Labview is referred to as a VI (virtual instrument), and a function that is called from the main program is referred to as a subVI. Labview is generally divided into two main parts; (1) block diagram and (2) front panel. The block diagram holds the code and can output indicators, graphs and plots to the front panel. The front panel is the graphical user interface (GUI), where the user can modify, edit and organize the controls and data visualization to its preferences. Labview has a large library of built-in functions and supports most communication protocols.

Labview was first introduced to NTNU Drillbotics by the 2018 team who built a control system designed for easy handover to next years team. In the *Drillbotics - Design report*, the team did thorough a review of the 2018 control system to learn and build on the experience the 2018 team acquired during last years competition. The report suggested the following improvements and new features:

- Make block diagram code more third party friendly by using more subVI's and increased signal bundling to reduce visible wiring.
- Implement a directional mode in the state machine.

- Include plots of the position in 3-axis on the front panel including a reference path.
- Implement a live log on the front panel.

The 2019 control system is programmed from scratch in Labview 2018 with inspiration from the 2018 state machine.

6.1.3 New Features

New features in the 2019 control system can be summarized as follows:

1. **Directional drilling state:** New state in the state machine to perform directional drilling.
2. **Safety sequence state:** New state in the state machine to have a steady state for the system to rest in the wait for an error to clear.
3. **Two-layered safety logic:** Medium and critical safety layers to improve overall system safety and operational uptime.
4. **Setpoint selector:** A discrete time boolean optimization algorithm for simple performance optimization of the weakest actuator in the system (EMM torque).
5. **Tab panels:** The front panel is build up by tab panels to increase the level of state customization for multi-phased state machines (vertical- and directional phase).
6. **Settings panel:** Improved settings panel where more user defined settings is moved from the block diagram to the front panel.
7. **Position tracking plots:** Live position tracking plots with user defined reference path.
8. **Live operation log:** Live operation log to print the operation status, next state and all changes and decisions made by the autonomous system.

6.1.4 General Code Setup

Before the control system is explained, the general code setup must be introduced. The code is generally build up by four sequences in a sequence structure; (1) initialize, (2) operation, (3) close and (4) export. See Figure 66. The init sequence setup the communication and create tasks for each actuator and some sensors. The Operation sequence is where the drilling operation takes place. A while loop runs with a 9-state state machine and terminates after the state criteria has been fulfilled. In general, all motor and sensor reads/writes happens inside the while loop (outside the state machine), and setpoints and termination criteria happens inside the state machine. The close sequence ends the tasks created in the init sequence. The export sequence exports all drilling data automatically to a text file.

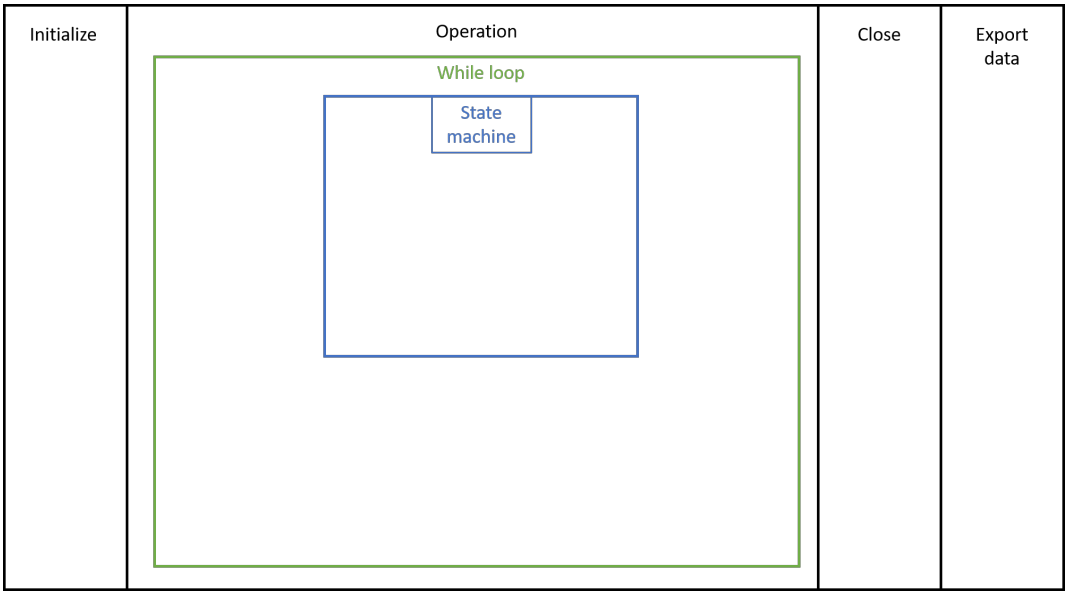


Figure 66: General code setup in Labview

6.2 Communication Setup

In this subsection, the communication setup for each actuator and sensor in Labview will be described. It must be stated that this has been a time consuming part of the project in terms of equipment delivery time, setup- and code debug time. Full version of each block diagram can be found in Appendix C. Details on each motor and sensor can be found in section 3.

6.2.1 Communication Flow Overview

An overview of the communication flow from Labview to each motor and sensor can be found in Figure 67 below.

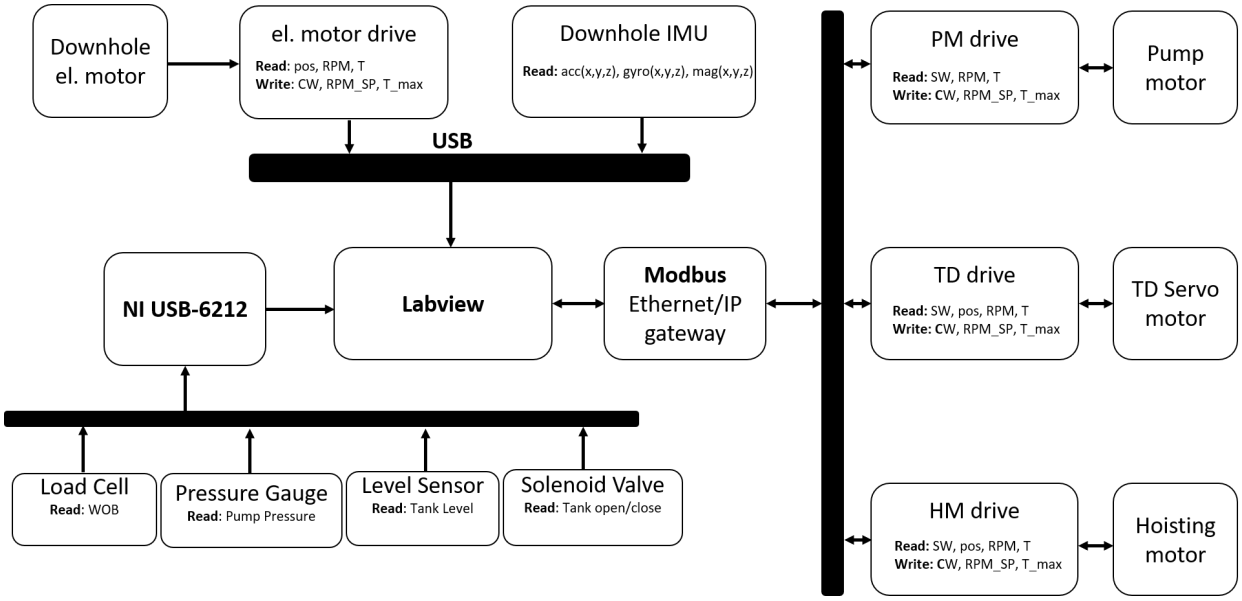


Figure 67: Communication flow chart

Modbus communication protocol: Modbus is a serial communication protocol developed by Modicon for PLC control [54]. The protocol consists of a master-slave hierarchy, where the requesting part is referred to as the master and the supplying part is referred to as the slave. In Labview, a Modbus master client instance is created by using built in Modbus VI library. By writing control words the different holding registers in the slave instance, the master can read/write data from/to the registers. This communication protocol will be used for the hoisting motor, pump motor and top drive.

6.2.2 Hoisting Motor

The hoisting motor communication is setup with modbus communication through a modbus-to-Ethernet/IP gateway. Modbus is a client/server hierarchy communication protocol where the Labview work as the client/master and the motor drive (Lenze) works as a server/slave. The communication setup for the hoisting motor is similar to the 2018 setup with minor changes. See Figure 68. A modbus client instance is created with the IP address of the Lenze drive. By writing different control words to different holding registers the team can read/write different parameters. Finally the master client instance is closed. For the general code setup presented in Figure 66, the modbus create, read/write and modbus close will be placed in the init-, operation- and close- sequence respectively. The hoisting motor reads/writes can be summarized as follows:

- **Read:** RPM, torque [Nm], scaled position [mm]
- **Write:** RPM SP, torque limit [Nm]

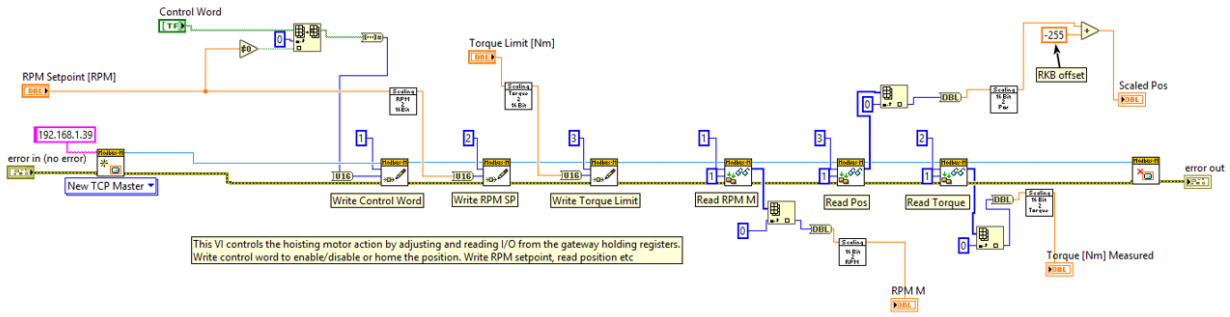


Figure 68: Hoisting motor communication subVI

6.2.3 Pump Motor

The pump motor communication is similar to the hoisting motor, where the only difference is the different holding registers. See Figure 69. The pump motor reads/writes can be summarized as follows:

- **Read:** RPM, torque [Nm]
- **Write:** RPM SP

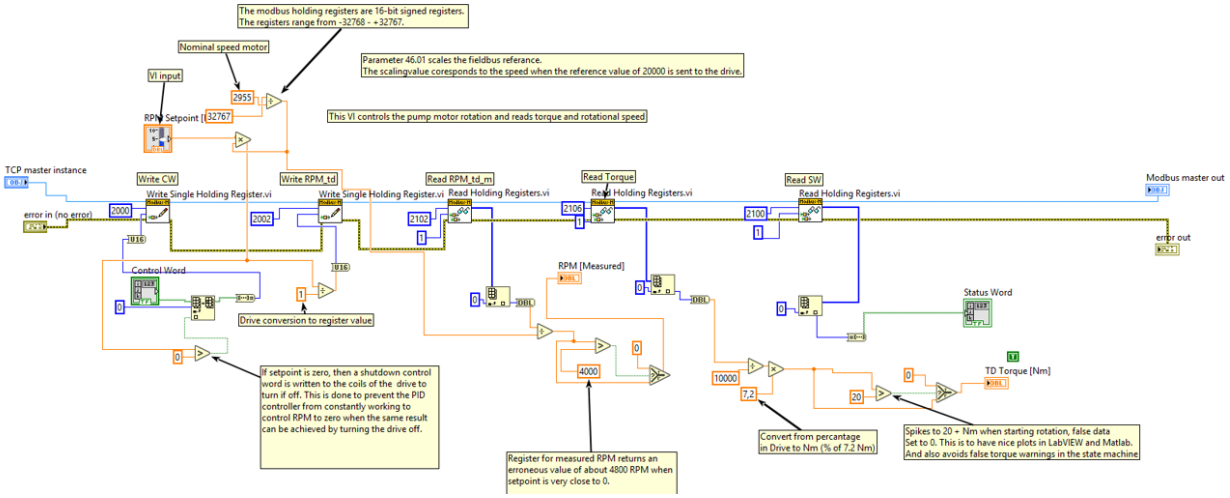


Figure 69: Pump motor communication subVI

6.2.4 Top Drive Servo Motor

The top drive communication setup differs from the hosting- and pump motor setup due to technical difficulties with Modbus and delivery time for spare parts and converters. The communication protocol was changed to CANopen last minute and is fully programmed by Steffen Værnes Moen. More info on CANopen can be found at the NI webpages [55].

The top drive communication script is dropped in as a subVI, where the script runs within its own while loop. A partly block diagram is presented in Figure 70. That is, the TD communication script works an external script, where the team uses global variables to control it. All external communication is programmed by the team. Since the top drive will be used for two purposes (vertical and directional drilling), the team can switch between different modes. The modes with a short description can be summarized as follows:

- **Velocity:** In velocity mode, the TD runs on a user defined velocity setpoint. An internal PID controls the velocity to stay at the setpoint. This mode will be used during the vertical drilling phase.
- **Position:** In position mode, the user can choose between absolute and relative position change. The team will used absolute position to control the toolface during the directional drilling phase. Position homing is however required. The position is changed by choosing a new setpoint and by toggling a bit in the system control word to activate the change.
- **Torque:** In torque mode, the TD runs on a user defined torque setpoint. This mode is not used by the team due to that the TD torque will not be an issue with the planned operation. Also, the EMM will be the weakest actuator in the system which means the TD will run on relative low nominal torque in comparison.
- **Homing:** In homing mode, the absolute position is zeroed. This is a requirement before going into the position mode.

The top drive reads/writes can be summarized as follows.

- **Read:** RPM, torque [Nm], absolute- and relative position [deg]
- **Write:** mode (position/homing/velocity/torque), RPM SP, torque SP, position abs/rel SP [deg]

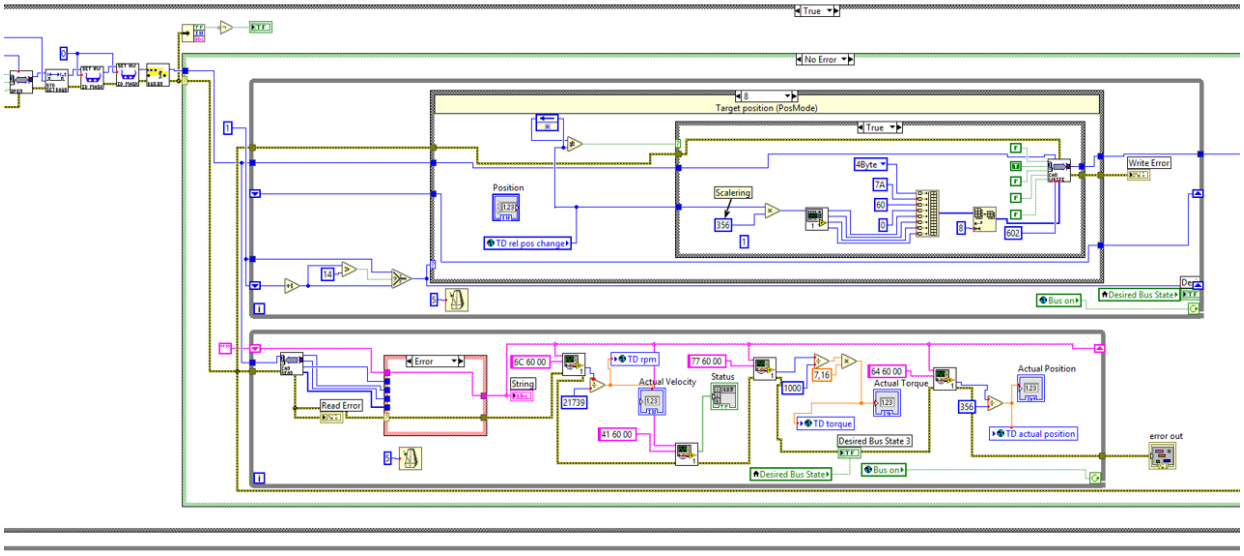


Figure 70: Top drive communication subVI (partly).

6.2.5 El. Mini Motor

The electrical mini motor (EMM) communication setup uses built-in serial communication blocks from EPOS Studio 4 [56]. An block diagram is presented in Figure 71. This communication has been setup by team member Hilde Eiken Helle with support from the Maxon support team. As explained in section 3 and subsubsection 4.1.4, the EMM comes with a decoder on top of the motor that can measure rpm and power consummations. The torque is calculates based on a linear relation with the current consumption. The EMM reads/writes can be summarized as follows:

- **Read:** RPM, current output [mA], torque [Nm] (calculated as a function of the current).
- **Write:** RPM SP, current input.

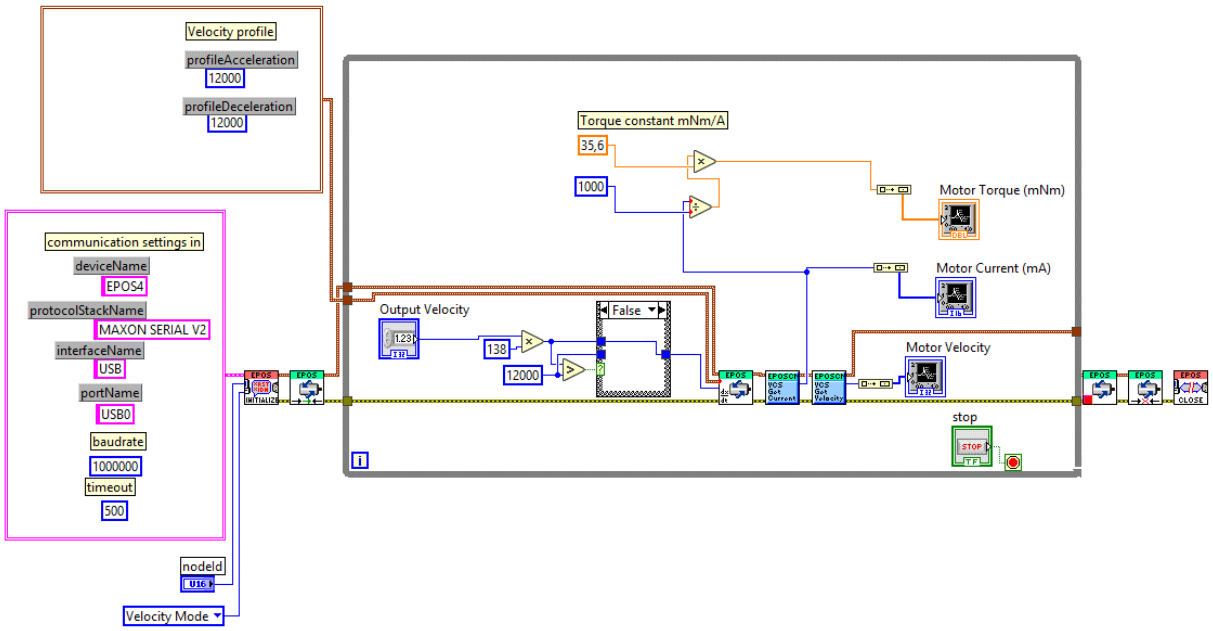


Figure 71: Electrical mini motor (EMM) communication subVI.

6.2.6 Topside Instrumentation Units

The communication setup for the topside instrumentation and solenoid valve is generally the same. The digital and analog I/O on NI USB-6212 DAQ device is used to read and write data signals from the sensors to Labview. The built-in DAQ assistant subVI is used to setup the analog and digital signals in Labview. The following units is setup with Labview DAQ assistant:

- Load cell
- Pressure gauges
- Solenoid valve

The load cell DAQ assistant subVI is presented in Figure 72 as an example. The DAQ sub outputs the measurement as a voltage signal in range +-10V. Linear interpolation is used to covert the voltage signal to a load measurement in [kg], as described in Equation 10 in section 3. The conversion principle is equal for the pressure gauge. The solenoid valve however uses a digital output (DO) to switch between high and low state. This is used to switch between closed and open solenoid valve. Block diagrams for all DAQ subVI's can be found in Appendix C. The DAQ assistant settings for each device can be found in Table 13.

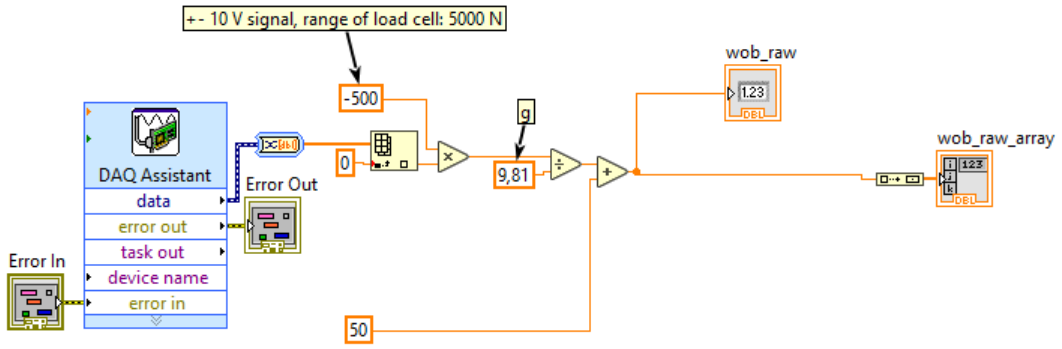


Figure 72: Load cell DAQ assistant subVI.

Table 13: Labview DAQ Assistant settings

Device:	Config:	Min:	Max:	Sample count:	Sample rate [Hz]:	Shunt resistor:
Load Cell:	AI	-10V	10V	1	1k	NA
Pressure gauge:	AI	4mA	20mA	1	1k	430 Ohm
Solenoid:	DO	0	5V	1	NA	NA

6.2.7 Down Hole Sensor Card

The downhole sensor card communication uses the NI VISA subVI's to setup and read the serial output from the IMU. An example of the IMU communication block diagram is presented in Figure 73 and the VISA settings is presented in Table 14. The VI is programmed to read the serial input as long as there is new data, or use the previous iteration's data. As explained in subsection 4.4, the sensor card outputs a serial print with all IMU data labeled with name (mx:) and a stop character (d) for each measurement. See example in Figure 74. That is, the VI is programmed to read the string object between the name and the stop character and convert it to a numeric value. This is done for each individual measurement, where the VI is programmed to parse out each individual serial string, convert it to a numeric value and define it as a variable (mx, my, mz etc.). An example is presented in Figure 75. At the end of each print, a termination character is printed to end the serial read (not a visual character in the serial print).

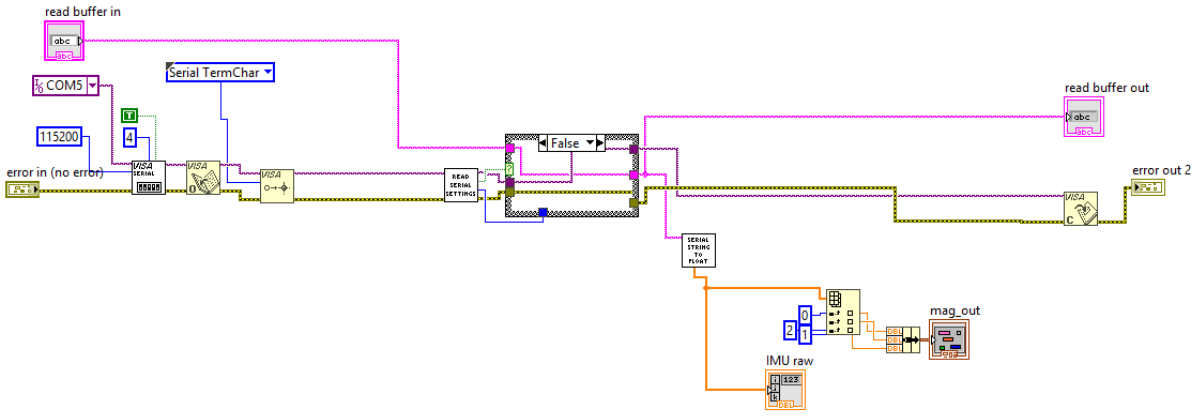


Figure 73: IMU individual subVI.

Table 14: IMU VISA serial settings

Device:	I/O	Baud rate	TermChar	Serial settings	Timeout	Convertion
IMU	COM X	115200	4	Bytes at Port	5ms	String to Numeric

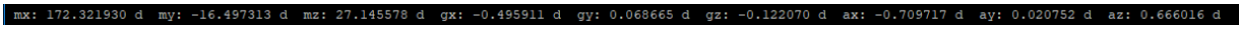


Figure 74: Serial print from IMU

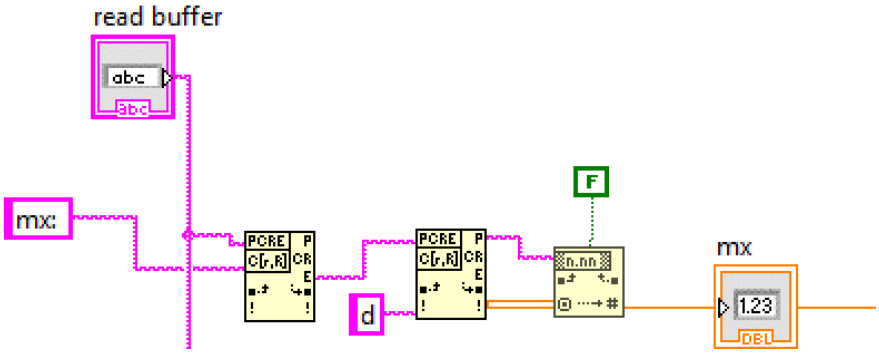


Figure 75: Example of serial read parsing, conversion from string to numeric and variable definition for the magnetometer x-axis.

6.3 Implementation of Filters and Controllers

6.3.1 Kalman Filter

The Kalman filter is implemented with the built-in discrete time Kalman filter subVI from the control and simulation package in Labview. The built-in subVI is based on the same estimation algorithm as described in subsection 2.1, where the user can define and customize the state-space model and the noise model. The advantage of using a built-in VI is that these blocks are optimized for real time operations. A hard-coded Kalman VI can be more computational heavy on the program and most likely result in a slower iteration time. An example of the Kalman filter sub-VI is presented in Figure 76. A user defined state-space model, noise model, prior estimate, initial estimate, system output and system input is fed into the filter block. The system matrices and noise covariance matrices is defined in subsection 5.2 and subsection 5.3. The system output (y) includes all 9 IMU measurements and the WOB measurement (in total 10 states). For a stationary un-modeled system, the system input (u) is set to zero. The Kalman filter block output the estimated state vector ($\hat{x}$) which is later used by the WOB- and position controller.

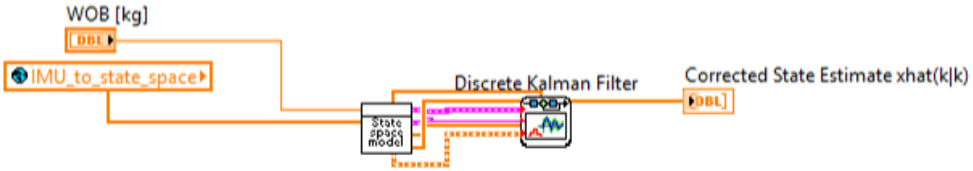


Figure 76: Labview Kalman subVI with user defined stochastic state-space model and second order statistic noise model.

6.3.2 WOB PID Controller

The WOB PID controller is hard-coded with inspiration from the 2018 team's controller. The reason for this is to add more user-customizability such as the ability to reset the integral effect in soft rock formations to avoid false integral build-up, and restrict the PID output with an upper and lower limit to avoid buckling. The VI can be divided into an outer and inner subVI to simplify the code. The outer core presented in Figure 77, holds the user defined settings such as PID gains, upper/lower saturation point, inputs and outputs. The inner core presented in Figure 78, holds the the actual integral and derivative action, and the equations is based on the theory and control schemes presented in subsection 2.1 and subsection 5.2. The Kalman filtered WOB measurement is fed into the controller, which outputs a new SP for the hoisting motor. The HM SP is limited to $HM_{SP} \in [-100, 200][\text{rpm}]$ to reduce the risk of buckling. If the WOB SP is set to zero, the HM SP will automatically be set to zero to prevent the PID from continuing the control process around the user corrected zero level. As described in subsection 3.3, the weight of the equipment in the rotary stack is subtracted from the measurement signal to zero out the measurement. If this correction is slightly on the minus side, the PID will try to correct for this resulting in a uncontrollable hoist-up movement that can pull off the BHA at the drill floor stabilizer. This function logic prevents that.

The inner PID core is equal for both the position controller and WOB controller. The user defined control gains will be presented as a result after testing in subsection 7.4.

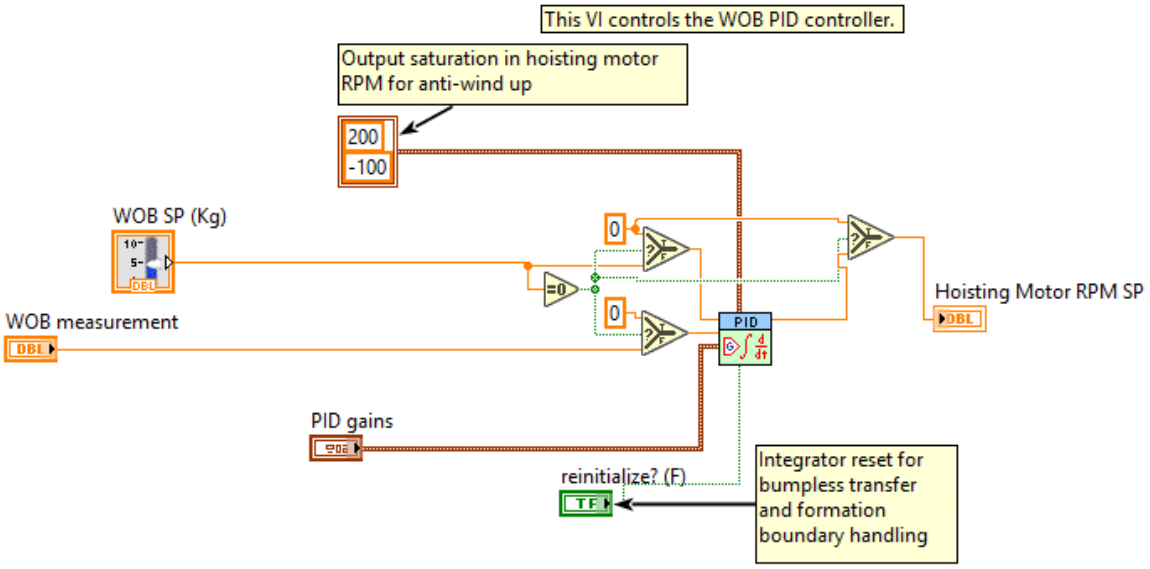


Figure 77: WOB controller subVI

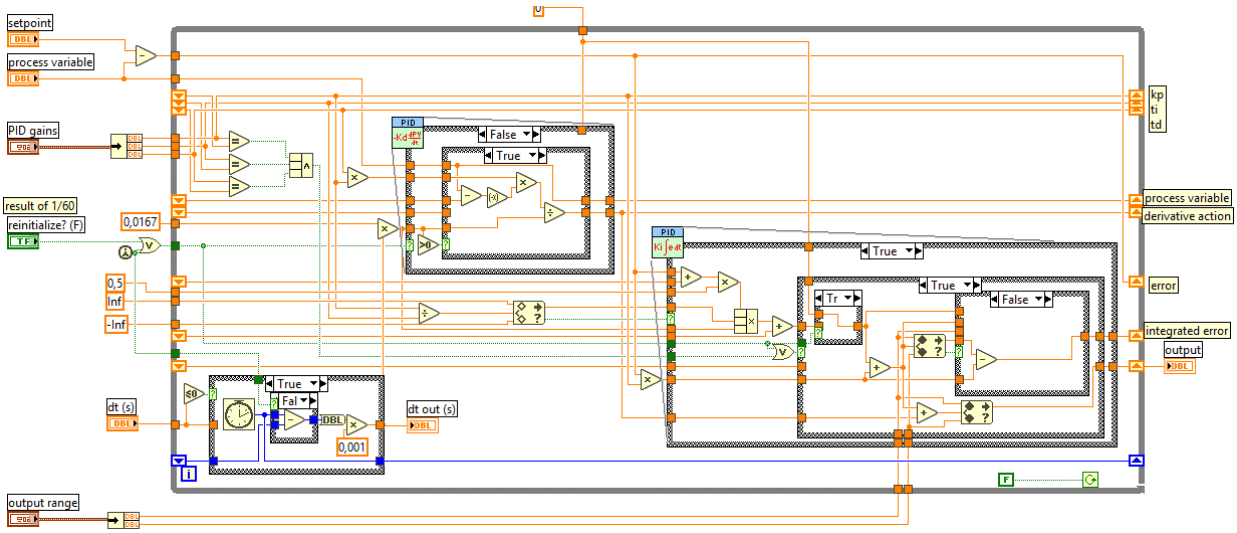


Figure 78: General PID inner core subVI

6.3.3 Position PI Controller

The position controller is based on the same inner core PID controller as for the WOB controller. The outer block diagram is presented in Figure 79. The derivative effect is however sat to zero, as explained in the position control design in subsection 5.3. The filtered IMU data is used to calculate the toolface angle (not shown) and is input to the inner core PID control block. The toolface SP is determined by the calculated azimuth (not shown). The PI output, or the top drive position SP, is limited to $TD_{SP} \in [-3600, 3600] deg$. This is done for two reasons; (1) to be able to correct for the maximum position/angle error the toolface can have after homing, which is +-180 deg, and (2) to prevent the controller to end up on the outer edge of the two PID saturation points of +-360 deg. This will give the controller at least +-180 deg correction space in the worst case scenario. In theory, it is not necessary with upper/lower limits of the PID output for this controller. This is mainly done based on the principle of it, and to prevent the PID to output an infinitely high position SP that can twist of the pipe in a tight borehole.

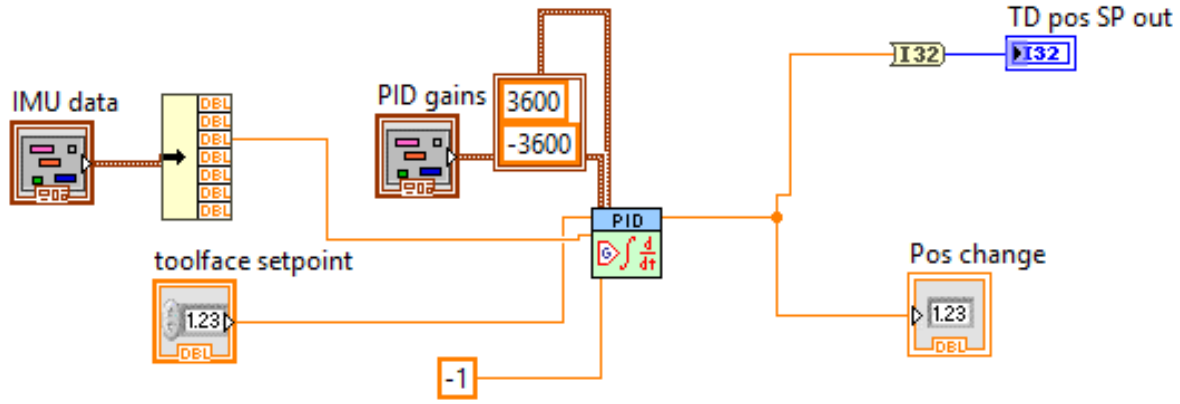


Figure 79: Position controller subVI

6.4 Manual Preparation for Autonomous Mode

Before the autonomous script can be run, two important manual preparation steps must be conducted; (1) manual calibration of the magnetometer after instalment in the BHA and (2) IMU-frame-BHA-bend-frame alignment and rig leveling. Both will be explained in the next subsections.

6.4.1 Magnetometer Calibration

The manual magnetometer calibration is a necessary step after each re-installment of the BHA stack. This is due to the internal distortion sources, where the magnetometer is dependent on the position and orientation of each distortion source. Since not all of the threaded BHA subs is mechanically fixed with an end-stop, the orientation of the sub might change after each installment based on how hard you torque it up. Also, the EMM must run continuously during the calibration process, as the alternating distortion from electric motor must be included in the calibration. The calibration process is programmed in an external script (not a part of the autonomous script), and the calibration front panel is presented in Figure 80. The calibration process goes through 3 manual calibration steps. The steps is presented below:

1. **Bias correction:** The BHA is rotated in a eight-figure motion around all axis to find the resolution (max/min values). The bias offset is automatically calculated and stored as constants.
2. **Orientation correction:** The bias corrected 3-dimensional magnitude plot of the magnetometer is manually rotated using rotation matrices in the program to prepare for linear scaling.
3. **Linear Scale correction:** The BHA is again rotated in a eight figure motion around all axis to find the resolution. The scale correction is automatically calculated and stored as constants.

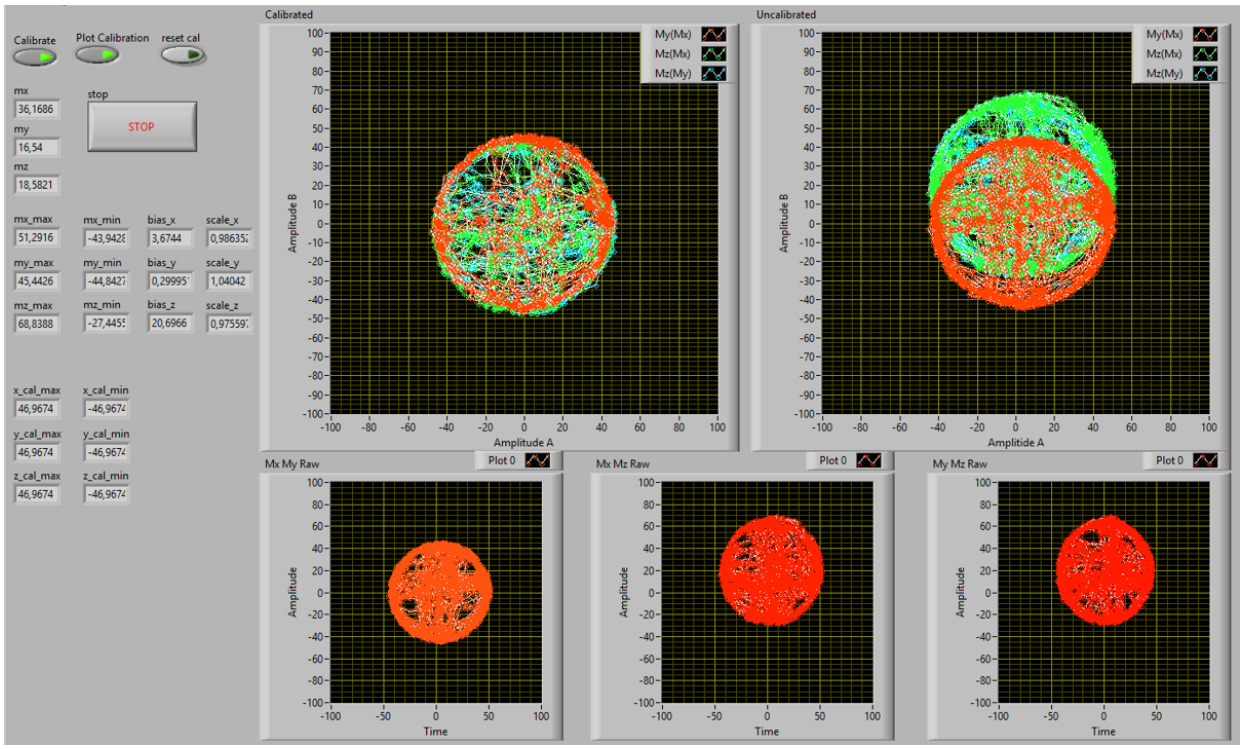


Figure 80: IMU calibration front panel in the middle of a calibration process.

6.4.2 Frame Alignment and Rig Leveling

The manual preparation for the rig leveling and alignment consists of 2 steps. An external compass and leveling bar is used for this process. The steps is explained below:

1. Level and align the rig with the gravitation and magnetic north direction. This is mainly done to reduce vibrations from when the rotary stack i rotated, but also to simplify the manual frame alignment that will be described further down.
2. Level and align the rock samples with the gravitation and magnetic north direction. This is done to get an accurate position tracking as the inertial frame is defined in the top center of the rock samples.

The IMU-frame-BHA-toolface-frame alignment is done in Labview script. The front panel is presented in Figure 81. The reason for this manual preparation step is the random orientation of the BHA toolface relative to the sensor sub after installment. When the BHA stack is torqued up, the bend-toolface will have a random orientation relative to the sensor sub, dependent on how hard the threads are torqued up. It will naturally be within a given offset-angle-area, but misalignment will influence the position control results to some extent. The alignment is done manually by the trial-and-error method. The full rotary stack is installed on the rig including BHA and sensor card and. Rig must be leveled and aligned, and sensor card must be calibrated. The steps is presented below:

1. **Body frame leveling:** Rotate the IMU body frame to level the IMU with the gravitation using Euler angle rotations in the program. The goal is to have the accelerometer bulls-eye placed in the xy-origin.
2. **Toolface controller:** Activate toolface controller. The BHA is rotated till the IMU X-axis is pointing in the magnetic north direction. If the rig is aligned, this will be equivalent to the front direction of the rig. The toolface will now have a random orientation from the magnetic north direction.
3. **Body frame rotation:** Manually rotate the IMU body frame using Euler angle rotations around the z-axis to align with the toolface frame. The toolface controller will automatically re-orient after each rotation. Trial-and-error until the toolface is pointing in the rig-front direction.

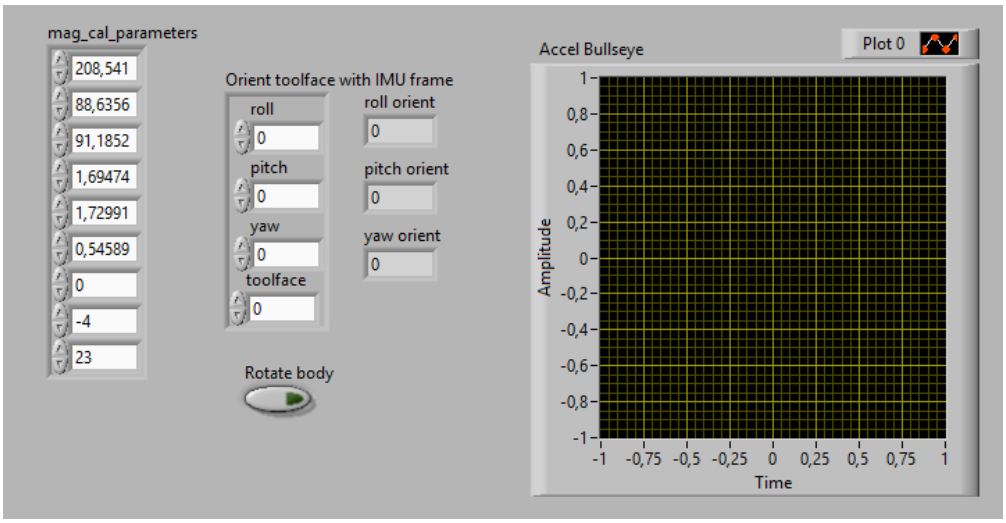


Figure 81: IMU and BHA toolface frame alignment front panel.

The visual determination of when the toolface is properly aligned with the rig/rock-front direction will naturally be a source of error. However, the toolface error will mainly be a problem in the beginning of the directional phase, as the position controller will correct for any deviation in azimuth by changing the toolface SP. That is, as long the accuracy of the manual frame alignment is within certain limit, e.g. +- 10deg, the position controller should be able to correct over a 50 cm long directionally drilled section.

6.5 Initialization Sequence

The first sequence of the autonomous competition script is the init sequence, as described in the general code setup in subsubsection 6.1.4. The main purpose of this sequence is to setup all Modbus, CANopen and VISA serial communication to avoid this from being ran each iteration of the operation while loop. For visual reference, see the competition script block diagram in Appendix C. The init code is summarized below:

- HM Modbus init
- PM Modbus init
- TD CANopen init
- EMM EPOS init
- IMU VISA serial init
- IMU reference path calculation
- Define state machine states

6.6 Operation Sequence - State Machine

The operation sequence is the main sequence in the autonomous script. This is where the actual drilling operation is carried out. A general overview is presented in Figure 82. The sequence mainly consists of a while loop that runs while true with a state machine placed inside. Tasks and instances (communication protocol setup) from the init section is stored in shift registers for the while loop to use for each iteration. Sensor outputs, safety logic and setpoint selection is generally placed before the state machine. It is important that these three items are ran at each iteration of the while loop for process control accuracy and general safety reasons. If a critical error occurs, the safety logic can automatically terminate the script in the middle of the operation.

The state machine consists of 9 states, where each state is programmed to perform a specific task with a specific termination criteria. The tasks can for example be to tag the rock, drill vertically or drill directionally. Termination criteria can be used to either terminate the script when the operation is finished, or to switch state when a current task is completed. The state machine takes in setpoints, sensor outputs and safety logic which is used for process/state control. The process control (PID control or constant SP) controls the setpoints for the relevant motors that is being used in the current state. The motor SP is written to each motor after the state machine. Plots and indicators takes in all relevant data (outputs, inputs, setpoints) for visual presentation on the front panel. All operation tasks, state status, safety concerns and changes made by the autonomous system is being logged and printed on the front panel.

Before the state machine states is introduced, the boolean safety logic and SP selector must be described. Both is used as an input to the state machine and can be found in the next two subsections.

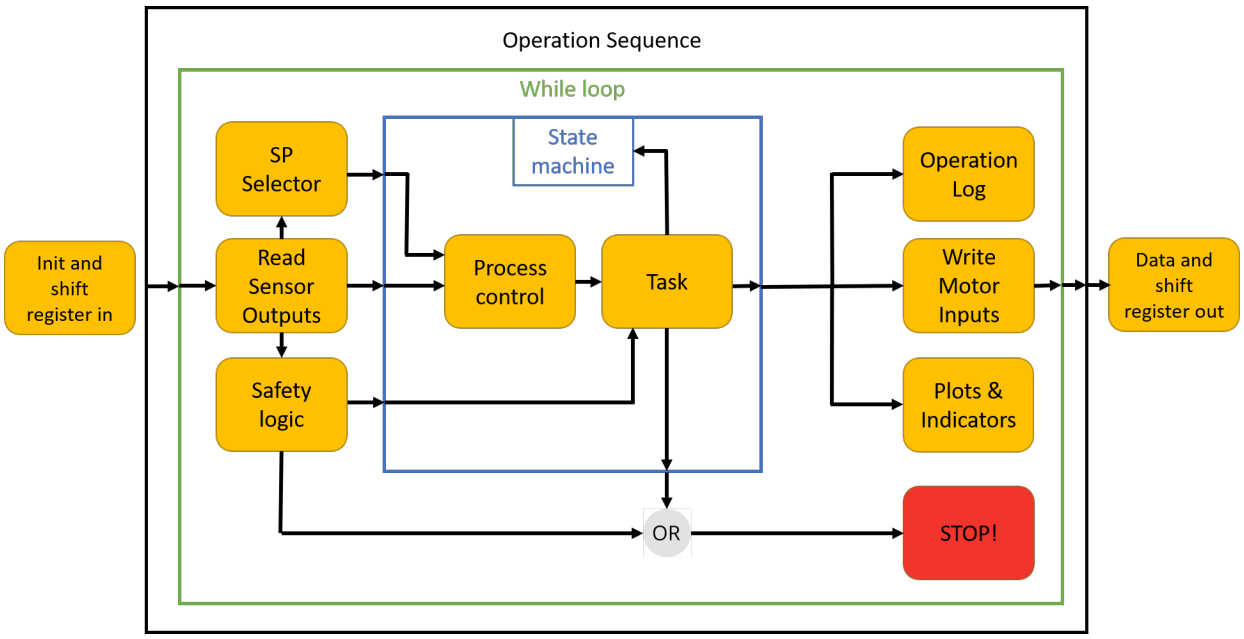
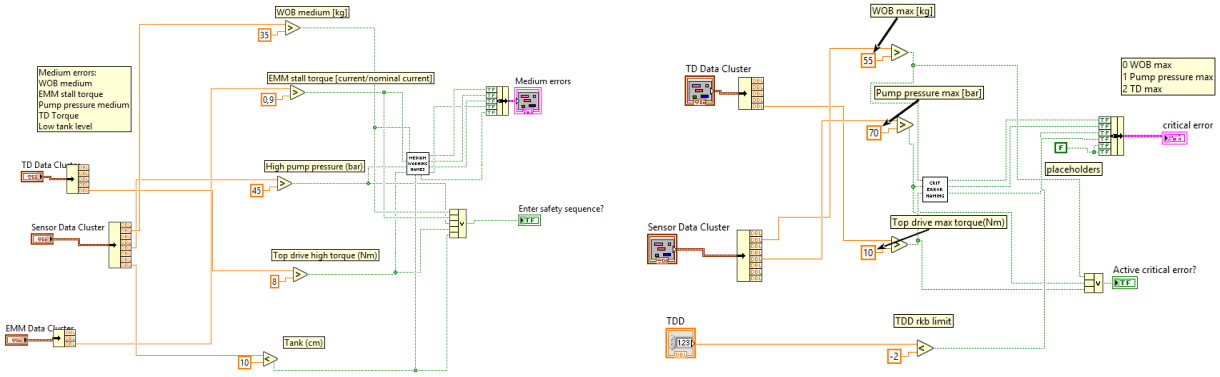


Figure 82: Operation sequence overview.

6.6.1 Safety Logic

A high priority for the 2019 control system is to improve the operation safety features. In the 2018 control system, boolean safety logic was implemented to terminate the script if a critical error occurred. The limitation with this logic is that the script will terminate even if the problem is caused by an error in the program, temporarily sensors failure, or even if the error can be fixed by quick manual intervention. To cope with this, the team have implemented a two-layered safety logic consisting of a medium and critical layer. The purpose of the medium error layer is to catch an error before it becomes a critical one and give the program a chance to jump to a steady state in the wait for the error to clear. This steady state is called the safety sequence and is presented as a part of the state machine later in this section. The purpose of the critical layer is to terminate the script immediately. The two layers runs in parallel, which means that an error can first be caught by the medium layer, and if the error continues to escalate the critical layer will catch it and terminate the script. The layer limits is user defined and is for each layer defined below:



(a) Medium safety layer.

(b) Critical safety layer.

Figure 83: Two-layered boolean safety logic block diagrams. Full version can be found in Appendix C.

Medium layer:

- $WOB > 35$ [kg]
- $T_{EMM} > 0,9$ [nominal]
- $P_{PM} > 45$ [bar]
- $T_{TD} > 8$ [Nm]
- $h_{tank} < 10$ [cm]

Critical layer:

- $WOB > 55$ [kg]
- $P_{PM} > 70$ [bar]
- $T_{TD} > 10$ [Nm]
- $h_{tank} < 5$ [cm]
- $TVD < -2$ [cm]

6.6.2 Setpoint Selector

As explained in subsection 5.2, the purpose of setpoint selector is to maximize the performance of the weakest actuator in the system, which is in the torque of the downhole power section (EMM), by changing WOB SP. The torque readings for this actuator and the stability of the downhole reaction torque on the bit is considered as too unreliable for it to be used as input in a torque PID controller. The setpoint algorithm was presented in Figure 54, and the user defined setpoint limits is summarized in Table 15.

Table 15: Setpoint selector user defined settings.

Increase WOB SP	Decrease WOB SP	Current WOB SP	Incremental change [kg]	Evaluation interval [s]
$T_{emm,nom} < 0,7$	$T_{emm,nom} > 0,85$	$0,7 < T_{emm,nom} < 0,85$	$+-1$	10

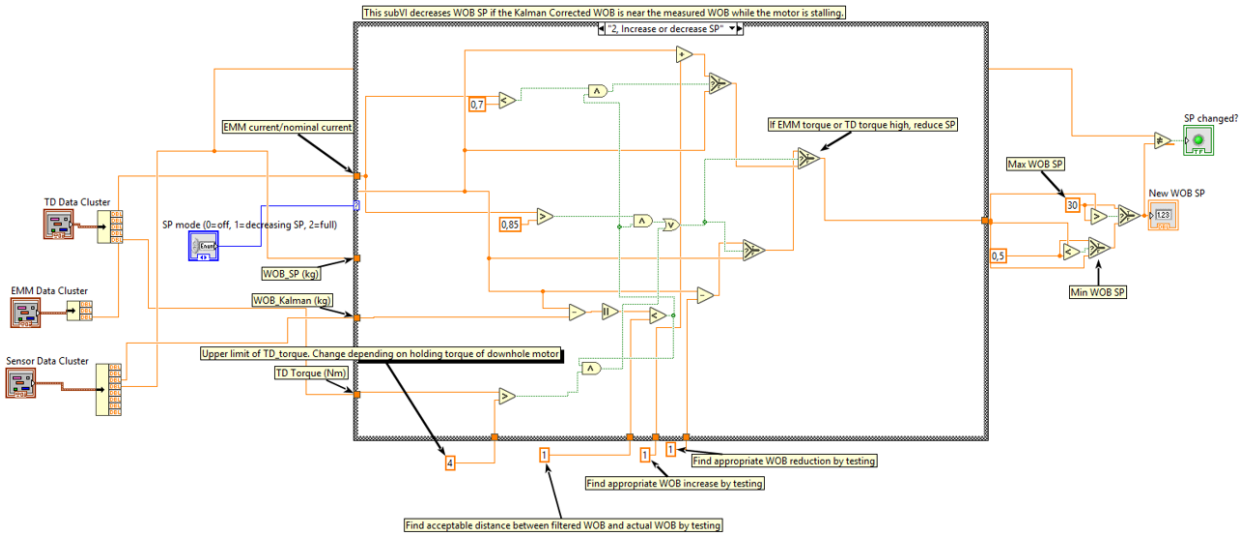


Figure 84: Setpoint selector block diagram. Full version in Appendix C.

6.6.3 State machine overview

In Figure 85, an overview of the 9 states in the state machine is presented. The state machine is build up by a linear sequence of states divided into a vertical- and directional phase, with the ability to jump to a steady state if an error occurs during one of the drilling states. In the next subsections, each state will be described individually. The block diagram for can be found along with the competition script in Appendix C. A summary of control mode, termination criteria and next action for each state can be found in Table 16.

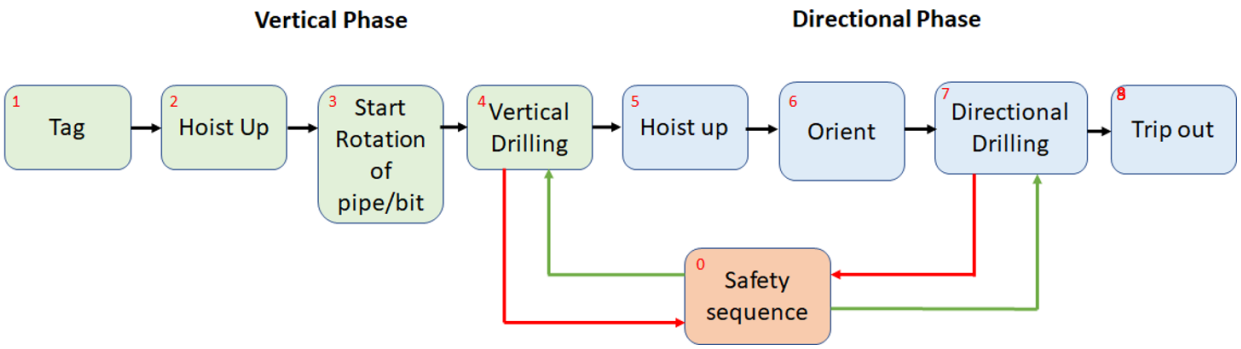


Figure 85: State machine overview

Table 16: State machine control mode, termination criteria and next action summary.

State	Control mode	Termination critera	Next action
1. Tag	Constant HM SP	$WOB > 8$ [kg]	2. Hoist up
2. Hoist up	HM SP	$t > 6$ [s]	3. Start rotation
3. Start rotation	Constant TD, EMM PM SP	$rpm = rpmSP$	4. Vertical drilling
4. Vertical drilling	WOB control	$TVD > 4$"	5. Hoist up
5. Hoist up	HM SP	$t > 6$ [s]	6. Orient
6. Orient	Toolface control	$toolface = toolfaceSP$ [deg]	7. Directional drilling
7. Directional drilling	WOB control Toolface control Position control	$TVD > 24$"	8. Trip out
8. Trip out	Constant HM SP	$TVD < Tagpos$ [cm]	Terminate script
0. Safety sequence	HM SP	Error = False	Return to previous state

6.6.4 Tag

The first state in the vertical phase is tag. The HM hoists down the rotary stack with constant velocity towards the rock surface until a user defined WOB limit gets exceeded. The tag WOB limit is currently set to 8 kg. All actuators besides the HM is set to zero. When the termination criteria is reached, the HM SP is automatically set to zero to stop the hoisting movement, and the HM position is saved and defined as tag position (will later be used during trip out). The state machine proceeds automatically to next state.

6.6.5 Hoist Up for Vertical Drilling

The second state in the vertical phase is hoist up. After the bit has tagged the rock, a short pre-programmed hoist-up-hoist-down movement is performed to create a small clearance between the bit and the rock before pipe/bit rotation is started. The hoist-down movement is done to zero the WOB measurement due to the offset dependency between the load cell and the hoisting direction. The WOB is currently zeroed when the ball screw pushes on the load cell while hoisting down. The termination criteria is time dependent and is currently set to 6 seconds; 4 second hoist-up and 2 second hoist-down. When the termination criteria is reached, the HM SP is automatically set to zero, and the state machine proceeds to next state.

6.6.6 Start Rotation

The third state in the vertical phase is start rotation. Pump motor, top drive and electrical mini motor is sat to a constant rpm SP to start the water circulation, pipe rotation and bit rotation respectively. The termination criteria is true when the rpm has reached or is within an defined error-limit from the rpm SP for each motor. When the termination criteria is reached, the state machine proceeds to next state.

6.6.7 Vertical Drilling

In the vertical drilling state the pipe and the bit is rotated to drill an over-gauged hole (due to the bend). The state runs on WOB control with automatic WOB SP selection. The PM, EMM and TD runs on constant rpm SP based on experience from testing. The termination criteria is set to 4" according to the competition guidelines. When the true vertical depth (TVD) has reached the target depth, the HM SP is automatically set to zero, and the state machine proceeds to next state.

When in vertical drilling state, the state has the opportunity to jump to the safety sequence in the wait for a medium error to clear. The state machine will automatically jump back to the vertical drilling state when the error is cleared.

6.6.8 Hoist Up for Directional Drilling

The first state in the directional phase is hoist up. After the target depth for vertical drilling is reached, a short pre-programmed hoist-up-hoist-down movement is performed. As for the hoist up for vertical drilling, the termination criteria is time dependent and is currently set to 6 seconds; 4 second hoist-up and 2 second hoist-down. When the termination criteria is reached, the HM SP is automatically set to zero, and the state machine proceeds to next state.

6.6.9 Orient Toolface

The second state in the directional phase is orient. The state runs on toolface control to orient the BHA toolface in the magnetic north direction before the directional drilling can start. The PM SP is set to zero to reduce vibrations and movement of the BHA due to the hydraulic force the bit jet nozzles acts on the BHA when the bit is not tagged. The toolface angle is evaluated every 10 seconds and the termination criteria is when the angle is within +- 1 deg from the toolface SP. The toolface SP is set to 0 is an initial value, but will later be changed by the position controller if the azimuth deviates from the north centerline. When the termination criteria is reached, the state machine proceeds to next state.

The bit rotation runs continuously throughout the orientation state to cope with the internal alternating distortion from the electrical motor. If the motor is stopped, the distortion will have a random orientation relative to the sensor card and thereby result in a random error in measured toolface angle. However, if the EMM runs continuously on a constant SP, the Kalman filter can be tuned to filter out the alternating distortion as process noise.

6.6.10 Directional Drilling

In the directional drilling state, a slide-while-drilling operation is carried out. The bit is rotated by the downhole power section while the state runs on WOB control with automatic WOB SP selection. To avoid wrong toolface SP correction by the position controller in the beginning of the directional well, the state will run on toolface control until an inclination of +4 deg is reached. The accelerometer vector will in the beginning fluctuate around the xy-origin, resulting in a random magnetic heading. That is, if the position controller is used, the toolface correction from the azimuth calculation will in the worst case scenario try

to correct for up to +- 180 deg. This is solved by using the the toolface controller for the first part of the directional well.

The termination criteria is set to 24" according to the competition guidelines. When the TVD has reached the vertical target depth of the directional well, the HM SP is automatically set to zero, and the state machine proceeds to next state.

As for the vertical drilling state, the directional drilling state has the opportunity to jump to the safety sequence in the wait for a medium error to clear. The state machine will automatically jump back to the directional drilling state when the error is cleared.

6.6.11 Trip Out

The trip put state is the last state in the state machine. After the target depth of the directional drilling state is reached, a slow hoist-up movement is performed to retrieve the tool from the borehole. The state runs on constants HM SP, EMM SP and PM SP to avoid the bit from getting stuck in the formation. The termination criteria is set to the tag position. When TVD reaches the tag position defined in the tag state, the script terminates automatically. The drilling operation is completed and the script jumps automatically to the close sequence, which will be described in next section.

6.6.12 Safety Sequence

The safety sequence is a steady state the state machine can jump to if an medium error occurs. The state is activated by the medium error safety layer and is constantly under evaluation during the vertical- and directional drilling state. First, state performs a quick hoist-up-hoist-down movement equivalent to the hoist up state, before the HM SP is set to zero. The state will return to the previous state if and only if all medium errors are cleared. If the state is activated by a high torque or WOB error, the WOB SP will automatically be decreased by 1 increment. This is done to reduce the risk of a limbo effect, where the state machine jumps in and out of the safety sequence. If the error is due to low tank level, the program will stay in the state until the level has reached a user defined level. This level is currently set to 30 cm. A counter keeps track of what errors that have activated the safety sequence. This is currently not used for anything, but can later be used to adjust the error and operation limits.

If the error does not clear automatically, the operator will have the opportunity to manually intervene. This is only recommended if the error is due to technical issues with a non-important sensor or similar.

6.7 Close Sequence

The main purpose of the close sequence is to end all tasks and instances that was initiated in the init sequence. Before the tasks are closed, all actuator Sp is set to zero for safety reasons. This is especially important if the while loop in the operation sequence is terminated by a critical error before it is able to complete all states in the state machine. For visual reference, see the competition script block diagram in Appendix C. The close code is summarized below:

- HM Modbus close
- PM Modbus close
- TD CANopen close
- EMM EPOS close
- IMU VISA serial close

6.8 Export Data Sequence

The last sequence of the autonomous competition script is the export data sequence. The main purpose of this sequence is to export all drilling data and save it as a text file. The sequence programmed with inspiration from the 2018 automatic data storing code. The sequence runs automatically after the close sequence ends, where a user prompt pop-up gives the operator the opportunity to save the operation data for later analysis. The user prompt pop-ups is presented in Figure 87. This year, the team outputs a lot more data than where a tunnel array out from the while loop stores a 38 column data array with one data point per iteration. The exported system outputs is presented in Figure 86. For visual reference of the export data sequence block diagram, see the competition script diagram in Appendix C. The init code is summarized below:

- HM Modbus init
- PM Modbus init
- TD CANopen init
- EMM EPOS init
- IMU VISA serial init
- IMU reference path calculation
- Define state machine states

0 time [s]	10 HM torque	20 TD rpm SP	30 azimuth [deg]
1 wob_raw [kg]	11 HM pos	21 TD position [deg]	31 heading [deg]
2 wob_filtered [kg]	12 HM ROP [cm/min]	22 TD position SP [deg]	32 inclination [deg]
3 wob_sp [kg]	13 Hole prog [cm/min]	23 TD torque [Nm]	33 roll [deg]
4 P pressure [bar]	14 Tag pos	24 EMM torque [Nm]	34 pitch [deg]
5 P tank [Pa]	15 TVD [cm]	25 EMM mean torque	35 yaw [deg]
6 Flow rate [l/min]	16 PM RMP	26 EMM nom current	36 accel bullseye x
7 Tank level [cm]	17 PM RPM SP	27 x pos [cm]	37 accel bullseye y
8 HM rpm	18 PM torque [Nm]	28 y pos [cm]	
9 HM rpm SP	19 TD rpm	29 z pos [cm]	

Figure 86: Exported system outputs.

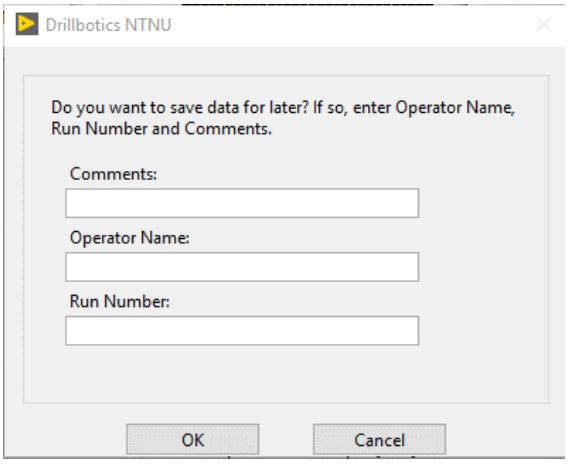


Figure 87: Export data user prompt.

6.9 Graphical User Interface

The graphical user interface, or the front panel, is the human-machine-interface between the operator the control system. The front panel contains all important drilling data presented as boolean, string or numeric values in graphical indicators, plots or prints. The team have focused on designing a clean and simple front panel to makes it easy for the operator to observe and control (if necessary) the drilling process. The front panel is presented in Figure 88 and is generally divided into 3 main panels:

1. **Main panel:** The main panel is located in the center/left side of the front panel and contains the most important indicators, plots and controls such as toolface orientation, position tracking, WOB plot, important measurement indicators, error warning indicators, communication status indicators, operation log and manual stop button. A settings tab is hidden behind the main panel if the operator wants to customize the drilling operation.
2. **B panel:** The B panel is located on the top right corner. The tab panel holds all indicators that is not important for the operator to see if not there is a problem with a specific part of the system. The tabs are organized with one tab for each motor and one tab for safety indicators. The safety indicator tab is standard and will automatically show on start up.
3. **C panel:** The C panel, or control panel, is a tab panel that holds all manual controls. As this is an autonomous system, a manual control panel is not really necessary. However, when implementing autonomous systems, the ability to manually override the system is an important safety feature and highly convenient during the system implementation phase. The SP control tab is standard and will function as SP indicators during the operation. In the state control tab, the operator has the ability to activate/deactivate automated SP selection, automated tank filling or manually jump into the safety sequence to "pause" the drilling process.

Figure 88: Control system front panel.

Front panel key features is summarized below, where Figure 88 is used as a reference.

Key Features:

- **Operation log:** Live operation log is one of the new key features in the 2019 control system. Every time the system performs an automated action, the event is logged and printed on the main panel

with a timestamp and a short description. The system logs every time the state machine starts a new state or completes the current state, if the SP selector changes SP, if a warning indicator is activated or if the system jumps into the safety sequence. In Figure 88, the system has logged that the state machine has jumped into the safety sequence due to low tank level. This feature is very important for autonomous systems, especially in the implementation phase, where the operator gets an overview of all autonomous actions performed by the system.

- **WOB control graph:** The WOB control graphs visualizes the SP, raw- and filtered WOB measurement on the main panel. The WOB controller is the main controller for both the vertical and directional drilling phase, and thereby the most important graph for the operator to observe. The SP selector will during both drilling phases perform step responses by changing the WOB SP. If the WOB controller is badly tuned, it can easily be observed on the WOB control graph and manually tuned by the operator in the C panel.
- **Position tracking:** Real time position tracking gives the operator a good overview of the progression of the drilling operation. Before drilling, the operator fills target depth for the vertical section and desired target position for the directional section. An optimized trajectory is automatically calculated and visualized on the position tracking graph. The reference path is used by the position controller to correct for deviation. As explained earlier, only azimuth will be corrected for with the current mechanical design of the directional tool, but the controller is designed to correct for deviation in 6 DOF.
- **Color coded indicators:** The 9 bar indicators on the bottom of the main panel is color coded based on the current risk level of each measurement. The risk is divided into three zones; (1) no risk zone indicated by green color, (2) the risk zone indicated by orange color, and (3) the critical zone indicated by red color. By color coding the risk, the operator will automatically focus on the red indicators in the front panel, resulting in simplified troubleshooting and decreased reaction time if manual intervention is needed. The boolean status indicators is also color coded with red and green to indicate problem/no problem.
- **Automated tab switching:** As explained above, the B panel holds all indicators and plots that is not needed for the operator to see during the main part of the drilling operation. However, when a specific task is carried out that is of the operators interest to follow closely, the B panel will automatically change tab to show the plots/indicators of interest. That is, the B panel can be customized to show specific plots and indicators for specific states. One example of this is the orient toolface state. The B panel will automatically jump into the top drive tab to show a plot of the toolface control action, and stay there during the whole directional drilling phase. If an error occurs and the state machine jumps into the safety sequence, the B panel will change tab back to error warning indicators to show what error that activated the safety state. In Figure 88, the safety sequence is activated and the B panel is changed accordingly.
- **Settings tab:** The settings tab is hidden behind the main panel and gives the operator the opportunity to customize the drilling operation by changing motor setpoint for motors that is not controlled, error limits, SP selector preferences and trajectory plan. The panel also holds the PID tuning and sensor calibration settings. The main purpose of the settings panel is to move all constants and variables that can be of interest for the operator to adjust or tweak on, from the block diagram to the front panel. Block diagrams are generally very difficult the follow, especially for a third party. By moving the customizable part of the code to the front panel, tuning of code gets simplified.
- **Manual override:** In the C panel, the operator has the opportunity to manually override the system if autonomous system malfunctions. This is an important safety feature for newly implemented autonomous systems that lacks of testing. The team have used this feature frequently during the implementation phase.

6.10 Simulation Script

One of the main design considerations for the 2019 control system is to implement a state machine simulations script. The purpose of this is for the team to be able to simulate boolean logic of the state machine without being dependent on the rig being operational. This is important for projects like Drillbotics, where the rig is non-operational 95 percent of the project time due to mechanical upgrading, delivery time for parts and equipment, drive communication setup time and when critical equipment or parts break. A simulation script is also great for implementation of individual controllers that only need parts of the rig, or state machine logic that does not need the rig at all.

The simulation script is programmed as a complete copy of the 2019 control system, where all motor and sensor inputs/outputs are replaced with manually controlled variables. Since the system is mostly unmodeled, the simulation script can only be used to simulate the boolean logic of the state machine, SP selector and safety logic. Block diagrams and front panel can be found in Appendix C. The following functions and controllers are implemented by only using the simulation script, and by including individual motors/sensors outside the rig:

- Setpoint selector
- Toolface controller (TD and IMU)
- Operation log print
- All boolean safety logic

7 Results and Discussion

In this section, the result from calibration, tuning and practical experiments will be presented. This includes IMU calibration, Kalman filter tuning, WOB control testing and tuning, position tracking, toolface control testing, magnetic ranging experiments and state machine testing. The design and implementation the downhole sensors card and the control system in Labview is a vital part of the scope for this project and can also be considered as results. Both can be found in section 4 and section 6 respectively. The individual results will be discussed within each subsection where an overall system analysis will be summarized in subsection 7.8.

Due to technical issues with vital equipment, rig shipping and difficulties with the German customs service, some part of the system remains untested on the rig. Untested parts of the system is summarized below:

- Position tracking during drilling
- Position control during drilling
- IMU Kalman filter tuning during drilling

7.1 Limit Testing

In this subsection the results from the PDM- and EMM performance testing is presented. This is not a part of the scope for this report, where the mechanical team have been responsible for both the design and test procedures. However, as the downhole power section is the weakest actuator in the system, the control system have been designed with this as its main consideration. Therefore, the results from PDM- and EMM performance testing will be summarized below. The results is taken from *Design and Implementation of an Autonomous Miniature Drilling Rig for Directional Drilling* by team members J.M. Montoza, H.E. Helle and U.M. Azam [57].

7.1.1 PDM- and EMM Performance Limit

Table 17: PDM- and EMM performance limits

DH power sector	RPM	Torque [Nm]	Status	Challenges
PDM (1:2)	>500	NA	Not operational	Internal leakage, cyclic wear and high m2m friction where the performance decreases over time.
PDM (4:5)	15	<0,1	Not operational	Internal leakage, internal friction, design-challenges regarding material combination, rotor tolerance and manufacturing method.
EMM	87	0,8	Operational	Relative low torque with high risk of stalling and challenges regarding water sealing around motor shaft.

7.2 IMU Calibration

7.2.1 Accelerometer Calibration

The accelerometer vector is one of two vectors used to described the orientation of the IMU relative to the world frame (North, West, UP). The sensors is factory calibrated, but after soldering of components on to the PCB additional calibration is sometimes necessary. Equations used for accelerometer calibration

can be found in Equation 12 in subsubsection 7.2.1. Below, the equipment used and calibration process is explained.

Equipment:

1. DH sensor card with ICM-20948
2. 3D printed calibration cube
3. A flat table
4. Leveling bar
5. Computer with calibration script

Calibration steps:

1. Level table manually with a leveling bar.
2. Place the sensor card inside the calibration cube as shown in Figure 89 and start the calibration script.
3. Measure the gravitational acceleration for each axis by placing the 6 flat cube sides on the table on at a time. Measure at least 1000 samples per side.
4. Bias offset and scale gain is automatically calculated and the results is presented in Table 18.

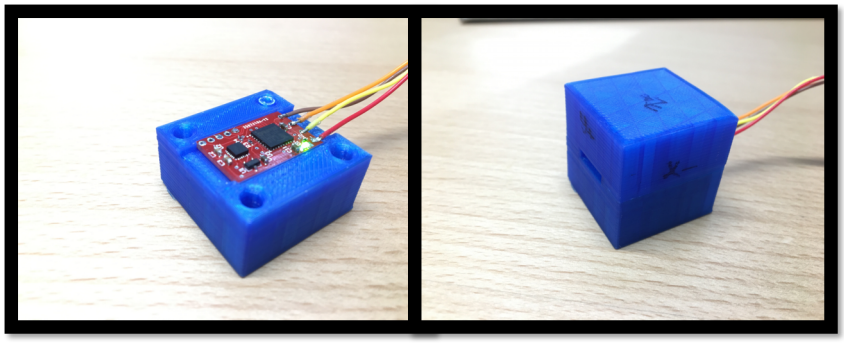


Figure 89: Accelerometer calibration setup using a 3D printed calibration cube.

Table 18: Accelerometer calibration result

Axis	**Max [g]**	**Min [g]**	**Bias [g]**	**Scale**	**Absolute raw [g]**	**Absolute calibrated [g]**
X	$0,995953$	$-1,008914$	$-0,0129614$	$0,997572$	$1,002793$	$1,00033$
Y	$1,00235$	$-1,00567$	$-0,00332146$	$0,996008$	$1,003835$	$0,9995815$
Z	$0,992286$	$-1,01267$	$-0,0203821$	$0,997529$	$1,004273$	$1,001681$
XYZ avg	$0,996863$	$-1,009083$	$-0,012221653$	$0,9970363$	$1,003634$	$1,000531$

The calibration results show that the factory calibration is pretty good. By looking at the absolute values of the measured gravitation for each axis, where 1g is the benchmark, the accuracy after calibration have increased by one decimal. Factory calibration shows an accuracy of 2 decimal, while the re-calibration shows an average accuracy of 3 decimals. The factory calibration will work perfectly fine and is more than accurate enough for this application. However, based on the increased accuracy after re-calibration, the calculated bias and scale correction will be used and implemented as constants in the program.

7.2.2 Gyroscope Calibration

The gyroscope is not used directly in the position tracking algorithm due to the problems with drift. However, the gyro is used to confirm downhole rotation and to evaluate the toolface angle rate during toolface control. The factory scale is assumed to good enough for this application. Hence, only bias offset will be corrected for. Equations used for gyroscope calibration can be found in Equation 13 in subsubsection 7.2.2. The calibration procedure consists of measuring the stationary angle rate for each axis. The sensor card is locked in a stationary position to avoid movement or rotation. The mean value is measured over 1000 samples for each axis. The result is presented in Table 19.

Table 19: Gyroscope calibration results

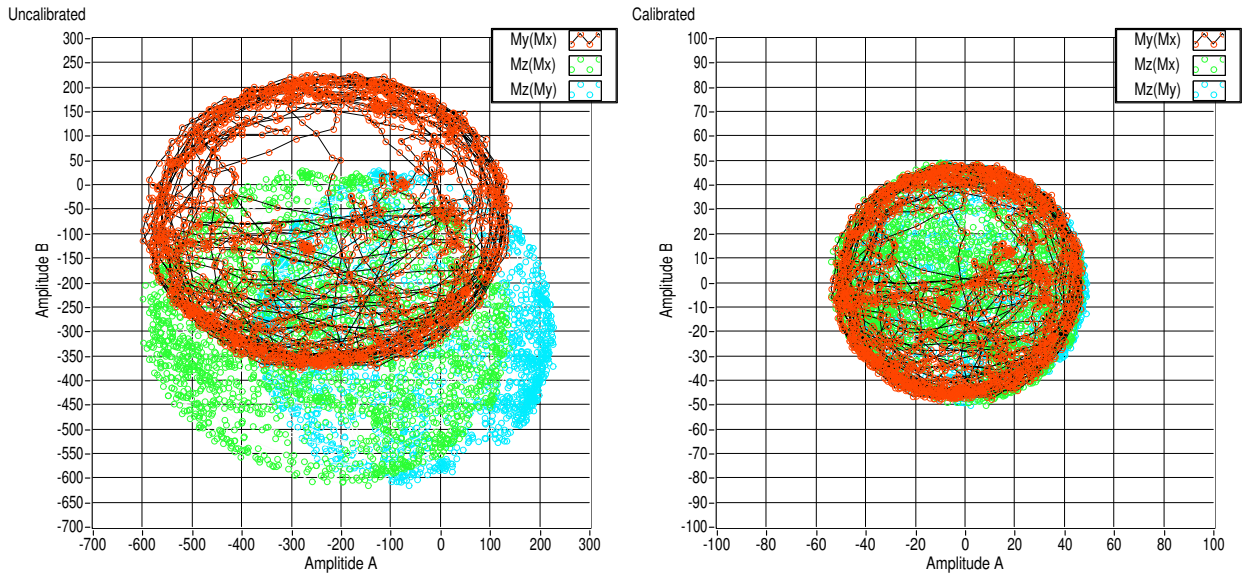
Axis	**Mean [deg]**	**Bias [deg]**	**Mean after calibration [deg]**	**Accumulated drift [deg/min]**
X	0,54120	0,27060	0,00689	0,339863
Y	-0,48761	-0,23587	-0,243805	-0,913121
Z	-0,30714	-0,15357	0,00054	0,0860555

The calibration result clearly shows the problem with gyro drift. The improved mean value will still cause drift over time. This can especially be observed for the y-axis, where the worst case scenario can lead to a drift of 1 dpm. The drift is mainly a result of the integration of near-dc bias instability and an effect called angular random walk (ARW) [58]. Since the gyro will not be used in the main position tracking algorithm, the calibration will not be analysed any further.

7.2.3 Magnetometer Calibration

The magnetometer vector is the second of two vectors used to described the orientation of the IMU relative to the world frame (North, West, UP). As explained in the theory section, the magnetometer is highly effected by hard- and soft iron distortion sources and needs to be re-calibrated after each installment in the BHA. Equations used for magnetometer calibration can be found in Equation 14 in subsubsection 7.2.3. Below, the equipment used and calibration process is explained. The magnetometer calibration will be divided into two part; (1) calibration outside the BHA and (2) calibration inside the BHA. Both is explained below:

Calibration outside the BHA:
In this experiment, the magnetometer is calibrated outside the BHA with minimal external distortion sources. The sensor card is rotated in an eight-figure motion around all axis to find the resolution of each axis. A bias offset is calculate to center around zero and a linear scale correction is calculated to scale the 3-dimensional magnitude plot of the rotated sensor into a perfect sphere. The results in presented in Figure 90.



(a) Raw

(b) Calibrated

Figure 90: Magnitude plots of magnetometer calibration outside the BHA with low magnetic distortion.

The result shows that the magnetometer is mainly affected by hard iron distortion which results in a bias offset from the origin. After calibration the 2-dimensional magnitude plots of each axis pair forms perfect equally large circles centered in the origin. In 3D, this will resemble to a sphere shape which is the overall benchmark of a perfect calibrated magnetometer. To summarize, the result shows that the calibration algorithm works for calibration outside the BHA.

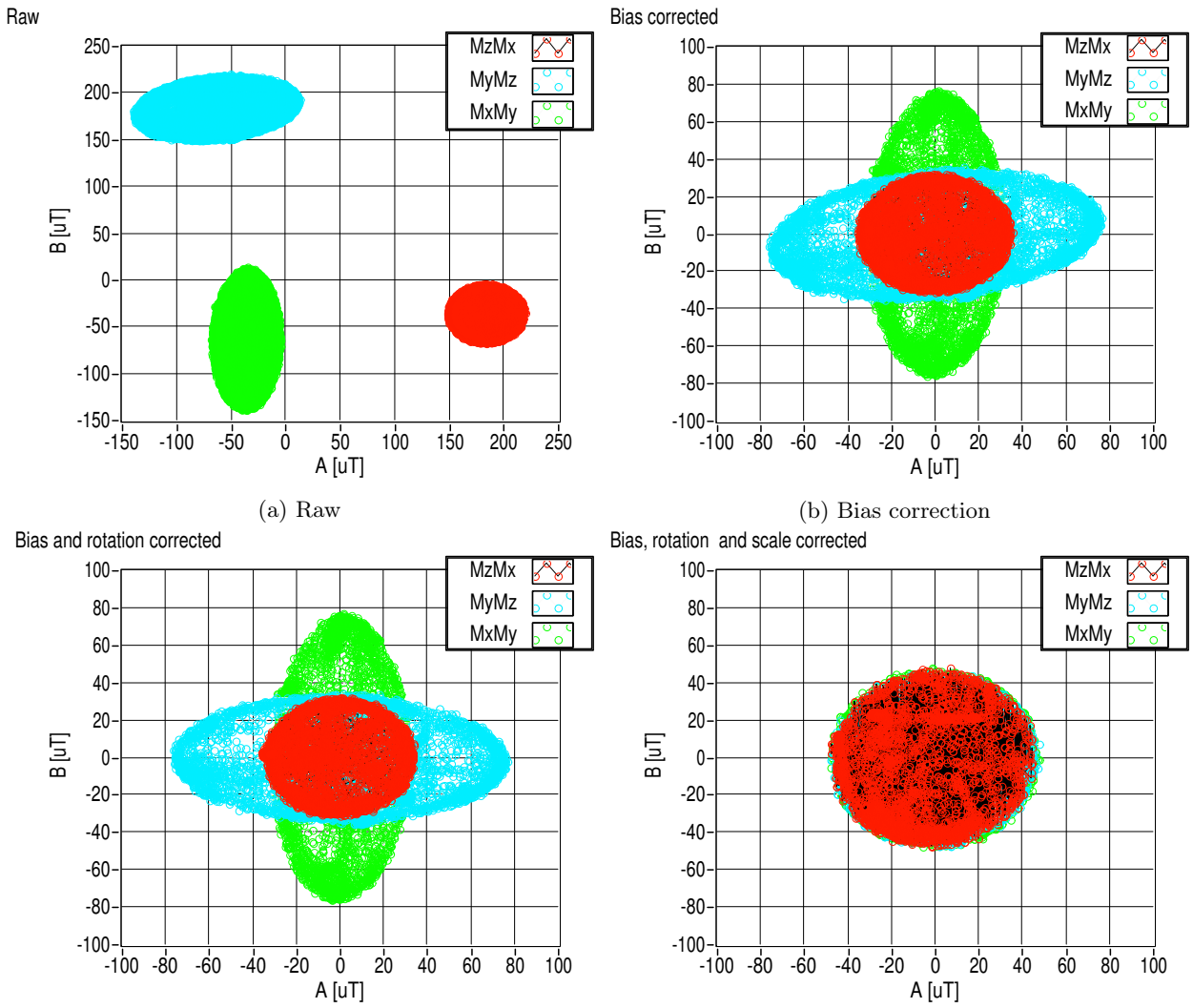
Calibration inside the BHA:
In this experiment, the magnetometer is calibrated inside the BHA with high internal distortion. To cope with the expected soft iron distortion, the calibration process is divided into 4 steps. A complete equipment list and calibration process is presented below. The results are presented in Figure 91 and will be used as a reference.

Equipment:

1. DH sensor card with ICM-20948
2. Full BHA stack including drillpipe
3. Computer with calibration script

Calibration steps:

1. Rotate the BHA in a eight-figure motion around all axis to find the resolution (Figure 91a). The result shows high hard iron distortion (offset from origin) and soft iron distortion (ellipsoidal stretch).
2. Calculate and correct for bias offset (Figure 91b). The magnitude plots is now centered in the origin.
3. Manually rotate the IMU body frame by using Euler angle rotations in the calibration script to correct for soft iron misalignment. Before linear scaling can be calculated, all ellipsoidal magnitude-pair that is not aligned with the x, y or z axis must be rotated. The result in Figure 91b shows that the yz-pair has a small tilt of approximate 4 degrees. The IMU body frame is rotated -4 degrees around the x-axis to correct for this. The result of the rotation is shown in Figure 91c.
4. Again, rotate the BHA in a eight-figure motion around all axis to find the resolution. Calculate and correct for linear scaling. The result is presented in Figure 91d. After scaling, the IMU body frame is rotated back to its original orientation using $R_x(4)$.



(a) Raw

(b) Bias correction

(c) Orientation correction

(d) Scale correction

Figure 91: Magnitude plots of magnetometer calibration inside the BHA with high magnetic distortion.

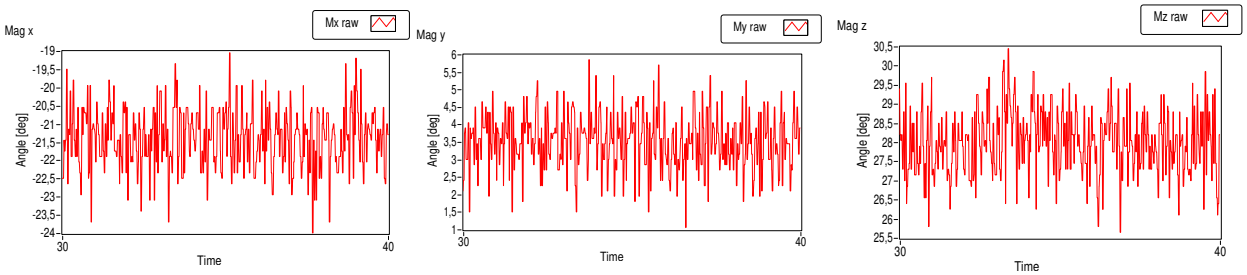
The result shows that the calibration algorithm is highly effective for calibrating a magnetometer effected by hard- and soft iron distortion inside the BHA. The orientation correction step is left out in most magnetometer calibration algorithms in order to simplify the process. However, when soft iron distortion is expected (ferromagnetic materials like iron), this step is critical to avoid loss of output resolution. A visual presentation of the calibration data is necessary for it to be accurate, like the one in the magnetometer calibration script. To summarize, the result shows that the calibration algorithm also works for calibration inside the BHA.

7.3 IMU Kalman filtering

Below, the Kalman filtering of IMU measurements and filter tuning results will be presented. In *Drillbotics - design report*, the team analysed the power spectral density (PSD) of the IMU measurements and determined that the measurement noise can be assumed to be white [9]. The subsection is divided into 3 parts; (1) magnetometer filter and tuning process results, (2) accelerometer filtered results and (3) IMU body frame filtered results. Since the tuning process is equal for both the magnetometer and accelerometer it will only

be described for the magnetometer.

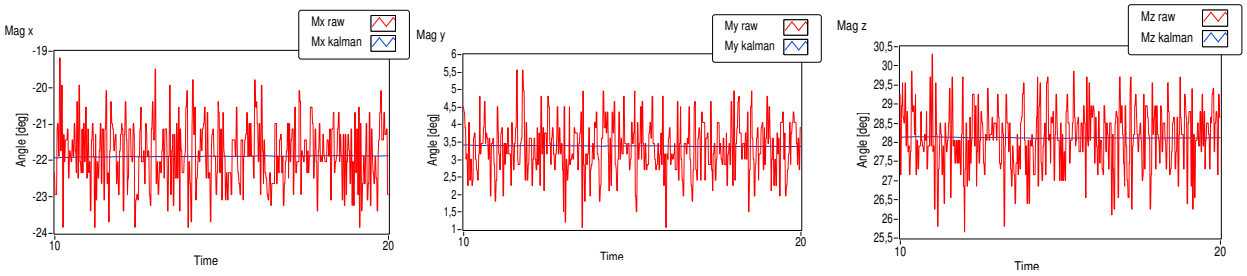
Magnetometer: Kalman filter and tuning
First, the raw magnetometer data for each axis is measured to calculated the stationary measurement noise. The results is presented in Figure 92. The variance is measured to be approximate 0,7 and is used in the diagonal measurement noise covariance matrix (R) as a part of the second-order statistic noise model in the control system.



(a) Mag x raw [uT]: Var=0,72 (b) Mag y raw [uT]: Var=0,68 (c) Mag z raw [uT]: Var=0,73

Figure 92: Raw magnetometer measurements with an average noise variance of $Var = 0,7$.

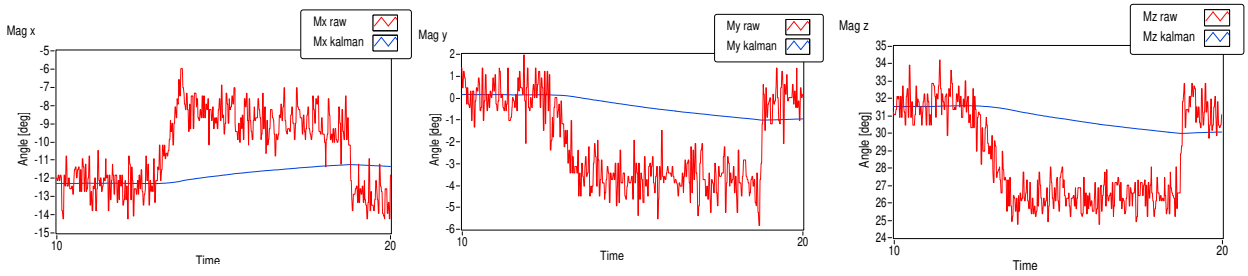
In Figure 93, the Kalman filter is applied according to the stationary model presented in subsection 5.3. The calculated variance is used for measurement noise (R) and the diagonal of the process noise covariance matrix (Q) is currently set to zero. The result shows that the filtered signal is centered in the mean of each axis and that all high-frequent noise is filtered out. If the Q is set to 1 on the diagonal, the estimated Kalman state will resemble to the raw data and thereby result in an unfiltered response. This is not shown. However, the sensor card is kept stationary in this example and thereby not affected by process noise.



(a) Mag x filtered (b) Mag y filtered (c) Mag z filtered

Figure 93: Filtered magnetometer measurements with R=Var(i) and Q=0 for $i \in [x, y, z]$.

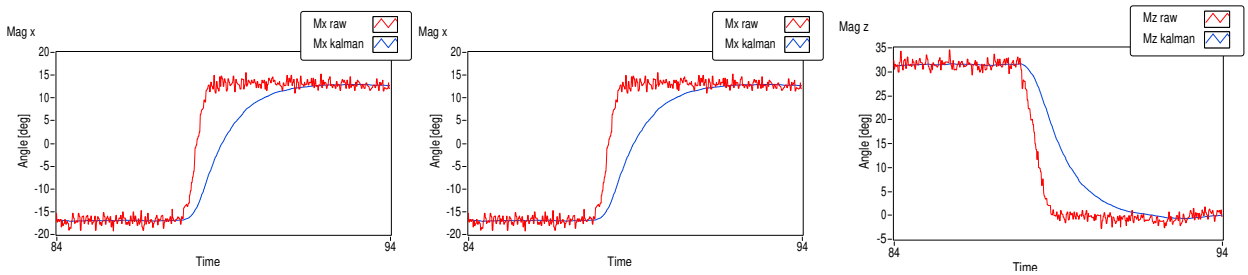
In Figure 94, a step response is performed which shows that the filtered signal is not able to follow the true measurement. This is mainly due to two reasons; (1) the system is un-modeled and (2) the process noise covariance matrix is set to zero. The un-modeled part of the system does not consider the system input (the angular rotation rate), which means the Kalman state estimation is completely relying on the system output (raw magnetometer measurement) in order to estimate the next state. This results in a time-delay dependent on the tuning of the process noise covariance matrix (currently set to zero). The filtered effect will somehow resemble to the effect from a regular low-pass filter. Currently the filter is tuned to not consider any process noise, and hence, the filtered signal is not able to follow a step response.



(a) Mag x filtered step (b) Mag y filtered step (c) Mag z filtered step

Figure 94: Filtered magnetometer step response with R=Var(i) and Q=0 for $i \in [x, y, z]$.

By tuning the diagonal process noise covariance matrix, the Kalman filter can output a desired response similar to an over-damped PID controlled step response. The expected time delay is not an issue when the objective is to calculate the orientation of the directional tool during an up to 3 hour long drilling operation with an expected ROP of less then 1 cm/min. In Figure 95, the tuned filter response is presented where the process noise variance is set to $Q = E - 4$. The filtered step response shows a slightly over-dampened state estimation with approximate zero high frequent noise. This specific tuning is mainly for demonstration purpose to shown a desired tuned step response result. For drilling, the filter will be tuned even "slower" to cope with the cyclic process noise from the EMM.



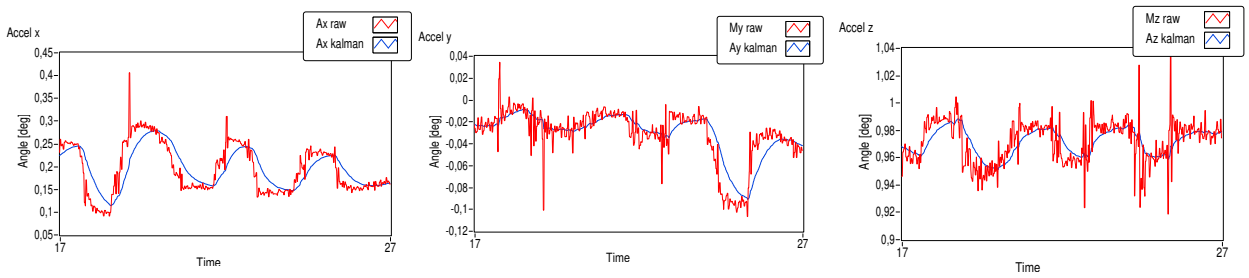
(a) Mag x tuned (b) Mag y tuned (c) Mag z tuned

Figure 95: Tuned magnetometer step response with R=Var(i) and Q=e-4 for $i \in [x, y, z]$.

Accelerometer: Tuned Kalman filter

As explained in the intro, the Kalman filter tuning process for the accelerometer is similar as for the magnetometer. First the raw measurement variance is calculated to be approximate $Var = E - 5$. Then the process noise covariance matrix is tuned to be $Diag(Q) = E - 8$. The filtered step response is presented in Figure 96. The result shows high frequent noise is filtered out on the estimated state. The main difference between the Kalman filter tuning for the accelerometer and magnetometer is the tuning for process noise. High frequent vibrations will be the main source of process noise for the accelerometer, where tuning of the filter must be conducted during drilling.

Since the electrical swivel had technical issues before the rig was lost to the German customs service, tuning of accelerometer process noise has not been conducted during drilling. However, based on the tuning principles presented above, the process noise covariance matrix would have been tuned accordingly "slow" to cope with the noise and to stay in the zero mean.

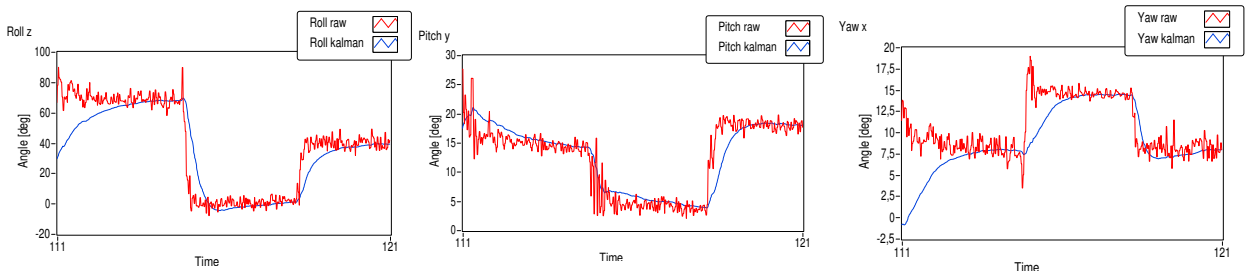


(a) Accel x tuned step (b) Accel y tuned step (c) Accel z tuned step

Figure 96: Tuned accelerometer step response with R=e-5 and Q=e-8 for $i \in [x, y, z]$.

IMU body frame: Tuned Kalman filter

As explained in subsubsection 5.3.4, the orientation of the IMU body frame relative to the world frame (north,west,up) is calculated from a double cross product by using the magnetometer- and accelerometer vector, and stored in a 3x3 rotation matrix R_w^b. By using the rotation matrix with the Kalman filtered IMU vectors, the Euler angle rotations (roll, pitch, yaw) is calculated. The filtered step response is presented in Figure 97. The result shows non-oscillating Kalman estimated states with the expected time-delay. The roll angle is due to the ZYX (roll,pitch,yaw) configuration of the Euler angles, equivalent to the toolface angle (before offset correction). It is important that this angle is well tuned to avoid high frequent setpoint shifting. This will be further analyzed when the toolface control result is presented in subsection 7.6.



(a) Roll angle. (b) Pitch angle. (c) Yaw angle.

Figure 97: Euler angle step response with Kalman filtered IMU data.

7.4 WOB Control Tuning

In this subsection, the results from WOB control testing and PID tuning will be presented. The tuning method is based on the Cohen-Coon tuning method and trial-and-error by following the PID theory. As explained in subsection 2.1, the Cohen-Coon tuning method calculates a theoretical ideal value for the PID gains by using the results from step response tests. That is, Cohen Coon tuning can mainly be used to find a starting point for the PID gains, where additional tuning is in most cases necessary. Based on the Cohen-Coon step response tests from the 2018 team, the 2019 team already have a starting point for the PID gains. Also, since the hoisting system is identical, the only difference is a new weight of the rotary stack due to new top drive and new BHA. That is, initial Cohen-Coon step response testing is not necessary and the team can jump straight into fine-tuning using PID theory and trial-and-error.

Due to limited time with an operational rig, the team lack of results from the starting point of the tuning process. The WOB controller have been tuned on the go in order to get the rig up and running and ready for shipping. Below, the results from the tuned WOB control response will be presented and analyzed for further fine-tuning. The trial-and-error tuning process has been based on the following principles.

Trial-and error tuning principles:

- Limit the PID control output based on the control purpose. For this controller, the PID output is HM rpm SP and have been limited to +-100 (rpm) in the beginning and later increased to +- 200 for increased ROP after the controller is tuned.
- Find a starting point for the proportional gain based on the expected error. In this project a proportional gain of $K_p = 1$ have been used. This will give a max HM velocity of 100 rpm and a hoisting speed of approximate 5,6 cm/min. The operator will have enough time to stop the operation in the control system or by using the emergency stop button to kill the rig power if it should be necessary.
- Only use filtered measurements as input to the PID controller to avoid high-frequent-high-amplitude shifting of the motor SP due to the derivative effect in the controller trying to dampen out the oscillations.
- Choose a starting point of the derivative effect and tune the controller by performing step responses. The team have tuned the the PD controller first before integral effect is added to simplify the tuning process.
- Add integral effect and tune to remove steady state error.
- If the controller is too slow/fast, increase/decrease the proportional gain and re-tune the derivative and integral effect. PID tuning is a iterative process.

In Figure 98, the WOB control step response is presented during a vertical drilling operation. The TD velocity is set to 100 rpm and the PM rpm is set to 1000 rpm. The WOB controller uses the Kalman filtered WOB signal as input to the PID controller. This filter tuning principle is equal as for the IMU measurements. Tag and safety logic runs on the raw WOB measurement to reduce the reaction time is an error occurs. The PID gains is set to $K_p = 1,6$, $T_d = 0,01$ and $T_i = 0,7$. That is, low proportional gain, very low derivative effect and low integral effect. The operation will be analyzed in time intervals:

- **40-60s:** The state machine is currently in the tag state with a WOB limit of 8 kg. Due to tagging in a pilot hole, the wall friction results in a measured WOB of 2 kg. The bit tags after 50 seconds and the tool is hoisted up. This results in a negative WOB of -10 kg as a result of wall friction and bias offset due to a dependency of the hoisting direction. The WOB is currently zeroed for positive hoisting direction. The WOB stabilizes on 2 Kg after a quick hoist down and the system jumps into the start-rotation-state.
- **60-90s:** The system is currently in standby mode and the vertical drilling state must be started manually. This is a part of the early version of the state machine where the operator manually confirms that all motors have started properly. At around 80 seconds, the vertical drilling state starts. The bit slips the wall and the WOB reaches zero in a short period. At 90 seconds, the bit tags the rock again and the drilling starts.
- **90-150s:** In this interval, two step responses is performed. Notice the change in process noise on raw WOB measurement compared to before drilling. The interval will be analyzed closer below.
- **150-250s:** The team performs manual step responses while drilling to test the stability of the WOB controller. The results shows a stable control response with varying steady sate error. The steady state error is especially large for WOB SP over 20 kg. This can be due to many reasons, but each setpoint is not held long enough in this test for the integral effect to correct. The integral effect is also set relative low in the current tuning.

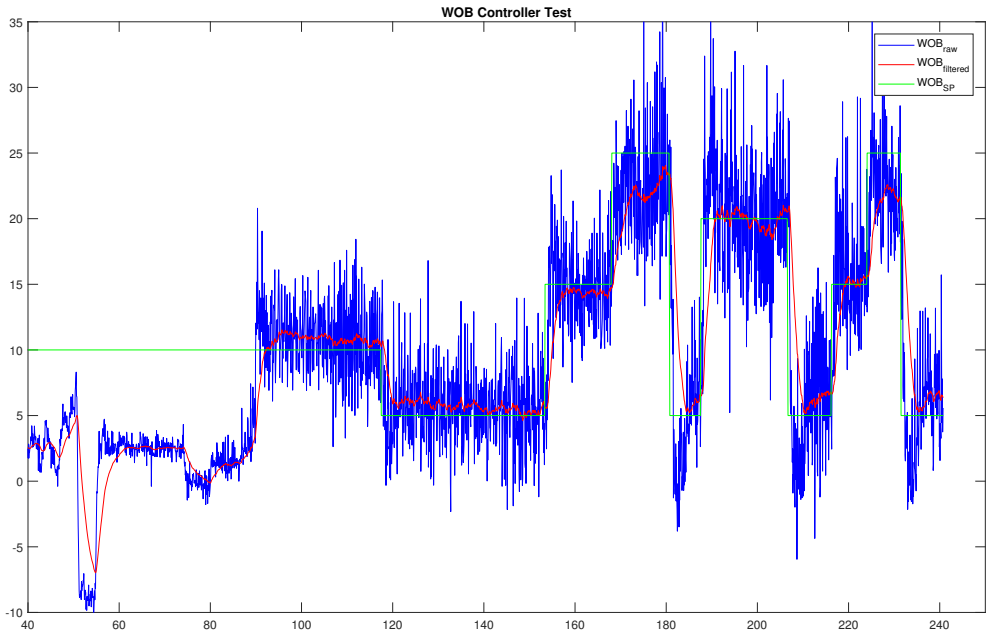
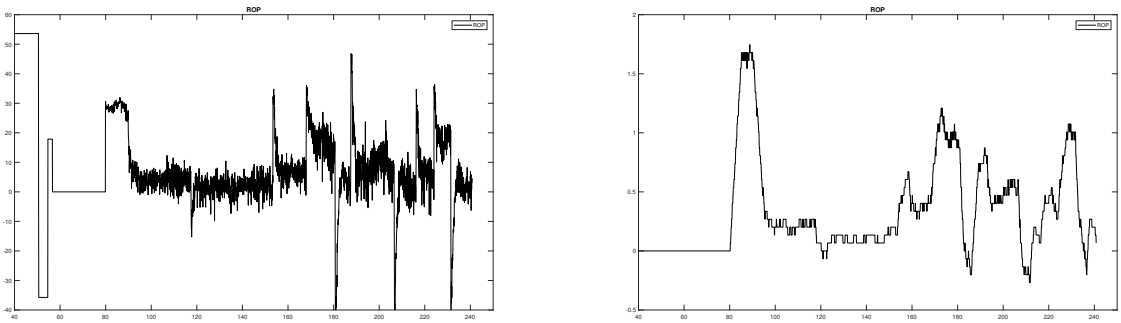


Figure 98: WOB tuned step response while drilling.

One of the main considerations for tuning of the WOB controller, is to reduce wear on the hoisting motor. In Figure 99, the HM rpm input and the ROP for the same time interval is presented. In Figure 99a, it can easily be seen when the WOB control mode is activated. The plot shows high frequently shifting PID output (equivalent to the HM input) with an amplitude of 5 rpm when in steady state. In the design report, the team analyzed the amplitude of the 2018 PID output to be around 10 rpm. The reason for this is that the 2018 team used a third-order butterworth low-pass filter on the WOB measurement. The low-pass filter was tuned to filter out high spikes on the WOB measurement resulting in a filtered steady state amplitude of aprox 3 kg. By feeding the low-pass filtered measurement into the PID controller, the derivative effect will try to dampen out the 3-kg-amplitude oscillations. The result is an amplified, equally oscillating PID output based on the tuning of the derivative effect. By using a Kalman filtered WOB measurement as PID input, the team have decreased the amplitude of the PID output, which will reduce the wear on the hoisting motor.



(a) HM rpm input

(b) Rate of penetration (ROP).

Figure 99: Hoisting motor input and rate of penetration.

By taking a closer look at the 90-150s interval, the current PID tuning can be analysed. In Figure 100, a zoomed-in plot is presented. The WOB controller perform a stable step response with a steady state error of approximate 1 kg. This is due to two reasons; (1) the accumulated integral of the error before the bit re-tags the rock and (2) low tuned integral effect. After the vertical drilling state is activated, it passes approximate 15 seconds before the bit tags the rock and starts drilling. During these 15 seconds, the integral of the error accumulates positive PID output. When the bit tags and WOB goes to SP, the accumulated error-integral is positive and shifts the WOB response in the positive direction. It will take some time before the integral effect is zeroed out due to the low error after the step response of aprox 1 kg, compared to the integral of a 10-kg-error over 15 seconds.

The team have fixed this problem by implementing a reset-integral-effect function when the bit re-tags the rock when vertical drilling starts and after a setpoint change. The integral effect is also tuned faster and is currently set to $T_i = 0,4$.

Another interesting observation is the overshoot of the raw WOB measurement compared to the Kalman filtered measurement. This is due to the time-delay between the Kalman filtered measurement and the raw measurement. By also considering the steady state error, it can be seen that the raw WOB overshoots with aprox 4 kg after a step response (compared to the filtered). This is one of the main risks by using slow tuned Kalman filtered measurement as control input. That is why all safety logic runs on the true measurement to obtain the fastest response time if an error occurs. However, by tuning the PID and Kalman filter together the process can be tuned to cope with the time-delay. This is explained below.

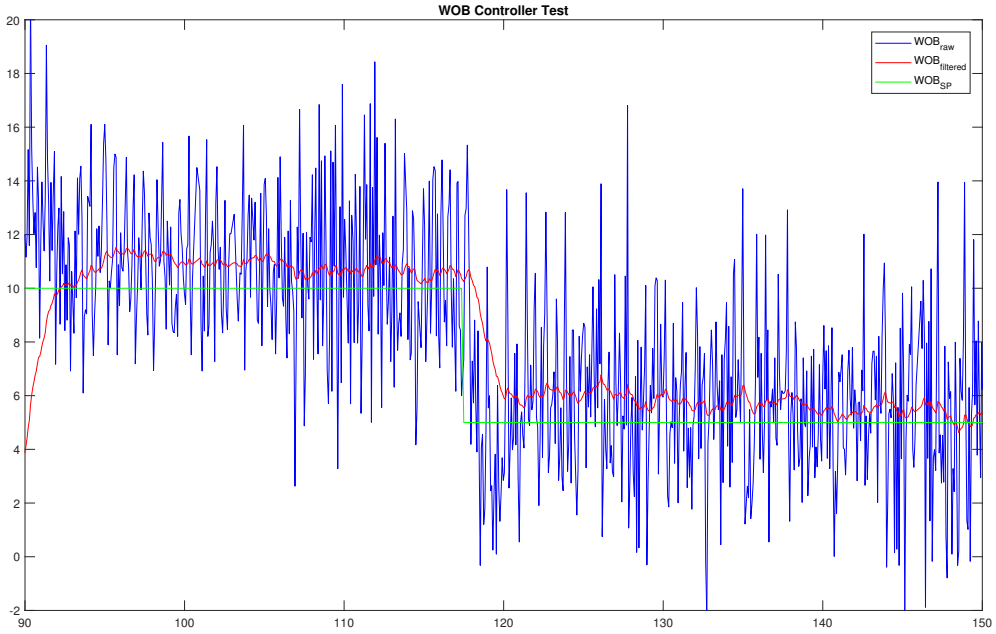


Figure 100: WOB tuned step response analysis.

In Figure 101, the PID is tuned to output an over-dampened WOB response. Since the true WOB measurement is unknown for the controller, an over-dampened response can be used to cope with the time delay and decrease the error between the true and filtered measurement during step responses. The current example is exported directly from Labview, where the sampling rate of the graph is reduced to increase the while-loop iteration rate. Consequently, the raw WOB measurements look lees noisy then they are. However, the plot clearly shows that by tuning for an over-dampened response, the raw WOB overshoot is reduced.

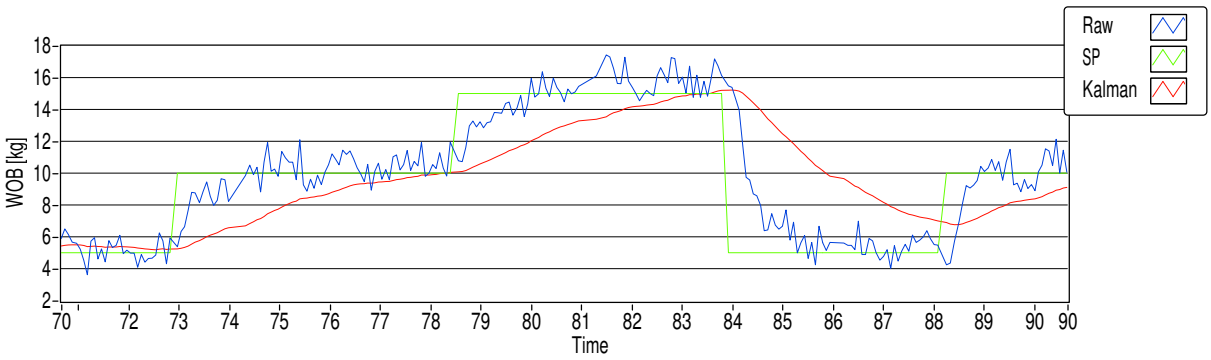


Figure 101: Over-dampened PID response to cope with the time-delay between the raw- and Kalman filtered measurement.

7.5 Position Tracking Testing

In this subsection, results from the position tracking tests with magnetic ranging will be presented. All tests are performed outside the rig due to the technical issues as described in the intro. The position tracking algorithm used for this experiment is presented in Equation 39 in subsection 5.3.

The objective with position tracking test is to test the accuracy of the position tracking algorithm by hoisting the sensor sub through a curved plastic tube to mimic a directionally drilled well. The linear hoisting velocity is set to 5 cm/min and the plastic well is manually measured to be 60 cm deep and 20 cm long. The position tracking test setup is presented in Figure 102. The test will be performed with three different configurations of the setup; (1) uncorrected, (2) passive magnetic ranging (PMR) and (3) active magnetic ranging (AMR). The objective with the uncorrected setup is to test how the local magnetic distortion from the rig frame influences the position tracking. Initial measurements of the magnetic field inside the rock sample chamber indicates that the magnetic flux lines slightly bends toward east as you move forward in the chamber. This is illustrated in Figure 103 (similar to Figure 32 in section 3). The objective with the PMR- and AMR setup is to analyze how magnetic ranging can be used to help position tracking in a highly magnetic distorted local environment.

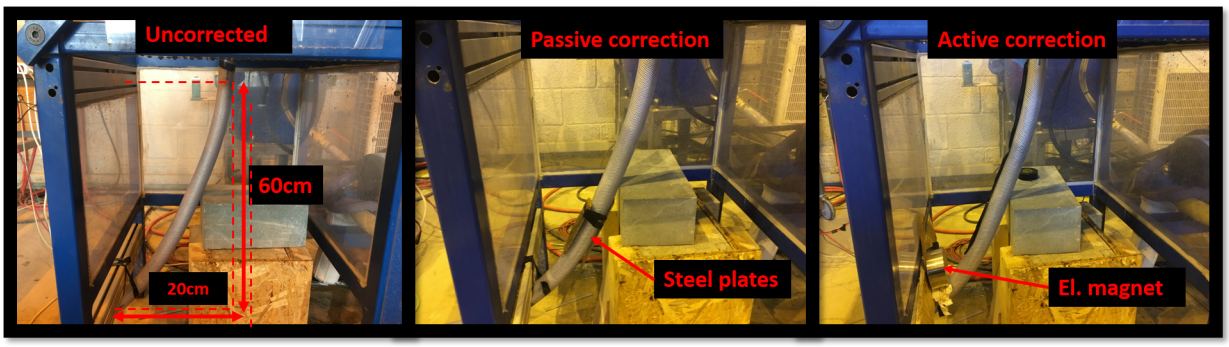


Figure 102: Position tracking test setup with uncorrected, passive- and active magnetic correction configurations. A plastic tube is used to simulate a deviated well.

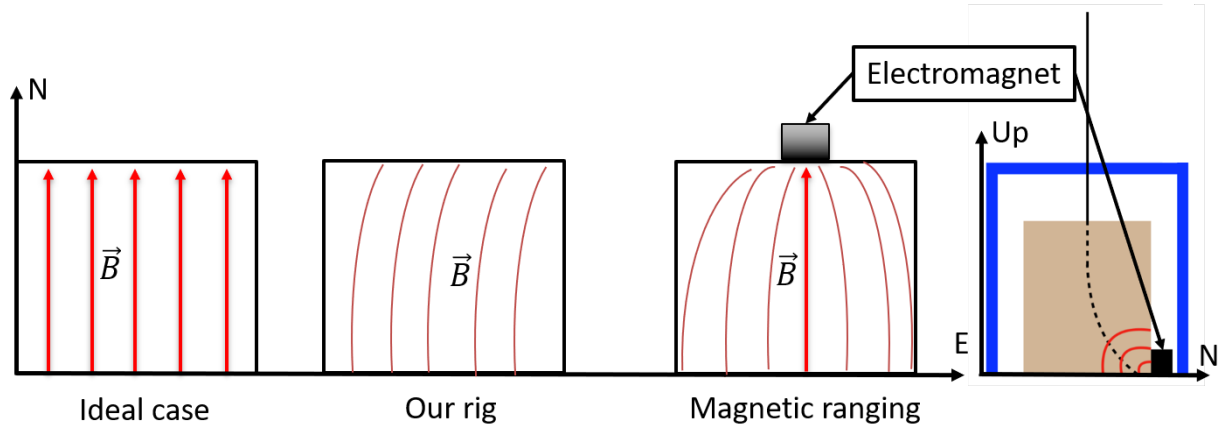
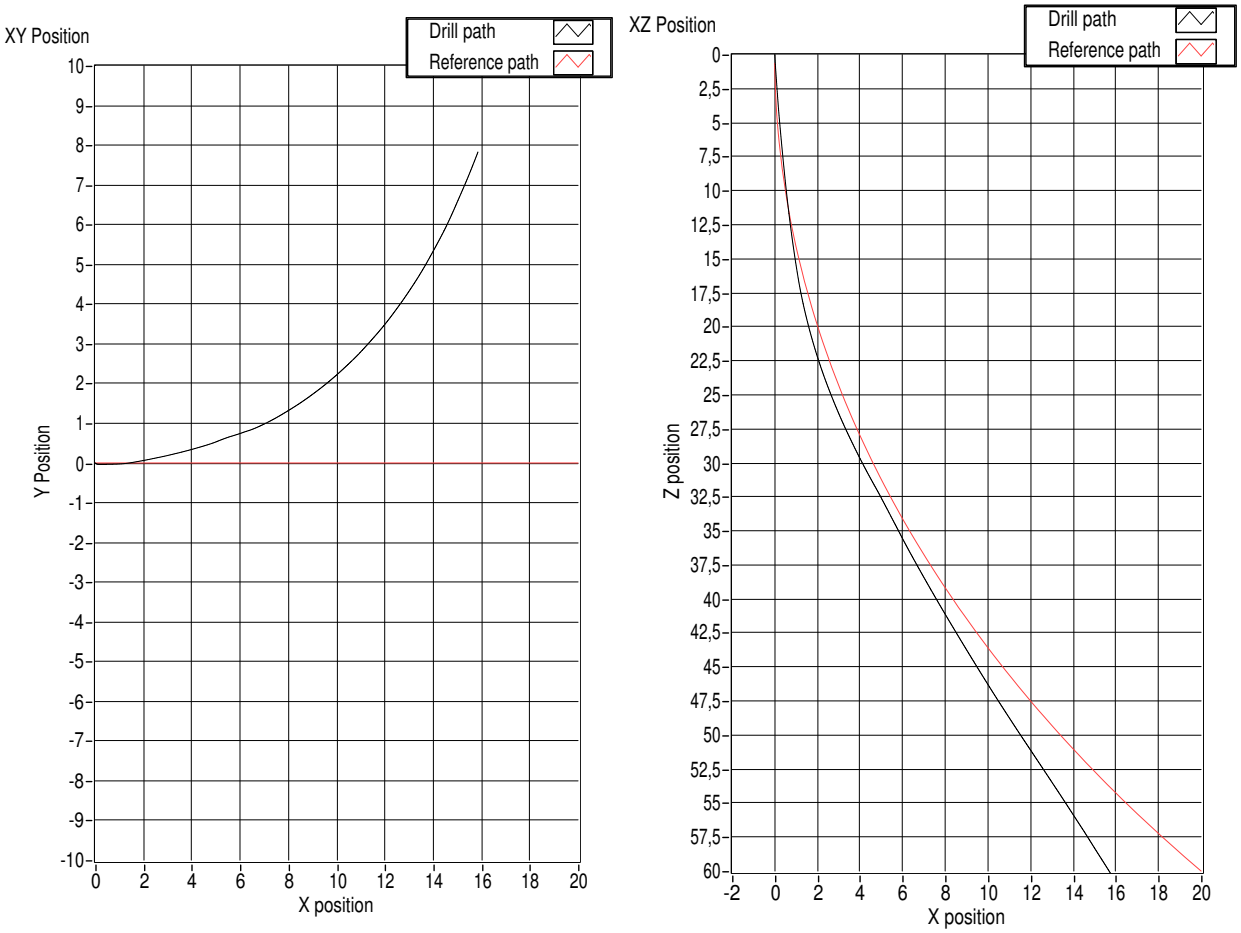


Figure 103: Active magnetic ranging setup using an electromagnet aligned with the magnetic north flux lines, and illustration of the magnetic flux lines for the ideal-, true- and magnetic ranging case.

Uncorrected:
The results for the uncorrected position tracking test is presented in Figure 104.



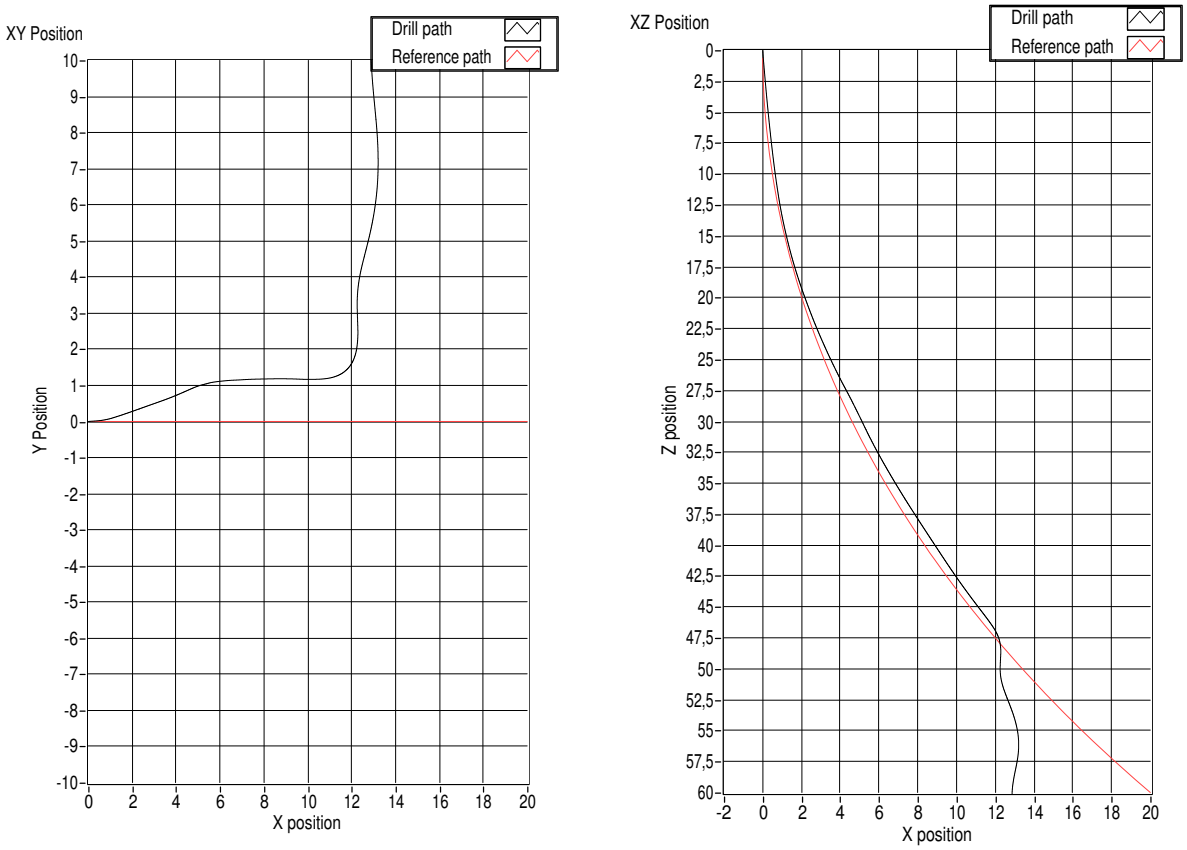
(a) XY-plot

(b) XZ-plot

Figure 104: Position tracking test with uncorrected passive rig distortion.

As expected from the initial measurement of the magnetic distortion inside the rock sample chamber, the result shows that the position tracking deviates from the centerline. The magnetic flux lines curves more and more towards east as the sub move forward in the chamber. When the magnetic north direction is changing towards east, the IMU will measure this as a movement towards west as shown in Figure 104a. Even if the curved plastic tube well deviates a lot more towards the steel frame than the planned drill path will, the test shows that the local magnetic distortion in the rock sample chamber effects the accuracy of the position tracking algorithm. Also, the azimuth angle rate is approximate constant during the whole operation, which confirms the curved flux lines theory.

Passive magnetic ranging:
The results for the PMR corrected position tracking test is presented in Figure 105. In this test, an elongated steel plate is attached to the top of the plastic tube to mimic a target well. The idea is that the IMU will lock on to this passive distortion source and treat is as a local north. The results in Figure 105a shows that the position tracking starts to deviate from the start, equal to the uncorrected case. However, when the sub reaches the 5 cm mark of the horizontal displacement, it stops to deviate and tracks the position straight forward until it reaches the 11 cm mark. After this point, the IMU tracks the sub to go in the west direction. The reason for this is that when the sub reaches 5 cm horizontal displacement, it enters into the magnetic hot-zone of the ferromagnetic material. The IMU locks on to the steel plate and treats it as a local north, and since the steel plate is place along with the direction the IMU is hoisted and the polarity is in front of the IMU, it will track the position to go straight forward. After 11 cm, the IMU goes past the north polarity and is on its way to exit the magnetic hot-zone of the steel bar. At the same time the sub gets closer to the rig frame where the original flux lines already bends towards east. The combination of both events results in a magnetic heading straight west.



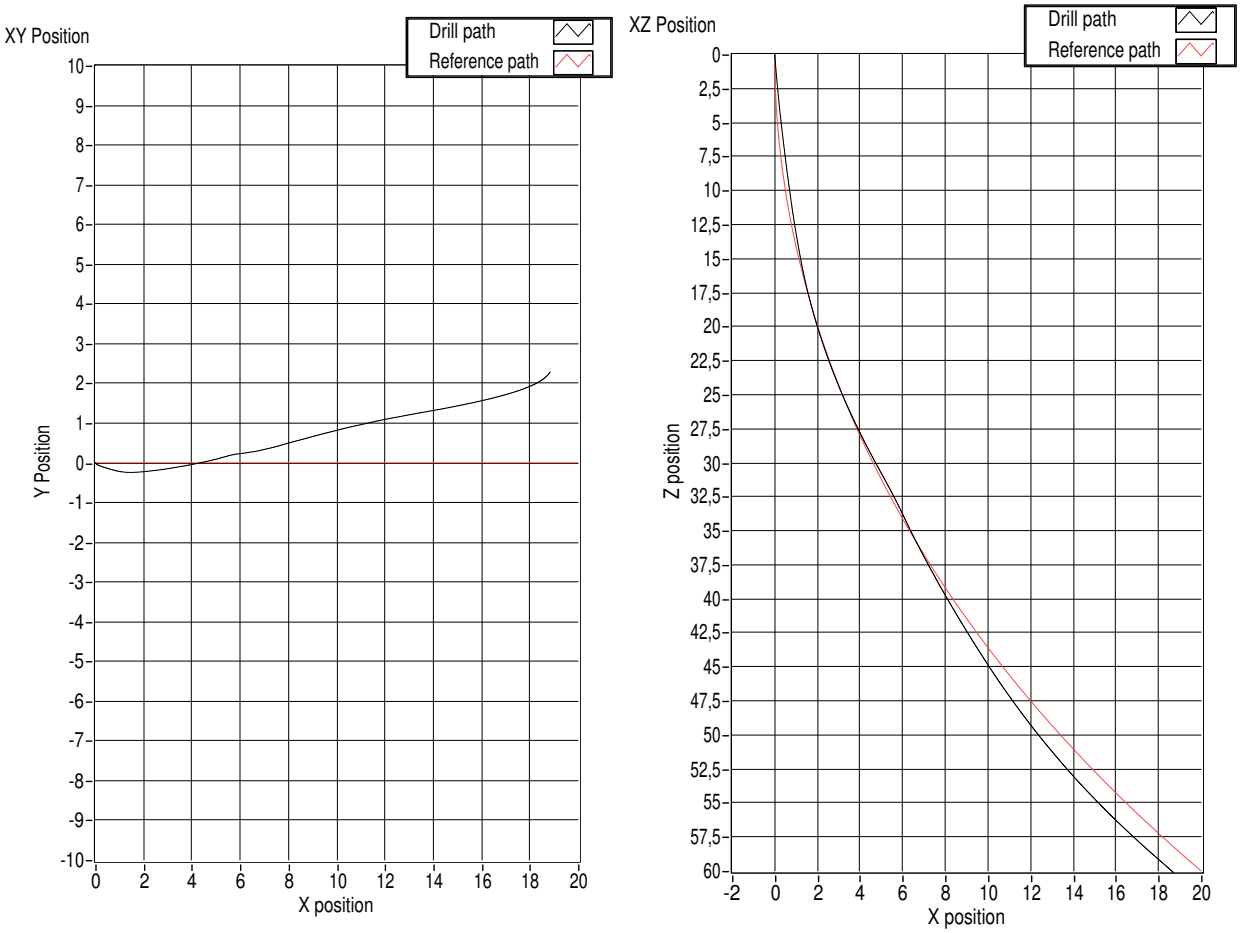
(a) XY-plot

(b) XZ-plot

Figure 105: Position tracking test with with passive ranging correction using two steel plates.

The take-away from this test is that a ferromagnetic material such as iron can be used as a guide (or target well) to lead the IMU trough a local environment with high magnetic distortion. The 5-11 cm part of the horizontal displacement for this test demonstrates that.

Active magnetic ranging:
The results for the AMR corrected position tracking test is presented in Figure 106. In this test, an electromagnet is placed in the front-bottom part of the well according to the test setup in Figure 102. The idea is that electromagnet will act a local north with sink properties that will distort the original magnetic field in a local environment, as visualized in Figure 103. As explained in subsection 2.2, magnetic field strength reduces cubicly over the distance from the magnetic source. That is, the electromagnet will work stronger as the closer the IMU gets closer.



(a) XY-plot

(b) XZ-plot

Figure 106: Position tracking test with active ranging correction using an electromagnet.

The result shows that the position tracking is highly improved by using an electromagnet in front of the well. From 2-18 cm horizontal displacement in x-direction, the magnetic heading is pretty much constant. This indicates that the flux lines inside the rock sample chamber is changed from the previous curved lines towards east. The IMU still deviates 2 cm towards west, but this is most likely due to misalignment between the electromagnet and the plastic tube. After 18 cm displacement, it can be seen that the heading starts to turn towards west. This is because the IMU is hoisted underneath the electromagnet, and hence past the polarity point it have been navigating relative to. In Figure 106b, it can be seen that the tracked xz-position is pretty accurate compared to the reference path. The slight inaccuracy is most likely due to the 2 cm deviation in the west direction and the fact that the curvature of tube-well is not shaped with constant

radius. To summarize, the result from the AMR position tracking shows that a electromagnet can be used change the local flux lines and hence improve the position tracking of an IMU device in a magnetic distorted local environment.

7.6 Toolface Control Testing

In this subsection, results from toolface control testing will be presented. The toolface controller is implemented on a test rig that was build after the rig was shipped for the competition. The test rig and test setup can be found in Figure 107. The simulation script is used for testing, where all motor- and sensor outputs besides top drive and IMU is replaced with manual adjustable variables. The toolface control testing is divided into three tests:

1. Orient from random position (the orient state in state machine)
2. Orient for passive magnetic ranging (PMR)
3. Orient for active magnetic ranging (AMR)

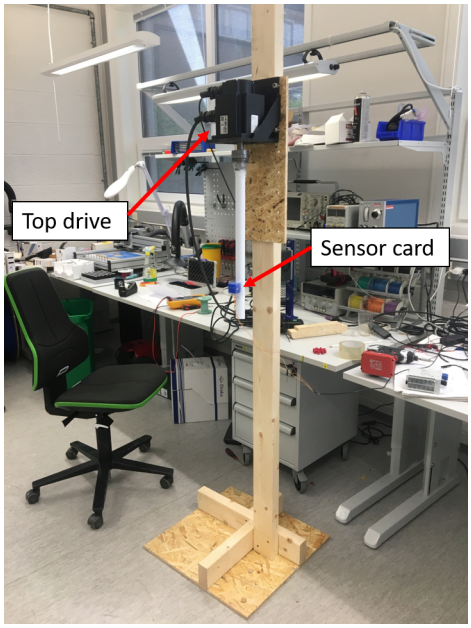


Figure 107: Setup for toolface control implementation and testing

Orient from random position:
In this test, the state machine is run trough the first 5 states in the state machine before it enters into the orient state. That is, the top drive have been rotated during the vertical drilling state and the IMU will have a random orientation relative to the magnetic north. In Figure 108, the result from the orientation from a random start position is presented. The result shows that after 96 seconds the orient state is activated, and the TD absolute position is homed (set to zero). The toolface SP is currently set to zero degrees from north and the toolface angle is currently around 80 degrees from north. The toolface PI controller takes in the toolface error and outputs a new SP for the TD position. The result shows that the PI controller steers the toolface towards the SP with an over-dampened response. Since the PI output is a position SP, overshooting is not an issue. That is the main reasons why the derivative effect is not needed in the controller. The internal position controller in the TD drive takes care of the position control.

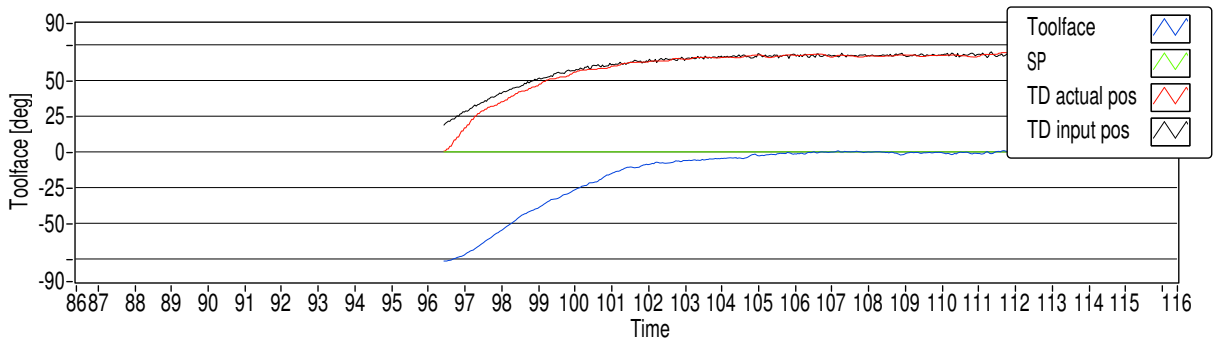


Figure 108: Orient toolface from random start position

Orient for passive magnetic ranging:
In this test, a steel rod is used to create a local distortion source that the IMU will lock on to treat as a local north. The result is presented in Figure 109. First, the toolface controller orients and settles at the SP which currently is sat to zero degrees from the magnetic north. At time 225 seconds, the steel rod is held close to the sensor with a 90 degree angle angle relative to the current toolface orientation. This is shown in Figure 110. The results show that the toolface controller orients towards the new local north and settles at the SP.

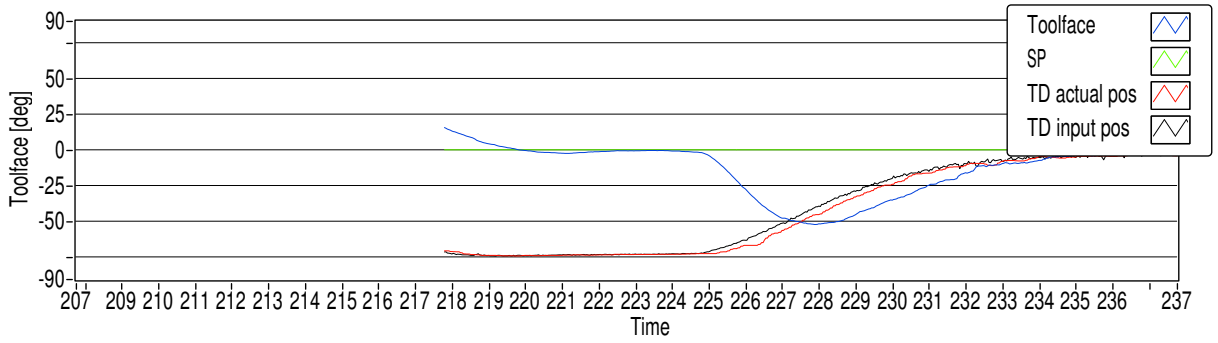


Figure 109: Toolface steering with passive ranging.

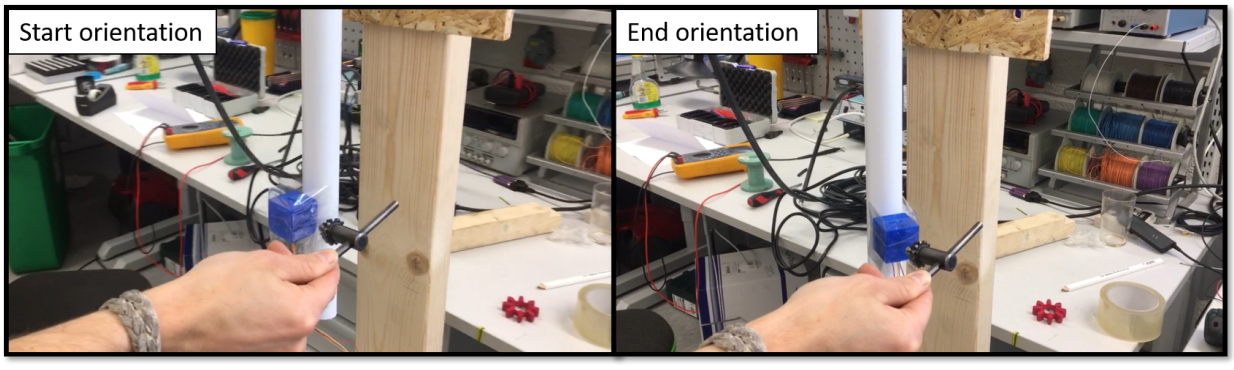


Figure 110: A steel rod is used to manipulate the IMU to make the toolface controller to steer towards it.

Orient for active magnetic ranging:
In this test, an electromagnet is used to create a local distortion source that the IMU will lock on to treat

as a local north. The result is presented in Figure 111. As for the passive test, the toolface controller have settled at the SP of zero degrees for north. The electromagnet is placed with a 90 degree orientation relative to the current toolface orientation. This is presented in Figure 112. At time 515 seconds, the electromagnet is turned on which results in a toolface error due to the new local north. The result shows that the toolface controller steer towards the new local north and settles at the SP.

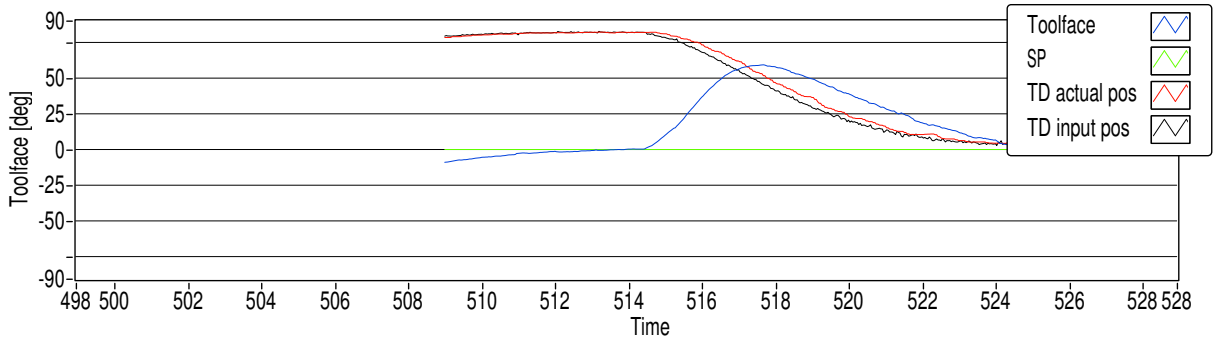


Figure 111: Toolface steering with active ranging.

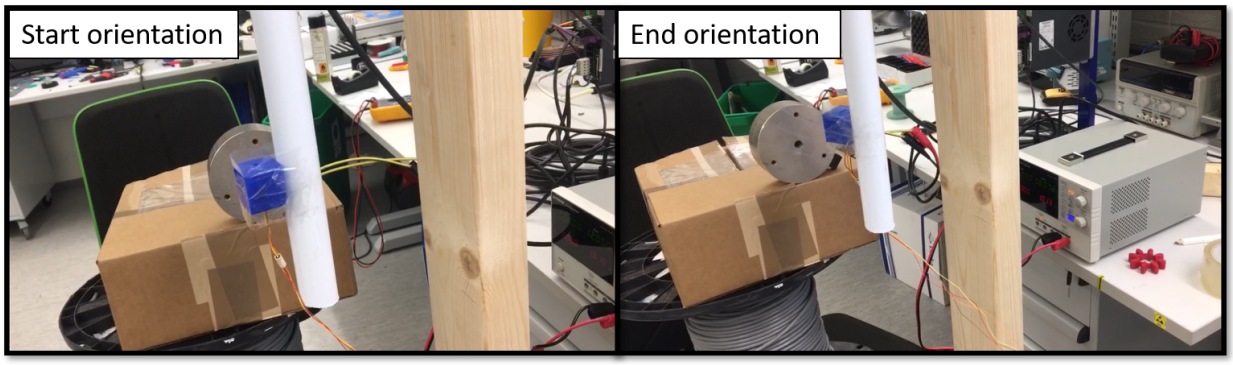


Figure 112: An electromagnet is used to create a new local north and make the toolface controller to steer towards it.

The toolface control tests shows that the toolface controller is well tuned and capable of steering the BHA toolface towards the magnetic north direction. That is, the toolface controller can be used to steering in a magnetic ranging setup.

7.7 State Machine Demo

On the Drillbotics competition day in Celle, Germany, the team presented a demo of the state machine by using the simulation script and an improvised demo rig with the rotary stack. The demo rig includes all equipment that was held back for further testing and implementation when the rig was shipped, The demo rig is presented in Figure 113. The front panel and block diagram for the simulation script can be found in Appendix C.

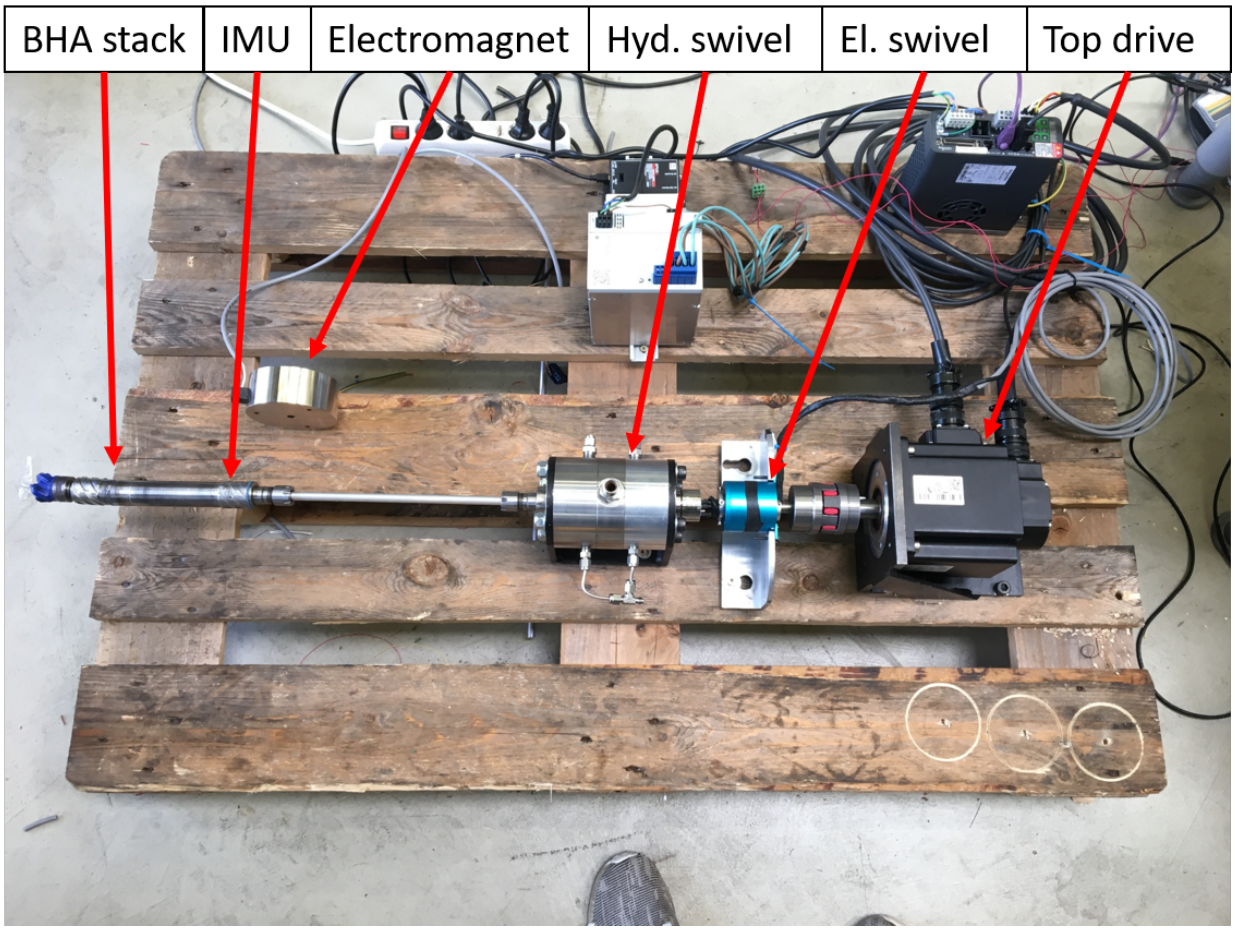


Figure 113: Demo rig on the competition day.

7.8 Overall system analysis

In this subsection, the an overall system analysis will be summarized. The analysis will focus on the challenges with the current system design and the uncertainties with the untested parts of the system. First, the steerability of the directional tool will be analysed.

Controlability and Observability of Directional tool:
A part of the scope for this thesis is to collaborate in the overall system design to ensure controllability and observability of the parts that will be used in the closed loop control of the drilling process. The directional tool, or the BHA with the bend sub, is a vital part of the position control and steerability of the drilling process. By using the IMU the hoisting motor velocity, the control system will have the opportunity to observe the orientation of the tool in real time and track the position by using the position tracking algorithm. Based on the result from the IMU calibration, Kalman filtration and position tracking tests the position of the drillbit will in theory be observable. However, there is still an uncertainty present regarding how the IMU will cope with the process noise from the drilling process. Testing of IMU inside the BHA during drilling with all motors running is a necessity before observability can be concluded on.

The controllability of the is mainly dependent on the mechanical and actuation design. With the current design the team have the ability to build inclination by using the physical principle behind the behind sub in a slide while drilling operation. By applying WOB the tool will build some angle based several dependencies that will not be analysed in this report. The team have also the ability to steer the toolface and thereby control in what direction the tool builds angle. The only thing the team can not control is the angle build

rate. The angle build rate can be manipulated by playing with the WOB SP, but not controlled with a angle of freedom. That is, the position control of the directional tool is partly controllable. However, if the competition problem statement is considered, the need to adjust the inclination is not necessary as long as the tool is designed to build enough angle to reach the max score zone.

The team have only been able to perform limit testing on the inclination build rate in a soft cement sample. This rock sample was too soft to drill in resulting in high rop with almost non WOB due to water jets and bit rotation flushed away the formation before the bit could settle on a WOB SP. The team reached ROP on up to 10 cm/min with minimal WOB. Without WOB, effect of the bend sub will not give the desired effect and result in a almost vertical drilled well. The team made a harder rock sample the weak before shipping but technical issues with vital motors and components resulted in that tests was postponed to the last day. In the middle of the process of rigging up for a performance test of the bend sub inclination build rate capabilities in a harder rock sample, the rig computer broke down. That is, the team team spent the whole night trying to retrieve all files for the control system before the rig was shipped to Celle. Due to this, the build rate capabilities of the bend sub also remains untested. Testing is generally needed to determine the the observability and controllability of the directional tool and can not be concluded on.

Weak links in the system design: There are generally two main weak link in the system based on the results from the implementation and testing of the rig from a control perspective. These are the downhole data signal transmission and downhole power section. The bend sub inclination build rate is not yet proven to be good or bad, and hence neglected from this analysis.

The issues with downhole data signal transmission is mainly centered around the wired setup from the IMU, up through the electrical swivel. The electrical swivel was tested at a early stage to make sure it worked as planned to. Unfortunately, technical issues with the electrical swivel appeared when the rig with all subsystems was assembled to be tested together at the end of the project. The connector out from the hydraulic swivel used fragile copper wires glued into small holes to transport signals from inside the pressurized zone and out. These wires turned out to be too fragile, resulting in wires snapping at the connection point. Another issue was with short circuits do to thin isolation on the copper wires. Both issues where solved after the rig was shipped where a new connector was made to remove this weak link from the system. On the competition day, the team could run the whole rotary stack including sensor card and electrical swivel without trouble. That is, if the rig had arrived for the competition it would have been operational seen from this side.

The other main weak link is the downhole power section where the torque generally too low. The downhole power section is not a part of the scope for this thesis, but the whole control system and setpoint selector algorithm is designed based this being the weak link in the system. The latest version of the control system was programmed and designed for the EMM too be used as the DH power section for the competition run. As explained in section 6, many features was implemented such as the safety sequence and to SP selector to keep the control system up and running even if the EMM stalled out. The team were unfortunately not able to test these features either do the shipping issues, and will also remain untested.

To summarize; the overall system lacks or have minimal time with collective testing with all subsystems presented. This makes it difficult to conclude around the overall system design and performance capabilities. Further testing is needed with all subsystems together, and with the new features of the control system . The downhole power section is generally took weak, even if the control system can keep the operation going. Next years team will have to makes this among the main focus areas of improvement.

8 Conclusion

This thesis has described the work related to the design and implementation of instrumentation setup and control system for the NTNU Drillbotics' contribution to the 2019 Drillbotics competition. Drillbotics has proven to be a challenging and educational project, where theoretical, practical and social skills are put to the test. The report will highlight experience with interdisciplinary teamwork, day-to-day problem solving and multidisciplinary work tasks within cybernetic-, electrical- and mechanical engineering as among the most awarding. According to the scope for this thesis, the following work have been carried out:

- **Downhole sensor card:** A downhole sensor card containing an inertial measurement unit (IMU) have been successfully designed, built and implemented in the miniature drilling rig. Results from practical experiments show that the IMU can measure the orientation of the BHA toolface, and track the position of the drillbit in 6 DOF. However, the IMU have minimal testing with live drilling due to technical problems with the electrical swivel on the rig. Further testing is therefore needed to confirm accuracy and robustness during drilling.
- **Magnetic ranging setup:** An active magnetic ranging setup have been implemented and tested. Results show that active magnetic ranging can be used to improve the accuracy of position tracking and position control in a miniature drilling rig with high internal magnetic distortion.
- **Topside actuation- and instrumentation setup:** A topside actuation- and instrumentation setup have been implemented and successfully tested, including a new top drive, a solenoid valve and a tank level sensor, in addition to the downhole sensor card. Actuation and instrumentation from the previous rig that has been reused for this year includes the hoisting motor, pump motor, load cell and pressure gauge. Steffen Wærnes Moen and Noralf Vedvik have helped with the electrical- and mechanical setup respectively.
- **Control design:** A PID WOB controller and PI position controller have been designed and implemented in the control system to control the WOB and drillbit position respectively. The result shows a stable WOB controller that only needs fine-tuning. The position controller is not tested due to technical issue with the electrical swivel. The toolface controller, which is the internal PI controller for the position controller, have been successfully tested by performing practically experiment in a test rig setup. Further testing is however necessary to confirm accuracy and stability during drilling.
- **Control system:** A control system has been designed and implemented in Labview to autonomously control the miniature drilling rig to solve the Drillbotics Competition 2019 problem statement; e.g. WOB-, toolface- and position control, with boolean safety logic and setpoint optimization of the weakest actuator in a 9-state state machine. Steffen Wærnes Moen have helped with Modbus and CANopen communication protocol setup in Labview for the motor drives. The control system is tested to be fully functional, but have minimal operational testing time with live drilling. Further testing is needed to confirm overall system stability, accuracy and safety.
- **Simulation script:** A simulation script have been implemented to simulate the boolean logic in the state machine. The simulation script is an un-modeled copy of the control system, where all motor and sensor inputs/outputs are replaced by manually controlled variables. The script have been used to successfully implement and test the boolean logic in the state machine, with two-layered safety logic, optimized setpoint selector and live operation log. The simulation script is a first step towards a real drilling simulator or a digital twin of the system, where modeling of the drilling process is necessary. This has not been a part of the scope for this thesis.
- **Steerability of directional tool:** A part of the scope for this thesis is to ensure controllability and observability of the directional tool from a control perspective point of view. The observability is ensured with the downhole IMU which can measure orientation and track position in 6 DOF. The controllability is only partly ensured with the current mechanical design. The bend-sub-design with top drive position- and velocity control of pipe and downhole power section velocity control of bit, gives the system the ability to build inclination and azimuth, control in what direction the tool should build inclination and azimuth and drill straight down/forward. The inclination build rate can not be

controlled, only manipulated by adjusting the WOB. To conclude, the directional tool is observable and partly controllable, but still capable of solving the challenge presented in the Drillbotics problem statement.

The 2018/19 problem statement by DSATS has proven to be a challenging task for all teams that entered the Drillbotics competition. The increased complexity with limitations for outer dimensions, time and budget have forced all teams to think out of the box. This clearly emerged during the European final in Celle, Germany, where all teams that entered the A-final presented their own solutions on how the task should be solved with fundamental differences in terms of steering, actuation and control algorithm. None of the teams presented a perfect solution without a vital trade-off. This shows how challenging the problem statement is, and why the Drillbotics competition can be a drive for innovation among students and within the oil- and gas industry.

A lot of work have been put into the overall system design and implementation of the 2018/19 NTNU rig by all team members and support team. However, the project will be handed over to next years team who will continue developing the system according the next years problem statement. Based on conversations with competition judges, it is expected that the problem statement for next years competition will be similar to this year with minor changes. Expected changes are multilayered rock samples, third party plug-and-play integration and changes in outer dimensions.

9 Further Work

The future work is based on a overall system analysis of the current design with focus on instrumentation, control system and the expected changes in next years competition problem statement. Future work is summarized in the list below:

- **Competition run:** Due to the problems with shipping, the rig remains untested for some subsystems including the final version of the control system. A first step to is test all un-tested parts of the system, such as position tracking and position control with the downhole sensor card during drilling. Secondly, a proper competition run with the 2019 rig setup is highly advised. The result and analysis of this test can be used to improve or redesign the rig setup for the 2020 problem statement.
- **Process modeling for Kalman filter:** The WOB and IMU can be modeled to improve the Kalman filtration. This is important if the competition guidelines reintroduces layers in the rock sample, where tuning of the process controls is important.
- **Downhole data transmission:** One of two main challenges this yeas was downhole power and data transmission. Limited space in the drillpipe and problems with short circuit in the electrical swivel makes this an area with big potential of improvement. An upgrading or redesign of the downhole data transmission is recommended for future projects.
- **Downhole power section:** Even if the downhole mechanical design is not a part of the scope for this report, the control system have been designed with special consideration on the weakest actuator in the system. From a control perspective, the downhole power section must be a focus area for future projects.
- **Downhole actuator for steering:** In the report, the directional tool is analysed to be partly controllable. Future teams is advised to further investigate options for downhole actuating of steering. This will improve the controllability of the process as steering with a topside actuator is very limited. If the competition guidelines increase outer dimensions of the BHA, there will be more space for downhole actuation.
- **Digital twin:** The simulation script has proven to be an important part of the early testing capabilities of the control system. A further development is this system is recommended by introducing process modeling to the system. If the drilling operation (for a miniature drilling rig) is modeled, a digital twin of the rig can be made. This can open up for more advanced optimization of the drilling process, where algorithms can be developed and tested inside the simulator before applied on the miniature drilling rig. The modeling of the system is however a complicated process, and can advertised as a standalone master thesis in parallel with the Drillbotics project.
- **Third party plug-and-play:** This will most likely be a requirement for next years competition and future project teams is advised to start on this early. A third party plug-and-play solution can potentially lead to a complete redesign of the control system structure. This is dependent the details provided by the competition guidelines for Drillbotics 2020.

References

[1] E24 Victoria Armstrong. *Nå er oljeprisen på sitt årsbeste for 2016*. May 2019. URL: https://e24.no/energi/oljebremsen/naa-er-oljeprisen-paa-sitt-aarsbeste-for-2016/23632839.

[2] Alexey Pavlov Bjarne Foss and more. *NTNU Strategy for Oil and Gas*. May 2019. URL: https://www.ntnu.edu/documents/1281387914/1281513667/BRU21+2017+NTNU+%28Print%29.pdf/4fc78ce5-2987-4f17-8695-67aec203f266.

[3] Equinor. *Automated drilling? There's an app for that*. Apr. 2018. URL: https://www.equinor.com/en/magazine/in-control-with-automatic-drilling.html.

[4] DSATS. *ABOUT DSATS*. May 2019. URL: https://connect.spe.org/dsats/home.

[5] Drillbotics. *ABOUT Drillbotics*. May 2019. URL: https://drillbotics.com/about-drillbotics/.

[6] Drillbotics. *2018 Drillbotics Winners Announced*. May 2019. URL: http://drillwell.no/home/great-results-at-drillbotics-2017.

[7] Drillbotics. *ABOUT Drillbotics*. May 2019. URL: https://drillbotics.com/2018-drillbotics-winners-announced/.

[8] Turøy Handeland and more. *Design and Optimization of a Miniature Autonomous Drilling Rig*. Jan. 2019. URL: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2559474.

[9] Mads Bertheussen Naamdal. "Design Report NTNU - Drillbotics 2018/19". NTNU, 2018.

[10] Drillbotics. *Finalist for 2018-2019 Announced*. Jan. 2019. URL: https://drillbotics.com/finalist-for-2018-2019-announced/.

[11] Mikkel Leite Arnoe. "The Design and Implementation of a Control System for an Autonomous Miniature Drilling Rig". MA thesis. NTNU, 2018.

[12] Drillbotics. *Blog*. June 2019. URL: https://drillbotics.com/blog/.

[13] 'National Instruments'. *'PID Theory Explained'*. Nov. 2018. URL: 'http://www.ni.com/white-paper/3782/en/'.

[14] Bjarne Foss' 'Jens G. Balchen Trond Andersen. *'Reguleringsteknikk'*. 'NTNU', 2016.

[15] Melda Ulusoy. *Understanding Kalman Filters, Part 3: An Optimal State Estimator*. Oct. 2017. URL: https://www.mathworks.com/videos/understanding-kalman-filters-part-3-optimal-state-estimator--1490710645421.html.

[16] Raghda Bakerey Khader Elssadig and Dr. Eltahir Mohammed Hussien. *Function using FOPDT, SOPDT and SKOGESTAD in Control System*. Dec. 2016. URL: http://sustech.edu/staff_publications/2017012606164628.pdf.

[17] 'Jacques Smuts'. *'Cohen-Coon Tuning Rules'*. Mar. 2011. URL: 'http://blog.opticontrols.com/archives/383'.

[18] Olav Egeland and Tommy Gravdahl. *Modeling and Simulation for Automatic Control*. NTNU, 2003.

[19] Matej Andrejašic. *MEMS ACCELEROMETERS*. May 2019. URL: http://mafija.fmf.uni-lj.si/seminar/files/2007_2008/MEMS_accelerometers-koncna.pdf.

[20] Electrics Tutorials. *Hall Effect Sensor*. May 2019. URL: https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html.

[21] Steven Earle. *9.3 Earth's Magnetic Field*. May 2019. URL: https://www.crystalinks.com/earthsmagneticfield.html.

[22] Steven Earle. *9.3 Earth's Magnetic Field*. May 2019. URL: https://opentextbc.ca/geology/chapter/9-3-earths-magnetic-field/.

[23] Christopher Konvalin. *Compensating for Tilt, Hard-Iron, and Soft-Iron Effects*. May 2019. URL: https://www.sensorsmag.com/components/compensating-for-tilt-hard-iron-and-soft-iron-effects.

[24] Millersville University. *EXPERIMENT OF THE MONTH*. May 2019. URL: https://www.millersville.edu/physics/experiments/023/index.php.

[25] Clinton Moss. *A Comparison of Active and Passive Magnetic Ranging Techniques in a Relief Well Application*. May 2019. URL: http://www.iscwsa.net/download/19d2462b-c1e3-11e6-9933-b9e85bce7ecb/.

[26] Schneider Electric. *BCH2MM1523CA6C*. May 2019. URL: https://docs-emea.rs-online.com/webdocs/14d7/0900766b814d71da.pdf.

[27] Schneider Electric. *LXM28AU15M3X*. May 2019. URL: https://www.schneider-electric.com/en/product/LXM28AU15M3X/motion-servo-drive---lexium-28---single-and-three-phase-200...230-v---1.5-kw/.
[28] Senring. *G012-12 Series Super High Speed Slip Ring(Max Speed 5000RPM)*. May 2019. URL: https://www.senring.com/high-speed-slip-ring/g012-12.html.
[29] 3D Metprint AB. *EMPOWERING GROWTH WITH 3D PRINTING OF METALS*. May 2019. URL: https://www.3dmetprint.com/.
[30] Maxon Motor. *Maxon DCX 19S*. May 2019. URL: https://www.maxonmotor.com/medias/sys_master/root/8830668505118/2018EN-78.pdf.
[31] Maxon Motor. *EPOS4 Module 50/5*. May 2019. URL: https://www.maxonmotor.com/medias/sys_master/root/8831824789534/EPOS4-Module-50-5-Hardware-Reference-En.pdf.
[32] Sebastian Knoop. "Design and Optimization of a Miniature Autonomous Drilling Rig-Contribution to the Drillbotics Competition 2018". MA thesis. NTNU, 2018.
[33] Lenze. *IE3 m550-P three-phase AC motors for inverter operation*. May 2019. URL: https://www.lenze.com/en-no/products/motors/ac-motors-inverter-operation/ie3-m550-p-ac-motors/.
[34] Lenze. *Automation systems Drive solutions*. May 2019. URL: https://www.lenze.com/fileadmin/lenze/documents/en/catalogue/CAT_GST_GFL_MF_15593808_en_GB.pdf.
[35] Lenze. *8400 TopLine frequency inverters*. May 2019. URL: https://www.lenze.com/en-no/products/inverters/control-cabinet-installation/8400-topline-frequency-inverters/.
[36] AEP Transducers. *TC4-AMP*. May 2019. URL: http://www.aeptransducers.com/force-transducers/104-tc4-amp.html.
[37] National Instruments. *USB-6212*. May 2019. URL: http://www.ni.com/en-no/support/model.usb-6212.html.
[38] RS. *RS PRO Solenoid Valve, 2 port , NC, 24 V dc*. May 2019. URL: https://uk.rs-online.com/web/p/solenoid-valves/1440801/.
[39] RS. *Omron Relay Socket for use with G2R-1-S Series*. May 2019. URL: https://no.rs-online.com/web/p/products/7948230/?grossPrice=Y&cm_mmc=NO-PPC-DS3A-_-google-_-3_NO_NO_Relay+Sockets_Omron_Exact-_-Omron+-+Relay+Sockets+-+7948230-_-p2rf05s&matchtype=e&kwd-306523915131&gclid=CjwKCAjw0ZfoBRB4EiwASUMdYWnB0LVI-fC-hJOhafG4ENo2YJhZ_CqipyiAqMOV-FimtZQGwMWxKxoCC5EQAvD_BwE&gclsrc=aw.ds.
[40] Aplicens A.S. *Pressure transmitter PCE-28*. May 2019. URL: https://www.aplisens.com/pc-28.html.
[41] TDK Invensense. *World's Lowest Power 9-Axis MEMS MotionTracking Device*. June 2017. URL: https://www.invensense.com/products/motion-tracking/9-axis/icm-20948/.
[42] TDK Invensense. *ICM-20948*. June 2017. URL: http://www.invensense.com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1.3.pdf.
[43] Silicon Labs. *EFM32 Gecko*. June 2018. URL: https://www.silabs.com/documents/public/data-sheets/efm32g-datasheet.pdf.
[44] 'Silicon Labs'. *'CP2104 USB-to-UART'*. June 2018. URL: 'https://www.silabs.com/documents/public/data-sheets/cp2104.pdf'.
[45] Wikipedia. *Photoresist*. Apr. 2019. URL: https://en.wikipedia.org/wiki/Photoresist.
[46] SEGGER. *J-Link Debug Probes*. Apr. 2019. URL: https://www.segger.com/products/debug-probes/j-link/.
[47] Digilent. *Digital Discovery*. Apr. 2019. URL: https://reference.digilentinc.com/reference/instrumentation/digital-discovery/start.
[48] 'JLC PCB'. *'JLCPCB.com'*. 2018. URL: 'https://jlcpcb.com/'.
[49] Silicon Labs. *Simplicity Studio 4*. Apr. 2019. URL: https://www.silabs.com/products/development-tools/software/simplicity-studio.
[50] Wikipedia. *I2C*. Apr. 2019. URL: https://no.wikipedia.org/wiki/I%C2%B2C.
[51] Circuit Basics. *Basics of UART Communication*. Apr. 2019. URL: http://www.circuitbasics.com/basics-uart-communication/.
[52] Mark Pedley. *High-Precision Calibration of a Three-Axis Accelerometer*. Apr. 2019. URL: http://cache.freescale.com/files/sensors/doc/app_note/AN4399.pdf.
[53] Wikipedia. *Hall effect*. May 2019. URL: https://en.wikipedia.org/wiki/Hall_effect.

[54] Schneider Electric. *What is Modbus and How does it work?* May 2019. URL: https://www.schneider-electric.co.in/en/faqs/FA168406/.

[55] National Instruments. *The Basics of CANopen*. June 2019. URL: https://www.ni.com/en-no/innovations/white-papers/13/the-basics-of-canopen.html.

[56] Maxon Motor. *EPOS*. May 2019. URL: https://www.maxonmotor.com/maxon/view/content/epos-detailsite.

[57] U.M. Azam J.M. Montoza H.E. Helle. *'Design and Implementation of an Autonomous Miniature Drilling RIg for Directional Drilling'*. 2019. URL: https://innsida.ntnu.no/litteratur%20(Posted%20during%20Autumn%202019.

[58] Ian Beavers. *The Case of the Misguided Gyro*. May 2019. URL: https://www.analog.com/en/analog-dialogue/raqs/raq-issue-139.html.

A Competition Guidelines

Society of Petroleum Engineers
Drilling Systems Automation
Technical Section (DSATS)
International University Competition
2018 – 2019

Drillbotics™ Guidelines

Revised 26 November 2018

1. Introduction

This year marks the fifth competition for the title of Drillbotics champion and a chance for students to learn about the drilling process from industry experts and for winning team(s) to travel and present a paper at the next SPE/IADC Drilling Conference and at an event organized by DSATS. The past years involved undergraduates, masters and doctoral students from a variety of disciplines who built innovative drilling machines and downhole tools while developing a deeper understanding of automating the drilling process. The university teams freely share lessons learned, which more rapidly advances the science of drilling automation. Everyone involved claims to have had a lot of fun while learning things that are not in the textbooks or published papers. Students also participated in related events at conferences, workshop meetings and networking with industry leaders in drilling automation. This year's contest promises to be just as challenging and hopefully as much fun.

How did the competition first come about? The origins began in 2008 when a number of SPE members established the Drilling Systems Automation Technical Section (DSATS) to help accelerate the uptake of automation in the drilling industry. DSATS' goal was to link the surface machines with downhole machines, tools and measurements in drilling systems automation (DSA), thereby improving drilling safety and efficiency. Later, at an SPE Forum in Paris, the idea of a student competition began to take shape. A DSATS sub-committee was formed to further develop the competition format and guidelines. Several universities were polled to find out the ability of academic institutions to create and manage multi-disciplinary teams. The Drillbotics committee began small in 2014-2015 to see if the format could succeed. With fine tuning, we continue along those lines as we start the 2019 process.

Version	Date	Section	Description
2019.01	10 Sept 2018	All	New Challenge, updated drillbit and formation, mandatory sensor requirements
2019.02	19 Nov 2018	3.1, 3.2, 3.4, 3.5, 3.7, 3.15, 8.2, Appendix A"	Modification to directional drilling target/objective/scoring, add details to bit specifications
2019.3	26 Nov 2018	3.2 3.5 and 3.6 3.7 7.0	Consolidated section Renumber/re-arrange section, allow for pilot hole and revise shipping info Bit diameter Timeline to reflect co-located tests

The 2019 competition has a few changes worth highlighting here:

- During previous competitions, the main focus was autonomously dealing with drillstring mechanics. Of course, this remains a concern, but the new test will focus on autonomous directional drilling.
- The 2019 rock sample shall be a homogeneous sandstone of known strength. Strength valuation will be provided well in advance of the competition date by the Drillbotics committee.
- Closed loop control of the rig based on downhole data is mandatory in this year's competition, not integrating this data set into the control algorithm is considered a "F - Failing grade" in this year's competition.
- Teams will kick off from vertical and are required to exit the rock sample within a defined target area, obtaining as much displacement as possible from well center along the north axis of the rock sample.
- The drill bit will be increased in diameter from last year's competitions by 1/8". Details will be provided via a post to the Drillbotics blog once the design is complete.
- Teams can choose to compete in a Group B Competition, which would use the prior year's rules/guidelines.
- The competition is to take place at a single industry facility in the USA and another facility in Europe. All teams are permitted to attend the presentations and Q&A of the other teams. All competitors at each location will start drilling at the same time.
- To attain a higher rating by the judges, the Phase I report should include a summary paragraph or table in the design report containing details of the control algorithm proposed. See section 3.4. This should be updated for the Phase II presentation to judges at the on-site test.
- The 2019 design should allow for third-party plug and play interface. See section 3.15.5. This is optional for 2019, but it will likely be mandatory in 2020.

The DSATS technical section believes that this challenge benefits students in several ways. Petroleum, mechanical, electrical and control engineers gain hands-on experience in each person's area of expertise that forms a solid foundation for post-graduate careers. They also develop experience working in multi-disciplinary teams, which is so important in today's technology driven industries. Winning teams must possess a variety of skills. The mechanical and electrical engineers need to build a stable, reliable and functional drilling rig. Control engineers need to architect a system for real-time control, including selection of sensors, data handling and fast-acting control algorithms. The petroleum engineers need an understanding of drilling dysfunctions and mitigation techniques. Everyone must work collectively to establish system functional requirements understood by each team member, properly model the drilling issues, and then to create a complete package working seamlessly together.

The oil and gas industry today seeks lower costs through efficiency and innovation. Many of the student competitors may discover innovative tools and control processes that will assist drillers to speed the time to drill and complete a well. This includes more than faster ROP, such as problem avoidance for dysfunctions like excessive vibrations, stuck pipe, and wellbore stability issues. Student teams built new downhole tools using 3D printing techniques of designs that would be difficult, if not impossible to machine. They used creative hoisting and lowering systems. Teams modeled drilling performance in particular formations and adjusted the drilling parameters accordingly for changing downhole conditions. While they have a lot to learn yet about our business, we have a lot to learn about their fresh approach to today's problems. Good Luck!

the DSATS Drillbotics Committee

Shashi Talya	(chair)	Pat Derkacz	Joachim Oppelt
Aaron Logan	(co-chair)	Frode Efteland	Neil Panchal
Fred Florence	(co-chair)	James Franks	Luis Pereira
Trey Adams		Jana Hochard	Marco Perez
Víctor Hugo Soriano Arámbulo		Mark Hutchinson	Bhavesh Ranka
Mike Attrell		Jayesh Jain	Geir Skaugen
Vimlesh Bavadiya		John Macpherson	Sami Sultan
Sylvian Chambon		George Michalopulos	Majid Tariq
John Clegg		Alex Ngan	Suresh Venugopal
Dmitriy Dashevskiy		Nii Nunoo	Kurt West

Contents

Objectives for the 2019 Competition

1.1. During the school year beginning in the fall of 2018, a team of students will organize themselves to solve a drilling related problem outlined in item 3 below. The team should preferably be a multi-disciplinary team that will bring unique skills to the group to allow them to design and construct hardware and software to demonstrate that they understand the underlying physics, the drilling issues and the usual means to mitigate the issues. We cannot stress enough the need to involve students with different technical training and backgrounds. They will need to develop skills to understand drilling dysfunctions and mitigation strategies, but they must also have the mechanical engineering capabilities to design the rig/drilling package. In past years, some entrants have not adequately considered the control network and algorithms needed for autonomous drilling. They have often misunderstood the need for calibrated sensors and fast, accurate data handling. All of this and more is needed to build and operate a complete automated drilling system.

1.2. The students could produce novel ideas leading to new drilling models, improved drilling machines and sensors, and the ability to integrate the data, models and machines that will hopefully create new, more efficient ways to drill wells in the future. Any such innovation will belong to the students and their university in accordance with the university's written policies. DSATS and SPE waive any claims to students' intellectual property.

1.3. The students, working as a multi-disciplinary team, will gain hands-on experience that will be directly applicable to a career in the upstream drilling industry.

2. Background

2.1. What is DSATS?

2.1.1. DSATS is a technical section of the Society of Petroleum Engineers (SPE) organized to promote the adoption of automation techniques using surface and downhole machines and instrumentation to improve the safety and efficiency of the drilling process. More information is available about DSATS at the DSATS homepage (http://connect.spe.org/DSATS/Home/).

2.1.2. The Drillbotics website at www.Drillbotics.com includes official updates to the competition guidelines and schedule, as well as FAQs, photos, and previous entrants' submittals and reports. Any updates to the guidelines posted on the Drillbotics website via blog entries from the

Committee is considered to be an official revision to these Guidelines. Questions and suggestions can be posted there, or teams can email the sub-committee at 2019@Drillbotics.com.

2.2. Why an international competition?

2.2.1. DSATS, as part of the SPE, is a group of volunteers from many nations, connected by their belief that drilling automation will have a long-term, positive influence on the drilling industry. This diversity helped to shape the direction of the organization. The group feels that the industry needs to attract young professionals from all cultures and disciplines to advance drilling practices in all areas of the world. The winners of the Group A competition will receive a grant for economy class transportation and accommodations to attend the next SPE Drilling Conference and will present an SPE paper that will be added to the SPE archives of One Petro[1]. Winners of Group B will publicly receive recognition of their achievement, and have the opportunity to publish an SPE paper that will be added to the SPE archives of One Petro. DSATS believes recognition at one of the industry's leading technical conferences will help encourage student participation. Also, the practical experience with drilling automation systems increases the students' visibility to the companies that are leading automation activities.

3. *Competition Guidelines*

3.1. Problem statement for the 2018-2019 competition:

Design a rig and related equipment to autonomously drill a well, using downhole sensors, that obtains as much horizontal displacement from surface as possible along the rock's "north" direction, as quickly as possible while maintaining borehole quality and integrity of the drilling rig and drillstring.

3.2. 2018-2019 High Level Challenge and Judging Changes

3.2.1. The competition will take place on the same day for all teams in North America, and the same day (likely different from the North American date) for all teams in Europe. (see 3.1.6, 3.5 and 7.0)

3.2.2. Inclination and Azimuth directionality is part of the competition for 2019. (see 3.5) The wellbore must be started vertically and then kicked off below a specified depth to build as much angle as

[1] Publication is subject to the SPE program committee's acceptance of the abstract/paper. If the abstract is not accepted, DSATS will solicit other SPE events try to get the paper into OnePetro.

possible along a specific well path. Teams score more points if the displacement is in the north direction along the centerline of the rock (see Appendix "A" for scoring details)

3.2.3. Downhole sensors are mandatory, and it is also mandatory to implement their data into the control algorithm of the rig. A severe penalty will be applied to teams who do not use downhole sensors. Closed loop control of the rig based on downhole data is mandatory in this year's competition, not integrating this data set into the control algorithm is considered a "F- Failing grade" in this year's competition.

3.2.4. A homogeneous sandstone Rock Sample will be provided by Drillbotics at the test sites. (see 3.6)

3.2.5. DSATS to provide a new bit with 1/8" larger diameter (1.25" new diameter) and 2" length. Students are permitted to use their own drillbit for the 2019 competition. (see 3.7)

3.2.6. Groups A&B

3.2.6.1. All returning teams must enter Group A, and are judged according to these guidelines. First time entrants may join Group A or Group B. Group B competitors will be judged by the previous year's guidelines (2017-2018).

3.2.6.2. New teams may choose in which Group they will compete with six (6) weeks' notice before the earliest drilling competition date. This notice is required for the committee to build and ship the custom rock sample.

3.2.6.3. Prizes are described in sections 9 and 10 below.

3.2.6.3.1. The "big prize" will be awarded to the winner of challenge Group A: an SPE Whitepaper published in OnePetro, economy class transportation and accommodations to attend and present at the next SPE Drilling Conference, with their rig presented at the drilling conference, subject to conference guidelines.

3.2.6.3.2. The winning Group B team will have an opportunity to present and publish a SPE Whitepaper, subject to the conference guidelines, and they will also receive recognition of the accomplishment at the conference

3.2.7. Additional information regarding the judging of the competition is detailed in section 3.16.

3.3. Two Project Phases

Fall Semester 2018

The first phase of the project is to organize a team to design an automatic drilling machine to solve the project problem. It is not necessary to build any equipment in this phase, but it is okay to do so. Design considerations should include current industry practices and the team should evaluate the advantages and shortcomings of today's devices. The design effort may be assisted by university faculty, but the students are encouraged to introduce novel designs for consideration. The level of student, faculty and technical staff involvement shall be reported when submitting the design. For returning teams, the Phase I Design should include an analysis of data and learnings from previous ("offset") wells drilled.

Spring Semester 2019

During the second phase, the finalist teams selected by DSATS proceed to the construction and drilling operation will use the previous semester's design to build an automated drilling machine. As per industry practices, it is common during construction and initial operations to run into problems that require a re-design. The team may change the design as needed in order to solve the problem subject to section 4.3.4. Teams may use all or part of a previous year's rig.

See section 7 for detailed timeline information.

3.4. Phase I – Design Competition

Design an automated drilling machine in accordance with the rules below.

3.4.1. DSATS envisions a small (perhaps 2 meters high) drilling machine that can physically imitate the functionality of full-scale rig machinery. (Since the winning machines will be presented at the SPE conference, there may be height restrictions imposed by the conference facility, so machines that are too tall may not be allowed on the exhibit floor.) The machine will be the property of the university and can be used in future research and competitions. New and novel approaches that improve on existing industry designs are preferred. While innovative designs are welcome, they should have a practical application to drilling for oil and gas.

3.4.2. The drilling machine will use electrical power from the local grid not to exceed 25 horsepower. Lower power consumption resulting from energy efficient designs will receive additional consideration.

3.4.3. The design must provide an accurate and continuous measurement of Weight-On-Bit (WOB), inclination, azimuth, and depth; as well as other drilling parameters, that should be presented as a digital record across the period of the test. All depth related measurements shall use the rig floor as the datum, not the top of the rock. Appropriate statistical measurements should be made at frequencies and with an accuracy and appropriate frequency content for the dynamics of the drilling system both at surface and downhole. Discussion of such choices should be included in the design report.

3.4.4. The proposed design must be offered in Phase I of the project, but changes are allowed in Phase II, as long as they are reported to the Committee via students' monthly reports. A summary of all significant changes, including the reason modifications were necessary, must be included in the students' final report.

3.4.5. Design submittal by the students shall include:

3.4.5.1. Engineering drawings of the rig concept, mechanical and electrical and auxiliary systems, if any

3.4.5.2. Design notes and calculations

3.4.5.2.1. All engineering calculations shall be included in the Phase I report, even if the rig is built using previous years' designs. This ensures that the 2019 team reviewed and understood the previous design assumptions and calculations.

Calculations should include each formula considered in the design, a reference that shows the origins of the formula, why is was chosen, what engineering assumptions were made, a definition of all variables and the values used in the calculation.

Example

Buckling limit	Euler's Equation	(1) cite a reference here or in the reference section of your design report

The critical buckling load, bcr, is calculated:

$Pbcr = \pi 2 * E * I / (K * L)^2$

$Pbcr$: Critical buckling load

E: Modulus elasticity of the aluminum drill pipe

I: Area moment of inertia

L: Length of the column

K: Column effective length factor (explain how you chose the appropriate k or n factor)

3.4.5.2.2. The report should include a table that summarizes ALL calculations.

Example

Calculations	***Formula***	***Reference***	***Results***
Moment of Inertia	$I = \pi / 64\ (dp\ 4 - idp\ 4)$	*Thin wall approx. or ID/OD calc separately or other? List your reference*	$0.000546\ in4$
Buckling Limit	$Pbcr = \pi 2 * E * I / (K * L)^2$	*Euler's Eq*	$18.9\ kg$

3.4.5.3. Control system architecture. (The response time of measurements, data aggregation and control algorithms should be estimated.)

3.4.5.4. Key features for any models and control software.

3.4.5.5. Proposed data handling and display.

3.4.5.6. Specification for sensors, signal processing and instrumentation, (verifying their accuracy, precision, frequency response and environmental stability), including the methods planned for calibration before and after the Phase II testing.

3.4.5.7. Plan for instrumentation of sensors in the BHA, as well as a method to synchronize all measurements and utilize both the surface and downhole sensors for real-time control of the drilling process.

3.4.5.8. An explanation of the implementation of the output of the BHA sensors to improve the trajectory of the wellbore, drilling efficiency and other drilling concerns.

3.4.5.9. An explanation of the algorithm used to autonomously control the drilling rig based on the output of the BHA sensors

3.4.5.10. An explanation of the principles being applied to directionally steer the wellbore into the defined target area (see Appendix "A") with the intent to score the maximum amount of points

3.4.5.11. Cost estimate and funding plan

3.4.5.12. A design summary video used to outline the design submittal not to exceed five (5) minutes in length. Videos shall be the property of the university, but DSATS shall have the rights to use the videos on its websites and in its meetings or events.

3.4.5.13. All design, construction and operation of the project are subject to the terms and conditions of section 11.

3.4.5.14. A safety case shall be part of the Phase I design. Include a review of potential hazards during the planned construction and operation of the rig, and for the unloading and handling of any rock samples or other heavy items. An example of a safety case will be posted on the Drillbotics.com website.

3.4.6. A committee of DSATS members (the Committee) will review the Phase I designs and select the top five (5) teams[2] who will progress to Phase II of the competition.

3.4.7. DSATS shall also award a certificate of recognition and publication on its website for the most innovative design. The design video will also be shown at the DSATS automation symposium at SPE conferences.

3.4.8. DSATS will not fund any equipment, tools, software or other material, including labor, for the construction of the rig. Student teams are encouraged to find external funding from industry participants and suppliers.

[2] The number of finalists could be increased or decreased by the DSATS Board of Directors subject to available funding.

3.5. Phase II – Drilling Competition

3.5.1. In the spring term of 2019, qualifying teams will build the rig and use it to drill rock samples provided by DSATS. Drilling a well, deviated toward the defined target area (see Appendix "A"), efficiently though the sample while controlling drilling dysfunctions is the primary technical objective of the competition. The exit point of the drillbit will be judged based on how much displacement from well center is obtained, with weighting applied according to the direction of the displacement. The use of both surface and downhole measurements to control the drilling process in real-time is mandatory, failure to do so will result in a failing grade. To avoid disqualification due to a downhole sensor failure, redundant or immediately replaceable items should be part of the design and implementation. Time to replace a sensor will be added to the drilling time for calculation of ROP.

3.5.2. The teams are to use manual control to pre-drill a vertical pilot hole not more than 1" deep measured from the rock's top face. This hole is to be drilled using the competition drilling rig. Location of this pilot hole will be marked on each sample by the committee at the intersection of two lines drawn from opposite corners of the rock sample.

3.5.3. Teams may use glue or use a mechanical fastener to attach a bell nipple or diverter housing to the top of the rock to allow connection of a flowline for return mud flow. The maximum allowable length of the bell nipple is 8 inches. If you use a fastener, be careful not to break the rock.

3.5.4. When the competition drilling begins, Teams competing in Group A will be required to continue to drill the pilot hole vertically to the kick off point. The kick off point may be at any depth greater than 4" below the surface of the rock.

3.5.5. Navigation shall be done autonomously, without any manual intervention.

3.5.6. No lateral forces are allowed to be applied above the rocks top face

3.5.7. No forces are allowed to be applied external to the rock that will force the drillbit in a particular direction

3.5.8. External magnetic field effects from the drilling rigs will be present on the directional sensors used to drill the wellbore. The industry has accepted practice of magnetic ranging, and this may be a technique worth investigating to improve the signal to noise of magnetic measurements

3.5.9. Once drilling commences, the test will continue until the drillbit exits the rock sample, or three (3) hours, whichever comes first.

3.5.10. Drilling performance will be observed and measured by Drillbotics judges invited to attend and witness the test.

3.5.11. DSATS will judge the competitors primarily on their ability to obtain as much displacement from well center as possible along the north axis of the rock sample; closer proximity to this axis and larger amounts of displacement point receives higher marks in this category (see Appendix "A" for details)

3.5.12. DSATS will run a flexible "casing" into the wellbore, and use this to gauge the borehole quality

3.5.12.1. Casing will be nearly equal in diameter to the 1.25" drillbit

3.5.12.2. An over gauge, and under gauge Casing will also be used as a no-go measurement

3.5.13. The final test will be scheduled late in the school year or soon after graduation. The test will occur at two locations, so teams must allow time to ship their rig from their university in accordance with the timeline per section 7 below.

3.6. Rock Samples

3.6.1. DSATS will prepare a set of nearly identical homogeneous sandstone samples appx. 12"W x 24"L x 24"H (30 x 60 x 60 cm) that will be shipped to each test site. It will not be sent to the schools. A smaller sample could be provided (no smaller than 12"W x 12"L x 24"H), which will be announced not later than March 1st.

3.6.2. The rock sample will be homogeneous sandstone, and rock compressive strength values will be provided for the sandstone samples furnished by DSATS. The Drillbotics committee will mark the surface of rock to indicate the well center where drilling will start. It will be located at the intersection of two lines drawn from opposite corners of the rock sample.

3.6.3. The university and/or students may acquire or produce rock samples as needed to verify the design and allow students to practice using their machine prior to the test. Drilling of the samples provided by DSATS prior to Phase II testing is not allowed and could lead to disqualification, except for the pilot hole.

3.6.4. The sandstone sample will be oriented during drilling so that it rests on a 12”x24” face so that the drilled depth will be 24”.

3.7. Bits

3.7.1. Upon request, DSATS will send a drillstring and bit to the finalist teams for use in Phase II. It is expected that the BHA and pipe will cause some difficulty, both for causing drilling dysfunction and for sensor integration and data telemetry. The judges will look for creative concepts supported by sound reasoning showing an understanding of how the BHA, bit and drillstring function together, and how the downhole system measures, samples and transmits the drilling data.

3.7.2. Upon request, the bit shall be returned to the Committee following Phase II testing for reconditioning for use in future competitions.

3.7.3. One (1) PDC bit will be provided by DSATS to be used during the Phase II tests. For 2018-2019 the bit will be:

3.7.3.1. A micro-bit 1.25" in (31.75 mm) diameter and 2.0” in total length.

3.7.3.2. Low axial aggressiveness and high side aggressiveness (i.e. high bit anisotropy).

3.7.3.3. An error when posting information about the bit diameter may have led to confusion and could affect teams who designed equipment based on the erroneous diameter. Therefore, teams may substitute a bit of their own design or a purchased bit not to exceed 1.5 inches in diameter and not more than 2 inches long. While this may allow some additional space for building the downhole equipment, it could also affect the build rate of the directional wellbore. It may also induce additional torque that could affect drillstring forces. Teams must evaluate the benefits and drawbacks of each and are encouraged to provide this analysis in their Phase I design report that describes what different forces are expected and their impact on drilling operations.

3.7.4. Students are encouraged to consider bit wear prior to the final test and its impact on drilling performance during the onsite testing. Based on prior competitions, bit wear should be minimal but some cutter damage is always possible.

3.7.5. Student teams may build or buy similar drill bits to test their design with the rock samples they sourced.

3.7.6. For the final competition, the students may use the directional drill bit provided by DSATS, or use their own bit design. However, the dimensions of their bits must not exceed 1.5 inches in diameter and 2 inches long. This provision is made to enable students to fully optimize the bit design for their specific directional system.

3.8. Drillpipe

3.8.1. The drill string provided by DSATS, if requested by the student teams, will be chosen to ensure drilling dysfunctions will be encountered. How these dysfunctions are mitigated is a key objective of the competition. Final details of the construction of this drill string will be furnished in late fall of 2018 to all entrants upon request. Preliminary specifications are listed below to assist with the mechanical and electrical design of the rig.

3.8.2. The drill pipe specifications for the 2018-2019 competition are subject to change, but should be:

3.8.3. Round Aluminum Tube 3/8 inch diameter x 36 inches long; 0.049 inch wall or equivalent

3.8.4. The material from KS Precision Metals is a typical low alloy material: "Our Aluminum tubing with wall thickness of .035 or .049 is 6061 T6"

3.8.5. DSATS will provide, upon request, the finalists four (4) joints of pipe. Any additional pipe needed can be purchased by the student teams or university if needed.

3.8.6. The use of a metric equivalent of the tubing is permitted.

3.8.7. Tubing is usually available from various hobby shops such as K-S Hobby and Craft Metal Tubing and via Amazon and other suppliers.

http://www.hobbylinc.com/htm/k+s/k+s9409.htm

ROUND ALUMINUM TUBING

OUTSIDE DIAMETER INCHES	WALL THICKNESS	ID
3/64 (.047)	.014	.019
1/16 (.0625)	.014	.035
5/64 (.078)	.014	.050
3/32 (.094)	.014	.066
	.016	.062
7/64 (.109)	.014	.081
1/8 (.125)	.014	.097
9/64 (.141)	.014	.113
5/32 (.156)	.014	.128
11/64 (.172)	.014	.144
3/16 (.187)	.014	.159
	.022	.143
	.035	.117
	.049	.089
13/64 (.203)	.014	.175
7/32 (.219)	.014	.191
	.022	.175
	.035	.149
15/64 (.235)	.014	.207
1/4 (.250)	.014	.222
	.016	.218
	.022	.206
	.035	.180
	.049	.152
9/32 (.281)	.014	.253
	.016	.249
5/16 (.312)	.014	.284
	.016	.280
	.035	.242
	.049	.214
11/32 (.344)	.016	.312
3/8 (.375)	.016	.343
	.035	.305
	.049	.277
13/32 (.406)	.016	.374
7/16 (.437)	.016	.405
	.035	.367
15/32 (.468)	.016	.436
1/2 (.500)	.016	.468
	.035	.430
17/32 (.531)	.016	.499
9/16 (.562)	.016	.530
5/8 (.625)	.016	.593

3.9. Tool joints

3.9.1. Students may design their own tooljoints as long as the design concept is included in the Phase I proposal.

3.9.2. Alternately, students may use commercially available connectors/fittings attached to the drillpipe using threads, epoxy cement or other material, and/or may use retaining screws if desired, as long as the design concept is included in the Phase I proposal.

3.9.2.1. A fitting used somewhat successfully in 2017 is available from Swagelock. In 2018, the winning team used a fitting from Vertex.

3.9.2.2. A fitting used successfully in 2016, but which did not work well in 2017, is available from Lenz (http://lenzinc.com/products/o-ring-seal-hydraulic-tube-fitting/hydraulic-straight-connectors) that uses a split-ring to allow a torque transfer across the fitting.

3.9.3. Students must state WHY they choose a tooljoint design in the Phase I proposal.

3.10. Bit sub/drill collar/stabilizers

3.10.1. It is expected that each team will design and build their own bit sub, instrumentation of the bit sub is ideal for directional sensors

3.10.2. Additional weight may be added to the bit sub, or surface weight/force (above the rock sample) may be applied to provide weight on bit and drillpipe tension

3.10.3. Stabilizers are permitted but will be limited in length. Advise the committee of your choice and why and include this in the Phase I design for committee consideration.

3.10.4. Students must add sensors to the drillstring, but are not permitted to instrument the rock samples. They must have a smaller diameter than the stabilizers and bit by at least 10%. Please include design concepts in the Phase I design.

3.10.5. The addition of along-string sensors to measure vibrations, verticality and/or tortuosity or other parameters will receive extra consideration. They must have a smaller diameter than the stabilizers and bit by at least 10%.

3.11. Automated Drilling

3.11.1. Drilling automation should be considered a combination of data, control AND dynamic modeling so that the control algorithm can determine how to respond to differences between the expected and actual performance. Process state detection can often enhance automation performance. Refer to documents posted on the DSATS website for more information.

3.11.2. Once drilling of the sample commences, the machine should operate autonomously. Remote operation and/or intervention is not allowed.

3.11.3. All directional steering should be autonomously controlled by the drilling rig

3.12. Sensors

3.12.1. The team may elect to use existing oilfield sensors or may look to other industries for alternate sensors.

3.12.2. The team may develop its own sensors if so desired.

3.12.3. Sensor quality differs from data quality. Both are important considerations in this competition.

3.12.4. The final report shall address which sensors were selected and why. The sensor calibration process shall also be explained.

3.13. Data collection and handling

3.13.1. The team may elect to use standard data collection and recording techniques or may develop their own. Data handling techniques and why they were chosen should be described in the Phase I submittal.

3.13.2. The final report shall address which data systems were selected and why.

3.13.3. The observed response time of measurements, data aggregation and control algorithms should be compared to the Phase I estimates.

3.14. Data visualization

3.14.1. Novel ways of presenting the data and progress of drilling in real time while drilling will receive particular attention from the judges.

3.14.2. Visualization of the processes (automation, optimization, drilling state, etc.) should be intuitive and easily understood by the judges, who will view this from the perspective of the driller operating a rig equipped with automated controls.

3.14.3. Data must be presented in a format that allows the judges to easily determine bit depth, elapsed drilling time, ROP, MSE, verticality/inclination, vibration, and any other calculated or measured variable used to outline the drilling rigs performance to the judges. Lack of an appealing and usable Graphic User Interface (GUI) will be noted to the detriment of the team.

3.14.4. All depths shall use the industry-standard datum of rotary/kelly bushing interface (RKB), which should be the top of the rig's "drill floor."

3.15. Measure and analyze the performance

3.15.1. The drilling machine should react to changing "downhole" conditions to select the optimal drilling parameters for improved performance, as measured by the rate of penetration (ROP), mechanical specific energy (MSE), verticality, cost per foot or meter, and other standard drilling measures or key performance indicators. Adding parameters such as MSE, or similar features, to the control algorithms will receive special attention from the judges.

3.15.2. Design limits of the drilling machine shall be determined and shall be incorporated in the programming of the controls during the construction phase.

3.15.3. Downhole measurements from directional sensors are to be used for adjusting drilling parameters and control of drilling machines used to aid in directional drilling

3.15.4. The final report (see Clause 3.19) shall outline drilling performance and efficiency criteria and measured results.

3.15.5. One of DSATS' goals is to promote plug and play capability to accelerate the implementation of drilling automation. A DSATS committee is preparing definitions and examples of proposed data communication protocols and interfaces. Once this is available, the Drillbotics competition will require the use of these standard protocols. This will not be a requirement for 2019 but it will be included in future competitions. Links to these standards will be added to the Drillbotics.com website when they are published.

3.16. The test well:

3.16.1. The competition will take place on the same day for all teams in North America, and the same day (likely different from the North American date) for all teams in Europe. The competition will take place at a facility capable of hosting all drilling rigs at the same time, and capable of having all drilling rigs start drilling at the same time (Group A and B). Location of drilling to be conducted indoors, in a location that does not have an unusually distorted magnetic field.

3.16.2. Prior to the commencement of the test, teams will attach a bell nipple per 3.5.3. They will then manually drill the pilot hole not to exceed 1" deep.

3.16.3. When the test begins, the teams will start drilling autonomously by continuing to drill the pilot hole, keeping the wellbore as vertical as possible until reaching the kick-off point. All rigs start the drilling competition at the same time (Group A and Group B)

3.16.4. The teams will kick off from vertical at any depth below the 4" vertical surface hole

3.16.5. The teams will target an exit of the directional wellbore within the defined target area. The target of exit is a primary judging metric, landing within the defined target area scoring points as described in Appendix "A"

3.16.6. No lateral forces may be applied above the rock.

3.16.7. Drilling will stop at 3 hours or when the last team exits the rock sample.

3.16.8. Should be drilled with a maximum allowable Weight-On-Bit dependent on the rig and dynamic drillstring integrity.

3.16.9. Will not require a closed-loop fluid circulation system, but could be of advantage for directional drilling, the bit and machinery should be cooled with air or fluid/water if needed. The design of the fluid system, if any, should be included in the Phase I design.

3.16.10. The rock sample will be homogeneous and will be capable of aiding in closed-loop fluid circulation. Note that the rock samples will leak once the drillbit punctures a rock face, so a rig design that includes a containment system is required.

3.16.11. Will require casing to fit in the directional wellbore. The ability to "run casing" is the secondary judging metric. Judges will run a "flexible" casing used as a gauge of borehole quality

3.16.12. Will not require a rig move, walking or skidding, but the mobility of the rig will be considered in the design phase.

3.17. Not included in the 2018-2019 competition

3.17.1. The drilling will not include automating the making or breaking of connections. If connections are necessary due to the rig and drillstring design, connections should be made manually, and the time involved with the connections will be included with respect to its effect on drilling performance (rate of penetration reduction).

3.18. Presentation to judges at Phase II Testing

3.18.1. The judges will arrive at the centralized test site facility to meet with the student teams and advisors immediately prior to the Phase II testing. DSATS will provide a suitable meeting room for discussion lasting about two hours.

3.18.2. The students will present a BRIEF summary of their final design, highlighting changes from their Phase I design, if any. Include an explanation of why any changes were necessary, as this indicates to the judges how much students learned during the design and construction process. Explain what measurement and control features have been deployed. Describe novel developments or just something learned that was worthwhile. Also include how actual expenses compared with the initial estimate. (Previous teams used a short PowerPoint presentation of about ten slides or so. Use any format you like.) Be sure to include all your team members as presenters, not just one spokesperson. At some time during your talk, let us know who the team members are and what background they have that pertains to the project.

3.18.3. Judges will ask questions to ascertain additional details about the design and construction process and to see if all team members have a reasonable understanding how all the various disciplines used for the rig design and construction fit together.

3.18.4. All teams may sit in for the presentations and Q&A of the other teams. The order of presentation will be determined by drawing lots.

3.19. Project report

3.19.1. The student team shall submit to DSATS a short monthly project report that is no more than one page in length (additional pages will be ignored) due on or before the last day of each month that will include:

3.19.2. Phase I

- Key project activities over the past month.
- Rig design criteria, constraints, tradeoffs, and how critical decisions were determined
- Cost updates
- Significant new learning, if any

3.19.3. Phase II

- Construction issues and resolution
- Summary of recorded data and key events
- Drilling parameters [such as WOB] and how they impact the test
- Other items of interest

3.19.4. Report content

3.19.4.1. To teach students that their work involves economic trade-offs, the monthly report should include at a minimum a summary estimate of team member labor hours for each step in the project: design, construction, testing, reporting, and a cost summary for hardware and software related expenditures. Also include labor for non-students that affect the cost of the project. Labor rates are not considered, as to eliminate international currency effects. Labor is not considered in the cost limits of item 6.1, but should be discussed in the report and paper.

3.19.5. File naming convention

3.19.5.1. To avoid extra work by the committee to rename all files, please use this convention for:

3.19.5.1.1. Monthly reports

Year-Month# University Name (abbreviated)

(note this is the competition year (spring term))

Example 2019-09 UDC

3.19.5.1.2. Design reports

Year University Name (abbreviated)

(note this is the competition year (spring term))

Example 2019 University of Drillbotics Competition

3.20. Final report and paper

3.20.1. The finalists shall prepare a project report that addresses the items below. We suggest you use the format of most SPE papers. For reference, please see http://spe.org/authors/resources/

3.20.2. The winning team of Group A and Group B shall update the report as needed to comply with SPE paper submittal guidelines to write a technical paper for publication by the SPE at its Annual Drilling Conference. SPE typically requires that the manuscript is due in the fall following the Phase II test. While the Drillbotics committee will make every effort to have the paper presented during the Drilling Conference, the SPE Program Committee has authority over which papers will be accepted by the conference. If the paper is not accepted by the conference, the Drillbotics committee will endeavor to have it presented at the DSATS Symposium and will use its contacts to have the paper published via other related SPE conferences.

3.20.3. The report, paper and all communications with DSATS shall be in the English language. The presentation will be made by at least one member of the student team.

3.20.4. The timing for submittal of the abstract and paper will be the published deadlines per the call for papers and conference guidelines as posted on the SPE's website (www.spe.org).

3.20.5. The abstract must generate sufficient interest with the SPE review committees to warrant publication, although DSATS will help promote acceptance where possible

3.20.6. The paper should address at a minimum

3.20.6.1. The technical and economic considerations for the rig design, including why certain features were chosen and why others were rejected.

3.20.6.2. The setup of the experimental test, the results and shortcomings.

3.20.6.3. Recommendations for improvements to the design and testing procedures.

3.20.6.4. Recommendations for improvements by DSATS of the competition guidelines, scheduling and provided material.

3.20.6.5. Areas of learning gained through the competition not covered in the university course material.

3.20.6.6. A brief bio or CV of the team members and their sponsoring faculty.

4. Team Members

4.1. DSATS envisions that the students would be at least senior undergraduate or Masters level, well versed in the disciplines needed for such a project. The maximum number of students per team is five (5) and the minimum shall be three (3). Any team that loses team members during the project can recruit a replacement.

4.2. At least one member of the team must be a Petroleum Engineering candidate with sufficient coursework completed to understand the physics relating to the drilling problems and the normal industry practices used to mitigate the problem.

4.3. Students with a background in mining, applied mathematics, mechanical and electrical engineering, as well as controls, mechatronics and automation or software development, are the most likely candidates, but students with any applicable background is encouraged.

4.4. A multi-disciplinary team simulates the working environment in the drilling industry today, as most products and services are produced with the cooperation of technical personnel from differing backgrounds and cultures.

4.5. A university may sponsor more than one team but must submit only one team/design for Phase II evaluation.

4.6. Students shall register their team not later than 30 November using the registration form on the Drillbotics website. Any changes to the team members or university supervisor over the course of the competition should be reported in the monthly reports.

5. Expenditures

5.1. Teams selected to advance to the second phase must limit the cost of the rig and materials to US$ 10,000 or its equivalent in other currencies. The students shall find a source of funding and report the source in the Phase I proposal. All funding and procurement should comply with university policy.

These funds are intended to cover the majority of expenses for hardware, software and labor to construct and operate the team's equipment. DSATS shall not be liable for any expenditure other than DSATS provided material and specified travel expenses.

5.2. DSATS will assist when possible to obtain free PLCs or similar control devices from suppliers affiliated with the DSATS organization. Such "in-kind" donations shall not be included in the team's project costs.

5.3. Students and universities may use other "in-kind" contributions which will not be included in the team's project costs. Such contributions may include modeling software, laboratory equipment and supplies, and similar paraphernalia usually associated with university laboratory projects.

5.4. Any team spending more than US$ 10,000, or its equivalent in other currencies, may be penalized for running over budget.

5.5. DSATS reserves the right to audit the team's and university's expenditures on this project.

5.6. Any devices built for the project will become the property of the university and can be used in future research and competitions. Any maintenance or operating costs incurred after the competition will not be paid by DSATS.

6. *Other Considerations*

6.1. The design concepts shall be developed by the student team under the supervision of the faculty. Faculty and lab assistants should review the designs to ensure student safety.

6.2. Construction of the equipment shall be supervised by the student team, but may use skilled labor such as welders and lab technicians. The use of outside assistance shall be discussed in the reports and the final paper. DSATS encourages the students to gain hands-on experience with the construction of the rig since this experience will be helpful to the career of individuals in the drilling industry.

6.3. University coursework and credit: Each university will decide whether or not this project qualifies as a credit(s) towards any degree program.

7. *Project Timeline*

Phase I - Design:	Fall 2018
Submit monthly reports	On or before the final day of each month
Submit final design to DSATS	31 Dec 2018, midnight UTC
Submit an abstract to DSATS*	31 Dec 2018, midnight UTC

*DSATS will submit an abstract to the SPE that will include excerpts from the student abstracts by the conference paper-submittal deadline, typically in mid-summer, for consideration of a paper by the conference program committee.

Phase II – Construction and Testing	Spring 2019
DSATS to announce finalists	On or about 15 Jan 2019
Construction	Spring 2019
Monthly reports	On or before the final day of each month
Drilling Test	Specific on-site test locations and dates for the North American and European locations will be arranged not later than 31 March 2019. The testing will typically occur in late May and/or early June.
The timeline for the Phase II tests:	Day 0 Students arrive Day 1 Students rig up; judges arrive Day 2 Students present to judges Day 3 Performance tests Day 4 Students rig down and depart
Shipping of rig to test site	The rig should be shipped to arrive no earlier than 10 days before the test date. Each team will coordinate with the committee and provide any documentation necessary.
Prepare and submit paper	Per SPE deadline*
Prepare and submit presentation	Per SPE deadline
Present paper at the Drilling Conf	Per SPE and DSATS schedule

8. *Evaluation Committee*

8.1. DSATS will select an evaluation committee from its membership

8.2. Criteria/Weighting (see chart):

Criteria	Parameter	Weighting
Phase I:		
a. Safety	Safety: construction and operation	10
b.Mobility of rig	Rig up, move, rig down	5
c.Design considerations and lessons learned		10
d.Mechanical design and functionality, versatility		25
e.Simulation/Model/Algorithm		25
f.Control scheme	Data, controls, response times	25
	Total	100%
Phase II:		
a.Creative Ability	Analysis, concepts, development	10
b.Engineering Skills	Problem/Goal, design criteria, feasibility	10
c.Construction Quality		10
d.Cost Control		10
e.Performance		30
Various parameters such as:	ROP, MSE, Landing Bit, Inclination, and other	
Are these used within the control algorithms		
Exit of drillbit within defined target area (see Appendix "A" for details)	Optimal landing of bit	
f.Quality of wellbore	Tested using the Go-No-Go flexible 'Casing'	10
	Verticality, tortuosity, caliper, other	
g. Data	Data handling, data visualization, data comparison to judges' wellbore logs, and other	20
h. Downhole Sensor Data Used in Control Algorithm	**Pass/Fail**	**Pass/Fail**
	Total	100%
Intangibles	Additional score may be added or subtracted by the judges at their discretion	

9. Group A Prizes

9.1. The winning team of Group A will be sponsored by DSATS to attend the next SPE/IADC Drilling Conference to present a paper that explains their project in detail.

9.1.1. The program committee of the Drilling Conference awarded the Drillbotics subcommittee a permanent slot in one of the drilling sessions at the conference. As per SPE's customary procedures, the paper will be archived in OnePetro. In addition, SPE has agreed to furnish a booth in the exhibition area during the conference where the team can erect their rig and describe its operation to the conference attendees. This is an excellent opportunity for students to network with the industry.

9.2. Upon submittal to DSATS of a valid expense statement (typically a spreadsheet supported by written receipts) of covered expenses will be reimbursed by the treasurer of DSATS for the following:

9.2.1. Reasonable shipping costs of the Drillbotics rig to and from the conference as long as charges are pre-approved by the chair or co-chair of the Drillbotics subcommittee.

9.2.2. Round trip economy airfare for the team and one university sponsor/supervisor to the gateway city of the next SPE/IADC Drilling Conference. Entrants should use the SPE approved carrier where possible to minimize cost. Airfares that exceed the SPE rate must be pre-approved by the committee or the reimbursement will be limited to the SPE rate. Information of reduced fare flights is available on the conference website. Please note that reservations must be made before the SPE published deadline. The departure point will be a city near the university, the student's home, or current place of work, subject to review by the Committee. Alternately, a mileage reimbursement will be made in lieu of airfare should the entrants decide to drive rather than fly to the conference. The reimbursement is based on current allowable mileage rates authorized by the US Internal Revenue Service.

9.2.3. One rental car/van at the gateway city for those teams that fly to the conference.

9.2.4. Lodging related to one hotel room per team member will be reimbursed at a rate not to exceed the SPE rate. Note that the room reservations are limited, so entrants must book their rooms early. Room and taxes for the night before the DSATS symposium, the night of the symposium and for the nights of the conference are covered. Charges for the room on the last day of the

conference need to be pre-approved by the Committee as most conference attendees depart on the last day of the conference unless there are unusual circumstances.

9.2.5. A per diem will be pre-approved by the Committee each year, which will vary with the cost of living in the gateway city. The per diem is intended to cover average meals (breakfast, lunch and dinner) and incidentals.

9.2.6. ATCE registration will be reimbursed. Students should register for the conference at the student rate. Early registration is appreciated.

9.3. Individual award certificates will be presented to all participants upon request, with special certificates given to all finalists.

9.4. DSATS may provide additional awards, at its sole discretion.

9.5. The evaluation and all decisions on any matter in the competition by the DSATS judges and DSATS board are final.

10. Group B Prizes

10.1. The winning team of Group B will submit a SPE Whitepaper that explains their project in detail. If the quality of the abstract is approved by the SPE Conference Program Committee, as per SPE's customary procedures, the paper will be archived in OnePetro

10.2. Individual award certificates will be presented to all participants upon request, with special certificates given to all finalists.

10.3. DSATS may provide additional awards, at its sole discretion.

10.4. The evaluation and all decisions on any matter in the competition by the DSATS judges and DSATS board are final.

11. Terms and conditions

11.1. In no event will SPE, including its directors, officers, employees and agents, as well as DSATS members and officers, and sponsors of the competition, be liable for any damages whatsoever, including without limitation, direct, indirect, special, incidental, consequential, lost profits, or punitive, whether based on contract, tort or any other legal theory, even if SPE or DSATS has been advised of the possibility of such damages.

11.2. Participants and Universities agree to indemnify and hold harmless SPE, its directors, officers, employees and agents, as well as DSATS members and officers, and sponsors of the competition, from all liability, injuries, loss damages, costs or expenses (including attorneys' fees) which are sustained, incurred or required arising out of participation by any parties involved in the competition.

11.3. Participants and Universities agree and acknowledge that participation in the competition is an agreement to all of the rules, regulations, terms and conditions in this document, including revisions and FAQs posted to the DSATS and Drillbotics websites (see section 2.1).

11.4. Winning teams and finalists must agree to the publication of their names, photographs and final paper on the DSATS web site.

11.5. All entries will be distributed to the Drillbotics Committee for the purpose of judging the competition. Design features will not be published until after all teams have been judged and a winner is announced. Previous years' submittals, reports, photos and similar documentation will be publicly available to foster an open exchange of information that will hopefully lead to faster learning for all participants, both new and experienced.

11.6. DSATS and the SPE cannot provide funding to sanctioned individuals and organization per current US law.

11.7. Participants must comply with all local laws applicable to this contest.

12. Marketing

12.1. Upon request, DSATS will provide a link on its website to all participating universities.

12.2. If university policy allows, various industry journals may send a reporter to witness the tests and interview students to publicize the project.

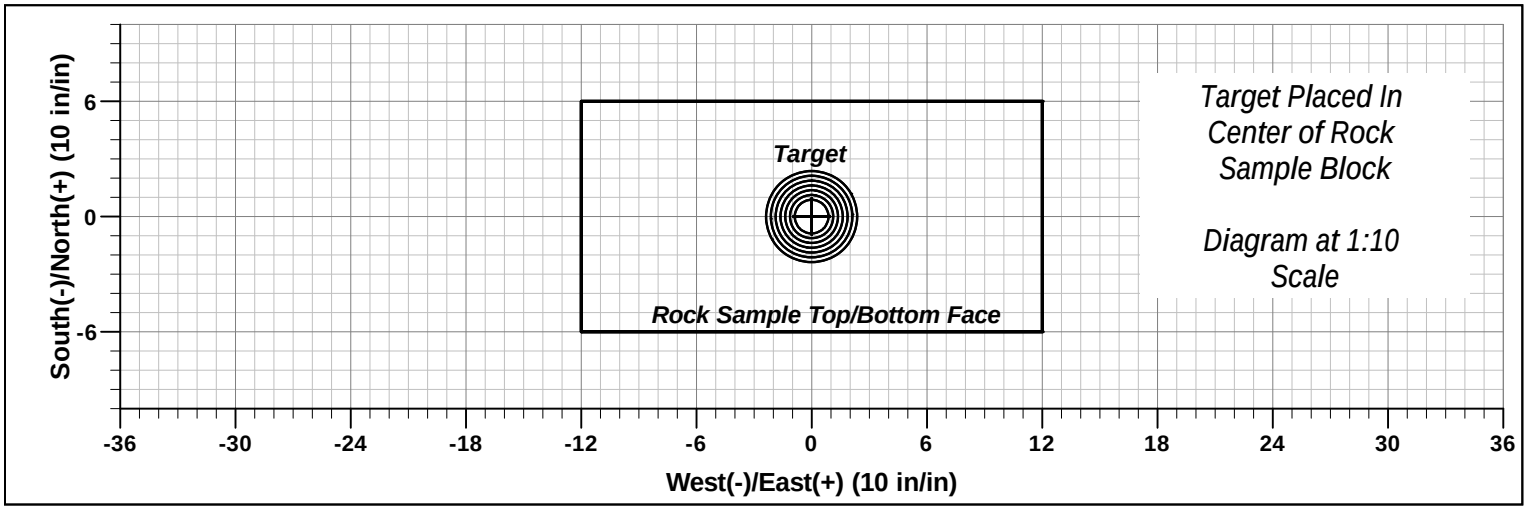
- End -

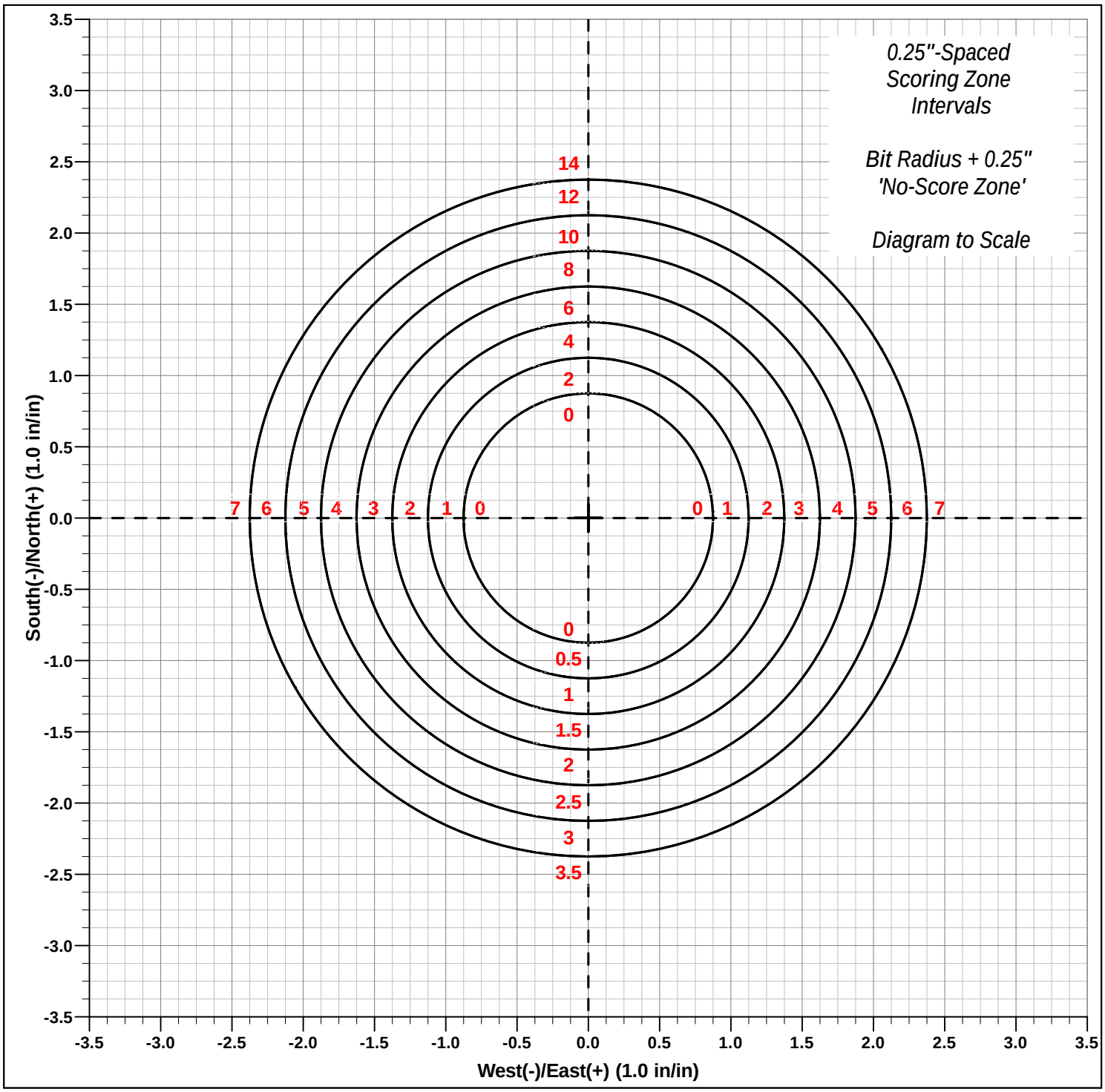
Appendix

A. Directional Objective Scoring

The following attached pages describe the directional target area, as well as the scoring for the directional competition objective. The maximum score is obtained by drilling at least 2 3/8” from well center along the north axis of the rock sample (see page A-A1 for details). Deviation from the north axis results in a reduced score for the same amount of displacement (see page A-A2 for a scoring example).

Directional Control Scoring Target





Directional Control Scoring Target Example

$$Total\ Score = [Min\ Score] + ([Max\ Score] - [Min\ Score]) * \frac{90° - [Deviation\ from\ Max\ Score]}{90°}$$

Max Score = 10 (North axis score)

Min Score = 5 (East axis score)

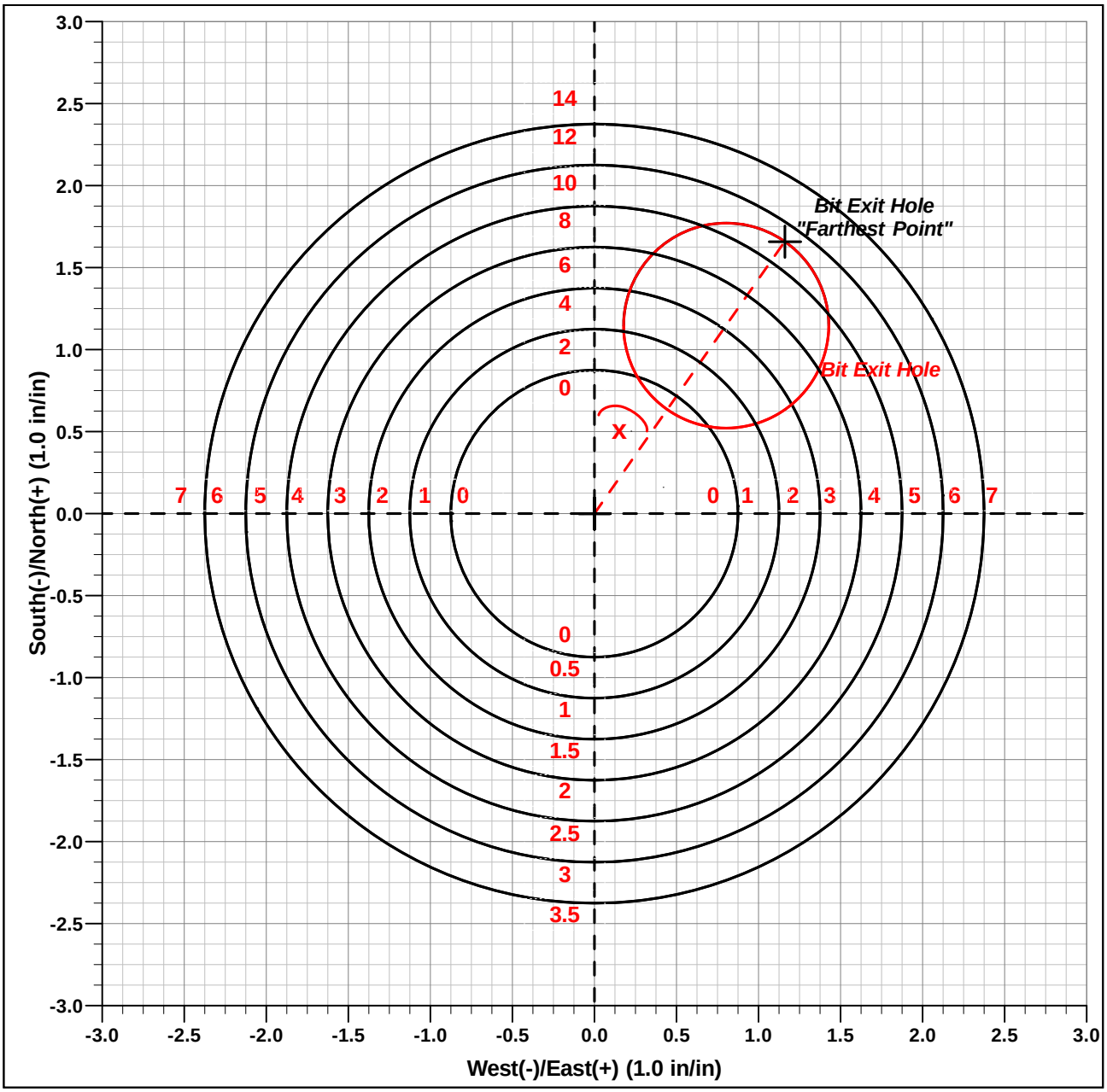
Deviation from Max Score = x = 35°

$$Total\ Score = [5] + ([10] - [5]) * \frac{90° - 35°}{90°}$$

$$Total\ Score = 5 + 5 * \frac{55°}{90°}$$

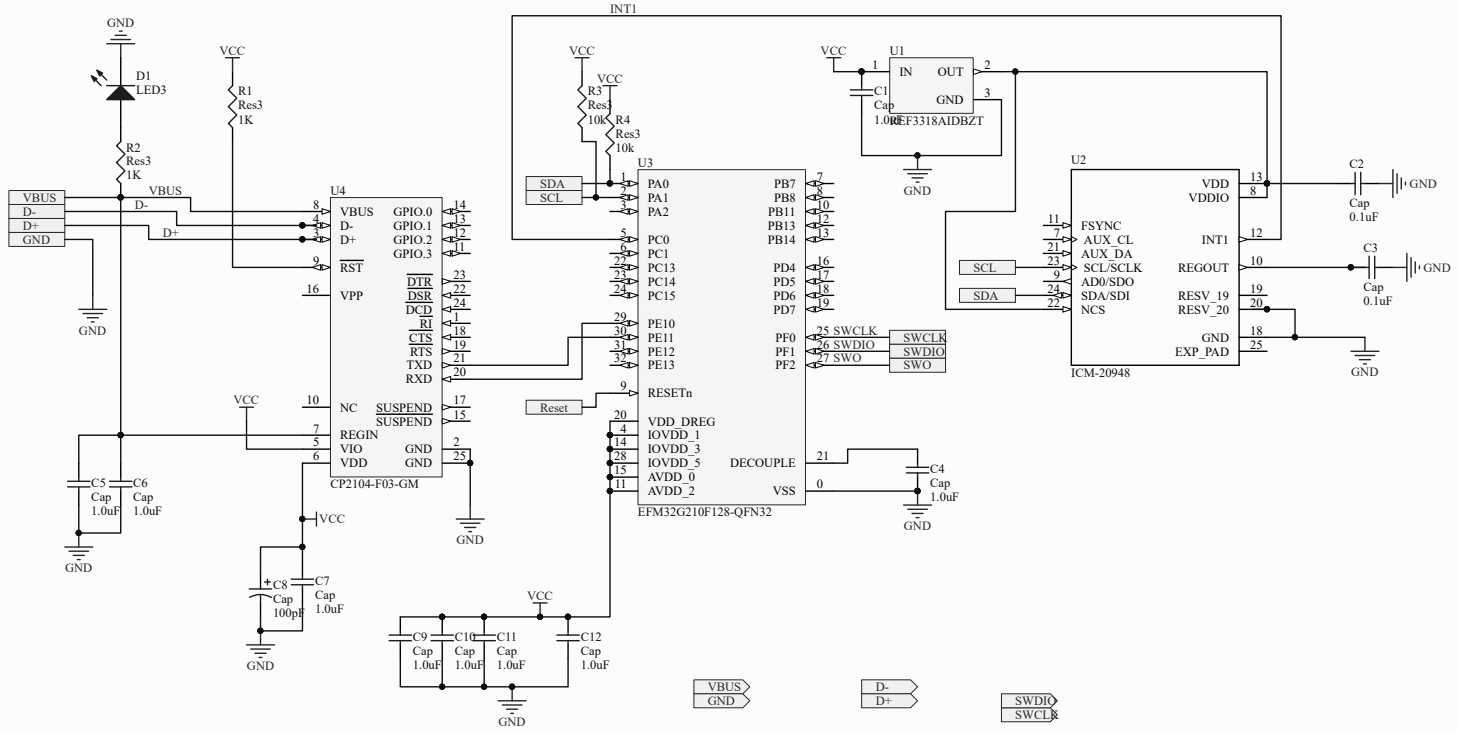
$$Total\ Score = 5 + 5 * \frac{55°}{90°}$$

$$\boldsymbol{Total\ Score = 8.06}$$



B Schematics and Drawings

B.1 PCB Schematics


INT1
GND
D1
LED3
R2
Res3
1K
VBUS
D-
D+
GND
R1
Res3
1K
VCC
U4
VBUS
D-
D+
RST
VPP
GPIO.0
GPIO.1
GPIO.2
GPIO.3
DTR
DSR
DCD
RI
CTS
RTS
TXD
RXD
NC
SUSPEND
SUSPEND
REGIN
VIO
VDD
GND
CP2104-F03-GM
C5
Cap
1.0uF
C6
Cap
1.0uF
C7
Cap
1.0uF
C8
Cap
100pF
R3
Res3
10k
R4
Res3
10k
SDA
SCL
U3
PA0
PA1
PA2
PC0
PC1
PC13
PC14
PC15
PE10
PE11
PE12
PE13
RESETn
VDD_DREG
IOVDD_1
IOVDD_3
IOVDD_5
AVDD_0
AVDD_2
PB7
PB8
PB11
PB13
PB14
PD4
PD5
PD6
PD7
PF0
PF1
PF2
DECOUPLE
VSS
EFM32G210F128-QFN32
Reset
SWCLK
SWDIO
SWO
C4
Cap
1.0uF
C9
Cap
1.0uF
C10
Cap
1.0uF
C11
Cap
1.0uF
C12
Cap
1.0uF
U1
IN
OUT
GND
REF3318AIDBZT
C1
Cap
1.0uF
U2
FSYNC
AUX_CL
AUX_DA
SCL/SCLK
AD0/SDO
SDA/SDI
NCS
VDD
VDDIO
INT1
REGOUT
RESV_19
RESV_20
GND
EXP_PAD
ICM-20948
C2
Cap
0.1uF
C3
Cap
0.1uF
VBUS
GND
D-
D+
SWDIO
SWCLK
SWO
Reset
Title
Size
A4
Number
Revision
Date: 21.06.2019
Sheet of
File: C:\Users\..\Downhole_2019_2.SchDoc
Drawn By:

C Code

C.1 Simplicity Studio C++ Code

```
#include "em_device.h"
#include "em_chip.h"
#include "InitDevice.h"
#include "em_usart.h"
#include "em_i2c.h"
#include "retargetserial.h"
#include "gpiointerrupt.h"
#include "em_cmu.h"
#include "math.h"
#include "stdio.h"
#include "stdarg.h"
#include "register_list.c"

//#include "iostream"
//#include "fstream"


#define DEBUG_BREAK           __asm__("BKPT #0");

#define ICM20948_ADDRESS    0x68 << 1
#define MAG_ADDRESS         0x0C << 1


// Set initial input parameters
enum Ascale {
  AFS_2G = 0,
  AFS_4G,
  AFS_8G,
  AFS_16G
};

enum Gscale {
  GFS_250DPS = 0,
  GFS_500DPS,
  GFS_1000DPS,
  GFS_2000DPS
};

enum Mscale {
    MFS_4900uT = 0
};


// Specify sensor full scale
int Gscale = GFS_250DPS;
int Ascale = AFS_2G;
int Mscale = MFS_4900uT;
float aRes, gRes, mRes;             // scale resolutions per LSB for the sensors

float a_tot;
float g_tot;

float yaw, pitch, roll;
float off_ang;

float roll2, pitch2, yaw2;

uint8_t magTest[6];

// Create vectors to store data
__int16_t accelCount[3];            // Stores the 16-bit signed accelerometer sensor
output
float ax, ay, az;                   // Stores the real accel value in g's
__int16_t gyroCount[3];             // Stores the 16-bit signed gyro sensor output
```

```
float gx, gy, gz;                   // Stores the real gyro value in degrees per
seconds
__int16_t magCount[3];              // Store the 16-bit signed  magnometer sensor
output
float mx, my, mz;                   // Stores the real mag value in uT
float mx_raw, my_raw, mz_raw;

float pp;

int tt;

float up[3], west[3], north[3];
float up_temp[3], west_temp[3], north_temp[3];
float up_norm[3], west_norm[3], north_norm[3];
float up_len, west_len, north_len;
float rot_matrix[3][3];
float q11, q12, q13, q21, q22, q23, q31, q32, q33;

float gyroBias[3], accelBias[3], magBias[3];    // Bias corrections for gyro and
accelerometer

float a_bias_x, a_bias_y, a_bias_z;
float g_bias_x, g_bias_y, g_bias_z;

float magScale[3];
float m_scale_x, m_scale_y, m_scale_z;
float m_bias_x, m_bias_y, m_bias_z;
float m_min_x, m_min_y, m_min_z, m_max_x, m_max_y, m_max_z;

__int16_t tempCount;                            // Stores the internal chip temperature
sensor output
float temperature;                              // Scaled temperature in degrees
Celsius

float SelfTest[6];                              // Gyro and accelerometer self-test
sensor output
__uint32_t count = 0;

float angle_pitch, angle_roll, angle_yaw;       //Store angles
long acc_total_vector;                          // Accelerometer vector
float angle_roll_acc, angle_pitch_acc;


// Array size storing i2c cmd and receive.
#define CMD_ARRAY_SIZE        1
#define DATA_ARRAY_SIZE       18

// Globals for persistent storage
uint8_t cmd_array[CMD_ARRAY_SIZE];
uint8_t data_array[DATA_ARRAY_SIZE];



__int16_t temp;
float temp_act;

__int16_t acc_x;
__int16_t acc_y;
__int16_t acc_z;

__int16_t gyr_x;
__int16_t gyr_y;
__int16_t gyr_z;
```

```
__int16_t mag_x;
__int16_t mag_y;
__int16_t mag_z;

float cal_acc_x;
float cal_acc_y;
float cal_acc_z;

float cal_gyr_x;
float cal_gyr_y;
float cal_gyr_z;

float cal_mag_x;
float cal_mag_y;
float cal_mag_z;

volatile __uint32_t msTicks; /* counts 1ms timeTicks */

double timer;
double dt;

void Delay(__uint32_t dlyTicks);


/*****************************************************************************//**
 * @brief SysTick_Handler
 * Interrupt Service Routine for system tick counter
 *****************************************************************************/
void SysTick_Handler(void)
{
  msTicks++;       /* increment counter necessary in Delay()*/
}
/*****************************************************************************//**
 * @brief Delays number of msTick Systicks (typically 1 ms)
 * @param dlyTicks Number of ticks to Delay
 *****************************************************************************/
void Delay(__uint32_t dlyTicks)
{
  __uint32_t curTicks;

  curTicks = msTicks;
  while ((msTicks - curTicks) < dlyTicks) ;
}

// Write a config register on an I2C device
// Tailored for the ADX345 device only i.e. 1 byte of TX
void i2c_write_register(__uint8_t address, __uint8_t reg_offset, __uint8_t write_data)
{
      cmd_array[0] = reg_offset;
      data_array[0] = write_data;
      i2c_transfer(address, cmd_array, data_array, 1, 1, I2C_FLAG_WRITE_WRITE);
      return;
}

// Read a config register on an I2C device
// Tailored for the ADX345 device only i.e. 1 byte of TX
uint8_t i2c_read_register(uint8_t address, uint8_t reg_offset)
{
      cmd_array[0] = reg_offset;
      i2c_transfer(address, cmd_array, data_array, 1, 1, I2C_FLAG_WRITE_READ);
      return data_array[0];
}

// Reads several registers on ICM20948
```

```
void i2c_read_registers(uint8_t address, uint8_t reg_offset, uint8_t count)
{
      cmd_array[0] = reg_offset;
      i2c_transfer(address, cmd_array, data_array, 1, count, I2C_FLAG_WRITE_READ);
      return;
}


// Read raw data from ICM-20948
void readAccelData(int16_t * destination)
{
    i2c_write_register(ICM20948_ADDRESS, REG_BANK_SEL, 0x00);           // Selects
Reg_BANK 0
    i2c_read_registers(ICM20948_ADDRESS, ACCEL_XOUT_H, 6);              // Read the six
raw data registers into data array
    destination[0] = (int16_t)((data_array[0] << 8) | data_array[1]) ;  // Turn the MSB
and LSB into a signed 16-bit value
    destination[1] = (int16_t)((data_array[2] << 8) | data_array[3]) ;
    destination[2] = (int16_t)((data_array[4] << 8) | data_array[5]) ;
}

void readGyroData(int16_t * destination)
{
    i2c_write_register(ICM20948_ADDRESS, REG_BANK_SEL, 0x00);           // Selects
Reg_BANK 0
    i2c_read_registers(ICM20948_ADDRESS, GYRO_XOUT_H, 6);               // Read the six
raw data registers sequentially into data array
    destination[0] = (int16_t)((data_array[0] << 8) | data_array[1]) ;  // Turn the MSB
and LSB into a signed 16-bit value
    destination[1] = (int16_t)((data_array[2] << 8) | data_array[3]) ;
    destination[2] = (int16_t)((data_array[4] << 8) | data_array[5]) ;
}


void readMagData(int16_t * destination)
{
    i2c_write_register(MAG_ADDRESS, CNTL2, 0x08);                               // Mode
2: continuous measurement 20 Hz
    Delay(10);

    i2c_read_registers(MAG_ADDRESS, 0x11, 6);                                   // Read the
six raw data registers into data array

    destination[0] = (int16_t)((data_array[1] << 8) | data_array[0]) ;      // Turn the
MSB and LSB into a signed 16-bit value
    destination[1] = (int16_t)((data_array[3] << 8) | data_array[2]) ;
    destination[2] = (int16_t)((data_array[5] << 8) | data_array[4]) ;
}

int16_t readTempData()
{
    i2c_write_register(ICM20948_ADDRESS, REG_BANK_SEL, 0x00);           // Selects
Reg_BANK 0
    i2c_read_registers(ICM20948_ADDRESS, TEMP_OUT_H, 2);                // Read the two
raw data registers sequentially into data array
    return ((int16_t)data_array[0]) << 8 | data_array[1] ;              // Turn the MSB
and LSB into a 16-bit value
}

//-------------------------------------------------

void initICM20948()
{
    // Initialize ICM-20948 device
```

```
    // get stable time source
    i2c_write_register(ICM20948_ADDRESS, REG_BANK_SEL, 0x00);           // Selects
Reg_BANK 0
    Delay(100);
    i2c_write_register(ICM20948_ADDRESS, PWR_MGMT_1, 0x01);             // Set clock
source to be PLL with x-axis gyroscope reference, bits 2:0 = 001


    // Set Gyro sample rate = gyroscope output rate/(1 + GYRO_SMPLRT_DIV[7:0])
    i2c_write_register(ICM20948_ADDRESS, REG_BANK_SEL, 0x02);           // Selects
Reg_BANK 2
    Delay(100);
    i2c_write_register(ICM20948_ADDRESS, GYRO_SMPLRT_DIV, 0x04);        // Use a 220 Hz
sample rate
    Delay(100);
    i2c_write_register(ICM20948_ADDRESS, GYRO_CONFIG_1, 0x00);          // Sets gyro
range to +/- 250 deg/s and enable DLPF
    Delay(100);
    i2c_write_register(ICM20948_ADDRESS, GYRO_CONFIG_2, 0x00);          // disable
self-test and averaging measurements
    Delay(100);

    // Set Accel sample rate = accel output rate/(1+ACCEL_SMPLRT_DIV[11:0])
    i2c_write_register(ICM20948_ADDRESS, ACCEL_SMPLRT_DIV_1, 0x00);     // MSB [11:8]
    Delay(100);
    i2c_write_register(ICM20948_ADDRESS, ACCEL_SMPLRT_DIV_2, 0x04);     // LSB [7:0]
Use a 220 Hz sample rate
    Delay(100);
    i2c_write_register(ICM20948_ADDRESS, ACCEL_CONFIG, 0x01);           // Set
accelerometer range to +/- 2 g and enable DLPF
    Delay(100);
    i2c_write_register(ICM20948_ADDRESS, ACCEL_CONFIG_2, 0x00);         // disable
self-test and averaging measurements
    Delay(100);

    // Set Mag settings
    i2c_write_register(ICM20948_ADDRESS, REG_BANK_SEL, 0x00);           // Selects
Reg_BANK 0
    Delay(100);
    i2c_write_register(ICM20948_ADDRESS, INT_PIN_CFG, 0x02);            // INT pin /
Bypass enable
    Delay(100);
    i2c_write_register(ICM20948_ADDRESS, INT_ENABLE, 0x01);             // Enable i2c
master interrupt
    i2c_write_register(ICM20948_ADDRESS, USER_CTRL, 0x00);              // I2C_MST_EN
    Delay(100);

    // Init magnetometer
    i2c_write_register(MAG_ADDRESS, CNTL3, 0x01);                          // Reset
AK09916
    Delay(1000);
    i2c_write_register(MAG_ADDRESS, CNTL2, 0x08);                          // Set mode
2: continuous measurement 20 Hz
    Delay(1000);


}

void calibrateAccGyro()
{

    Delay(1000);
```

```
    uint8_t data[12]; // data array to hold accelerometer and gyro x, y, z, data
    uint16_t ii, packet_count, fifo_count;
    int32_t gyro_bias[3] = {0, 0, 0}, accel_bias[3] = {0, 0, 0};

    // reset device
    i2c_write_register(ICM20948_ADDRESS, REG_BANK_SEL, 0x00);           // Selects
Reg_BANK 0
    i2c_write_register(ICM20948_ADDRESS, PWR_MGMT_1, 0x80); // Write a one to bit 7
reset bit; toggle reset device
    Delay(1000);

    // get stable time source
    // Set clock source to be PLL with x-axis gyroscope reference, bits 2:0 = 001
    i2c_write_register(ICM20948_ADDRESS, PWR_MGMT_1, 0x01);
    i2c_write_register(ICM20948_ADDRESS, PWR_MGMT_2, 0x00);
    Delay(200);

    // Configure device for bias calculation
    i2c_write_register(ICM20948_ADDRESS, INT_ENABLE, 0x00);   // Disable all interrupts
    i2c_write_register(ICM20948_ADDRESS, FIFO_EN_1, 0x00);      // Disable FIFO
    i2c_write_register(ICM20948_ADDRESS, FIFO_EN_2, 0x00);      // Disable FIFO
    i2c_write_register(ICM20948_ADDRESS, PWR_MGMT_1, 0x00);   // Turn on internal clock
source
    i2c_write_register(ICM20948_ADDRESS, I2C_MST_CTRL, 0x00); // Disable I2C master
    i2c_write_register(ICM20948_ADDRESS, USER_CTRL, 0x00);    // Disable FIFO and I2C
master modes
    i2c_write_register(ICM20948_ADDRESS, USER_CTRL, 0x0A);    // Reset FIFO and DMP
    Delay(15);

    // Configure ICM20948 gyro and accelerometer for bias calculation
    i2c_write_register(ICM20948_ADDRESS, REG_BANK_SEL, 0x02);           // Selects
Reg_BANK 2
    i2c_write_register(ICM20948_ADDRESS, GYRO_SMPLRT_DIV, 0x00);    // Set sample rate
to 1,1 kHz
    i2c_write_register(ICM20948_ADDRESS, GYRO_CONFIG_1, 0x21);          // Set low-pass
filter to 188 Hz, Set gyro full-scale to 250 deg/s, maximum sensitivity

    i2c_write_register(ICM20948_ADDRESS, ACCEL_SMPLRT_DIV_1, 0x00);
    i2c_write_register(ICM20948_ADDRESS, ACCEL_SMPLRT_DIV_2, 0x00); // Set sample rate
1,125 kHz
    i2c_write_register(ICM20948_ADDRESS, ACCEL_CONFIG, 0x21); // Set accelerometer
full-scale to 2 g, maximum sensitivity

    uint16_t  gyrosensitivity  = 131;   // = 131 LSB/degrees/sec
    uint16_t  accelsensitivity = 16384;  // = 16384 LSB/g

    // Configure FIFO to capture accelerometer and gyro data for bias calculation
    i2c_write_register(ICM20948_ADDRESS, REG_BANK_SEL, 0x00);           // Selects
Reg_BANK 0
    i2c_write_register(ICM20948_ADDRESS, USER_CTRL, 0x40);      // Enable FIFO
    i2c_write_register(ICM20948_ADDRESS, FIFO_EN_2, 0x1E);     // Enable gyro and
accelerometer sensors for FIFO  (max size 1024 bytes in MPU-6050)
    Delay(80); // accumulate 80 samples in 80 milliseconds = 960 bytes

    // At end of sample accumulation, turn off FIFO sensor read
    i2c_write_register(ICM20948_ADDRESS, FIFO_EN_2, 0x00);        // Disable gyro and
accelerometer sensors for FIFO
    i2c_read_registers(ICM20948_ADDRESS, FIFO_COUNTH, 2); // read FIFO sample count
    fifo_count = ((uint16_t)data_array[0] << 8) | data_array[1];
    packet_count = fifo_count/12;// How many sets of full gyro and accelerometer data
for averaging

    for (ii = 0; ii < packet_count; ii++)
    {
```

```
        int16_t accel_temp[3] = {0, 0, 0}, gyro_temp[3] = {0, 0, 0};
        i2c_read_registers(ICM20948_ADDRESS, FIFO_R_W, 12); // read data for averaging
        accel_temp[0] = (int16_t) (((int16_t)data_array[0] << 8) | data_array[1]  ) ;
// Form signed 16-bit integer for each sample in FIFO
        accel_temp[1] = (int16_t) (((int16_t)data_array[2] << 8) | data_array[3]  ) ;
        accel_temp[2] = (int16_t) (((int16_t)data_array[4] << 8) | data_array[5]  ) ;
        gyro_temp[0]  = (int16_t) (((int16_t)data_array[6] << 8) | data_array[7]  ) ;
        gyro_temp[1]  = (int16_t) (((int16_t)data_array[8] << 8) | data_array[9]  ) ;
        gyro_temp[2]  = (int16_t) (((int16_t)data_array[10] << 8) | data_array[11]) ;

        accel_bias[0] += (int32_t) accel_temp[0]; // Sum individual signed 16-bit
biases to get accumulated signed 32-bit biases
        accel_bias[1] += (int32_t) accel_temp[1];
        accel_bias[2] += (int32_t) accel_temp[2];
        gyro_bias[0]  += (int32_t) gyro_temp[0];
        gyro_bias[1]  += (int32_t) gyro_temp[1];
        gyro_bias[2]  += (int32_t) gyro_temp[2];

    }
    accel_bias[0] /= (int32_t) packet_count; // Normalize sums to get average count
biases
    accel_bias[1] /= (int32_t) packet_count;
    accel_bias[2] /= (int32_t) packet_count;
    gyro_bias[0]  /= (int32_t) packet_count;
    gyro_bias[1]  /= (int32_t) packet_count;
    gyro_bias[2]  /= (int32_t) packet_count;

    if(accel_bias[2] > 0L) {accel_bias[2] -= (int32_t) accelsensitivity;}  // Remove
gravity from the z-axis accelerometer bias calculation
    else {accel_bias[2] += (int32_t) accelsensitivity;}

    // Construct the gyro biases for push to the hardware gyro bias registers, which
are reset to zero upon device startup
    data[0] = (-gyro_bias[0]/4  >> 8) & 0xFF; // Divide by 4 to get 32.9 LSB per deg/s
to conform to expected bias input format
    data[1] = (-gyro_bias[0]/4)       & 0xFF; // Biases are additive, so change sign on
calculated average gyro biases
    data[2] = (-gyro_bias[1]/4  >> 8) & 0xFF;
    data[3] = (-gyro_bias[1]/4)       & 0xFF;
    data[4] = (-gyro_bias[2]/4  >> 8) & 0xFF;
    data[5] = (-gyro_bias[2]/4)       & 0xFF;


    gyroBias[0] = (float) gyro_bias[0]/(float) gyrosensitivity; // construct gyro bias
in deg/s for later manual subtraction
    gyroBias[1] = (float) gyro_bias[1]/(float) gyrosensitivity;
    gyroBias[2] = (float) gyro_bias[2]/(float) gyrosensitivity;

    // Construct the accelerometer biases for push to the hardware accelerometer bias
registers. These registers contain
    // factory trim values which must be added to the calculated accelerometer biases;
on boot up these registers will hold
    // non-zero values. In addition, bit 0 of the lower byte must be preserved since it
is used for temperature
    // compensation calculations. Accelerometer bias registers expect bias input as
2048 LSB per g, so that
    // the accelerometer biases calculated above must be divided by 8.

    int32_t accel_bias_reg[3] = {0, 0, 0}; // A place to hold the factory accelerometer
trim biases
    i2c_write_register(ICM20948_ADDRESS, REG_BANK_SEL, 0x01);           // Selects
Reg_BANK 1
    i2c_read_registers(ICM20948_ADDRESS, XA_OFFS_H, 2); // Read factory accelerometer
trim values
```

```
    accel_bias_reg[0] = (int16_t) ((int16_t)data_array[0] << 8) | data_array[1];
    i2c_read_registers(ICM20948_ADDRESS, YA_OFFS_H, 2);
    accel_bias_reg[1] = (int16_t) ((int16_t)data_array[0] << 8) | data_array[1];
    i2c_read_registers(ICM20948_ADDRESS, ZA_OFFS_H, 2);
    accel_bias_reg[2] = (int16_t) ((int16_t)data_array[0] << 8) | data_array[1];

    uint32_t mask = 1uL; // Define mask for temperature compensation bit 0 of lower
byte of accelerometer bias registers
    uint8_t mask_bit[3] = {0, 0, 0}; // Define array to hold mask bit for each
accelerometer bias axis

    for(ii = 0; ii < 3; ii++)
    {
        if(accel_bias_reg[ii] & mask) mask_bit[ii] = 0x01; // If temperature
compensation bit is set, record that fact in mask_bit
    }

    // Construct total accelerometer bias, including calculated average accelerometer
bias from above
    accel_bias_reg[0] -= (accel_bias[0]/8); // Subtract calculated averaged
accelerometer bias scaled to 2048 LSB/g (16 g full scale)
    accel_bias_reg[1] -= (accel_bias[1]/8);
    accel_bias_reg[2] -= (accel_bias[2]/8);

    data[0] = (accel_bias_reg[0] >> 8) & 0xFF;
    data[1] = (accel_bias_reg[0])      & 0xFF;
    data[1] = data[1] | mask_bit[0]; // preserve temperature compensation bit when
writing back to accelerometer bias registers
    data[2] = (accel_bias_reg[1] >> 8) & 0xFF;
    data[3] = (accel_bias_reg[1])      & 0xFF;
    data[3] = data[3] | mask_bit[1]; // preserve temperature compensation bit when
writing back to accelerometer bias registers
    data[4] = (accel_bias_reg[2] >> 8) & 0xFF;
    data[5] = (accel_bias_reg[2])      & 0xFF;
    data[5] = data[5] | mask_bit[2]; // preserve temperature compensation bit when
writing back to accelerometer bias registers

    // Push accelerometer biases to hardware registers; doesn't work well for
accelerometer
    // Are we handling the temperature compensation bit correctly?
    //  i2c_write_register(ICM20948_ADDRESS, XA_OFFSET_H, data[0]);
    //  i2c_write_register(ICM20948_ADDRESS, XA_OFFSET_L_TC, data[1]);
    //  i2c_write_register(ICM20948_ADDRESS, YA_OFFSET_H, data[2]);
    //  i2c_write_register(ICM20948_ADDRESS, YA_OFFSET_L_TC, data[3]);
    //  i2c_write_register(ICM20948_ADDRESS, ZA_OFFSET_H, data[4]);
    //  i2c_write_register(ICM20948_ADDRESS, ZA_OFFSET_L_TC, data[5]);

    // Output scaled accelerometer biases for manual subtraction in the main program
    accelBias[0] = (float)accel_bias[0]/(float)accelsensitivity;
    accelBias[1] = (float)accel_bias[1]/(float)accelsensitivity;
    accelBias[2] = (float)accel_bias[2]/(float)accelsensitivity;

    g_bias_x = gyroBias[0];
    g_bias_y = gyroBias[1];
    g_bias_z = gyroBias[2];

    a_bias_x = accelBias[0];
    a_bias_y = accelBias[1];
    a_bias_z = accelBias[2];
}

void calibrateMagnetometer()
{
    uint16_t ii = 0, jj = 0, samplecount = 2000;
```

```
    int16_t magBiases[3] = {0, 0, 0}, magScales[3] = {0, 0, 0};
    int16_t magMax[3] = {-32752, -32752, -32752}, magMin[3] = {32752, 32752, 32752},
magTemp[3] = {0, 0, 0};

    float magScaleFactor = 0.15;        // uT/LSB

    print("Mag calibration: Wave device in figure eight for 30 sec. Starting in 2
sec:");
    print("\n");
    Delay(2000);
    print("Start moving device");
    print("\n");



    // Shoot for 15 seconds to find max/min values
    for(ii = 0; ii < samplecount; ii++)
    {
        readMagData(magTemp);
        pp = (float)(ii)/2000*100;
        print("Calibration status: "); print("%.f",pp); print(" percent"); print("\r");
        for(jj = 0; jj < 3; jj++)
        {
            if(magTemp[jj] > magMax[jj]) magMax[jj] = magTemp[jj];
            if(magTemp[jj] < magMin[jj]) magMin[jj] = magTemp[jj];
        }
        Delay(12);      // new data available evry 10 ms
    }

    // Get hard iron correction
    magBiases[0]  = (magMax[0] + magMin[0])/2;  // get average x mag bias in counts
    magBiases[1]  = (magMax[1] + magMin[1])/2;  // get average y mag bias in counts
    magBiases[2]  = (magMax[2] + magMin[2])/2;  // get average z mag bias in counts

    magBias[0] = (float)magBiases[0]*magScaleFactor;        // mag bias in uT for later
subtraction
    magBias[1] = (float)magBiases[1]*magScaleFactor;
    magBias[2] = (float)magBiases[2]*magScaleFactor;

    // Get soft iron correction
    magScales[0]  = (magMax[0] - magMin[0])/2;  // get average x axis max chord length
in counts
    magScales[1]  = (magMax[1] - magMin[1])/2;  // get average y axis max chord length
in counts
    magScales[2]  = (magMax[2] - magMin[2])/2;  // get average z axis max chord length
in counts

    float avg_rad = (float)(magScales[0] + magScales[1] +
magScales[2])*magScaleFactor/3;

    magScale[0] = (float)avg_rad/(magScales[0]*magScaleFactor);  // get soft iron scale
x axis
    magScale[1] = (float)avg_rad/(magScales[1]*magScaleFactor);  // get soft iron scale
y axis
    magScale[2] = (float)avg_rad/(magScales[2]*magScaleFactor);  // get soft iron scale
z axis

    Delay(100);

    // Store bias and scale globally


    print("\n");
```

```
    print("Bias_x %f", magBias[0]);
    print(" Bias_y %f", magBias[1]);
    print(" Bias_z %f", magBias[2]);
    print("\n");

    print("Scale_x %f", magScale[0]);
    print(" Scale_y %f", magScale[1]);
    print(" Scale_z %f", magScale[2]);
    print("\n");

    Delay(1000);

    m_bias_x = magBias[0];
    m_bias_y = magBias[1];
    m_bias_z = magBias[2];
    m_scale_x = magScale[0];
    m_scale_y = magScale[1];
    m_scale_z = magScale[2];
    m_min_x = magMin[0]*magScaleFactor;
    m_min_y = magMin[1]*magScaleFactor;
    m_min_z = magMin[2]*magScaleFactor;
    m_max_x = magMax[0]*magScaleFactor;
    m_max_y = magMax[1]*magScaleFactor;
    m_max_z = magMax[2]*magScaleFactor;

    print("m_min_x %f", m_min_x); print("m_max_x %f", m_max_x); print("\n");
    print("m_min_y %f", m_min_y); print("m_max_y %f", m_max_y); print("\n");
    print("m_min_z %f", m_min_z); print("m_max_z %f", m_max_z); print("\n");
    print("n");

}


// Used by the read_register and write_register functions
// data_array is read data for WRITE_READ and tx2 data for WRITE_WRITE
void i2c_transfer(uint16_t device_addr, uint8_t cmd_array[], uint8_t data_array[],
uint16_t cmd_len, uint16_t data_len, uint8_t flag)
{
      // Transfer structure
      I2C_TransferSeq_TypeDef i2cTransfer;

      // Initialize I2C transfer
      I2C_TransferReturn_TypeDef result;
      i2cTransfer.addr          = device_addr;
      i2cTransfer.flags         = flag;
      i2cTransfer.buf[0].data   = cmd_array;
      i2cTransfer.buf[0].len    = cmd_len;

      // Note that WRITE_WRITE this is tx2 data
      i2cTransfer.buf[1].data   = data_array;
      i2cTransfer.buf[1].len    = data_len;

      // Set up the transfer
      result = I2C_TransferInit(I2C0, &i2cTransfer);

      // Do it until the transfer is done
      while (result != i2cTransferDone)
      {
            if (result != i2cTransferInProgress)
            {
                DEBUG_BREAK;
            }
            result = I2C_Transfer(I2C0);
      }
```

```
}

void icm_Power_on(void)
{

}


// Setup routine for MCU and MPU
void Setup()
{
    CHIP_Init();

    enter_DefaultMode_from_RESET();

    //RETARGET_SerialInit();

    gpioSetup();

    if (SysTick_Config(CMU_ClockFreqGet(cmuClock_CORE) / 1000))
        {
        DEBUG_BREAK;
        }

    // Code to interface I2C device goes here...

    Delay(1000);

    I2C_Init_TypeDef i2cInit = I2C_INIT_DEFAULT;
    I2C_Init(I2C0, &i2cInit);
    i2c_write_register(ICM20948_ADDRESS, REG_BANK_SEL, 0x00); // // Selects Reg_BANK 0
    Delay(1000);
    uint8_t c = i2c_read_register(ICM20948_ADDRESS, WHO_AM_I);  // Read WHO_AM_I
register for ICM-20948
    if (("i", c) == 234)
    {
        print("\n");
        print("I AM ICM-20948");
        //print("%i", c);
        print("\n");
    } else
    {
        print("Failed to connect to ICM-20948");
    }

    Delay(1000);

      //MPU6050SelfTest();
      //if(SelfTest[0] < 1.0f && SelfTest[1] < 1.0f && SelfTest[2] < 1.0f &&
SelfTest[3] < 1.0f && SelfTest[4] < 1.0f && SelfTest[5] < 1.0f) {
        //  printf("Pass Selftest!");
      //}


//    calibrateAccGyro();
//    Delay(1000);

      initICM20948();
//    Delay(1000);
//
//    calibrateMagnetometer();
//    Delay(500);

    return;
```

```
}

void getGres() {
  switch (Gscale)
  {
    // Possible gyro scales (and their register bit settings) are:
    // 250 DPS (00), 500 DPS (01), 1000 DPS (10), and 2000 DPS  (11).
    case GFS_250DPS:
          gRes = 250.0/32768.0;
          break;
    case GFS_500DPS:
          gRes = 500.0/32768.0;
          break;
    case GFS_1000DPS:
          gRes = 1000.0/32768.0;
          break;
    case GFS_2000DPS:
          gRes = 2000.0/32768.0;
          break;
  }
}

void getAres() {
  switch (Ascale)
  {
    // Possible accelerometer scales (and their register bit settings) are:
    // 2 Gs (00), 4 Gs (01), 8 Gs (10), and 16 Gs  (11).
    case AFS_2G:
          aRes = 2.0/32768.0;
          break;
    case AFS_4G:
          aRes = 4.0/32768.0;
          break;
    case AFS_8G:
          aRes = 8.0/32768.0;
          break;
    case AFS_16G:
          aRes = 16.0/32768.0;
          break;
  }
}

void gpioSetup(void)
{
  /* Enable GPIO in CMU */
  CMU_ClockEnable(cmuClock_GPIO, true);

  /* Initialize GPIO interrupt dispatcher */
  GPIOINT_Init();

  /* Configure PB10 and PB11 as input */
  GPIO_PinModeSet(gpioPortE, 0, gpioModeInput, 0);

  /* Set falling edge interrupt for both ports */
  GPIO_IntConfig(gpioPortE, 0, true, true, true);
  }

void send_string(char * string)
{
      while (*string != 0)
      {
            if (*string == '\n')
            {
                  USART_Tx(USART0, '\r');
```

```
            }
            USART_Tx(USART0, *string++);
        }
}

void print(const char* format, ...)
{
    char        msg[130];
    va_list     args;

    va_start(args, format);
    vsnprintf(msg, sizeof(msg), format, args); // do check return value
    va_end(args);

    send_string(msg);
}

/**************************************************************************//**
 * @brief  Main function
 *****************************************************************************/
int main(void)
{
    Setup();

    while(1)
    {

    dt = (msTicks - timer) / 1000;
    timer = msTicks;

    readAccelData(accelCount);  // Read the x/y/z adc values
    getAres();

    // Now we'll calculate the accleration value into actual g's
//    ax = (float)accelCount[0]*aRes - accelBias[0];  // get actual g value, this
depends on scale being set
//    ay = (float)accelCount[1]*aRes - accelBias[1];
//    az = (float)accelCount[2]*aRes - accelBias[2];
    ax = (float)accelCount[0]*aRes;  // get actual g value, this depends on scale being
set
    ay = (float)accelCount[1]*aRes;
    az = (float)accelCount[2]*aRes;
//    a_tot = (float)sqrt(ax*ax+ay*ay+az*az);

    readGyroData(gyroCount);  // Read the x/y/z adc values
    getGres();

    // Calculate the gyro value into actual degrees per second
//    gx = (float)gyroCount[0]*250.0/32768.0 - gyroBias[0];  // get actual gyro value
in dpm, this depends on scale being set
//    gy = (float)gyroCount[1]*250.0/32768.0 - gyroBias[1];
//    gz = (float)gyroCount[2]*250.0/32768.0 - gyroBias[2];
    gx = (float)gyroCount[0]*250.0/32768.0;  // get actual gyro value in dpm, this
depends on scale being set
    gy = (float)gyroCount[1]*250.0/32768.0;
    gz = (float)gyroCount[2]*250.0/32768.0;

//    tempCount = readTempData();  // Read the x/y/z adc values
//    temperature = ((float) tempCount) / 340. + 36.53; // Temperature in degrees
Centigrade

//    angle_pitch += gx * 0.0000611;
//    angle_roll += gy * 0.0000611;
//    angle_yaw += gz * 0.0000611;
```

```
    readMagData(magCount);
//    mx_raw = (float)magCount[0]*4912.0/32752.0;
//    my_raw = (float)magCount[1]*4912.0/32752.0;
//    mz_raw = (float)magCount[2]*4912.0/32752.0;
//    mx = (float)(magCount[0]*4912.0/32752.0 - magBias[0])*magScale[0];
//    my = (float)(magCount[1]*4912.0/32752.0 - magBias[1])*magScale[1];
//    mz = (float)(magCount[2]*4912.0/32752.0 - magBias[2])*magScale[2];
    mx = (float)(magCount[0]*4912.0/32752.0);
    my = (float)(magCount[1]*4912.0/32752.0);
    mz = (float)(magCount[2]*4912.0/32752.0);



//  Calculates the cross product between accel and magnet vector:
//    up_temp[0] = az, up_temp[1] = ax*(-1), up_temp[2]= ay*(-1);
//    north_temp[0] = mz*(-1), north_temp[1] = mx*(-1), north_temp[2] = my;
//
//    up[0] = up_temp[0], up[1] = up_temp[1], up[2] = up_temp[2];
//    west[0] = (float)up[1]*north_temp[2]-up[2]*north_temp[1];
//    west[1] = (float)up[2]*north_temp[0]-up[0]*north_temp[2];
//    west[2] = (float)up[0]*north_temp[1]-up[1]*north_temp[0];
//
//    north[0] = (float)west[1]*up[2]-west[2]*up[1];
//    north[1] = (float)west[2]*up[0]-west[0]*up[2];
//  north[2] = (float)west[0]*up[1]-west[1]*up[0];
//
////    Calculates vector lengths
//    up_len = (float)sqrt(up[0]*up[0] + up[1]*up[1] + up[2]*up[2]);
//    west_len = (float)sqrt(west[0]*west[0] + west[1]*west[1] + west[2]*west[2]);
//    north_len = (float)sqrt(north[0]*north[0] + north[1]*north[1] +
north[2]*north[2]);
//
////    Normalizes vectors
//    up_norm[0] = (float)up[0]/up_len; up_norm[1] = (float)up[1]/up_len; up_norm[2] =
(float)up[2]/up_len;
//    west_norm[0] = (float)west[0]/west_len; west_norm[1] = (float)west[1]/west_len;
west_norm[2] = (float)west[2]/west_len;
//    north_norm[0] = (float)north[0]/north_len; north_norm[1] =
(float)north[1]/north_len; north_norm[2] = (float)north[2]/north_len;
//
////    Defines a rotation matrix that can tell the orientation of the card relative to
the inertial frame
//    rot_matrix[0][0] = (float)north_norm[0]; rot_matrix[1][0] = north_norm[1];
rot_matrix[2][0] = north_norm[2];
//    rot_matrix[0][1] = (float)west_norm[0]; rot_matrix[1][1] = west_norm[1];
rot_matrix[2][1] = west_norm[2];
//    rot_matrix[0][2] = (float)up_norm[0]; rot_matrix[1][2] = up_norm[1];
rot_matrix[2][2] = up_norm[2];
//
////    Calculates euler angles; roll(z), pitch(y) and yaw/x)
//    roll = (float)atan2(rot_matrix[1][0],rot_matrix[0][0])*180/3.14159;
//    pitch = (float)atan2((-1)*rot_matrix[2][0],sqrt(rot_matrix[2][1]*rot_matrix[2]
[1]+rot_matrix[2][2]*rot_matrix[2][2]))*180/3.14159;
//    yaw = (float)atan2(rot_matrix[2][1],rot_matrix[2][2])*180/3.14159;
//
////    Name all components in the rotation matrix for further calculations in Labview
//    q11 = (float)rot_matrix[0][0]; q12 = (float)rot_matrix[0][1]; q13 =
(float)rot_matrix[0][2];
//    q21 = (float)rot_matrix[1][0]; q22 = (float)rot_matrix[1][1]; q23 =
(float)rot_matrix[1][2];
//    q31 = (float)rot_matrix[2][0]; q32 = (float)rot_matrix[2][1]; q33 =
(float)rot_matrix[2][2];
```

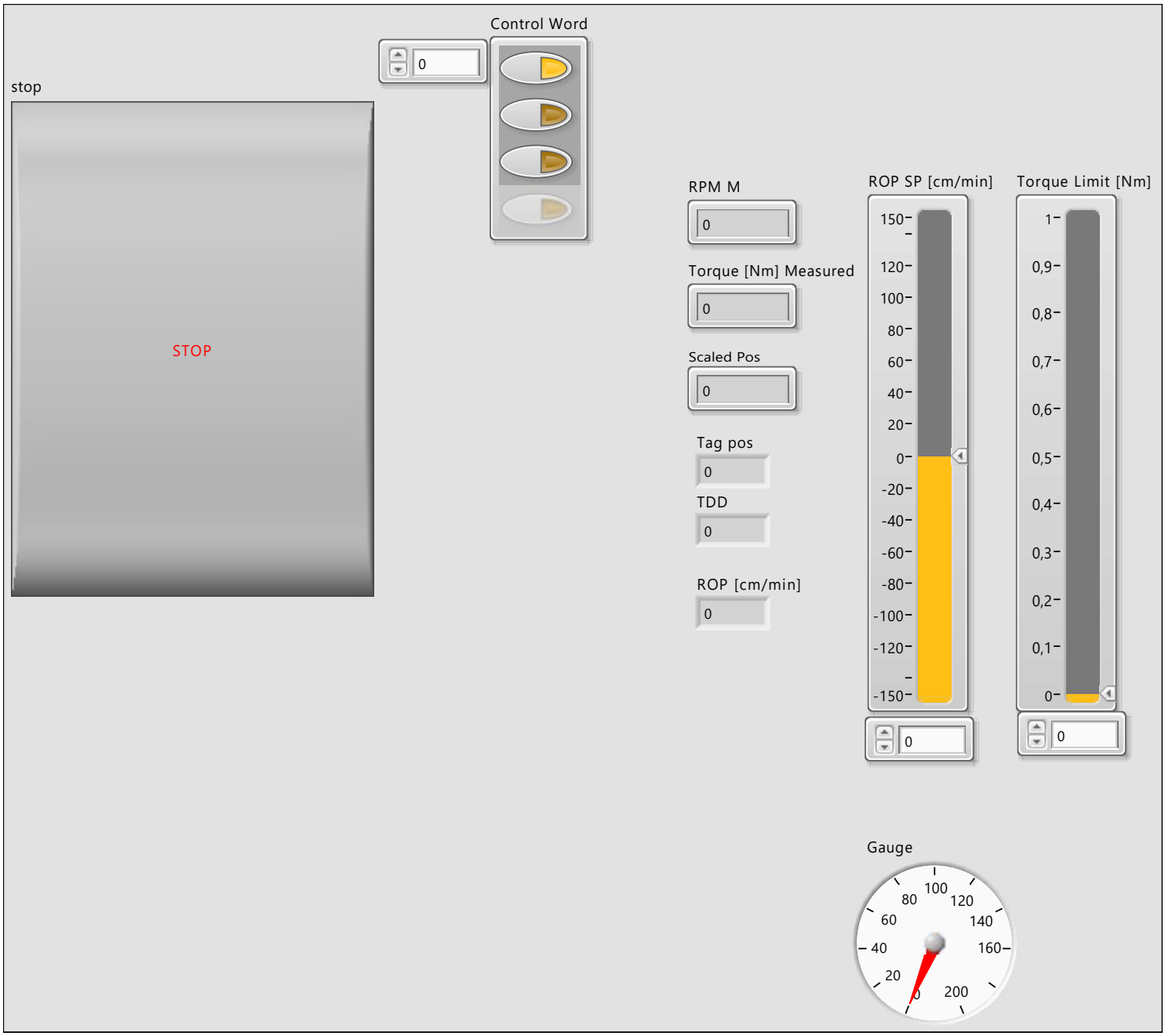
```
//  Serial prints that Labview can parse and analyze
//    print(" Azimuth: %f", roll); print(" d ");
//    print(" Pitch: %f", pitch); print(" d ");
//    print(" Yaw: %f", yaw); print(" d ");
//    print(" q11: %f", q11); print(" d ");
//    print(" q12: %f", q12); print(" d ");
//    print(" q13: %f", q13); print(" d ");
//    print(" q21: %f", q21); print(" d ");
//    print(" q22: %f", q22); print(" d ");
//    print(" q23: %f", q23); print(" d ");
//    print(" q31: %f", q31); print(" d ");
//    print(" q32: %f", q32); print(" d ");
//    print(" q33: %f", q33); print(" d ");
    print(" mx: %f", mx); print(" d ");
    print(" my: %f", my); print(" d ");
    print(" mz: %f", mz); print(" d ");
//    print(" mx_raw: %f", mx_raw); print(" d ");
//    print(" my_raw: %f", my_raw); print(" d ");
//    print(" mz_raw: %f", mz_raw); print(" d ");
    print(" gx: %f", gx); print(" d ");
    print(" gy: %f", gy); print(" d ");
    print(" gz: %f", gz); print(" d ");
    print(" ax: %f", ax); print(" d ");
    print(" ay: %f", ay); print(" d ");
    print(" az: %f", az); print(" d ");
    print("\n");
    print("%c", 0x04);

    }

}
```

C.2 Labview Code

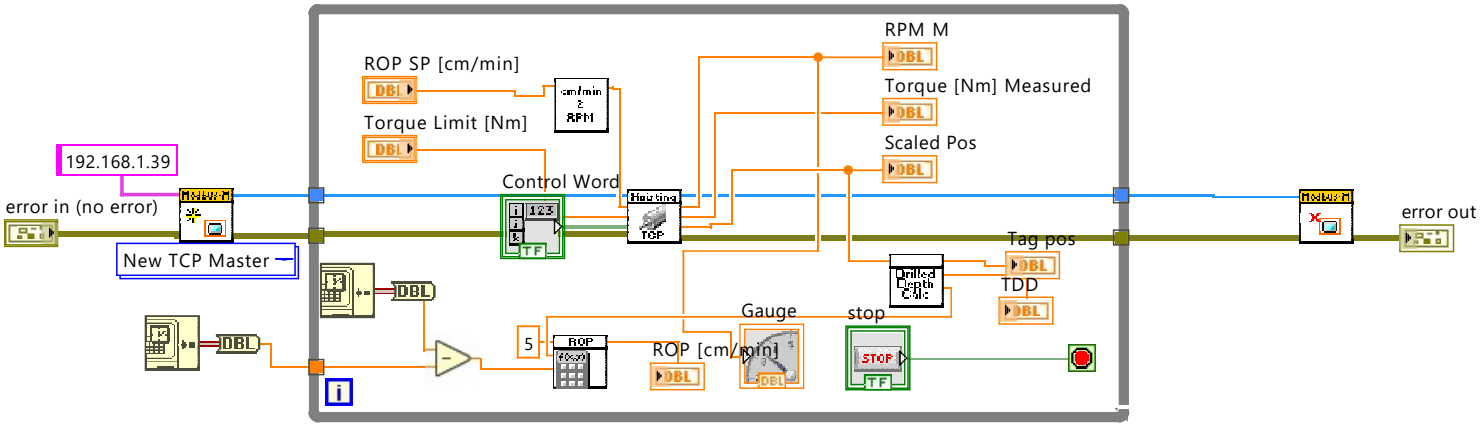
C.2.1 Hoisting Motor Manual

HM_Manual.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Hoisting Motor\HM_Manual.vi
Last modified on 30.05.2019 at 09.01
Printed on 18.06.2019 at 10.50

HM_Manual.vi



HM_Manual.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Hoisting Motor\HM_Manual.vi
Last modified on 30.05.2019 at 09.01
Printed on 18.06.2019 at 10.50



Modbus Master.lvclass:Close.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Modbus\master\Close.vi

Modbus Master.lvclass:Create Master Instance (TCP).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Modbus\master\Create Master Instance (TCP).vi

Modbus Master.lvclass:Create Master Instance.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Modbus\master\Create Master Instance.vi

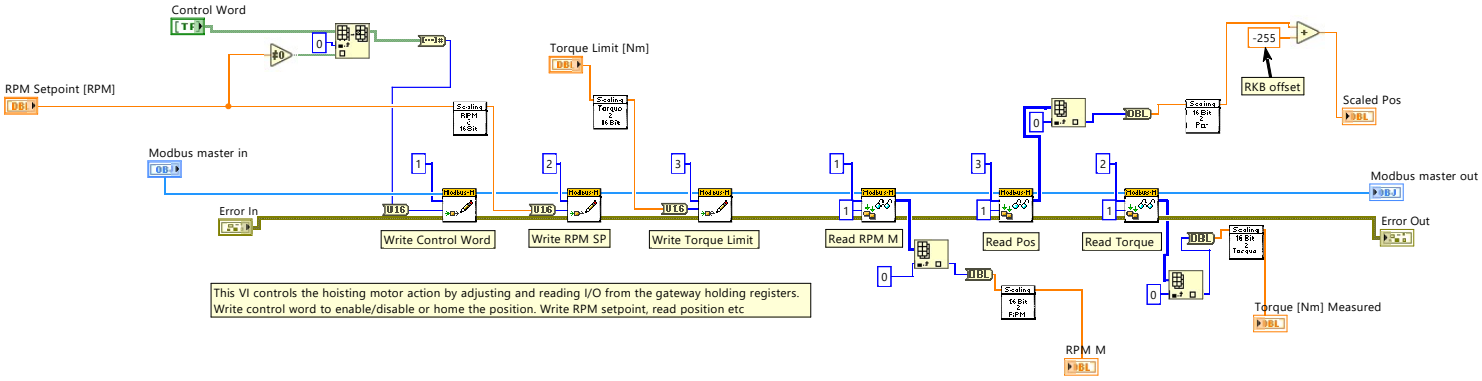
ROP_from_pos_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\ROP_from_pos_2019.vi

drilled_depth_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\drilled_depth_sub.vi

HM_TCP_SubVI.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Hoisting Motor\SubVI's\HM_TCP_SubVI.vi

cm_per_min_2_hoisting_RPM.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\LabVIEW Drillbotics 2018\Conversions\cm_per_min_2_hoisting_RPM.vi

HM_TCP_SubVI.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Hoisting Motor\SubVI's\HM_TCP_SubVI.vi
Last modified on 25.05.2019 at 16.20
Printed on 18.06.2019 at 10.52



Modbus Master.lvclass:Read Input Registers.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Modbus\master\Functions\Read Input Registers.vi

Modbus Master.lvclass:Write Single Holding Register.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Modbus\master\Functions\Write Single Holding Register.vi

scaling16bit2Torque_TCP_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Convertion subVIs\scaling16bit2Torque_TCP_2019.vi

scaling16bit2pos_TCP_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Convertion subVIs\scaling16bit2pos_TCP_2019.vi

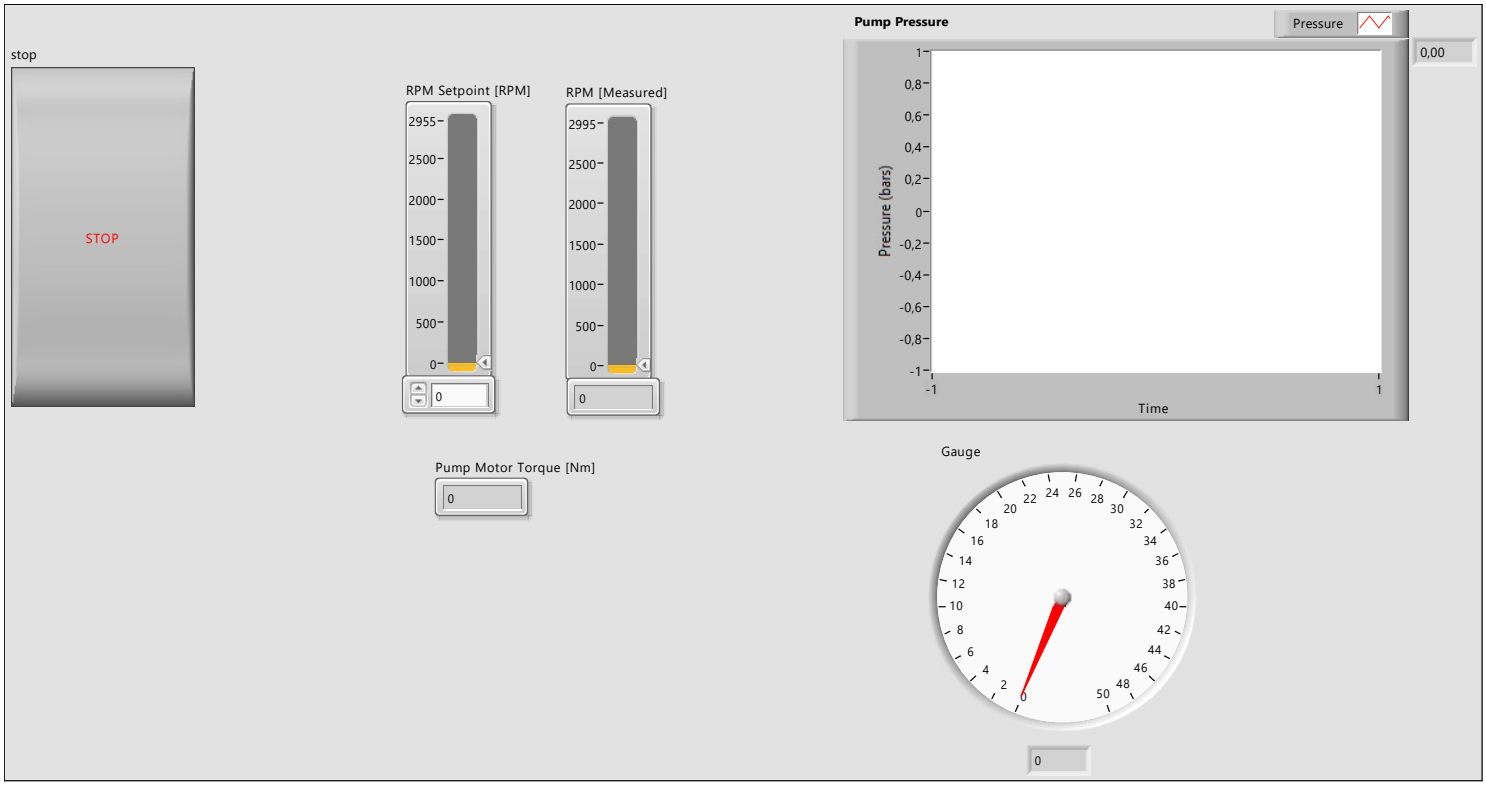
scaling16bit2RPM_TCP.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\LabVIEW Drillbotics 2018\Conversions\scaling16bit2RPM_TCP.vi

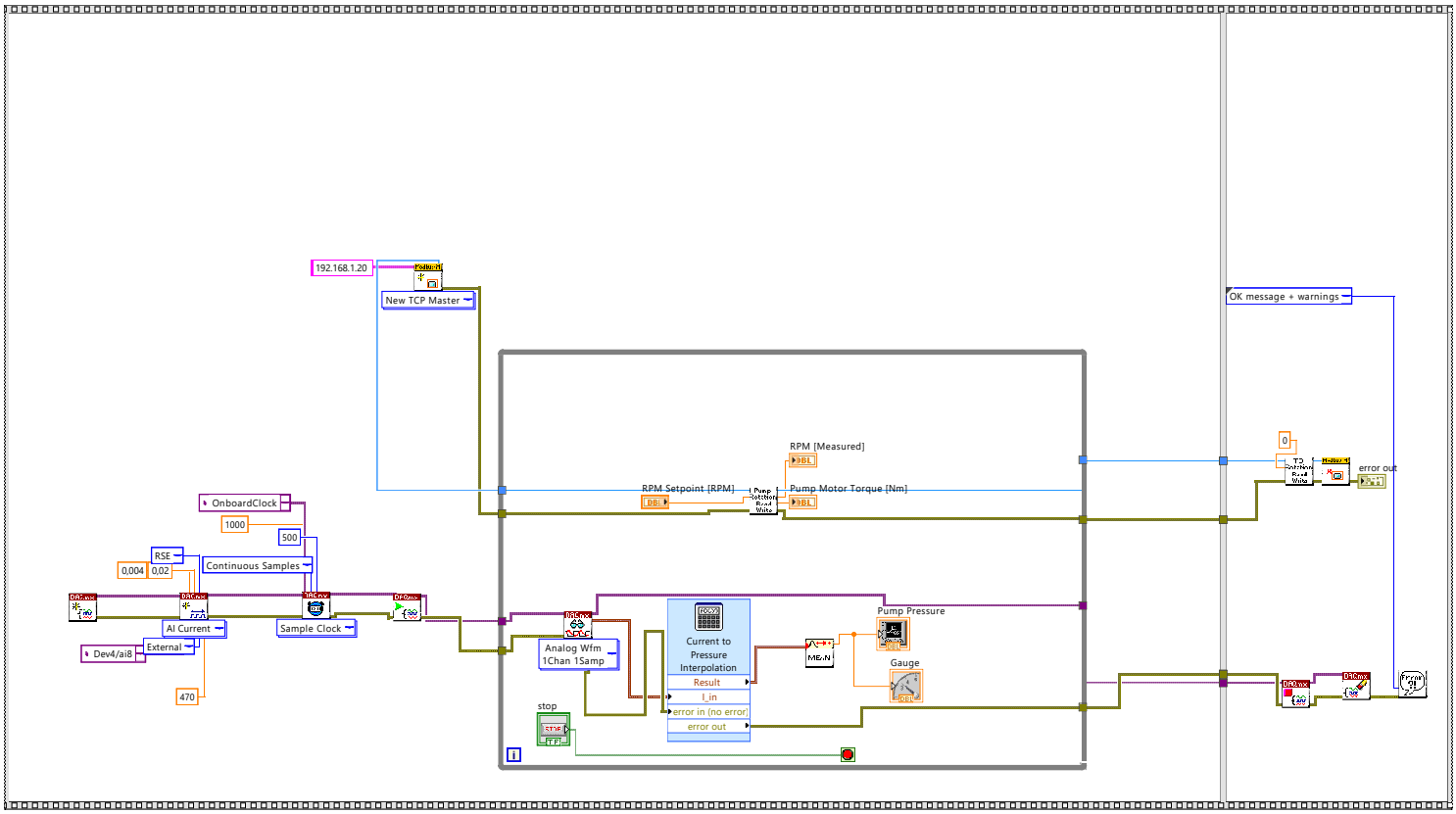
scalingTorque216bit_TCP_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Convertion subVIs\scalingTorque216bit_TCP_2019.vi

scalingRPM216bit_TCP_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Convertion subVIs\scalingRPM216bit_TCP_2019.vi

C.2.2 Pump Motor Manual

Pump_manual.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Pump Motor\Pump_manual.vi
Last modified on 18.06.2019 at 12.27
Printed on 18.06.2019 at 12.41





Simple Error Handler.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Utility\error.llb\Simple Error Handler.vi

Modbus Master.lvclass:Close.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Modbus\master\Close.vi

Pump_manual.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Pump Motor\Pump_manual.vi
Last modified on 18.06.2019 at 12.27
Printed on 18.06.2019 at 12.41

Modbus Master.lvclass:Create Master Instance.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Modbus\master\Create Master Instance.vi

Modbus Master.lvclass:Create Master Instance (TCP).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Modbus\master\Create Master Instance (TCP).vi

DialogType.ctl
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Utility\error.llb\DialogType.ctl

DAQmx Clear Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Clear Task.vi

DAQmx Fill In Error Info.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\miscellaneous.llb\DAQmx Fill In Error Info.vi

DAQmx Stop Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Stop Task.vi

TD_TCP_SubVI.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\TD Motor Old\SubVI's\TD_TCP_SubVI.vi

NI_PtbyPt.lvlib:Mean PtByPt.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\ptbypt\Probability & Statistics.llb\Mean PtByPt.vi

ex_Modify Signal Name.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\express\express shared\ex_Modify Signal Name.vi

Pump_TCP_SubVI.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Pump Motor\SubVI's\Pump_TCP_SubVI.vi

DAQmx Start Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Start Task.vi

DAQmx Timing (Sample Clock).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\timing.llb\DAQmx Timing (Sample Clock).vi

DAQmx Timing.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\timing.llb\DAQmx Timing.vi

DAQmx Create Channel (AI-Current-Basic).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\channels.llb\DAQmx Create Channel (AI-Current-Basic).vi

DAQmx Create Virtual Channel.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\channels.llb\DAQmx Create Virtual Channel.vi

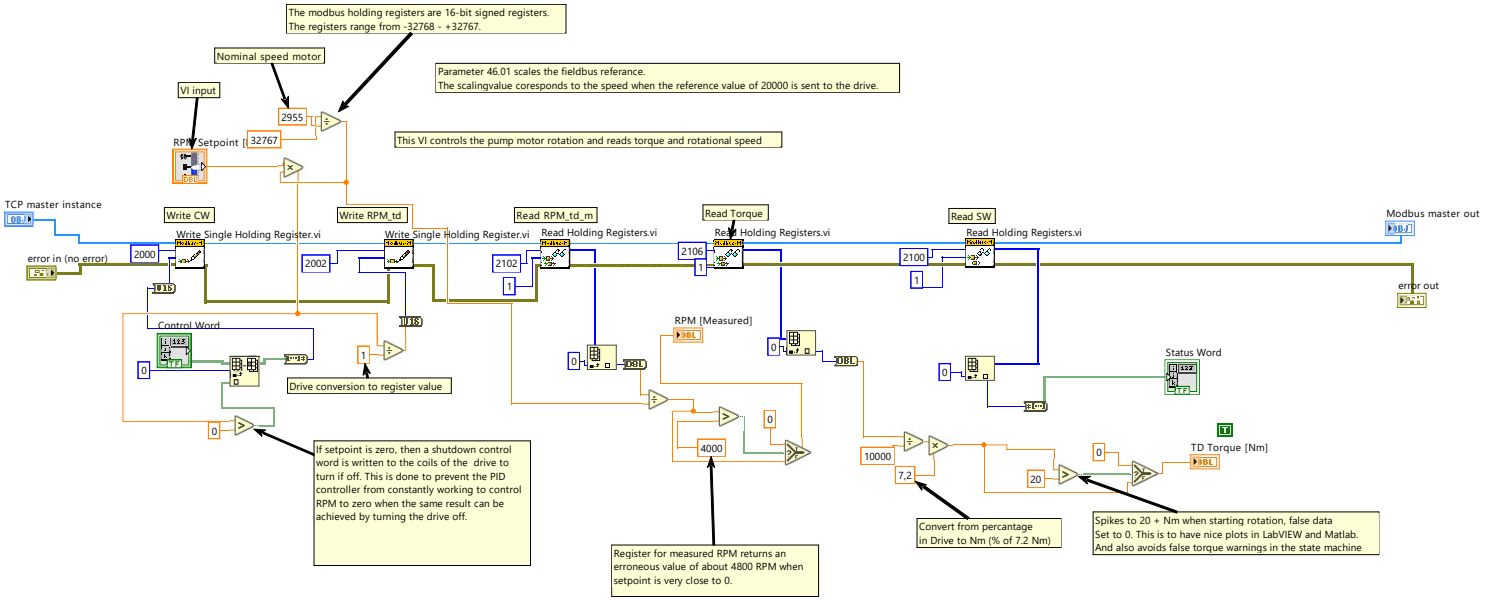
Pump_manual.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Pump Motor\Pump_manual.vi
Last modified on 18.06.2019 at 12.27
Printed on 18.06.2019 at 12.41

DAQmx Create Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\task.llb\DAQmx Create Task.vi

DAQmx Read (Analog Wfm 1Chan 1Samp).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\read.llb\DAQmx Read (Analog Wfm 1Chan 1Samp).vi

DAQmx Read.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\read.llb\DAQmx Read.vi

Pump_TCP_SubVI.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Pump Motor\SubVI's\Pump_TCP_SubVI.vi
Last modified on 30.05.2019 at 14.47
Printed on 18.06.2019 at 12.42

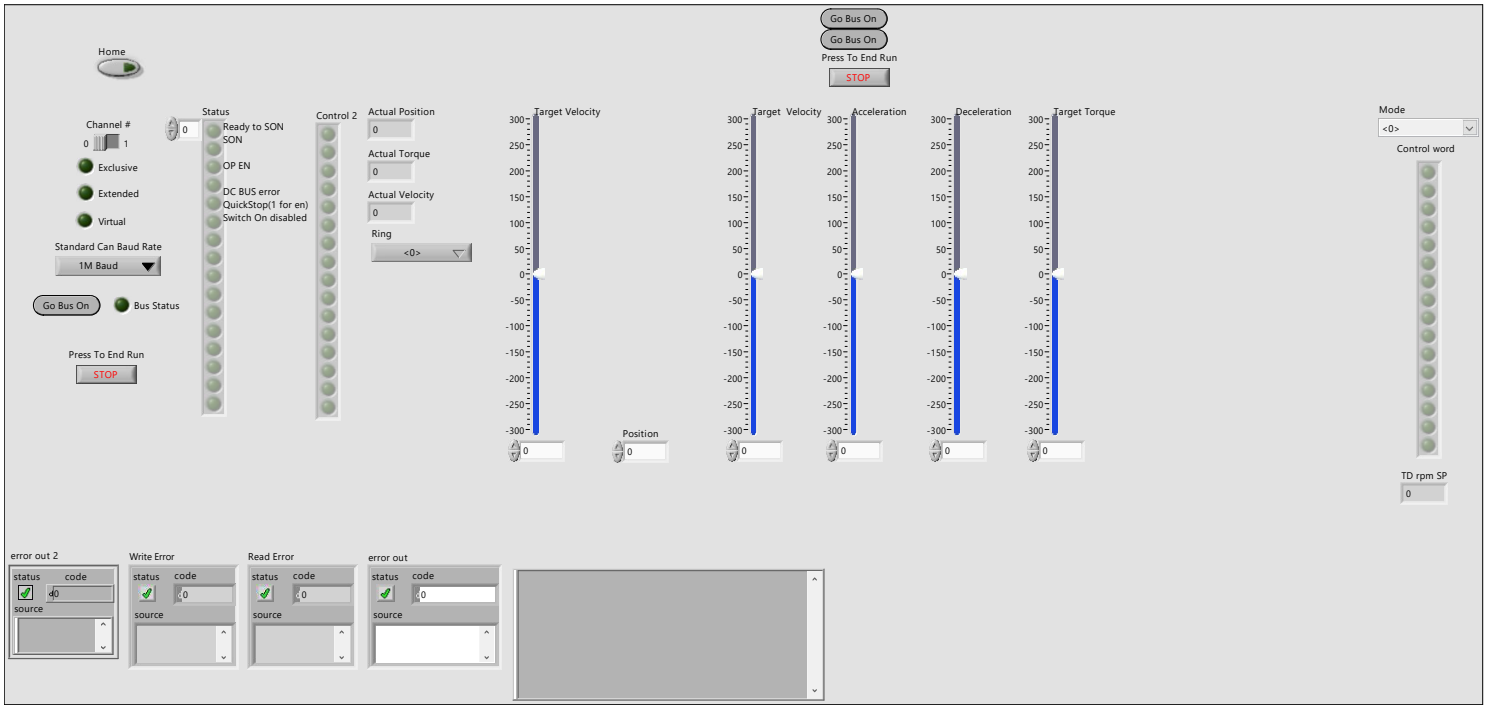


Modbus Master.lvclass:Read Holding Registers.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Modbus\master\Functions\Read Holding Registers.vi

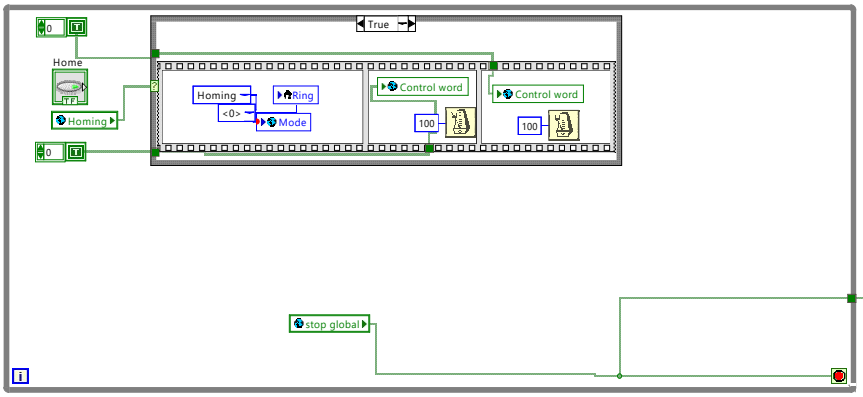
Modbus Master.lvclass:Write Single Holding Register.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Modbus\master\Functions\Write Single Holding Register.vi

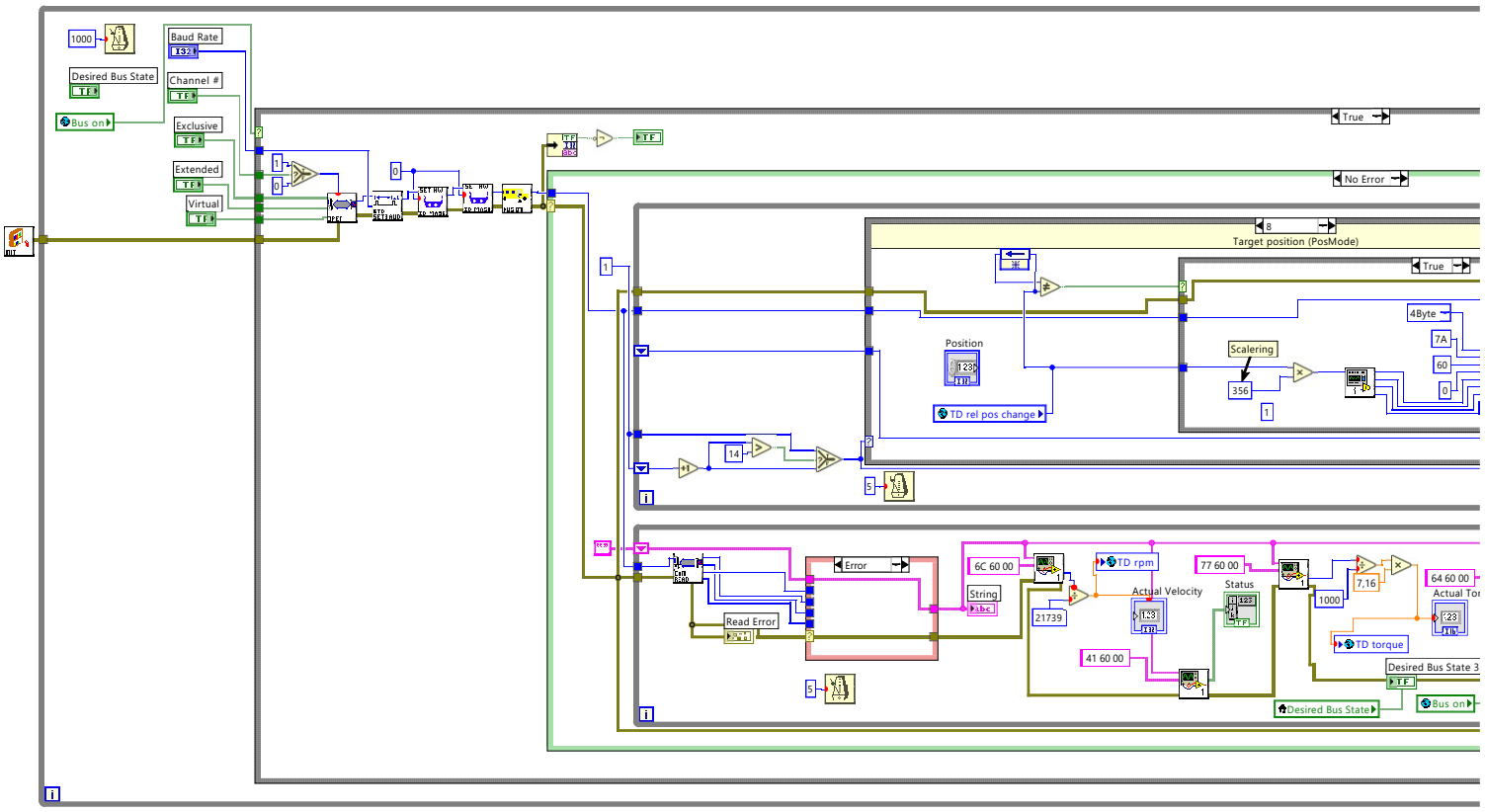
C.2.3 Top Drive Manual

TD_main.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\TD_main.vi
Last modified on 05.06.2019 at 11.12
Printed on 18.06.2019 at 12.51



TD_main.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\TD_main.vi
Last modified on 05.06.2019 at 11.12
Printed on 18.06.2019 at 12.51





kvCanBusOff.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\kvCanVI.llb\kvCanBusOff.vi

MatchString.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\MatchString.vi

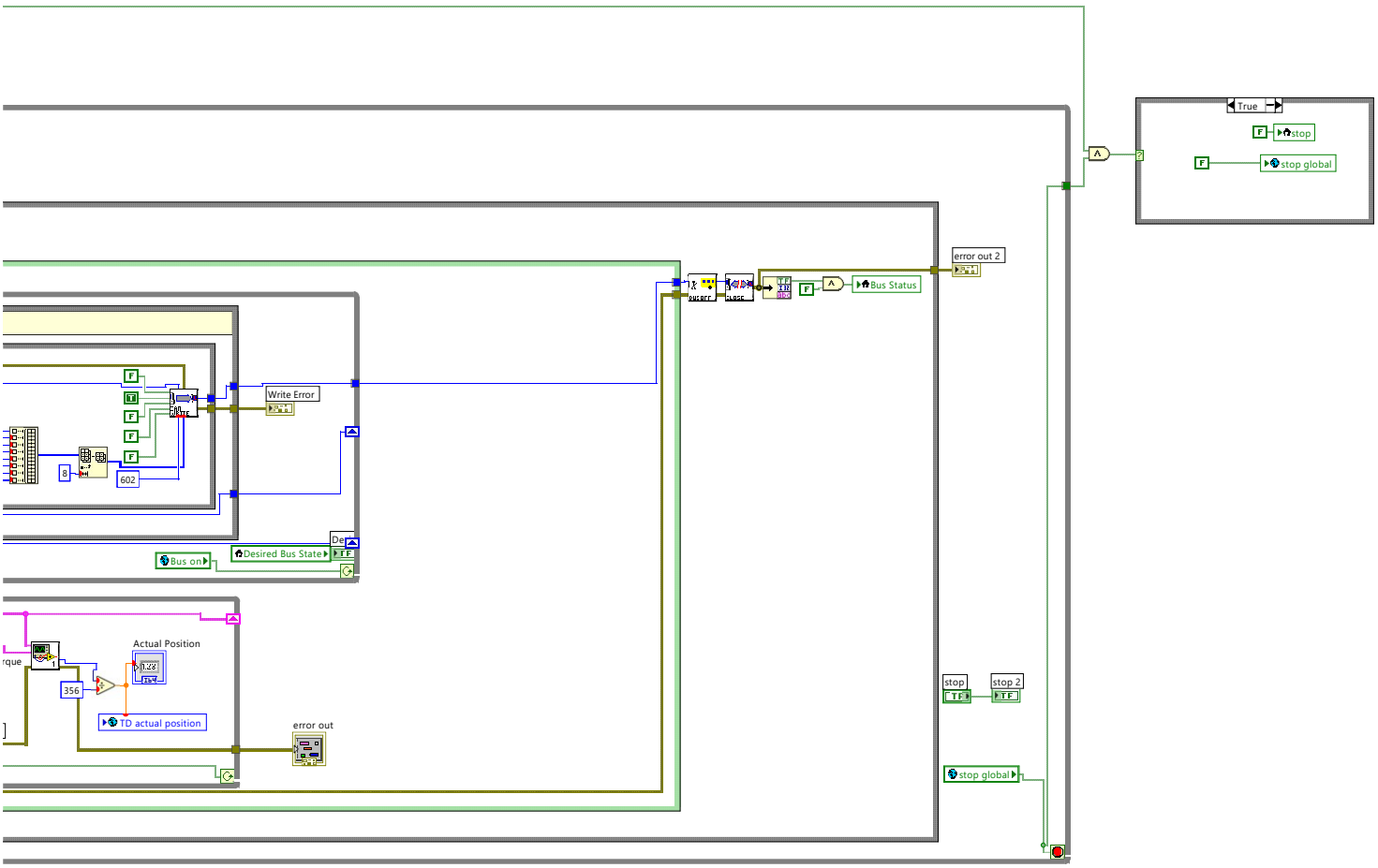
global_td_torque.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td_torque.vi

kvCanSetStdBusParams.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\kvCanVI.llb\kvCanSetStdBusParams.vi

16_8HILO.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\16_8HILO.vi

kvCanClose.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\kvCanVI.llb\kvCanClose.vi

TD_main.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\TD_main.vi
Last modified on 05.06.2019 at 11.12
Printed on 18.06.2019 at 12.51



TD_main.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\TD_main.vi
Last modified on 05.06.2019 at 11.12
Printed on 18.06.2019 at 12.51

32i_8i4byte.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\32i_8i4byte.vi

kvCanBusOn.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\kvCanVI.llb\kvCanBusOn.vi

kvCanRead.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\kvCanVI.llb\kvCanRead.vi

global_td_rel_pos_change.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td_rel_pos_change.vi

global_td_homing.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td_homing.vi

kvCanMsgDisplayBox.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\kvCanVI.llb\kvCanMsgDisplayBox.vi

global_td_control_word.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td_control_word.vi

global_td_rpm.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td_rpm.vi

global_td_rpm_sp.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td_rpm_sp.vi

global_td__mode_selector.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td__mode_selector.vi

kvCanAccept.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\kvCanVI.llb\kvCanAccept.vi

kvCanWrite.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\kvCanVI.llb\kvCanWrite.vi

global_stop_button.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_stop_button.vi

global_td_bus_on.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td_bus_on.vi

global_td_actual_pos.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td_actual_pos.vi

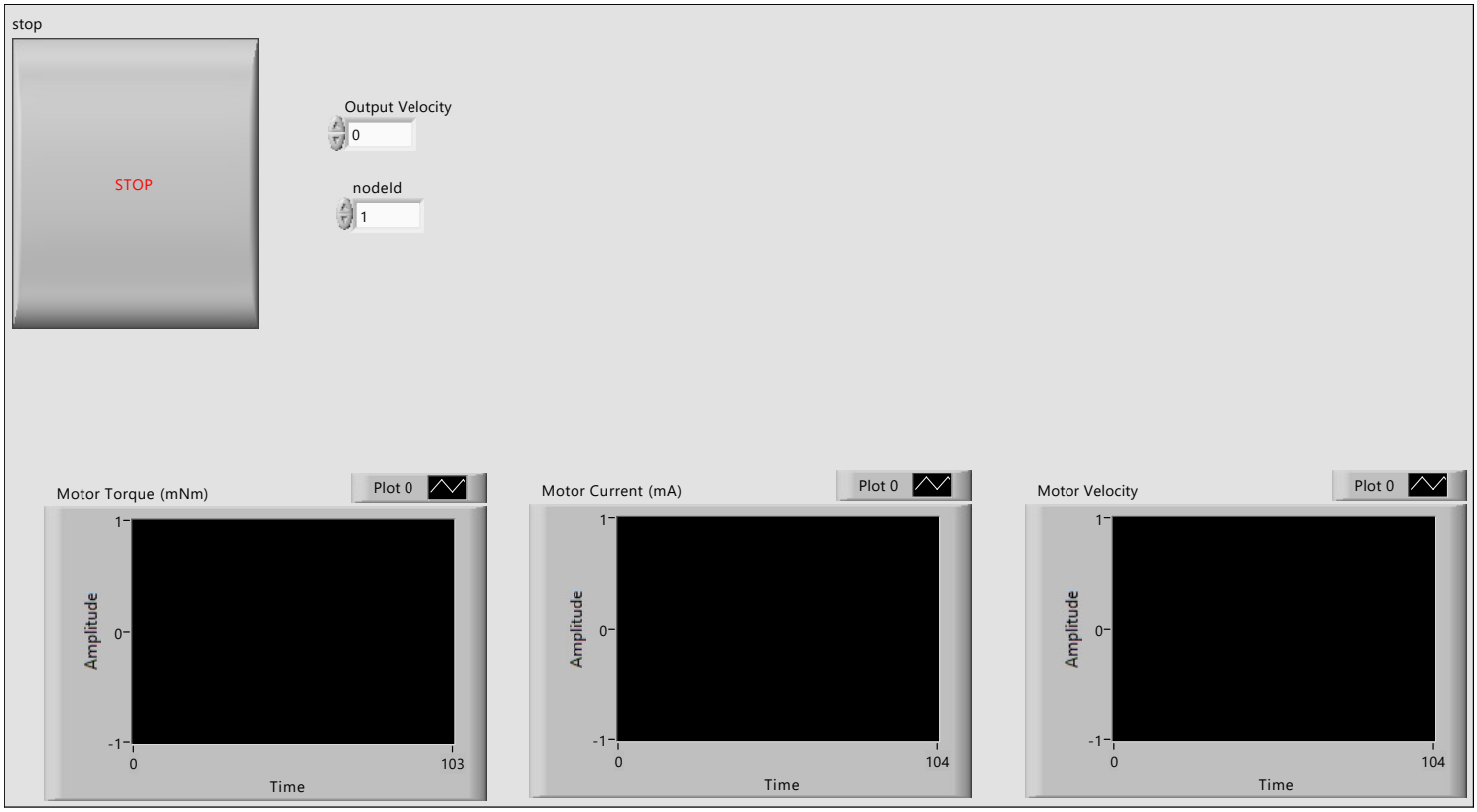
TD_main.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\TD_main.vi
Last modified on 05.06.2019 at 11.12
Printed on 18.06.2019 at 12.51

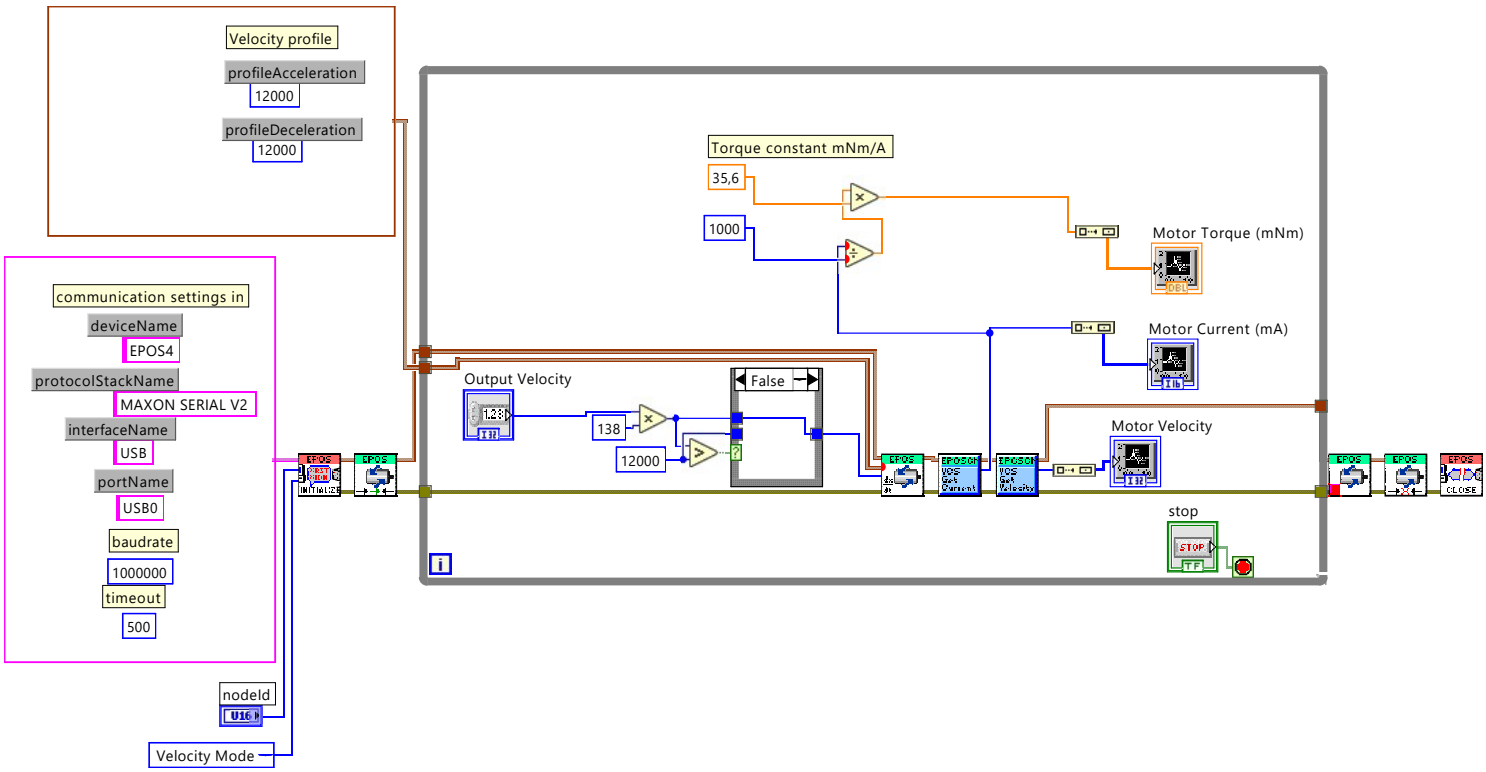
kvCanOpenChannel.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\kvCanVI.llb\kvCanOpenChannel.vi

kvCanInitialize.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\kvCanVI.llb\kvCanInitialize.vi

C.2.4 Electrical Mini Motor Manual

Maxon Manual with sensor.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\Maxon Manual with sensor.vi
Last modified on 18.06.2019 at 13.52
Printed on 18.06.2019 at 13.54





StopAxis.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\EPOS-LabVIEW-Instrument-Driver-En\LabVIEW\maxon EPOS\Operation\Motion Info\StopAxis.vi

Maxon Manual with sensor.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\Maxon Manual with sensor.vi
Last modified on 18.06.2019 at 13.52
Printed on 18.06.2019 at 13.54

DisableAxis.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\EPOS-LabVIEW-Instrument-Driver-En\LabVIEW\maxon EPOS\Operation\State Machine\DisableAxis.vi

Initialize.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\EPOS-LabVIEW-Instrument-Driver-En\LabVIEW\maxon EPOS\Initialization\Initialize.vi

MoveWithVelocity.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\EPOS-LabVIEW-Instrument-Driver-En\LabVIEW\maxon EPOS\Operation\Profile Velocity Mode\MoveWithVelocity.vi

VCS Get Current Is.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\EPOS-LabVIEW-Instrument-Driver-En\LabVIEW\maxon EPOS\Operation\Motion Info\VCS Layer\VCS Get Current Is.vi

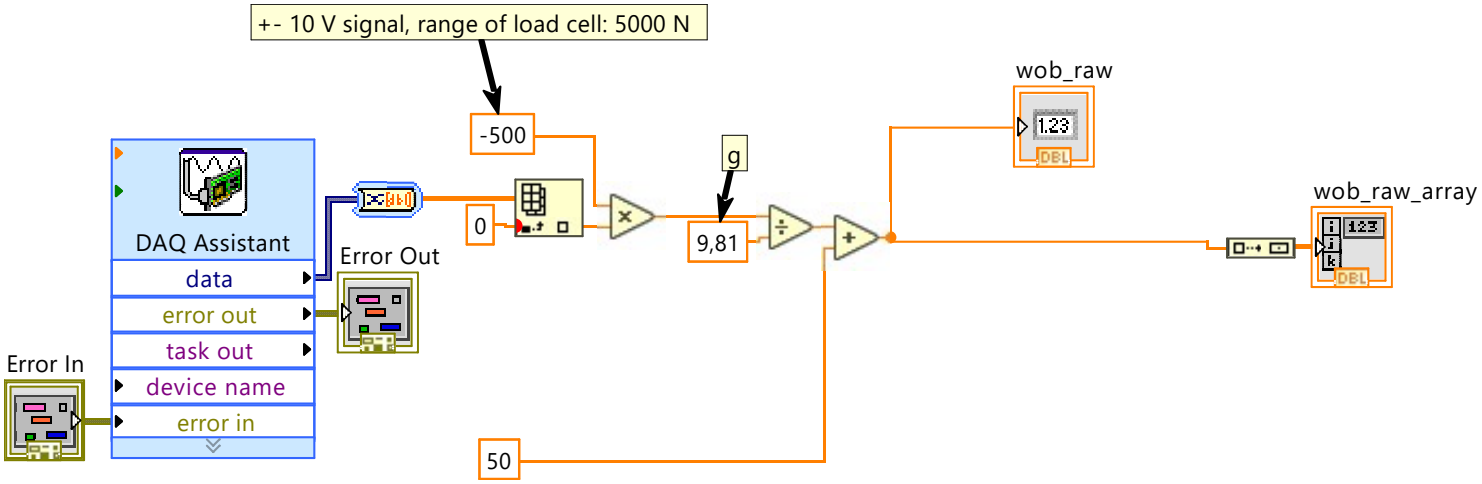
EnableAxis.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\EPOS-LabVIEW-Instrument-Driver-En\LabVIEW\maxon EPOS\Operation\State Machine\EnableAxis.vi

Close.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\EPOS-LabVIEW-Instrument-Driver-En\LabVIEW\maxon EPOS\Initialization\Close.vi

VCS Get Velocity Is.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\EPOS-LabVIEW-Instrument-Driver-En\LabVIEW\maxon EPOS\Operation\Motion Info\VCS Layer\VCS Get Velocity Is.vi

C.2.5 Load Cell Individual

Load_Cell_DAQassistant_subVI.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Load Cell\SubVI's\Load_Cell_DAQassistant_subVI.vi
Last modified on 27.05.2019 at 18.18
Printed on 18.06.2019 at 16.10



Dynamic To Waveform Array.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\express\express shared\transition.llb\Dynamic To Waveform Array.vi

DAQmx Clear Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Clear Task.vi

DAQmx Stop Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Stop Task.vi

DAQmx Fill In Error Info.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\miscellaneous.llb\DAQmx Fill In Error Info.vi

Convert 1DDbl to DDT 2.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\miscellaneous.llb\Convert 1DDbl to DDT 2.vi

DAQmx Create Channel (AI-Voltage-Basic).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\channels.llb\DAQmx Create Channel (AI-Voltage-Basic).vi

DAQmx Start Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Start Task.vi

DAQmx Create Channel (TEDS-AI-Voltage-Basic).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\channels.llb\DAQmx Create Channel (TEDS-AI-Voltage-Basic).vi

DAQmx Create Virtual Channel.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\channels.llb\DAQmx Create Virtual Channel.vi

DAQmx Create Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\task.llb\DAQmx Create Task.vi

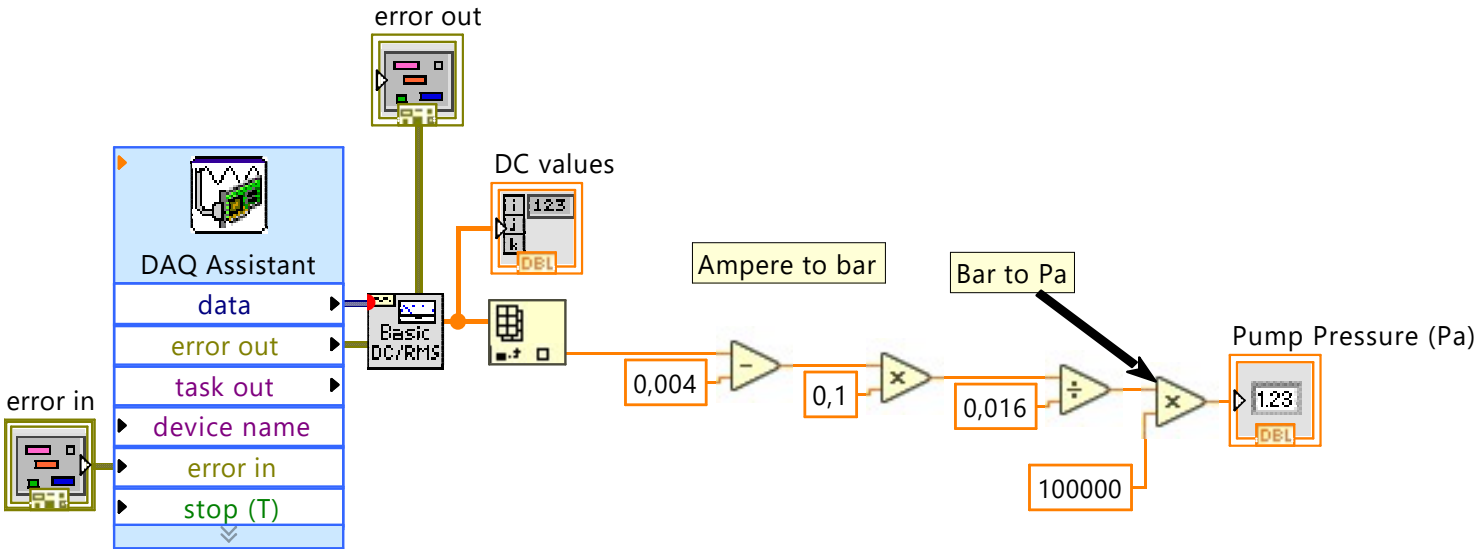
DAQmx Read (Analog 1D DBL NChan 1Samp).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\read.llb\DAQmx Read (Analog 1D DBL NChan 1Samp).vi

Load_Cell_DAQassistant_subVI.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Load Cell\SubVI's\Load_Cell_DAQassistant_subVI.vi
Last modified on 27.05.2019 at 18.18
Printed on 18.06.2019 at 16.10

DAQmx Read.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\read.llb\DAQmx Read.vi

C.2.6 Pressure Gauge Individual

Pump pressure DAQ sensor.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Pump Pressure Gauge\Pump pressure DAQ sensor.vi
Last modified on 14.05.2019 at 08.58
Printed on 18.06.2019 at 16.29



NI_MAPro.lvlib:Basic Averaged DC-RMS.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\measure\madcrms.llb\Basic Averaged DC-RMS.vi

DAQmx Clear Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Clear Task.vi

DAQmx Fill In Error Info.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\miscellaneous.llb\DAQmx Fill In Error Info.vi

DAQmx Stop Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Stop Task.vi

Convert 1DDbl to DDT 2.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\miscellaneous.llb\Convert 1DDbl to DDT 2.vi

DAQmx Start Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Start Task.vi

DAQmx Create Channel (AI-Current-Basic).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\channels.llb\DAQmx Create Channel (AI-Current-Basic).vi

DAQmx Create Channel (TEDS-AI-Current-Basic).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\channels.llb\DAQmx Create Channel (TEDS-AI-Current-Basic).vi

DAQmx Create Virtual Channel.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\channels.llb\DAQmx Create Virtual Channel.vi

DAQmx Create Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\task.llb\DAQmx Create Task.vi

Pump pressure DAQ sensor.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Pump Pressure Gauge\Pump pressure DAQ sensor.vi
Last modified on 14.05.2019 at 08.58
Printed on 18.06.2019 at 16.29

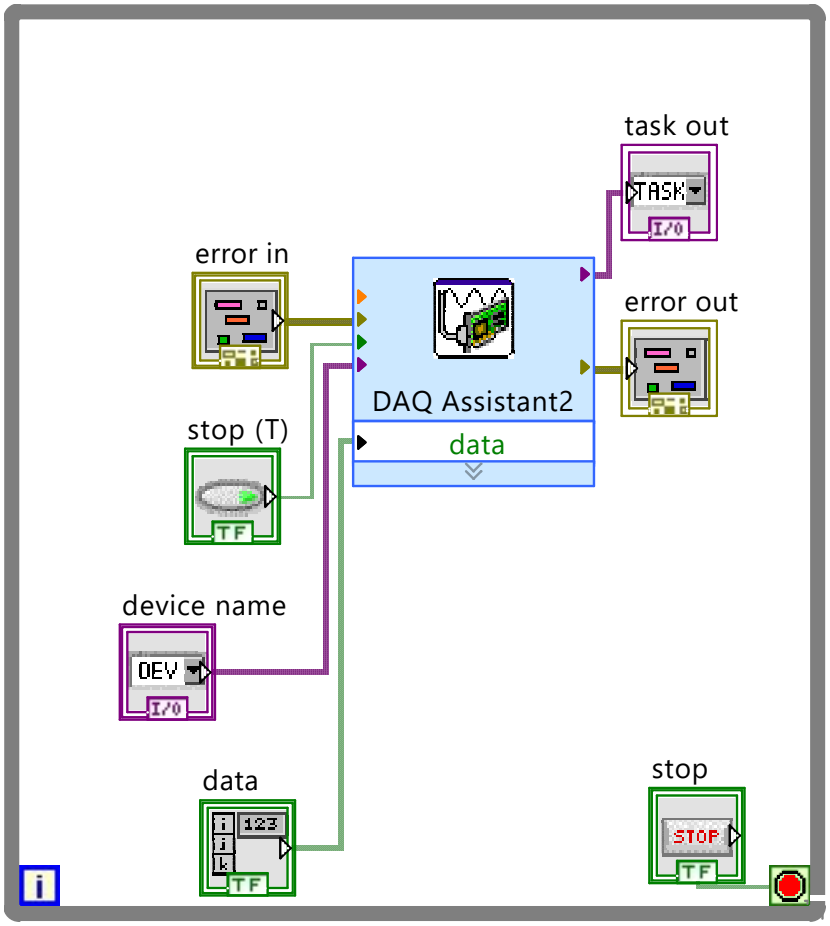
DAQmx Read (Analog 1D DBL NChan 1Samp).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\read.llb\DAQmx Read (Analog 1D DBL NChan 1Samp).vi

DAQmx Read.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\read.llb\DAQmx Read.vi

NI_MAPro.lvlib:Basic Averaged DC-RMS for N Chan.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\measure\madcrms.llb\Basic Averaged DC-RMS for N Chan.vi

C.2.7 Solenoid Valve Individual

Valve Manual - DAQ.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Solenoid Valve\Valve Manual - DAQ.vi
Last modified on 30.04.2019 at 14.48
Printed on 18.06.2019 at 16.34



DAQmx Create Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\task.llb\DAQmx Create Task.vi

DAQmx Write (Digital 1D Bool NChan 1Samp 1Line).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\write.llb\DAQmx Write (Digital 1D Bool NChan 1Samp 1Line).vi

DAQmx Write.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\write.llb\DAQmx Write.vi

DAQmx Create Virtual Channel.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\channels.llb\DAQmx Create Virtual Channel.vi

DAQmx Start Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Start Task.vi

DAQmx Create Channel (DO-Digital Output).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\create\channels.llb\DAQmx Create Channel (DO-Digital Output).vi

DAQmx Clear Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Clear Task.vi

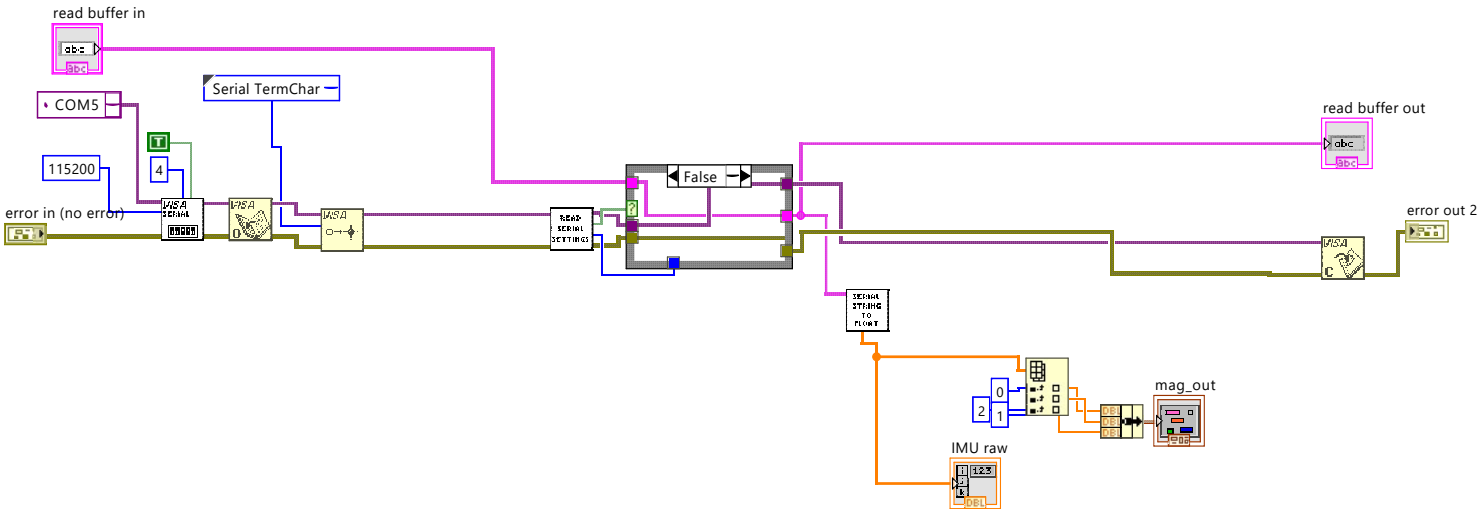
Valve Manual - DAQ.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Solenoid Valve\Valve Manual - DAQ.vi
Last modified on 30.04.2019 at 14.48
Printed on 18.06.2019 at 16.34

DAQmx Fill In Error Info.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\miscellaneous.llb\DAQmx Fill In Error Info.vi

DAQmx Stop Task.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\DAQmx\configure\task.llb\DAQmx Stop Task.vi

C.2.8 Down Hole Sensor Card Individual

Untitled 1
Last modified on 18.06.2019 at 17.25
Printed on 18.06.2019 at 17.25



VISA Configure Serial Port
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Instr_visa.llb\VISA Configure Serial Port

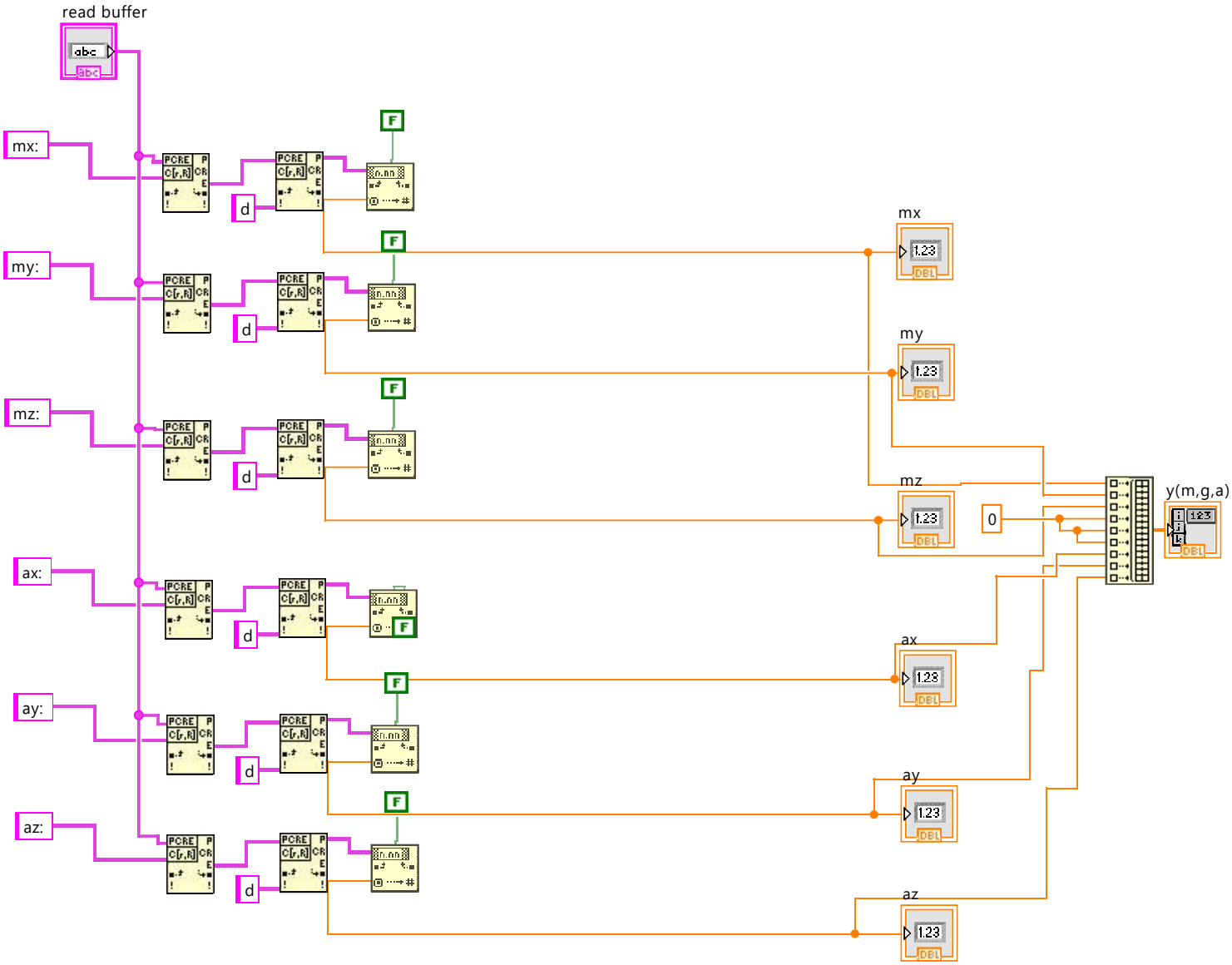
VISA Configure Serial Port (Instr).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Instr_visa.llb\VISA Configure Serial Port (Instr).vi

Select Event Type.ctl
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Instr_visa.llb\Select Event Type.ctl

serial_string_to_mag_raw.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\serial_string_to_mag_raw.vi

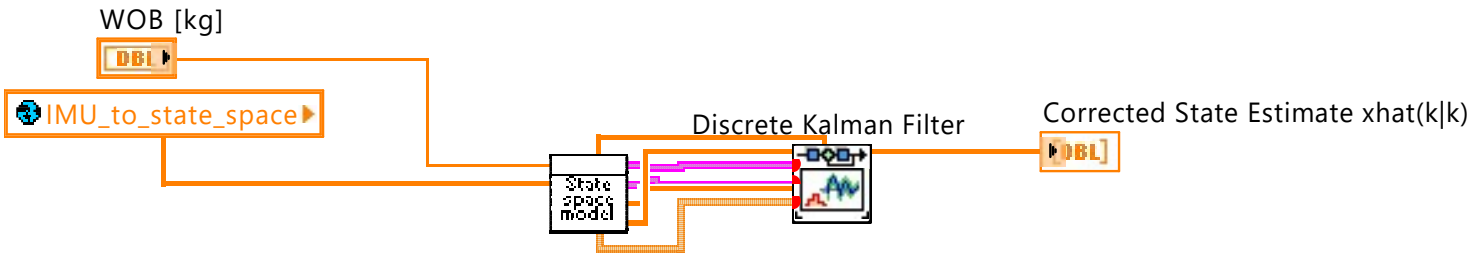
read_serial_inlet.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\read_serial_inlet.vi

serial_string_to_mag_raw.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\serial_string_to_mag_raw.vi
Last modified on 13.06.2019 at 09.39
Printed on 18.06.2019 at 17.29



C.2.9 Kalman filter subvi

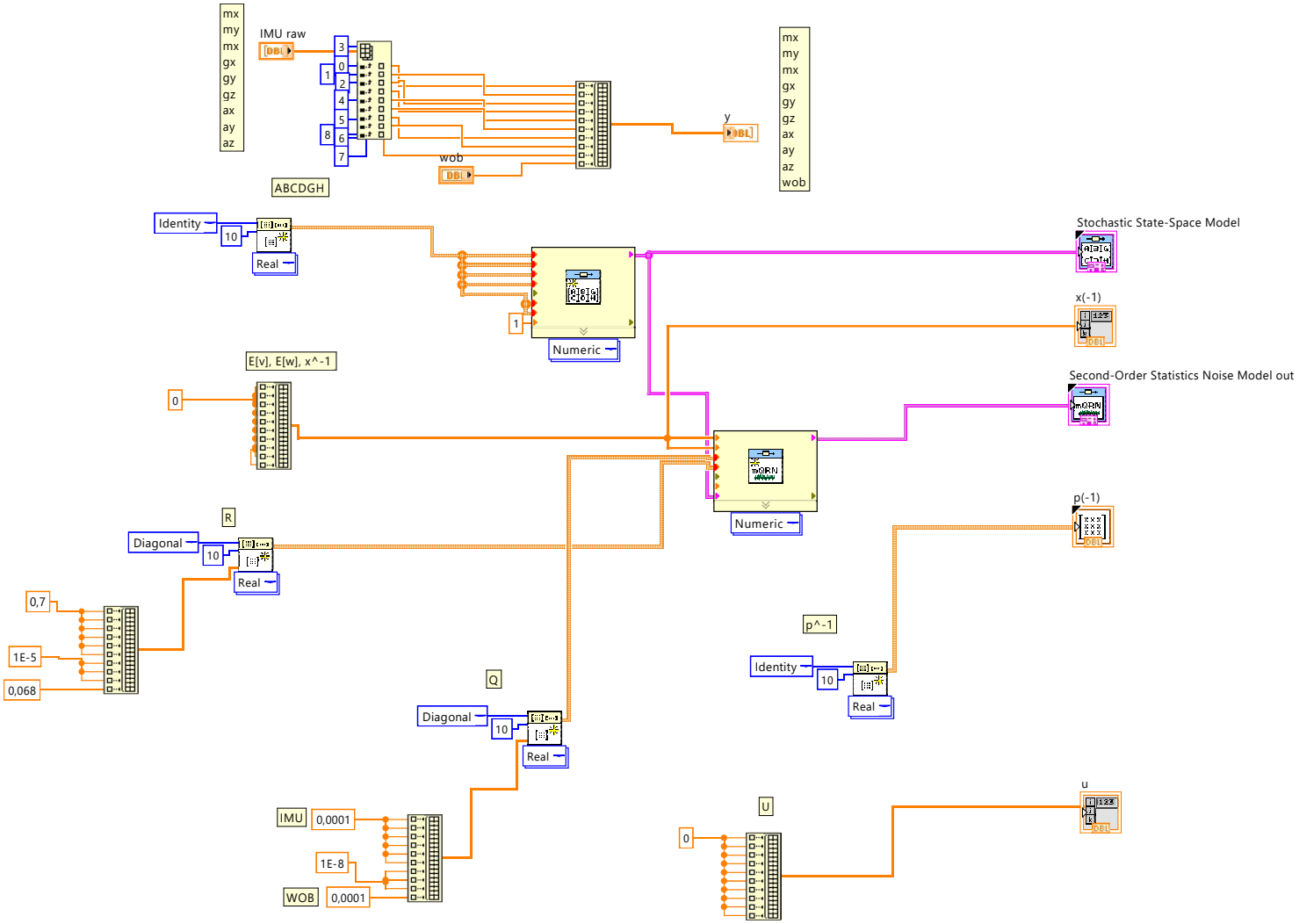
Untitled 2
Last modified on 18.06.2019 at 19.50
Printed on 18.06.2019 at 19.50



global_imu_raw_data.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_imu_raw_data.vi

state_space_model_kalman_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\state_space_model_kalman_sub.vi

state_space_model_kalman_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Sub-VIs\state_space_model_kalman_sub.vi
Last modified on 05.06.2019 at 18.33
Printed on 18.06.2019 at 19.50



NI_Matrix.lvlib:RealMatrix.ctl
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Analysis\Matrix\Datatypes\RealMatrix.ctl

NI_AALPro.lvlib:Create Special Matrix.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Analysis\7linalg.llb\Create Special Matrix.vi

NI_CD_Stochastic Systems.lvlib:cd_Stochastic State-Space Model.ctl
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\addons\Control Design_Stochastic Systems\Stochastic Systems.llb\Stochastic Systems SubVIs\cd_Stochastic State-Space Model.ctl

NI_CD_Stochastic Systems.lvlib:cd_Second-Order Statistics Noise Model.ctl
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\addons\Control Design_Stochastic Systems\Stochastic Systems.llb\Stochastic Systems SubVIs\cd_Second-Order Statistics Noise Model.ctl

NI_AALPro.lvlib:Create Special Real Matrix.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Analysis\7linalg.llb\Create Special Real Matrix.vi

NI_CD_Stochastic Systems.lvlib:CD Construct Stochastic Model.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\addons\Control Design_Stochastic Systems\Stochastic Systems.llb\CD Construct Stochastic Model.vi

state_space_model_kalman_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Sub-VIs\state_space_model_kalman_sub.vi
Last modified on 05.06.2019 at 18.33
Printed on 18.06.2019 at 19.50

NI_CD_Stochastic Systems.lvlib:CD Construct Noise Model (Numeric).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\addons\Control Design_Stochastic Systems\Stochastic Systems.llb\CD Construct Noise Model (Numeric).vi

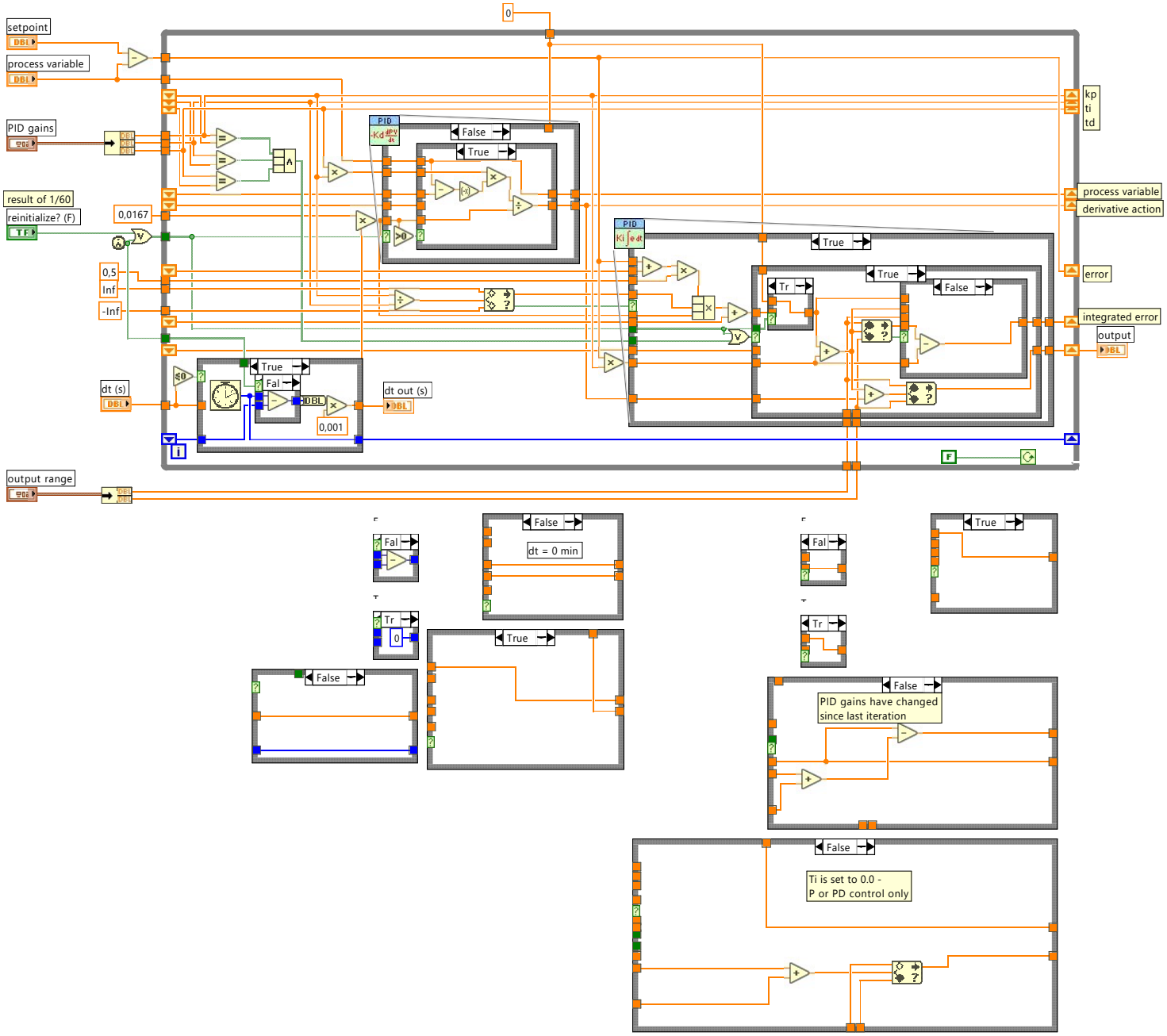
NI_CD_Stochastic Systems.lvlib:CD Construct Stochastic Model (Numeric).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\addons\Control Design_Stochastic Systems\Stochastic Systems.llb\CD Construct Stochastic Model (Numeric).vi

NI_CD_Stochastic Systems.lvlib:CD Construct Noise Model.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\addons\Control Design_Stochastic Systems\Stochastic Systems.llb\CD Construct Noise Model.vi

C.2.10 WOB Controller subVI

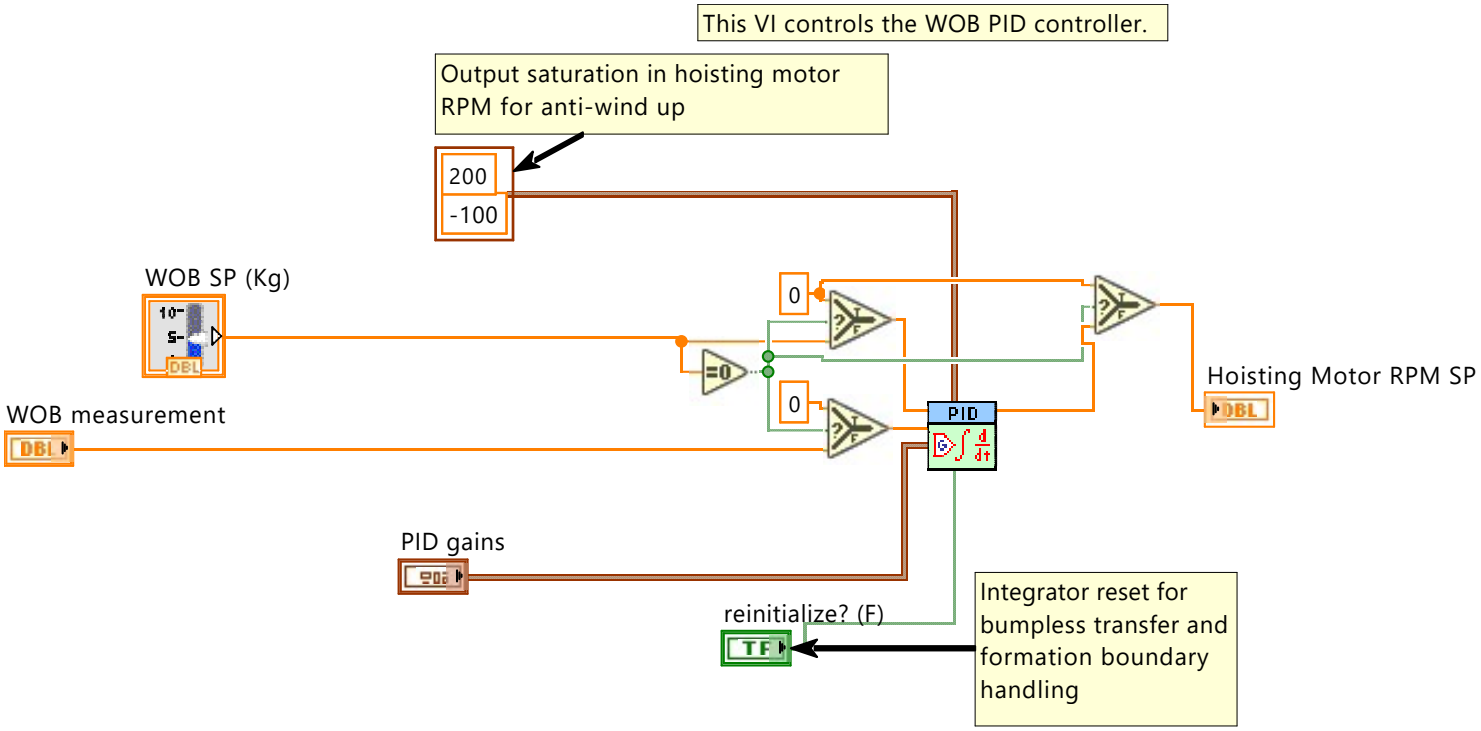
NI_PID_pid.lvlib:PID (DBL).vi:7250002
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\addons\control\pid\pid.llb\PID (DBL).vi
Last modified on 08.03.2018 at 18.59
Printed on 18.06.2019 at 20.08

Block Diagram



List of SubVIs and Express VIs

PID_WOB (SubVI).vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\PID_WOB (SubVI).vi
Last modified on 04.06.2019 at 21.10
Printed on 18.06.2019 at 20.10

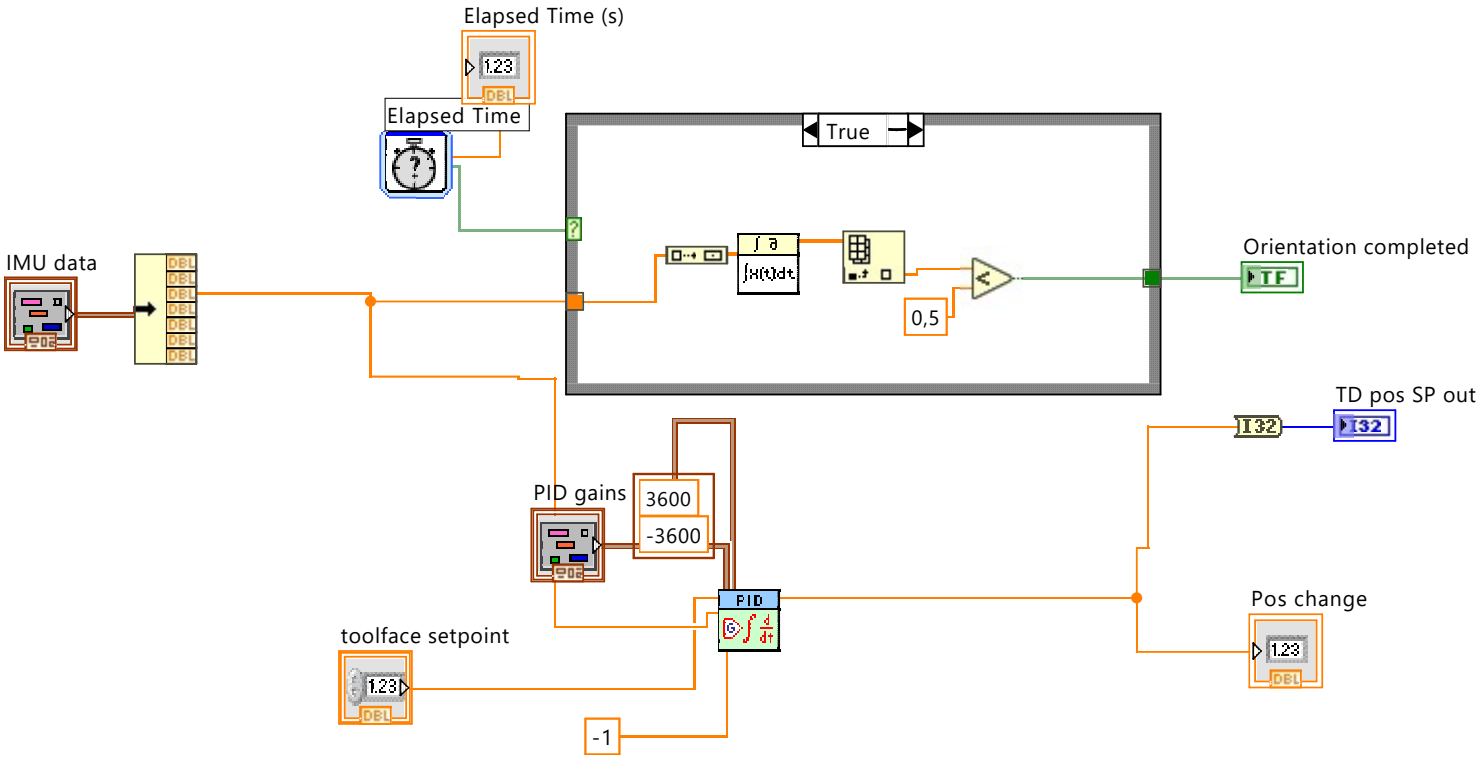


NI_PID_pid.lvlib:PID (DBL).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\addons\control\pid\pid.llb\PID (DBL).vi

NI_PID_pid.lvlib:PID.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\addons\control\pid\pid.llb\PID.vi

C.2.11 Position Controller subVI

position_control_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\position_control_sub.vi
Last modified on 08.06.2019 at 21.43
Printed on 18.06.2019 at 20.20



NI_AALPro.lvlib:Integral x(t).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Analysis\2dsp.llb\Integral x(t).vi

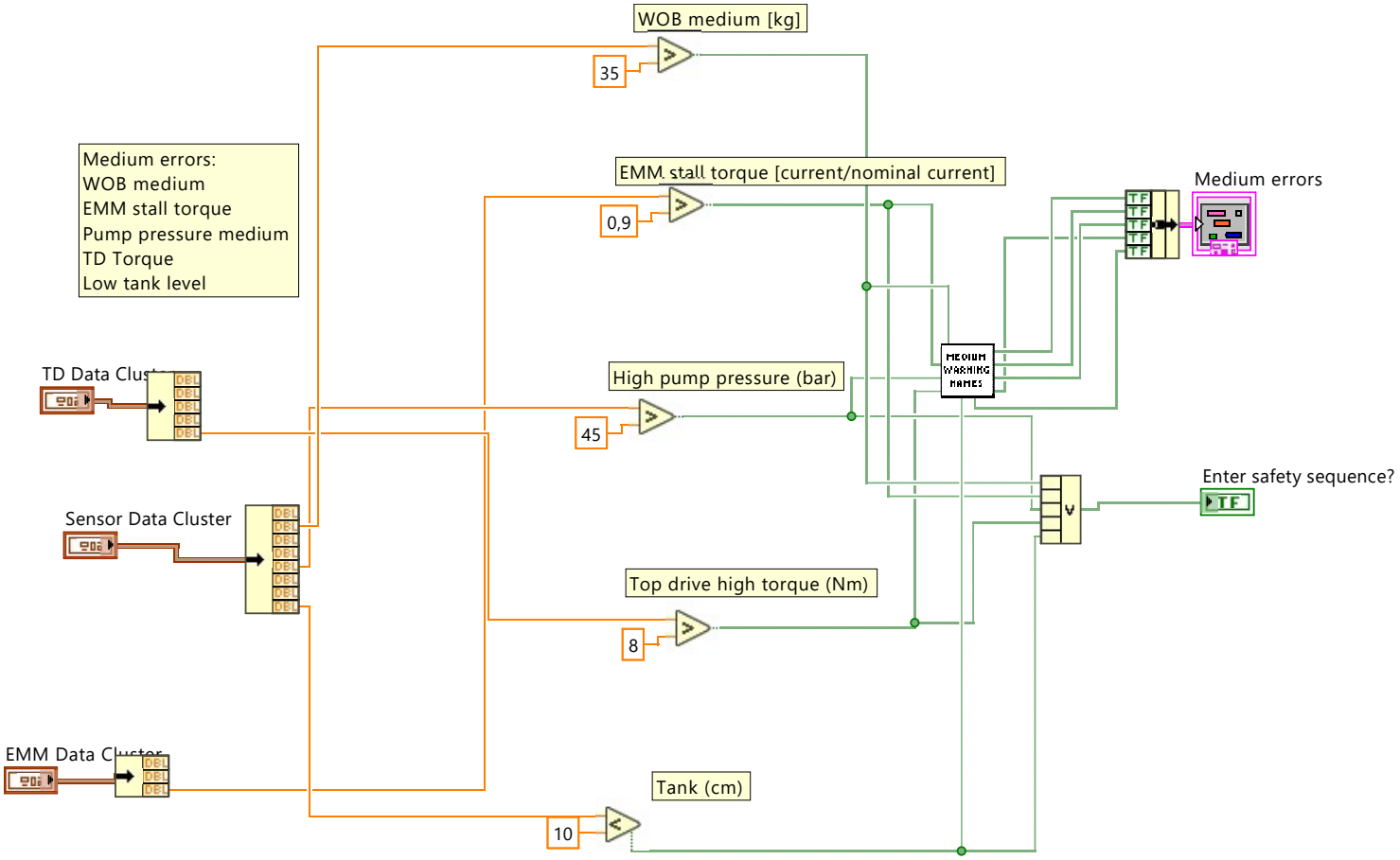
subElapsedTime.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\express\express execution control\ElapsedTimeBlock.llb\subElapsedTime.vi

NI_PID_pid.lvlib:PID (DBL).vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\addons\control\pid\pid.llb\PID (DBL).vi

NI_PID_pid.lvlib:PID.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\addons\control\pid\pid.llb\PID.vi

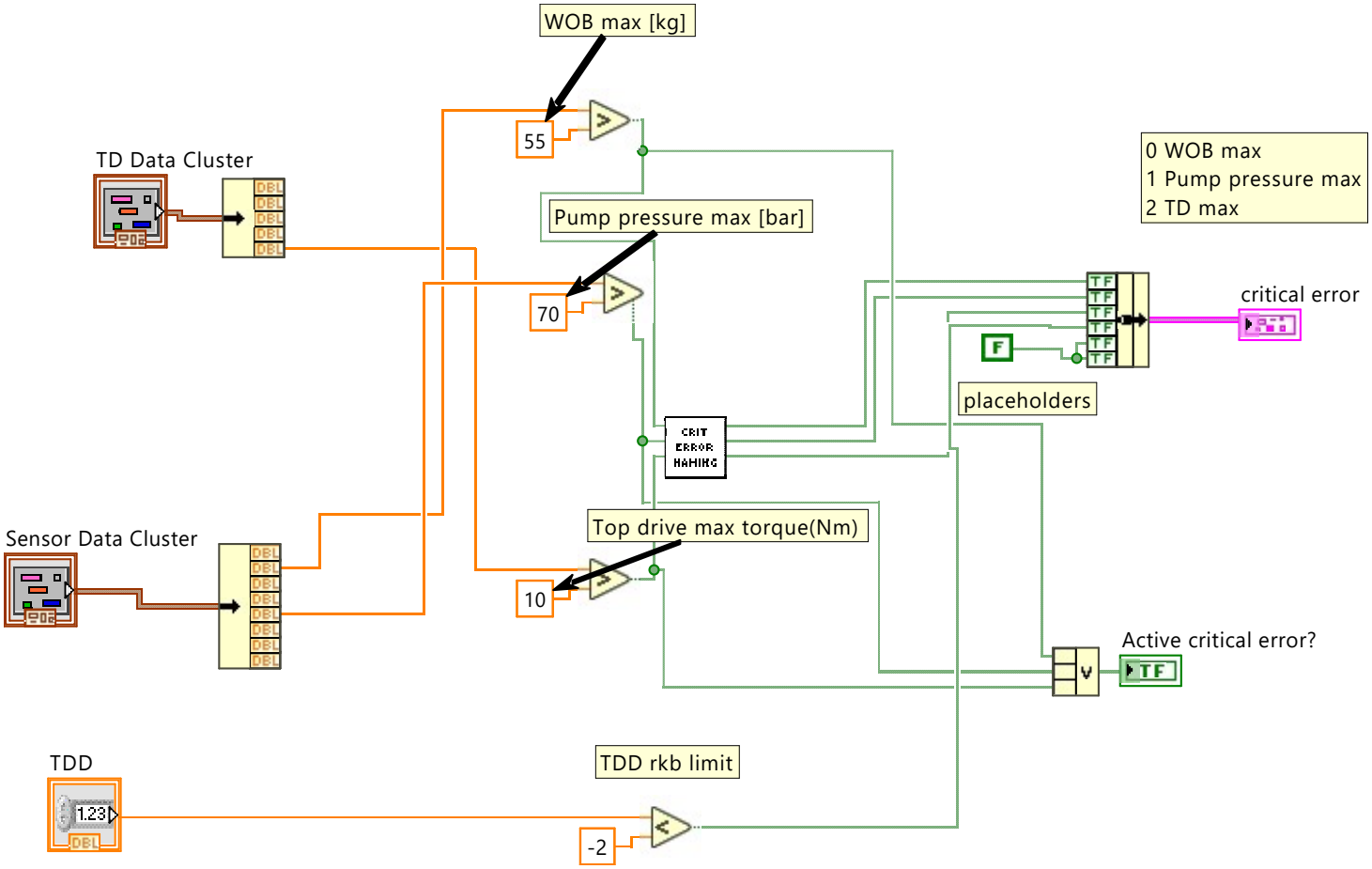
C.2.12 Safety Logic subVI

detect_medium_warning.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\detect_medium_warning.vi
Last modified on 03.06.2019 at 20.24
Printed on 19.06.2019 at 12.41



Medium_warning_naming_sub_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\Medium_warning_naming_sub_Hilde.vi

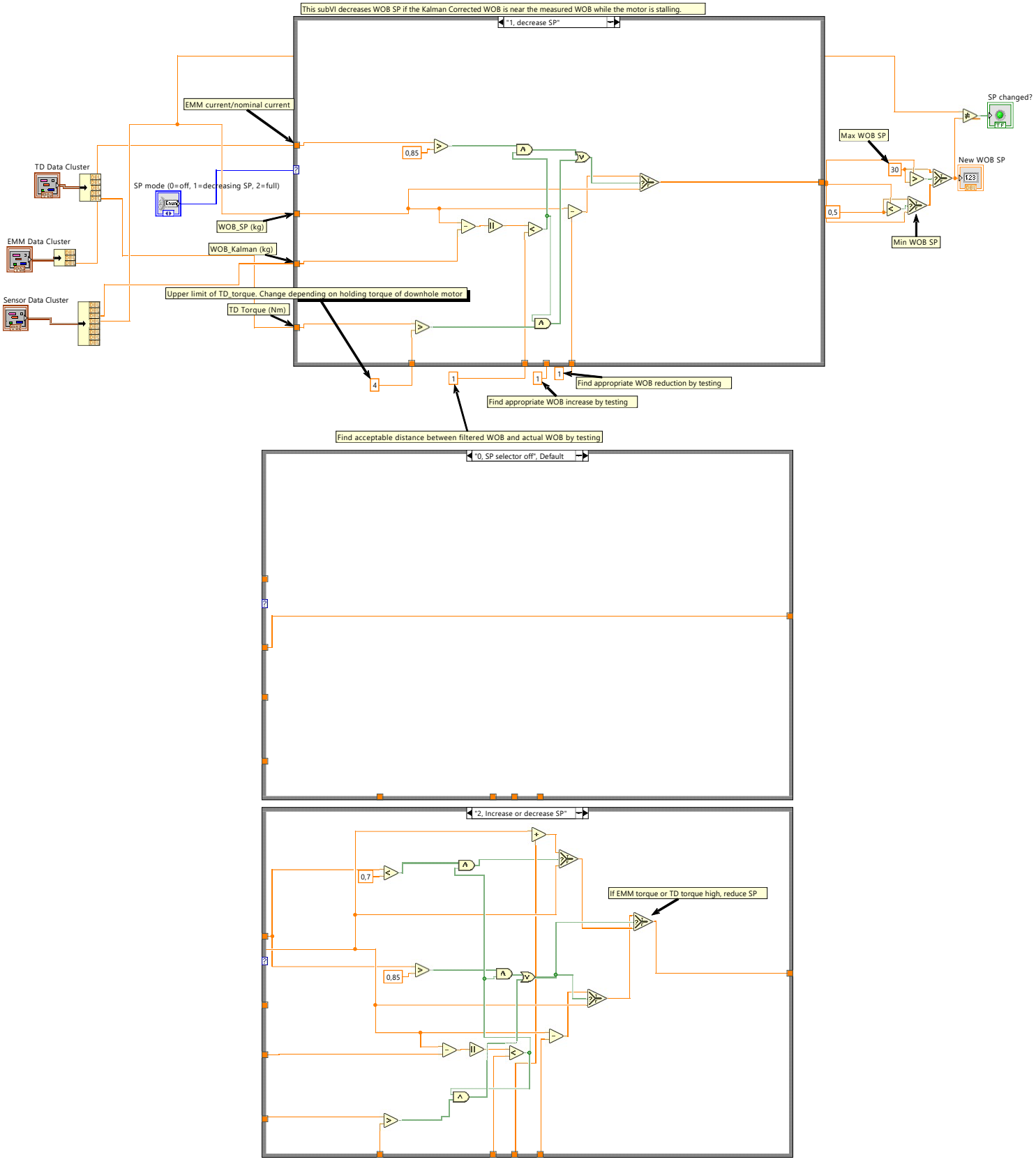
Critical_error_sub_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Sub-VIs\Hilde subVI 29mai\Critical_error_sub_Hilde.vi
Last modified on 06.06.2019 at 09.46
Printed on 19.06.2019 at 12.44



Crit_error_naming_sub_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\Crit_error_naming_sub_Hilde.vi

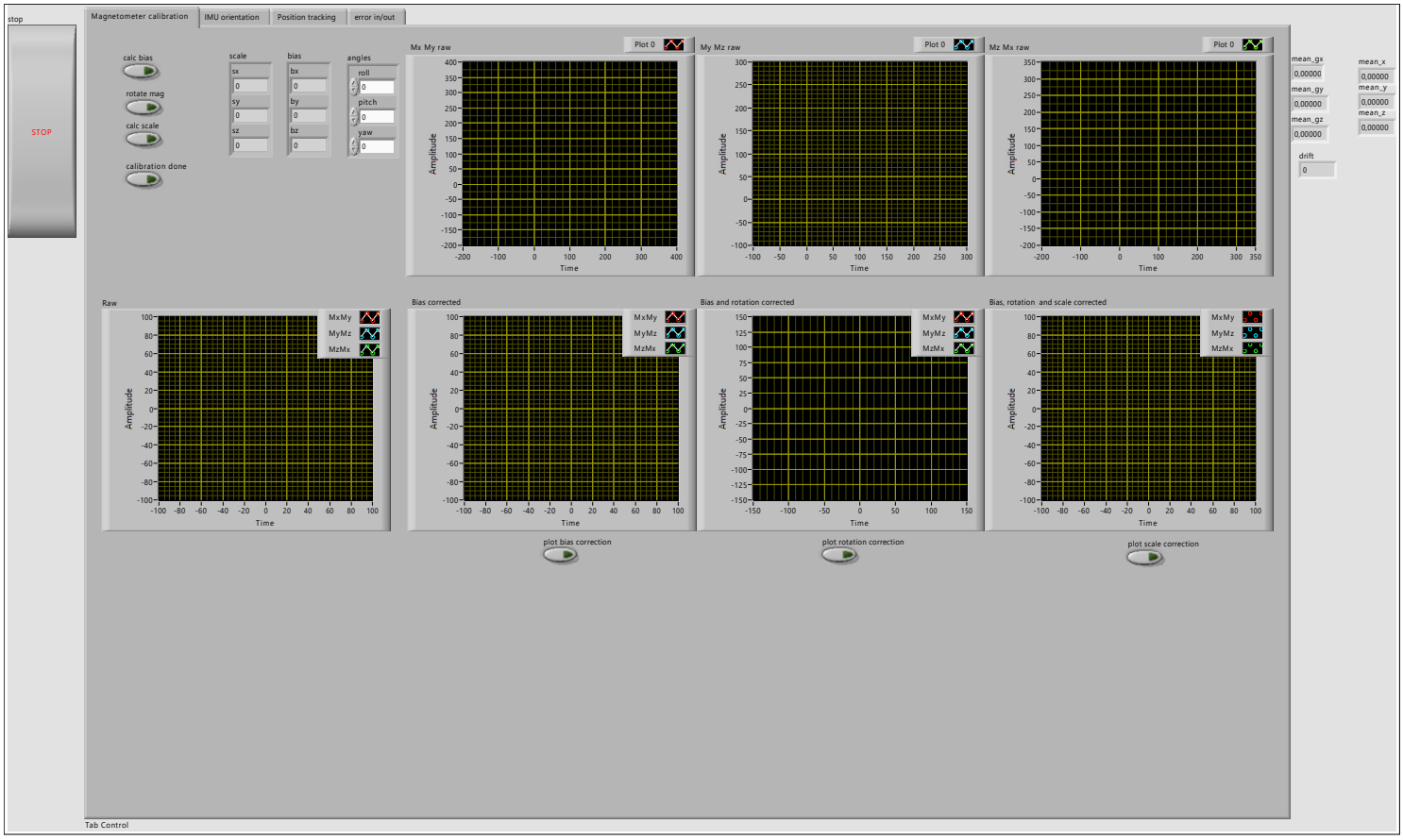
C.2.13 Setpoint selector subVI

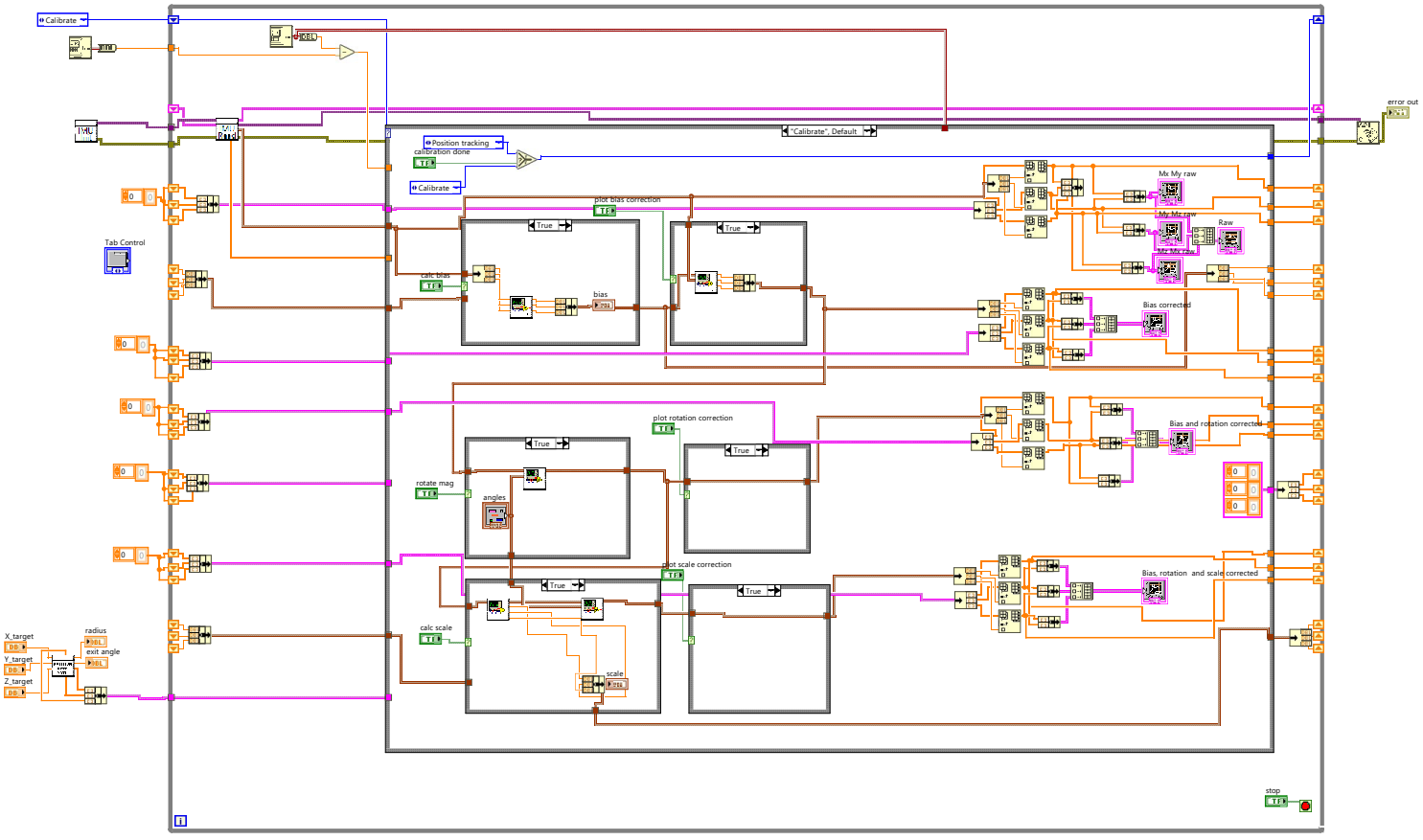
WOB_SP_selector__vertical_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Sub-VIs\Hilde subVI 29mai\WOB_SP_selector__vertical_Hilde.vi
Last modified on 04.06.2019 at 16.37
Printed on 19.06.2019 at 13.27



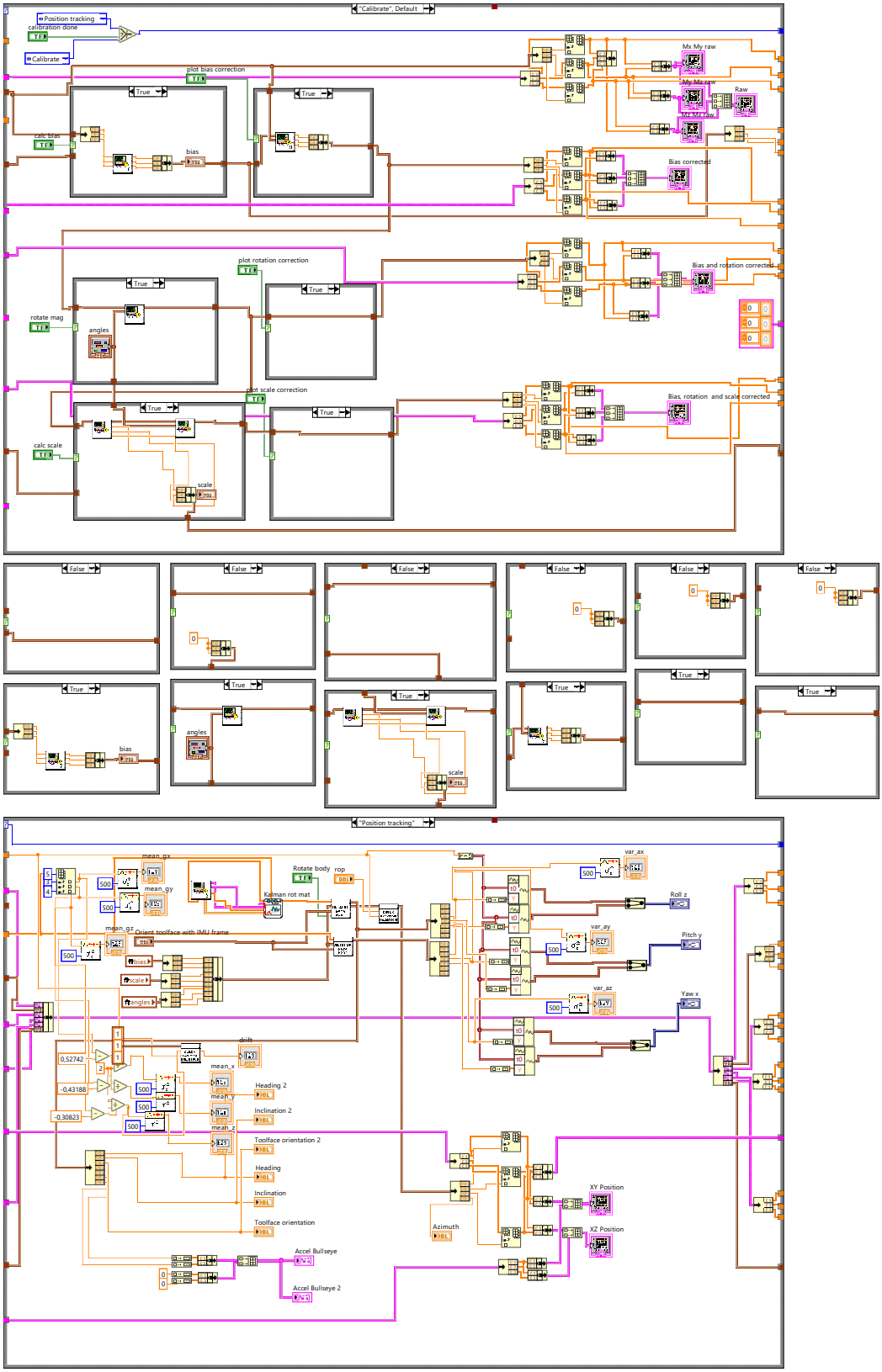
C.2.14 IMU Calibration Script

soft_iron_calibration_2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\soft_iron_calibration_2.vi
Last modified on 13.06.2019 at 09.30
Printed on 18.06.2019 at 17.42





soft_iron_calibration_2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\soft_iron_calibration_2.vi
Last modified on 13.06.2019 at 09.30
Printed on 18.06.2019 at 17.43



kalman_data_processing_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\kalman_data_processing_sub.vi

kalman_data_processing_sub - Copy.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\kalman_data_processing_sub - Copy.vi

soft_iron_calibration_2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\soft_iron_calibration_2.vi
Last modified on 13.06.2019 at 09.30
Printed on 18.06.2019 at 17.43

NI_PtbyPt.lvlib:Variance PtByPt.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\ptbypt\Probability & Statistics.llb\Variance PtByPt.vi

euler_position_equation_competition_script_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\euler_position_equation_competition_script_sub.vi

state_space_model_sensor_in_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\state_space_model_sensor_in_sub.vi

imu_init_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\imu_init_sub.vi

ref_path_init_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\ref_path_init_sub.vi

mag_rotated_back.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\mag_rotated_back.vi

mag_cal_scale.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\mag_cal_scale.vi

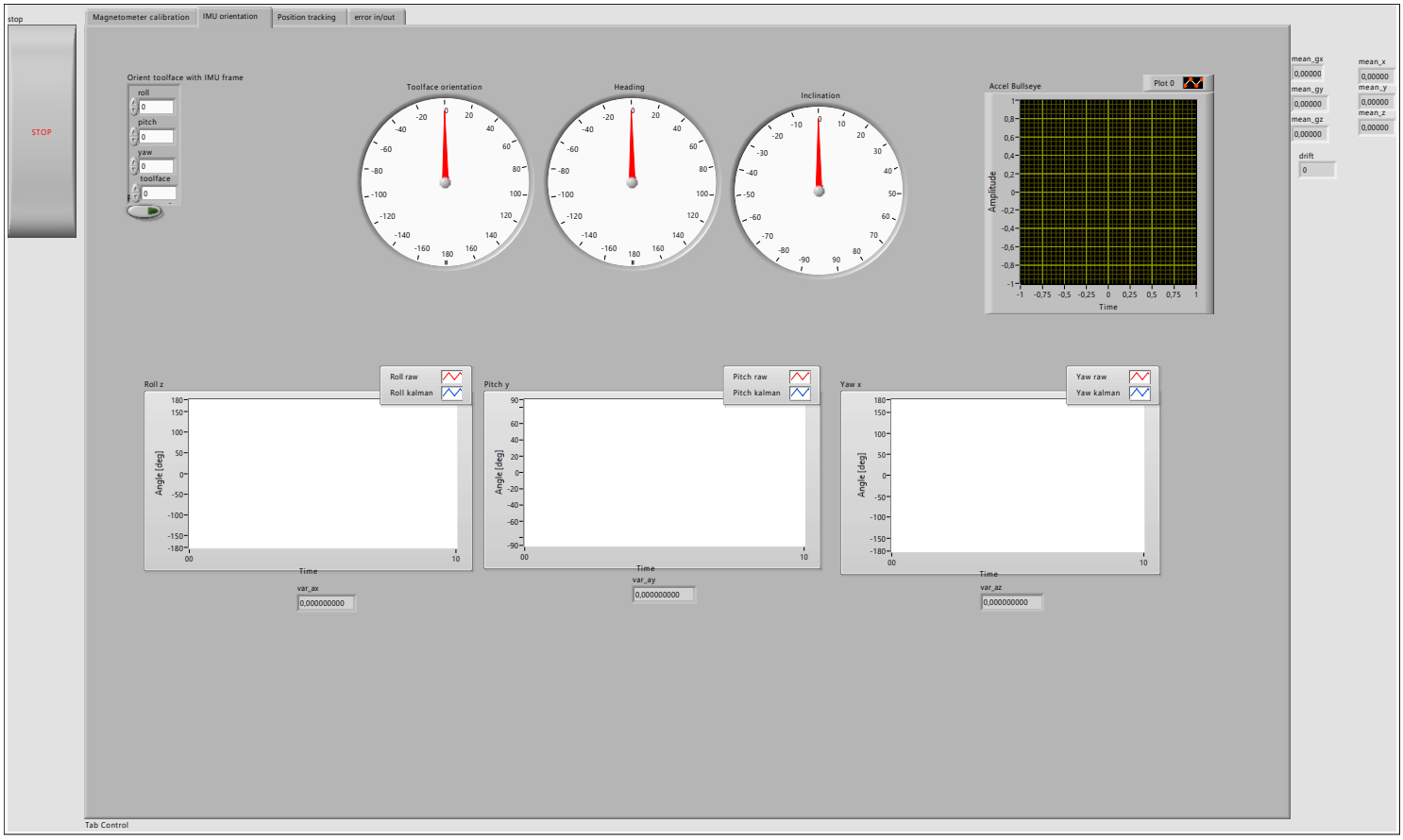
mag_cal_rotate.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\mag_cal_rotate.vi

bias_correct_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\bias_correct_sub.vi

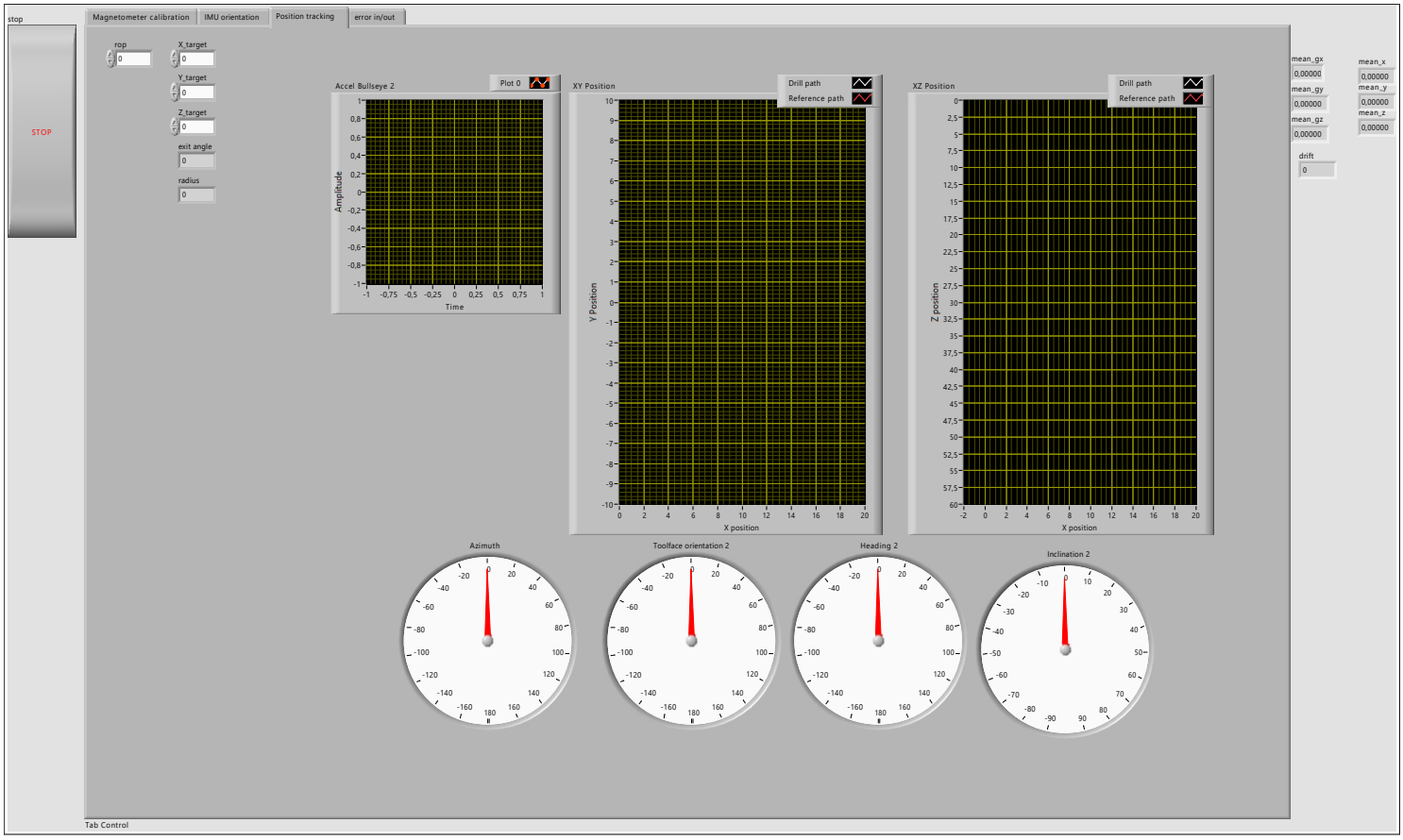
mag_cal_bias.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\mag_cal_bias.vi

imu_read_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\imu_read_sub.vi

soft_iron_calibration_2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\soft_iron_calibration_2.vi
Last modified on 13.06.2019 at 09.30
Printed on 18.06.2019 at 17.44

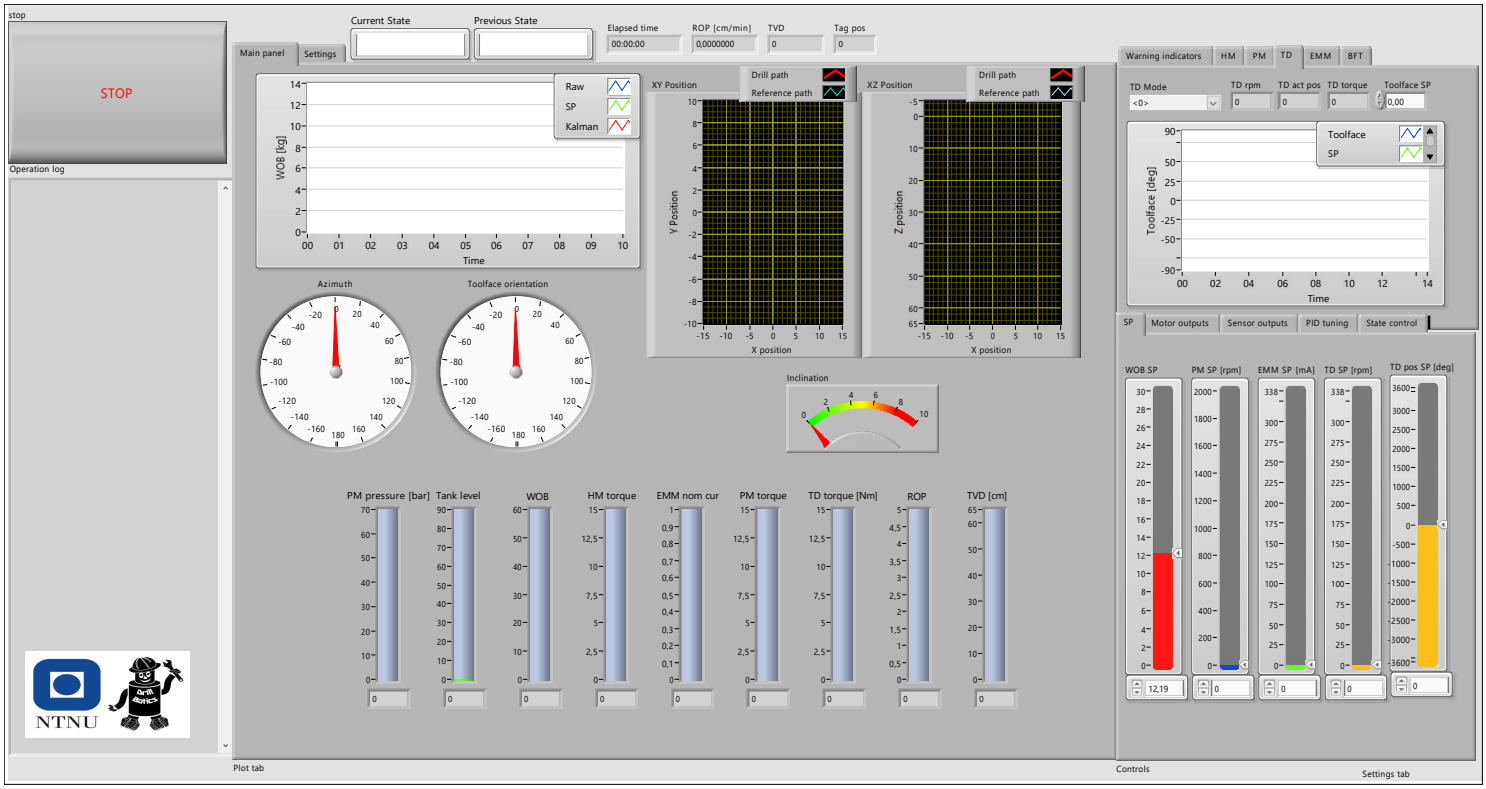


soft_iron_calibration_2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\soft_iron_calibration_2.vi
Last modified on 13.06.2019 at 09.30
Printed on 18.06.2019 at 17.45

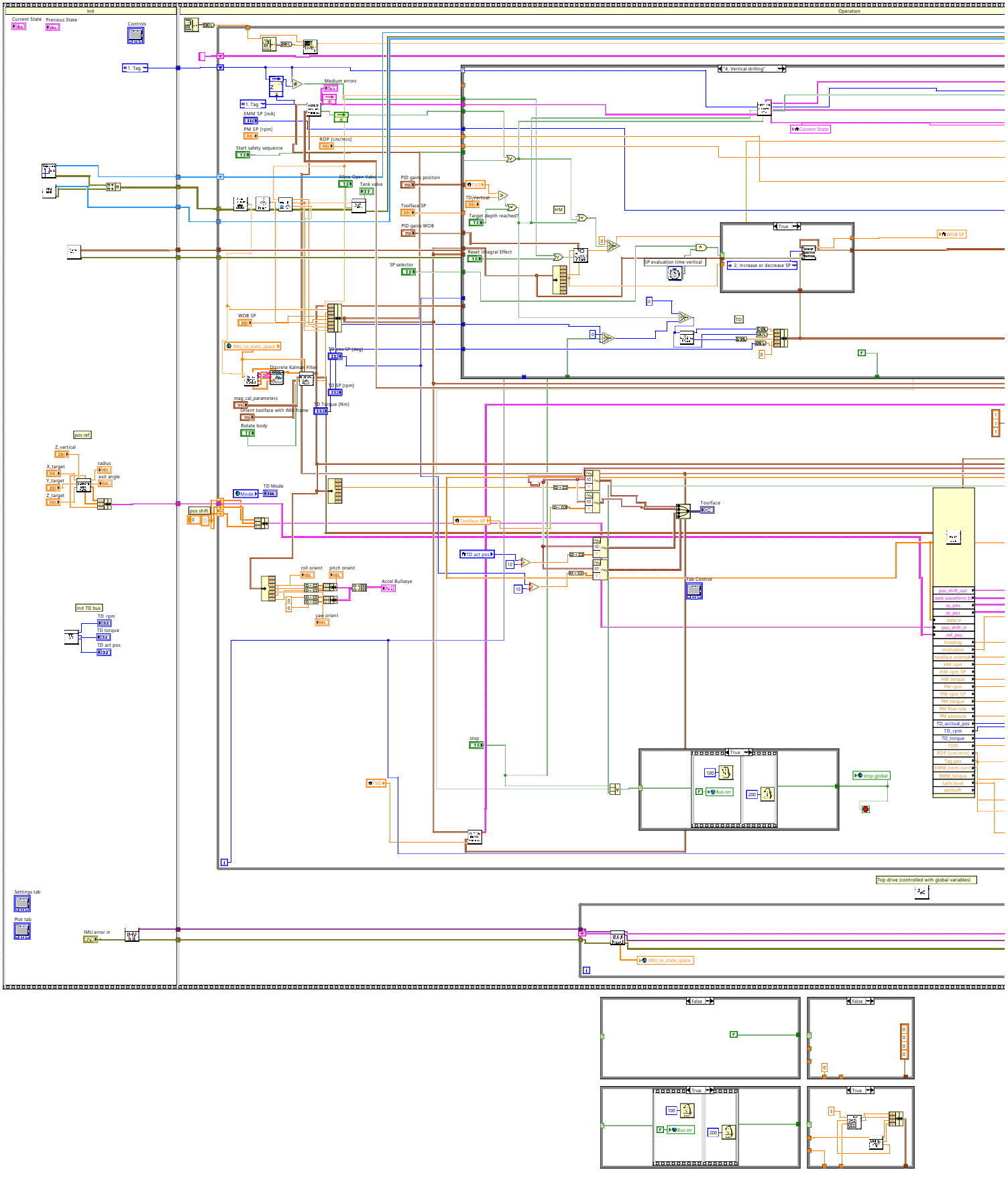


C.2.15 Competition Script

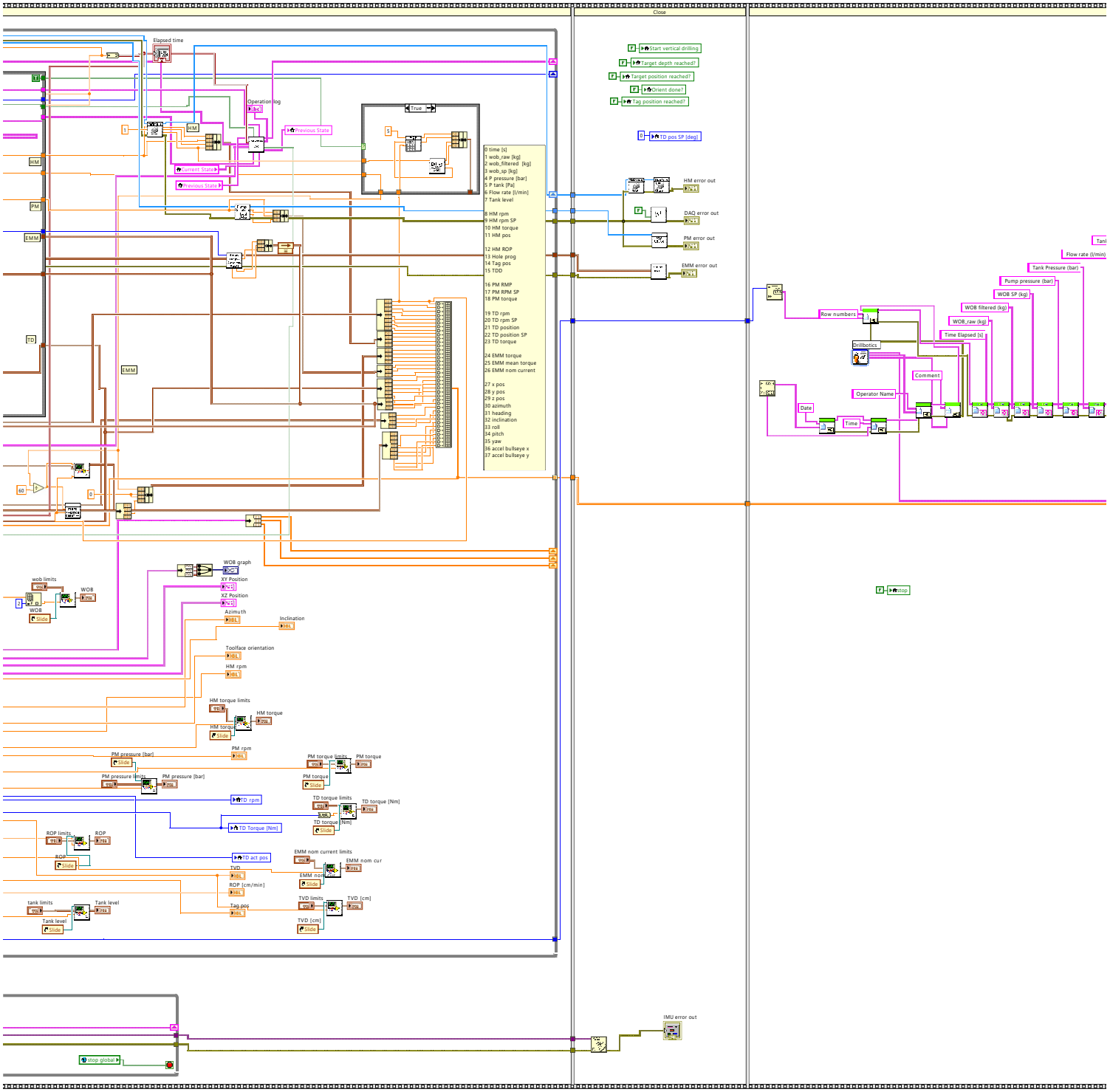
Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.47



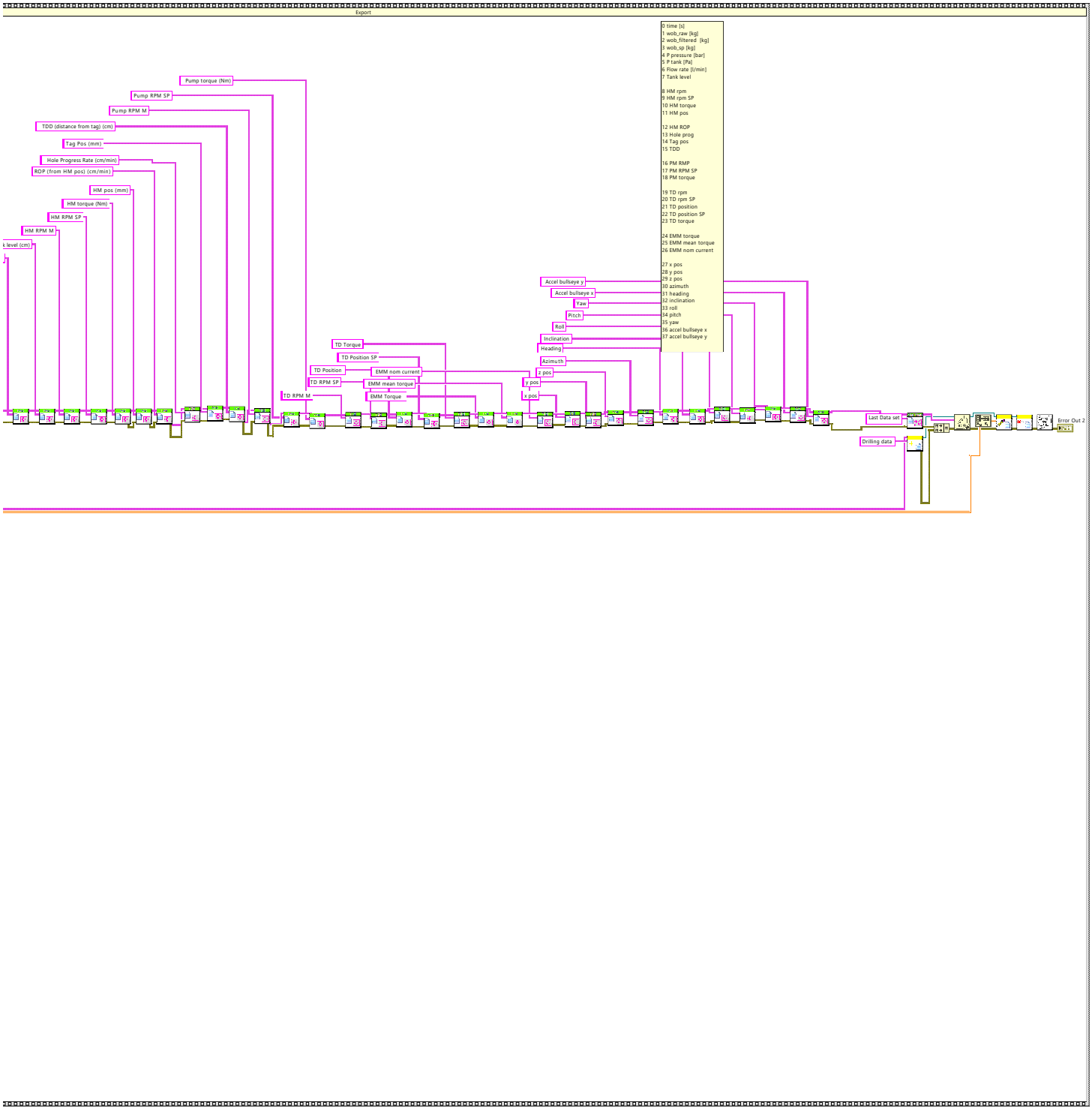
Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.47



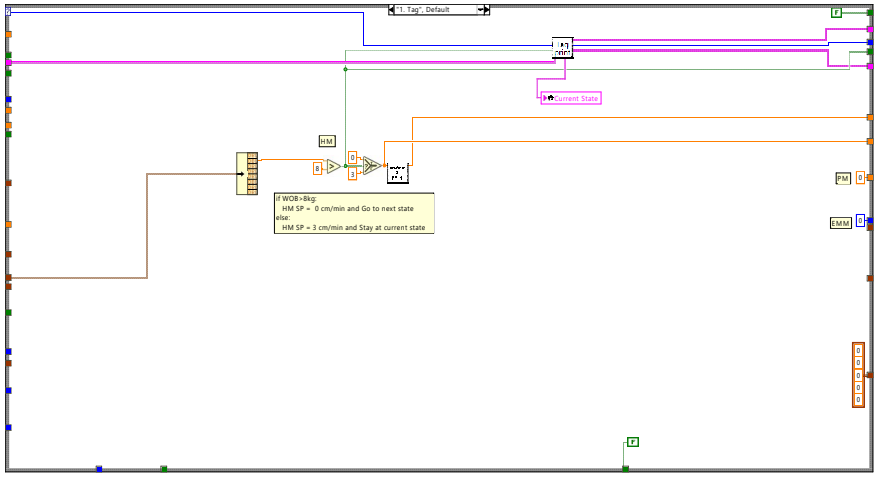
Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.48

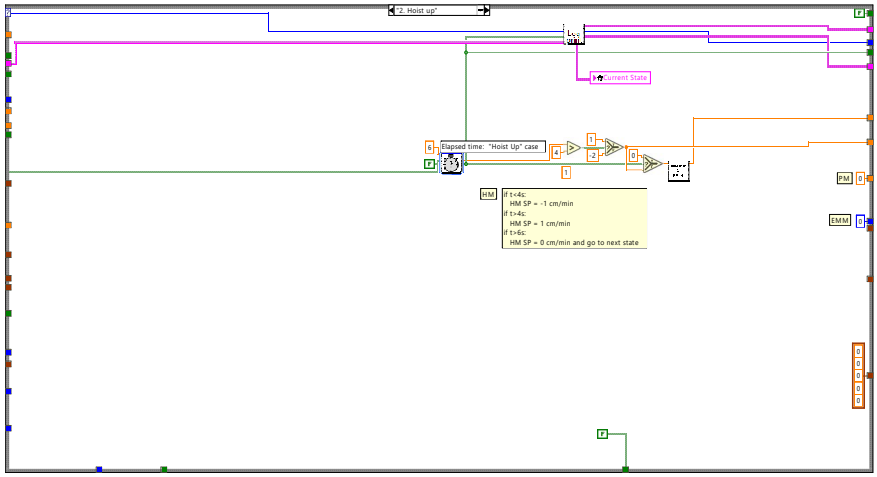


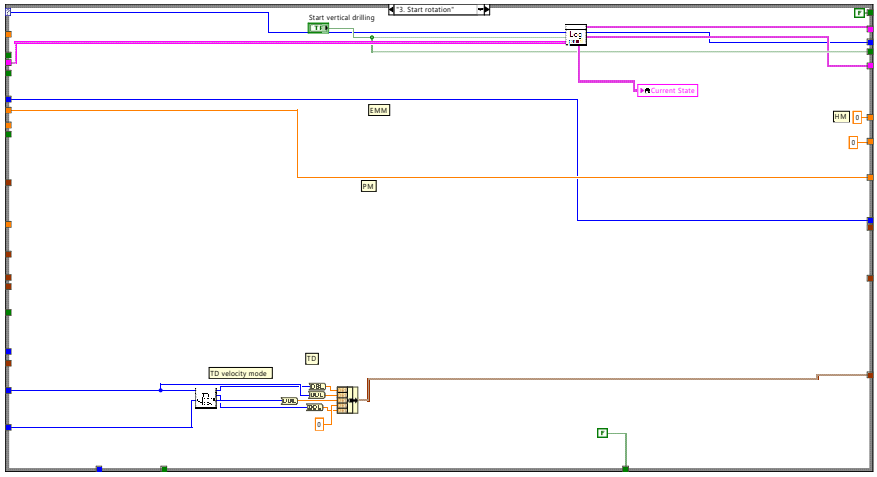
Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.48



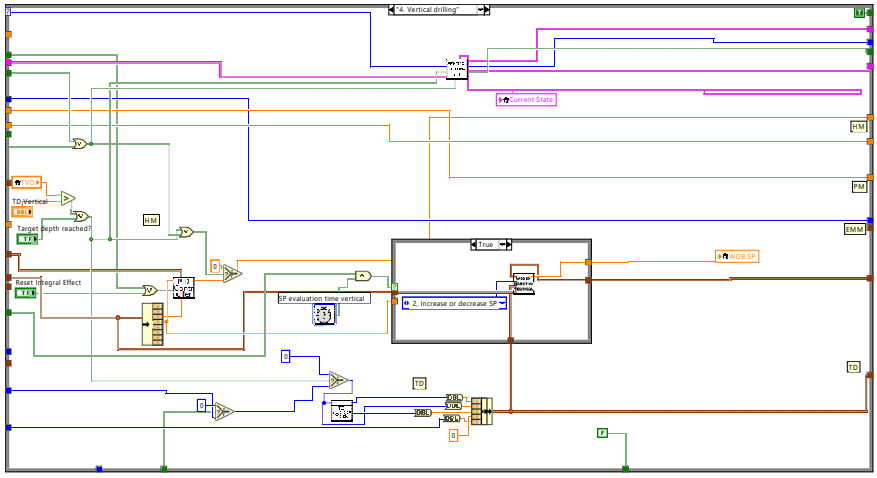
Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.48

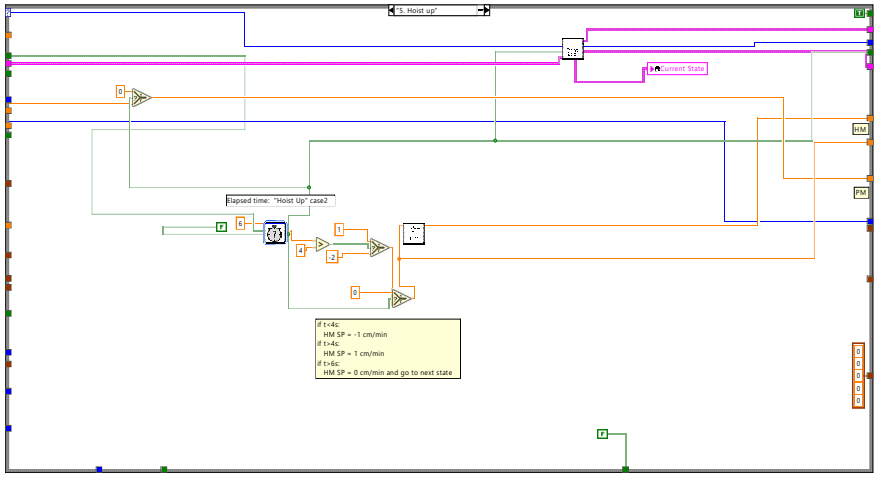


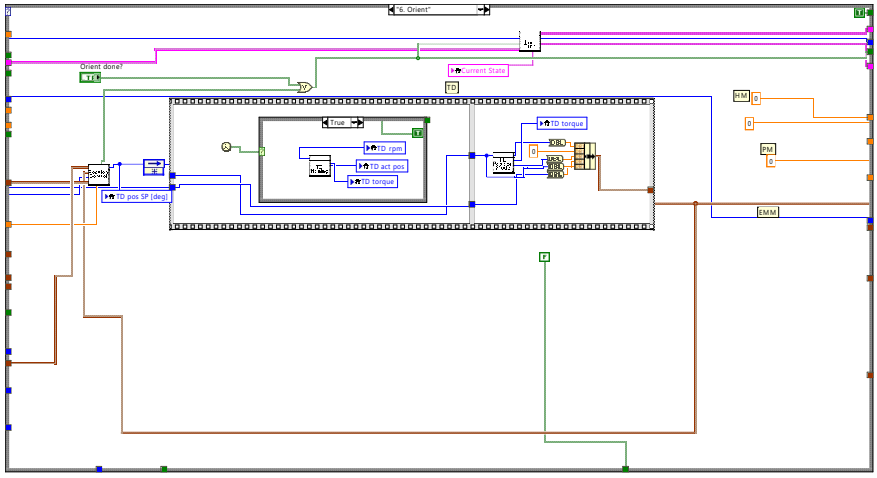




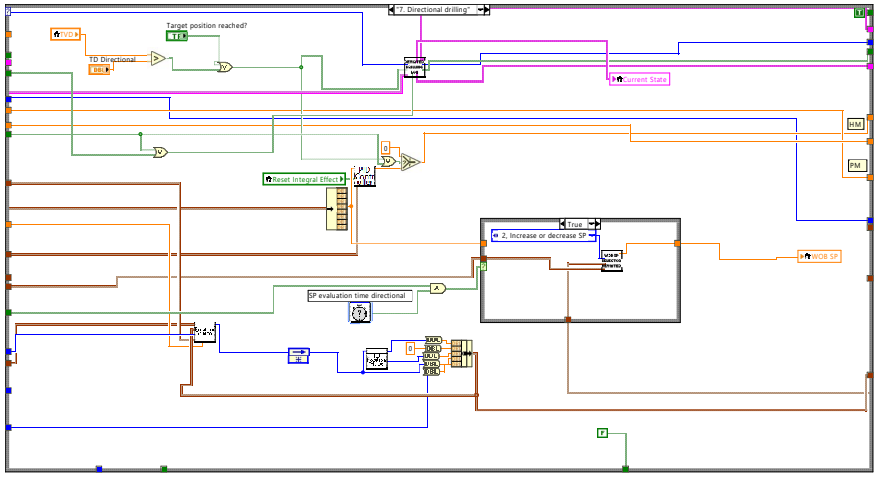
Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.48

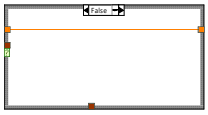


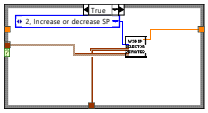


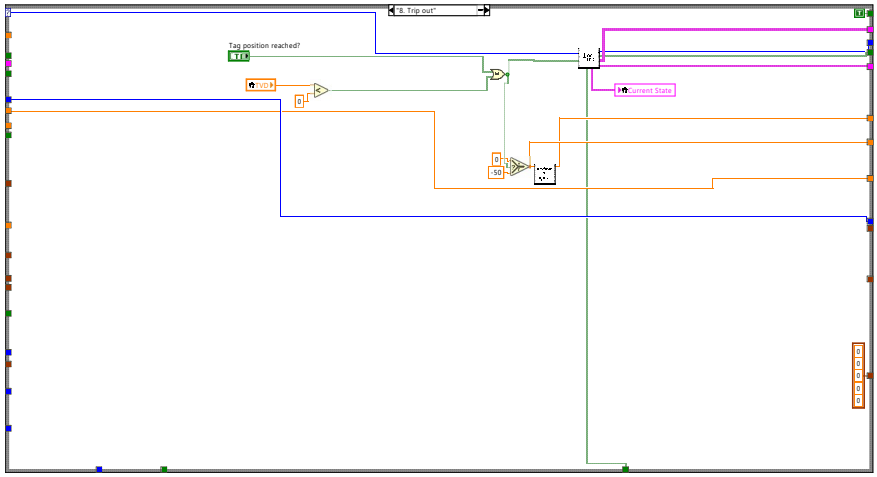


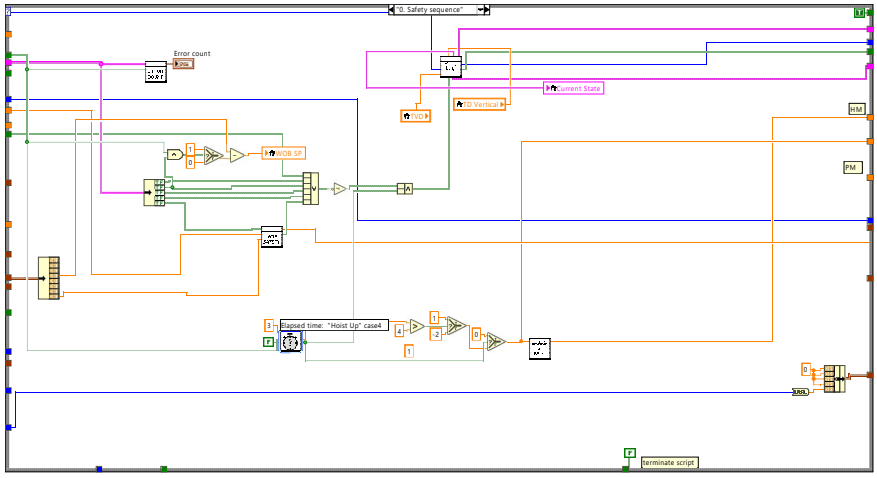
Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.48











hoist_close_Subvi.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Hoisting Motor\SubVI's\hoist_close_Subvi.vi

Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.48

Format File Property_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\Format File Property_2019.vi

Create General Data File_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\Create General Data File_2019.vi

Add Colum to String_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\Add Colum to String_2019.vi

Add End of Line to String_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\Add End of Line to String_2019.vi

MultiChannel Write_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\MultiChannel Write_2019.vi

Close Data File_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\Close Data File_2019.vi

Simple Error Handler.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Utility\error.llb\Simple Error Handler.vi

td_homing.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\td_homing.vi

start_bit_pipe_bit_log_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\start_bit_pipe_bit_log_sub.vi

WOB_SP_selector__deviated_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\WOB_SP_selector__deviated_Hilde.vi

tank_safety_sequence_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\tank_safety_sequence_sub.vi

euler_position_equation_competition_script_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\
euler_position_equation_competition_script_sub.vi

trip_out_log_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\trip_out_log_sub.vi

Maxon torque control subVi.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\Maxon torque control subVi.vi

Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.48

position_control_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\position_control_sub.vi

global_td__mode_selector.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td__mode_selector.vi

drilled_depth_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\drilled_depth_sub.vi

orient_log_subs.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\orient_log_subs.vi

HM_TCP_SubVI.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Hoisting Motor\SubVI's\HM_TCP_SubVI.vi

error_counter_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\error_counter_sub.vi

deviated_drilling_log_sub_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\deviated_drilling_log_sub_Hilde.vi

ROP_from_pos_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\ROP_from_pos_2019.vi

subElapsedTime.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\express\express execution control\ElapsedTimeBlock.llb\subElapsedTime.vi

safety_log_sub_TDD_new state.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\safety_log_sub_TDD_new state.vi

TD_position_mode_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\TD_position_mode_sub.vi

hoist_up_log_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\hoist_up_log_sub.vi

tag_log_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\tag_log_sub.vi

PID_WOB (SubVI).vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\PID_WOB (SubVI).vi

WOB_SP_selector__vertical_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\WOB_SP_selector__vertical_Hilde.vi

Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.48

vertical_drilling_log_sub_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\vertical_drilling_log_sub_Hilde.vi

TD_velocity_mode_sub_simulator.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Simulator VIs\TD_velocity_mode_sub_simulator.vi

hoist_up_2_log_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\hoist_up_2_log_sub.vi

cm_per_min_2_hoisting_RPM.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\LabVIEW Drillbotics 2018\Conversions\cm_per_min_2_hoisting_RPM.vi

tank_valve_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\tank_valve_sub.vi

Maxon end subVi.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\Maxon end subVi.vi

Operation_log_sub_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\Operation_log_sub_Hilde.vi

Critical_error_sub_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\Critical_error_sub_Hilde.vi

detect_medium_warning.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\detect_medium_warning.vi

tdd_simulation_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\tdd_simulation_sub.vi

PM_close.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\New Pump motor\subvi\PM_close.vi

PM_read_write.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\New Pump motor\subvi\PM_read_write.vi

plot_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\plot_sub.vi

indicator_color_sub_2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\indicator_color_sub_2.vi

global_td_bus_on.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td_bus_on.vi

Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.48

Tank pressure subVI.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Tank Pressure Gauge\sub VIs\Tank pressure subVI.vi

time_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\time_sub.vi

kalman_data_processing_sub_competition_script.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\
kalman_data_processing_sub_competition_script.vi

state_space_model_kalman_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\state_space_model_kalman_sub.vi

Pump pressure subVI.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Pump Pressure Gauge\SubVI's\Pump pressure subVI.vi

Load_Cell_DAQassistant_subVI.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Load Cell\SubVI's\Load_Cell_DAQassistant_subVI.vi

global_imu_raw_data.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_imu_raw_data.vi

imu_read_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\imu_read_sub.vi

global_stop_button.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_stop_button.vi

TD_main.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\kvCanVi\TD_main.vi

Maxon init subVi.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Maxon Motor\Maxon init subVi.vi

PM_init.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\New Pump motor\subvi\PM_init.vi

hoist_init_Subvi.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Manual Motor Control\Hoisting Motor\SubVI's\hoist_init_Subvi.vi

td_init.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\td_init.vi

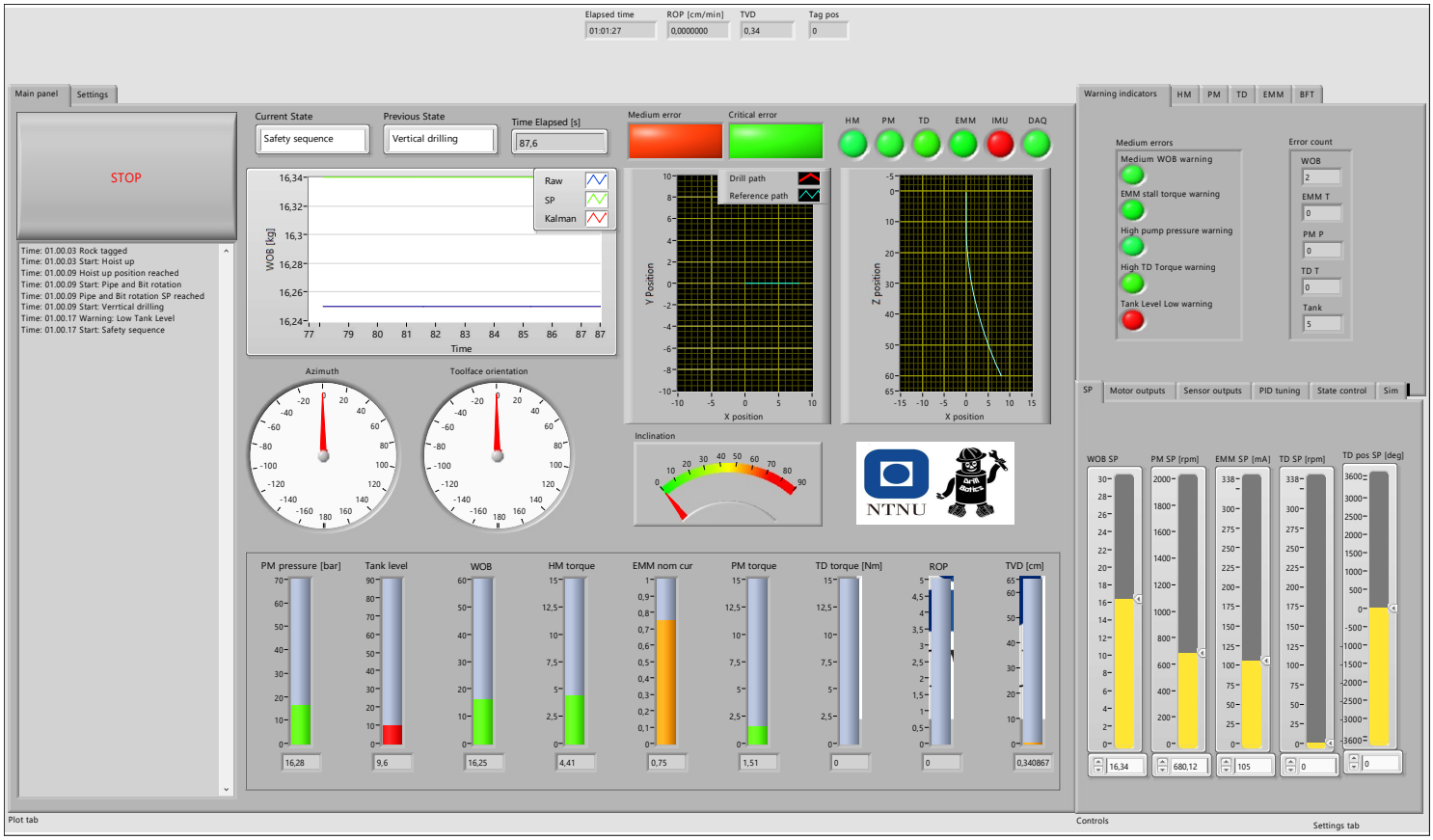
Competition_Script_final.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_final.vi
Last modified on 19.06.2019 at 09.44
Printed on 19.06.2019 at 09.48

ref_path_init_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\ref_path_init_sub.vi

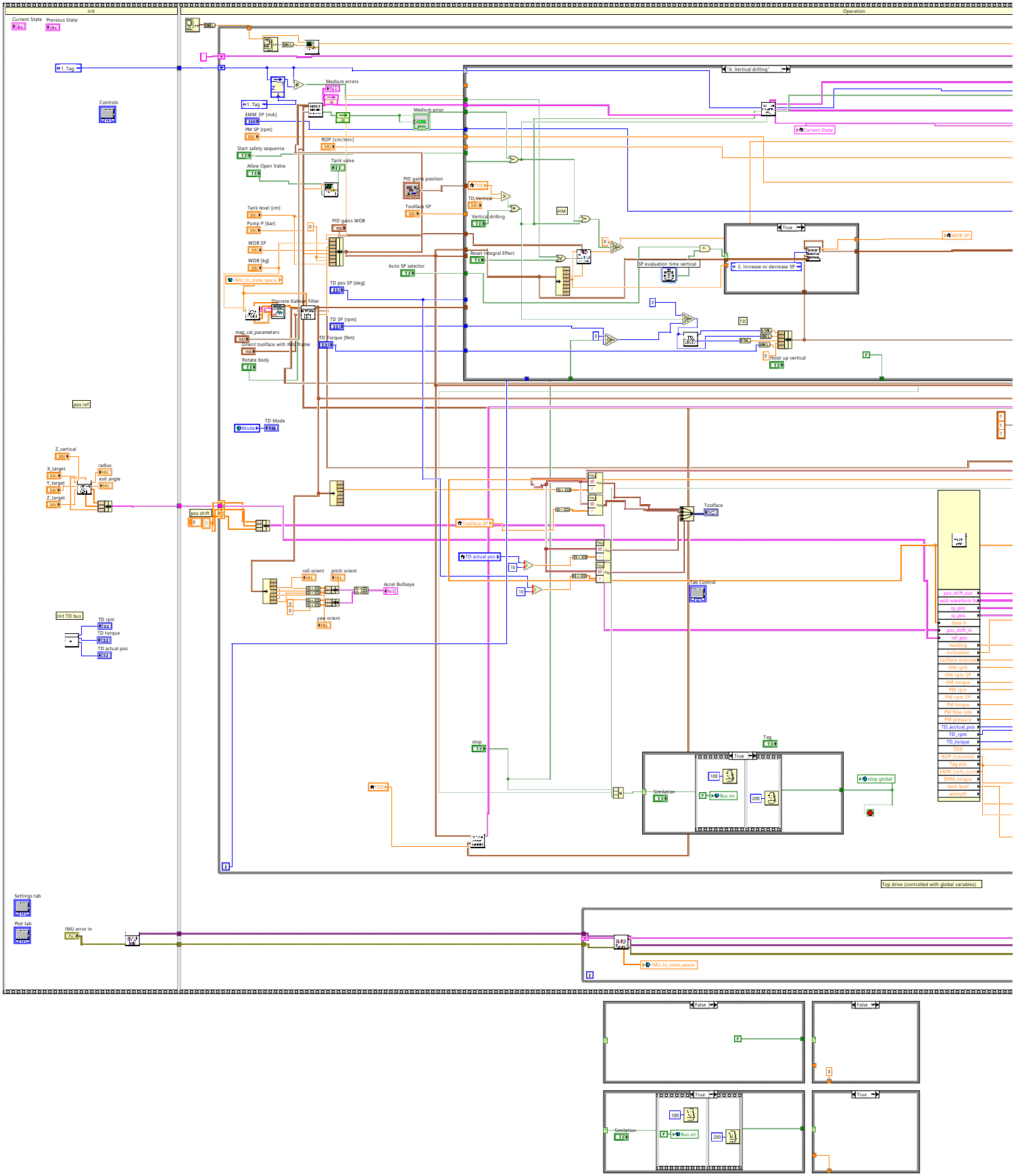
imu_init_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\imu_init_sub.vi

C.2.16 Simulation Script

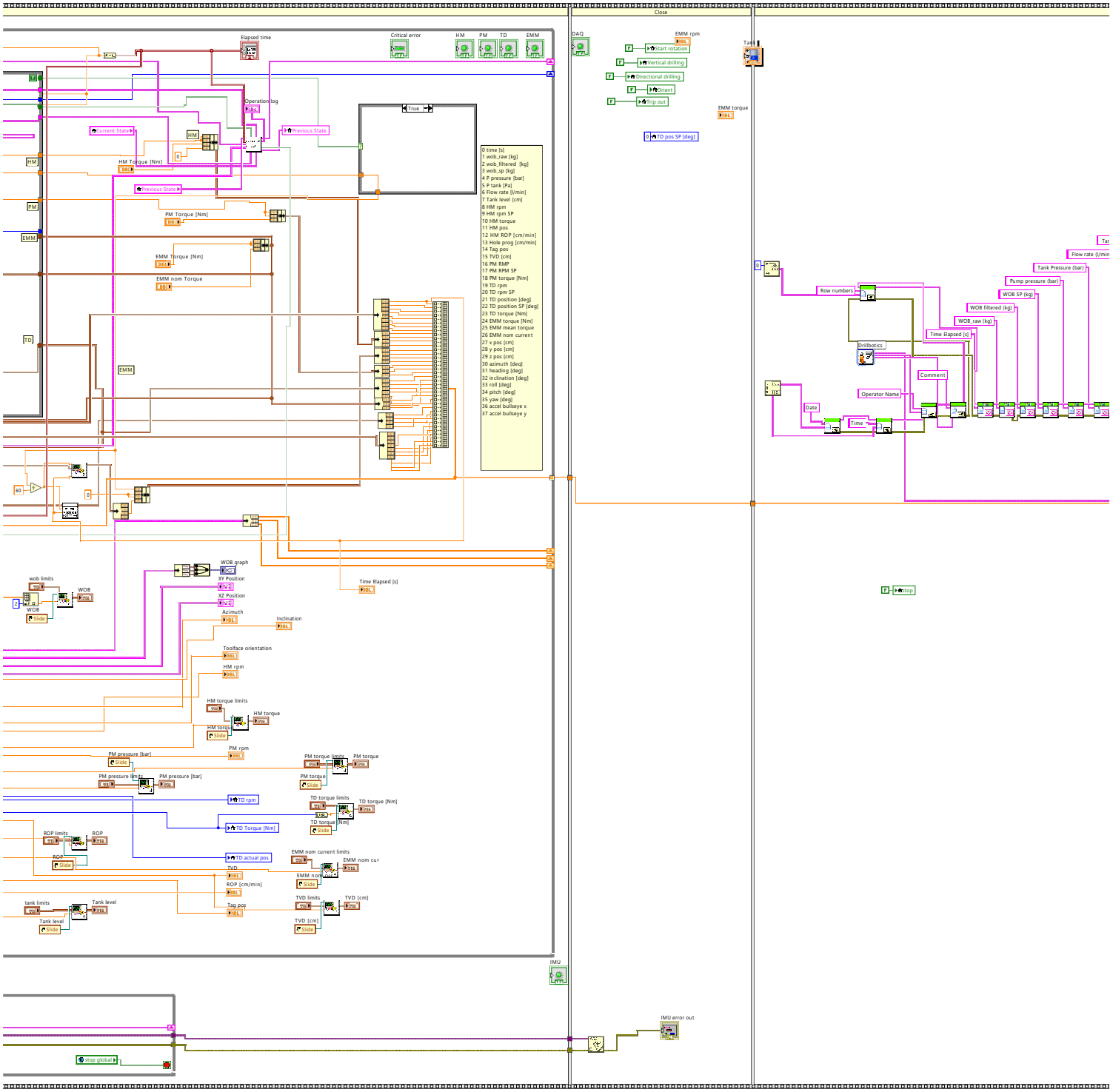
Competition_Script_simualtion_v2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_simualtion_v2.vi
Last modified on 19.06.2019 at 21.59
Printed on 20.06.2019 at 01.02



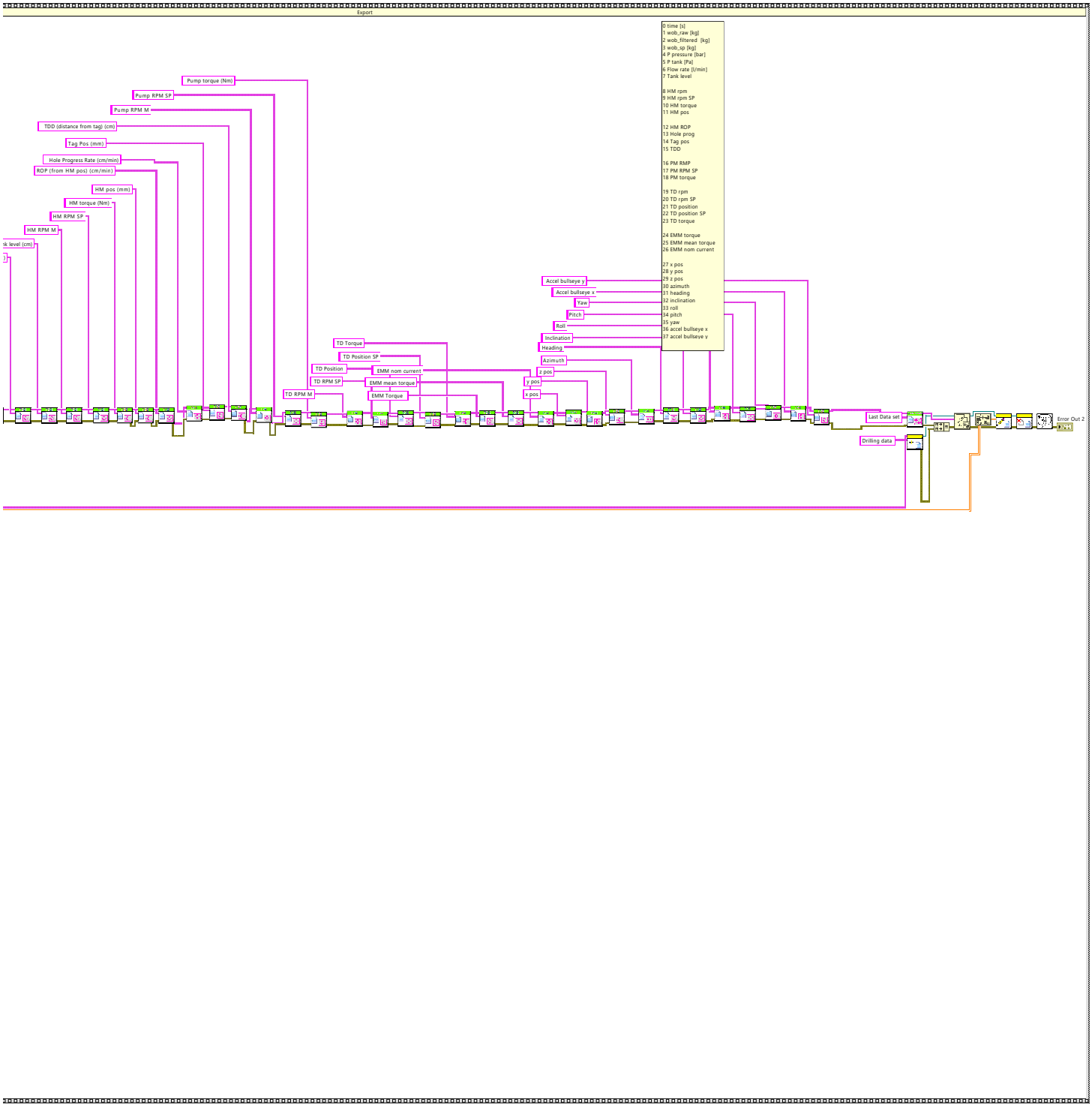
Competition_Script_simualtion_v2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_simualtion_v2.vi
Last modified on 19.06.2019 at 21.59
Printed on 20.06.2019 at 01.02



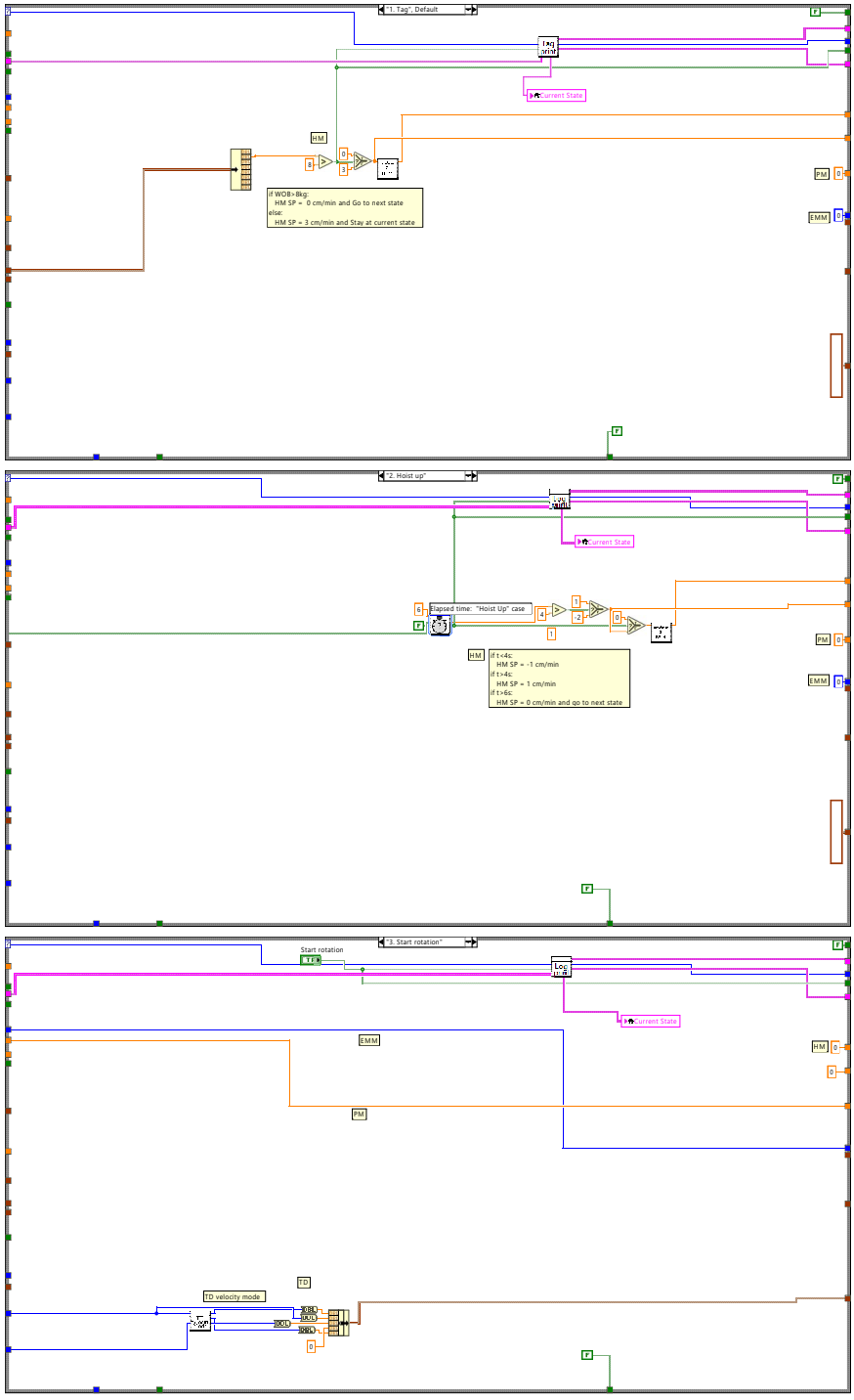
Competition_Script_simualtion_v2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_simualtion_v2.vi
Last modified on 19.06.2019 at 21.59
Printed on 20.06.2019 at 01.02



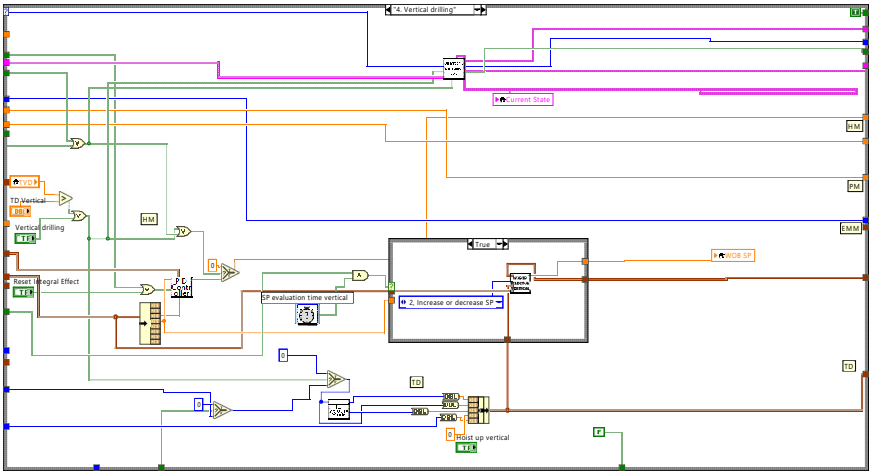
Competition_Script_simualtion_v2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_simualtion_v2.vi
Last modified on 19.06.2019 at 21.59
Printed on 20.06.2019 at 01.02

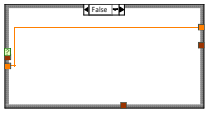


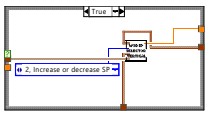
Competition_Script_simualtion_v2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_simualtion_v2.vi
Last modified on 19.06.2019 at 21.59
Printed on 20.06.2019 at 01.02

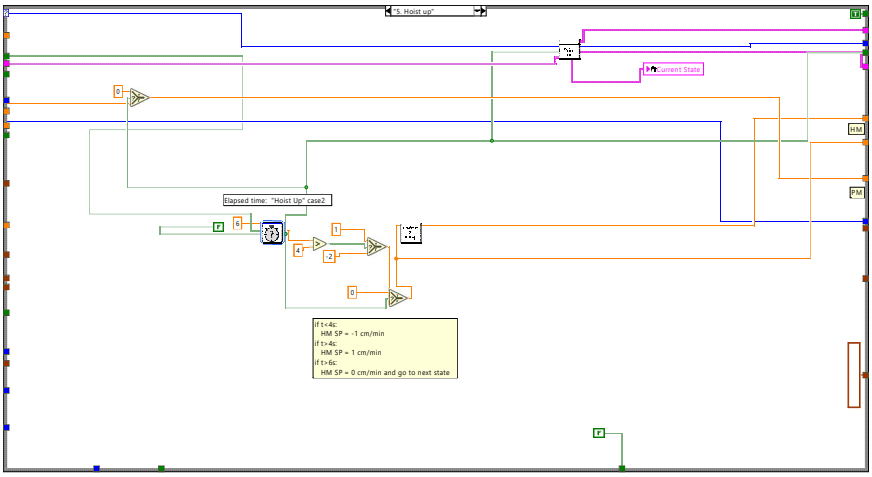


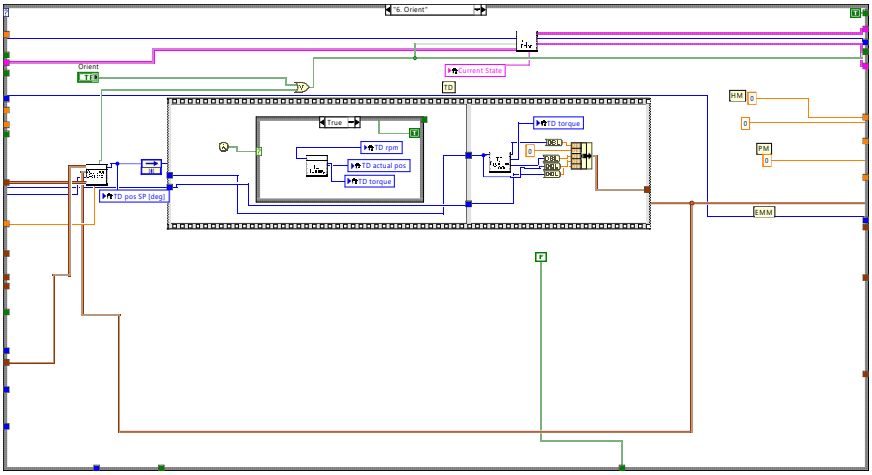
Competition_Script_simualtion_v2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_simualtion_v2.vi
Last modified on 19.06.2019 at 21.59
Printed on 20.06.2019 at 01.02

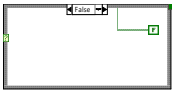


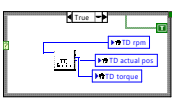




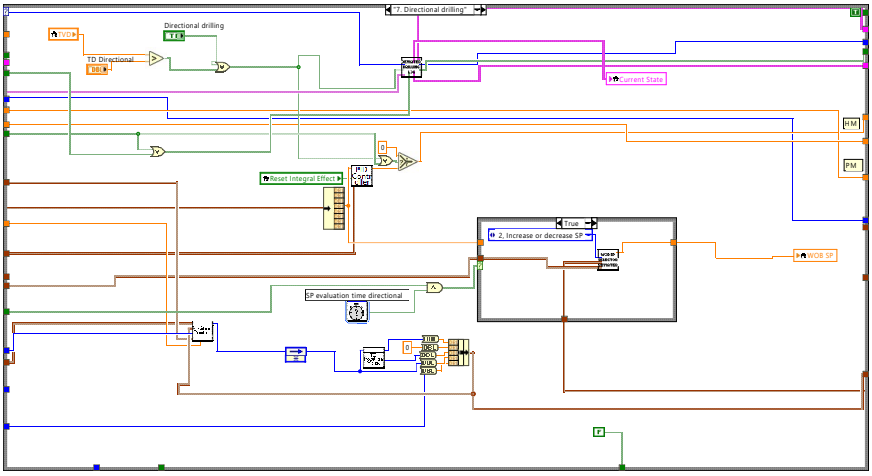


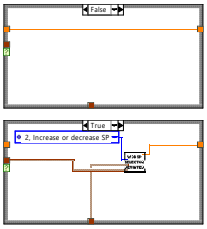


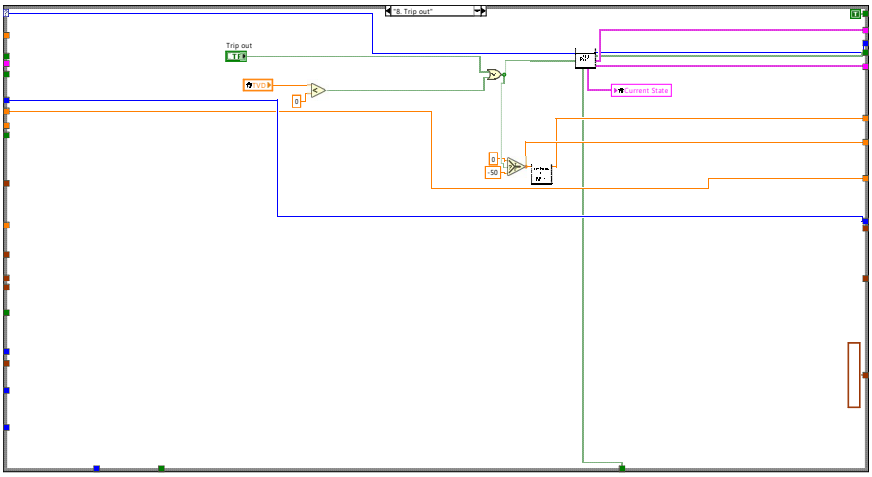


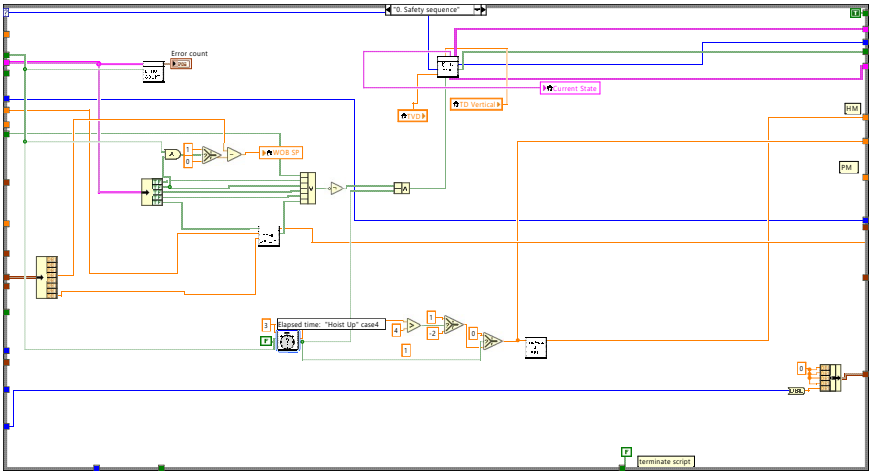


Competition_Script_simualtion_v2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Competition Script\Competition_Script_simualtion_v2.vi
Last modified on 19.06.2019 at 21.59
Printed on 20.06.2019 at 01.02









Add Colum to String_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\Add Colum to String_2019.vi

Competition_Script_simualtion_v2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_simualtion_v2.vi
Last modified on 19.06.2019 at 21.59
Printed on 20.06.2019 at 01.02

Operation_log_sub_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\Operation_log_sub_Hilde.vi

MultiChannel Write_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\MultiChannel Write_2019.vi

plot_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\plot_sub.vi

global_td_bus_on.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td_bus_on.vi

detect_medium_warning.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\detect_medium_warning.vi

Add End of Line to String_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\Add End of Line to String_2019.vi

Close Data File_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\Close Data File_2019.vi

indicator_color_sub_2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\indicator_color_sub_2.vi

Format File Property_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\Format File Property_2019.vi

tdd_simulation_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\tdd_simulation_sub.vi

Create General Data File_2019.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Data Handling\Create General Data File_2019.vi

Critical_error_sub_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\Critical_error_sub_Hilde.vi

state_space_model_kalman_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\state_space_model_kalman_sub.vi

tank_safety_sequence_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\tank_safety_sequence_sub.vi

vertical_drilling_log_sub_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\vertical_drilling_log_sub_Hilde.vi

Competition_Script_simualtion_v2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_simualtion_v2.vi
Last modified on 19.06.2019 at 21.59
Printed on 20.06.2019 at 01.02

deviated_drilling_log_sub_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\deviated_drilling_log_sub_Hilde.vi

global_td__mode_selector.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_td__mode_selector.vi

time_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\time_sub.vi

kalman_data_processing_sub_competition_script.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\
kalman_data_processing_sub_competition_script.vi

PID_WOB (SubVI).vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\PID_WOB (SubVI).vi

hoist_up_2_log_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\hoist_up_2_log_sub.vi

safety_log_sub_TDD_new state.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\safety_log_sub_TDD_new state.vi

WOB_SP_selector__vertical_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\WOB_SP_selector__vertical_Hilde.vi

subElapsedTime.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\express\express execution control\ElapsedTimeBlock.llb\subElapsedTime.vi

tag_log_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\tag_log_sub.vi

cm_per_min_2_hoisting_RPM.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\LabVIEW Drillbotics 2018\Conversions\cm_per_min_2_hoisting_RPM.vi

TD_position_mode_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\TD_position_mode_sub.vi

error_counter_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\error_counter_sub.vi

WOB_SP_selector__deviated_Hilde.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Hilde subVI 29mai\WOB_SP_selector__deviated_Hilde.vi

Competition_Script_simualtion_v2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_simualtion_v2.vi
Last modified on 19.06.2019 at 21.59
Printed on 20.06.2019 at 01.02

td_homing.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\td_homing.vi

TD_velocity_mode_sub_simulator.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Simulator VIs\TD_velocity_mode_sub_simulator.vi

start_bit_pipe_bit_log_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\start_bit_pipe_bit_log_sub.vi

trip_out_log_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\trip_out_log_sub.vi

hoist_up_log_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\hoist_up_log_sub.vi

orient_log_subs.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\Live_log_print_subs\orient_log_subs.vi

position_control_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\position_control_sub.vi

euler_position_equation_competition_script_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\
euler_position_equation_competition_script_sub.vi

Simple Error Handler.vi
C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\Utility\error.llb\Simple Error Handler.vi

global_stop_button.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_stop_button.vi

tank_filling_simulation_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\tank_filling_simulation_sub.vi

imu_read_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\imu_read_sub.vi

global_imu_raw_data.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\global_imu_raw_data.vi

imu_init_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\imu_init_sub.vi

Competition_Script_simualtion_v2.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\
Autonomous Control\Competition Script\Competition_Script_simualtion_v2.vi
Last modified on 19.06.2019 at 21.59
Printed on 20.06.2019 at 01.02

td_init.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Autonomous Control\Sub-VIs\td_init.vi

ref_path_init_sub.vi
C:\Users\Bruker\Desktop\Drillbotics backup 090619\Drillbotics backup\Drillbotics 2019\Individual Sensor Control\Down Hole Sensors\IMU_mads\subVi\ref_path_init_sub.vi