




Universitetet
i Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: Petroleum Technology, Drilling	Spring / Autumn semester, 2020 Open/ Confidential
Author: Magomed Adamovitsj Khadisov	 (signature of author)
Programme coordinator: Anita Malde Supervisor(s): Dan Sui	
Title of master's thesis: Directional Drilling: Trajectory Design and Position Uncertainty Study for a Laboratory Drilling Rig.	
Credits (ECTS): 30	
Keywords: Directional Drilling Trajectory Optimization Trajectory Design Position Uncertainty Calculations ISCWSA Error Model Drilling Automation Drillbotics	Number of pages: 94 + supplemental material/other: 19 Stavanger, 19.june 2020 date/year

Abstract

This thesis investigates how trajectory optimisation and position uncertainty calculations can be performed on a laboratory drilling rig.

The improvements made are to be used on the drilling rig designed and manufactured by UiS Drillbotics team, for the international Drillbotics competition. This year's competition presented a new set of challenges, including a pre-planned trajectory, hitting target points before exiting the well and an integrated error model. The requirements are presented in detail in the guidelines provided by Drilling Systems Automation Technical Section (DSATS) committee.

The models presented in this thesis are intended to advance the current system, where automated optimisation of trajectories and position uncertainty calculations can easily be performed and visualised. Using Bézier curves for trajectory optimisation and the ISCWSA error model, these calculations are presented and tested for both field cases, and generated trajectories with the dimensions of the laboratory drilling rig.

The solutions presented offers a possibility to intuitively visualise the results to the driller and observers. The trajectory optimisation is a process conducted before any drilling is initiated, and imported to the main GUI for comparison between planned trajectory and drilled trajectory. The position uncertainty calculations are flexible, and can be specifically configured for different sensor packages and confidence levels. The uncertainty calculations are plotted together with the trajectory, which enables the driller and system to identify deviations and reconsider the trajectory if necessary.

Acknowledgments

I would like to thank Professor Dan Sui from the University of Stavanger, for her supervision, guidance and support throughout this semester and my time on this project.

The past three years on this project have been a great experience, both academically and socially. I leave this project with hands-on experience, new knowledge and experience from working in a multidisciplinary team. A special thanks to all old and new UiS Drillbotics members throughout the years.

Lastly, I would like to thank Andrzej Tunkiel for taking the time to join the team meetings and discussions and the Department of Energy and Petroleum Engineering (IEP) for their support.

Contents

Abstract	ii
Acknowledgements	ii
List of Symbols and Abbreviations	x
1 Background	1
1.1 UiS Drillbotics	1
1.2 Guidelines	2
1.2.1 Competition Rock	2
1.2.2 Bit	3
1.2.3 Drill pipe	4
1.2.4 Automated Drilling	4
1.2.5 Data Visualization	4
2 Laboratory Scale Drilling Rig	6
2.1 Rotational System	6
2.2 Hoisting System	7
2.3 Circulation System	8
2.4 Bottom Hole Assembly	8
2.5 Rig Sensors	10
3 Directional Drilling	13
3.1 Application of Directional Drilling	14
3.2 Types of Well Trajectories	15
3.2.1 Vertical	15
3.2.2 J-shape	15
3.2.3 S-shape	15
3.2.4 Horizontal Wells	16
3.3 Survey Calculations	16
3.3.1 Tangential Method	16

3.3.2	Average Angle Method	17
3.3.3	Balanced Tangential Method	18
3.3.4	Radius of Curvature Method	18
3.3.5	Minimum Curvature Method	19
4	Error Models and Uncertainties	22
4.1	Sensor Errors	22
4.1.1	Bias Error	23
4.1.2	Scale Factor-Error	23
4.2	ISCWSA Model	24
4.2.1	Assumptions and Limitations of the Model	25
4.2.2	Details of the Mathematical Framework	25
4.2.2.1	Notations Used	26
4.2.3	Evaluating Position Uncertainty	26
4.3	Weighting Functions	28
4.3.1	Weighting Functions for Sensor	28
4.4	Summation of Uncertainty Terms and Propagation Modes	30
4.5	Ellipse and Ellipsoid of Uncertainty	31
4.5.1	2D Ellipse of Uncertainty	33
4.5.2	3D Ellipsoid of Uncertainty	35
5	Case Study: Position Uncertainty - Field Data	36
5.1	Field Data	36
5.2	Confirmation of Computational Method	37
5.2.1	Covariance matrix with NEV-axes	37
5.2.2	Covariance matrix with HLA-axes	40
5.3	Ellipsoid of Uncertainty	41
5.3.1	Discussion: Uncertainty Dimensions	50
6	Well Trajectory Design	52
6.1	Bézier Method	52
6.2	Generating Trajectories	54
6.2.1	Assumptions	54
6.2.2	Trajectory Examples	55
7	Case Study: Trajectory Optimisation and Position Uncertainty for Laboratory Drilling Rig	62
7.1	Assumptions and Limitations	62
7.1.1	Competition Rock and Guidelines	62
7.1.2	Mechanical Limitations	65
7.2	Trajectories: Competition Rock	65

7.2.1	Definition of Trajectory Parameters	66
7.2.2	Simple Trajectories	66
7.2.3	Complex Trajectories	70
7.3	Implementation of Trajectory Optimization	77
7.4	Position Uncertainty Calculations: Generated Wells	81
7.5	Considering Weighting Functions	81
7.5.1	Accelerometer Bias Error	81
7.5.2	Accelerometer Scale-Factor Error	82
7.5.3	Magnetometer Bias Error	82
7.5.4	Magnetometer Scale-Factor Error	83
7.5.5	Limitations	84
7.6	Magnetic Variables	84
7.7	Trajectory	84
7.7.1	Data Formatting	85
7.8	Results and Discussion	85
8	Conclusion and Recommendations	92
8.1	Conclusion	92
8.2	Recommendations	93
A	List of Error Sources and Weighting Functions	i
B	Code: Trajectory Optimization	iii
C	Code: Well Trajectory By Bzier	vi
D	Survey Data	xi
E	Code: Position Uncertainty Calculations	xii

List of Figures

1.1	The provided Baker Hughes bit [21].	3
2.1	Pneumatic system overview [29].	7
2.2	3D render of the new BHA (J.E Karlsen 2020).	8
2.3	Top section, covering the pneumatic motor (J.E Karlsen 2020).	9
2.4	Bend section, which will cover the mechanical joint of the BHA (J.E Karlsen 2020).	9
2.5	Bottom section, a stabilizer positioned below the bend (J.E Karlsen 2020).	9
2.6	Full assembly with and without the exterior.	10
2.7	Rig sensors [29].	11
2.8	Inclination and azimuth given in degrees.	12
3.1	Different configurations of multilateral wells [2].	15
3.2	Types of Well Trajectories [15].	16
3.3	Left: Average angle method, Right: Tangential angle method [16].	18
3.4	Radius of curvature method [16].	19
3.5	Minimum curvature method [16].	20
3.6	Comparing the accuracy of the different methods presented [7].	21
4.1	Bias- and scalar factor error visualised [40].	23
4.2	Definition of axes and toolface angle [4].	25
4.3	Normal Distribution Curve [5].	33
4.4	Overview of an ellipse, highlighting the axes [39].	34
5.1	ISCWSA Test #1 Well	37
5.2	Last 8 rows with matrix elements from excel [24].	38
5.3	Last 8 rows with matrix elements from MATLAB	38
5.4	Difference in the last 8 rows.	39
5.5	Matrix Elements and MD from excel spreadsheet [24].	39
5.6	Matrix Elements and MD from MATLAB.	40
5.7	Last 8 rows from excel (HLA-axes) [24].	40

5.8	Last 8 rows from MATLAB (HLA-axes).	41
5.9	Difference in the last 8 rows (HLA-axes).	41
5.10	Arranging the 1x6 array into a 3x3 covariance matrix in MATLAB.	42
5.11	Ellipsoid of uncertainty at 1-standard deviation, $k = 1.8779$	43
5.12	Semi-major and semi-minor axis with respect to MD, $k = 1.8779$ scaled for 3x3 matrix.	44
5.13	Semi-major and semi-minor axis with respect to MD, $k = 1.5151$ scaled for 2x2 matrix.	45
5.14	Trajectory plotted in N/E with ellipses of uncertainty in 2D, $k = 1.5151$	45
5.15	From 2x2 covariance matrix, $k = 1.5151$. Ellipses oriented with respect to eigenvectors.	46
5.16	Showing that both ellipses are equal, $k = 1.5151$ for 2x2 covariance matrix.	46
5.17	Based on 3x3 covariance matrix, $k = 2.7954$	47
5.18	Semi-major and semi-minor axis, $k = 2.7954$	48
5.19	Ellipsoid plotted for every survey point, $k = 2.7954$	49
5.20	Average Depth of Crude Oil and Natural Gas Wells [1].	50
6.1	Third order Bézier curves with varying C_S and fixed C_E . Attractor C_{S4} is not shown for B_4 [34].	53
6.2	3D-trajectory with d_s and d_e as 201 and 251.	55
6.3	Cross sections of the same trajectory.	56
6.4	Table with data from the survey stations.	56
6.6	Trajectory (c) from a different viewpoint.	57
6.5	Trajectories with the same initial and finale conditions as previously defined, this time with varying d_S and d_E	58
6.7	Trajectory (d) from a different viewpoint.	59
6.8	Same trajectory as in fig 6.5, but with d_S and d_E lower than calculated values.	60
7.1	Well profile using 7° whipstock in cement [29].	63
7.2	A simple representation of a well after kick-off point with fixed angle.	64
7.3	The simplest form of trajectory, with no change in azimuth and a small displacement along the east-axis.	66
7.4	$d_S = d_E = 120$	67
7.5	Same as figure 7.4, but with axes limits equal to that of the competition rock.	67
7.6	V/E and V/N plotted with survey points, $d_S = d_E = 120$	68
7.7	$d_S = d_E = 3$	69
7.8	Survey measurements for trajectory 7.7.	69

7.9	Scoring system used for the 2018-2019 competition [12].	70
7.10	Unoptimised complex trajectory with two target points.	71
7.11	Visualisation used for designing complex trajectories	72
7.12	Complete trajectory, axes limits scaled up to that of the competition rock.	73
7.13	Trajectory 7.12 from V/E and V/N perspective.	74
7.14	Trajectory 7.12 in sections.	75
7.15	Improved (T3-E): $d_S = d_E = 121$	76
7.16	Trajectory 7.12 from V/E and V/N perspective with new T3-E section.	77
7.17	How a GUI for trajectory optimization can be presented, this example is for the simple case.	78
7.18	The Digital Well Environment-GUI that is currently used on the rig [29].	79
7.19	Flowchart for trajectory optimisation implementation.	80
7.20	Trajectory which position uncertainty calculations will be done for.	85
7.21	Ellipsoids of uncertainty, $k = 2.7954$	86
7.22	Covariance matrix 3x3 elements, $k = 2.7954$	86
7.23	Semi-major and semi-minor axes, $k = 2.7954$ from 3x3 covariance matrix (NEV).	87
7.24	Based on 2x2 covariance matrix, $k = 2.4477$	88
7.25	Semi-major and semi-minor axes, $k = 2.4477$ from 2x2 covariance matrix.	88
7.26	Semi-major and semi-minor axes, $k = 2.4477$ from 2x2 covariance matrix in centimeters.	89
7.27	Pre-planned trajectory in blue, drilled trajectory and ellipsoids of uncertainty in red.	91

List of Tables

1.1	Rock strength data provided by the committee. 3 samples each in vertical and horizontal section of the rock [11].	3
4.1	Propagation modes [41].	30
4.2	Number of standard deviations. Confidence level is implicit [9].	32
4.3	Confidence level at specified number of dimensions. Number of standard deviations are implicit [9].	32
7.1	Length of MD shown in figure 7.2, with varying β . Total MD is including the vertical section.	64
7.2	Target points.	73
7.3	d_S and d_E values for the different intervals	73
7.4	Targets point with new E-point.	76
7.5	Accelerometer specifications for the ICM-20948 sensor, with focus on bias error.	82

List of Symbols and Abbreviations

A Azimuth.

B_x Magnetometer measurements in x-axis.

B_y Magnetometer measurements in y-axis.

B_z Magnetometer measurements in z-axis.

C_E Attractor point at E.

C_S Attractor point at S.

F Ratio factor.

G_x Accelerometer measurements in x-axis.

G_y Accelerometer measurements in y-axis.

G_z Accelerometer measurements in z-axis.

I Inclination.

R_i Length of semi-axes of an ellipsoid along principal axes.

$[C]_{hla}$ Error covariance matrix in HLA-axes.

$[C]_{nev}$ Error covariance matrix in NEV-axes.

$[T]_{hla}^{nev}$ Direction cosine matrix to transform NEV matrix to HLA..

ΔE Easting.

ΔN Northing.

ΔV Vertical Depth.

β Angle at kick-off point.

$\Delta \mathbf{r}$ Borehole displacement between successive survey stations.

$e_{i,l,k}$ The error due to the i th error source at the k th survey station at the l th survey leg.

$\frac{\delta p}{\varepsilon_i}$ Weighting function - effect of the i th error source on the survey measurement.

$\frac{dr}{dp}$ The effect on the borehole position due to changes in survey measurement vector.

λ Eigenvalue of a given covariance matrix.

ϕ Dogleg.

ρ_i Propagation coefficients.

σ Magnitude of the weighting function.

σ_E^2 Variance in E-measurements.

σ_N^2 Variance in N-measurements.

σ_V^2 Variance in V-measurements.

θ Magnetic dip angle.

ε^{bias} Bias error term.

$\varepsilon^{scale\ factor}$ Scale factor error term.

d_E Parameter to determine position of attractor point C_E .

d_S Parameter to determine position of attractor point C_S .

k Confidence level constant.

t_E Tangnet vector at E.

t_S Tangnet vector at S.

BHA Bottom Hole Assembly.

DAQ Data Acquisition.

DL Dogleg.

DLS Dogleg Severity.

DSATS Drilling Systems Automation Technical Section.

EMI Electromagnetic Interference.

EOR Enhanced Oil Recovery.

EOU Ellipsoid of Uncertainty.

GUI Graphic User Interface.

HSE Health, Safety and Environment.

I2C Inter-Integrated Circuit.

ID Inner Diameter.

IMU Inertial Measurement Unit.

ISCWSA Industry Steering Committee for Wellbore Survey Accuracy.

KOP Kick off point.

LC Load Cell.

LPM Liters Per Minute.

LSB Least Significant Bit.

MD Measured depth.

MWD Measurement While Drilling.

OD Outer Diameter.

PCB Printed Circuit Board.

PDC bit Polycrystalline Diamond Compact bit.

PLC Programmable Logic Controller.

PUM Position Uncertainty Model.

RKB Rotary Kelly Bushing.

ROP Rate Of Penetration.

RPM Rotations Per Minute.

SPE Society of Petroleum Engineers.

SPI Serial Peripheral Interface.

TVD True Vertical Depth.

WOB Weight On Bit.

Chapter 1

Background

1.1 UiS Drillbotics

UiS Drillbotics is a non-profit multidisciplinary student organization at the University Of Stavanger. Our organization competes in the international Drillbotics competition, where universities design and build a laboratory scale drilling rig. Using sensors and control algorithms to autonomously drill a rock sample provided by SPE's Drilling System Automation Technical Section (DSATS). The objective of the competition is described on Drillbotics web-page as: "Design a drilling rig and related equipment to autonomously drill a vertical well as quickly as possible while maintaining borehole quality and integrity of the drilling rig and drillstring." [11]. A more general description of the competition is as follows: "Drillbotics is an international competition for universities to design and build a small drilling rig that uses sensors and control algorithms to autonomously drill a rock sample provided by SPE's Drilling Systems Automation Technical Section (DSATS)" [11].

For the 2019-2020 competition, UiS Drillbotics representing the University of Stavanger will compete with twenty-eight (28) universities world wide. This years team consists of seven (7) students, including both Bachelor of Science and Master of Science students from the Institute of Energy and Petroleum (IEP), the Institute of Computer Science and Electrical Engineering (IDE) and the Institute of Mechanical and Structural Engineering and Materials Science (IMBM). This years competition will be a continuation of the progress made in the last three years. We will continue to improve our existing rig, adding new systems and methodologies for the new challenges that will be part of the 2019-2020 competition. A brief introduction and description of the current rig will be presented in this thesis.

Being an international competition, there are certain rules, guidelines and limita-

tions to abide by. These are presented yearly, and define the outline of the competition. This includes factors such as design, components, budget, organisation and how the competition will be arranged.

1.2 Guidelines

This will be a high level description, highlighting what is determined to be important. For further reading please see [13].

The 2020 competition introduces a major change in how it is organised. Students are given the opportunity to compete in two groups, A and B.

- Group A: Design and build a miniature drilling rig and autonomously drill a directional well through a homogeneous rock sample to a given plan.
- Group B: Design, model, and simulate controls for a miniature directional drilling rig and demonstrate on a virtual drilling system.

This year, the UiS Drillbotics team will be divided into two teams. One team for each part of the competition. Team A will focus on the physical rig, and team B will mainly work with developing models for simulations. Continuing this section, we will focus on the rules that mainly considers team A. For team A, these are the main points that will be essential for succeeding with the competition.

- A closed loop control system for the downhole data is mandatory, participants that fail to meet this criteria will get a failing grade.
- Design a rig and related equipment to autonomously drill a well, using downhole sensors, that is able to hit multiple directional targets at $[X,Y,Z]$ as quickly as possible, while maintaining borehole quality, and ensuring the integrity of the drilling rig and drillstring.
- Navigation shall be done autonomously, using both surface and downhole measurements to control the drilling process in real-time.
- The test will continue until the drill bit exits the rock sample, or the time runs out (three hours).

1.2.1 Competition Rock

On the competition day DSATS will present the competition rock. This will be a homogeneous sandstone with dimensions of 12" W x 24" L x 24" H (30 x 60 x 60 cm). The team will not have access to the rock, until the day of competition. Rock compressive strength values are provided by the committee and given in table 1.1.

The well center for the pilot hole, which is set to 1" (0.0254 m) will be marked by the committee. For the competition, the rock sample will be oriented so that it rests on a 30 cm x 60 cm face, resulting in drilling depth of 60 cm [13].

Test #	Compressive Strength (psi)	Young's Modulus (psi)	Poisson's Ratio
1	2298	1.70E+6	0.13
2	3095	1.52E+6	0.26
3	1845	1.31E+6	0.26
4	6230	1.72E+6	0.20
5	4553	1.27E+6	0.14
6	5305	1.50E+6	0.21

Table 1.1: Rock strength data provided by the committee. 3 samples each in vertical and horizontal section of the rock [11].

1.2.2 Bit

DSATS will send a bit by Baker Hughes to all finalists. The provided bit will be a PDC bit with diameter of 1.5" (38.1 mm) and 2.0" (50.8 mm) in total length, with 4 nozzles for fluid to exit. The bit is characterised by the short length and high bit anisotropy. The axial and side aggressiveness of the bit can be changed, using the tungsten carbide elements on the gage pads and the bit face. See figure 1.1 for the bit.

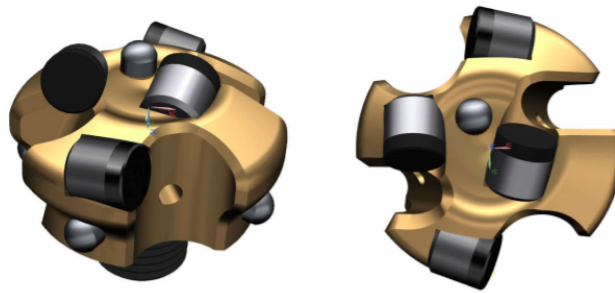


Figure 1.1: The provided Baker Hughes bit [21].

In addition, the team has decided to design a bit in collaboration with Lyng Drilling. Information about the self designed and manufactured bit is presented in section 2.

1.2.3 Drill pipe

In previous competitions the drill pipe has been an aluminum tube of 6061-T6 alloy. With length of 914.4mm, OD of 9.53mm and wall thickness 1.2446 mm. This year, use of stainless steel tubing is permitted and will be used by the UiS Drillbotics team. Details about the drill string are provided in section 2.1.

1.2.4 Automated Drilling

For the 2020 competition, directional drilling and well trajectory calculations will be an essential part. The team must design a system that is capable of drilling a directional well, following the planned trajectory presented to us at the day of the competition. The team will be scored on how accurately the target points are hit and how these intersect with the calculated well trajectory. The guidelines specify multiple requirements that must be part of the system, these include, providing a definitive directional surveys (TXT, LAS or CSV format) where a error model is associated with the well trajectory (ISCWSA error model or a self developed model). All teams are required to provide the pre-planned trajectory and an as-drilled trajectory, these shall be plotted on the same TVD vs. Vertical Section plot. In addition the two trajectories must be visualised on a X/Y plot, including grid north reference to *block north* and a [0,0] well center reference.

The team is free to use existing well trajectory models, or develop new models. It is required to include an appropriate survey station interval to accurately represent the drilled wellbore position [13]. All drilling control operations must operate autonomously, including directional drilling control. Directional surveys acquired while drilling, and data from the downhole sensors must be used as feedback for the steering control. This also includes set-points commands for parameters such as WOB, RPM and ROP. The drilling parameters must be optimized for the current drilling situation, and have integrated dynamic controls to mitigate or avoid drilling incidents.

1.2.5 Data Visualization

The guidelines states that intuitive and creative solutions for data visualization will give contestants a higher score. Standard drilling parameters ought to be presented in a format which is easy to read and understand. An appealing and easy to use Graphic User Interface (GUI) must be developed. Depths should adhere to industry standards for datum RKB which is equivalent of the rig's drill floor.

The aim of this chapter was to give the reader an idea of the project, background and motivation. For the detailed guidelines document see [13], and for more information about the competition and UiS Drillbotics please see references [29, 20, 18].

Chapter 2

Laboratory Scale Drilling Rig

This chapter will present the drilling rig built by the UiS Drillbotics team. The reader will be presented with a short description of the different systems. As the rig has received yearly upgrades, including both hardware and software. For an extensive description of the rig, see [18, 25, 20, 29].

2.1 Rotational System

The rotational system consists of two motors. A conventional top drive that is for vertical drilling and a downhole pneumatic motor installed last year for directional drilling. The top drive is a hollow-shaft brushless motor used to rotate the assembly, which provides 2.86 Nm and a maximum instantaneous torque of 8.59 Nm [29]. The hollow-shaft allows for circulation of drilling fluid, all the way from the pumps, through the top drive and out through the bit. The top drive can provide a RPM up to 3500 RPM, where the system limit is set to 1500 RPM. The top drive is part of the rotational system which is controlled using a programmable logic controller (PLC). By transmitting signals from the PLC it is possible to control the RPM and torque output of the top drive. The drill pipe is a 91.6 cm long stainless steel tube, type 316. With OD 10 mm, ID 7mm and wall thickness 1.5 mm. Destructive testing of the stainless pipe was performed and showed yield strength of 310 MPa, ultimate tensile strength of 363 MPa. Compression tests shows that buckling occurs at approximately at 100 MPa. Torsional tests resulted in yield strength of 32 Nm and shear at 88 Nm. For a detailed description of the tests performed and data for the steel pipe please refer to [8, 25].

The downhole motor is a pneumatic motor installed for vertical and directional drilling, by only rotating the bit. A crossover was designed and manufactured to

connect the bit to the pneumatic motor. Initially two models of the same motor were bought, type 302916D and 302916F with nominal torque ratings of 2.1 and 4.9 Nm, with 750 and 330 RPM [10]. Through experiments it was shown that the higher RPM model yielded better results, thus is the preferred model [29]. The speed and power output from the downhole motor is regulated by varying the flow rate or pressure of the system. The pneumatic system on the rig is designed with a compressor that delivers 345 LPM, a hydraulic maintenance unit which separates water from the compressed air, lubricates the motor and choke the inlet pressure to the pneumatic motor. Other components that are part of the pneumatic system are shown on figure 2.1.

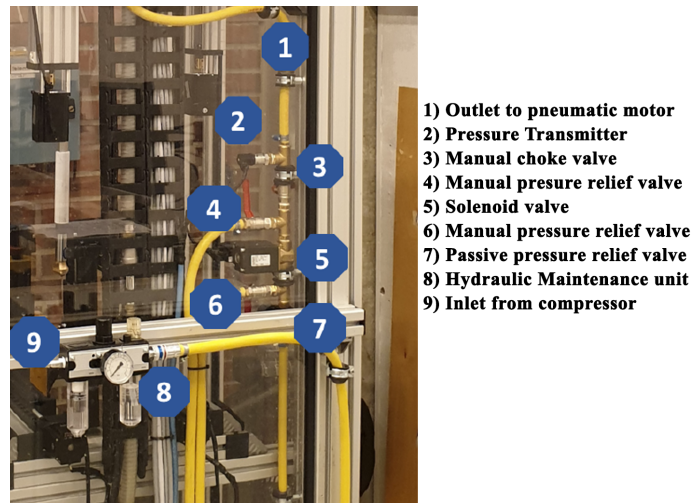


Figure 2.1: Pneumatic system overview [29].

2.2 Hoisting System

The rig is equipped with a hoisting system consisting of three actuators. For each of the actuators, a stepper motor is installed with a brake to reduce the holding torque when the system is not running. The actuators raise and lower the top plate where the top drive is installed. The top plate is connected to three brackets, each with a dedicated tri-axial load cells that measures the free hanging weight and hook load of the top plate [29]. The actuators are used to apply WOB by lowering the top plate against the formation. The actuators are controlled by the stepper motor, where the step-angle is 1.8° , for every 1.8° a total of 10 micro-steps are transmitted. The stepper motor is connected to a lead screw, where one revolution of the screw results in 8 mm of travel length of the top plate, this results in a resolution $4 \mu\text{m}$ [29]. With this configuration one has precise control of MD and WOB control. The

WOB is measured using load cells from HBM of model U9C, which measures both tensile and compressive force. These are the latest addition to the rig, and were installed February, 2020. Each of the three load cells have a capacity of 200N [19].

2.3 Circulation System

The rig was previously installed with two Xylem Flojet diaphragm electrically operated positive displacement pumps, that were used to circulate fresh water as mud [18]. These are now uninstalled as they were not used after the pneumatic motor was added to the system. Cuttings transportation is now done by redirecting the air exiting the exhausts on the pneumatic motor [37, 29].

2.4 Bottom Hole Assembly

The concept is based on having a mechanical section in the BHA that allows the assembly to bend and build an angle. Whereas previous year, a knuckle-joint with a whipstock to initiate the build section was used. The new BHA has an integrated bent section, that is fixed at 8° when no force is applied. It is designed to straighten when force is applied, for more details see [8, 25].

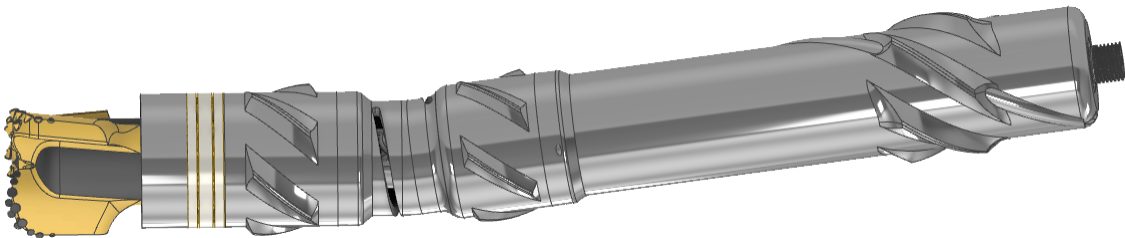


Figure 2.2: 3D render of the new BHA (J.E Karlsen 2020).

The top section (far right) covering the pneumatic motor, bent section with interior consisting of the bend configuration and axles connecting the pneumatic motor and bit for rotation. The bottom section is a stabilizers that connects to the thrust bearings and bit. The bit on figure 2.2 is a generic bit used to showcase the BHA. A custom bit is designed for the competition together with Lyng Drilling. The bit is a PDC bit with twelve cutters and four gauge insert. For extensive description of the custom bit, see [25].

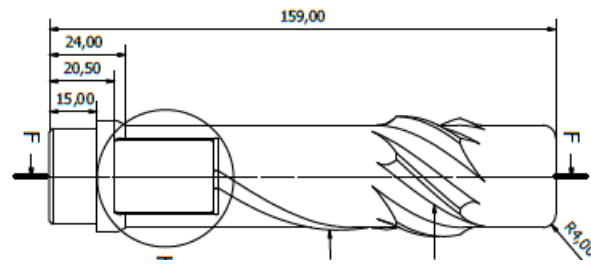


Figure 2.3: Top section, covering the pneumatic motor (J.E Karlsen 2020).

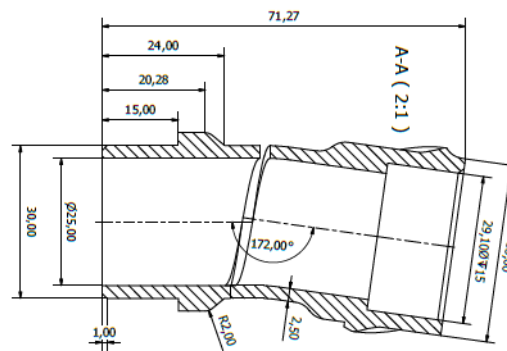


Figure 2.4: Bend section, which will cover the mechanical joint of the BHA (J.E Karlsen 2020).

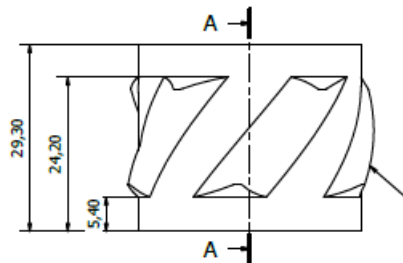


Figure 2.5: Bottom section, a stabilizer positioned below the bend (J.E Karlsen 2020).

The full assembly, including a thrust bearing and bit measures 290.65 mm. This will vary based on what bit is used and type of thrust bearing. All drawings of the BHA are designed and generated by UiS Drillbotics Team A member Jon E. Karlsen.



Figure 2.6: Full assembly with and without the exterior.

The exterior of the BHA is in figure 2.5 printed in carbon, while the finale product will be printed using 17-4 PH stainless steel. The bushing, the silver part with a black center on figure 2.5. Is installed to reduce friction between two surfaces. Left for the bushing, in silver is the thrust roller bearing. With purpose of guiding rotating axles and to transfer load. The bearing depicted here accommodates axial loads only [25].

2.5 Rig Sensors

The rig is equipped with a multitude of different sensor for different tasks. Sensors that are essential and prioritized are connected to a high frequency data acquisition module (DAQ) from HBM. Remaining sensors such as the downhole sensors are connected directly to the PC. This is accomplished using I2C protocol from sensor to the micro-controller which then transmits the signal via USB to PC. A complete overview of all sensors installed for the 2018-2019 competition are given on figure 2.7.

Sensor:	Description:	Range:	Units:
W_1, W_2, W_3	Surface WOB load-cells – <i>Forsentek FNZ 100N</i>	-300 – 300	N
T_1	Surface torque – <i>APM-HE09ACH inbuilt encoder</i>	0 – 8.59	Nm
T_2	Drill pipe torque – <i>TAM Active-3 Advanced</i>	10^{-4} – 10^3	MPa
H	Vertical depth – <i>Sick DT35-B15251</i>	0 – 100	cm
ω	Surface RPM – <i>APM-HE09ACH inbuilt encoder</i>	0 – 3500	RPM
P_{mud}	Mud Pump pressure – <i>Gems 10 bar, IP65, G1/4</i>	0 – 10	bar
$P_{\text{pneumatic}}$	Pneumatic pressure – <i>Gems 16 bar, IP65, G1/4</i>	0 – 16	bar
$\text{Acc}_x, \text{Acc}_y, \text{Acc}_z$	Accelerometer (downhole) – <i>LSM9DS0</i>	2 – 16	G
$\text{Gyr}_x, \text{Gyr}_y, \text{Gyr}_z$	Gyroscope (downhole) – <i>LSM9DS0</i>	245 – 2000	deg/s
$\text{Mag}_x, \text{Mag}_y, \text{Mag}_z$	Magnetometer (downhole) – <i>LSM9DS0</i>	2 – 12	gauss
T	Temperature (downhole) – <i>LSM9DS0</i>	-40 – 85	C
L	Virtual step counting – <i>Arduino Due</i>	0 – 900	mm
A	Acoustic sensor* for communication and bit-rock interaction wasted energy dissipation estimation	126 146	dB
V	High-speed camera as a vibration sensor*	0 – 530	frames/sec

(*Non-invasive, portable and underdevelopment sensor).

Figure 2.7: Rig sensors [29].

The directional drilling control measurements all rely on sensor data for inclination and azimuth. Previously a 9-DOG sensor, model LSM9DS0 was used for this purpose. From E. A. Løken and J. Løkkevik, 2019 [29]: "The downhole sensors are configured with sampling rate of 60 Hz using the Adafruit Trinket M0 and the pulse-algorithm implemented to request and receive measurements synced across all sensor sources (downhole sensor package and surface sensors)." The magnetometer measure in all three directions B_x, B_y and B_z and are mounted orthogonally to decide which way the magnetic North is. The same principle applies to the accelerometer displaying the data as following parameters, G_x, G_y and G_z .

A new upgraded sensor is being installed for the 2020 competition. The sensor is an Inertial Measurement Unit (IMU) from TDK InvenSense. The decision was made to change the old sensor, due to differences in dimensions. The new sensor being considerable smaller than the old one, with dimensions of 3 mm x 3 mm x 1 mm. In addition the new sensor has a higher sampling rate, which utilizes the same I2C protocol as the old sensor, with a high speed SPI at 7 MHz or 400 kHz fast mode I2C [23]. It comes with a 3-axis gyroscope, accelerometer and compass. The goal is to have a sensor which is more reliable and with less drift. Drift being the accumulation of errors in the sensor data, that increases with time and use. For the 2018-2019 competition, a set of algorithms to calculate the inclination and azimuth was implemented by the team. For further reading see [29], and for conversion of sensor data to Cartesian coordinates using the new sensor, see [25].

For directional drilling the sensor utilises the accelerometer and magnetometer to calculate inclination and azimuth. It will be mounted on the BHA on a custom designed printed circuit board, which is flexible [25]. Following formulas are implemented [29],

$$I = \frac{180}{\pi} \cdot \cos^{-1} \left(\frac{G_z}{\sqrt{G_x^2 + G_y^2 + G_z^2}} \right) \quad (2.1)$$

$$A = \frac{180}{\pi} \cdot \tan^{-1} \left(\frac{B_y}{B_x} \right) \quad (2.2)$$

Using these formulas both inclination and azimuth can be plotted in degrees. Initial tests where the sensor is mounted on a plate, and rotated 360° by hand have yielded promising results. These calculations are done using the raw data with no post-processing and calibration.

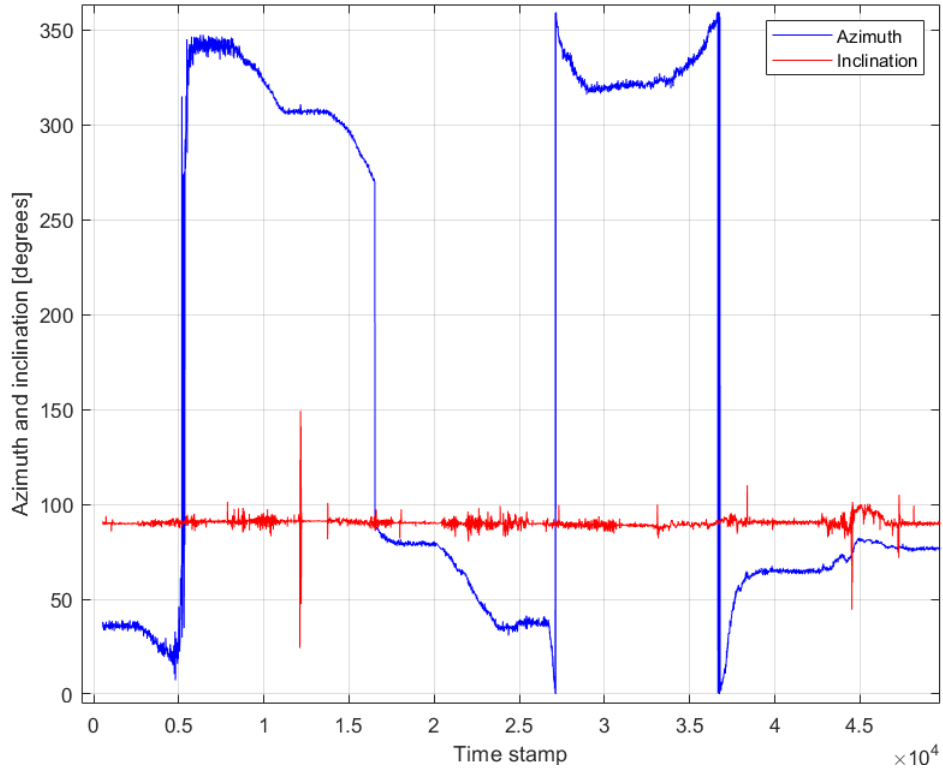


Figure 2.8: Inclination and azimuth given in degrees.

Chapter 3

Directional Drilling

This chapter will present the theoretical background for directional drilling and well trajectory calculations. The following literature presented will include terminology specifically used for directional drilling and survey calculations. The following definitions are from F. O. Farah, 2013 [16].

- **Azimuth:** The angle ($^{\circ}$) between the north direction and the plane containing the vertical line through the wellhead and the vertical line through the target.
- **Build-up rate:** The build-up rate ($^{\circ}/30\text{m}$ or 100ft) is the rate at which the angle is built.
- **Drop-off point:** The depth where the inclination starts to drop off (i.e. the well profile tending to vertical)
- **Displacement:** The horizontal distance between the vertical lines passing through wellhead and the target.
- **Inclination:** Angle ($^{\circ}$) made by the tangential section of the hole with the vertical.
- **Kick-off point (KOP):** The depth at which the well is first deviated from the vertical and build section is initiated.
- **Measured depth (MD):** Actual depth (length) of the well along the well path.
- **True-vertical depth (TVD):** Vertical distance between the top of the well and bottom of well.
- **Tangent section or Hold section:** Section of the well where the well path is maintained at a certain inclination.

3.1 Application of Directional Drilling

Directional drilling is an essential part of modern drilling operations. It can be defined as the practice of drilling non-vertical wells, or the practice of controlling the direction and deviation of the well to a predetermined underground target. Major applications of directional drilling include:

- Sidetracking
- Geology/Salt Domes
- Relief Well
- Horizontal Drilling
- Multilateral

Sidetracking may be required if a fishing operation is unsuccessful, due to a collapsed well or to avoid certain geological features. A cement plug is placed on top of the section and is allowed to set firmly. This provides the well with a foundation from which the new section of the hole can be kicked off. Often using a whipstock or downhole motor together with a bent sub. Whereas sidetracking is often due to failed drilling operations, directional drilling is also used for avoiding geological problems. As hydrocarbon reservoirs are sometimes associated or in close proximity to salt dome structures. Rather than drilling through a salt dome, which may lead to drilling incidents such as washouts, lost circulation and corrosion. It is preferred to avoid the salt formation by drilling a deviated well. Furthermore, conventional vertical drilling through a steeply dipping plane, may lead to movement or slippage along that plane. This can be avoided using directional drilling [22]. Directional drilling can be used for tertiary well control, by drilling relief wells. These are drilled adjacent to an existing well where the primary and secondary well barrier have failed. The goal being to divert the pressurized hydrocarbons up a new wellbore in a controlled manner, like killing a well that has taken a kick [3].

Majority of wells are drilled to an inclination of approximately 60° , inclination beyond this increases the exposure to drilling incident and the cost of the well. Yet, there are benefits to horizontal wells. Such as increasing the drainage area of the reservoir, prevention of water and gas coning, enhanced oil recovery (EOR) and increased penetration of reservoir [22]. The cost of drilling a horizontal well, must be justified by increasing the productivity and production of the well. Horizontal wells can be combined with multilateral wells. Where a multilateral well, is a well with two or multiple laterals. These can be horizontal vertical or deviated. Multilateral wells are used in fields with complex geological features, and contribute to exposing more of the reservoir and increasing production from a single platform slot

[2]. Multilateral wells can be drilled as opposed dual laterals, stacked dual laterals, multilaterals and branched as seen of figure 3.1.

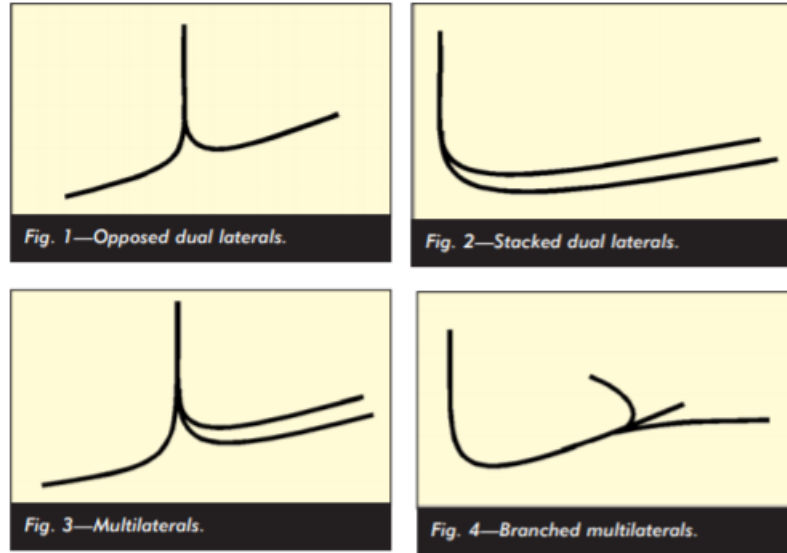


Figure 3.1: Different configurations of multilateral wells [2].

3.2 Types of Well Trajectories

Wells can be drilled using a multitude of different well profiles. The most common well trajectories are vertical, J-shape, S-shape and horizontal.

3.2.1 Vertical

Vertical wells are simply vertical wells with no inclination, in reality minimal changes in inclination do occur in vertical wells..

3.2.2 J-shape

J-type wells are widely used and characterized by a straight vertical section, a kick-off point KOP with build section and a hold section.

3.2.3 S-shape

S-shape wells are more complicated than J-shaped wells. First section is a vertical section, followed by a kick-off point and a build section. When desired angle is

reached the angle is held until desired target is reached. The last part of the S-shaped well consists of a drop section. Where inclination of the well is dropped until the well is vertical. Then vertical drilling can proceed. S-shaped wells can be used for avoiding salt domes, hitting multiple targets or to avoid faults.

3.2.4 Horizontal Wells

Horizontal wells consist of a vertical section with a KOP. At the KOP the well starts the build section until the well turn horizontal. At 90° there is no drop section, but small variations may occur in the inclination. Horizontal wells are used for production in thin lateral reservoirs, avoiding gas and water coning and increasing production in reservoir with low permeability [22].

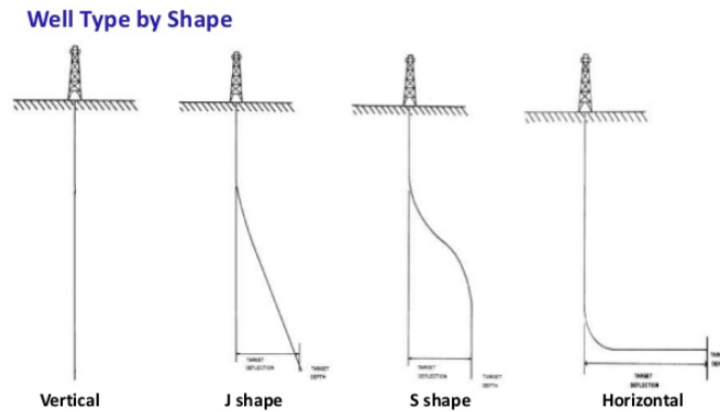


Figure 3.2: Types of Well Trajectories [15].

3.3 Survey Calculations

This section will present multiple methods used for survey calculations. Highlighting the advantages and disadvantages of the five most commonly used methods. For this section, inclination will be given as I and azimuth by A . Subscripts will be given by 1 and 2, denoting upper and lower survey stations. ΔMD is the length between two stations, this is the measured depth. ΔV , ΔN and ΔE are the incremental distance between stations along the three axes (vertical, east and north) [22].

3.3.1 Tangential Method

The tangential method assumes that the wellpath can be described as a straight line using inclination and azimuth from the lower survey station. Due to not including

data from upper survey station, this method yields significant errors in wells with substantial changes in short intervals, as in deviated wells. Thus, this method should not be used, unless the course lengths are shorter than the length of the survey tool [16].

$$\Delta V = \Delta MD \cdot \cos(I_2) \quad (3.1)$$

$$\Delta N = \Delta MD \cdot \sin(I_2) \cdot \cos(A_2) \quad (3.2)$$

$$\Delta E = \Delta MD \cdot \sin(I_2) \cdot \sin(A_2) \quad (3.3)$$

3.3.2 Average Angle Method

Like the tangential method, average angle method also assumes a straight line. The straight line intersects both the upper and lower survey station. It is calculated using the average inclination and azimuth from both survey stations.

$$\Delta V = \Delta MD \cdot \cos\left(\frac{I_1 + I_2}{2}\right) \quad (3.4)$$

$$\Delta N = \Delta MD \cdot \sin\left(\frac{I_1 + I_2}{2}\right) \cdot \cos\left(\frac{A_1 + A_2}{2}\right) \quad (3.5)$$

$$\Delta E = \Delta MD \cdot \sin\left(\frac{I_1 + I_2}{2}\right) \cdot \sin\left(\frac{A_1 + A_2}{2}\right) \quad (3.6)$$

Figure 3.3 shows the difference between the two methods. The figure denotes inclination as I_i , azimuth as Az_i , ΔTVD as our ΔV and measured depth as ΔMD . The dashed lines illustrates calculated wellbore, and the solid line represents the actual wellbore.

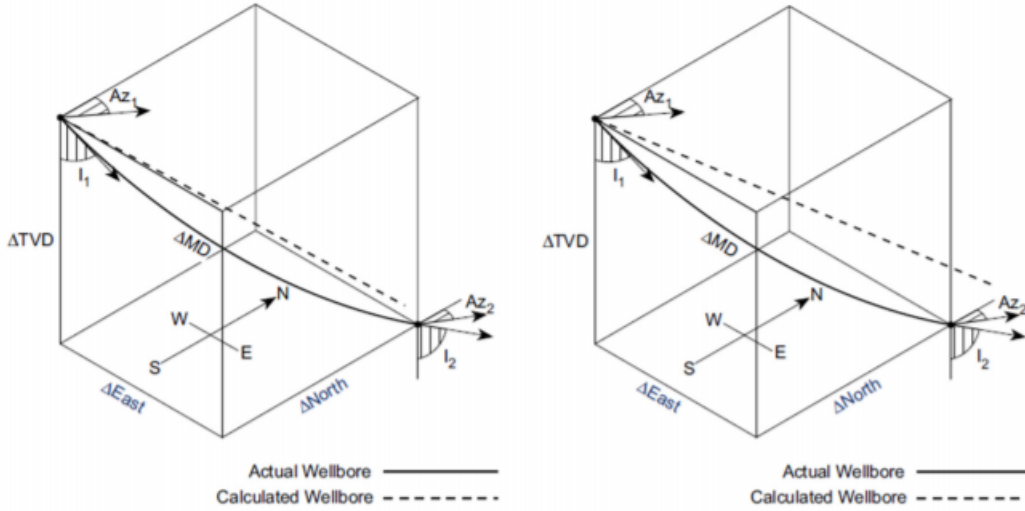


Figure 3.3: **Left:** Average angle method, **Right:** Tangential angle method [16].

3.3.3 Balanced Tangential Method

A method that is considerably more accurate than the tangential method, is the balanced tangential method. It takes both survey stations into account, and assumes that the actual wellpath can be approximately calculated using two straight lines of equal length [22].

$$\Delta V = \frac{1}{2} \cdot \Delta MD \cdot (\cos(I_1) + \cos(I_2)) \quad (3.7)$$

$$\Delta N = \frac{1}{2} \cdot \Delta MD \cdot (\sin(I_1)\cos(A_1) + \sin(I_2)\cos(A_2)) \quad (3.8)$$

$$\Delta E = \frac{1}{2} \cdot \Delta MD \cdot (\sin(I_1)\sin(A_1) + \sin(I_2)\sin(A_2)) \quad (3.9)$$

3.3.4 Radius of Curvature Method

Contradictory to the previously mentioned methods, the radius of curvature method states that the wellpath can not be calculated using straight lines, but assumes the wellpath to be a circular arc. This method assumes that the wellpath lies on a cylinder with a vertical axis, and with a radius equal to that of the curvature in the horizontal plane [16]. The final equations are presented below, but are calculated by separately calculating two sets of different equations for the horizontal and vertical

plane. For further reading on calculations for horizontal and vertical plane, please see [22].

$$\Delta V = \frac{\Delta MD \cdot (\sin(I_2 - I_1))}{I_2 - I_1} \cdot \left(\frac{180}{\pi}\right) \quad (3.10)$$

$$\Delta N = \frac{\Delta MD \cdot (\cos(I_1) - \cos(I_2)) \cdot (\sin(A_2) - \sin(A_1))}{(I_2 - I_1) \cdot (A_2 - A_1)} \cdot \left(\frac{180}{\pi}\right)^2 \quad (3.11)$$

$$\Delta E = \frac{\Delta MD \cdot (\cos(I_1) - \cos(I_2)) \cdot (\cos(A_1) - \cos(A_2))}{(I_2 - I_1) \cdot (A_2 - A_1)} \cdot \left(\frac{180}{\pi}\right)^2 \quad (3.12)$$

Radius of curvature method provides a more accurate method to approximate the actual wellpath, compared to the average angle method. However the method assumes the radius stays constant throughout the whole arc, which might not be accurate for wells with long build section [22].

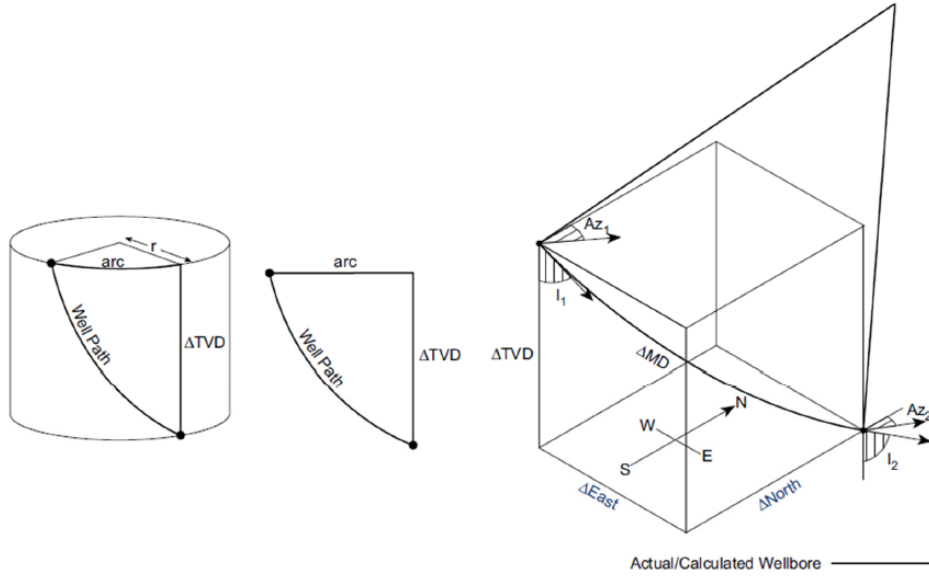


Figure 3.4: Radius of curvature method [16].

3.3.5 Minimum Curvature Method

Minimum curvature method builds on the principles from the balanced tangential method. Rather than strictly assuming the wellpath can be approximated by two straight lines, the wellpath is calculated using a circular arc. This is done by applying

a ratio factor F , that is based on the bending between the two stations (dog-leg angle). Where the formula for dogleg is given as

$$\phi = \cos^{-1}[\cos(I_1)\cos(I_2) + \sin(I_1)\sin(I_2)\cos(A_2 - A_1)] \quad (3.13)$$

The ratio factor F , can be calculated using figure 3.5. Where F can be expressed as

$$F = \frac{2}{\phi} \cdot \left(\frac{180}{\pi}\right) \cdot \tan\left(\frac{\phi}{2}\right) \quad (3.14)$$

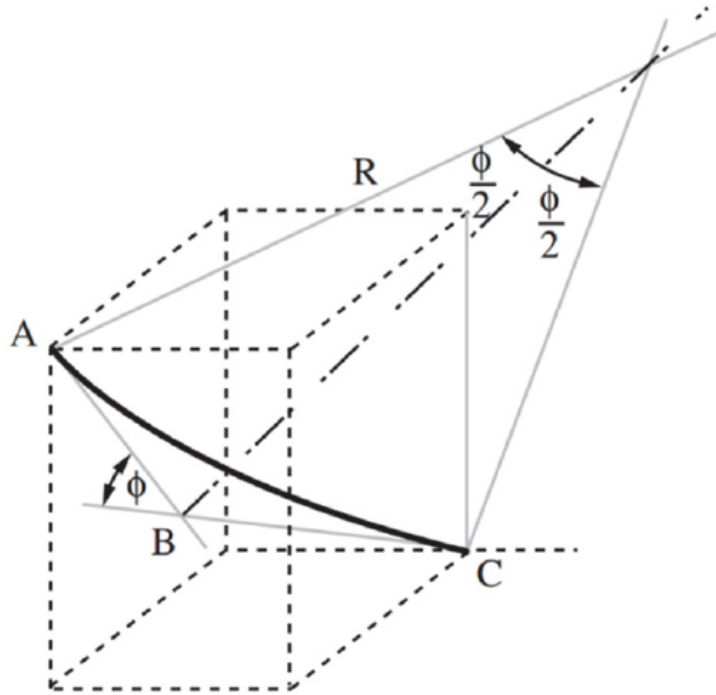


Figure 3.5: Minimum curvature method [16].

The ratio factor is then applied to the equations for ΔV , ΔN and ΔE given by the balanced tangential method, resulting in following equations [22]

$$\Delta V = F \cdot \frac{\Delta MD}{2} \cdot (\cos(I_1) + \cos(I_2)) \quad (3.15)$$

$$\Delta N = F \cdot \frac{\Delta MD}{2} \cdot (\sin(I_1)\cos(A_1) + \sin(I_2)\cos(A_2)) \quad (3.16)$$

$$\Delta E = F \cdot \frac{\Delta MD}{2} \cdot (\sin(I_1)\sin(A_1) + \sin(I_2)\sin(A_2)) \quad (3.17)$$

Minimum curvature method is regarded as one of the most accurate methods by the industry, and is thus commonly used [6]. As seen on figure 3.6 for a specific well, the minimum curvature method provided the most accurate results with no error.

<u>Calculation Methods</u>	<u>TVD Error (ft)</u>	<u>Displacement Error (ft)</u>
Tangential	-4.76	+14.99
Balanced tangential	-0.11	-0.03
Average angle	0.00	-0.25
Curvature radius	-0.04	-0.31
Minimum curvature	—	—

Well Specifications: well = Sperry-Sun test hole, total survey depth = 6,023 ft, maximum angle = 26°, survey interval = approximately 62 ft, and well profile = vertical hole to 4,064 ft building to 26° at 6,023 ft.

Figure 3.6: Comparing the accuracy of the different methods presented [7].

Using this method one is able to obtain the 3D-coordinates of a well using only measured depth, inclination and azimuth between two survey stations as input. The output is $\Delta V / \Delta TVD$, ΔN , ΔE , DL, DLS. Despite the accuracy of the minimum curvature method, it is decided that for the laboratory drilling rig the balanced tangential method will be used. This is due to the uncertainty calculations which are presented in chapter 4 are based on the balanced tangential method. Using this method one can visualise the wellpath in real time, using data as inclination, azimuth and measured depth from our sensors.

Chapter 4

Error Models and Uncertainties

As mentioned in 1.2.4 the competition requires the directional drilling calculations to be associated with an error model. In this thesis the Industry Steering Committee for Wellbore Survey Accuracy model, also known as the SPE Wellbore Positioning Technical Section will be presented [41]. The ISCWSA produces, maintains and publishes standards and the mathematical framework for survey error models. This section will describe the overall error model that can be used for survey errors, and present the theoretical approach for uncertainties regarding the downhole measurements.

4.1 Sensor Errors

Prior to presenting the ISCWSA model, some sensor errors are presented. These are some of the errors that are included in the error model presented later in this chapter. A short introduction to terms such as bias and scale factor-error is given. The error sources will be affected by the equipment used for data collection. For the UiS Drillbotics rig, the IMU sensor-package that will be located downhole acting as the survey tool. The sensor-package includes an accelerometer, magnetometer and a gyroscope. These have error sources which contribute to errors in the surveys. This section will present the two main sensor errors that are considered in this thesis.

An example of how data is recorded by a sensor for a z-accelerometer from [41].

$$G_z = (1 + \varepsilon_{G_z}^{scale\ factor})G_z^{true} + \varepsilon_{G_z}^{bias} \quad (4.1)$$

Where G_z is the value that the user read from the sensor, G_z^{true} is the true value and $\varepsilon_{G_z}^{scale\ factor}$ and $\varepsilon_{G_z}^{bias}$ represents the residual errors of the sensor after calibration.

4.1.1 Bias Error

A bias error for a gyroscope and accelerometer would be the finite measured output, even when no rotation or acceleration is applied to the sensor. For a magnetometer this would be the output even in zero-field conditions [38]. This is a systematic error that is regarded as an residual error that will be present after calibration. The bias error is visualised on figure 4.1 with the scale factor error. If measured value from a sensor is y , and the real value that is to be measured is x . The bias error can be expressed as $y = x + 2$. Where 2 is the value of the bias error.

4.1.2 Scale Factor-Error

The scale factor-error can be represented as a constant multiplier which scales the real measurement. Again, if y is measured value by the sensor, and x is the real value. Ideally $y = x$. In reality a scale factor-error is present, which can be expressed as $y = 1.05x$, where 0.05 is the scale factor term. This is what is shown in equation 4.1 as $\varepsilon_{G_z}^{scale\ factor}$.

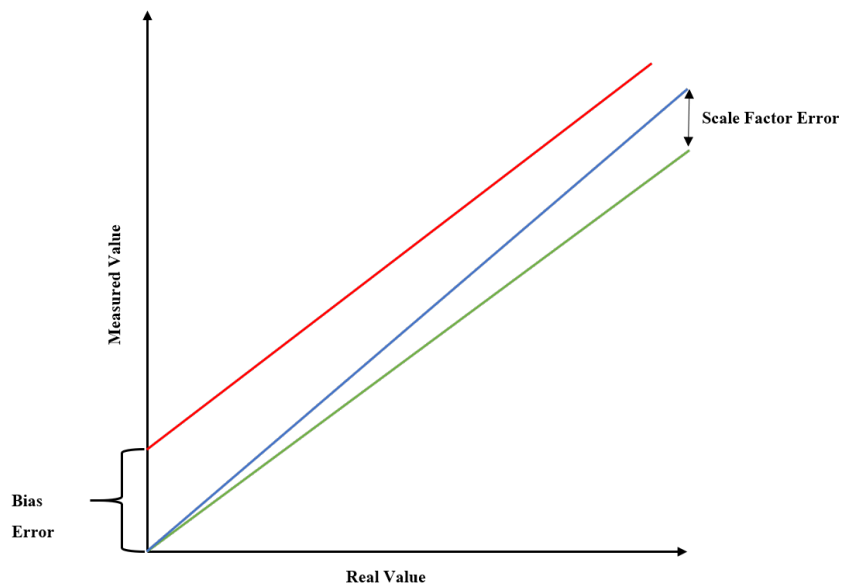


Figure 4.1: Bias- and scalar factor error visualised [40].

4.2 ISCWSA Model

The purpose of an error model is to evaluate and consider the effect of various factors that could lead to errors in the survey measurement, and hence to quantify the uncertainty. For a given survey tool, the downhole sensor in our case, multiple factors may lead to errors. These are to be evaluated at the particular survey station and used to determine position uncertainties. Errors from the survey stations will accumulate along the wellpath as MD increases, and the contribution of all the individual errors are summed to determine the total uncertainty for the wellbore position [41]. With the ISCWSA model, these uncertainties are expressed as a covariance matrix which describes the error along each of the coordinate axes. This matrix can be used to determine the uncertainty ellipsoid with a given confidence level. A specific survey tool, or sensors will exhibit a bias error, even when calibrated. This offset or bias error can be determined from data sets or from data-sheets, by first determining the typical range of bias then quantifying it as a standard deviation [41].

The term "accumulated uncertainties" along the well, are accumulations on a statistical basis. If the x-accelerometer bias error is expected to have the same value point to point, the uncertainties are added arithmetically. If the error is un-correlated and randomise from point to point. In this case, the uncertainties are root summed squared together [41]. The combination of all contributions due to individual error sources, an assumption is made for the model, that all of the isolated error sources are independent from each other. Meaning an error in x-accelerometer is independent from y-accelerometer and vice versa. From here on, Position Uncertainty Model will be referred to as PUM.

To calculate the PUM, following terms must be defined.

- For a specific survey tool/sensor there are a number of **error sources** which effect the data. These can be identified and will lead to a set error in model. An example is the residual sensor error after calibration of the sensor.
- **Error magnitude** which is the standard deviation of that error, determined from test data.
- **Weighting functions** which are a set of mathematical equations used to define the impact of a given error source, and its impact on measured depth, inclination and azimuth.
- Each error should be assigned a **propagation mode** which defines how an error changes from survey station to survey station, this is used to calculate the total accumulation of errors.

4.2.1 Assumptions and Limitations of the Model

The documentation published by ISCWSA states that their model is designed to be a practical method that can be relatively easily implemented in existing software, to be used by well planners and directional drillers. It is intended to be viable for a wide range of tools, and is not developed with specific tools in mind. The model is only applicable for survey runs under normal industry best-practise procedures. These includes: sensors that have been regularly and thoroughly calibrated, sufficiently short survey intervals, field quality control checks, such as total magnetic field, gyro drifts, magnetic dip and the use of non-magnetic spacing for surveys. The model assumes that the wellpath can be described by a arc between the survey stations, and as a rule of thumb that the survey interval is 100 ft [41]. Given these assumptions and limitations, being industry standards and practices.

4.2.2 Details of the Mathematical Framework

In the ISCWSA model following axes sets are used, see figure 4.2 [4].

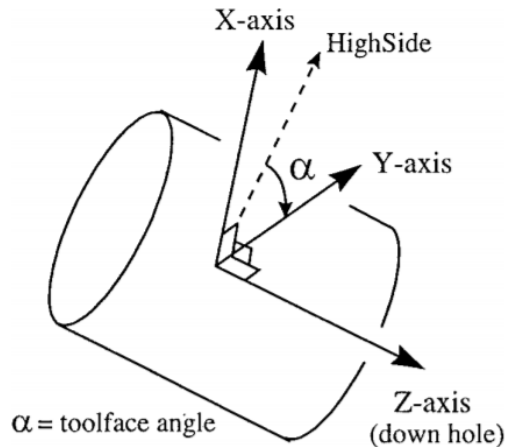


Figure 4.2: Definition of axes and toolface angle [4].

The axes sets are divided into three groups, to differentiate between the reference used. In Definition of the ISCWSA Error Model, Revision 4.3 [41] these are defined as:

- **Body Reference Frame (tool axes)** The z-axis is coincident with the along the well axis of the survey tool and the x and y-axes are perpendicular to z and to each other. This axes set is used to define orientation of the various sensors.

- **Earth Centred Reference Frame (nev)** The x-axis is in the horizontal plane and points towards the true north, the y-axis is also in the horizontal plane and points towards true east. The z-axis points downwards.
- **Borehole Reference Frame (hla)** The z-axis is aligned along the borehole axis. The x-axis is perpendicular to z and points toward the high side. The y-axis is perpendicular to both of these and hence is laterally aligned across the borehole.

4.2.2.1 Notations Used

Following subscripts i,k and l will be used throughout the section for the mathematical framework. Where i is used to index different error sources from 1...I, k used to index different survey stations in a survey leg from 1...K and l is used to index different survey legs in a well from 1...L. In addition the terms below are used in the error model as presented by ISCWSA [41].

- σ_i the magnitude of the ith error source.
- **3x1 vectors:** bold typeface is used to identify vectors quantities.

$\mathbf{e}_{i,l,k}$ the error due to the ith error source at the kth survey station at the lth survey leg.

$\frac{\delta \mathbf{p}}{\varepsilon_i}$ weighting function - effect of the ith error source on the survey measurement vector

$\Delta \mathbf{r}$ borehole displacement between successive survey stations.

- **3x3 matrices:**

$\frac{d\mathbf{r}}{d\mathbf{p}}$ the effect on the borehole position due to changes in the survey measurement vector

$[C]_{new}$ error covariance matrix in NEV-axes

$[T]_{hla}^{nev}$ NEV to HLA transformation direction cosine matrix.

4.2.3 Evaluating Position Uncertainty

The core equation used for the error evaluation is,

$$\mathbf{e}_i = \sigma_i \frac{d\mathbf{r}}{d\mathbf{p}} \frac{\delta \mathbf{p}}{\delta \varepsilon_i} \quad (4.2)$$

Where ε represents the error source (magnetometer, accelerometer etc.), i the index for the particular error source considered and σ_i is the magnitude of the uncertainty for the i th error. For example σ_i for a magnetometer calibration error could be set to 70nT [41]. The weighting function $\frac{\delta \mathbf{p}}{\delta \varepsilon}$ is a 3x1 vector that we express with respect to the survey measurements for depth, inclination and azimuth.

$$\frac{\delta \mathbf{p}}{\delta \varepsilon_i} = \begin{bmatrix} \frac{\delta D}{\delta \varepsilon_i} & \frac{\delta I}{\delta \varepsilon_i} & \frac{\delta A}{\delta \varepsilon_i} \end{bmatrix} \quad (4.3)$$

The last part of equation 4.2 $\frac{d\mathbf{r}}{d\mathbf{p}}$ is the effect of the survey errors in measured depth, inclination and azimuth on the wellbore position in NEV-axes. This is expressed as an 3x3 matrix [4].

$$\frac{d\mathbf{r}}{d\mathbf{p}} = \begin{bmatrix} \frac{dN}{dMd} & \frac{dN}{dInc} & \frac{dN}{dAz} \\ \frac{dE}{dMd} & \frac{dE}{dInc} & \frac{dE}{dAz} \\ \frac{dV}{dMd} & \frac{dV}{dInc} & \frac{dV}{dAz} \end{bmatrix} \quad (4.4)$$

Next step is to calculate the matrix in equation 4.4. Wellbore positions can be calculated using numerous methods, as presented in section 3.3, thus the $\frac{d\mathbf{r}}{d\mathbf{p}}$ matrix will depend on the interval between two survey stations. Where $d\mathbf{r}$ can be expressed as Δr_k for the displacement between survey station $k-1$ and k , and Δr_{k+1} for the displacement between stations k and $k+1$. Equation 4.2 can be written with respect to Δr_k and Δr_{k+1} .

$$\mathbf{e}_{i,l,k} = \sigma_{i,l} \left(\frac{d\Delta \mathbf{r}_k}{d\mathbf{p}_k} + \frac{d\Delta \mathbf{r}_{k+1}}{d\mathbf{p}_k} \right) \frac{\delta \mathbf{p}_k}{\delta \varepsilon_i} \quad (4.5)$$

In the definition of the ISCWSA model [41], the balanced tangential method is used to calculate the $\frac{d\mathbf{r}_k}{d\mathbf{p}_k}$ due to its simplicity. It is also mentioned that doing this leads to no significant loss of accuracy in the end results. The balanced tangential model gives following equation for displacement between two survey stations $j-1$ and j in the NEV-axes.

$$\Delta \mathbf{r}_j = \begin{bmatrix} \Delta N \\ \Delta E \\ \Delta V \end{bmatrix} = \frac{D_j - D_{j-1}}{2} \begin{bmatrix} \sin I_{j-1} \cos A_{j-1} + \sin I_j \cos A_j \\ \sin I_{j-1} \sin A_{j-1} + \sin I_j \sin A_j \\ \cos I_{j-1} + \cos I_j \end{bmatrix} \quad (4.6)$$

The final 3x3 matrix equations of $\frac{d\Delta \mathbf{r}_k}{d\mathbf{p}_k}$ and $\frac{d\Delta \mathbf{r}_{k+1}}{d\mathbf{p}_k}$ can be expressed be as (where $j=k$):

$$\frac{d\Delta \mathbf{r}_k}{d\mathbf{p}_k} = \frac{1}{2} \begin{bmatrix} \sin I_{k-1} \cos A_{k-1} + \sin I_k \cos A_k & (D_k - D_{k-1}) \cos I_k \cos A_k & -(D_k - D_{k-1}) \sin I_k \sin A_k \\ \sin I_{k-1} \sin A_{k-1} + \sin I_k \sin A_k & (D_k - D_{k-1}) \cos I_k \sin A_k & (D_k - D_{k-1}) \sin I_k \cos A_k \\ \cos I_{k-1} + \cos I_k & -(D_k - D_{k-1}) \sin I_k & 0 \end{bmatrix} \quad (4.7)$$

$$\frac{d\Delta \mathbf{r}_{k+1}}{d\mathbf{p}_k} = \frac{1}{2} \begin{bmatrix} -\sin I_k \cos A_k - \sin I_{k+1} \cos A_{k+1} & (D_{k+1} - D_k) \cos I_k \cos A_k & -(D_{k+1} - D_k) \sin I_k \sin A_k \\ -\sin I_k \sin A_k - \sin I_{k+1} \sin A_{k+1} & (D_{k+1} - D_k) \cos I_k \sin A_k & (D_{k+1} - D_k) \sin I_k \cos A_k \\ -\cos I_k - \cos I_{k+1} & -(D_{k+1} - D_k) \sin I_k & 0 \end{bmatrix} \quad (4.8)$$

The detailed derivation of equations 4.7 and 4.8, including all steps can be found in [41].

4.3 Weighting Functions

Each error source has a three weighting functions which are used to define how a specific error source affects the measured depth, inclination and azimuth measurements [4].

4.3.1 Weighting Functions for Sensor

Case 1: For depth sensor error:

The weighting functions is

$$\begin{aligned} \frac{\delta \mathbf{p}}{\delta \varepsilon_i} &= \begin{bmatrix} \delta D & \delta I & \delta A \\ \delta \varepsilon_i & \delta \varepsilon_i & \delta \varepsilon_i \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{aligned} \quad (4.9)$$

The corresponding σ is the magnitude of the depth error source.

Case 2: For accelerometer and magnetometer error source:

The inclination can be calculated using following formula,

$$I = \cos^{-1} \left(\frac{G_z}{\sqrt{G_x^2 + G_y^2 + G_z^2}} \right) \quad (4.10)$$

Similarly the equation for azimuth,

$$A = \tan^{-1} \left(\frac{(G_x B_y - G_y B_x) \sqrt{G_x^2 + G_y^2 + G_z^2}}{B_z (G_x^2 + G_y^2) - G_z (G_x B_x + G_y B_y)} \right) \quad (4.11)$$

where G_x , G_y and G_z are measurements from the accelerometer and B_x , B_y and B_z are the measurements from the magnetometers. Their inverse relations are

$$G_x = -G \sin(I) \sin(\alpha) \quad (4.12)$$

$$G_y = -G \sin(I) \cos(\alpha) \quad (4.13)$$

$$G_z = G \cos(I) \quad (4.14)$$

Where G is the gravity field strength, and α is toolface angle [4].

$$B_x = B \cos(\theta) \cos(I) \cos(A) \sin(\alpha) - B \sin(\theta) \sin(I) \sin(\alpha) + B \cos(\theta) \sin(A) \cos(\alpha) \quad (4.15)$$

$$B_y = B \cos(\theta) \cos(I) \cos(A) \cos(\alpha) - B \sin(\theta) \sin(I) \cos(\alpha) - B \cos(\theta) \sin(A) \sin(\alpha) \quad (4.16)$$

$$B_z = B \cos(\theta) \sin(I) \cos(A) + B \sin(\theta) \cos(I) \quad (4.17)$$

B is the magnetic field strength and θ is magnetic dip angle. Considering G_z (ABZ) from the accelerometer as an example, G_z appears in both 4.10 and 4.11. The accelerometer readings do not have an effect on depth readings, thus the weighting functions for depth is equal to zero for this set of equations. Weighting functions for I and A is determined by taking the partial derivatives of 4.10 and 4.11 with respect to G_z [41]. Resulting in equation

$$\frac{\delta \mathbf{p}}{\delta \varepsilon_i} = ABZ = \frac{1}{G} \begin{bmatrix} 0 \\ -\sin(I) \\ \tan(\theta) \sin(I) \sin(A) \end{bmatrix} \quad (4.18)$$

The associated σ is the magnitude of Z-accelerometer error source [4].

4.4 Summation of Uncertainty Terms and Propagation Modes

The finale step is the summation of all error sources. In addition, consider how the different errors relate to each other and how they accumulate along the wellpath. The equation is formulated as:

$$[C_k]_{nev} = \sum_{errors_i} \sum_{k_1 \leq K} \sum_{k_2 \leq K} \rho(\varepsilon_{i,l_1,k_1}, \varepsilon_{i,l_2,k_2}) e_{i,l_1,k_1} \cdot e_{i,l_2,k_2}^T \quad (4.19)$$

Where $\rho(\varepsilon_{i,l_1,k_1})$ is the coefficient for the correlation for the error source between k_1, k_2 and l_1 and l_2 . This summation will give a 3x3 covariance matrix with each axis down the diagonal and the correlations between these values. The correlation coefficient is set to either 0 for uncorrelated, or 1 for correlated error terms between survey stations. For case 1. where $\rho = 1$ it is assumed that the same error value is present for each station, this could for example be a bias error or a residual sensor error which stays constant. In this case, the uncertainty contributions is summed using addition.

$$e_{total} = e_1 + e_2 \quad (4.20)$$

If the error are not correlated, $\rho = 0$. This can results in both causing a positive error or that one of them causes a negative error. For this case the error contributions must be root sum squared together [41].

$$e_{total} = \sqrt{e_1^2 + e_2^2} \quad (4.21)$$

An error source may be independent at all survey stations, or correlated between survey all survey station on a survey leg, or within the whole well. This can be expressed as four propagation modes.

Propagation Mode	Identifier	ρ_1	ρ_2	ρ_3	
Random	R	0	0	0	always independent
Systematic	S	1	0	0	correlated from survey station to survey station
Well by Well	W	1	1	0	correlated from leg to leg
Global	G	1	1	1	correlated over all wells

Table 4.1: Propagation modes [41].

Where the propagation coefficients are defines as:

ρ_1 correlation between survey stations within the same survey leg

ρ_2 correlation between survey stations in different legs in the same well

ρ_3 correlation between survey stations within different wells in the same field

Equation 4.19 can be transformed into a 3x3 covariance matrix, which describes the error ellipsoid at a particular station. Following equation is for the NEV-axes [26]. The computational steps and mathematical details can be found in [41].

$$[C]_{nev} = \begin{bmatrix} \sigma_N^2 & Cov(N, E) & Cov(N, V) \\ Cov(N, E) & \sigma_E^2 & Cov(E, V) \\ Cov(N, V) & Cov(E, V) & \sigma_V^2 \end{bmatrix} \quad (4.22)$$

Where σ_N^2 , σ_E^2 and σ_V^2 are the variances in their respective axes, with uncertainty in north axis at 1 standard deviation being $\pm\sqrt{\sigma_N^2}$. The rest of the terms are the covariances, which represent the orientation of the ellipsoid with respect to the principle axes. Equation 4.22 can be transformed into borehole reference using following equation.

$$[C]_{hla} = [T]_{hla}^{nevT} [C]_{nev} [T]_{hla}^{nev} \quad (4.23)$$

The direction cosine matrix equation 4.24 is used to transform the $[C]_{nev}$ matrix into a matrix for borehole reference frame (HLA) [41].

$$[T]_{hla}^{nev} = \begin{bmatrix} \cos(I)\cos(A) & -\sin(A) & \sin(I)\cos(A) \\ \cos(I)\sin(A) & \cos(A) & \sin(I)\sin(A) \\ -\sin(I) & 0 & \cos(I) \end{bmatrix} \quad (4.24)$$

With the covariance matrices one can present and visualise both the 2D and 3D error ellipse/ellipsoid.

4.5 Ellipse and Ellipsoid of Uncertainty

An error ellipse is a 2D or 3D ellipsoid representing the volume of uncertainty in a wellbore position at a specific depth [36]. The volume occupied by the ellipse is a result of the error sources, such as measurements of inclination and azimuth. In addition to the the chosen confidence level of the ellipse. The inclination error creates a high side dimension, azimuth error causes a lateral dimension and the measured depth error creates a third component along the axis of the wellbore [7]. These errors will propagate and accumulate with increasing depth, as shown by the

mathematical framework from ISCWSA, meaning the total area or volume of the ellipse will become larger as MD increases.

The magnitude of the ellipse will be defined by the confidence level. In tables 4.2 and 4.3 from [9] confidence levels with respect to dimensions are presented. A standard value for confidence level is 95% and would for 2D and 3D error ellipse yield standard deviation of 2.4477 and 2.7954. This assumes that the sensor errors follow a normal distribution [28].

Probability	Standard deviations				
dimensions	± 1	± 2	± 2.58	± 2.79	± 3
1	68.27%	95.45%	99.081	99.47%	99.73%
2	39.35%	86.47%	96.41%	97.96%	98.89%
3	19.87%	73.85%	91.63%	94.93%	97.07%

Table 4.2: Number of standard deviations. Confidence level is implicit [9].

Probability	Confidence level				
dimensions	68.3%	90.0%	95.0%	99.0%	99.7%
1	0.9999	1.5448	1.9599	2.5758	2.9677
2	1.5151	2.1459	2.4477	3.0348	3.4086
3	1.8779	2.5002	2.7954	3.3682	3.7325

Table 4.3: Confidence level at specified number of dimensions. Number of standard deviations are implicit [9].

Tables 4.3 and 4.2 can be presented as a curve. Figure 4.3 illustrates normal distribution.

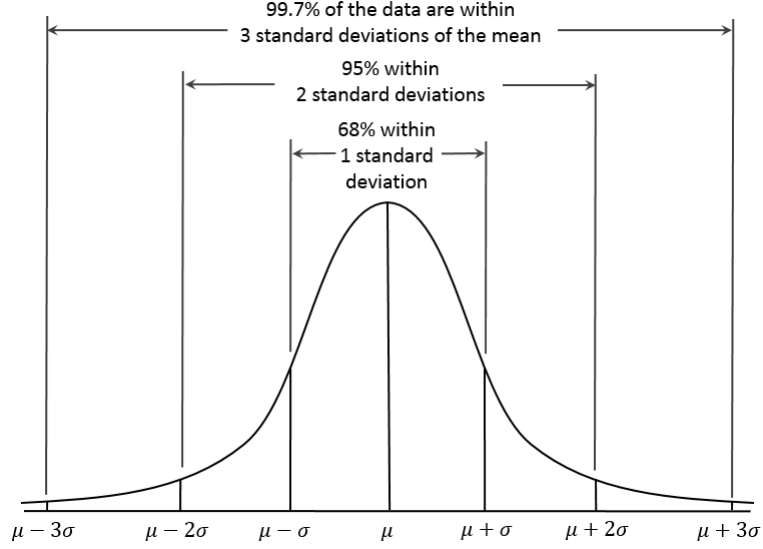


Figure 4.3: Normal Distribution Curve [5].

4.5.1 2D Ellipse of Uncertainty

For the calculation and visualisation of the 2D-error ellipse the method of eigenvalues and eigenvector will be used. The method is based on that the roots of the eigenvalues in matrix 4.25 represents the major and minor axes of the error ellipse. Furthermore, the eigenvectors are used to define the direction and orientation of the ellipse [27].

$$\begin{bmatrix} \sigma_N^2 & Cov(N, E) \\ Cov(N, E) & \sigma_E^2 \end{bmatrix} \quad (4.25)$$

To find the eigenvalues the matrix, the determinant of the matrix must be set to equal zero.

$$\begin{bmatrix} \sigma_N^2 - \lambda & Cov(N, E) \\ Cov(N, E) & \sigma_E^2 - \lambda \end{bmatrix} = 0 \quad (4.26)$$

Finding the determinant of the matrix

$$(\sigma_N^2 - \lambda)(\sigma_E^2 - \lambda) - Cov(N, E)^2 = 0 \quad (4.27)$$

$$\lambda^2 - \lambda(\sigma_N^2 + \sigma_E^2) + (\sigma_N^2\sigma_E^2 - Cov(N, E)^2) = 0 \quad (4.28)$$

From [28, 36] the two eigenvalues λ_1 and λ_2 can be found using the quadratic formula

$$\lambda_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (4.29)$$

Where

$$a = 1$$

$$b = -(\sigma_N^2 + \sigma_E^2)$$

$$c = \sigma_N^2 \sigma_E^2 - Cov(N, E)^2$$

Using a,b and c following equation 4.29 is obtained.

$$\lambda_{1,2} = \frac{(\sigma_N^2 + \sigma_E^2) \pm \sqrt{(\sigma_N^2 + \sigma_E^2)^2 + 4(Cov(N, E)^2 - \sigma_N^2 \sigma_E^2)}}{2} \quad (4.30)$$

As shown on figure 4.4 an ellipse can be expressed as semi-major axis a, and semi-minor axis b.

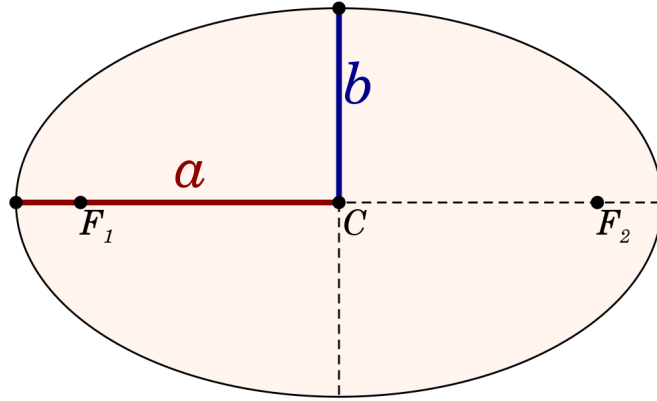


Figure 4.4: Overview of an ellipse, highlighting the axes [39].

These can be calculated using the eigenvalues $\lambda_{1,2}$ and the confidence constant k.

$$Semi - majoraxis = \sqrt{\lambda_1} \cdot k \quad (4.31)$$

$$Semi - minoraxis = \sqrt{\lambda_2} \cdot k \quad (4.32)$$

4.5.2 3D Ellipsoid of Uncertainty

For visualisation of an ellipsoid, the methodology is similar to a 2D ellipse. The 2x2 covariance matrix is replaced by a 3x3 matrix with a third component being V - vertical depth.

$$[C]_{nev} = \begin{bmatrix} \sigma_N^2 & Cov(N, E) & Cov(N, V) \\ Cov(N, E) & \sigma_E^2 & Cov(E, V) \\ Cov(N, V) & Cov(E, V) & \sigma_V^2 \end{bmatrix} \quad (4.33)$$

From L. Xiuashan, 2019 [42], by obtaining the eigenvalues for the covariance matrix in NEV-axes, the three axis radii can be expressed as:

$$R_i = k\sqrt{\lambda_i} \quad (i = 1, 2, 3) \quad (4.34)$$

Where λ_1 is radius in N-axis, λ_2 in E-axis and λ_3 is for the depth element.

Chapter 5

Case Study: Position Uncertainty - Field Data

This chapter is dedicated to the position uncertainty calculations based on the ISCWSA model. A MATLAB script has been written for the theory presented in Chapter 4. It is based on an excel spreadsheet for calculations according to the ISCWSA model. The spreadsheet is published by ISCWSA-representatives and can be found at [24]. The calculations in the spreadsheet have been verified on ISCWSA test wells. This makes it possible to confirm that the computational methods presented in MATLAB yields correct results.

5.1 Field Data

The field is labeled **ISCWSA Test #1**, and has total MD of the well is 8000 m and TVD is 3521 m. Initial inclination and azimuth is 0° , inclination is 90° and azimuth is 75° at finale depth. The well is characterised by a vertical section down to 1230 m TVD, before a build-section is held until inclination reaches 60° at a TVD of 1945 m. The inclination is held until 3444 m TVD where a new build-section starts. Complete horizontal drilling is reached at TVD 3521 m and 5400 m MD. This section is drilled until a MD of 8000 m is reached. The well has a geomagnetic field intensity B of 50000 nT, magnetic dip and declination of 72° and -4° . The meridian convergence is set to 0° . The well trajectory is visualised in figure 5.1. The data set provides a complete overview weighting functions, magnitudes, units and sources. A total of 27 weighting functions are considered for this case. Survey data is given in appendix D, for full access to the excel spreadsheet see [24].

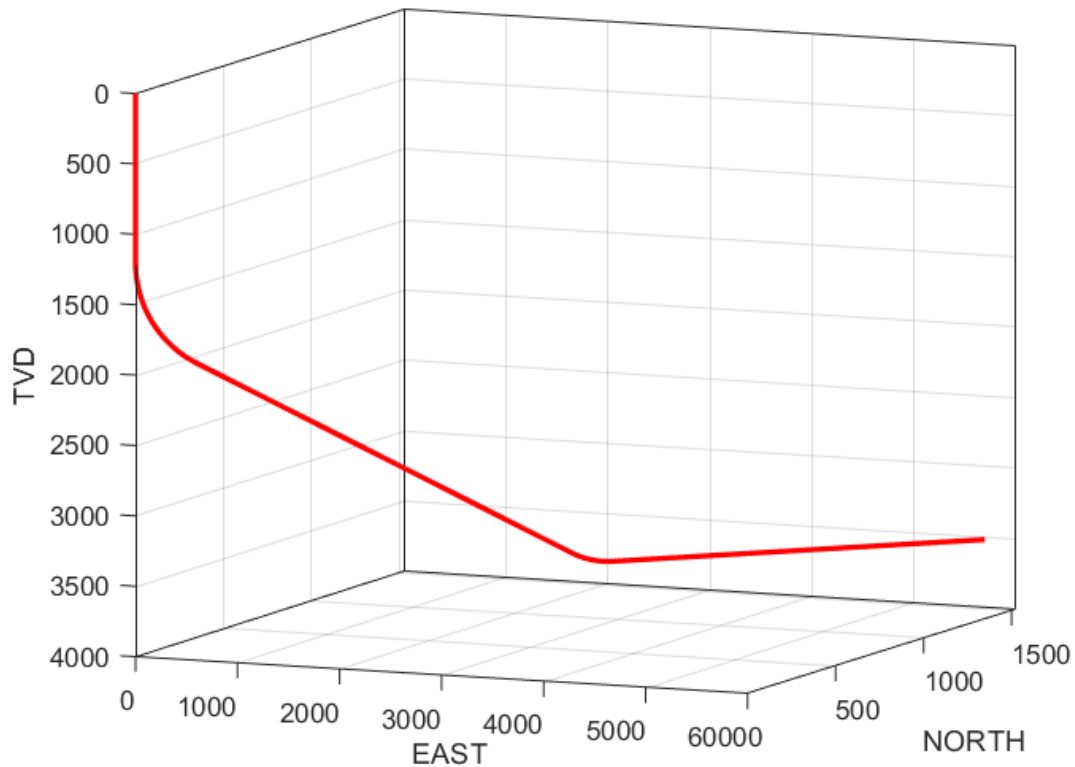


Figure 5.1: ISCWSA Test #1 Well

5.2 Confirmation of Computational Method

The spreadsheet contains calculations for the different matrices, but the ellipsoid of uncertainty itself is not calculated nor visualised. Validation of the values yielded by the script will be done by comparing the matrices. If the calculations performed in MATLAB are equal to the confirmed values in the spreadsheet, all other equations that are functions of these matrices should yield correct results. This assumes the calculation methods used are correct.

5.2.1 Covariance matrix with NEV-axes

The first matrix is the 3x3 covariance matrix which describes the error ellipsoid at specified station with respect to the [E,N,V] coordinate system. The covariance matrix is given as 5.1.

$$[C]_{nev} = \begin{bmatrix} \sigma_N^2 & Cov(N, E) & Cov(N, V) \\ Cov(N, E) & \sigma_E^2 & Cov(E, V) \\ Cov(N, V) & Cov(E, V) & \sigma_V^2 \end{bmatrix} \quad (5.1)$$

For following data, the variances σ_N^2, σ_E^2 and σ_V^2 are all denoted as NN,EE and VV. And Cov(N,E), Cov(N,V) and Cov(E,V) are presented as NE,NV and EV. Figure 5.5 shows the matrix elements calculated using the spreadsheet, and the same plot using values generated by MATLAB on figure 5.6. By observation of the plots 5.5 and 5.6 both sets of matrix elements follow the same trend. By closer inspection of the matrix elements, relatively minor changes can be observed when comparing the elements from excel and MATLAB. Figure 5.2 and 5.3 is an overview of the 8 last rows with data points for the covariance matrix elements. The values are sorted from the left: NN,EE,VV,NE,NV and EV, the values are to be multiplied by 10^3 . Figure 5.4 is the difference between the two sets of matrix elements. It is calculated by subtracting the excel values from MATLAB. The resulting values are the difference between the two sets of calculated values. The differences accumulates at the end rows, the highest difference is for the 268th row. Where the highest difference in values are for the NN and NE with 0.18%. The source of the error is not identified, the difference in use of significant figures between the spreadsheet and MATLAB script can be one potential source responsible for marginally deviating results. As the deviation from validated values are 0.18% at highest, the results are approved.

7.9548	0.6677	0.4356	-2.0944	-0.0503	-0.1190
8.0424	0.6743	0.4404	-2.1177	-0.0507	-0.1197
8.1304	0.6810	0.4452	-2.1411	-0.0510	-0.1203
8.2189	0.6877	0.4501	-2.1647	-0.0513	-0.1209
8.3080	0.6945	0.4550	-2.1884	-0.0516	-0.1215
8.3975	0.7013	0.4599	-2.2123	-0.0520	-0.1221
8.4875	0.7081	0.4648	-2.2363	-0.0523	-0.1227
8.5478	0.7127	0.4681	-2.2523	-0.0525	-0.1231

Figure 5.2: Last 8 rows with matrix elements from excel [24].

7.9696	0.6687	0.4356	-2.0983	-0.0504	-0.1190
8.0573	0.6754	0.4404	-2.1217	-0.0507	-0.1197
8.1455	0.6821	0.4452	-2.1452	-0.0510	-0.1203
8.2342	0.6888	0.4501	-2.1688	-0.0513	-0.1209
8.3234	0.6956	0.4550	-2.1926	-0.0517	-0.1215
8.4131	0.7024	0.4599	-2.2165	-0.0520	-0.1221
8.5033	0.7092	0.4648	-2.2405	-0.0523	-0.1227
8.5637	0.7138	0.4681	-2.2566	-0.0526	-0.1231

Figure 5.3: Last 8 rows with matrix elements from MATLAB

14.7942	1.0567	0.0000	-3.9540	-0.0209	0.0056
14.9595	1.0686	0	-3.9982	-0.0211	0.0056
15.1256	1.0805	0	-4.0427	-0.0212	0.0057
15.2927	1.0925	0.0000	-4.0874	-0.0214	0.0057
15.4607	1.1045	-0.0000	-4.1324	-0.0216	0.0058
15.6297	1.1166	0.0000	-4.1776	-0.0218	0.0058
15.7996	1.1288	0	-4.2231	-0.0220	0.0059
15.9134	1.1370	0.0000	-4.2536	-0.0221	0.0059

Figure 5.4: Difference in the last 8 rows.

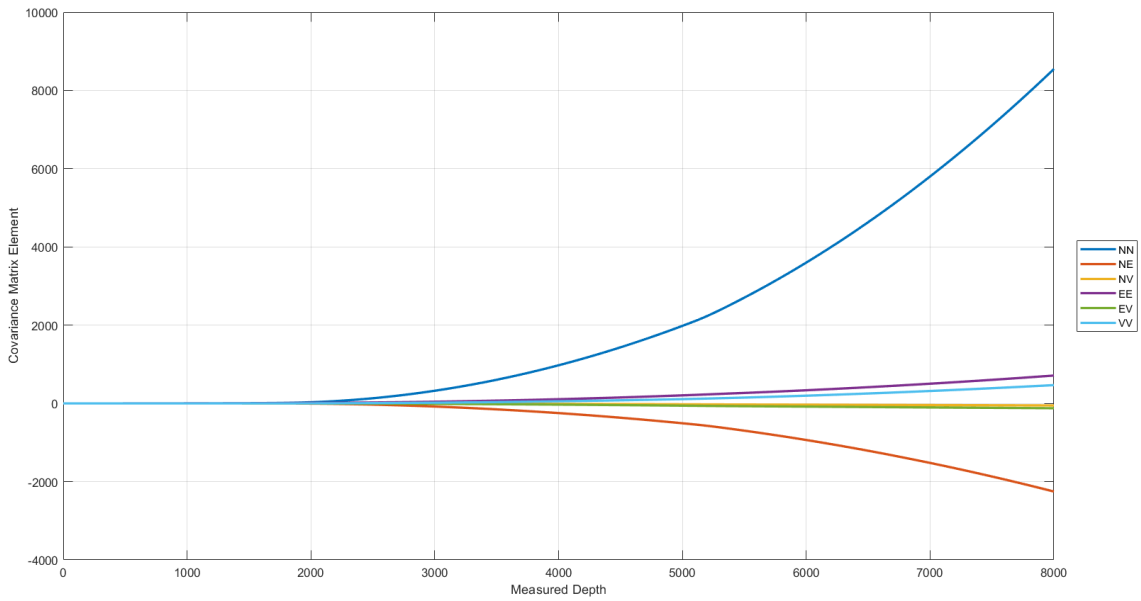


Figure 5.5: Matrix Elements and MD from excel spreadsheet [24].

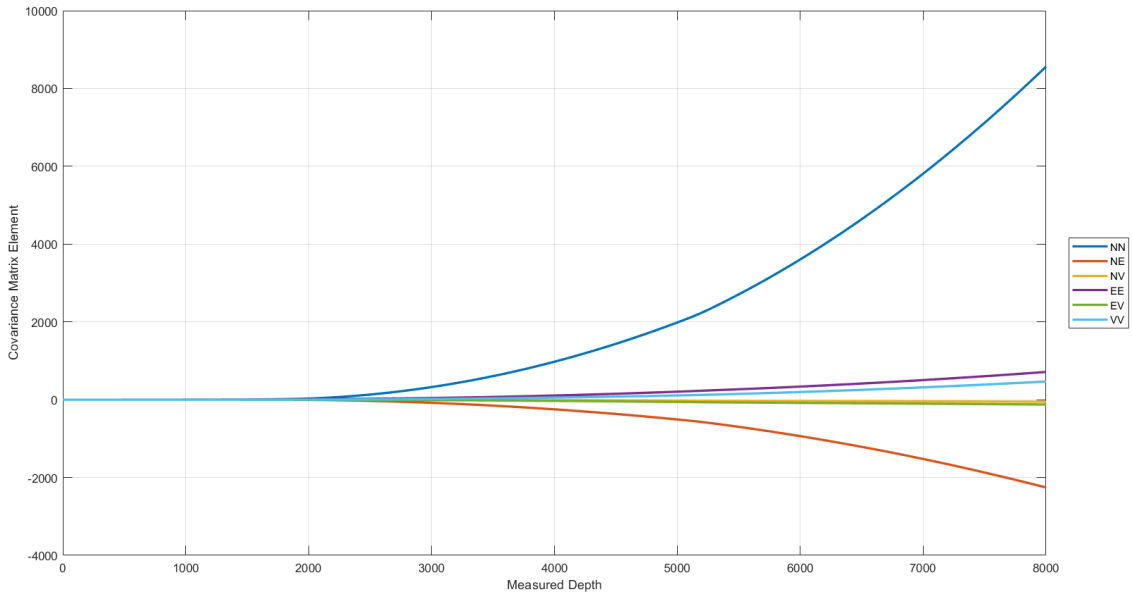


Figure 5.6: Matrix Elements and MD from MATLAB.

5.2.2 Covariance matrix with HLA-axes

This subsection will showcase the values for the matrix elements and their differences in the spreadsheet and MATLAB. The HLA-axes is the local coordinates in the borehole, commonly referred to as lateral, high-side and along hole. The procedure of presenting the 8 last rows with matrix elements, as the error accumulates and is largest towards the end of the depth. The same pattern is observed for the covariance matrix with the HLA-axes. Figure 5.7 and 5.8 are to be multiplied with 10^3 , and are sorted from left: HH,LL,AA,HL,HA and LA. Figure 5.9 show the difference in matrix elements from the two sources. Majority of the deviation is to be found for LL, which is approximately an 0.18% increase compared to the spreadsheet values.

0.4356	8.5138	0.1086	-0.0178	0.1280	-0.0080
0.4404	8.6076	0.1090	-0.0180	0.1287	-0.0081
0.4452	8.7019	0.1094	-0.0181	0.1294	-0.0081
0.4501	8.7968	0.1099	-0.0183	0.1301	-0.0081
0.4550	8.8922	0.1103	-0.0184	0.1307	-0.0081
0.4599	8.9881	0.1107	-0.0186	0.1314	-0.0082
0.4648	9.0845	0.1111	-0.0188	0.1321	-0.0082
0.4681	9.1491	0.1114	-0.0189	0.1325	-0.0082

Figure 5.7: Last 8 rows from excel (HLA-axes) [24].

0.4356	8.5297	0.1086	-0.0178	0.1280	-0.0080
0.4404	8.6237	0.1090	-0.0180	0.1287	-0.0081
0.4452	8.7182	0.1094	-0.0181	0.1294	-0.0081
0.4501	8.8132	0.1099	-0.0183	0.1301	-0.0081
0.4550	8.9087	0.1103	-0.0185	0.1307	-0.0081
0.4599	9.0048	0.1107	-0.0186	0.1314	-0.0082
0.4648	9.1014	0.1111	-0.0188	0.1321	-0.0082
0.4681	9.1661	0.1114	-0.0189	0.1325	-0.0082

Figure 5.8: Last 8 rows from MATLAB (HLA-axes).

0.0000	15.8510	-0.0000	-0.0216	0	-0.0101
0	16.0280	-0.0000	-0.0218	0	-0.0102
0	16.2061	-0.0000	-0.0220	0	-0.0102
0.0000	16.3852	-0.0000	-0.0222	0.0000	-0.0102
-0.0000	16.5652	-0.0000	-0.0224	0.0000	-0.0103
0.0000	16.7463	-0.0000	-0.0226	0.0000	-0.0103
0	16.9284	-0.0000	-0.0228	0	-0.0103
0.0000	17.0503	-0.0000	-0.0229	0	-0.0104

Figure 5.9: Difference in the last 8 rows (HLA-axes).

The data clearly shows the variance to be largest for NN and LL followed by EE and HH. Which is expected as the accumulation of error is usually largest in the lateral dimension followed by the high side dimension, the opposite can be observed if inclination measurements are more precise than azimuth measurements [17]. Consequences of this is visualised in the following section, where the covariance matrix is used to plot the ellipsoid of uncertainty.

5.3 Ellipsoid of Uncertainty

For visualisation of the ellipsoid of uncertainty the covariance matrices previously presented are used. Initially in the MATLAB script the covariance matrix elements are stored as a 1x6 array. Before an ellipsoid of uncertainty is visualised, the array must be rearranged. This is done by taking the elements from the array and sorting them in a 3x3 matrix which is the covariance matrix. The process is shown on figure 5.10 for the matrix in NEV-axes. The process for rearranging 1x6 arrays into 3x3 matrices is performed for all error sources and error models.

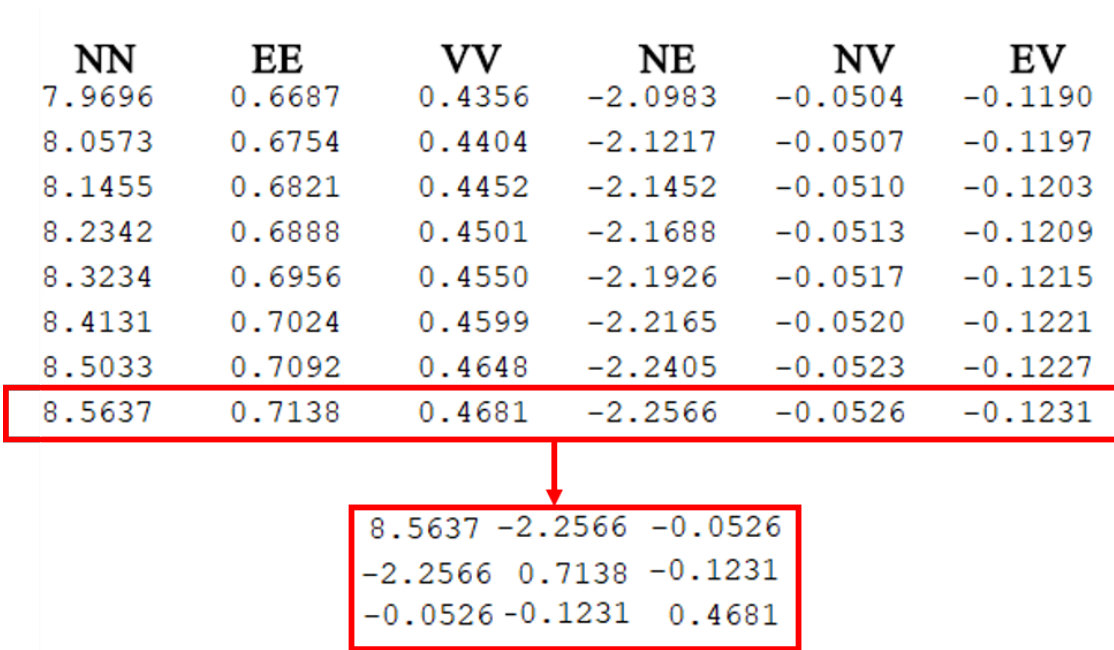


Figure 5.10: Arranging the 1x6 array into a 3x3 covariance matrix in MATLAB.

The covariance matrix shown in figure 5.10 is a summation of all error sources combined into one finale NEV-axes matrix. Including sensor, azimuth reference, alignment, depth and magnetic interference errors. By finding the eigenvalues of the total covariance matrix in NEV-axes, the trajectory can be visualised in a NEV coordinate system. Following figure shows the trajectory with ellipsoids of uncertainty for every 25th element of the covariance matrix at 1-standard deviation. Figure 5.11 is two figures showcasing the trajectory with EOU visualised. The green lines running along the trajectory is the semi-minor and major axes of the ellipsoid. The largest uncertainty is in the N-axis, the ellipsoids increase in size as depth from surface increases. Due to the independent sources of errors propagating and increasing in proportion with how far the survey point is from the origin [17]. The EOU visualised in figure 5.11 are not oriented with respect to their eigenvalues. Only considering the dimensions of the radii with center-points E,N,V.

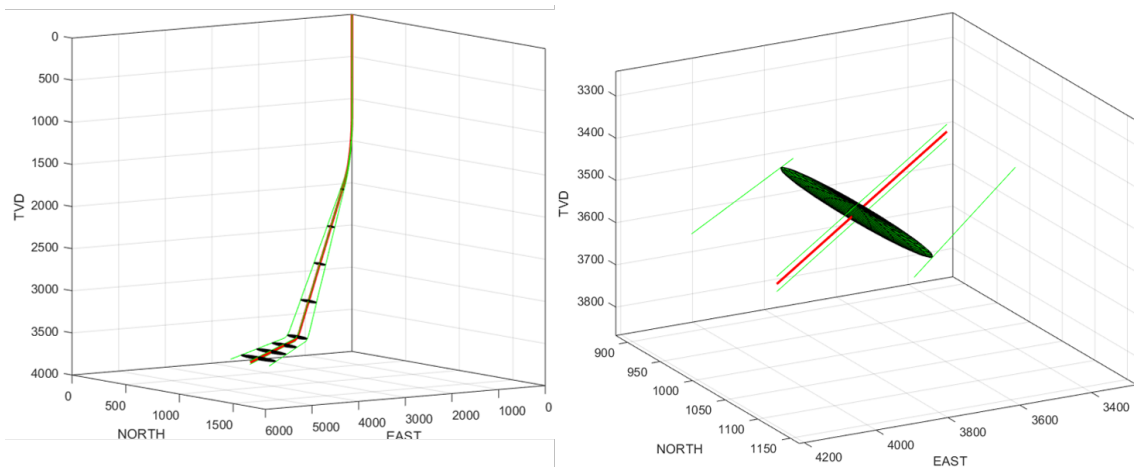


Figure 5.11: Ellipsoid of uncertainty at 1-standard deviation, $k = 1.8779$.

Figure 5.11 is visualised with $k = 1.8779$ at 1-standard deviation, statistically this gives the ellipsoid a confidence level of 68.3% that the well is in fact within the volume occupied by it. The semi-minor and major axes are visualised below using k at 1-standard deviation in 3 dimensions. The green line represents the semi-minor axis visualising uncertainty along the E-axis, and the red line is for the semi-major axis for uncertainty along the N-axis.

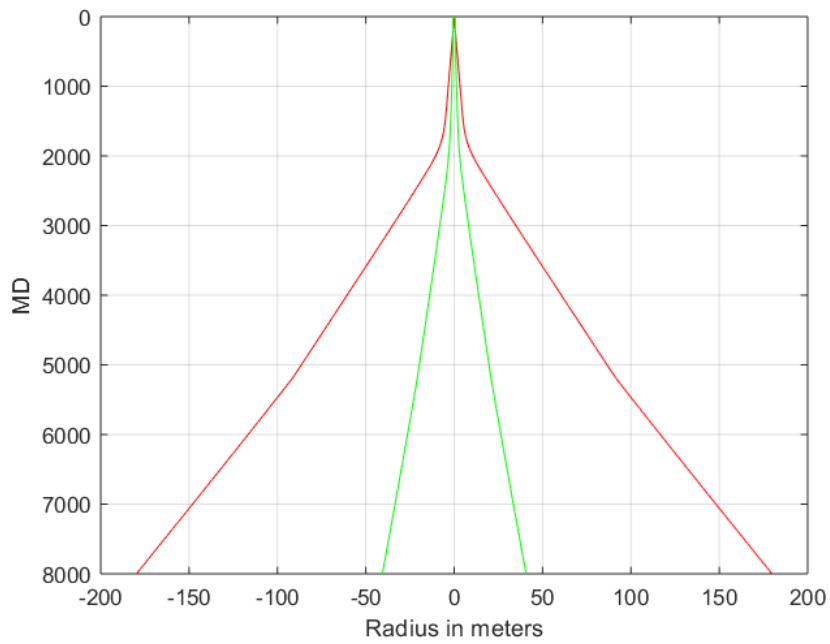


Figure 5.12: Semi-major and semi-minor axis with respect to MD, $k = 1.8779$ scaled for 3x3 matrix.

If higher precision is required for N/E direction, k can be set to 1.5151 for 68.3% confidence level (1-standard deviation) in 2 dimensions. This is done by sorting the calculated covariance matrix elements into a 2x2 matrix, as presented in 4.5.1.

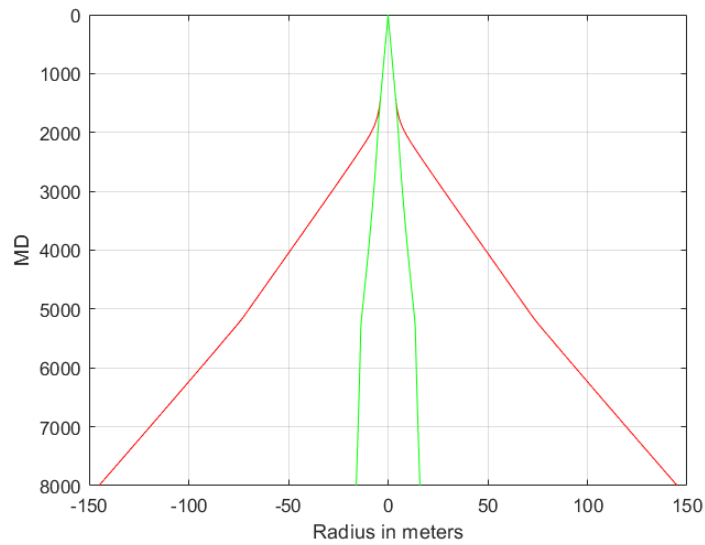


Figure 5.13: Semi-major and semi-minor axis with respect to MD, $k = 1.5151$ scaled for 2x2 matrix.

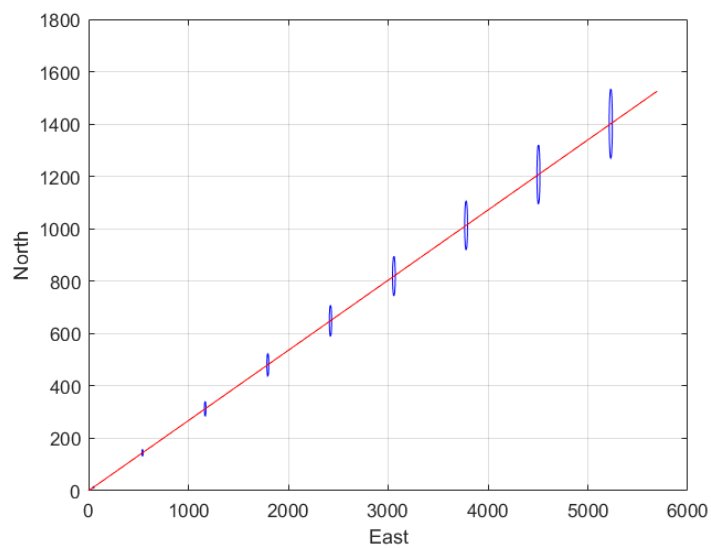


Figure 5.14: Trajectory plotted in N/E with ellipses of uncertainty in 2D, $k = 1.5151$.

The plots presented thus far have not taken the eigenvectors into consideration when visualising the ellipses. Figure 5.14 can be recreated using the calculated eigenvalues and eigenvector from the 2x2 covariance matrix.

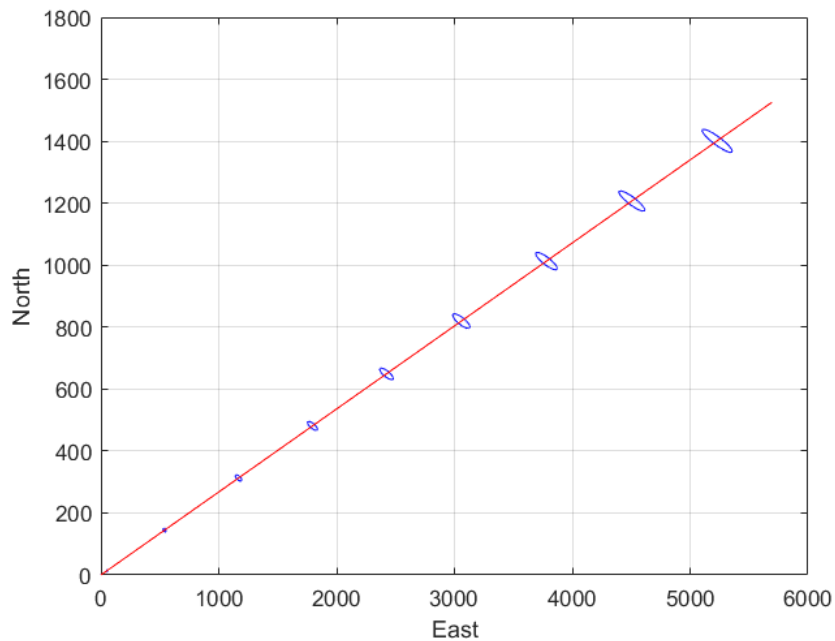


Figure 5.15: From 2x2 covariance matrix, $k = 1.5151$. Ellipses oriented with respect to eigenvectors.

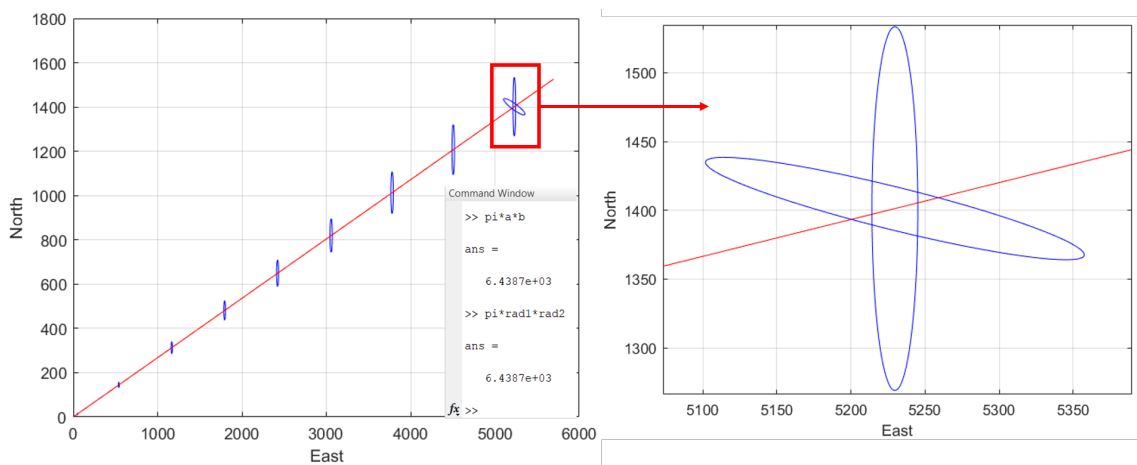


Figure 5.16: Showing that both ellipses are equal, $k = 1.5151$ for 2x2 covariance matrix.

Due to how the axes are scaled in MATLAB, initially the ellipses of 5.15 appear smaller than those of 5.14. This tendency is clear when one rotated ellipse is plotted with the non-rotated. It can be proven that these ellipses are equal by simply

calculating the area of the ellipse using $\pi \cdot r_1 \cdot r_2$. This is shown on figure 5.16, where rad1/rad2 are for the non-rotated case, and a/b are the radii of the ellipses rotated using the eigenvectors. If a practical approach is to be adopted, the confidence level of 68.3% must be increased. For a confidence level of 95% for an ellipsoid (3D) the k is set to 2.7954. This generates figure 5.17. With the semi-major and semi-minor axes given on figure 5.18.

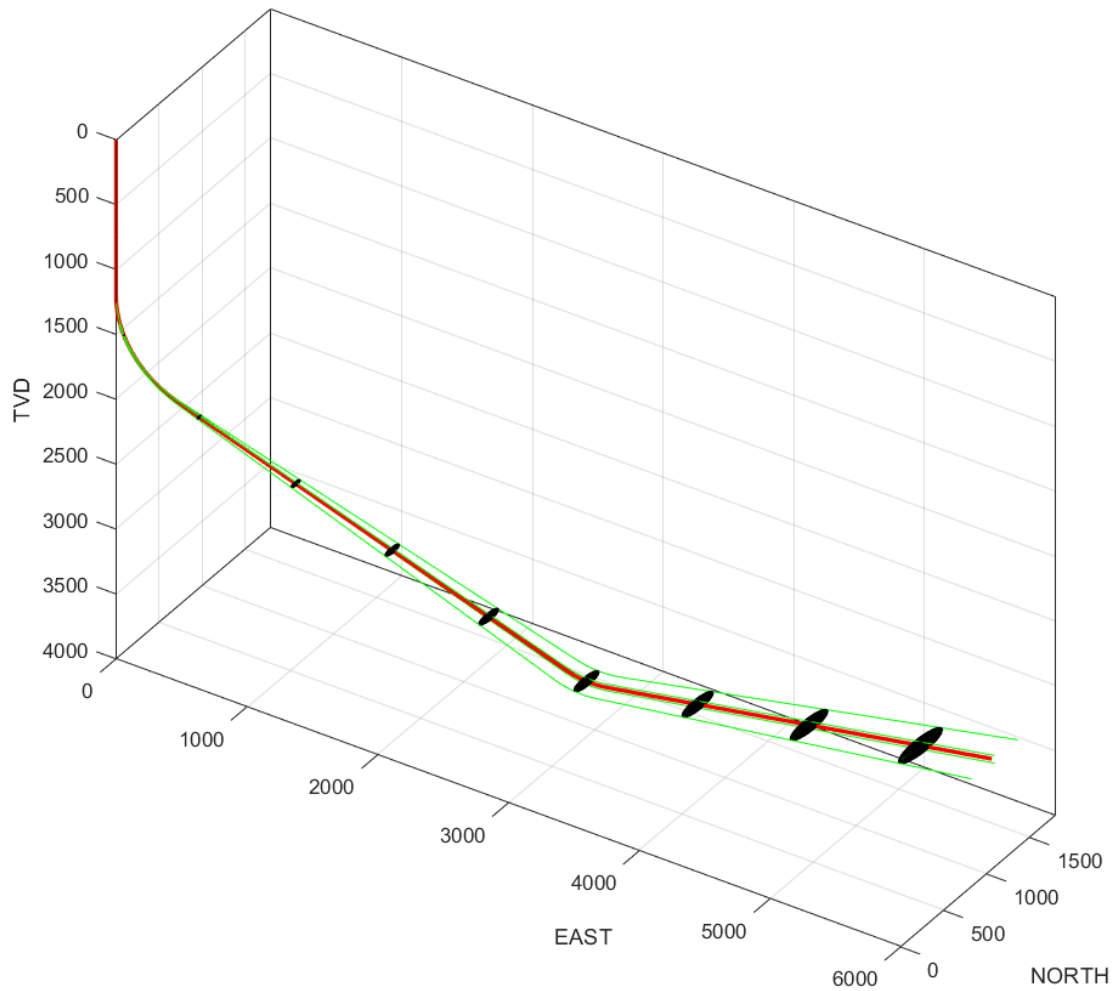


Figure 5.17: Based on 3x3 covariance matrix, $k = 2.7954$

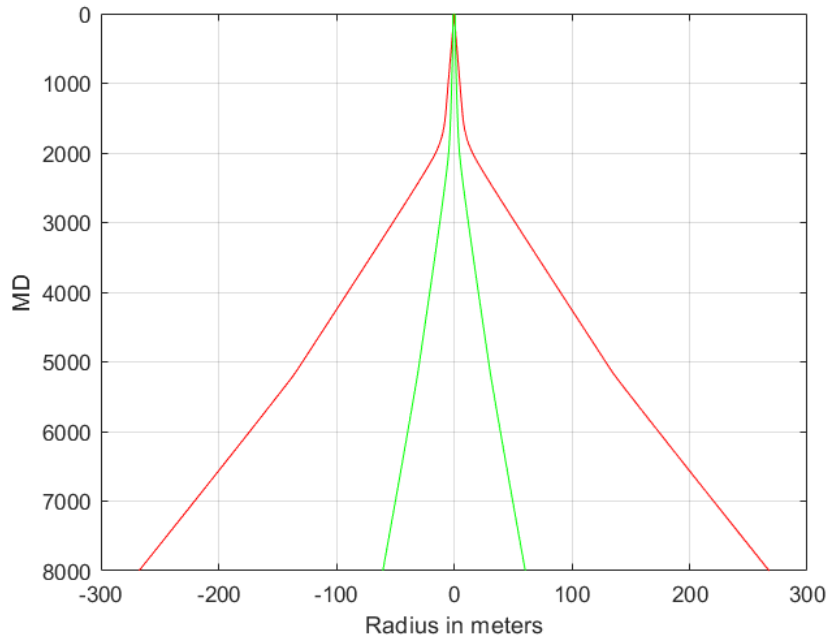


Figure 5.18: Semi-major and semi-minor axis, $k = 2.7954$

The major contributor to the overall uncertainty is along the north-axis. With a sharp increase at approximately 1650 m MD. This can be due to errors accumulating when the well goes from a vertical section to a direction well. The deviated drilling starts at 1230 m TVD, but the changes are minimal between MD and TVD until 1650 m. It is at 1650 m MD the displacement along the horizontal plane accelerates. At 8000 m MD the semi-major axis is 267 m (N) and semi-minor axis is 60.5 m (E). The uncertainty ellipsoids form a cone shape around well if an ellipsoid is plotted for every survey point. This visualises clearly how the error accumulates along the wellbore with increasing depth. This is shown on figure 5.19, where the black frame on the ellipsoid is turned off. The volume occupied by the ellipsoids of uncertainty is visualised by the green colour.

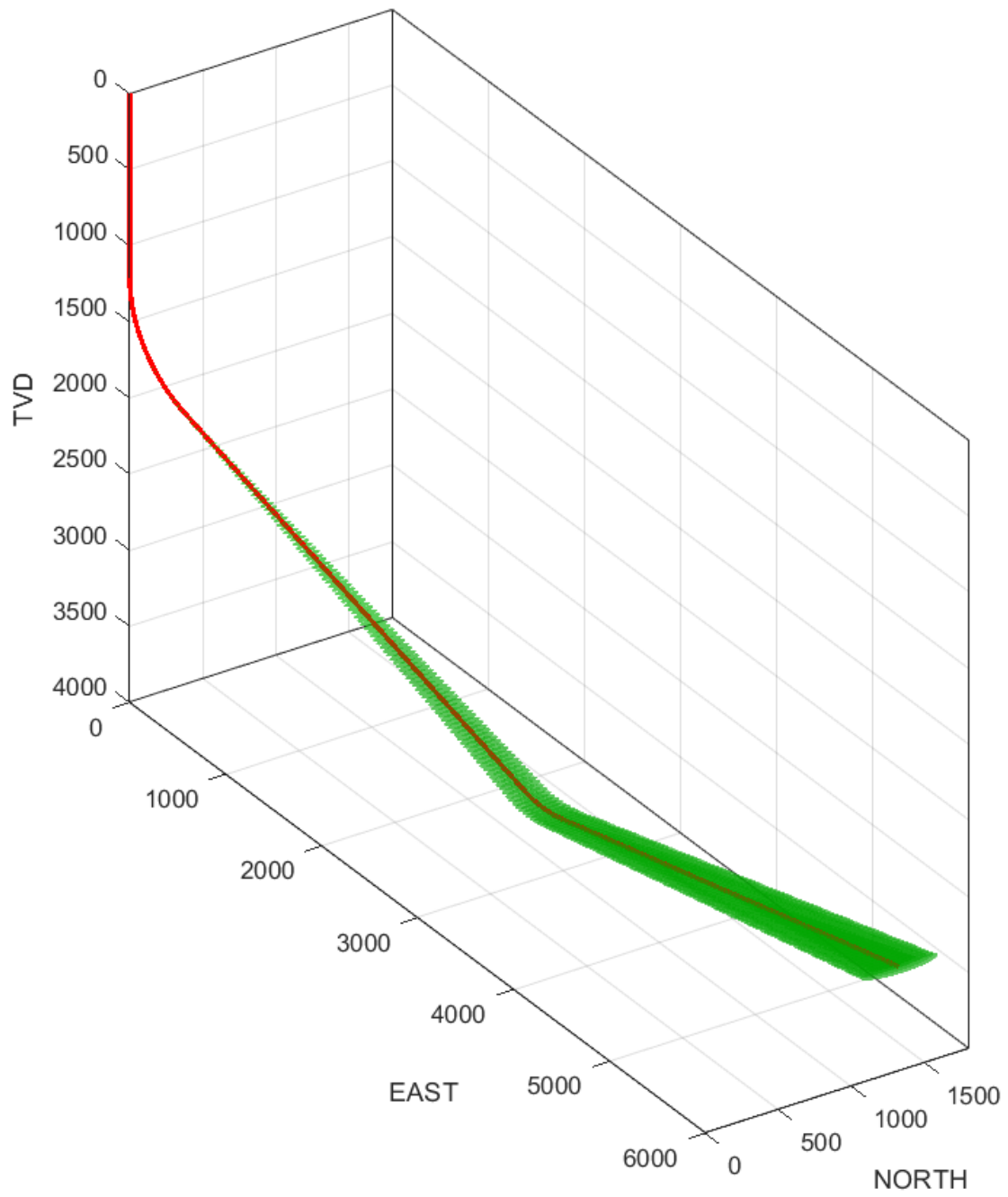
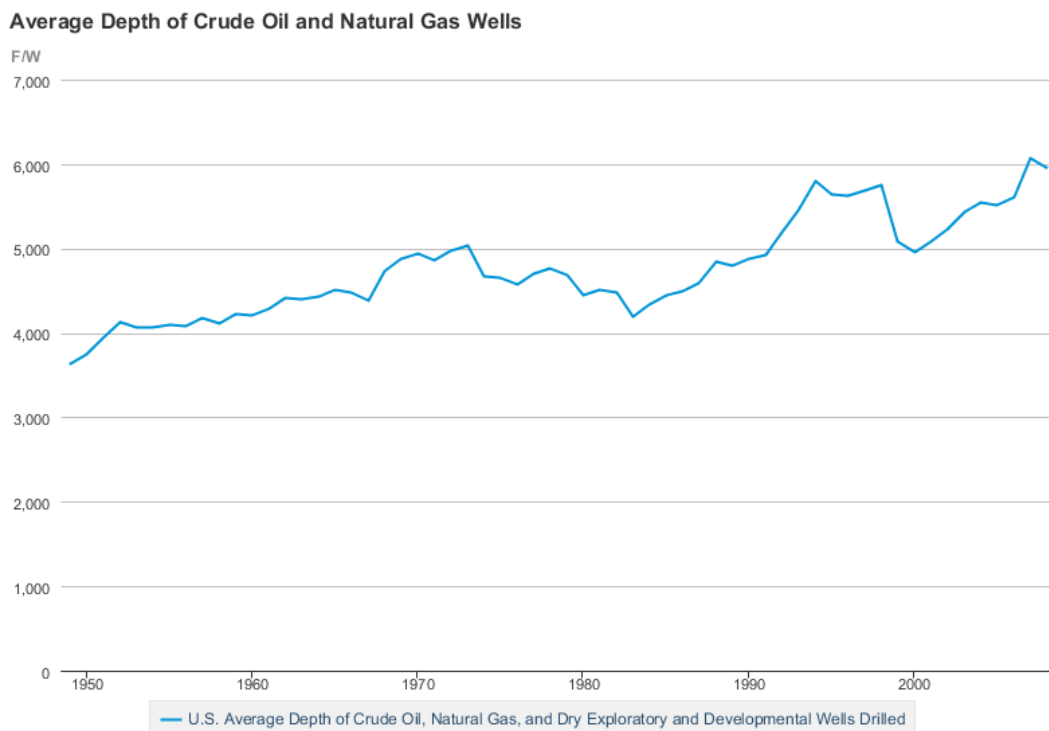


Figure 5.19: Ellipsoid plotted for every survey point, $k = 2.7954$

5.3.1 Discussion: Uncertainty Dimensions

The values for the covariance matrices have been confirmed to be correct, thus should the error ellipses and ellipsoids calculated based on them be. Yet the calculated uncertainty dimensions are above generally accepted values in the industry. From A. Buchanan 2013, [14]: "For typical well depths and step-out, or horizontal reach, the dimensions of the uncertainty envelope may be on the order of 100 ft [30 m] or more unless action is taken to correct error sources and run high-accuracy surveys.". The paper excludes mentioning a range for typical well depths. From the U.S Energy Information Administration [1] following data for average well depths in the period 1949-2008 can be found.



 Source: U.S. Energy Information Administration

Figure 5.20: Average Depth of Crude Oil and Natural Gas Wells [1].

The y-axis shows feet/well, where majority of the depths are between 4000 ft (1219.2 m) to 6000 ft (1828.8 m). If a rough approximation is made, where 6000 ft well could potentially have an error radius of 30 m, assuming this increases linearly. The **ISCWSA Test #1** field with a depth of 3521 m TVD would have maximum radius between 57.8 m - 86.7 m.

From [14], the case of 30 m radius for ellipsoid of uncertainty is regarding wells with insufficient survey-accuracy and error correction. This range is further supported by L. Xiushan, 2019 [42] for following well:

”A horizontal well was surveyed and calculated according to the industry norms, and the calculation results with the true north as the reference direction are shown in Table 1. The well had a geomagnetic field intensity of 56 356.51 nT, magnetic declination of -10.60° , magnetic dip of 64.72° , meridian convergence of 0.876° , and the scaling factor of error ellipsoid of 2.0.” [42].

The well described in [42] has a MD of 10161.32 m and TVD of 8000 m. With total displacement in N-axis 1155 m and E-axis 2000.52 m. The well ends with a horizontal section where inclination is 90° and azimuth 68.76° . For this well, the paper presents following radii along the principal axes at MD 10161.32 m. $R_1 = 27.78$ m, $R_2 = 60.68$ m and $R_3 = 39.16$ m. A direct comparison with data presented in this thesis is difficult. Mainly due to the lack of information regarding the specific tools, quality control of the data, if any pre- or post-processing has been done and what specific weighting functions and magnitudes have been used. These factors may also contribute to the large radii for the calculated ellipsoids and ellipses based on the spreadsheet. The spreadsheet is mainly published to show the calculations and mathematical framework of the ISCWSA methodology to calculate covariance matrices, and not to visualise the EOU. Thus, it can not be confirmed that the data has been a subject of sufficient data quality control that would be expected before position uncertainty is calculated.

Chapter 6

Well Trajectory Design

This chapter will present computer generated well trajectories using the Bézier method [34]. Main focus being optimising well trajectory by minimizing the MD (e.g tortuosity), limiting DLS and changes to azimuth. The Bézier method will be used to generate trajectories based on the mechanical limitations of our rig, and the dimensions of the competition rock.

6.1 Bézier Method

This section will give a brief introduction to the Bézier method. The code for well trajectory optimisation, and visualisation is given in appendix B and C. The core code used is written by E. Elsayed who is part of the UiS Drillbotics Team B, with some modifications made by the writer of this thesis. To obtain the 3D well trajectories used in this case study, second and third order Bézier curves are used, where the second order curve is defined as

$$B(u) = (1 - u)^2S + 2(1 - u)uC_S + u^2E \quad (6.1)$$

And third order,

$$B(u) = (1 - u)^3S + 3(1 - u)^2uC_S + 3(1 - u)u^2C_E + u^3E \quad (6.2)$$

Second order curves are typically used for free end trajectories, where the end point of the trajectory is not defined. While the third order curves are used for instances where the end of the trajectory is set and defined [34]. This section will focus on the third order equation, as this case study makes use of a defined end point. The

parameters in the equations are: u is a dimensionless parameter in the interval $[1,0]$, S is the defined starting point, E is the ending point of the curve, C_S and C_E are the attractors. The dashed-line is the support line of the trajectory, a tangent line to the starting point S or ending point E , going through the attractors, the direction of the dashed line is determined by the inclination and azimuth at S . The distance of the attractor points from S will dictate the direction of the trajectory to E [34].

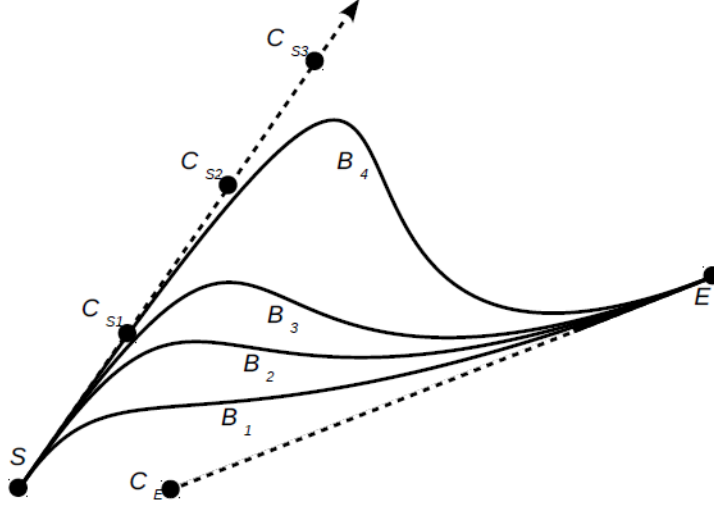


Figure 6.1: Third order Bézier curves with varying C_S and fixed C_E . Attractor C_{S4} is not shown for B_4 [34].

To obtain the well trajectories, some data points must be defined. The initial starting point S must be given as $[v_S, n_S, e_S]$, initial inclination I_s and azimuth A_s . Lastly, the target point of the trajectory, must as S be given in following coordinates $E = [v_E, n_E, e_E], I_E$ and A_E . The inclination and azimuth at any point of the trajectory, is used to determine an unit tangent vector, where t_i can be set to t_S or t_E depending on if one is considering free-end case, or defined-end case [34].

$$t_i = (\cos(I_i); \sin(I_i)\cos(A_i); \sin(I_i)\sin(A_i)) \quad (6.3)$$

This tangent vector and an arbitrary scalar parameter d_i is used to determine the position of the attractor point C_i , which controls the shape and curvature of the trajectory. The position of attractor point C_S with respect to I_S and A_S is given by

$$C_S = S + d_S t_S \quad (6.4)$$

For the defined end point case an additional tangent vector is defined

$$t_E = (\cos(I_E); \sin(I_E)\cos(A_E); \sin(I_E)\sin(A_E)) \quad (6.5)$$

Giving the position of attractor C_E as

$$C_E = E - d_E t_E \quad (6.6)$$

The C_S and C_E are determined by the tangents at each end point of the trajectory (S and E), and their distance to their respective ends provide two degrees of freedom, which are used to control the shape of the 3D trajectory [34]. The code in B are used to find the optimal values for d_S and d_E with respect to DLS, maximum change in azimuth and reducing MD between the start and end-point S and E. This is shown in figure 6.1, where C_E is set, and C_S have varying values, due to varying d_S . Resulting in trajectories with higher or lower degree of curvature. Examples of this in 3D-space will be given later in this chapter. For further reading regarding theory and the mathematical framework of this method, see [34, 35].

6.2 Generating Trajectories

In this section the code will be showcased to the reader, to give further understanding of the theory presented in section 6.1. Where example wells are generated to show what the code is capable of, and its functions. Before generating trajectories for the competition rock and the Drillbotics competition specifically.

6.2.1 Assumptions

The 3D-space the trajectories are generated in, have no physical attributes assigned to it, such as hardness, porosity or permeability. The trajectory is generated in an environment with no geological features, such as faults or salt domes, thus trajectories have no geological features to avoid or interact with. This must be taken into account when the *optimal* trajectory is generated and interpreted, where properties of rock mass and soil or geological features are not considered. The trajectories will not be perfectly applicable in the real world, where such variables have to be considered. The only considerations and limitations of the code, are the initial and finale conditions (e.g coordinates) of the well, inclination and azimuth and finally the DLS limit assigned to a specific trajectory. The code will be used to minimize the MD with respect to these parameters.

6.2.2 Trajectory Examples

First step is to define our start- and end-point, S and E. In this example $S = [v_S, n_S, e_S] = [0, 0, 0]$, and $E = [v_E, n_E, e_E] = [1000, 150, -60]$. Initial and finale inclination and azimuth are both set to $[0, 10]$. Final boundary condition, is the maximum DLS for the trajectory, in this case set to $5^\circ/30\text{m}$. An addition to the code, is the parameter **md0**, which defines the measured depth at our initial vertical depth. An example being, if initial vertical depth is set to 0 m, and **md0** = 100. The measured depth at initial vertical depth of 0 m will be 100 m and not 0 m. For calculations in this example **md0** = 0 m. Final parameter to define is **celln**, which is used to define the desired survey points, set to 30 in this example. The code for this section is given in appendix B, can find the optimal d_S and d_E , in the code these are named **k0** and **k1**.

With set limitations the code calculates the optimal d_S and d_E to be 201 and 251. These are then used to calculate the well trajectory with code given in appendix C. Calculating following trajectory,

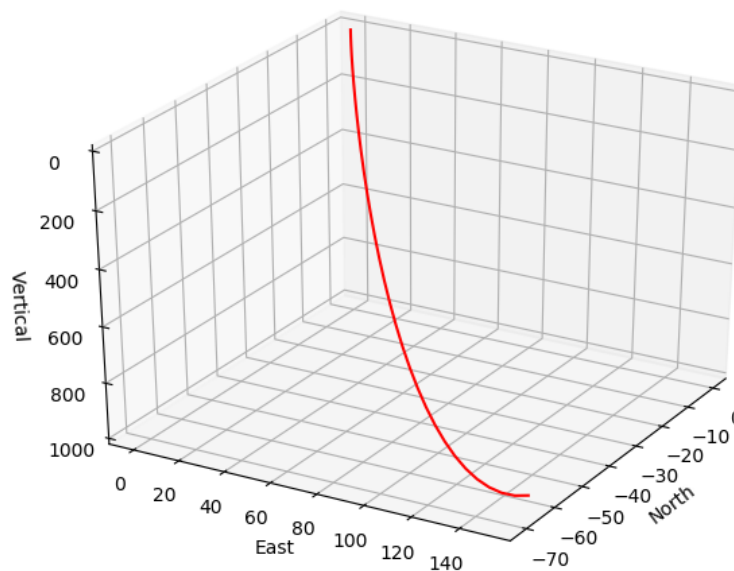


Figure 6.2: 3D-trajectory with d_s and d_e as 201 and 251.

With cross sections,

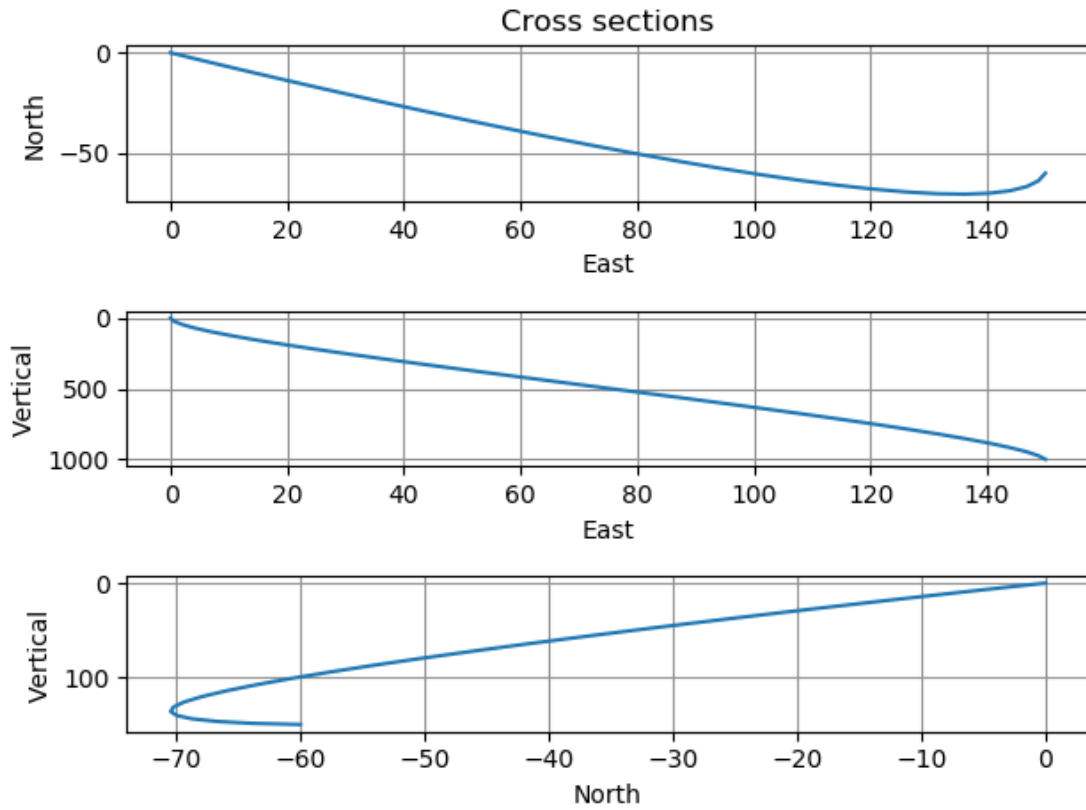


Figure 6.3: Cross sections of the same trajectory.

And survey measurements generated by the code,

	U	v	n	e	md	INC	AZ	dls	TF		U	v	n	e	md	INC	AZ	dls	TF
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	4.984	2.197	16	0.533	521.528	-50.089	79.475	530.584	12.026	-61.889	0.219	357.761
1	0.033	21.245	-0.334	0.465	21.253	2.893	-54.370	3.349	359.995	17	0.567	560.384	-53.883	86.752	570.297	11.814	-63.073	0.272	357.678
2	0.067	44.683	-1.299	1.817	44.750	5.084	-54.607	2.337	359.989	18	0.600	599.103	-57.423	93.930	609.835	11.535	-64.472	0.333	357.632
3	0.100	70.169	-2.839	3.996	70.375	6.777	-54.863	1.680	359.979	19	0.633	637.539	-60.654	100.949	649.040	11.188	-66.151	0.403	357.612
4	0.133	97.556	-4.899	6.939	97.997	8.104	-55.140	1.236	359.965	20	0.667	675.547	-63.521	107.747	687.757	10.771	-68.199	0.485	357.611
5	0.167	126.700	-7.425	10.586	127.477	9.154	-55.440	0.924	359.947	21	0.700	712.980	-65.969	114.262	725.831	10.282	-70.742	0.582	357.628
6	0.200	157.454	-10.361	14.873	158.667	9.988	-55.768	0.698	359.920	22	0.733	749.693	-67.942	120.433	763.111	9.721	-73.973	0.699	357.662
7	0.233	189.673	-13.650	19.741	191.417	10.650	-56.125	0.530	359.883	23	0.767	785.540	-69.385	126.197	799.447	9.094	-78.181	0.843	357.719
8	0.267	223.211	-17.239	25.127	225.574	11.171	-56.518	0.401	359.828	24	0.800	820.376	-70.243	131.494	834.694	8.415	-83.815	1.024	357.803
9	0.300	257.923	-21.072	30.970	260.982	11.572	-56.951	0.300	359.744	25	0.833	854.056	-70.460	136.261	868.710	7.725	-88.443	1.255	357.926
10	0.333	293.662	-25.094	37.207	297.484	11.872	-57.430	0.222	359.611	26	0.867	886.432	-69.981	140.437	901.358	7.111	-77.648	1.554	358.105
11	0.367	330.284	-29.249	43.778	334.922	12.083	-57.963	0.163	359.390	27	0.900	917.361	-68.750	143.961	932.511	6.748	62.921	1.950	358.353
12	0.400	367.642	-33.482	50.620	373.137	12.213	-58.561	0.126	359.030	28	0.933	946.695	-66.714	146.770	962.050	6.920	44.690	2.484	358.663
13	0.433	405.591	-37.738	57.673	411.970	12.268	-59.235	0.118	358.561	29	0.967	974.290	-63.815	148.804	989.871	7.941	25.894	3.216	358.984
14	0.467	443.986	-41.961	64.874	451.262	12.254	-60.000	0.137	358.160	30	1.000	1000.000	-60.000	150.000	1015.890	10.000	10.000	4.240	359.254
15	0.500	482.680	-46.096	72.162	490.853	12.173	-60.876	0.173	357.907										

Figure 6.4: Table with data from the survey stations.

From figure 6.4 one can see that our initial- and end-point conditions are met. The DLS stays below the $5^\circ/30\text{m}$ limit set initially. To further illustrate how d_S and d_E affect the trajectory, three more trajectories are visualised and presented below. These have the same initial and finale conditions as the trajectory just presented. But d_S and d_E values are changed manually, without considering how realistic the trajectory is, but rather to showcase the planned trajectory with varying values for d_S and d_E . Figure 6.5 clearly shows the effect of changing parameters d_S and d_E beyond the optimal values. With case (a) being the optimal case previously explained, it is the benchmark that cases (b),(c) and (d) are compared against.

For case (b), from observation alone, it is clear that the trajectory has a higher value for MD, where the trajectory moves further down the North axis, negative values indicating movement in south direction. It is first when the parameters for case (c) are applied that trajectories beyond what is practical are generated. Trajectory (c) is observed from a different angle in figure 6.6. In the interval 0 m to 245 m along the vertical axis, an inclination of 98° is achieved. At this point the eastern and northern coordinates are 29 m and -61 m. The trajectory continues to build an angle up to 121.7° before dropping of. At the inclination of 121.7° , the vertical depth is 214 m. The finale section is defined by a sharp declination down to the specified end-point of [1000,150,-60].

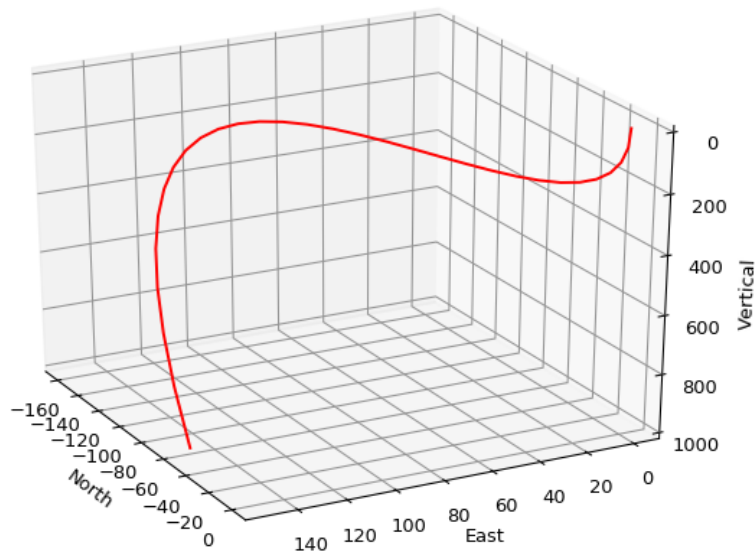
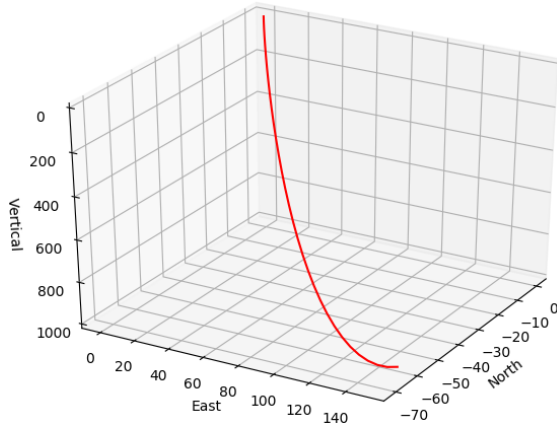
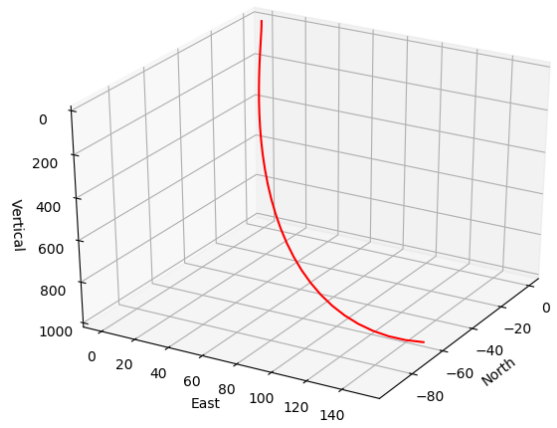


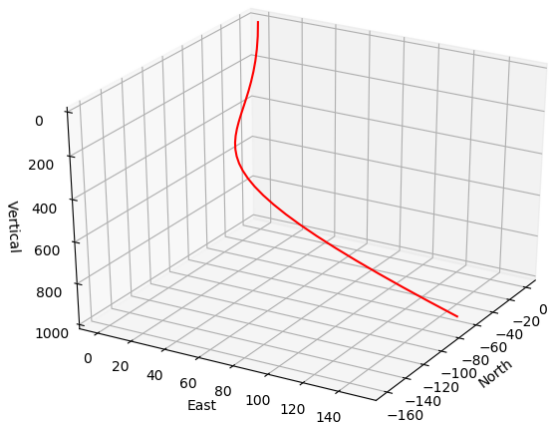
Figure 6.6: Trajectory (c) from a different viewpoint.



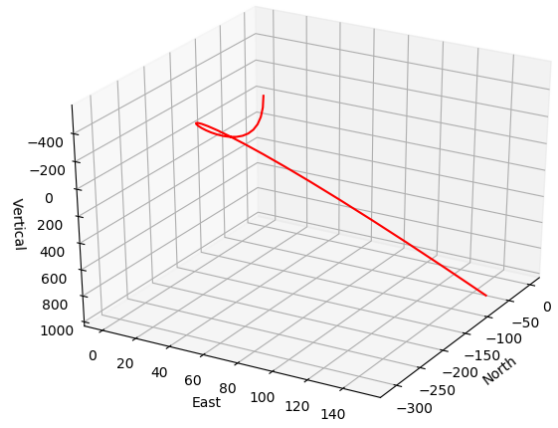
(a) $d_S = 201$ and $d_E = 251$



(b) $d_S = 200$ and $d_E = 600$



(c) $d_S = 700$ and $d_E = 1500$



(d) $d_S = 1000$ and $d_E = 3500$

Figure 6.5: Trajectories with the same initial and final conditions as previously defined, this time with varying d_S and d_E .

The major challenge with trajectory (c) is the DLS throughout the wellpath. With values ranging from $0-78^\circ/30\text{m}$, the swift changes in inclination over short intervals are beyond what is realistic for a well trajectory. The trajectory also adds to the total MD compared to the optimal case (a). Trajectory (d) continues the trend set by (c). With a section where inclination $> 90^\circ$, at the highest 163° . The direction of the well changes from positive values along the vertical-axis to negative. The lowest depth along the vertical-axis is -568 m , in practice this mean drilling 568 m above set datum plane.

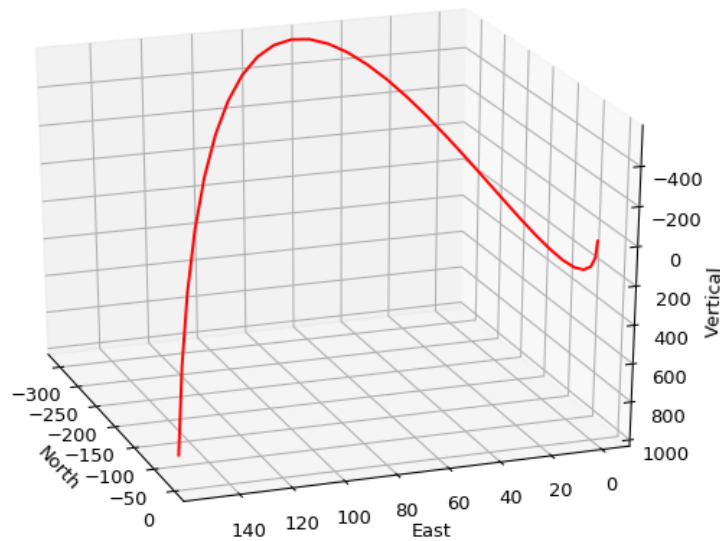
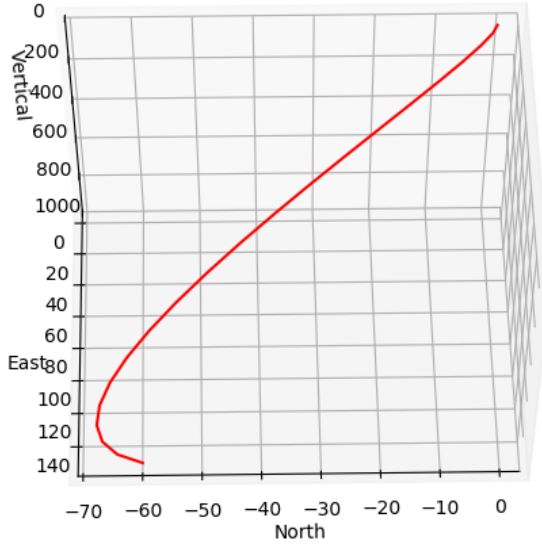


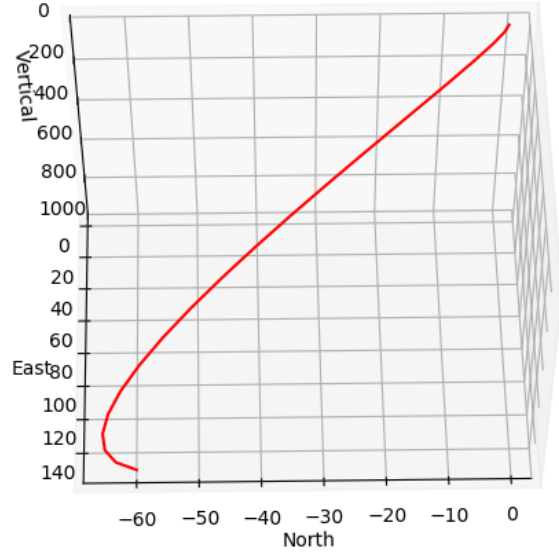
Figure 6.7: Trajectory (d) from a different viewpoint.

Figure 6.7 shows upwards trend of trajectory (d). This dramatically increases the inclination and MD. Compared to the optimal case (a), where the total measured depth was 1015 m , whereas MD for (d) is 2616 m . This further highlights the importance of finding the optimal values for d_S and d_E . As mentioned in assumptions for this subject, these trajectories are not necessarily applicable in real life. This can be due to limitations of equipment, geological features or rock properties. But the idea is to see what is theoretically possible, then applying the overall trend of generated trajectories.

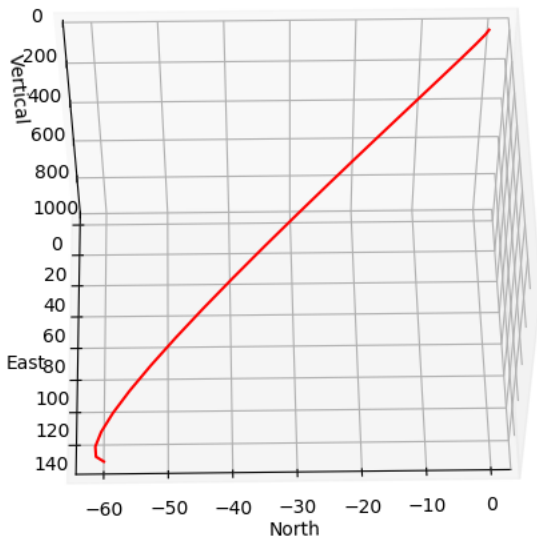
There is yet another option with the code. That is manually choosing values for d_S and d_E that are below, the calculated optimal values. This results in a lower degree of curvature, and the tendency for straight line between initial and end point increases with lower values. This is shown on figure 6.8.



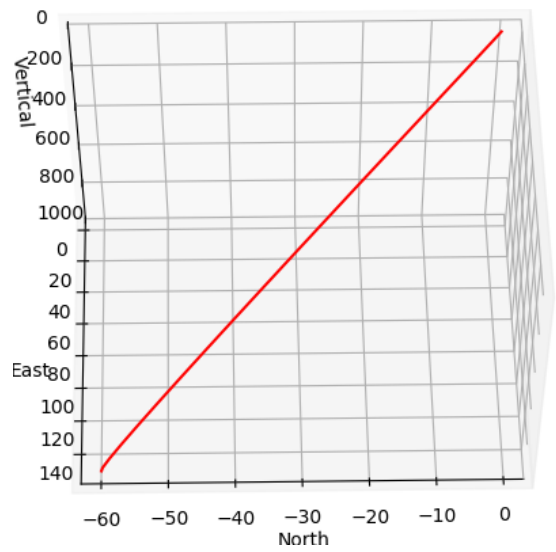
(e) $d_S = 180$ and $d_E = 200$



(f) $d_S = 140$ and $d_E = 160$



(g) $d_S = 50$ and $d_E = 70$



(h) $d_S = 5$ and $d_E = 10$

Figure 6.8: Same trajectory as in fig 6.5, but with d_S and d_E lower than optimal values.

The third order Bézier curve staying strictly between S and E when assigned lower d_S and d_E values, can be explained mathematically. If we revisit the quadratic Bézier equation from [34].

$$B(u) = (1 - u)^3 S + 3(1 - u)^2 u C_s + 3(1 - u) u^2 C_e + u^3 E \quad (6.7)$$

If we assume that the initial position S and finale position E are constant and u is set to be in the interval [0,1]. What is left to define is C_S and C_E . From following equations,

$$C_S = S + d_S t_S \quad (6.8)$$

$$C_E = E - d_E t_E \quad (6.9)$$

Only parameters that can be changed independently are d_S and d_E , as the tangential unit vector t_S and t_E are functions of A_S , A_E , I_{E_S} and I_E which also are constant as those parameters are defined for initial and finale point. Thus, smaller values of d_S and d_E will result in C_S and C_E that will be closer to S and E. Inserting these in the cubic equation will yield increasing values for B(u) in the interval [S,E] where u = [0,1], the opposite being, large values of d_S and d_E will cause fluctuations in C_S and C_E in the interval u = (0,1), which may result in values for B(u) outside the interval [S,E]. Nonetheless both cases will yield S for u = 0, and E for u = 1.

Chapter 7

Case Study: Trajectory Optimisation and Position Uncertainty for Laboratory Drilling Rig

7.1 Assumptions and Limitations

When using this method for the UiS Drillbotics rig, there are two main limitations to consider. The first one is the mechanical and technical limitations of the rig, and the second being the dimensions of the competition rock. In addition to the physical limitations, the framework of the guidelines will be discussed. This section will give an overview over the limitations to consider, and the effect of these on the trajectory optimisation process.

7.1.1 Competition Rock and Guidelines

As presented in 1.2.1, the dimensions of the homogeneous sandstone are set to (30 x 60 x 60) cm. For drilling, the rock will be resting on the 30 cm x 60 cm face, giving a total depth of 60 cm. In 1.2.1 some physical properties of the rock were presented, which will not be considered when trajectories are generated. From previous experiments drilling in cement samples as a substitute for sandstone, and drilling in the competition rock 2019. The experience is that sandstone is easy to drill in and the rig can obtain high values for ROP. Based on vertical drilling from 2018 [18], the rig could achieve ROP of 4.454 cm/min in cement rock samples. From this experience, a benchmark of what constitutes a realistic trajectory will be

defined, as no drilling has been conducted due to the worldwide pandemic in 2020.

In [29] documentation of an experiment using concrete is presented. The concrete sample made using Portland cement is used for deviated well drilling with the whipstock. When drilling was completed, and inspection of the rock was made. It was discovered that 55 mm horizontal build was achieved along the east-axis over a TVD of 490 mm [29]. Unintentionally, during the experiment displacement along the north-axis of approximately 10 mm was observed, potentially due to connections in knuckle-joint or BHA experiencing a twist or loose connections.

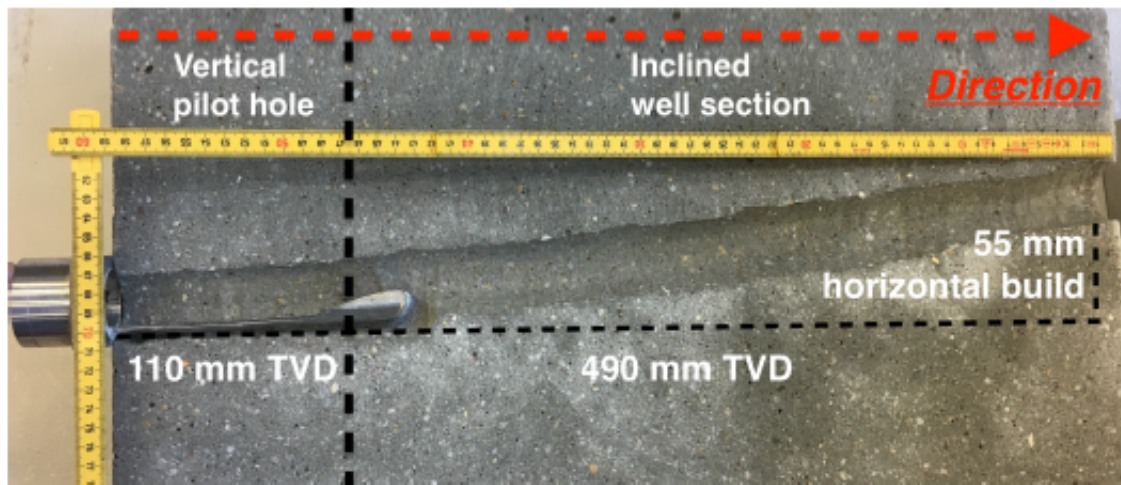


Figure 7.1: Well profile using 7° whipstock in cement [29].

For directional drilling, the ROP in the same cement is reduced to 0.833 cm/min [29]. If we use this ROP as the benchmark, and multiply it with the amount of time that is allowed for drilling according to the guidelines, 3 hours or 180 min [13]. All trajectories shorter than 150 cm in MD should be theoretically possible. This assumes that a ROP of 0.833 cm/min is kept constant for 180 minutes of the competition, which is unlikely. But rather gives an approximation of what the maximum measured depth of a trajectory can be. Figure 7.1 can be presented as a simple schematic.

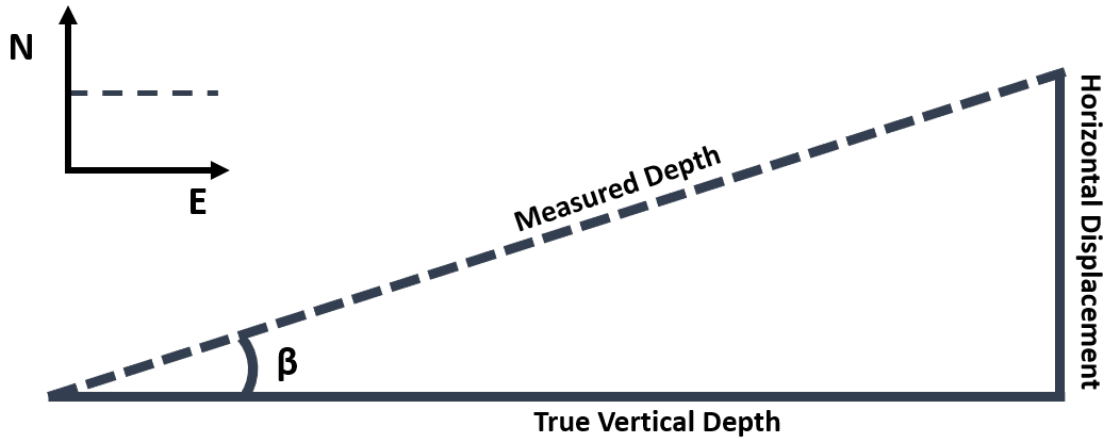


Figure 7.2: A simple representation of a well after kick-off point with fixed angle.

Figure 7.2 is a simple representation of the well after KOP. According to the guidelines, deviated drilling can start at any point below or at the defined 4" (101.6 mm) vertical section. It assumes no change in azimuth (e.g stays constant along the north-axis). And that the angle β at KOP stays constant throughout the well. With simple trigonometry we can calculate multiple values for measured depth and horizontal displacement, with a set TVD and varying β . If KOP is right after the 101.6 mm vertical section. The TVD used in table 7.1 is 498.4 mm.

β [°]	MD [mm]	Δh [mm]	Total MD [mm]
7.000	502.143	61.196	603.743
8.000	503.298	70.046	604.898
9.000	504.613	78.939	606.213
10.000	506.089	87.881	607.689
12.000	509.535	105.938	611.135
14.000	513.658	124.265	615.258

Table 7.1: Length of MD shown in figure 7.2, with varying β . Total MD is including the vertical section.

The total MD values in table 7.1 are all well below 150 cm. Taking into account that these calculations are based on the KOP being exactly at a depth of 101.6 mm, and that the angle at KOP stays constant. These calculations are approximations intended to show that most wells within the guidelines are below the set 150 cm limit.

7.1.2 Mechanical Limitations

The main mechanical components to consider, are the downhole tools. Mainly the BHA-assembly. It consists of three sections, a top, mid and bottom section. The dimensions of the three parts given in following figures 2.3, 2.4 and 2.5. The metal and total length of the BHA limits the mobility of the drilling rig. The design is optimised for a well trajectory which favours straight sections with constant inclination. The static angle for the bent section at 8° will be the deciding factor horizontal displacement by just rotating the downhole motor. For change in azimuth both the pneumatic and top-drive will rotate simultaneously. This method of drilling is not time efficient and limits the potential ROP of the system. Considering these factors, a trajectory should be designed by minimizing change in azimuth for time efficiency.

7.2 Trajectories: Competition Rock

To drill the vertical section of the rock, both the top-drive and pneumatic will rotate simultaneously. When a TVD of 101.6 mm is reached, the top drive will be turned off and drilling will commence using the downhole pneumatic motor. The simplest and preferred trajectory, is a trajectory with a kickoff point at the 101.6 mm mark, followed by a linear build section down to the finale depth. With close to no change in azimuth and movement along the north-axis. Generating wells for the competition rock will be done by scaling up the dimensions, due to the code being optimised for integers and not floats. All real values in meters will be multiplied by 10^4 for the code. Example: dimensions of the rock will be set to (3000 x 6000 x 6000) m in the code. Likewise, to recreate a horizontal displacement of 70 mm, this will be equal to 700 m in the code.

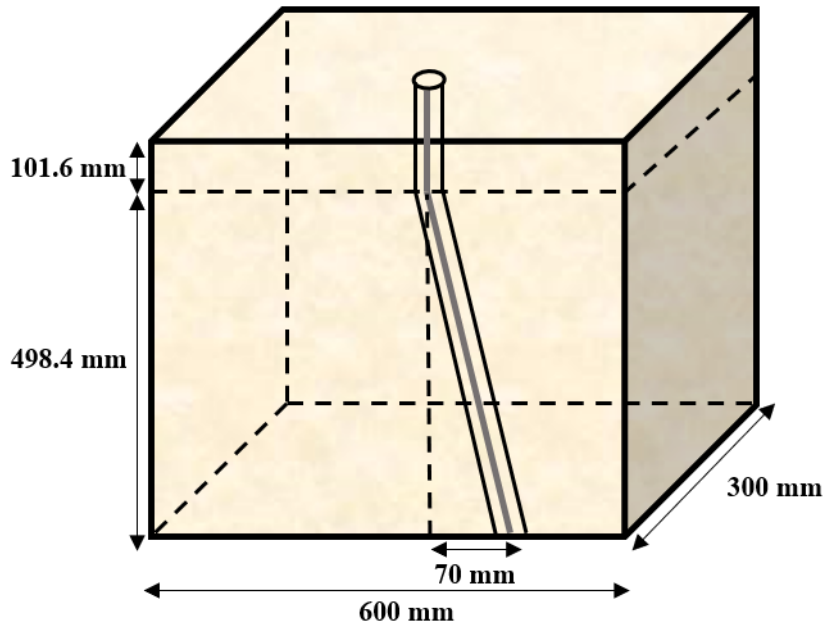


Figure 7.3: The simplest form of trajectory, with no change in azimuth and a small displacement along the east-axis.

7.2.1 Definition of Trajectory Parameters

The parameters of the trajectory are defined as, $S = [v_S, n_S, e_S] = [0, 0, 0]$ and $E = [4984, 100, 700]$. A change along the north-axis is added, based on experience from previous year. A few of the drilled deviated wells have some deviation along the north-axis. In some cases due to problems related to loose connection or vibrations. Thus, change in azimuth must may be considered. Initial and finale inclination are both set to 8 degrees, and initial and finale azimuth are set to 90 degrees. At v_S the parameter m_d is set to 1016 m. Survey points are set to 20, and upper DLS limit is $8^\circ/30m$.

7.2.2 Simple Trajectories

With given values, the optimal d_S and d_E are both calculated to be 120. Resulting in following trajectory.

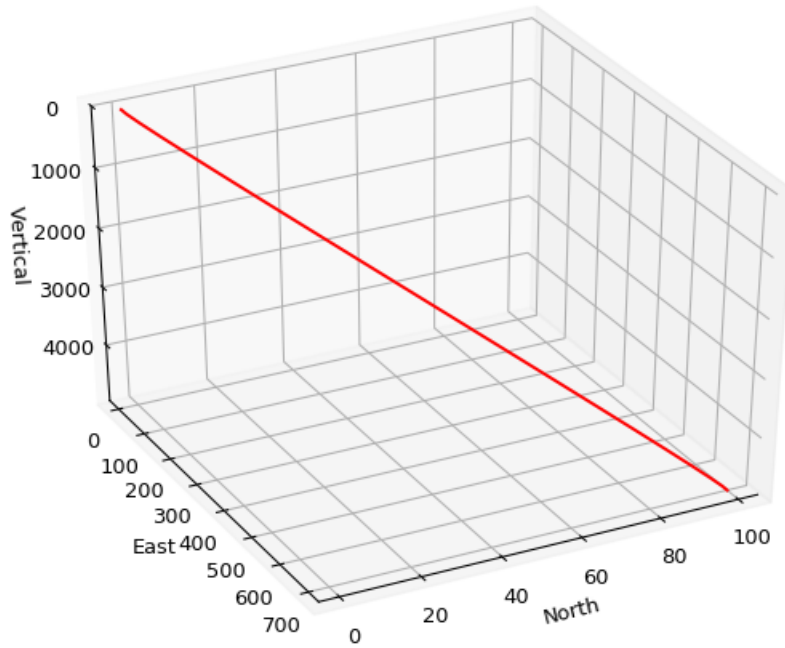


Figure 7.4: $d_S = d_E = 120$

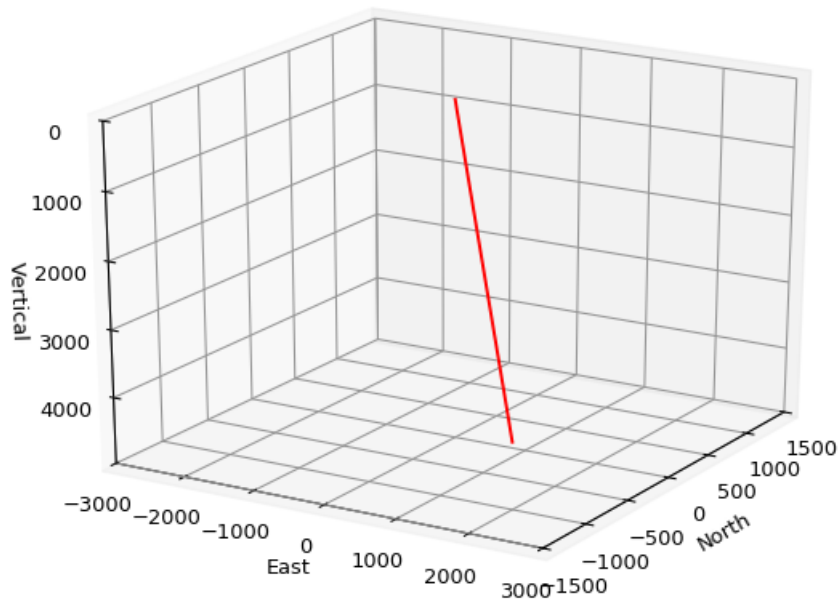


Figure 7.5: Same as figure 7.4, but with axes limits equal to that of the competition rock.

Figure 7.5 is to emphasize that the displacement along the north-axis is minimal, relative to the dimensions of the competition rock. The total measured depth of this trajectory was 604.991 mm when scaled down to millimeters, nearly identical to the 604.894 mm, calculated using a constant angle of 8 degrees and no change in azimuth. The slight increase in MD is due to the displacement along the north-axis. A trajectory of 604.991 mm in MD is well within what the rig is capable of with respect to the ROP and time limit of the competition.

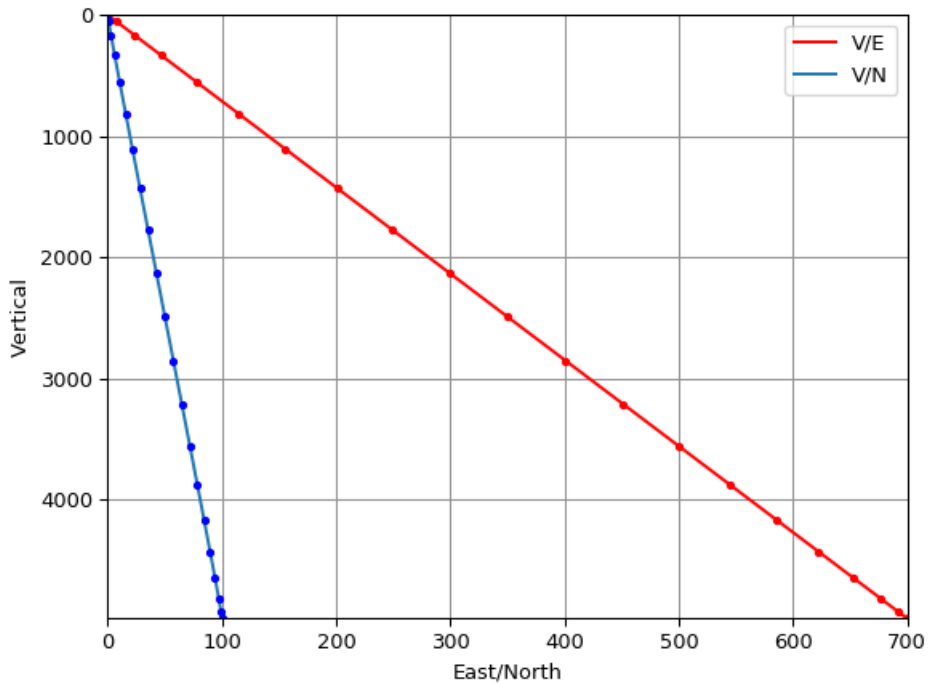


Figure 7.6: V/E and V/N plotted with survey points, $d_S = d_E = 120$.

The MD can be further reduced by assigning d_S and d_E with lower values, and having no displacement along the north-axis. The new trajectory shown on figure 7.7 is a straight line from our zero-point, where MD is 101.6 mm and down to finale vertical depth of 498.4 mm. The finale total measured depth is 604.892 mm. Compared to the previous trajectory with 10 mm displacement along the north-axis and MD of 604.991 mm, the decrease in MD is insignificant. All these values were scaled down to millimeters.

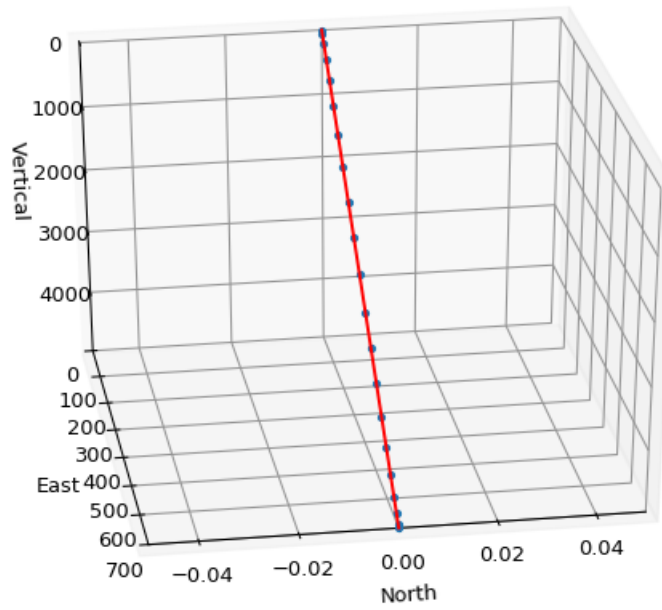


Figure 7.7: $d_S = d_E = 3$.

	U	v	n	e	md	INC	AZ
0	0.00	0.000	0.0	0.000	1016.000	8.000	90.0
1	0.05	37.404	0.0	5.253	1053.771	7.995	90.0
2	0.10	141.691	0.0	19.901	1159.082	7.995	90.0
3	0.15	305.429	0.0	42.898	1324.427	7.995	90.0
4	0.20	521.188	0.0	73.201	1542.304	7.995	90.0
5	0.25	781.535	0.0	109.766	1805.206	7.995	90.0
6	0.30	1079.039	0.0	151.551	2105.630	7.995	90.0
7	0.35	1406.270	0.0	197.510	2436.072	7.995	90.0
8	0.40	1755.794	0.0	246.600	2789.027	7.995	90.0
9	0.45	2120.181	0.0	297.778	3156.991	7.995	90.0
10	0.50	2492.000	0.0	350.000	3532.459	7.995	90.0
11	0.55	2863.819	0.0	402.222	3907.927	7.995	90.0
12	0.60	3228.206	0.0	453.400	4275.890	7.995	90.0
13	0.65	3577.730	0.0	502.490	4628.845	7.995	90.0
14	0.70	3904.961	0.0	548.449	4959.287	7.995	90.0
15	0.75	4202.465	0.0	590.234	5259.711	7.995	90.0
16	0.80	4462.812	0.0	626.799	5522.613	7.995	90.0
17	0.85	4678.571	0.0	657.102	5740.489	7.995	90.0
18	0.90	4842.309	0.0	680.099	5905.834	7.995	90.0
19	0.95	4946.596	0.0	694.747	6011.145	7.995	90.0
20	1.00	4984.000	0.0	700.000	6048.916	8.000	90.0

Figure 7.8: Survey measurements for trajectory 7.7.

7.2.3 Complex Trajectories

In this paper, complex trajectories are defined to be: trajectories which have one or multiple target points between start-point S and end-point E. This is how the competition was set to be, where on the day of the competition teams would be given one or multiple [X,Y,Z] coordinates to hit before exiting the rock. The position of these target points must be within a reasonable interval, based on the limitations of the drilling rig. The old guidelines for the 2018-2019 competition will be used to get a general idea of the deviation from [0,0,0] that can be expected.

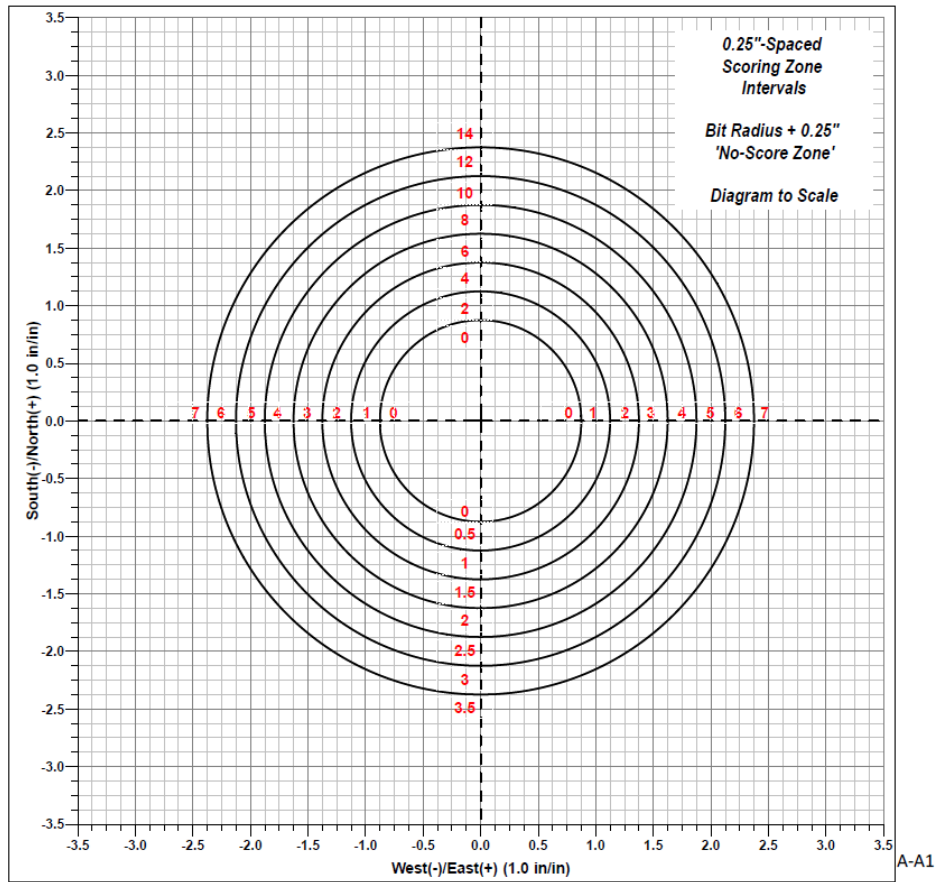


Figure 7.9: Scoring system used for the 2018-2019 competition [12].

Figure 7.9 (2018-2019 guidelines) shows the maximum score that can be obtained for deviation along the north-south axis and east-west axis. For both axes the maximum score is given in interval $\pm 2.4''$, which in millimeters is ± 60.96 mm. Negative displacement in the vertical-axis is not mentioned in the guidelines, thus the interval for the vertical-axis is set to [0,498.4] from the KOP. The complex trajectories are

generated and visualised in parts. For example: $[X_S, Y_S, Z_S]$ to $[X_i, Y_i, Z_i]$, $[X_i, Y_i, Z_i]$ to $[X_{i+1}, Y_{i+1}, Z_{i+1}]$ and $[X_{i+1}, Y_{i+1}, Z_{i+1}]$ to $[X_E, Y_E, Z_E]$. Between every iteration, the survey data is exported to an Excel file, where all the survey data is collected. Finally, using a simple python script. The data is pulled from the Excel file and used to visualise the full trajectory.

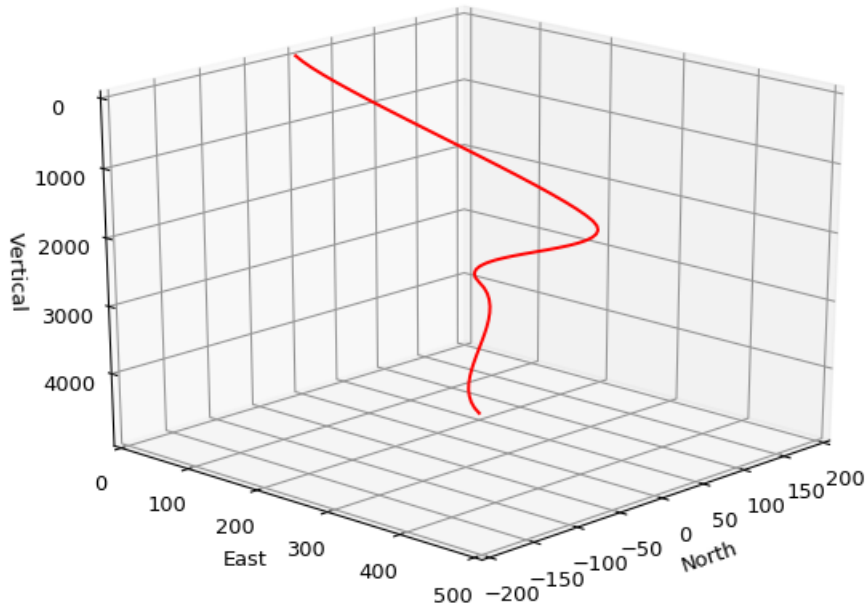


Figure 7.10: Unoptimised complex trajectory with two target points.

Trajectory 7.10 was generated with no regards to optimal d_S and d_E , in this case those were set to $d_S = d_E = 120$. This trajectory was generated with target points [E,N,V] being $[300, 100, 2000]$ and $[200, 50, 3000]$, with end-point $[200, 50, 4984]$. Purpose of this trajectory was to test the capabilities of the code with regards to using it for optimisation between target points and a chosen exit point. Total MD of this trajectory was 604.007 mm.

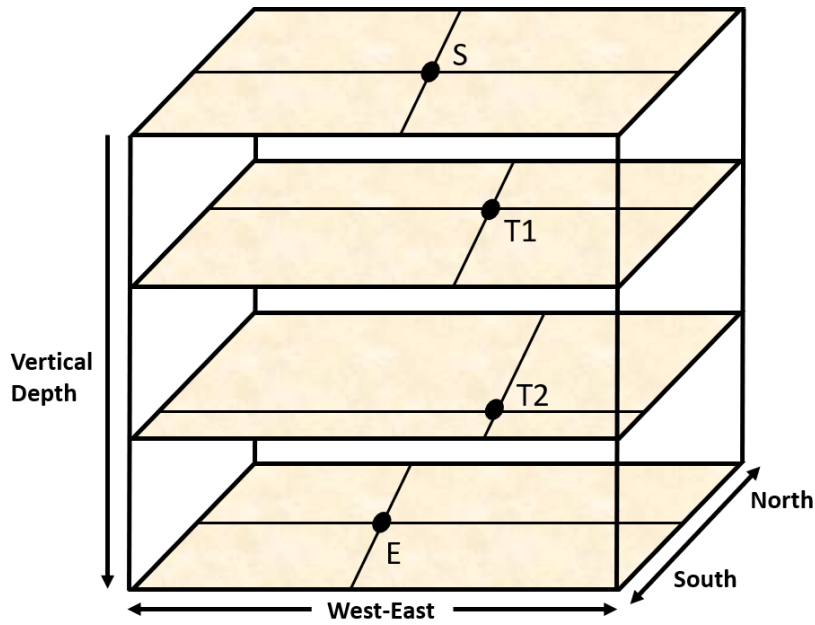


Figure 7.11: Visualisation used for designing complex trajectories

When designing a complex trajectory, one can imagine each point on a isolated layer in the rock. This is visualised in figure 7.11. Where S and E are start and end-points and T1 and T2 are target points 1 and 2. For this figure only two targets points were chosen. Each point is defined by $[E,N,V]$ coordinates on that layer. The trajectory must go through these selected target points. The figure shows "West-East" and "North-South", where "West" and "South" are indicated by negative values on their respective axes.

Table 7.2 shows a planned well with target point which a trajectory will be generated for. Between S and E, three targets point have been defined. The code will be run a total of four times, the intervals are S-T1, T1-T2, T2-T3 and T3-E. For every interval the optimal d_S and d_E will be obtained and used for trajectory generation. **md0** for the first interval is 1016 m, for the second interval **md0** will be set to the total MD of previous interval. Same procedure is repeated for all intervals.

	E	N	V	I1	I2	A1	A2
S	0	0	0	8	8	0	45
T1	300	300	1500	8	9	45	60
T2	450	400	3000	9	9	60	75
T3	600	500	4000	9	9	75	80
E	600	600	4984	9	9	80	80

Table 7.2: Target points.

All trajectories are generated using optimal d_S and d_E , given in table 7.3.

	d_S	d_E
S-T1	211	181
T1-T2	91	91
T2-T3	71	91
T3-E	161	161

Table 7.3: d_S and d_E values for the different intervals

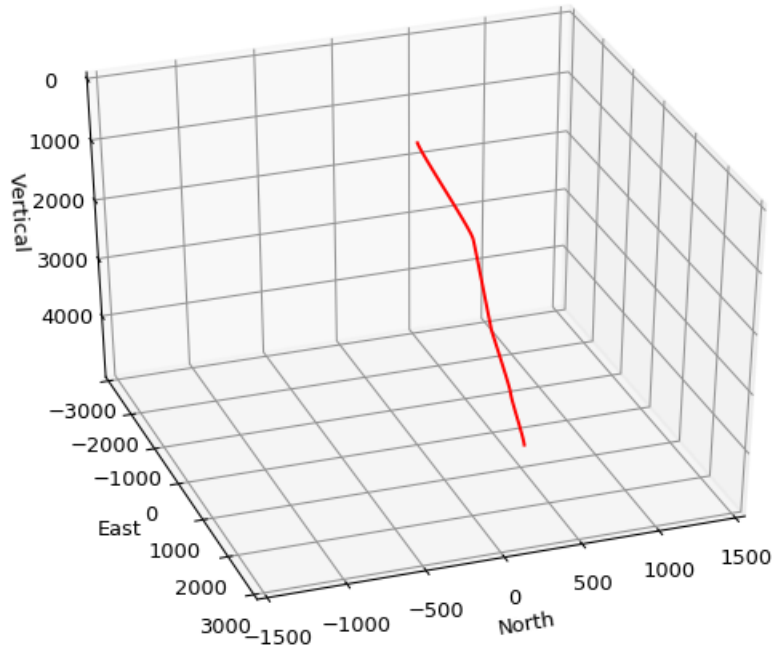


Figure 7.12: Complete trajectory, axes limits scaled up to that of the competition rock.

Figure 7.12 shows the complete trajectory starting at S, going through all three target points and ending at E. A noticeable change in direction is observable at point T1 at 1500 m. This is clearly seen on figure 7.13 as well. The MD of the well, including the vertical section down to 1016 m, is 6092.825 m. When scaled down to millimeters, this is equal to 609.2825 mm. Approximately 5 mm longer than previous trajectories. This is due to displacement along North and East-axis that corresponds to the maximum displacement expected, based on the guidelines 2018-2019 [12].

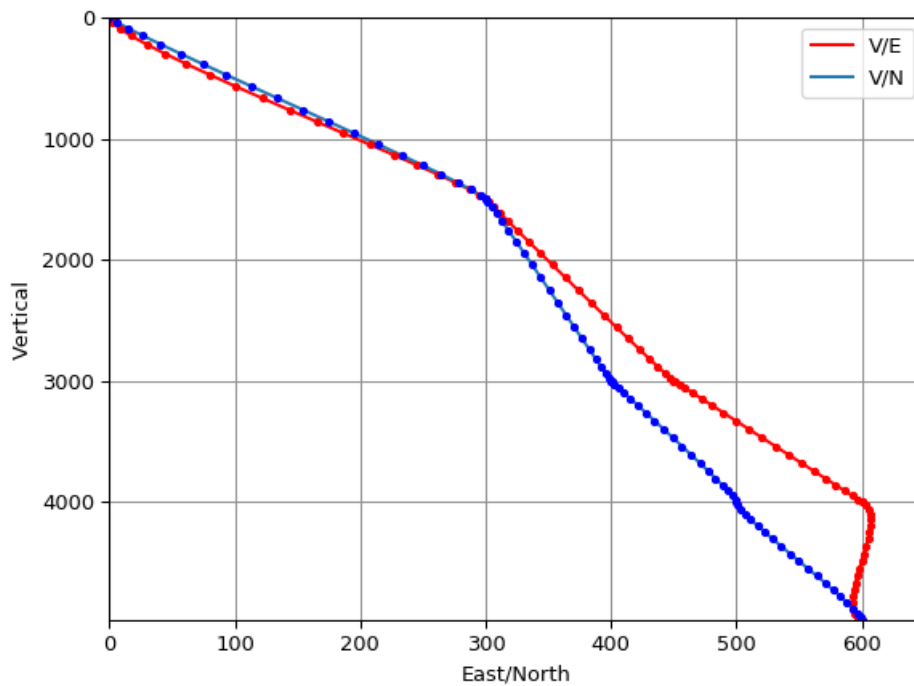
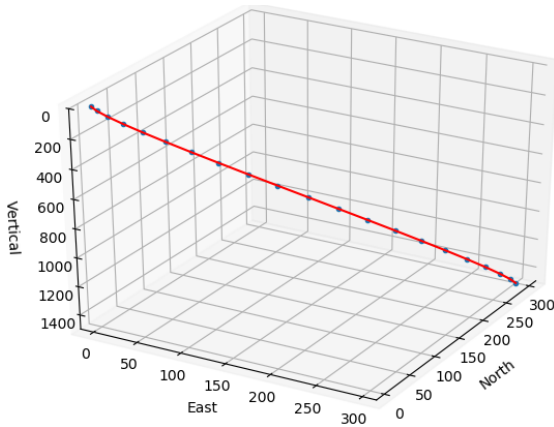
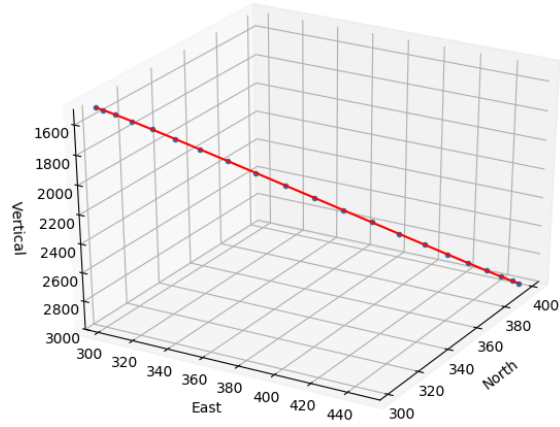


Figure 7.13: Trajectory 7.12 from V/E and V/N perspective.

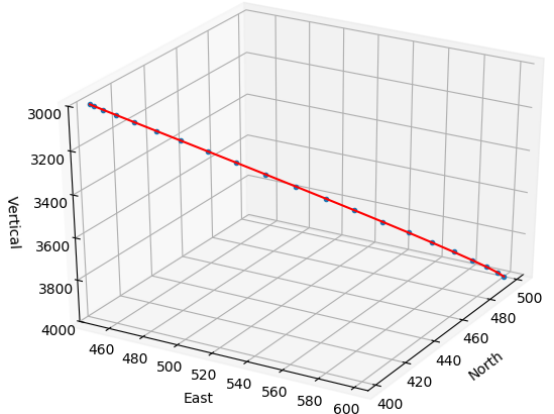
Section T3-E is characterised by its curvature, shown on figure 7.14. The first half of the trajectory is above 600 m on the East-axis, then it turns around and moves under the 600 m on the same axis, finally it moves along the positive direction of the East-axis and ends on 600 m. All values for this section, including the DLS are within the defined limits. As the trajectory is generated in 3D space with no physical attributes, trajectories with high degree of curvature must be discussed. Section T3-E would require the BHA-assembly to rotate 180° twice. This is not time efficient or a practical solution considering the technical capabilities of the rig.



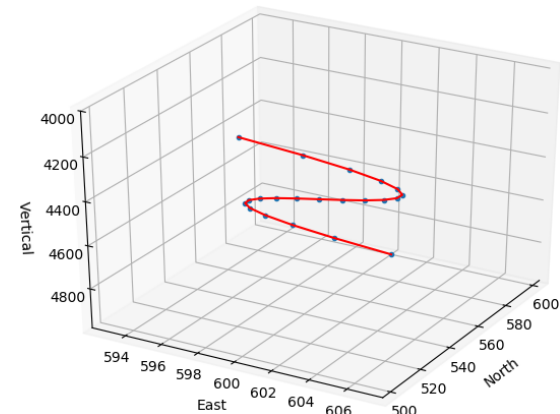
(S-T1) $d_S = 211$ and $d_E = 181$



(T1-T2) $d_S = 91$ and $d_E = 91$



(T2-T3) $d_S = 71$ and $d_E = 91$



(T3-E) $d_S = 161$ and $d_E = 161$

Figure 7.14: Trajectory 7.12 in sections.

To achieve a linear and practical trajectory for the T3-E section. The eastern-coordinates for T3 and E can be changed to increase or decrease in one direction. Else, if the eastern-coordinates are set to stay constant at T3 and E, as the northern-coordinates increase, a trajectory with a S-shape is inevitable. This can be done, by changing the E coordinates to $[E,N,V] = [700,600,4984]$. Only change applied is the increase in displacement along the eastern-axis with 100 m. No changes have been applied for S, T1, T2 and T3. The new trajectory for T3-E on figure 7.15 is a more time efficient option, compared to the shorter alternative which required a 180 degree rotation of the BHA. The total trajectory from S to E with new T3-E section has a MD of 6097.141 m or 609.714 mm when scaled down. An increase of approximately 0.5 mm, insignificant compared to overall time efficiency.

	E	N	V	I1	I2	A1	A2
S	0	0	0	8	8	0	45
T1	300	300	1500	8	9	45	60
T2	450	400	3000	9	9	60	75
T3	600	500	4000	9	9	75	80
E	700	600	4984	9	9	80	80

Table 7.4: Targets point with new E-point.

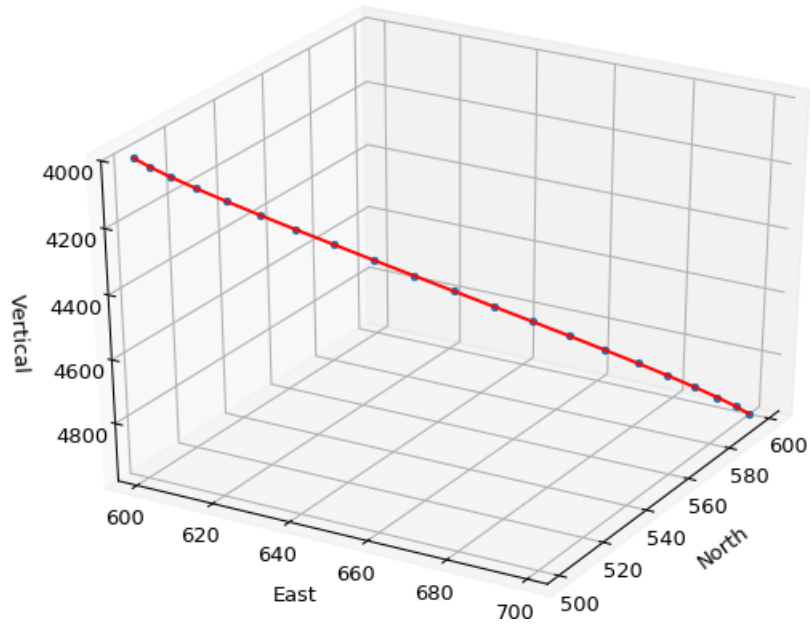


Figure 7.15: Improved (T3-E): $d_S = d_E = 121$

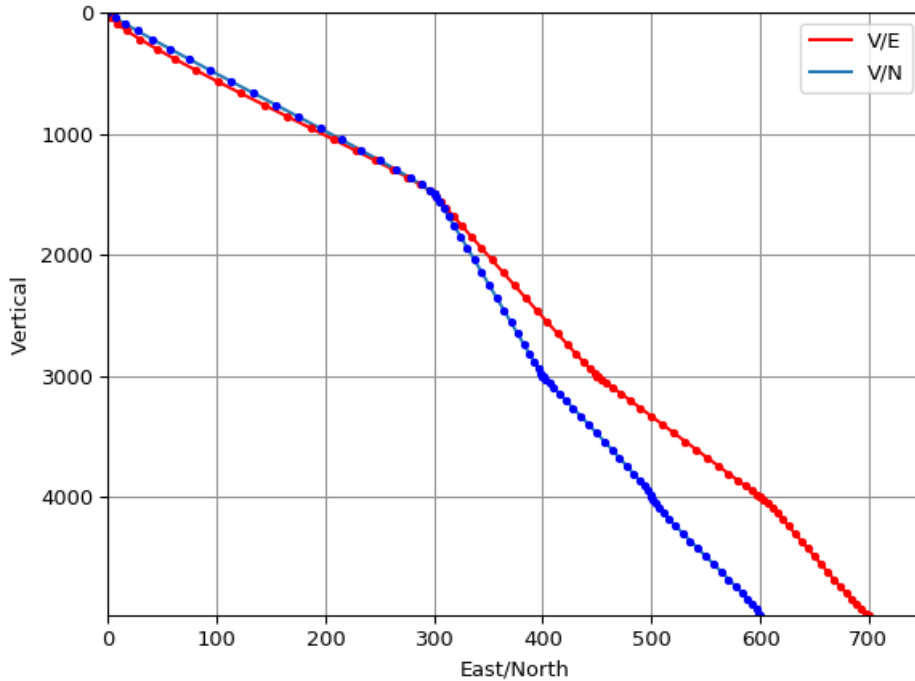


Figure 7.16: Trajectory 7.12 from V/E and V/N perspective with new T3-E section.

7.3 Implementation of Trajectory Optimization

The method presented in this chapter can be implemented and used for planning of a trajectory. The user must take into account that generated trajectories do not consider the physical properties of the rock or the capabilities of the rig. Thus, the trajectories generated are to be used as an approximation of what the finale drilled well might look like. The survey measurements generated by the code, can be used as a template to follow and for confirmation that the drilling is commencing according to the generated survey measurements. The code can be implemented as a separate section in the GUI before drilling is initiated. The person responsible for the drilling session, will be assigned to define the target points. The GUI must be intuitive and give the user the choice between a simple or complex trajectory. Simple trajectory requiring to define S and E, together with values for azimuth, inclination, DLS, survey points and **md0**.

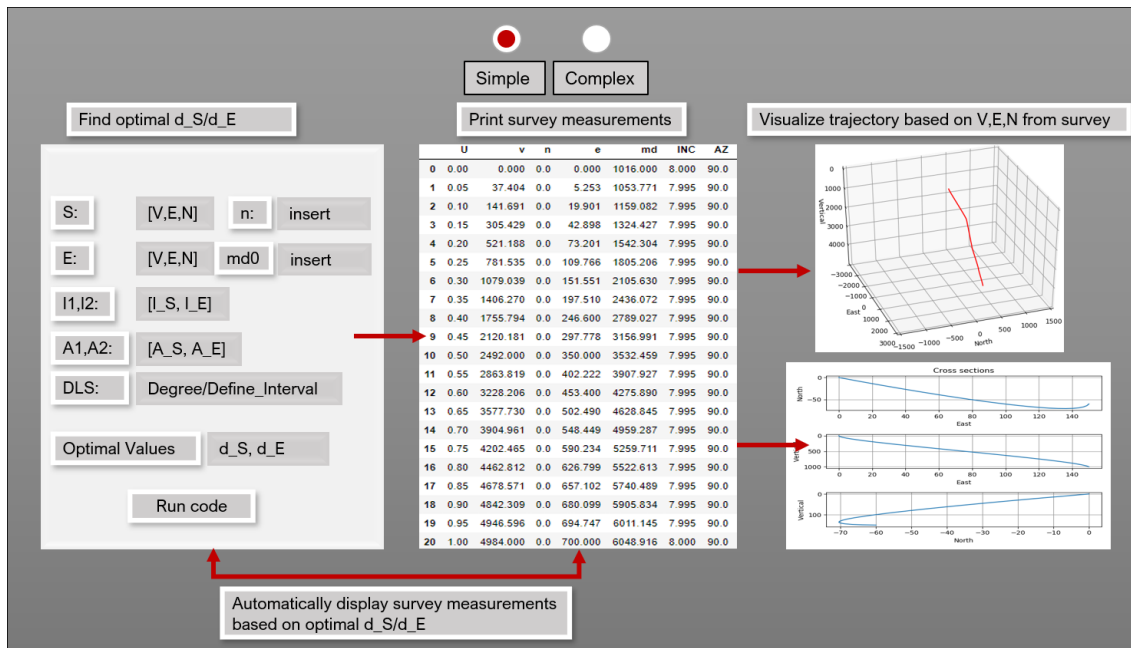


Figure 7.17: How a GUI for trajectory optimization can be presented, this example is for the simple case.

The grey rectangles in figure 7.17 are for data input. The idea is to integrate the code into the existing control system, where all survey measurements and visualisation of the trajectory is accessible to the user. The data from the generated survey measurements can be imported to the main GUI. So that the trajectory that is visualised in real time, with data from the sensor package can be plotted together with the generated trajectory for real time comparison. The GUI that is used today, and designed for the 2019 competition is shown on figure 7.18.

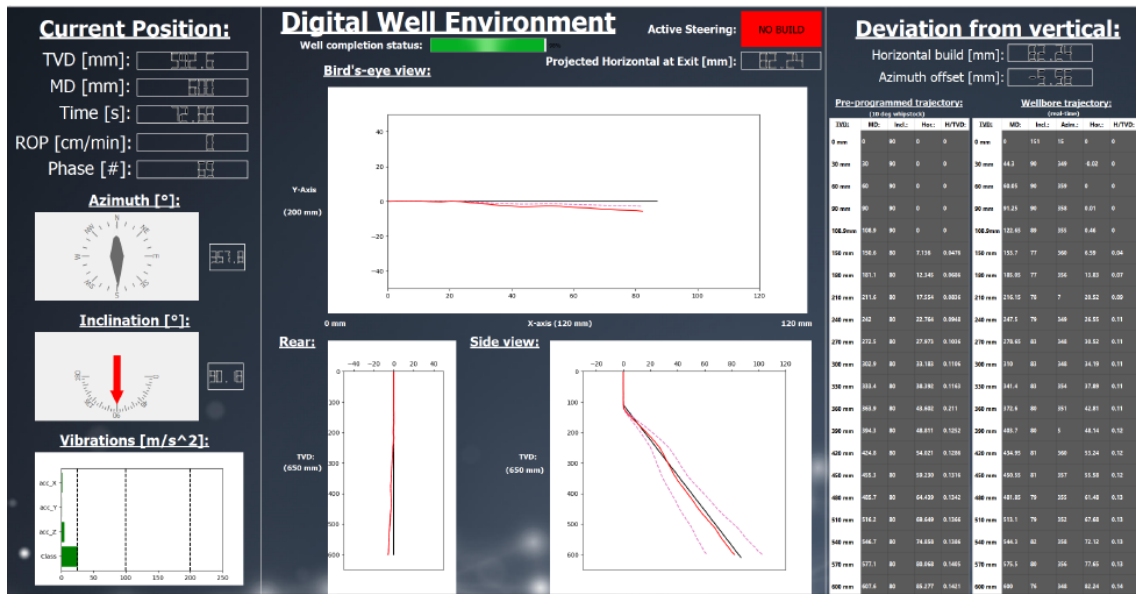


Figure 7.18: The Digital Well Environment-GUI that is currently used on the rig [29].

The "Digital Well Environment" includes visualisation of the well being drilled from different angles, by importing data from generated survey, both can easily be presented to the driller and audience. For the complex trajectory, the GUI can be designed with the same structure as on figure 7.17. A system must be in place to run multiple iterations to find d_S and d_E between all intervals. A flowchart has been designed 7.19 to visualise the logic. The user is asked define all start and end-point conditions. Then n-amount of target points (T-points) should be defined, next step is to assign values to the target points. The system should warn the user, if input will lead to a trajectory with high degree of curvature, as discussed in 7.2.3. If that is not the case, or the user chooses to ignore the warning, proceeding step is calculating optimal d_S and d_E values. Using these, a survey will be printed, where the coordinates will be used to visualise the trajectory. With approval of the user, the data will be exported to main GUI, if not the option will be given to hard code d_S/d_E . If user manually assigns values to d_S/d_E the code will repeat previous step, otherwise a choice will be given to redefine target points or all points, repeating the whole process.

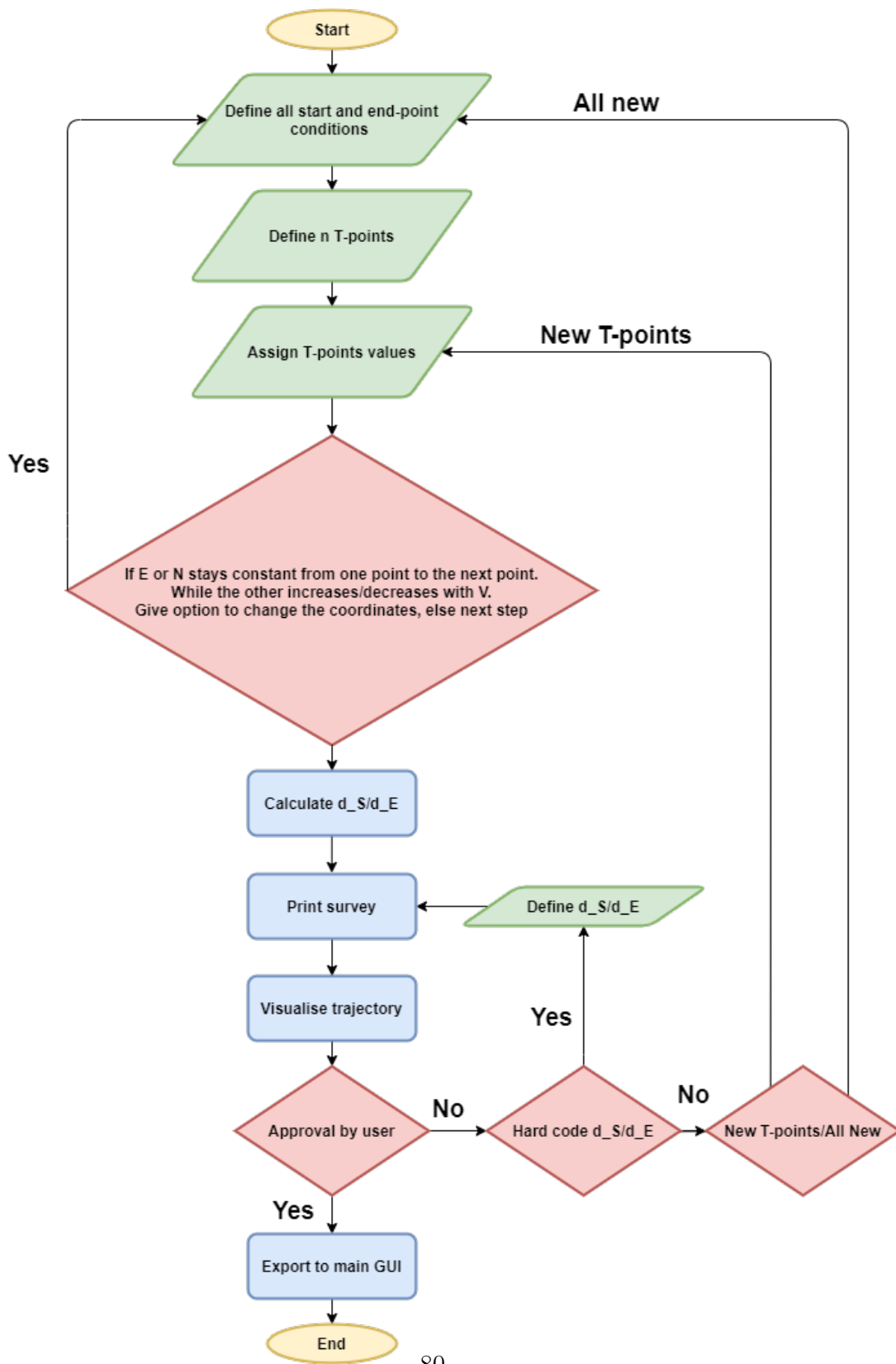


Figure 7.19: Flowchart for trajectory optimisation implementation.

7.4 Position Uncertainty Calculations: Generated Wells

This part is dedicated to the visualisation of ellipses and ellipsoids of uncertainty, presented in chapter 5 on generated wells. As the work on the downhole sensor package has been delayed and can not be used this semester to collect data while drilling. Generated wells within the capabilities of the rig will be used as data. The MATLAB script previously presented will be used, by importing survey points from generated wells. A substantial change to the calculations will be the change in number of weighting functions. As a majority of the weighting functions used on the field data from the Excel spreadsheet [24], are for MWD-tools developed for full scale drilling rigs. And can not be applied for the laboratory drilling rig.

7.5 Considering Weighting Functions

For the rig, the main weighting functions considered are those for the accelerometer and magnetometer. The weighting function $\frac{\delta \mathbf{p}}{\delta \varepsilon_i}$ is given with labels according to the error it represents. For example ABXY is the weighting function $\frac{\delta \mathbf{p}}{\delta \varepsilon_i}$ for accelerometer bias error for x and y-axis. In this case study, only bias and scale-factor errors for the accelerometer and magnetometer will be considered. These have been chosen due to the importance for acquiring data such as azimuth and inclination. In addition to inclination and azimuth data being the largest uncertainties for the laboratory drilling rig.

7.5.1 Accelerometer Bias Error

Uncorrected accelerometer bias will contribute to position error and pitch-or-roll angle error for orientation systems. Where roll is rotation about the x-axis and roll is rotation with respect to the y-axis. The bias error for accelerometers are measured in mg, which is milli-g's [38]. In addition a bias error magnitude must be specified for the MWD error model, to be used in the weighting functions. Where the the gravitational acceleration is often set to $g = 9.81m/s^2$. The accelerometer bias errors and their respective weighting functions has to be defined for all three axes of the sensor. In the data-sheet for our sensor the following values are given for the bias errors [23].

Bias	Description	Magnitude	Unit
Initial Tolerance	Board-level, all axes	± 50	mg

Table 7.5: Accelerometer specifications for the ICM-20948 sensor, with focus on bias error.

The magnitude of the bias error can be set to $\sigma_{bias} = 0.05m/s^2$. Having identified the magnitude of the bias error for the specific sensor used. Following weighting functions for accelerometer bias error for x,y and z are presented [41].

$$ABXY = \frac{1}{G} \begin{bmatrix} 0 \\ -\cos(I) \\ \tan(\theta)\cos(I)\sin(A) \end{bmatrix} \quad (7.1)$$

$$ABZ = \frac{1}{G} \begin{bmatrix} 0 \\ -\sin(I) \\ \tan(\theta)\sin(I)\sin(A) \end{bmatrix} \quad (7.2)$$

7.5.2 Accelerometer Scale-Factor Error

Similar to the bias error, the scale-factor error contributes to errors in position calculations. Where the bias error is a constant offset, the scale-factor error increases as the input accelerations increases. Thus, the total error effect is when the acceleration range is wide [38]. In [23], the data sheet provides us with the scale-factor error values (e.g. sensitivity) for the accelerometer where $\sigma = 0.5\% = 0.005$. Weighting functions for the scale-factor error from the accelerometer are [41]:

$$ASXY = \begin{bmatrix} 0 \\ \frac{\sin(I)\cos(I)}{\sqrt{2}} \\ -\frac{\tan(\theta)\sin(I)\cos(I)\sin(A)}{\sqrt{2}} \end{bmatrix} \quad (7.3)$$

$$ASZ = \begin{bmatrix} 0 \\ -\sin(I)\cos(I) \\ \tan(\theta)\sin(I)\cos(I)\sin(A) \end{bmatrix} \quad (7.4)$$

7.5.3 Magnetometer Bias Error

For the magnetometer the data sheet provides us with a zero-field output of ± 2000 . The sensor is a digital 16-bit device, where the magnetometer has a fixed range of $\pm 4900 \mu T$, giving a full range of $9800\mu T$ [23]. Giving us

$$\frac{9800\mu T}{2^{16}} = 0.14953\mu T/bit \quad (7.5)$$

With a tolerance of $\pm 2000LSB$ in zero-field conditions, LSB can be converted to μT with following equation.

$$0.14953\mu T/bit \cdot 2000LSB = \pm 299\mu T \quad (7.6)$$

Following terms are weighting functions for the magnetometer bias errors. Where $\sigma = 299 \mu T$.

$$MBXY = \begin{bmatrix} 0 \\ 0 \\ \frac{-\cos(I)\sin(a)}{B\cos(\theta)} \end{bmatrix} \quad (7.7)$$

For the z-axis:

$$MBZ = \begin{bmatrix} 0 \\ 0 \\ \frac{-\sin(I)\sin(a)}{B\cos(\theta)} \end{bmatrix} \quad (7.8)$$

7.5.4 Magnetometer Scale-Factor Error

For a 16-bit sensor the general sensitivity will be

$$\frac{1}{2^{16}} = 0.00152 \quad (7.9)$$

The scale-factor error of the magnetometer is $\sigma = 0.00152$, and weighting functions from [41]:

For XY, assuming scale-factor error is systematic

$$MSXY = \begin{bmatrix} 0 \\ 0 \\ \frac{\sin(I)\sin(A)(\tan(\theta)\cos(I)+\sin(I)+\cos(A))}{\sqrt{2}} \end{bmatrix} \quad (7.10)$$

For scale-factor error z-axis

$$MSZ = \begin{bmatrix} 0 \\ 0 \\ -(\sin(I)\cos(A) + \tan(\theta)\cos(I))\sin(I)\sin(A) \end{bmatrix} \quad (7.11)$$

7.5.5 Limitations

The spreadsheet from ISCWSA states "Although an attempt has been made to ensure that the formulation is general, no assumptions should be made as to the validity of the data on any other well paths or with any other error models" [24]. For this study, a majority of the weighting functions will not be included, and the ones mentioned will have their magnitudes changed based on the data-sheet [23]. The magnitude σ of the error is according to ISCWSA determined by finding the standard deviation of an error, based on test data [41]. Due to time constraints, experiments have not been conducted to map the error distribution of the sensors.

7.6 Magnetic Variables

Values for magnetic dip, geomagnetic field strength and declination are all collected from maps developed by National Center For Environmental Information et.al. The data which is used, will be approximately for the Stavanger area, and not precisely the location of the drilling rig at the University of Stavanger. Magnetic dip angle is set to 72° [30], geomagnetic field strength is 51000 nT [32] and declination is set to 2° [31]. Convergence is 0° as reference is true north. The gravitational acceleration is set to $9.81m/s^2$ based on data from [33].

7.7 Trajectory

For calculation of EOU a simple trajectory with a start- S and end-point E will be generated. A constant azimuth of 90° and inclination of 8° is preferred. As previously demonstrated, the dimensions for the competition rock will be scaled up. A simple trajectory is chosen for this part of the study, as the main task is to apply the methodology of uncertainty calculations for a laboratory rig, using generated data. The trajectory will be generated from the KOP. The trajectory has following coordinates, S = [E,N,V] = [0,0,0] and E = [700,1,4984]. Initial and finale inclination at 8° , initial and finale azimuth 90° . Survey points n = 100, and **md0** at S is 0 m. Using calculated optimal $d_S = 11$ and $d_E = 21$.

7.7.1 Data Formatting

The survey measurements from the generated survey is exported from the Python script to Excel for adequate formatting. Based on the azimuth along the trajectory, following terms have to be calculated: AzT, AzG and AzMag. Which are the azimuth values for True North, Grid North and Magnetic North. Found using convergence and declination values. All parameters with degrees as unit, are converted to radians. Data must be formatted in Excel, such that the MATLAB script can easily extract necessary data such as MD, AzT, AzG, AzMag, dip angle, gravitational acceleration, total field strength, σ (magnitude of weighting functions), inclination and TVD.

7.8 Results and Discussion

The trajectory itself is visualised on figure 7.20. The axes are scaled up, and the displacement of 700 m along the East-axis is equal to 70 mm when scaled down to the capabilities of the rig.

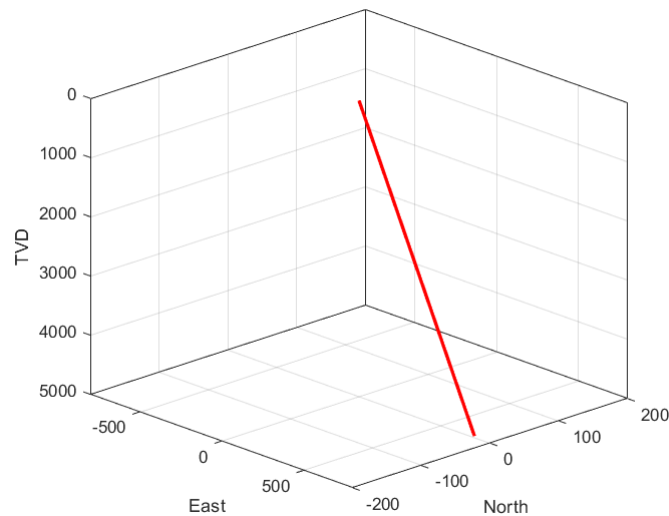


Figure 7.20: Trajectory which position uncertainty calculations will be done for.

Figure 7.21 shows the calculated ellipsoids using the 3x3 covariance matrix with a confidence level of 95%, $k = 2.7954$. The ellipsoids are plotted for every survey point, based on the data generated using the Bezier method.

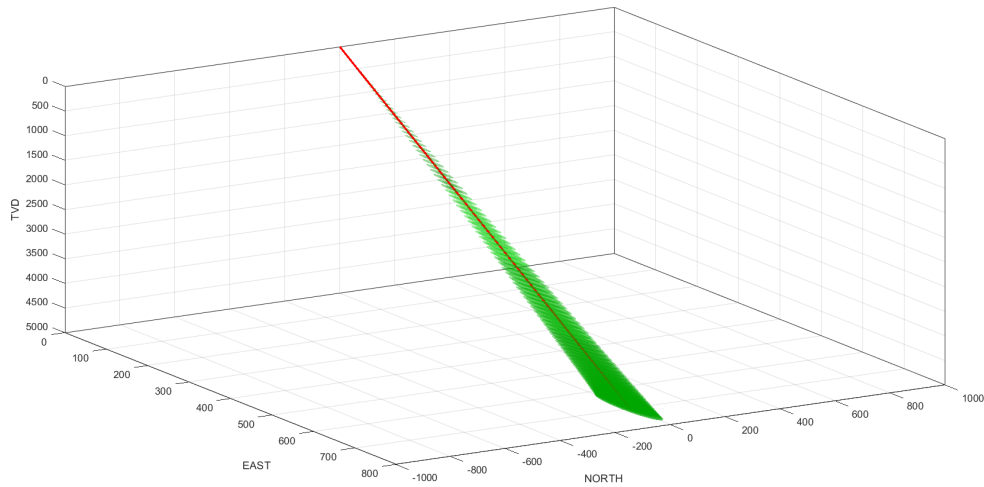


Figure 7.21: Ellipsoids of uncertainty, $k = 2.7954$.

Due to displacement along East-axis and change in weighting functions and their respective magnitudes. The uncertainty is largest along the E-axis. This is clear from observing the covariance matrix sorted in 1x6 array as discussed earlier in the thesis. The second column from left, being the EE or σ_E^2 is the largest element in the 3x3 covariance matrix. The high variance σ_E^2 means that the data in eastern coordinates (NEV-axes) have a higher spread compared to the other variances. Resulting in a higher uncertainty of the true position along the East-axis.

293.8904	642.5318	12.6903	278.7440	-39.2084	-90.2991
296.1218	647.4104	12.7867	280.8604	-39.5061	-90.9848
298.0476	651.6208	12.8698	282.6869	-39.7630	-91.5765
299.6577	655.1411	12.9394	284.2140	-39.9778	-92.0712
300.9422	657.9497	12.9948	285.4324	-40.1492	-92.4659
301.8919	660.0261	13.0359	286.3331	-40.2759	-92.7577
302.4981	661.3517	13.0620	286.9081	-40.3568	-92.9440
302.7523	661.9080	13.0730	287.1493	-40.3907	-93.0222

Figure 7.22: Covariance matrix 3x3 elements, $k = 2.7954$

The change in radii of the ellipsoids is visualised in figure 7.23. Where the red lines are the semi-major axis along the East-axis (80.1 m) and the green lines are the semi-minor axis along the North-axis (33.5 m). The distance between two lines of the same colour is equal to the total length of the ellipse along that axis.

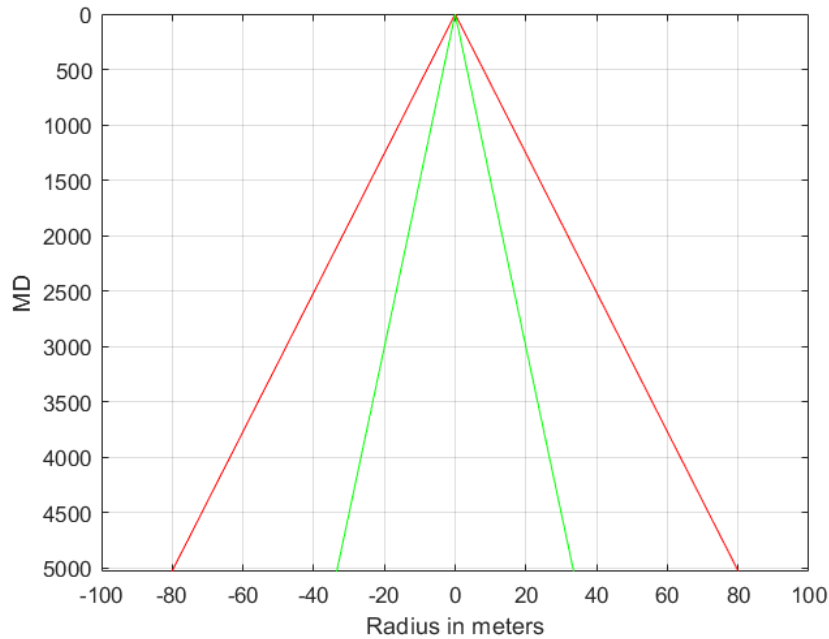


Figure 7.23: Semi-major and semi-minor axes, $k = 2.7954$ from 3×3 covariance matrix (NEV).

Compared to the field case, the uncertainty is considerably smaller. This is mainly due to the elimination of over twenty weighting functions that would lead to a larger accumulation of error with increasing well depth. In addition to the ellipsoids, the ellipses of uncertainty can be plotted based on the 2×2 covariance matrix. Where $k = 2.4477$ for 95% confidence level for two dimensions. The uncertainty ellipses are rotated with respect to their eigenvalues on figure 7.24. The ellipses are plotted for N/E axes for every survey point. This forms a cone shape around the trajectory, which with 95% probability the trajectory will be within. As the trajectory moves along the East-axis, the uncertainty ellipse shifts its orientation making the uncertainty largest along the East-axis. As k is smaller for a 95% confidence level for two dimensions, compared to three dimensions. Logically, the axes of the uncertainty ellipse are smaller for the 2×2 covariance matrix. As the formula for semi-axes is $Semi - axis_i = \sqrt{\lambda_i} * k$.

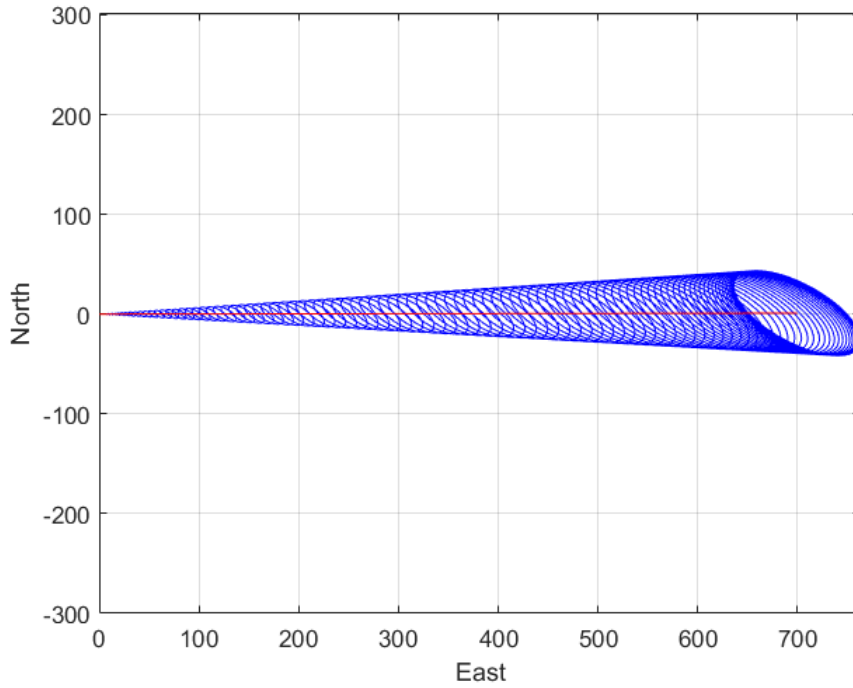


Figure 7.24: Based on 2x2 covariance matrix, $k = 2.4477$.

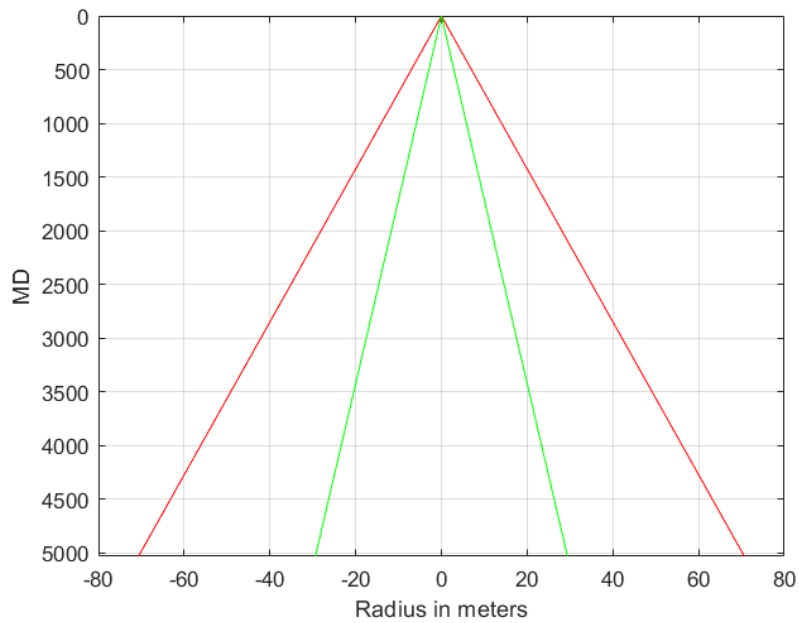


Figure 7.25: Semi-major and semi-minor axes, $k = 2.4477$ from 2x2 covariance matrix.

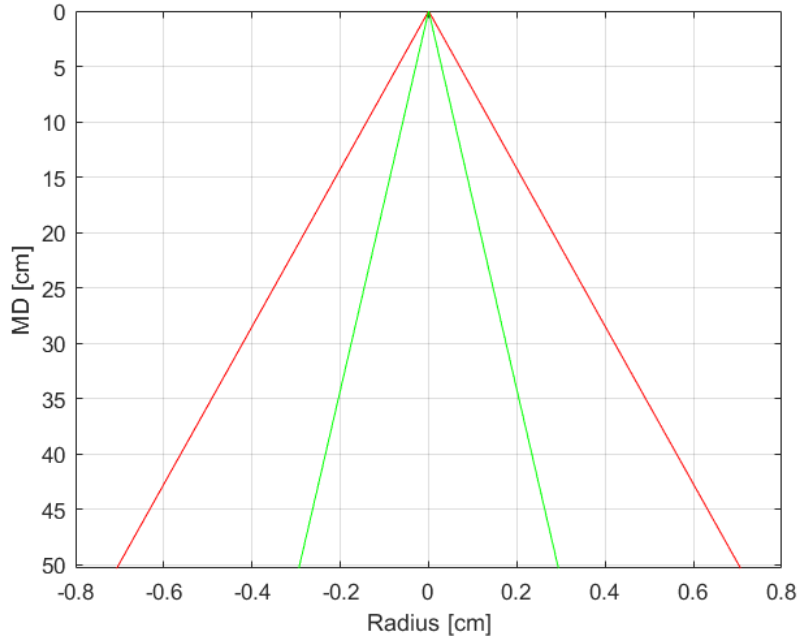


Figure 7.26: Semi-major and semi-minor axes, $k = 2.4477$ from 2×2 covariance matrix in centimeters.

For figure 7.24 the semi-major (E) axis is 70.6 m and semi-minor (N) axis 29.4 m. These are smaller than the 80.1 m and 30.5 m axes calculated from the 3×3 covariance matrix. As the 2×2 covariance matrix do not include the variance for depth measurements, thus the total uncertainty is smaller compared to the 3×3 case. If the trajectory is scaled down, the 700 m displacement along the East-axis corresponds to 70 mm. Assuming the uncertainty stays proportional and can be scaled down using the same factor as for the trajectory coordinates. A semi-major axis of 80.1 m in the MATLAB script would corresponds to a 0.00801 m or 8.01 mm uncertainty on the rig, shown on figure 7.26. Where the data is scaled up, where 0.07 m displacement on the rig, is equal to 700 m displacement in the script. The small values for uncertainty for the rig are expected, as the overall scale is considerably smaller when compared to field data. The precise and small uncertainty values for this case, is due to good survey measurements from the generated data. Technical challenges that must be solved before implementing following system on the laboratory drilling rig. Is the calculation of well coordinates in [E,N,V] format, as discussed in chapter 3. The quality of the uncertainty measurement are directly related to the quality of the survey measurements. If the sensor-package used on the rig, leads to unreliable data for coordinates of the well and inclination and azimuth values. The uncertainty calculations will be unreliable and not representative of the downhole conditions. As the uncertainties calculated are with magnitudes σ

from the data-sheet of the proposed sensor, and not based on the proposed methods from ISCWSA, where the errors of the sensors is mapped through experiments. The calculated ellipsoids and ellipses for the rig case are not necessarily correct.

This case study considered a total of 8 weighting functions with magnitudes from the data-sheet of the sensor. Yet, with a small amount of weighting functions the uncertainty along the E-axis is up 80.1 m. Compared to the industry cases discussed earlier, the uncertainty is disproportionately larger than the field case mentioned from [42]. Preferably the code used to generate the trajectories for test data, is optimised for small scale use. As scaling up and down the data can lead to errors. In addition, it is not verified that the uncertainty calculations scale proportionally. ISCWSA highlights the importance of correct survey interval, as this need to be sufficiently short for accurate calculations. The report recommends using the industry standard of 100 ft or 30 m as a general rule [41]. Whereas the script used to generate survey measurements currently do not offer the options to set survey measurement interval, and is randomly defined based on the trajectory boundaries and selected survey points to print. Currently, the ΔMD is different throughout the $u = [0,1]$ interval. Meaning that the depth between survey point 1 and 2 can be 20 m, then between survey point 2 and 3 the depth can be 45 meters. For these calculations, a constant survey interval should be programmed, as this is the standard the ISCWSA model follows.

As the ISCWSA is mainly developed for industry cases, it must be confirmed that these methods can be applied for smaller case systems. An error in one direction of 8.01 mm when the total MD is over 60 cm is negligible for the laboratory drilling rig. If the downhole sensor output lacks in data quality for azimuth and inclination, the uncertainty will be larger. Moving average filters are used to average out the data output from the sensors to eliminate heavy fluctuations, as downhole vibrations can produce data with noise. Disadvantage of using filters to average data output is the delay of data, depending on the defined window of the filter. Correct and precise inclination and azimuth values are essential to calculate trustworthy values for easting (E) and northing (N) coordinates. With improved sensor data the simple task of visualising the trajectory in 3D would be simple, and helpful for the driller and audience. The trajectories currently drilled with the rig, are simple and are characterised by a vertical section, a KOP and a small displacement along the horizontal plane. As the rig becomes more complicated and capable of more demanding complex trajectories, the uncertainty calculations will become valuable to determine the true position of the bit. As the now cancelled Drillbotics competition 2020 was based on hitting target points decided by the jury. Having the possibility to calculate uncertainty of the bit position would be an advantage to decide if the target is hit with a predefined confidence level.

Or if the driller should consider tripping out a small section of the well, and initiate directional drilling from a new depth, to increase the probability of hitting the target.

The diameter of the well does not stay constant during drilling. Based on previous experiments, hole diameter enlargement is often due to reaming of the well or vibrations. On figure 7.1 the well diameter is larger than the bit diameter in some sections. As directional drilling is initiated, lateral vibrations are introduced. The larger well diameter will be included in the uncertainty calculations. Likewise, the magnitude of which the well diameter has increased will be reflected in the error ellipsoids. As the well diameter increases in size over the bit diameter, the larger the uncertainty calculations will become. As more experimental data is collected, when these models and the sensor are installed on the BHA. A threshold or interval can be defined for what dimensions of uncertainty is expected at different depths with given sensor configuration. If anomalies are detected, where the uncertainty calculations are larger than expected, it can be a sign of well diameter above bit diameter. Figure 7.27 illustrates how both pre-planned and drilled trajectory can be visualised on the same plot, including position uncertainty calculations.

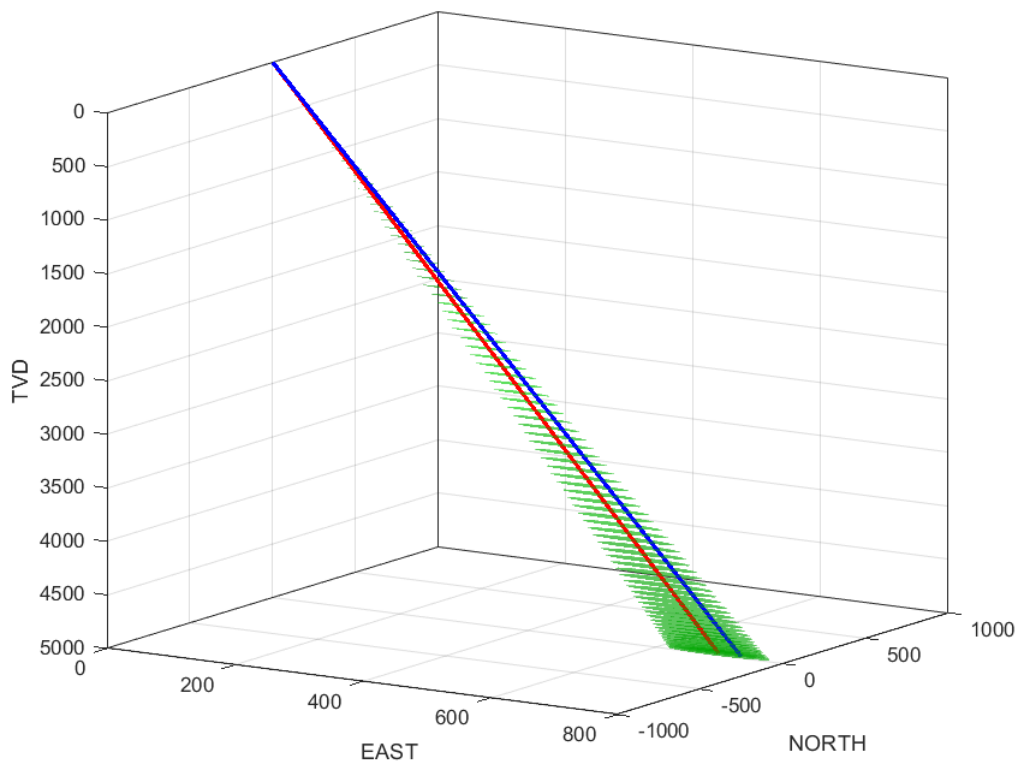


Figure 7.27: Pre-planned trajectory in blue, drilled trajectory and ellipsoids of uncertainty in red.

Chapter 8

Conclusion and Recommendations

8.1 Conclusion

By implementing error models adopted by the industry, uncertainty calculations have been performed for both field data and generated trajectories. For the generated trajectories, the magnitudes and number of weighting functions were chosen with the physical laboratory rig in mind. The purpose of these test were to apply the methods used in by the industry for small scale use. To make the transition easier when the models presented are in the future fully integrated on the rig.

As the industry and Drillbotics competition moves towards a digital future, these methods will be valuable for time efficient and user friendly well planning. The well planning of trajectories is an easy task where only a few parameters are to be defined. The generated trajectory can be plotted with the real trajectory drilled to stay on course. By being able to compare both trajectories in real time, the driller can decide if the operation is going according to the plan. As presented, currently the code for trajectory optimisation is only functional for larger dimensions. The scalability of depth and displacement in the horizontal plane, are believed to scale proportionally. The main concern for having to scale up the system for trajectory optimisation, is how it affects the position uncertainty calculations. Thus, the code should be optimised for floats, so the user can give the depth of the rock as 0.6 m rather than 6 m, 60 m, 600 m etc. Due to extraordinary circumstances, the work related to the new sensor was not finalised. Thus, the proposed methods have not been tested with data from the sensor during drilling.

With methods to calculate position uncertainty calculations on small scale system, the aim is to recreate realistic industry scenarios in a laboratory setup, opening the possibility to drill multiple wells from the same pilot hole. The code for plotting the

uncertainty ellipses/ellipsoids is working, and can be used for real field data given that the necessary data is available. The code calculates the covariance matrix in NEV and HLA axes, the latter calculated using the cosine directional matrix. These can be used to calculate the radii of the ellipse or ellipsoid along the principal axes. With the possibility to visualise the ellipsoids together with the trajectory, the driller gets a clear picture of the well. For future competitions and research projects, this opens the possibility to do research on uncertainty calculations in a safe and controlled laboratory environment. Where different sensors and configurations can be used to drill different well profiles for research, but also to optimise the rig for the Drillbotics competition. With the current rig setup the trajectories are limited to simple wells with straight section. As the deviated drilling techniques and mechanical components improve, the uncertainty generations will play an increasingly important role. In addition the methods presented in this thesis is highly relevant for UiS Drillbotics Team B, as they operate in an exclusively digital environment.

8.2 Recommendations

This section is dedicated to personal recommendations from the writer to future members of the project.

- **Sensors:** Implementation of a sensor-package with high precision and data quality. A thorough method for calibration of sensor should be defined. A proposed and discussed method is to buy a robot arm with integrated mapping system. Mount the proposed sensor to be used downhole on the robot arm and calibrate accordingly. The data from the sensor attached to the arm should be calibrated with respect to the mapping system of the robot. All sensor errors must be defined based on test data. As this data is necessary to define the standard deviation to determine the magnitude σ used for the weighting function for calculations of position uncertainty. Shielding of the sensor components. It has been observed that the electromagnetic interference from the top drive when turned on, affects the data from the sensor. A proper shielding solution of the sensor components and wires, or the top drive itself should be worked out.
- **Trajectory Optimisation:** The current state of the trajectory generation script for well planning and visualisation works well. Yet, the script should be optimised for smaller scale use, as currently the script cannot correctly interpret floats, just integers. Other options that can be added: 1. Define the survey measurement interval, as in take a measurement for every X meters. 2. An integrated solution to add multiple points that the trajectory should hit, before ending at end-point E.

- **Uncertainty Calculations:** Streamline the script and translate it to Python, so that it can be implemented directly into the control system of the laboratory rig. Develop an intuitive GUI where user can simply choose which weighting functions should be considered and define their respective magnitudes. For ellipsoid visualisation, this should be done with respect to their orientation and eigenvalues. This was only implemented for the 2D-cases this year. More testing is necessary, collaboration with a company that is willing to give access to survey measurements and their calculated ellipsoids for comparison and fine tuning. Relies on quality data from the sensor, data quality from the sensor should be prioritised for reliable uncertainty calculations.
- **Mechanical:** The main mechanical part to reconsider is the BHA. A short BHA is preferred as it allows the system to be more flexible and mobile considering directional drilling. Steel is great for durability and for strength, but challenging for directional drilling at this scale. A more unorthodox approach, experimenting with different materials can be valuable. Where ductility and flexibility is prioritised, at the same time the material should have durability to handle vibrations, erosion and heat. By keeping the same design philosophy used today, where the BHA consists of multiple parts screwed together. A new design could consist of stabilizers made of steel, and the rest of the BHA made using softer and more flexible material with smaller diameter. To avoid friction and heating problems from having contact with the rock. The distance between the sensor package and the bit should be minimized. By placing the sensors close to the bit, the data $[E,N,V]$ would be a more precise representation of the bits location. A recommendation was made last year by E. Løken and J.Løkkevik, 2019 [29] to place a sensor immediately behind the bit with internal memory that could be extracted after a drilling session was completed. This was to avoid problems with wiring over rotating parts. This would be an interesting approach to fine tune the uncertainty calculations model with reliable data.

After the implementation of the pneumatic motor (downhole motor) for the 2019 competition, vibrations are less of an issue. As some vibrations are still present during drilling, this leads to data with noise, or rapid fluctuations in data. This problem is especially prevalent for the WOB measurements, as axial vibrations are the main issue. These external forces may damage equipment such as the sensors which can result in data of low quality, or loss of downhole data. A different approach to drilling should be considered and experimented with. Where the priority is higher RPM and lower WOB. A spring element in the BHA can be a potential solution for dampening the axial vibrations.

References

- [1] U.S Energy Information Administration. *Average Depth of Crude Oil and Natural Gas Wells*. URL: https://www.eia.gov/dnav/pet/pet_crd_welldep_s1_a.htm.
- [2] D. Denney et al. "Adopting Multilateral Technology". In: *Journal of Petroleum Technology* 50.07 (1998), pp. 50–51.
- [3] V. Flores et al. "Relief Well Planning". In: *IADC/SPE Drilling Conference and Exhibition*. Society of Petroleum Engineers. 2014.
- [4] W. S. Hugh et al. "Accuracy prediction for directional measurement while drilling". In: *SPE Drilling & Completion* 15.04 (2000), pp. 221–233.
- [5] M. Alsaleh. *Normal Distribution Curve*. URL: <https://mohsinalsaleh.com/normal-distribution-curve/>.
- [6] R. Amorin. "Application of Minimum Curvature method to Well-path Calculations". PhD thesis. 2009.
- [7] J. Bryla. *Directional Drilling and Deviation Control*. PowerPoint Presentation. 2019.
- [8] J. Byman. "Design, manufacturing and optimization of the drill string assembly for a miniature drilling rig". University of Stavanger, 2020.
- [9] ISCWSA Committee. *Collision Avoidance Calculations - Current Common Practice, 2013*. URL: <https://www.iscwsa.net/media/files/files/b6fb074d/current-common-practice-in-collision-avoidance-calculations-oct-2017.pdf>.
- [10] Deprag. *Air Motors - Customized drive solutions*. URL: https://www.deprag.com/fileadmin/bilder_content/emedi/broschueren_pics/emedi_druckluftmotoren/D6000/D6000en.pdf.
- [11] Drillbotics.com. *About Drillbotics*. URL: <https://drillbotics.com/about-drillbotics/>.
- [12] Drillbotics.com. *Guidelines 2018-2019*. URL: https://drillbotics.com/download/guidelines/guidelines_archive/Drillbotics-Guidelines-2019_4.pdf.
- [13] Drillbotics.com. *Guidelines 2019-2020*. URL: <https://drillbotics.com/download/guidelines/2020-Drillbotics-Guidelines.pdf>.

- [14] A. Buchanan et.al. “Geomagnetic referencing—the real-time compass for directional drillers”. In: *Oilfield Review* 25.3 (2013), pp. 32–47.
- [15] PPT Exploration and Production. *Basic Drilling for WE ADP 2014*. PowerPoint Presentation. 2014. URL: <https://www.slideshare.net/pakmek/basic-drilling-for-we-adp-2014-k-edit3>.
- [16] F. O. Farah. “Directional well design, Trajectory and survey calculations, with a case study in Fiale, Asal rift, Djibouti”. In: *Ministry of Energy and Natural Resources* 27 (2013).
- [17] C. P. Gooneratne and L. Bodong et.al. *Instruments, Measurement Principles and Communication Technologies for Downhole Drilling Environments*. Springer, 2019.
- [18] A. Jakobsen H. Hagen and M. Khadisov. “Laboratory drilling rig construction, testing and modeling for optimization and problem management.” University of Stavanger, 2018.
- [19] HBM. *U9C Miniature Load Cell: for Measurement of Tensile and Compressive Force*. URL: <https://www.hbm.com/en/3926/u9c-miniature-force-sensor-for-tensile-and-compressive-forces/>.
- [20] O. A. Hjelm and S. J. Nilsen. “Further Development and Testing of an Autonomous Drilling Rig and Control Algorithms for Improved Drilling Performance”. University of Stavanger, 2018.
- [21] Baker Hughes. *PDC Micro-Bit for the 2020 SPE Drillbotics Competition*. URL: <https://drillbotics.com/download/guidelines/2020-Drillbotics-Microbits-Baker-Hughes.pdf>.
- [22] T. Inglis. *Directional drilling*. Vol. 2. Springer Science & Business Media, 2013.
- [23] TDK Invensense. *World’s Lowest Power 9-Axis MEMS MotionTracking™ Device*. URL: https://invensense.tdk.com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1.3.pdf?ref_disty=digikey.
- [24] A. McGregor ISCWSA. *Error Model Example File 1 - MWDREV4 ISCWSA*. URL: <https://www.iscwsa.net/articles/error-model-example-file-1-mwdrev4-iscwsa/>.
- [25] A. Jakobsen and H. Hagen. “Design and optimization of an autonomous laboratory-scale drilling rig and bit design for directional drilling”. MA thesis. University of Stavanger, 2020.
- [26] A. Jamieson. “Introduction to Wellbore Positioning (e-book)”. In: *Research Office in Inverness, Scotland: University of the Highlands and Islands* (2012).
- [27] M. A. Jassim. “The Ellipse of Position Error”. In: *ZANCO Journal of Pure and Applied Sciences* 31.s3 (2019), pp. 324–328.
- [28] R. G. R. Kuhnast. “Comparison of Separation Factor Estimations based on Scenario Analysis”. Montan Universitat, 2015.

- [29] E. A. Løken and J. Løkkevik. “Optimization of an Intelligent Autonomous Drilling Rig: Testing and Implementation of Machine Learning and Control Algorithms for Formations Classification, Downhole Vibrations Management and Directional Drilling”. University of Stavanger, 2019.
- [30] NOAA/NCEI and CIRES. “US/UK World Magnetic Model - Epoch 2015.0 Main Field Inclination (I)”. URL: https://upload.wikimedia.org/wikipedia/commons/d/de/World_Magnetic_Inclination_2015.pdf.
- [31] NOAA/NCEI and CIRES. “US/UK World Magnetic Model - Epoch 2020.0 Main Field Declination (D)”. URL: https://www.ngdc.noaa.gov/geomag/WMM/data/WMM2020/WMM2020_D_BoZ_MILL.pdf.
- [32] NOAA/NCEI and CIRES. “US/UK World Magnetic Model - Epoch 2020.0 Main Field Total Intensity (F)”. URL: https://www.ngdc.noaa.gov/geomag/WMM/data/WMM2020/WMM2020_F_BoZ_MILL.pdf.
- [33] Geological Survey of Norway - NGU. “Gravimetry”. URL: <https://www.ngu.no/en/topic/gravimetry>.
- [34] J. HB. Sampaio. “Designing three-dimensional directional well trajectories using Bézier curves”. In: *Journal of Energy Resources Technology* 139.3 (2017).
- [35] J. HB. Sampaio. “Planning 3D well trajectories using cubic functions”. In: (2006).
- [36] M. Shaine. “Error and Ellipses of Uncertainty Analysis in Far North”. MA thesis. NTNU, 2017.
- [37] S. Skjørestad. “Directional drilling capabilities of a rebuilt and optimized autonomous laboratory scale drilling rig.” University of Stavanger, 2019.
- [38] V. Tigadi. *Evaluating inertial measurement units*. URL: <https://www.edn.com/evaluating-inertial-measurement-units/>.
- [39] M. W. Toews. *The semi-major (a) and semi-minor axis (b) of an ellipse*. URL: https://en.wikipedia.org/wiki/Semi-major_and_semi-minor_axes#/media/File:Ellipse_semi-major_and_minor_axes.svg.
- [40] H. Van. “Uncertainty analysis in determining the position of a drill bit”. In: (2018).
- [41] Industry Steering Committee for Wellbore Survey Accuracy. *Definition of the ISCWSA Error Model*. URL: <https://www.iscwsa.net/media/files/files/454df66c/definition-of-iscwsa-error-model-rev4.pdf>.
- [42] L. Xiushan. “Borehole trajectory uncertainty and its characterization”. In: *Petroleum exploration and development* 46.2 (2019), pp. 407–412.

Appendix A

List of Error Sources and Weighting Functions

Following weighting functions are from [41].

11 List of Error Sources and Weighting Functions

Refer to section 4.2 for a details of the notation used below.

11.1 Error Sources Common to Both Gyro and MWD Models

Terms common to both gyro and MWD models:

	Error Code	Description	Propagation Mode	Weighting Function		
				MD	Inc	Azimuth
1	XYM1	xy misalignment 1	S/R	0	w_{12}	0
2	XYM2	xy misalignment 2	S/R	0	0	$-w_{12}/\sin I$
3	XYM3	xy misalignment 3 <i>(singular when vertical)</i>	S/R	0	$w_{34} \cos A$	$-w_{34} \sin A / \sin I$
4	XYM4	xy misalignment 4 <i>(singular when vertical)</i>	S/R	0	$w_{34} \sin A$	$w_{34} \cos A / \sin I$
5	SAG / VSAG	Vertical sag (SAG in MWD model)	S	0	$\sin I$	0
6	DRF-R	Depth random error	R	1	0	0
7	DRF-S	Depth systematic reference	S	-	0	0
8	DSF-W	Depth scale	S/W	ΔD	0	0
9	DST-G	Depth stretch type	G	$(D+D\cos I)$	ΔD	0

11.2 MWD Error Sources

	Error Code	Description	Propagation Mode	Weighting Function		
				MD	Inc	Azimuth
10	ABXY-TI1	Accelerometer bias – term1	S/R	0	$-\frac{\cos I}{G}$	$\frac{\tan \theta \cos I \sin A_m}{G}$
11	ABXY-TI2	Accelerometer bias – term2 <i>(singular when vertical)</i>	S/R	0	0	$\frac{\cot I - \tan \theta \cos A_m}{G}$
12	ABZ	Accelerometer bias z-axis	S	0	$-\frac{\sin I}{G}$	$\frac{\tan \theta \sin I \sin A_m}{G}$

13	ASXY-TI1	Accelerometer scale factor – term1	S	0	$\frac{\sin I \cos I}{\sqrt{2}}$	$-\frac{\tan \theta \sin I \cos I \sin A_m}{\sqrt{2}}$
14	ASXY-TI2	Accelerometer scale factor – term2	S/R	0	$\frac{\sin I \cos I}{\sqrt{2}}$	$-\frac{\tan \theta \sin I \cos I \sin A_m}{2}$
15	ASXY-TI3	Accelerometer scale factor – term3	S/R	0	0	$\frac{\tan \theta \sin I \cos A_m - \cos I}{2}$
16	ASZ	Accelerometer scalefactor z-axis	S	0	$-\sin I \cos I$	$\frac{\tan \theta \sin I \cos I \sin A_m}{-\cos I \sin A_m}$
17	MBXY-TI1	Magnetometer bias – term1	S/R	0	0	$\frac{B \cos \theta}{\cos A_m}$
18	MBXY-TI2	Magnetometer bias – term2	S/R	0	0	$\frac{B \cos \theta}{\sin A_m}$
19	MBZ	Magnetometer bias z-axis	S	0	0	$\frac{B \cos \theta}{\sin I \sin A_m (\tan \theta \cos I + \sin I \cos A_m)}$
20	MSXY-TI1	Magnetometer scale factor – term1	S	0	0	$\frac{\sqrt{2}}{\sin A_m (\tan \theta \sin I \cos I - \cos^2 I \cos A_m - \cos A_m)}$
21	MSXY-TI2	Magnetometer scale factor – term2	S/R	0	0	$\frac{2}{(\cos I \cos^2 A_m - \cos I \sin^2 A_m - \tan \theta \sin I \cos A_m)}$
22	MSXY-TI3	Magnetometer scale factor – term3	S/R	0	0	$\frac{2}{-(\sin I \cos A_m + \tan \theta \cos I) \sin I \sin A_m}$
23	MSZ	Magnetometer scalefactor z-axis	S	0	0	$\frac{\sin I \sin A_m}{B \cos \theta}$
24	AMIL	Axial magnetic interference	S	0	0	$\frac{\cos^2 I \sin A_m (\tan \theta \cos I + \sin I \cos A_m)}{G(1 - \sin^2 I \sin^2 A_m)}$
25	ABIXY-TI1	Accelerometer bias – axial interference correction – term1	S/R	0	$-\frac{\cos I}{G}$	$\frac{-(\tan \theta \cos A_m - \cot I)}{G(1 - \sin^2 I \sin^2 A_m)}$
26	ABIXY-TI2	Accelerometer bias – axial interference correction – term2 (singular when vertical)	S/R	0	0	
27	ABIZ	Accelerometer bias z-axis when axial interference correction applied	S	0	$-\frac{\sin I}{G}$	$\frac{\sin I \cos I \sin A_m (\tan \theta \cos I + \sin I \cos A_m)}{G(1 - \sin^2 I \sin^2 A_m)}$
28	ASIXY-TI1	Accelerometer scale factor – axial interference correction – term1	S	0	$\frac{\sin I \cos I}{\sqrt{2}}$	$-\frac{\sin I \cos^2 I \sin A_m (\tan \theta \cos I + \sin I \cos A_m)}{\sqrt{2}(1 - \sin^2 I \sin^2 A_m)}$
29	ASIXY-TI2	Accelerometer scale factor – axial interference correction – term2	S/R	0	$\frac{\sin I \cos I}{2}$	$-\frac{\sin I \cos^2 I \sin A_m (\tan \theta \cos I + \sin I \cos A_m)}{2(1 - \sin^2 I \sin^2 A_m)}$

30	ASIXY-TI3	Accelerometer scale factor – axial interference correction – term3	S/R	0	0	$\frac{(\tan \theta \sin I \cos A_m - \cos I)}{2(1 - \sin^2 I \sin^2 A_m)}$
31	ASIZ	Accelerometer scalefactor z-axis when axial interference correction applied	S	0	$-\sin I \cos I$	$\frac{\sin I \cos^2 I \sin A_m (\tan \theta \cos I + \sin I \cos A_m)}{(1 - \sin^2 I \sin^2 A_m)}$
32	MBIXY-TI1	Magnetometer bias – axial interference correction – term1	S/R	0	0	$\frac{\cos I \sin A_m}{B \cos \theta (1 - \sin^2 I \sin^2 A_m)}$
33	MBIXY-TI2	Magnetometer bias – axial interference correction – term2	S/R	0	0	$\frac{\cos A_m}{B \cos \theta (1 - \sin^2 I \sin^2 A_m)}$
34	MSIXY-TI1	Magnetometer scale factor – axial interference correction – term1	S	0	0	$\frac{\sin I \sin A_m (\tan \theta \cos I + \sin I \cos A_m)}{\sqrt{2}(1 - \sin^2 I \sin^2 A_m)}$
35	MSIXY-TI2	Magnetometer scale factor – axial interference correction – term2	S/R	0	0	$\frac{\sin A_m (\tan \theta \sin I \cos I - \cos^2 I \cos A_m - \cos A_m)}{2(1 - \sin^2 I \sin^2 A_m)}$
36	MSIXY-TI3	Magnetometer scale factor – axial interference correction – term3	S/R	0	0	$\frac{(\cos I \cos^2 A_m - \cos I \sin^2 A_m - \tan \theta \sin I \cos A_m)}{2(1 - \sin^2 I \sin^2 A_m)}$
37	DEC	Constant declination error	G/R	0	0	1
38	DBH	Declination error dependant on the horizontal component of Earth's field	G/R	0	0	$\frac{1}{B \cos \theta}$
39	MFI	Earth's total magnetic field when axial interference correction applied	G/R	0	0	$\frac{\sin I \sin A_m (\tan \theta \cos I + \sin I \cos A_m)}{B(1 - \sin^2 I \sin^2 A_m)}$
40	MDI	Dip angle when axial interference correction applied	G/R	0	0	$\frac{\sin I \sin A_m (\cos I - \tan \theta \sin I \cos A_m)}{(1 - \sin^2 I \sin^2 A_m)}$

Appendix B

Code: Trajectory Optimization

```
1 # To add a new cell, type '# %%'  
2 # To add a new markdown cell, type '# %% [markdown]'  
3  
4 # %%  
5 import numpy as np  
6 import matplotlib.pyplot as plt  
7 import seaborn as sns  
8 from matplotlib.colors import LogNorm  
9 import math as m  
10 import random  
11 import pandas as pd  
12  
13  
14 # %%  
15 #given data  
16 v0,v1=0,1200      # vertical depth of point 0,1  
17 e0,e1=0,200      # east coordinates of point 0,1  
18 n0,n1=0,510      # north coordinates of point 0,1  
19 inc0,inc1=np.radians(0.0001),np.radians(90)      # inclination angles  
      of point 0,1 in deg (transfer to radian)  
20 az0,az1=np.radians(0.0001),np.radians(0.0001)      # azimuth angles  
      of point 0,1in deg (transfer to radian)  
21  
22 ### we will get the optimum values of k0 and k1 from the coding of  
      trajectory optimization  
23 md0= 101.6      # measure depth of point 0  
24 celln=50      #number of survy points  
25  
26  
27 # %%  
28 k0=range(100,5000,100)      # free parameter at point 0  
29 k1=range(100,5000,100)      # free parameter at point 1
```

```

30 csv,csn,cse,cev,cen,cee=[],[],[],[],[],[]
31 optiValues = []
32 iter = 0
33 for i in k0:
34     for j in k1:
35         w0v=m.cos(inc0)
36         w0n=m.sin(inc0)*np.cos(az0)
37         w0e=m.sin(inc0)*np.sin(az0)
38         k0w0v=i*w0v
39         k0w0n=i*w0n
40         k0w0e=i*w0e
41         w1v=m.cos(inc1)
42         w1n=m.sin(inc1)*np.cos(az1)
43         w1e=m.sin(inc1)*np.sin(az1)
44         k1w1v=j*w1v
45         k1w1n=j*w1n
46         k1w1e=j*w1e
47         csv.append(v0+k0w0v)
48         csn.append(n0+k0w0n)
49         cse.append(e0+k0w0e)
50         cev.append(v1-k1w1v)
51         cen.append(n1-k1w1n)
52         cee.append(e1-k1w1e)
53         breaker = True
54         bv,bn,be,Bv,Bn,Be,B,Bm,tv,tn,te,bbv,bbn,bbe,BB,kv,kn,ke,km
= [], [], [], [], [], [], [], [], [], [], [], [], [], [], [], [], []
55         U,v,n,e,md,dls,Dls,MD=[],[],[],[],[],[],[],[]
56         for num in range(0,celln+1):
57             u=float(num/celln)
58             U.append(u)
59             v.append(v0*(1-u)**3+3*u*csv[iter]*(1-u)**2+3*cev[iter
]* (1-u)*u**2+u**3*v1)
60             n.append(n0*(1-u)**3+3*u*csn[iter]*(1-u)**2+3*cen[iter
]* (1-u)*u**2+u**3*n1)
61             e.append(e0*(1-u)**3+3*u*cse[iter]*(1-u)**2+3*cee[iter
]* (1-u)*u**2+u**3*e1)
62             if (num > 0):
63                 md.append(md[num-1]+m.sqrt((v[num]-v[num-1])**2+(n[
num]-n[num-1])**2+(e[num]-e[num-1])**2))
64             else:
65                 md.append(md0)
66                 bv.append(-3*v0*(1-u)**2+3*(1-u)*(1-3*u)*csv[iter
]+3*(2-3*u)*u*cev[iter]+3*v1*u**2)
67                 bn.append(-3*n0*(1-u)**2+3*(1-u)*(1-3*u)*csn[iter
]+3*(2-3*u)*u*cen[iter]+3*n1*u**2)
68                 be.append(-3*e0*(1-u)**2+3*(1-u)*(1-3*u)*cse[iter
]+3*(2-3*u)*u*cee[iter]+3*e1*u**2)
69                 Bv.append(bv[num]*bv[num])
70                 Bn.append(bn[num]*bn[num])

```



```

71     Be.append(be[num]*be[num])
72     B.append(Bv[num]+Bn[num]+Be[num])
73     Bm.append(np.sqrt(B[num]))
74     tv.append(bv[num]/Bm[num])
75     tn.append(bn[num]/Bm[num])
76     te.append(be[num]/Bm[num])
77     bbv.append(6*v0*(1-u)-6*csv[iter]*(2-3*u)+6*cev[iter
] *(1-3*u)+6*u*v1)
78     bbn.append(6*n0*(1-u)-6*csn[iter]*(2-3*u)+6*cen[iter
] *(1-3*u)+6*u*n1)
79     bbe.append(6*e0*(1-u)-6*cse[iter]*(2-3*u)+6*cee[iter
] *(1-3*u)+6*u*e1)
80     BB.append(bv[num]*bbv[num]+bn[num]*bbn[num]+be[num]*bbe
[num])
81     kv.append(((1/B[num])*bbv[num])-((BB[num]/(B[num])**2)*
bv[num]))
82     kn.append(((1/B[num])*bbn[num])-((BB[num]/(B[num])**2)*
bn[num]))
83     ke.append(((1/B[num])*bbe[num])-((BB[num]/(B[num])**2)*
be[num]))
84     km.append(np.sqrt((kv[num]*kv[num])+(kn[num]*kn[num])+(
ke[num]*ke[num])))
85     dls.append(km[num]*(180*30/np.pi))
86     if (dls[num] >= 5):
87         breaker = False
88     if (breaker == True):
89         optiValues.append([i,j,md[num]]) # the optimum values
of k0,k1 and its correspond md
90         iter += 1
91 min = optiValues[0][2]
92 opti = []
93 for l in range(len(optiValues)):
94     if optiValues[l][2] <= min:
95         opti.append(optiValues[l])
96         min = optiValues[l][2] # calling for k0,k1 whose
the min md among the optimum values
97 print(opti[-1])
98
99
100 # %%
101 k0=opti[-1][0]
102 print(k0)
103
104
105 # %%
106 k1=opti[-1][1]
107 print(k1)

```

Appendix C

Code: Well Trajectory By Bzier

```
1 # To add a new cell, type '# %%'  
2 # To add a new markdown cell, type '# %% [markdown]'  
3 # %%  
4 from IPython import get_ipython  
5  
6  
7 # %%  
8 import numpy as np  
9 import matplotlib.pyplot as plt  
10 import seaborn as sns  
11 from matplotlib.colors import LogNorm  
12 import math as m  
13 import random  
14 import pandas as pd  
15  
16  
17 # %%  
18 #given data  
19 v0,v1=0,1200      # vertical depth of point 0,1  
20 e0,e1=0,200      # east coordinates of point 0,1  
21 n0,n1=0,510      # north coordinates of point 0,1  
22 inc0,inc1=np.radians(0.0001),np.radians(90)      # inclination angles  
                of point 0,1 in deg (transfer to radian)  
23 az0,az1=np.radians(0.0001),np.radians(0.0001)      # azimuth angles  
                of point 0,1in deg (transfer to radian)  
24 k0,k1=400,500    # free parameters of point 0, 1 (control dls and  
                md of the curvature)  
25 ### we will get the optimum values of k0 and k1 from the coding of  
                trajectory optimization  
26 md0=101.6        # measure depth of point 0  
27 celln=50         #number of survy points  
28
```

```

29
30 # %%
31 # calculations
32 w0v=m.cos(inc0)
33 w0n=m.sin(inc0)*np.cos(az0)
34 w0e=m.sin(inc0)*np.sin(az0)
35 k0w0v=k0*w0v
36 k0w0n=k0*w0n
37 k0w0e=k0*w0e
38 w1v=m.cos(inc1)
39 w1n=m.sin(inc1)*np.cos(az1)
40 w1e=m.sin(inc1)*np.sin(az1)
41 k1w1v=k1*w1v
42 k1w1n=k1*w1n
43 k1w1e=k1*w1e
44 csv=v0+k0w0v
45 csn=n0+k0w0n
46 cse=e0+k0w0e
47 cev=v1-k1w1v
48 cen=n1-k1w1n
49 cee=e1-k1w1e
50
51
52
53 # %%
54
55 U,bv,bn,be,Bv,Bn,Be,B,Bm,tv,tn,te
   =[],[],[],[],[],[],[],[],[],[],[],[]
56 bbv,bbn,bbe,BB,Bm,kv,kn,ke,km,inc,az,dls,TF
   =[],[],[],[],[],[],[],[],[],[],[],[],[]
57 R,cv,cn,ce,hv,hn,he,PHI,hcv,hcn,hce,hct
   =[],[],[],[],[],[],[],[],[],[],[],[]
58 v,n,e,md,INC,AZ,TolFace=[],[],[],[],[],[],[]
59 def trajectory(*args):
60     for num in range(0,celln+1):
61         u=float(num/celln)
62         U.append(u)
63         v.append(v0*(1-u)**3+3*u*csv*(1-u)**2+3*cev*(1-u)*u**2+u
   **3*v1)
64         n.append(n0*(1-u)**3+3*u*csn*(1-u)**2+3*cen*(1-u)*u**2+u
   **3*n1)
65         e.append(e0*(1-u)**3+3*u*cse*(1-u)**2+3*cee*(1-u)*u**2+u
   **3*e1)
66         if (num > 0):
67             md.append(md[num-1]+m.sqrt((v[num]-v[num-1])**2+(n[num]
   ]-n[num-1])**2+(e[num]-e[num-1])**2))
68         else:
69             md.append(md0)
70         bv.append(-3*v0*(1-u)**2+3*(1-u)*(1-3*u)*csv+3*(2-3*u)*u*

```

```

cev+3*v1*u**2)
71     bn.append(-3*n0*(1-u)**2+3*(1-u)*(1-3*u)*csn+3*(2-3*u)*u*
cen+3*n1*u**2)
72     be.append(-3*e0*(1-u)**2+3*(1-u)*(1-3*u)*cse+3*(2-3*u)*u*
cee+3*e1*u**2)
73     Bv.append(bv[num]*bv[num])
74     Bn.append(bn[num]*bn[num])
75     Be.append(be[num]*be[num])
76     B.append(Bv[num]+Bn[num]+Be[num])
77     Bm.append(np.sqrt(B[num]))
78     tv.append(bv[num]/Bm[num])
79     tn.append(bn[num]/Bm[num])
80     te.append(be[num]/Bm[num])
81     bbv.append(6*v0*(1-u)-6*csv*(2-3*u)+6*cev*(1-3*u)+6*u*v1)
82     bbn.append(6*n0*(1-u)-6*csn*(2-3*u)+6*cen*(1-3*u)+6*u*n1)
83     bbe.append(6*e0*(1-u)-6*cse*(2-3*u)+6*cee*(1-3*u)+6*u*e1)
84     BB.append(bv[num]*bbv[num]+bn[num]*bbn[num]+be[num]*bbe[num]
])
85     kv.append(((1/B[num])*bbv[num])-((BB[num]/(B[num])**2)*bv[
num]))
86     kn.append(((1/B[num])*bbn[num])-((BB[num]/(B[num])**2)*bn[
num]))
87     ke.append(((1/B[num])*bbe[num])-((BB[num]/(B[num])**2)*be[
num]))
88     km.append(np.sqrt((kv[num]*kv[num])+(kn[num]*kn[num])+(ke[
num]*ke[num])))
89     inc.append(m.acos(tv[num]))
90     az.append(m.atan(te[num]/tn[num]))
91     dls.append(km[num]*(180*30/np.pi))
92     R.append(1/km[num])
93     cv.append(kv[num]/km[num])
94     cn.append(kn[num]/km[num])
95     ce.append(ke[num]/km[num])
96     hv.append((tv[num]/np.tan(inc[num]))-(1/np.sin(inc[num])))
97     hn.append(tn[num]/np.tan(inc[num]))
98     he.append(te[num]/np.tan(inc[num]))
99     PHI.append(m.acos(hv[num]*cv[num]+hn[num]*cn[num]+he[num]*
ce[num]))
100     hcv.append((hn[num]*ce[num])-(he[num]*cn[num]))
101     hcn.append((he[num]*cv[num])-(hv[num]*ce[num]))
102     hce.append((hv[num]*cn[num])-(hn[num]*cv[num]))
103     hct.append(hcv[num]*tv[num]+hcn[num]*tn[num]+hce[num]*te[
num])
104     if inc[num]>0:
105         if hcv[num]>=0:
106             TF.append(PHI[num])
107         else:
108             TF.append(360-PHI[num])
109     INC.append(inc[num]*180/np.pi)

```

```

110     AZ.append(az[num]*180/np.pi)
111     TolFace.append(TF[num]*180/m.pi)
112
113
114     v[num]=round(v[num] ,3)           # vertical depth
115     n[num]=round(n[num] ,3)           # north coordinate
116     e[num]=round(e[num] ,3)           # east coordinate
117     md[num]=round(md[num] ,3)         # measure depth
118     inc[num]=round(inc[num] ,3)        # inclination
119     az[num]=round(az[num] ,3)         # azimuth
120     dls[num]=round(dls[num] ,3)       # dog leg severity
121     PHI[num]=round(PHI[num] ,3)
122     TF[num]=round(TF[num] ,3)         # tool face angle
123
124     return list(zip(U,v,n,e,md,inc,INC,az,AZ,dls,PHI,TF,TolFace))
125
126
127
128 # %%
129 writer = pd.ExcelWriter('OutputFromWell.xlsx')
130
131 #configuration = [v0,v1,n0,n1,e0,e1,inc0,inc1,az0,az1,md0,k0,k1]
132 #then your function will be
133 #def calculate_stuff(u, *configuration):
134 #    #[v0,v1,n0,n1,e0,e1,inc0,inc1,az0,az1,md0,k0,k1]= configuration
135 #    #unpackng all the variables
136 #    #return v, n, e, md, INC, AZ
137 result= trajectory(U,v,n,e,md,inc,INC,az,AZ,dls,PHI,TF,TolFace)
138 df=pd.DataFrame(list(zip(U,v,n,e,md,inc,INC,az,AZ,dls,PHI,TF,
139 TolFace)),columns=['U','v','n','e','md','inc','INC','az','AZ','
140 dls','PHI','TF','TolFace'])
141 print(df)
142
143 df.to_excel(writer, sheet_name='Data', index=False)
144 writer.save()
145 writer.close()
146
147 plt.plot(e,n)
148 plt.plot(e,n,v)
149 plt.xlabel('East')
150 plt.ylabel('North')
151
152 # %%
153 from mpl_toolkits import mplot3d
154 get_ipython().run_line_magic('matplotlib', 'notebook')

```

```

155 plt.figure(1)
156 ax = plt.axes(projection='3d')
157
158 ax.plot3D(e, n, v, c='r')
159
160 ax.set_xlabel('East')
161 ax.set_ylabel('North')
162 ax.set_zlabel('Vertical')
163 ax.invert_zaxis()
164 ax.zaxis._axinfo['juggled'] = (1,2,0)
165
166 plt.tight_layout()
167
168
169 # %%
170 from matplotlib.ticker import (MultipleLocator, FormatStrFormatter,
171                               AutoMinorLocator)
172
173 fig = plt.figure(2)
174
175
176 plt.subplot(3, 1, 1)
177 plt.title('Cross sections')
178 plt.plot(e, n, '-')
179 plt.xlabel('East')
180 plt.ylabel('North')
181 plt.grid()
182
183 plt.subplot(3, 1, 2)
184 plt.plot(e, v, '-')
185 plt.xlabel('East')
186 plt.ylabel('Vertical')
187 plt.gca().invert_yaxis()
188 plt.grid()
189
190 plt.subplot(3, 1, 3)
191 plt.plot(n, e, '-')
192 plt.xlabel('North')
193 plt.ylabel('Vertical')
194 plt.gca().invert_yaxis()
195 plt.grid()
196 plt.tight_layout()
197
198
199
200 # %%

```


Appendix E

Code: Position Uncertainty Calculations

Section of the code, including 8 weighting functions for laboratory drilling rig.

```
1 clear
2 clc
3 close all
4
5 rho=xlsread ('error-model-example-mwdrev4-iscwsa-1CASE','Model','W4
   :W30');
6 MD=xlsread ('error-model-example-mwdrev4-iscwsa-1CASE','Wellpath','
   K3:K103');
7 Inc=xlsread ('error-model-example-mwdrev4-iscwsa-1CASE','Wellpath','
   L3:L103');
8 AzT=xlsread ('error-model-example-mwdrev4-iscwsa-1CASE','Wellpath','
   M3:M103');
9 AzG=xlsread ('error-model-example-mwdrev4-iscwsa-1CASE','Wellpath','
   N3:N103');
10 AzMag=xlsread ('error-model-example-mwdrev4-iscwsa-1CASE','Wellpath
   ','O3:O103');
11 TVD=xlsread ('error-model-example-mwdrev4-iscwsa-1CASE','Wellpath','
   P3:P103');
12
13 %% other parameters
14
15 G=9.81;
16 B=51000; %magnetic field strength
17 Dip=1.257; %dip angle
18 VInc=0.0001;%inclination limit
19 Sing=10;%singular matrix condition
20 SF=2.7954; % k-constant for confidence level
21
```



```

22 %% caclulate drdp_k from eq (10)
23
24 drdp_k=drdpk(Inc, AzT, MD);
25
26 %% caclulate drdp_k+1 from eq (13)
27
28 drdp_k1=drdpk1(Inc, AzT, MD, drdp_k);
29
30 %% Determine weighting matrix dpde %case:DRFR
    -----
31
32 dpde_DRFR=[];
33 for i=1:length(Inc)
34     dpde_DRFR=[dpde_DRFR;1 0 0];
35 end
36
37 %% Determine weighting matrix dpde -systematic errors %case:ABXY1
38
39 dpde_ABXY1=[0, -cos(Inc(1))/G, tan(Dip)*cos(Inc(1))*sin(AzMag(1))/G];
40 for i=2:length(Inc)
41     dpde_ABXY1=[dpde_ABXY1;0, -cos(Inc(i))/G, tan(Dip)*cos(Inc(i))*
        sin(AzMag(i))/G];
42 end
43
44 %% Determine covariance matrix %case:ABXY1
45 index=4;
46 type=2;%systematic error
47
48 COV_ABXY1=conmatrix(dpde_ABXY1, index, type, drdp_k, drdp_k1, rho, MD,
    Inc, AzT, AzG, AzMag, TVD, Sing, G, B);
49
50 %% Determine weighting matrix dpde -systematic errors %case:ABZ
51
52 dpde_ABZ=[];
53 for i=1:length(Inc)
54     dpde_ABZ=[dpde_ABZ;0, -sin(Inc(i))/G, tan(Dip)*sin(Inc(i))*sin(
        AzMag(i))/G];
55 end
56
57 %% Determine covariance matrix %case:ABZ
58 index=6;
59 type=2;%systematic error
60
61 COV_ABZ=conmatrix(dpde_ABZ, index, type, drdp_k, drdp_k1, rho, MD, Inc,
    AzT, AzG, AzMag, TVD, Sing, G, B);
62
63 %% Determine weighting matrix dpde -systematic errors %case:ASXY1
64
65 dpde_ASXY1=[];

```

```

66 for i=1:length(Inc)
67     dpde_ASXY1=[dpde_ASXY1;0, sin(Inc(i))*cos(Inc(i))/sqrt(2), -tan(
        Dip)*sin(Inc(i))*cos(Inc(i))*sin(AzMag(i))/sqrt(2)];
68 end
69
70 %% Determine covariance matrix %case:ASXY1
71 index=7;
72 type=2;%systematic error
73
74 COV_ASXY1=conmatrix(dpde_ASXY1,index,type,drdp_k,drdp_k1,rho, MD,
        Inc, AzT, AzG, AzMag, TVD,Sing,G,B);
75
76 %% Determine weighting matrix dpde -systematic errors %case:ASZ
77
78 dpde_ASZ=[];
79 for i=1:length(Inc)
80     dpde_ASZ=[dpde_ASZ;0, -sin(Inc(i))*cos(Inc(i)), tan(Dip)*sin(Inc(
        i))*cos(Inc(i))*sin(AzMag(i))];
81 end
82
83 %% Determine covariance matrix %case:ASZ
84 index=10;
85 type=2;%systematic error
86
87 COV_ASZ=conmatrix(dpde_ASZ,index,type,drdp_k,drdp_k1,rho, MD, Inc,
        AzT, AzG, AzMag, TVD,Sing,G,B);
88
89 %% Determine weighting matrix dpde -systematic errors %case:MBXY1
90
91 dpde_MBXY1=[];
92 for i=1:length(Inc)
93     dpde_MBXY1=[dpde_MBXY1;0,0, -cos(Inc(i))*sin(AzMag(i))/B/cos(Dip
        )];
94 end
95
96 %% Determine covariance matrix %case:MBXY1
97 index=11;
98 type=2;%systematic error
99
100 COV_MBXY1=conmatrix(dpde_MBXY1,index,type,drdp_k,drdp_k1,rho, MD,
        Inc, AzT, AzG, AzMag, TVD,Sing,G,B);
101
102 %% Determine weighting matrix dpde -systematic errors %case:MBZ
103
104 dpde_MBZ=[];
105 for i=1:length(Inc)
106     dpde_MBZ=[dpde_MBZ;0,0, -sin(Inc(i))*sin(AzMag(i))/B/cos(Dip)];
107 end
108

```

```

109 %% Determine covariance matrix %case:MBZ
110 index=13;
111 type=2;%systematic error
112
113 COV_MBZ=conmatrix(dpde_MBZ,index,type,drdp_k,drdp_k1,rho, MD, Inc,
    AzT, AzG, AzMag, TVD,Sing,G,B);
114
115 %% %% Determine weighting matrix dpde -systematic errors %case:
    MSXY1
116
117 dpde_MSXY1=[];
118 for i=1:length(Inc)
119     dpde_MSXY1=[dpde_MSXY1;0,0,sin(Inc(i))*sin(AzMag(i))*(tan(Dip)*
        cos(Inc(i))+sin(Inc(i))*cos(AzMag(i)))/sqrt(2)];
120 end
121
122 %% %% Determine covariance matrix %case:MSXY1
123 index=14;
124 type=2;%systematic error
125
126 COV_MSXY1=conmatrix(dpde_MSXY1,index,type,drdp_k,drdp_k1,rho, MD,
    Inc, AzT, AzG, AzMag, TVD,Sing,G,B);
127
128
129 %% Determine weighting matrix dpde -systematic errors %case:MSZ
130
131 dpde_MSZ=[];
132 for i=1:length(Inc)
133     dpde_MSZ=[dpde_MSZ;0,0,-(sin(Inc(i))*cos(AzMag(i))+tan(Dip)*cos(
        Inc(i)))*sin(Inc(i))*sin(AzMag(i))];
134 end
135
136 %% Determine covariance matrix %case:MSZ
137 index=17;
138 type=2;%systematic error
139
140 COV_MSZ=conmatrix(dpde_MSZ,index,type,drdp_k,drdp_k1,rho, MD, Inc,
    AzT, AzG, AzMag, TVD,Sing,G,B);
141
142 %% sum up all convariant matrix-global/local coordinate
143 rn=length(Inc);
144 COV_total=[COV_ABXY1;COV_ABZ;COV_ASXY1;COV_ASZ;COV_MBXY1;COV_MBZ;
    COV_MSXY1;COV_MSZ];
145 [COV,COV_L]=covsum(COV_total,rn,MD, Inc, AzT, AzG, AzMag, TVD);
146
147 %% reformulate covariance matrix
148 [COV_3D,COV_2D,COV_L3D, eig_v]=covmatrix(COV,COV_L,rn,SF);
149
150 %% calculate surevey point([E,N,V])

```

```

151 [E,N,V]=mcm(MD, Inc, AzT,rn);
152
153 %% 3D plot
154
155 for i=1:1:rn-1
156
157     plot3(E(i),N(i),V(i),'r'); hold on;
158     [x, y, z] = ellipsoid(E(i),N(i),V(i),eig_v(i,1),eig_v(i,2),
159     eig_v(i,3));
160     surf(x, y, z)
161     mymap = [ 0 0.6 0
162             0 0.6 0
163             0 0.8 0
164             0 0.7 0
165             0 0.65 0];
166     colormap(mymap)
167     view(90,0)
168     shading(gca,'interp') %interp colour // faceted for border
169     alpha 0.4
170     set(gca, 'ZDir', 'reverse')
171     xlabel('EAST')
172     ylabel('NORTH')
173     zlabel('TVD')
174     grid on
175     box on
176     hold on
177     plot3(E,N,V,'Linewidth',2,'color','red')
178 end
179 %% Plot major/minor axes vs tvd this is for 3d
180 figure(3)
181 e1 = eig_v(:,1)/100;
182 e2 = eig_v(:,2)/100;
183 e3 = eig_v(:,3)/100;
184 e11 = -1*e1;
185 e22 = -1*e2;
186 e33 = -1*e3;
187 plot(e1,MD/100,'color','red')
188 hold on
189 % plot(e2,MD,'color','blue')
190 plot(e2,MD/100,'color','green')
191 plot(e11,MD/100,'color','red')
192 % plot(e22,MD,'color','blue')
193 plot(e22,MD/100,'color','green')
194 set(gca, 'YDir', 'reverse')
195 xlabel('Radius [cm]')
196 ylabel('MD [cm]')
197 ylim([0 50.33])
198 grid on

```

```

199 hold off
200
201
202 %% Plot 2D rotated ellipse
203 figure(7)
204 data=[];
205 special_e=[];
206     for i=1:1:rn
207         data(i, :, :) = [ COV_L(i,1), COV_L(i,4);
208                         COV_L(i,4), COV_L(i,2)];
209         covariance = [ COV_L(i,1), COV_L(i,4);
210                      COV_L(i,4), COV_L(i,2)];
211
212
213 special_e_temp = eig(covariance);
214 special_e = [special_e; [sqrt(special_e_temp(2))*2.4477, sqrt(
215     special_e_temp(1))*2.4477]];
216 [eigenvec, eigenval] = eig(covariance);
217 [largest_eigenvec_ind_c, r] = find(eigenval == max(max(eigenval)));
218 largest_eigenvec = eigenvec(:, largest_eigenvec_ind_c);
219
220
221 largest_eigenval = max(max(eigenval));
222
223 if(largest_eigenvec_ind_c == 1)
224     smallest_eigenval = max(eigenval(:,2))
225     smallest_eigenvec = eigenvec(:,2);
226 else
227     smallest_eigenval = max(eigenval(:,1))
228     smallest_eigenvec = eigenvec(1,:);
229 end
230
231 angle = atan2(largest_eigenvec(2), largest_eigenvec(1));
232
233 if(angle < 0)
234     angle = angle + 2*pi;
235 end
236
237 %% Get the 95% confidence interval error ellipse
238 chisquare_val = 2.4477;
239 theta_grid = linspace(0,2*pi);
240 phi = angle;
241 X0=E(i);
242 Y0=N(i);
243 a=chisquare_val*sqrt(largest_eigenval);
244 b=chisquare_val*sqrt(smallest_eigenval);
245
246 ellipse_x_r = b*cos(theta_grid);

```

```

247 ellipse_y_r = a*sin( theta_grid );
248
249 R = [ cos(phi) sin(phi); -sin(phi) cos(phi) ];
250
251
252 r_ellipse = [ellipse_y_r;ellipse_x_r]' * R;
253
254
255 plot(r_ellipse(:,1) + X0,r_ellipse(:,2) + Y0,'-','color','blue')
256 hold on;
257
258
259 grid on
260
261 hXLabel = xlabel('East');
262 hYLabel = ylabel('North');
263
264 plot(E,N,'color','red')
265 axis equal
266
267     end
268
269 %%
270 making=[];
271 sorting=[];
272 for i=1:10:rn-1
273 figure(9)
274 making(i,:,:) = [ COV(i,1),COV(i,4);
275                 COV(i,4),COV(i,2)];
276                 sorting=[ COV(i,1),COV(i,4);
277                          COV(i,4),COV(i,2)];
278 f_eig = eig(sorting);
279 rad1 = sqrt(f_eig(1,:))*SF;
280 rad2 = sqrt(f_eig(2,:))*SF;
281 t_angle = 0:((2*pi)/(267)):2*pi;
282 centerx = E(i);
283 centery = N(i);
284 el_x = centerx + rad1*cos(t_angle);
285 el_y = centery + rad2*sin(t_angle);
286 plot(el_x,el_y,'color','blue')
287 ylim([0 1800])
288 xlim([0 6000])
289 axis equal
290 grid on
291 xlabel('East')
292 ylabel('North')
293 hold on
294 plot(E,N,'color','red')
295 plot(r_ellipse(:,1) + X0,r_ellipse(:,2) + Y0,'-','color','blue')

```

```
296 end
297
298 %% just traj
299
300 figure(10)
301 plot3(E,N,V, 'color', 'red', 'Linewidth', 2)
302 grid on
303 box on
304 ylabel('North')
305 xlabel('East')
306 zlabel('TVD')
307 set(gca, 'ZDir', 'reverse')
```