



Maanalaisten poralaitteiden navigointimenetelmät ja niiden visualisointi

Ville Heikkinen

OPINNÄYTETYÖ
Huhtikuu 2022

Sähkö- ja automaatiotekniikan tutkinto-ohjelma
Automaatiotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkö- ja automaatiotekniikan tutkinto-ohjelma
Automaatiotekniikka

HEIKKINEN, VILLE:

Maanalaisen poralaitteiden navigointimenetelmät ja niiden visualisointi

Opinnäytetyö 57 sivua, joista liitteitä 0 sivua
Huhtikuu 2022

Maanalaisessa kaivos- ja tunnelinrakennustoiminnassa käytetään louhintamenetelmänä poraus- ja räjäytysmenetelmää. Menetelmässä porataan reiät, joihin räjäytyspanokset sijoitetaan, ja räjäytys itsessään tekee louhinnan raskaan työosuuden. Reiät porataan maanalaisilla poralaitteilla. Ehtona ekonomiselle louhinnalle on porattavien reikien tarkkuus sijainnin, syvyyden ja kulman suhteen. Suurin tarkkuus saavutetaan tietokoneavusteisella poraamisella, jossa hyödynnetään maanalaista navigointia reikien poraamiseen. Yksi poralaitteita valmistava yritys on Sandvik Mining and Construction Oy.

Opinnäytetyössä kehitettiin visualisointityökalu neljälle Sandvik Mining and Construction Oy:n käyttämälle maanalaiselle navigointimenetelmälle. Yrityksellä oli olemassa Matlab-koodit käsiteltävistä neljästä navigointimenetelmästä, ja työkalulla on kyettävä visualisoimaan kyseisten koodien tuottamia navigointituloksia. Työkalua on tarkoitus käyttää koodien jatkokehitykseen ja laaduntarkastukseen aikaisessa kehitysvaiheessa ennen navigointikoodien pääsyä tuotantoon.

Opinnäytetyön tuloksena luotiin jokaisella navigointimenetelmälle oma Matlab liivescript -tiedosto. Tiedosto sisältää koodin, joka toteuttaa navigointimenetelmän visualisoinnin. Tiedostossa on myös kuvalliset ohjeet työkalun käyttöön. Visualisoinnin tulos näkyy Matlabin erillisessä kuvaajaikkunassa, jonka työkalu avaa. Käyttäjä kykenee muuttamaan kuvaajanäkymää zoomaamalla, pyörittämällä tai liikuttamalla sitä.

Työkalussa on käyttäjälle syötekentät, joiden avulla muutetaan navigointimenetelmän käyttämiä parametreja. Työkalu reagoi parametrien muutoksiin reaaliajassa muuttaen kuvaajaikkunan näkymää.

Visualisointityökalu täyttää sille asetetut tavoitteet, mutta kehityskohteita työkalussa silti on. Yhtenä tärkeimpänä lisättävänä ominaisuutena on käyttäjän syötteen tarkistus. Tällä hetkellä työkalu vaatii käyttäjältä hyvää navigointimenetelmien ja niihin liittyvien parametrien tuntemusta. Syötteen tarkistuksella voidaan helpottaa työkalun käyttöä ja laajentaa työkalun käyttäjäryhmää.

Asiasanat: maanalainen navigointi, visualisointi, Matlab, ohjelmointi

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Electrical and Automation Engineering
Automation Engineering

HEIKKINEN, VILLE:
Navigation Methods of Underground Drill Rigs and Their Visualization

Bachelor's thesis 57 pages
April 2022

Drill and blast excavation is used when developing underground mine or tunnel. This method includes drilling holes where explosive charges are then placed. To achieve economical excavation, it is important to control location, depth and angle of the drilled holes. Most accurate results are achieved when using computer aided drilling and underground navigation. Sandvik Mining and Construction Oy is one of the few companies who manufacture these underground drill rigs.

The purpose of this study was to develop a visualization tool for four underground navigation methods used by Sandvik Mining and Construction Oy. The company has Matlab codes for the four methods used in this study. The visualization tool must be able to visualize results of those codes. The visualization tool is intended to be used for further development and quality control of underground navigation methods.

As a result of the study, a dedicated Matlab livescript file was created for each navigation method. The file contains a code that implements the visualization and input fields for the user. The result of the visualization is displayed in a separate Matlab figure window which opens automatically. The user can manipulate the figure view by zooming, rotating or moving it.

The visualization tool meets the requirements set for it but it can still be improved. In the future the most important improvement would be user input validation. Currently the tool requires the user to be familiar with the navigation methods and related parameters. Adding an input validation would expand the user group of the tool and make its use easier.

Key words: underground navigation, visualization, Matlab, programming

SISÄLLYS

1	JOHDANTO	6
2	SANDVIK MINING AND CONSTRUCTION OY	8
3	NAVIGOINTI, KOORDINAATISTOT JA VISUALISOINTI	10
	3.1 Maanalainen paikannus ja navigointi	10
	3.2 Koordinaatistot ja koordinaatistomuunnokset.....	12
	3.3 Visualisointi	16
4	KÄYTETYT NAVIGOINTIMENETELMÄT	18
	4.1 Käytettävät koordinaatistot	18
	4.2 Tunnelilinja	19
	4.3 Porakruununavigointi	20
	4.4 Lasernavigointi ilman kaarretaulukkoa	22
	4.5 Lasernavigointi kaarretaulukon kanssa	23
	4.6 Takymetrinavigointi	24
5	NAVIGOINTITAPAHTUMIEN VISUALISOINTI	26
	5.1 Matlab	26
	5.2 Yhteiset funktiot.....	27
	5.3 XML-tiedoston parsiminen	32
	5.4 Porakruununavigoinnin visualisointi	37
	5.5 Lasernavigoinnin visualisointi ilman kaarretaulukkoa.....	40
	5.6 Lasernavigoinnin visualisointi kaarretaulukon kanssa.....	43
	5.7 Takymetrinavigoinnin visualisointi	48
6	POHDINTA	53
	LÄHTEET	56

LYHENTEET JA TERMIT

Allokointi	Ohjelmoinnissa muistin ennakkovaraaminen
Argumentti	Ohjelmoinnissa funktiolle annettava arvo/muuttuja, jota kyseinen funktio käyttää
CAD	Tietokoneavusteinen suunnittelu (Computer Aided Design)
iSURE	Tietokoneohjelma tunnelirakennuksen suunnitteluun (Intelligent Sandvik Underground Rock Excavation software)
IREDES	Kaivostoiminnassa käytettävä tietostandardi (International Rock Excavation Data Exchange Standard)
Katko	Tunnelin etenemä yhden räjäytyksen aikana
Matlab	Numeerisen laskennan ohjelma (Matrix Laboratory)
Modulaarinen	Ohjelman jakaminen pienempiin osiin, joita voidaan hyödyntää useamman kerran
Paalunumero	Sijainti tunnelissa
Paluuarvo	Ohjelmoinnissa funktion tuottama arvo, joka palautetaan funktiota kutsuneelle taholle
Parsiminen	Ohjelmoinnissa tietorakenteen jäsentely uuteen muotoon
Porauskaavio	Määrää porattavien reikien paikat ja sijoitetaan porattavaan tunnelin seinämään
Tunnelilinja	Joukko pisteitä 3D-koordinaatistossa, joiden kautta tunneli rakennetaan
XML	Merkintäkielien standardi (Extensible Markup Language)

1 JOHDANTO

Opinnäytetyössä kuvataan Sandvikin käyttämiä maanalaisia navigointimenetelmiä. Sandvik valmistaa maanalaisia poralaitteita ja näissä käytetään tietokonetta apuna poraamiseen. Yksi tärkeä osa tietokoneavusteisessa poraamisessa on paikannus ja navigointi. Sandvikilla onkin useita eri menetelmiä, joilla saavutetaan tarvittava paikannus. Työn tarkempuna tavoitteena on esittää näiden navigointimenetelmien tulos uudessa esitysmuodossa. Aikaisemmin melkein kaikki tulokset on esitetty pelkästään numeerisesti, mikä on ihmiselle hankalasti tulkittavaa. Työn tavoitteena on kehittää navigointimenetelmien visualisointityökalu. Visualisointityökalu toteutetaan käyttämällä numeerisen laskennan Matlab-ohjelmistoa. Työkalun käyttöön tarvitsee Matlab-ympäristön.

Sandvik Mining and Construction Oy:llä on olemassa Matlab-koodit navigointialgoritmeista. Koodit ovat yksittäisiä funktioita, jotka yhdessä muodostavat useita eri kokonaisuuksia. Osana työtä on myös selvittää, miten nämä koodit toimivat ja mitä ne tuottavat tuloksena. Tämän jälkeen voidaan käyttää, muokata ja yhdistää koodeja opinnäytetyön tarkoituksen mukaiseksi.

Työn tavoitteisiin ei kuitenkaan sisälly olemassa olevien koodien tarkastaminen tai korjaaminen. Koodeja muutetaan vain yhteensopivuuden varmistamiseksi. Opinnäytetyön tuloksia voidaan käyttää tulevaisuudessa koodien tarkastamiseen ja korjaamiseen.

Valmista työkalua käytettäisiin navigointimenetelmien jatkokehitykseen. Työkalulla pystyttäisiin näkemään muutokset eri versioiden välillä ja voidaan verrata mahdollisia hyötyjä tai haittoja, joita kehityksen mukana ilmenee. Työkalulla voidaan varmistaa, että navigointimenetelmät tuottavat oletetun kaltaisen lopputuloksen. Työkalu olisi myös yksi lisäväline mahdollisten ongelmien ja ohjelmointivirheiden huomaamiseen. Työkalua käytettäisiin varhaisessa kehitysvaiheessa, jolloin mahdolliset virheet on helpompi korjata. Toisin sanoen visualisointia voitaisiin käyttää yhtenä laaduntarkistusmenetelmänä.

Visualisointityökalun toinen käyttötarkoitus voisi suuntautua aiheiden opetukseen. Maanalaisten poralaitteiden navigointimenetelmät on hankala aihe ja käsite

ymmärtää nopeasti ja selkeästi. Työkalua voitaisiinkin käyttää asiaa selittäessä ja opettaessa. Työkalua voidaan myös käyttää havainnollisten kuvien tuottamiseen erilaisia käyttötarkoituksia varten.

2 SANDVIK MINING AND CONSTRUCTION OY

Sandvik AB on vuonna 1862 perustettu ruotsalainen teollisuuskonserni. Sandvik AB:lla on laaja kirjo osaamista, johon sisältyy mm. työkalujen valmistus, kaivostoiminnan tuotteiden valmistus, kivien ja mineraalien käsittelyyn tarkoitettujen laitteiden valmistus ja erilaisten materiaalien valmistus. (Sandvik AB ei pvm.)

Vuonna 2021 Sandvik AB:n liikevaihto oli 99,105 miljardia ruotsin kruunua. Bruttotuloksen osuus oli 40,5 % eli 40,173 miljardia kruunua. Liikevoittoa oli 18,654 miljardia kruunua. Sandvik investoi tuotekehitykseen n. 3,7 miljardia kruunua. Vuonna 2021 Sandvik AB työllisti 44 133 työntekijää. (Sandvik AB ei pvm.)

Opinnäytetyö tehdään yritykselle Sandvik Mining and Construction Oy. Sandvik Mining and Construction Oy kuuluu teollisuuskonserni Sandvik AB:hen. Vuonna 1997 Tampereella sijaisi Tampella Tamrock Oy, joka oli talousvaikeuksiin ajautuneen Oy Tampella AB:n tytäryhtiö. Sandvik AB osti Tampella Tamrock Oy:n ja nimesi yhtiön ensin Tamrock Oy:ksi ja tämän jälkeen Sandvik Tamrock Oy:ksi. Yhtiö sai nimen Sandvik Mining and Construction Oy vuonna 2006. (Tamrock Oy (Oy Tampella Ab) — Osakehistoria ei pvm.)

Sandvik Mining and Construction Oy valmistaa maanalaisia ja maanpäällisiä poralaitteita ja muuta tarvittavaa laitteistoa mm. kaivostoimintaan, louhintaan ja rakentamiseen. (Sandvik Mining and Rock Technology ei pvm.)

Sandvik Mining and Construction Oy on yksi muutamasta yrityksestä maailmalla, joka kykenee valmistamaan maailman terävintä kärkeä olevia maanalaisia poralaitteita. Tampereen tehtaalla sijaitsee myös Sandvikin testikaivos, joka on uniikki maailmassa. Testikaivoksen olosuhteet ovat samat kuin oikeassa kaivoksessa ja kaikki tehtaalta lähtevät laitteet käyvät läpi testauksen kaivoksessa. (Sandvik Mining and Rock Technology ei pvm., Sandvik Mining and Construction Oy 2015, 4)

Tarkempana opinnäytetyön paikkana on Sandvikin Tampereen tehdas. Tampereen tehtaalla valmistetaan pääasiassa maanpäällisiä ja maanalaisia poralaitteita kaivos, työmaa ja rakennus tarkoituksiin. Tähän opinnäytetyöhön liittyen ollaan

kiinnostuneita maanalaisista poralaitteista. Tekstissä jatkossa Sandvikilla viitataan yritykseen Sandvik Mining and Construction Oy.

3 NAVIGOINTI, KOORDINAATISTOT JA VISUALISOINTI

Poralaitteella on yksi tehtävä tunnelia rakennettaessa. Poralaite valmistaa tunnelin perän räjähdyspanoksia varten. Poraaminen on siis keino mahdollistaa räjäytystyö, jolla tunnelia rakennetaan. Räjähdyksen tarkoitus on irrottaa kivimassaa oikean profiilin mukaan ja mahdollisimman pienellä vahingolla ympäröivään kivimassaan. (Maidl, Thewes, ja Maidl 2013, 192)

Yleensä tunnelin halutaan sijoittuvan tiettyyn paikkaan ja kulkevan tiettyyn suuntaan. Vaatimuksena on räjäytyksen tarkka hallinta. Edellytyksenä tähän on reikien poraaminen tarkasti haluttuun paikkaan räjäytettävässä tunnelin seinässä. Paikan lisäksi reiän syvyys ja kulma täytyy olla juuri oikeat. Tämä tarkoittaa, että poralaitteen ja porattavan pinnan sijainnit toisiinsa nähden täytyy tietää tarkasti. (O'reilly ja Miitel 2018, 1–2)

Sijaintien selvittämiseen tarvitaan navigointia. Normaalisti navigointimenetelmät käyttävät muodossa tai toisessa satelliitteja hyödyksi. Maanpinnan alapuolella satelliittisignaalit ovat hyvin heikkoja tai niitä ei ole ollenkaan. Näin ollen täytyy käyttää vaihtoehtoisia menetelmiä. Yleensä menetelmästä riippumatta maanmittaaja määrittelee vaihtelevin keinoin tunnelin suunnan ja merkkää viitepisteitä. Kun poralaite on sijoitettu karkeasti porattavan pinnan eteen, täytyy poralaitteen ja maanmittaajan määrittelemien pisteiden suhde toisiinsa selvittää. (O'reilly ja Miitel 2018, 2)

3.1 Maanalainen paikannus ja navigointi

Opinnäytetyöhön liittyvässä navigoinnissa ei tarkoiteta perinteistä navigointia. Perinteisellä navigoinnilla viitataan navigointiin, joka ohjaa käyttäjän paikasta a paikkaan b. Sen sijaan navigoinnilla työn yhteydessä tarkoitetaan kohteiden sijaintien määrittämistä toisiinsa nähden. Maanalaisissa poralaitteissa on monenlaisia navigointimenetelmiä valmistajasta, käyttäjästä ja käyttökohteesta riippuen.

Navigointia käytetään tietokoneavusteisissa poralaitteissa, joiden määrä kasvaa, mutta on yhä olemassa poralaitteita, joissa ei käytetä minkäänlaista paikannusta.

Tarkka poraaminen on kuitenkin perusta ekonomiselle räjäytykselle ja tunnelin rakentamiselle. Tietokone ei kuitenkaan yksin kykene porausta toteuttamaan vaan se vaatii poraussuunnitelman, jota se kykenee seuraamaan. Tätä varten luodaan porauskaavio. Porauskaavio sisältää porattavien reikien sijainnit ja muut tarvittavat tiedot. Porauskaavio voidaan suunnitella toimistossa CAD-ohjelmia hyödyntäen ja siirtää porauslaitteen tietokoneella haluttua mediaa käyttäen. (Maidl, Thewes, ja Maidl 2013, 281)

Tietokoneavusteinen poraaminen tuo monia etuja tarkkuuden lisäksi porausoperaatioon. Reikiä ei enää tarvitse merkata käsin ja porauskaaviota ei tarvitse luoda käsin tunnelin seinämään. Porauksen aikaiset muutokset on helppo toteuttaa, sillä kaikki poraukseen liittyvä data on helposti saatavilla. (Maidl, Thewes, ja Maidl 2013, 282)

Poralaitteen tietokone tarvitsee suunnitelman lisäksi poralaitteen sijainnin suhteessa porauskaavioon ja tunnelin seinämään. Yksi tapa on käyttää lasersädettä tunnelissa. Jos lasersäteen sijainti on tunnettu, voidaan sitä käyttää määrittelemään poralaitteen sijainti. Poralaitteen puomissa sijaitseva ohjaukko ohjaa lasersäteeseen, jolloin poralaitteen sijainti saadaan selville. (Maidl, Thewes, ja Maidl 2013, 281)

Laserin käyttö ei kuitenkaan ole ainut keino selvittää poralaitteen sijainti. Toinen yleinen tapa on käyttää takymetria tai teodoliittia. Takymetri on yleisesti rakennustyömailla käytetty maanmittauslaite. Takymetri kykenee elektronisesti mittaamaan kulmia ja etäisyyksiä kohteeseen sekä suorittamaan joitakin laskentoja integroidulla prosessorilla (Kennie ja Petrie 2014, 28). Teodoliitti kykenee yleensä takymetrin poiketen mittaamaan pelkästään kulmia.

Esimerkiksi yritys Gebr. Eickhoff Maschinenfabrik und Eisengießerei GmbH käyttää teodoliittia poralaitteen paikantamiseksi. Menetelmässä poralaitteessa sijaitsee kaksi prismaa, joiden sijainti suhteessa poralaitteeseen tunnetaan. Tunnelissa sijaitsee motorisoitu teodoliitti, joka mittaa prismojen sijainnin. Teodoliitin sijainti tunnelissa tiedetään, jolloin mittauksen jälkeen tiedetään myös poralaitteen sijainti tunnelissa. Tunnelin valmiissa osassa sijaitsee navigointipisteitä, joiden

avulla teodoliitin sijainti voidaan tarvittaessa määrittää uudelleen. (Maidl, Thewes, ja Maidl 2014, 265)

Poralaite sisältää useita eri mittalaitteita ja antureita mm. kaltevuusantureita, puomin kulma-antureita ja puomin sylinterien asentoantureita. Poralaitteen tietokone pystyy näiden anturitietojen ja teodoliitin mittaustietojen avulla laskemaan puomin pään koordinaatit. Järjestelmä pystyy nyt näyttämään puomin pään sijainnin tunneliprofilissa. (Maidl, Thewes, ja Maidl 2014, 265–266)

Suurin osa poralaitteissa käytetyistä navigointimenetelmistä perustuu tavalla tai toisella joko tunnelilaserien tai maanmittausinstrumentin käyttöön. Maanmittausinstrumentti voi olla aikaisemmin mainittu teodoliitti tai modernimpi takymetri. Esim. yrityksen ZED Instruments Ltd laitteissa käytetään laseria poralaitteen ja porattavan kohteen sijaintien ratkaisemiseen. Järjestelmään on kuitenkin mahdollista lisätä teodoliitti, jolloin voidaan myös selvittää poralaitteen sijainti globaalissa koordinaatistossa. (Maidl, Thewes, ja Maidl 2014, 267)

3.2 Koordinaatistot ja koordinaatistomuunnokset

Navigointia varten tarvitsee määrittää erilaisten kohteiden sijainti toisiinsa nähden esim. poralaitteen ja porattavan seinämän. Kuitenkin esim. porattavan seinämän määritelmä on varsin abstrakti ja se vaihtelee tapauskohtaisesti. Sen sijaan kohteille voidaan määritellä omat koordinaattisysteemit ja esittää käsiteltävät kohteet omissa koordinaatistoissaan. Käytännössä kohteet esitetään omissa koordinaatistoissaan kolmiulotteisina pisteinä tai vektoreina. Laskemalla koordinaatistojen väliset sijainnit toisiinsa nähden voidaan määritellä tarvittavien kohteiden väliset sijainnit.

Koordinaatistojen väliseen muuntamiseen liittyy kaksi asiaa, koordinaatiston kierto ja koordinaatiston origon siirtyminen. Jotta pistettä tai vektoria voidaan kuvata toisessa koordinaatistossa, täytyy se muuntaa koordinaatistojen välisen kierron ja siirron mukaan.

Kiertoa 2- tai 3-ulotteisessa avaruudessa voidaan kuvata matriisilla. Tällä matriisilla voidaan kertoa pistettä ja lopputuloksena saadaan pisteen sijainti kohdekoordinaatistossa kierron jälkeen. Navigointimenetelmien tapauksessa käsiteltävät koordinaatistot ovat 3-ulotteisia, jolloin kiertomatriisi on 3x3 matriisi. (Boyd ja Vandenberghe 2018, 129)

$$M_r = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \quad (1)$$

Kiertomatriisia käytetään kertomalla sillä vektoria tai pistettä. Esimerkiksi kaavan (1) matriisia voidaan käyttää kolmiulotteisen vektorin (kaava 2) kanssa kaavojen (3) ja (4) mukaisesti. Vektorin alkioit edustavat koordinaatiston x-, y- ja z-koordinaatteja.

$$v = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2)$$

$$v_r = M_r \cdot v \quad (3)$$

$$v_r = \begin{bmatrix} m_{11}x + m_{12}y + m_{13}z \\ m_{21}x + m_{22}y + m_{23}z \\ m_{31}x + m_{32}y + m_{33}z \end{bmatrix} \quad (4)$$

Kaavasta (4) saatu vektori v_r on nyt kiertynyt matriisin M_r mukaisesti. Koordinaatistomuunnoksessa tämä riittää, jos molempien koordinaatistojen origot sijaitsevat samassa pisteessä. Navigointimenetelmissä käytetyissä koordinaatistoissa origot eivät sijaitse samassa pisteessä, jolloin koordinaatistojen välillä tapahtuu myös origon siirto.

Koordinaatiston origon siirtäminen on yksinkertaisempaa. Tähän ei tarvita kuin yksi 3-ulotteinen vektori (kaava 5).

$$v_t = \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \quad (5)$$

Jos kiertoa ei tapahdu niin vektori v muunnetaan uuteen koordinaatistoon kaavojen (6) ja (7) mukaisesti.

$$v_r = v + v_t \quad (6)$$

$$v_r = \begin{bmatrix} x + dx \\ y + dy \\ z + dz \end{bmatrix} \quad (7)$$

Työssä tarvitsee käytännössä aina suorittaa molemmat operaatiot. Koordinaatiston kiertäminen ja siirtäminen voidaan toteuttaa yhdellä laskutoimituksella. Tämä vaatii, että siirrettävästä vektorista tehdään ns. neliulotteinen. Vektoriin lisätään neljäs alkio, jonka arvo on aina 1 (kaava 8). (Zahneisen ja Ernst 2016, 275)

$$v = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (8)$$

Tämä täytyy tehdä, koska uusi käytettävä muunnosmatriisi tulee olemaan 4x4 matriisi. Tällä 4x4 matriisilla ei voi kertoa perinteistä 3-ulotteista vektoria. Myöhemmin todistetaan miksi neljäs lisätty alkio mahdollistaa koordinaatiston siirron samassa operaatiossa kierron kanssa. Tässä täytyy kuitenkin olla tarkkana, sillä neljännen alkion täytyy olla 1.

$$A = \begin{bmatrix} M_r & v_t \\ 0 & 1 \end{bmatrix} \quad (9)$$

Kaavassa (9) on esitetty tämä uusi matriisi A yksinkertaistetussa muodossa. Kaavasta nähdään, että matriisi sisältää sekä kiertomatriisin M_r ja siirtovektorin v_t . Tämä voidaan avata kaavan (10) mukaisesti.

$$A = \begin{bmatrix} m_{11} & m_{12} & m_{13} & dx \\ m_{21} & m_{22} & m_{23} & dy \\ m_{31} & m_{32} & m_{33} & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Muunnosmatriisilla A voidaan nyt kertoa haluttua neliulotteista vektoria ja saadaan tulokseksi neliulotteinen vektori, joka on siirretty kohdekoordinaatistoon. Operaation suoritus näkyy kaavoissa (11) ja (12).

$$v_r = \begin{bmatrix} m_{11}x + m_{12}y + m_{13}z + dx \cdot 1 \\ m_{21}x + m_{22}y + m_{23}z + dy \cdot 1 \\ m_{31}x + m_{32}y + m_{33}z + dz \cdot 1 \\ 0 \cdot x + 0 \cdot y + 0 \cdot z + 1 \cdot 1 \end{bmatrix} \quad (11)$$

$$v_r = \begin{bmatrix} m_{11}x + m_{12}y + m_{13}z + dx \cdot 1 \\ m_{21}x + m_{22}y + m_{23}z + dy \cdot 1 \\ m_{31}x + m_{32}y + m_{33}z + dz \cdot 1 \\ 1 \end{bmatrix} \quad (12)$$

Kaavoista (11) ja (12) nähdään muutama tärkeä asia. Ensinnäkin siirretyn vektorin neljäs alkio on edelleen 1. Tämä toteutuu aina riippumatta kiertomatriisista tai siirtovektorista. Sen sijaan halutaan tarkastella ensimmäistä 3 alkioita. Näille alkioille tapahtuu kierto samalla tavalla kuin aikaisemminkin kaavassa (4), mutta nyt jokaiseen alkioon lisätään yksi luku. Tämä luku on vektorin neljäs alkio kerrottuna siirtovektorin alkioilla. Tässä tapahtuu koordinaattisysteemin origon siirron lisääminen muunnettavaan vektoriin. Tämän takia vektorin neljännen alkion täytyy olla 1. Muuten toteutettava siirto skaalautuisi ja lopputulos ei olisi enää luotettava. Tällä yhdellä operaatiolla päästään samaan lopputulokseen kuin toteuttamalla kierto ja siirto erikseen. Tätä 4x4 matriisia kutsutaan homogeeniseksi siirtomatriisiksi (Zahneisen ja Ernst 2016, 275).

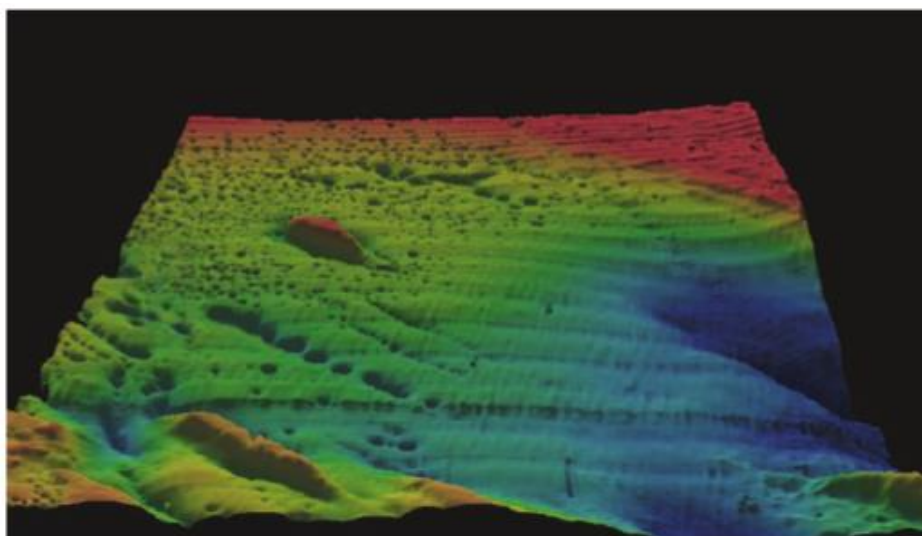
Matriisilaskentaa voidaan käsitellä usealla eri ohjelmalla. Tässä työssä päädyttiin käyttämään Matlab-ohjelmaa. Tämän päätöksen takana on muutamia perusteita. Tärkeimpänä on se, että työssä hyödynnettävät Sandvikin koodit on tehty Matlabilla. Toisekseen Matlab on tarkoitettu numeeriseen laskentaan erityisesti matriiseilla, joka puhuu päätöksen puolesta vahvasti. Matlabissa on myös useita toimintoja, jotka on tarkoitettu visualisointiin.

3.3 Visualisointi

Ihminen harvoin oppii sulkemalla ympäristönsä täysin pois. Sen sijaan oppiminen tapahtuu vuorovaikuttamalla erilaisia työkaluja käyttäen esim. piirtämällä, laske-
malla ja tietokoneavusteisesti. Näköön perustuva tiedonsiirto on ylivoimaisesti
paras tietokoneen ja ihmisen välillä. Ihminen saa näön kautta enemmän tietoa
kuin muilla aisteilla yhteensä. Visualisointi tarkoittaaakin tiedon tai käsitteen kuval-
lista esittämistä. (Ware 2012, 2)

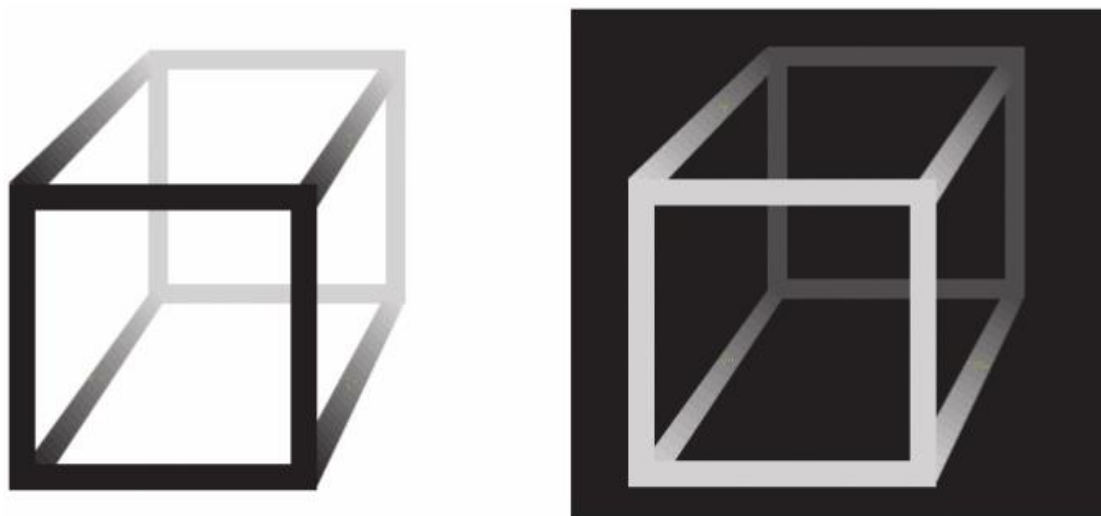
Visualisoinnilla on huomattavia hyötyjä informaation käsittelyssä. Se helpottaa
isojen tietomäärien ymmärtämistä, eri datapisteiden välisten suhteiden ja skaalo-
jen näkyvyyttä, virheiden huomaamista jne. Visualisoinnilla voidaan myös var-
mentaa halutun lopputuloksen saavuttaminen. Monesti ihminen hahmottaa mie-
lessään, miltä lopputuloksen kuuluisi näyttää, mutta ei kykene sitä pelkästä saa-
dusta informaatiosta näkemään. Tuloksen oikeellisuuden tarkastamisessa voi-
daan käyttää visualisointia hyödyksi. (Ware 2012, 3–4)

Visualisointia on monenlaista, mutta opinnäytetyön kannalta oleellista visualisoin-
tia on 3D-visualisointi. Kuvassa 1 nähdään esimerkki meren pohjan visualisoin-
nista kaikuluotaimilla tehtyjen mittauksien perusteella. Normaalisti tämän tyyppi-
nen data esitetään merikarttana hyödyntäen korkeuskäyriä. Kuvassa 1 on sen
sijaan muutettu data graafiseksi esitykseksi tietokoneella. Visualisoinnista näh-
dään välittömästi mm. erilaiset muodot, joita merenpohjaan on muodostunut.
(Ware 2012, 3)



KUVA 1. Esimerkki merenpohjan visualisoinnista (Ware 2012, 3)

Kuvassa 2 nähdään esimerkki syvyysvaikutelman luomisesta visualisoinnissa. Esimerkissä pienennetään kaukaisten kohteiden kontrastia, joka auttaa syvyysvaikutelman luomisessa. (Ware 2012, 280)



KUVA 2. Esimerkki valoisuuden ja kontrastin vaikutuksesta visualisoinnissa (Ware 2012, 280)

4 KÄYTETYT NAVIGOINTIMENETELMÄT

Opinnäytetyössä visualisoidaan neljää Sandvikilla käytettävää navigointimenetelmää. Menetelmiin liittyy koordinaatistoja ja tunnelilinja, joiden ymmärtäminen helpottaa navigointimenetelmien ymmärtämistä.

4.1 Käytettävät koordinaatistot

Työssä käsiteltävissä navigointimenetelmissä käytetään kolmea eri koordinaatistoa, joiden välisiä koordinaatistomuunnoksia työssä visualisoidaan.

Projektikoordinaatisto

Tunnelilinja suunnitellaan projektikoordinaatistoon. Projektikoordinaatistona käytettävä koordinaatisto riippuu kohteesta ja asiakkaasta. Projektikoordinaatistona voi toimia esim. maan globaali koordinaatisto. Projektikoordinaatisto voi olla vasemman- tai oikeankätinen ja tämä otetaan huomioon navigointitapahtumassa. Projektikoordinaatisto voi ajatella ns. pääkoordinaatistona ja kaikki muut koordinaatistot sijaitsevat jossain tässä koordinaatistossa. Projektikoordinaatisto ei kuitenkaan ole pakollinen. (Sandvik Mining and Construction ei pvm., 3)

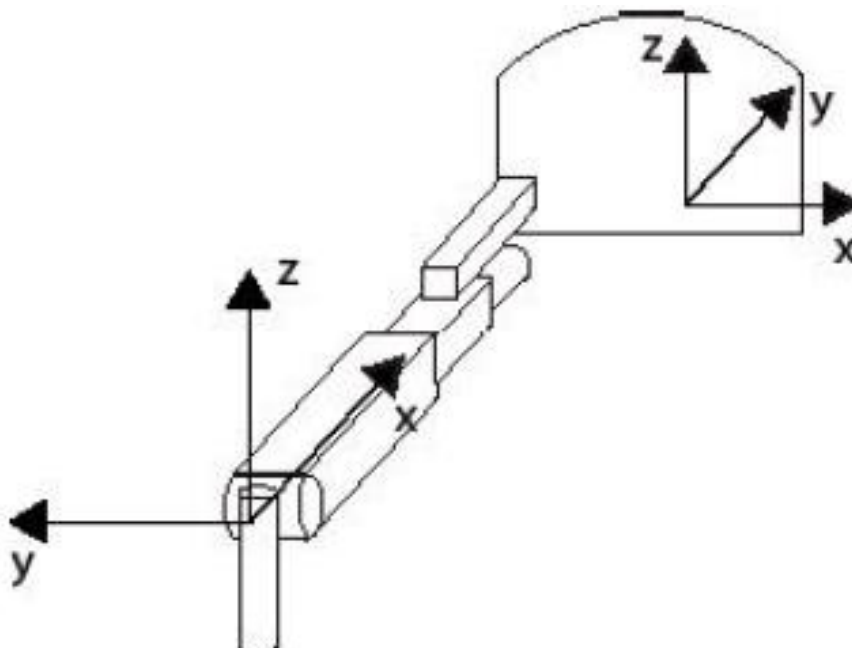
Poraussuunnitelman koordinaatisto

Poraussuunnitelman koordinaatiston y-akseli osoittaa tunnelin suuntaan, x-akseli y-akselista kohtisuoraan oikealla ja z-akseli kohtisuoraan ylöspäin xy-tasoon verrattuna. Koordinaatisto on oikeankätinen. Navigointitaso luodaan poraussuunnitelman koordinaatistoon. Navigointitaso sijaitsee poraussuunnitelman koordinaatiston xz-tasossa. Navigointitasossa sijaitsee porauskaavio, johon on merkitty porattavien reikien sijainnit. (Sandvik Mining and Construction ei pvm., 3)

Poralaitteen koordinaatisto

Poiketen poraussuunnitelman koordinaatistosta poralaitteen koordinaatiston x-akseli osoittaa poralaitteen mukaisesti eteenpäin, y-akseli x-akselista kohtisuoraan vasemmalle ja z-akseli kohtisuoraan ylöspäin xy-tasoon verrattuna. Poralaitteen koordinaatisto on oikeakätinen. Kuvasta 3 nähdään hyvin poralaitteen ja poraussuunnitelman koordinaatistojen akselien erot. Poralaitteen koordinaatiston

origo sijaitsee vasemman puomin kääntöakselissa puomin nostoliitoksen korkeudella. (Sandvik Mining and Construction ei pvm., 3)



KUVA 3. Poralaitteen ja poraussuunnitelman koordinaatistot (Sandvik Mining and Construction ei pvm., 5)

4.2 Tunnelilinja

Navigointisuunnitelmaa varten Sandvikilla on iSure-ohjelmisto. Ohjelmalla luodaan tunnelin tekoa varten poraussuunnitelma (Zou 2017, 540). Yhtenä ulostulona ohjelma antaa tunnelilinjan, joka sisältää kaarretaulukon. Tätä taulukkoa käytetään työssä käsiteltävistä navigointimenetelmistä kahdessa. Tunnelilinja on tallennettuna XML-tiedostoon, joka noudattaa IREDES-standardia.

IREDES (International Rock Excavation Data Exchange Standard) standardin on tarkoitus helpottaa datan vaihtoa yleisten kaivoslaitteiden välillä. Standardi mahdollistaa tiedon vaihdon laitteen valmistajasta riippumatta. Se on suunniteltu kansainväliseksi ja mahdollisimman kattavaksi. (McBain ja Timusk 2012, 4)

IREDES käyttää XML (eXtensible Markup Language) kieltä, joka on yleinen ohjelmisto- ja tietokantajärjestelmissä. XML:n muotoilu itsessään on standardoitu, joka tekee siitä laajasti käytettävään. XML on myös suoraviivaista ja yksinkertaista

ja näin ollen ihmiselle luettavissa. IREDES rakentaa XML-tiedoston XML-kaavioiden avulla (Carter 2019, 48). Kaaviot määrittävät tarkemmin XML-tiedostossa sallitut ominaisuudet ja datatyypit.

```
<CurveTable>
  <IR:CurvePointId>1</IR:CurvePointId>
  <IR:PointCoord>
    <IR:PointX>6822079.492</IR:PointX>
    <IR:PointY>24477134.673</IR:PointY>
    <IR:PointZ>114.082</IR:PointZ>
  </IR:PointCoord>
  <IR:PointG>0</IR:PointG>
  <IR:PegNumber>2</IR:PegNumber>
  <IR:Comment />
</CurveTable>
```

KUVA 4. Esimerkki XML-tiedostosta (Sandvik Mining and Construction Oy)

Esimerkissä (kuva 4) nähdään tunnelilinjatiedostossa olevan kaarretaulukon yksi piste. Näitä pisteitä voi yhdessä tunnelilinjatiedostossa olla satoja. Kuvasta nähdään, että XML-kieli on selvästi luettavaa. XML:ssä käytetään avaus- ja sulkuta-geja, joiden väliin on kirjoitettu kyseisen elementin arvo. Kaarretaulukon pisteessä on luettavissa pisteen id (kuva 4: CurvePointId), koordinaatit projektikoordinaatistossa (kuva 4: PointCoord), kallistuskulma (kuva 4: pointG), paalunumero (kuva 4: PegNumber) ja vapaaehtoinen kommentti (kuva 4: Comment).

Paalunumero kuvaa sijaintia tunnelissa. Paalunumero on yleensä tunnelin pituus xy-tasossa eli ylhäältä päin katsottuna. Paalunumero ei ota siis huomioon tunnelin korkeuden vaihteluja. Usein tunnelia rakentaessa porauskohdat eivät osu kaarrepisteiden kohdalle. Tällöin kaarretaulukosta interpoloidaan tarvittava tieto annetulla paalunumerolla. (Sandvik Mining and Construction ei pvm., 7)

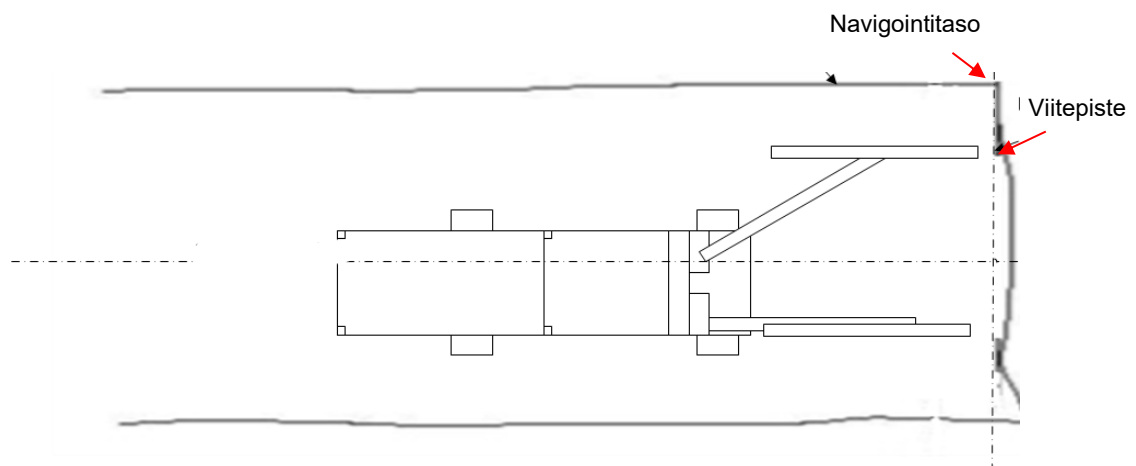
4.3 Porakruununavigointi

Porakruununavigointi on käsiteltävistä tavoista yksinkertaisin. Porakruununavigointi voidaan suorittaa ilman projektikoordinaatistoa. Tämä tarkoittaa, että tun-

nelin ei tarvitse sijaita tarkassa paikassa muuhun ympäristöön nähden. Porakruunavigointi on yleisin käsiteltävistä neljästä tavasta. (Sandvik Mining and Construction ei pvm., 9)

Menetelmässä maanmittaaja mittaa etukäteen tunnelin ja suorittaa navigoinnin vaatimat mittaukset. Maanmittaaja merkkää tunnelin kulkusuunnan viivana tunnelin lattiaan tai kattoon sekä yhden viitepisteen tunnelinperään. Viitepisteen koordinaatit merkataan poraussuunnitelmaan poraussuunnitelman koordinaatistossa. Navigointitaso sijaitsee tässä samassa koordinaatistossa. Maanmittaajan käyttämällä menetelmällä tämän pisteen mittaamiseen ja merkkäämiseen ei ole merkitystä navigoinnin kannalta. (Sandvik Mining and Construction ei pvm., 9)

Poralaitteen operaattori sijoittaa visuaalisesti poralaitteen samansuuntaiseksi maanmittausinsinöörin merkkääman viivan kanssa (kuva 5). Poralaitteen puomi liikutetaan viitepisteeseen. Nyt viitepisteen koordinaatit tiedetään myös poralaitteen koordinaatistossa. Nämä koordinaatit syötetään poralaitteen ohjausjärjestelmään. (Sandvik Mining and Construction ei pvm., 10)



KUVA 5. Porakruunavigointi (Sandvik Mining and Construction ei pvm., 10, muokattu)

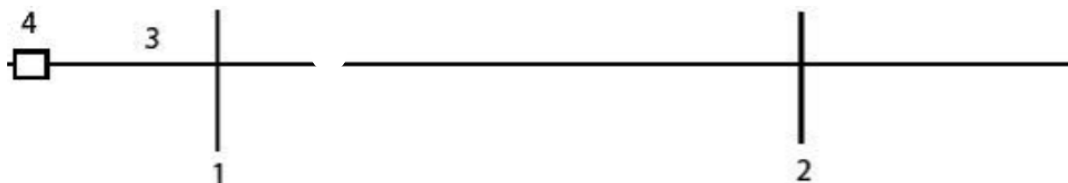
Viitepisteen koordinaatit tiedetään nyt molemmissa koordinaatistoissa. Viitepisteen koordinaateilla ja poralaitteen anturitiedoilla järjestelmä kykenee laskemaan muunnosmatriisin poralaitteen koordinaatistosta poraussuunnitelman koordinaatistoon. (Sandvik Mining and Construction ei pvm., 10)

Menetelmä voidaan myös toteuttaa ilman viitepisteiden merkkausta etukäteen. Tällöin operaattori visuaalisesti ja iteroivasti sijoittaa poraussuunnitelman tunne-
linseinään manuaalisesti säätämällä viitepisteen koordinaatteja. Tämä ei ole kui-
tenkaan opinnäytetyön kannalta oleellista. (Sandvik Mining and Construction ei
pvm., 10)

4.4 Lasernavigointi ilman kaarretaulukkoa

Lasernavigaatiota ilman kaarretaulukkoa voidaan käyttää erityistilanteissa esim.
poratessa suoraa tunnelia (kuva 6) tai jos laserin risteyskohdat ja kulmat on las-
kettu etukäteen maanmittaajan toimesta nykyiselle sijainnille tunnelissa. (Sandvik
Mining and Construction ei pvm., 7)

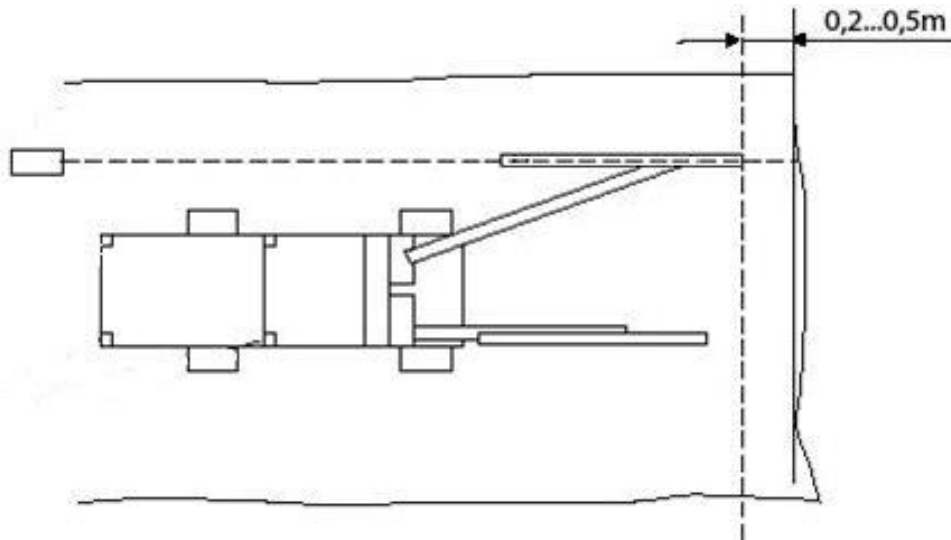
1. Navigointitaso ensimmäisen porauksen aikana
2. Navigointitaso toisen porauksen aikana
3. Laserlinja
4. Yhteinen tunnelilaseri kaikille porauksille



KUVA 6. Lasernavigointi ilman kaarretaulukkoa suorassa tunnelissa (Sandvik Mi-
ning and Construction ei pvm., 8, muokattu)

Poraussuunnitelmassa voi olla tunnelilaserin leikkauspisteen koordinaatit ja leik-
kauskulmat. Leikkauspiste on piste, jossa tunnelilaseri ja navigointitaso kohtaa-
vat. Jos tunnelilaserien tietoja ei ole suunnitelmassa valmiina, maanmittaaja las-
kee kyseiselle sijainnille laserin leikkauspisteen ja antaa tiedot poralaitteen ope-
raattorille. Operaattori tallentaa tunnelilaserin tiedot poraussuunnitelmaan.
(Sandvik Mining and Construction ei pvm., 16)

Poralaitteen puomi liikutetaan niin, että tunnelilaseri kulkee puomissa sijaitsevien
tähtämien läpi. Navigointitaso luodaan syöttökiskon etupisteen kohdalle (kuva
7). (Sandvik Mining and Construction ei pvm., 17)



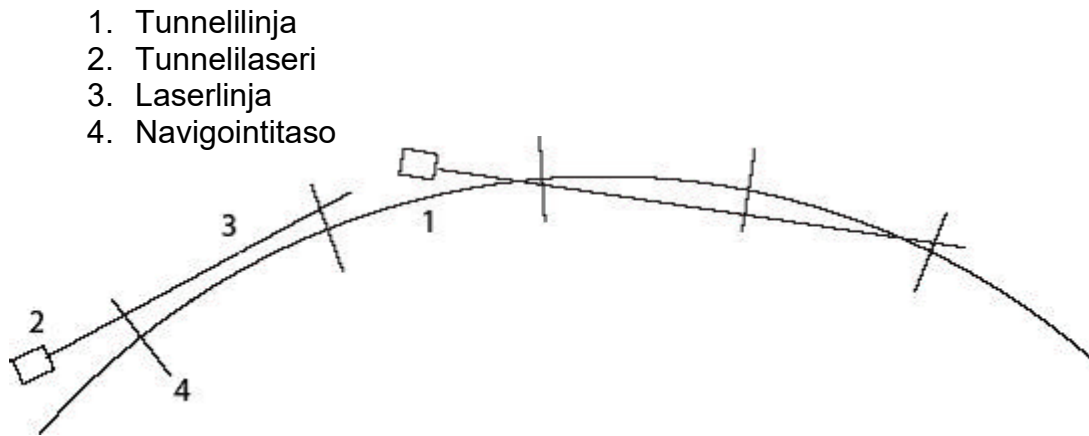
KUVA 7. Lasernavigointi ilman kaarretaulukkoa (Sandvik Mining and Construction ei pvm., 17)

Järjestelmä kykenee laskemaan valitun puomin sijainnin ja tunnelilaserin datan avulla muunnosmatriisin poralaitteen koordinaatistosta poraussuunnitelman koordinaatistoon. (Sandvik Mining and Construction ei pvm., 16)

4.5 Lasernavigointi kaarretaulukon kanssa

Poiketen edellisistä menetelmistä, joissa laskettiin yksi muunnosmatriisi poralaitteen koordinaatistosta poraussuunnitelman koordinaatiston, tarvitsee nyt laskea kaksi muunnosmatriisiä. Ensin suoritetaan kaarrelaskenta, joka palauttaa muunnosmatriisin projektikoordinaatistosta poraussuunnitelman koordinaatistoon. (Sandvik Mining and Construction ei pvm., 19–20)

Poralaitteen järjestelmään on ennen navigointitapahtumaa ladattu tunnelisuunnitelma, joka sisältää navigoinnin tarvitsevan tunnelilinjan (Sandvik Mining and Construction ei pvm., 7). Kuvassa 8 on esimerkkikuva tunnelilinjasta. Siinä näkyy tunnelilaserit sekä navigointitasot. Poraus ja navigointi suoritetaan navigointitasojen kohdalla.



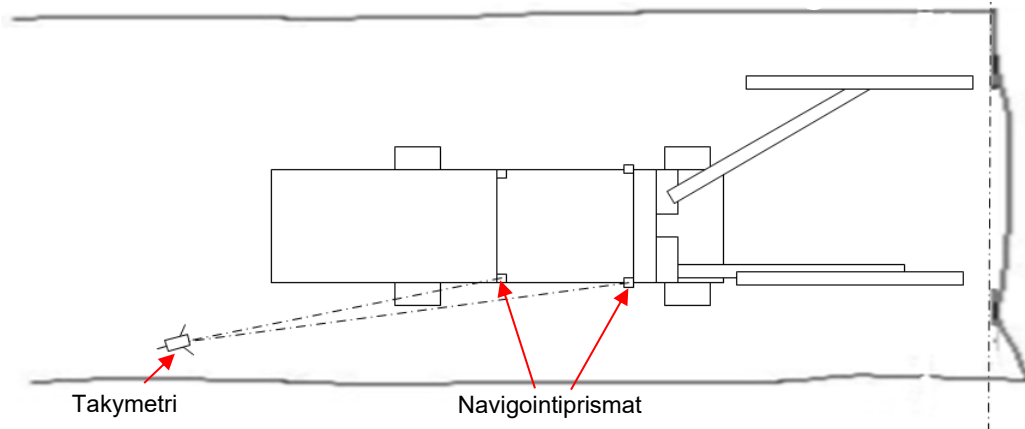
KUVA 8. Tunnelilinja (Sandvik Mining and Construction ei pvm., 7, muokattu)

Navigointia varten tarvitaan nykyisen porauspinnan paalunumero. Paalunumeron määrittelee maanmittaaja tai se voidaan arvioida edellisen tunnetun paalunumeron avulla. Poralaitteen järjestelmä kykenee laskemaan tunnelilaserien risteyspisteet ja kulmat navigointitason kanssa paalunumeron avulla. Seuraavaksi voidaan laskea projektikoordinaatiston ja poraussuunnitelman koordinaatiston välinen muunnosmatriisi. (Sandvik Mining and Construction ei pvm., 7)

Poralaitteen sijainti navigointitasoon selvitetään samalla tavalla kuin lasernavigoinnissa ilman kaarretaulukkoa eli liikutetaan puomi lasersäteeseen, jonka jälkeen pystytään laskemaan muunnosmatriisi poralaitteen koordinaatiston ja poraussuunnitelman koordinaatiston välille. Näillä kahdella matriisilla pystytään muuttamaan koordinaatteja kaikkien kolmen koordinaatiston välillä. (Sandvik Mining and Construction ei pvm., 20)

4.6 Takymetrinavigointi

Takymetrinavigoinnissa ei hyödynnetä tunnelilasereita ollenkaan, vaan paikannus toteutetaan sarjalla takymetrimittauksia. Ensin selvitetään takymetrin sijainti projektikoordinaatistossa. Tämä toteutetaan mittaamalla kaksi tai useampaa tunnettua pistettä tunnelissa takymetrillä. (Sandvik Mining and Construction ei pvm., 8)



KUVA 9. Takymetrinavigointi (Sandvik Mining and Construction ei pvm., 8, mukattu)

Seuraavaksi selvitetään takymetrin sijainti poralaitteeseen nähden. Poralaitteessa sijaitsee kaksi prismaa, joiden sijainti poralaitteen koordinaatistossa tunnetaan. Takymetri ja poralaitteen navigointiprismat näkyvät kuvassa 9. Takymetrillä suoritetaan mittaukset molempiin prismoihin, jolloin poralaitteen sijainti saadaan selville. Näiden mittausten avulla poralaitteen järjestelmä kykenee nyt laskemaan poralaitteen koordinaatiston ja projektin koordinaatiston välisen muunnosmatriisin. (Sandvik Mining and Construction ei pvm., 8)

Navigointitasen sijainti selvitetään seuraavaksi. Tämä voidaan selvittää mittaamalla navigointitasen paikka takymetrillä tai osoittamalla navigointitasen paikka poralaitteen puomilla. Poralaitteen järjestelmä voi tämän jälkeen laskea muutoksen poraussuunnitelman koordinaatistosta projektikoordinaatiston. Näillä kahdella lasketulla muunnosmatriisilla voidaan määrittää porauksen vaatimat tiedot. (Sandvik Mining and Construction ei pvm., 8)

5 NAVIGOINTITAPAHTUMIEN VISUALISOINTI

Seuraavaksi käsitellään edeltävässä kappaleessa esiteltyjen navigointimenetelmien visualisointi. Navigointimenetelmät käyttävät osittain samoja funktioita ja toimintoja, jolloin on järkevää hyödyntää modulaarista ohjelmointia. Kahdessa navigointitavassa tarvitaan XML-tiedosto, joka parsitaan ensin käytettävään muotoon.

Yhteiset funktiot luodaan Matlabin script-tiedostoon. Tiedostoon kirjoitetaan pelkästään ohjelmakoodia. Navigointimenetelmien visualisointi tehdään Matlabin livescript-tiedostolla. Tiedosto eroaa normaalista script-tiedostosta, sillä siihen voidaan sisällyttää normaalia tekstiä ja koodia sekaisin. Tällä tavalla saadaan luotua siistimpi työkalu, johon voidaan lisätä esim. työhjeita ja kuvia avustamaan työkalun käytössä. Lisäksi livescript-tiedosto tukee muita ominaisuuksia, joita työssä hyödynnetään.

5.1 Matlab

Opinnäytetyön päätyökaluna on Matlab-ohjelmisto. Matlab on kaupallinen numeerisen laskennan ohjelma. Ohjelmisto jakaa nimensä siinä käytettävän kielen kanssa. Ohjelmistoa käytetään pääasiassa matemaattisten ongelmien ratkaisuun ja ohjelman nimi Matlab onkin lyhenne sanoista "matrix laboratory". Ohjelmassa tyypillinen käytetty tiedostorakenne onkin matriisi. Matlab-ohjelmaa voidaan käyttää puhtaasti matemaattisten ongelmien lisäksi mm. mallintamiseen, simulointiin, data-analysointiin, visualisointiin ja algoritmien kehitykseen. (Gilat 2014, luku Johdanto)

Matlabissa on komentoikkuna, jota voidaan käyttää erilaisten komentojen ja funktioiden suorittamiseen. Komentoriville kirjoitetaan haluttu koodi ja tämä toteutuu enter-näppäimen painalluksella. Komentorivi on hyvä työkalu lyhyiden operaatioiden tai valmiiden ohjelmien käyttöön. Koodin kasvaessa komentorivi käy kuitenkin epäkäytännölliseksi. (Mathworks ei pvm)

Matlabissa on oma tekstinkäsittelyohjelma, jolla kirjoitetaan ohjelmakoodeja käyttäen Matlab-ohjelmointikieltä. Nämä koodit voidaan tallentaa omiksi ohjelmikseen .m-loppuisiin tiedostoihin. Matlab-koodeihin voidaankin tästä syystä viitata .m-tiedostoina (Turk 2019, luku 1). Näitä tiedostoja voi kuka tahansa hyödyntää kenellä on pääsy Matlab-ohjelmaan.

5.2 Yhteiset funktiot

Yhteiset funktiot ovat toimintoja, joita käytetään useammassa navigointimenetelmässä. Ei ole järkevää kirjoittaa jokaiselle menetelmällä samaa toimintoa uudelleen. Sen sijaan toiminnot toteutetaan funktioilla, joita voidaan käyttää mahdollisimman useassa eri käyttötarkoituksessa. Yhteisiä funktioita ovat poralaitteen piirto, navigointitason piirto ja tunnelilinjan piirto.

Poralaitteen piirto

Poralaitteen piirtäminen toteutetaan funktiolla `draw_rig_tm(transform_matrix)`. Funktiolle annetaan argumenttina 4x4-muunnosmatriisi. Jos muunnosta ei haluta tehdä annetaan argumenttina 4x4-yksikkömatriisi. Paluuarvona saadaan piirretty poralaite ja poralaitteen origon sijainti kohdekoordinaatistossa.

Poralaitetta esitetään visualisoinnissa laatikoilla. Laatikot toteutetaan piirtämällä ne yksi sivu kerrallaan. Sivua piirretään käyttämällä Matlabin funktiota `patch`. Kuvassa 10 näkyy laatikon ylä- ja alasivun piirto. Piirtoa varten funktiolle `patch` annetaan sivun neljän kulman koordinaatit. Koordinaatit on määritelty kiinteästi poralaitteen koordinaatistossa ja argumenttina annetun muunnosmatriisin avulla ne muutetaan kohdekoordinaatistoon.

Kuvassa 10 asetetaan muuttujille `x_upper`, `y_upper`, `z_upper` nämä lasketut koordinaatit. Komennolla `"rect(1) = patch(x_upper,y_upper,z_upper,box_color,'FaceAlpha',transparency)"` toteutetaan sivun piirto. Funktion `patch` kolme ensimmäistä argumenttia sisältävät pisteiden `x`, `y` ja `z`-koordinaatit. Jälkimmäiset argumentit eivät ole pakollisia ja liittyvät piirrettävän sivun ulkonäköön. `box_color` muuttujaan on tallennettuna haluttu väri vektorina. `'FaceAlpha'` ja

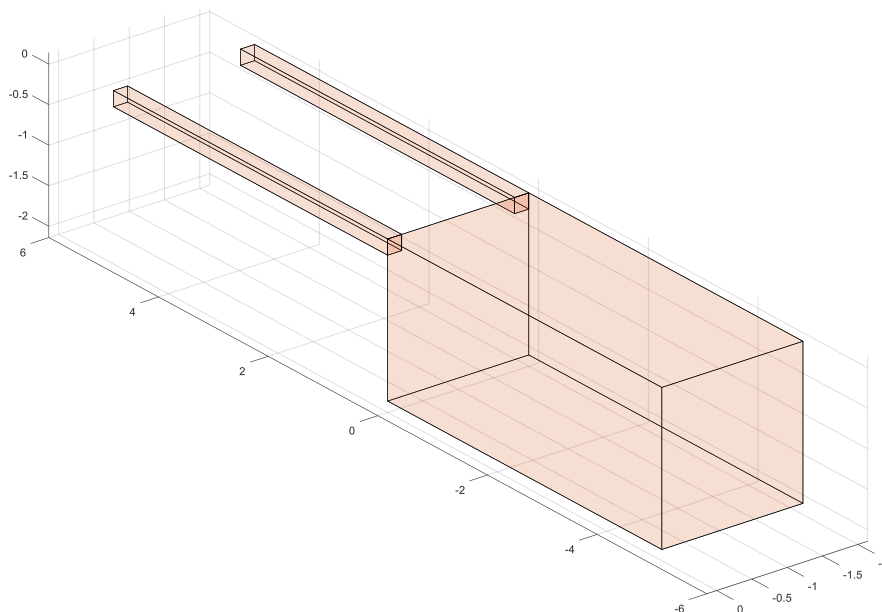
transparency argumentit yhdessä muuttavat piirrettävän sivun läpinäkyvyyttä muuttujaan transparency tallennetun arvon mukaan.

```
% Point coordinates for upper points
x_upper = [point1(1) point2(1) point3(1) point4(1)];
y_upper = [point1(2) point2(2) point3(2) point4(2)];
z_upper = [point1(3) point2(3) point3(3) point4(3)];
% Point coordinates for lower points
x_lower = [point5(1) point6(1) point7(1) point8(1)];
y_lower = [point5(2) point6(2) point7(2) point8(2)];
z_lower = [point5(3) point6(3) point7(3) point8(3)];

%"roof" of the rectangle
rect(1)=patch(x_upper,y_upper,z_upper,box_color,'FaceAlpha',transparency);
hold on
%% "floor" of the rectangle
rect(2)=patch(x_lower,y_lower,z_lower,box_color,'FaceAlpha',transparency);
origin_point = [x_upper(1),y_upper(1),z_upper(1)];
```

KUVA 10. Laatikon sivun piirto

Poralaitteen kaikki sivut piirretään samaa periaatetta käyttäen. Oikeassa käytössä yksittäisiä sivuja ja näiden piirtoja ei erota. Sen sijaan koko poralaite ilmestyy kerralla ja sitä käsitellään yhtenä kappaleena. Kuvassa 11 on piirretty esimerkkikuva funktiolla draw_rig_tm. Argumenttina on käytetty 4x4-yksikkömatrisiä, jolloin koordinaatistomuunnosta ei ole.



KUVA 11. Esimerkki poralaitteen piirrosta

Navigointitason piirto

Navigointitaso piirretään funktiolla draw_plane_tm(t_matrix,planetype), kun tunneliinjaa ei käytetä. Funktiolle annetaan kaksi argumenttia, 4x4-muunnosmatriisi

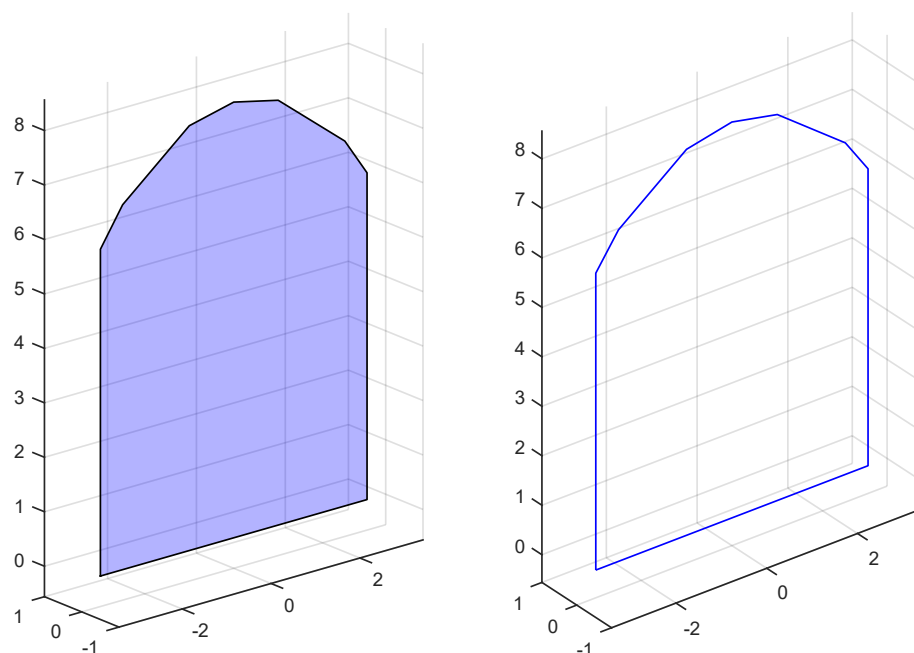
ja navigointitason piirtotyyppi. Funktion paluuarvona saadaan piirretty navigointitaso. Navigointitason piirron toiminta on hyvin samankaltaista poralaitteen piirron kanssa.

Navigointitason profiili luodaan usealla pisteellä. Kuvassa 12 näkyy navigointitason profiilin luonnissa käytetyt pisteet. Navigointitason profiilista voi halutessaan luoda tarkemman lisäämällä näitä pisteitä. Käytetty pistemäärä on kuitenkin riittävä työn tarkoitukseen. Profiilin pisteet kerrotaan argumenttina saadulla muunnosmatriisilla, jolloin navigointitaso siirtyy haluttuun koordinaatistoon.

```
x = [-3 -3 -2.5 -1 0 1 2.5 3 3 -3];
y = [0 0 0 0 0 0 0 0 0 0];
z = [0 6 6.7 7.8 8 7.8 6.7 6.0 0 0];
```

KUVA 12. Navigointitason profiili

Funktion toinen argumentti planetype kertoo piirrettävän navigointitason tyyppin. Ensimmäinen vaihtoehtoista on sama kuin poralaitteen piirroksessa käytetty eli patch funktio. Toinen vaihtoehto on käyttää plot3 funktiota, jonka avulla piirretään ainoastaan navigointitason ulkoreuna. Erot vaihtoehtojen välillä näkyy kuvassa 13. Kaksi vaihtoehtoa tuo visualisointiin mahdollisuuden muokata lopputulosta käyttäjän mieltymyksen mukaan.



KUVA 13. Navigointitasot

Navigointitason piirto tunnelilinjaan

Navigointimenetelmän käyttäessä tunnelilinjaa navigointitaso luodaan funktiolla `plot_nav_plane`. Funktiolla on muutamia lisäominaisuuksia funktioon `draw_plane_tm` verrattuna. Kyseisten kahden funktion sijaan voitaisiin käyttää yhtä funktiota, joka sopisi kaikille navigointimenetelmille. Työmäärä pysyy kuitenkin suurin piirtein samana riippumatta siitä ohjelmoitaisiinko yksi yhteinen funktio tai kaksi eri funktiota. Erottelemalla funktio kahteen osaan selkeytetään myös selvästi funktion käyttökohdetta, mutta tässä kohtaa on enemmän kyse henkilökohtaisesta ohjelmointitavasta.

Funktiolla on kahdeksan argumenttia. Muunnosmatriisin ja tason piirtotyyppin argumentit ovat samat kuin edellisessä funktiossa `draw_plane_tm`. Näiden lisäksi funktiolla `plot_nav_plane` on argumentteina katkon pituus, katkon alku- ja loppupisteen koordinaatit, koordinaatiston kätisyys ja laserin kahden pisteen koordinaatit. Paluuarvoina on piirretyn tason lisäksi mahdollisesti piirretty laser.

Funktio piirtää navigointitason samalla tavalla kuin funktio `draw_plane_tm`. Lisäksi koordinaatiston kätisyys tarkistetaan ja tarvittaessa x-akseli käännetään. Tunnelilaserin ollessa käytössä piirretään se argumentteina annettujen pisteiden mukaan. Tämä on valinnainen ominaisuus funktiossa. Laser piirretään, jos laserpisteet on annettu funktion argumentteina. Kuvassa 14 on ohjelmanosuus, jolla tarkastetaan tämä. Matlabin funktio `nargin` palauttaa funktiolle annettujen argumenttien määrän. Määrän ollessa kahdeksan on laserpisteet annettu, jolloin laser piirretään. Laser piirretään aiemmin mainitulla `plot3` funktiolla.

```

if (nargin==8)
    % If laser is used
    laser = plot3([laserb(1) lasera(1)],[laserb(2) lasera(2)],[laserb(3) lasera(3)], 'r', LineWidth=1);
end

```

KUVA 14. Funktion argumenttimäärän tarkistus

Lisäksi funktio piirtää katkon näkyviin argumentteina annettujen katkon alku- ja loppupisteiden avulla funktiolla `plot3`. Katkon pituus kirjoitetaan katkon päähän näkyviin. Katkon piirtoon liittyvät komennot näkyvät kuvassa 15. Uutena ominaisuutena on nyt funktio `text` (kuva 15: rivi 3). Kyseiselle funktiolle annetaan x-, y- ja z-koordinaatit, sekä kirjoitettava teksti. Katkon pituus on tallennettuna muuttujassa `roundl`. Muuttuja on tyypiltään `double`-muuttuja. Se täytyy muuttaa string-

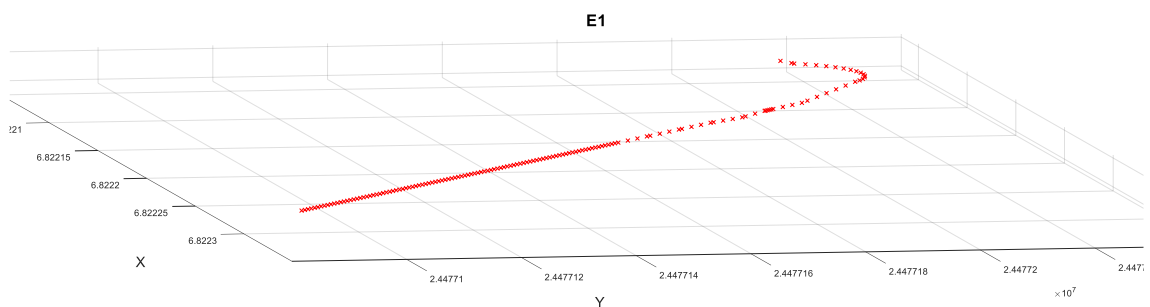
tyyppiseksi, jotta sitä voidaan käyttää funktion `text` kanssa. Tätä varten on Matlabin funktio `num2str`, joka toteuttaa kyseisen muunnoksen.

```
nav_plane(1) = plot3(np1(1),np1(2),np1(3),'mo',LineWidth=1);
nav_plane(2) = plot3([np1(1) np2(1)],[np1(2) np2(2)],[np1(3) np2(3)],'m',LineWidth=1);
nav_plane(3) = text(np2(1),np2(2),np2(3),num2str(round1));
```

KUVA 15. Katkon piirtäminen

Tunnelilinjan piirto

Tunnelilinja piirretään funktiolla `draw_tunnel_line_array`. Funktiolla ei ole paluuarvoa. Funktiolle annetaan argumentteina kaarretaulukko ja kaarretaulukon tiedot. Nämä molemmat saadaan funktion `parse_xml` paluuarvoina, joka esitellään myöhemmin tässä kappaleessa. Funktio piirtää tunnelilinjan pisteet kaarretaulukkoon tallennettujen koordinaattien avulla funktiolla `plot3`. Kuvassa 16 näkyy esimerkki piirretystä tunnelilinjasta.



KUVA 16. Esimerkki tunnelilinjasta

Lisäksi kaarretaulukon tiedoista saadaan akselien nimet ja tunnelilinjan nimi. Kuvassa 17 näkyy miten kuvaajan akselit ja otsikko nimetään. Esim. x-akseli nimetään funktiolla `xlabel`. Funktiolle annetaan argumentteina akselin haluttu nimi. Valinnaisena argumenttina funktiolle on annettu uusi fonttikoko, joka helpottaa akselin nimen lukemista. Kuvaajan nimi annetaan funktiolla `title`. Funktiolle annetaan samat argumentit kuin akselin nimeämisessä. Tämän lisäksi käytetään valinnaista argumenttia, joka ottaa tekstin tulkitsijan pois käytöstä. Jos tulkitsija on käytössä, tunnelilinjan nimessä mahdollisesti esiintyvät erikoismerkit tuottavat halutusta poikkeavaa käyttäytymistä.

```
xlabel(curve_info.x_axis_name,'FontSize',16);
ylabel(curve_info.y_axis_name,'FontSize',16);
xlabel(curve_info.z_axis_name,'FontSize',16);
title(curve_info.plan_id,'FontSize',18,'Interpreter','none');
```

KUVA 17. Kuvaajan akselien ja otsikon nimeäminen

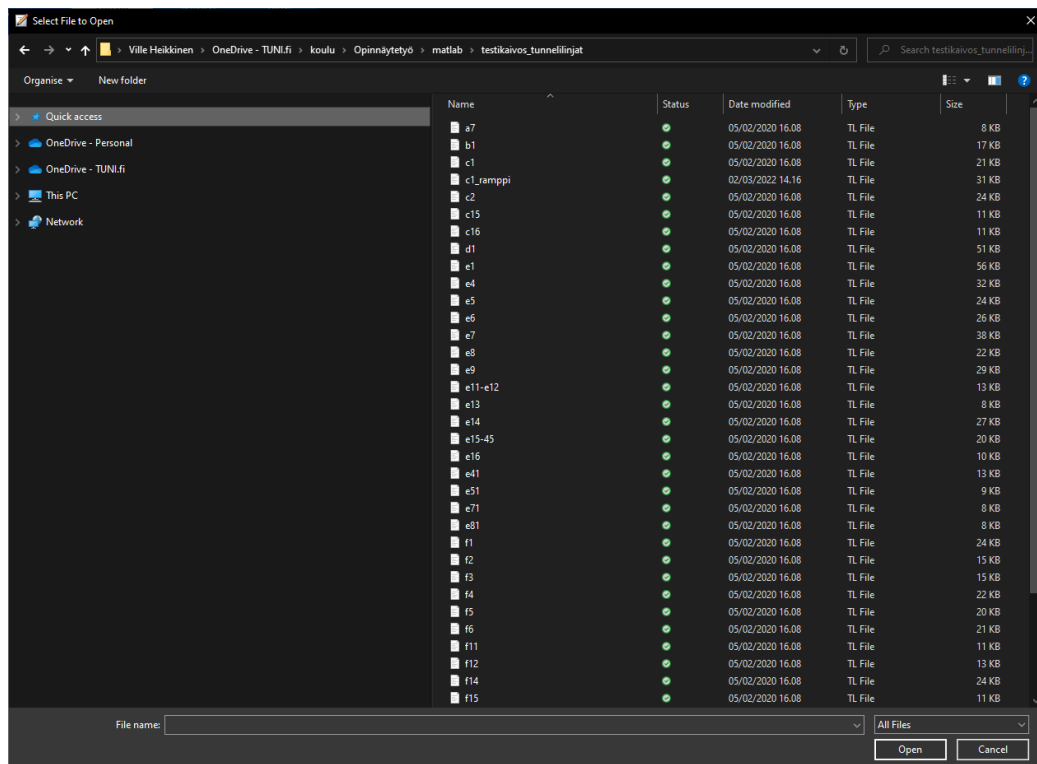
5.3 XML-tiedoston parsiminen

Kaarretaulukko sijaitsee suunnitteluohjelmasta saadussa tunneliinjassa. Tunneliinija on XML-tiedosto, jossa on kaarretaulukon lisäksi muita hyödyllisiä tietoja. Kaarretaulukko täytyy muuttaa tietorakenteeseen, jota navigointifunktiot kykenevät hyödyntämään. Toimintoa varten tehdään funktio `parse_xml`. Funktiolla on kaksi paluuarvoa ja nolla argumenttia.

Ensin käyttäjältä kysytään tiedostoa, jota halutaan käyttää. Seuraavaksi kyseinen tiedosto parsitaan tietorakenteeksi, jota Matlab pystyy käyttämään. Kuvassa 18 näkyy edellä mainitut toimenpiteet. Tiedoston valinta toteutetaan Matlabin sisäisellä funktiolla `uigetfile` (kuva 18: rivi 1), joka avaa tiedostonvalinta näkymän (kuva 19).

```
[file,path] = uigetfile();
path_to_file = strcat(path,file);
unedited_tunnel_line = xml2struct(path_to_file);
```

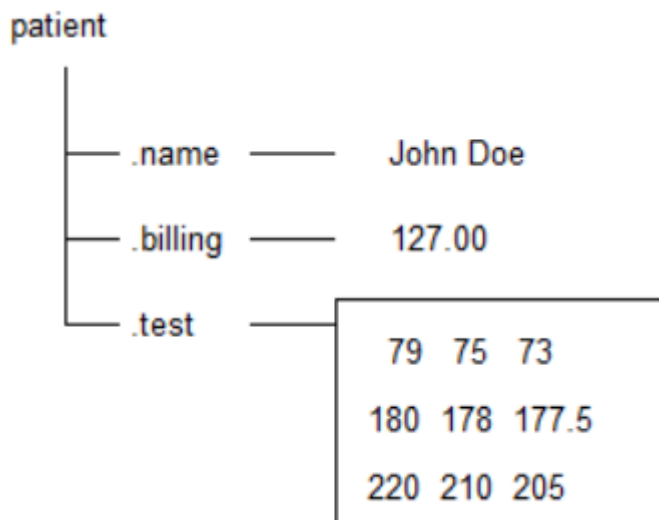
KUVA 18. Tiedoston valitseminen



KUVA 19. Tunneliinijan tiedoston valintanäkymä

Funktiolla `uigetfile` on kaksi paluuarvoa, tiedoston nimi (kuva 18: `file`) ja polku (kuva 18: `path`) tiedostolle. Parsimiseen käytettävä funktio tarvitsee polun tiedostolle, joka sisältää myös tiedoston nimen. Kuvan 18 toinen koodirivi yhdistää aiemmin saadut polun ja tiedostonimen yhdeksi muuttujaksi (kuva 18: `path_to_file`), jota voidaan käyttää parsimisfunktiossa. Funktiota `xml2struct` (kuva 18: rivi 3) käytetään nyt XML-tiedoston parsimiseen. Funktio on ladattu Matlab yhteisön sivuilta ja on kaikkien vapaassa käytössä. Funktiolle annetaan argumenttina XML-tiedoston polku.

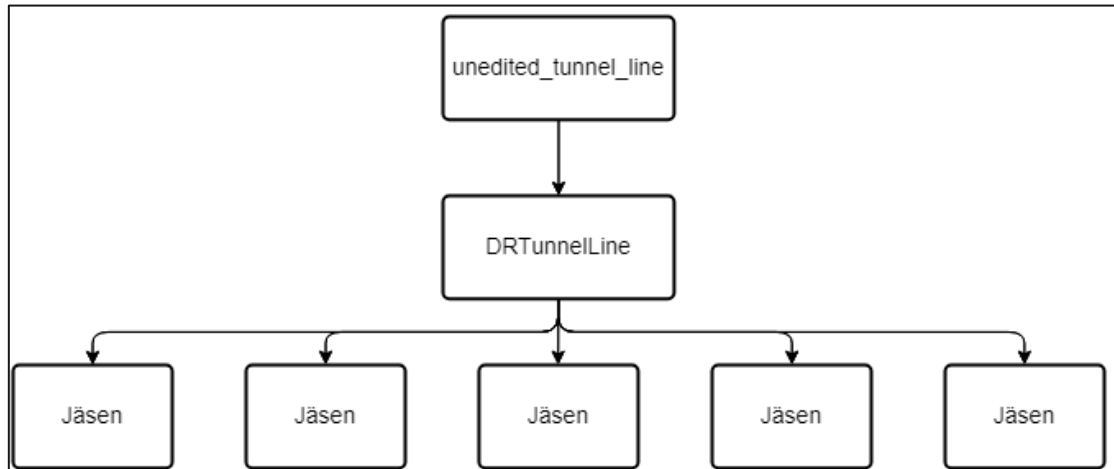
Funktio antaa paluuarvona XML-tiedoston sisällön struct-tietorakenteena (kuva 18: `unedited_tunnel_line`). Matlabin struct-tietorakennetta käsitellään pisteen avulla. Tietorakenne on tyypiltään puumainen. Sillä on yksi muuttuja, jonka alla sijaitsee eri tasoissa jäsen ja arvo pareja. Kuvassa 20 on esimerkki struct-tietorakenteesta. Esimerkissä on muuttuja `patient`, jonka alla on kolme jäsentä. Muuttujan jäsenillä on niille osoitetut arvot. Muuttujan jäseniä käsitellään pisteoperaattorin avulla. Esimerkin `name` jäsenen arvo saadaan komennolla `"patient.name"`. Tämä palauttaa arvon `"John Doe"`. Tässä esimerkissä tietorakenteessa on vain yksi alataso, mutta tasoille ei ole rajoituksia syvyys tai leveys suunnassa.



KUVA 20. Esimerkki Matlabin struct-tietorakenteesta. (Mathworks ei pvm.)

Funktion `xml2struct` palauttama struct-tietorakenne on kuvan 21 mukainen. Siinä on XML-tiedoston muodon takia jäsen `DRTunnelLine`, jolla itsellään ei ole arvoa. Sen sijaan halutut arvot sijaitsevat `DRTunnelLine`-jäsenen alapuolella sijaitsevilla jäsenillä. Komennolla `"unedited_tunnel_line = unedited_tunnel_line.DRTunnelLine"` poistetaan tämä ylimääräinen taso välistä

pois. Tason poistamisella helpotetaan tietorakenteen käsittelyä myöhemmissä vaiheissa.



KUVA 21. Funktion xml2struct palauttama tietorakenne.

Funktion parse_xml toiseen paluuarvoon tunnel_line_info tallennetaan yleistä tietoa tunnelilinjasta. Muuttuja tunnel_line_info on tyypiltään struct-tietorakenne. Tämä informaatio sisältää mm. projektikoordinaatiston akselien nimet, minimi- ja maksimipaalunumerot, tunnelilaserit jne. Kuvassa 22 näkyy miten muuttujaan tunnel_line_info tallennetaan halutut tiedot. Suurin osa tietojen tallentamisesta on varsin yksinkertaista. Esimerkiksi tunnelilinjan nimen tallennus onnistuu yksinkertaisesti komennolla "tunnel_line_info.plan_name = unedited_tunnel_line.IRu_colonu_PlanName.Text". Muuttujaan tunnel_line_info voidaan luoda uusia jäseniä käyttämällä pisteoperaattoria ja näille voidaan määrätä arvo käyttämällä yhtäsuuruusmerkkiä.

```

% General information about the tunnel line
tunnel_line_info.plan_name= unedited_tunnel_line.IRu_colonu_PlanName.Text;
tunnel_line_info.plan_id= unedited_tunnel_line.IRu_colonu_PlanId.Text;

if unedited_tunnel_line.CoordSystem.IRu_colonu_CsysType.Text == 'L'
    tunnel_line_info.coord_sys_type = -1; % Left handed
else
    tunnel_line_info.coord_sys_type = 1; % Right handed
end

tunnel_line_info.x_axis_name = unedited_tunnel_line.CoordSystem.IRu_colonu_AxisNames.IRu_colonu_X.Text;
tunnel_line_info.y_axis_name = unedited_tunnel_line.CoordSystem.IRu_colonu_AxisNames.IRu_colonu_Y.Text;
tunnel_line_info.z_axis_name = unedited_tunnel_line.CoordSystem.IRu_colonu_AxisNames.IRu_colonu_Z.Text;
  
```

KUVA 22. Yleistiedon parsiminen XML-tiedostosta

XML-tiedostossa on projektikoordinaatiston kätsisyys osoitettu kirjaimella. Myöhemmin ohjelmassa halutaan kuitenkin käsitellä koordinaatiston kätsisyyttä nume-

romuodossa. Tähän ratkaisuna käytetään kuvassa 22 näkyvää if-lausetta. If-lauseelle annetaan ehto, tässä tapauksessa tarkistetaan, onko CsysType-jäsenen sisältämä arvo kirjain L. Jos ehto on tosi, annetaan paluuarvon tunnel_line_info jäsenelle coord_sys_type arvo -1. Jos if-lauseen ehto on epätosi, annetaan jäsenelle arvo 1.

Osassa XML-tiedostoissa on tunnelilaserien tiedot. Tiedot halutaan tallentaa paluuarvoon tunnel_line_info tässä tapauksessa. Tietojen tallennus ei kuitenkaan onnistu aivan yhtä helposti kuin aikaisemmissa tapauksissa. Tätä vaikeuttaa se, että kaikissa XML-tiedostoissa ei tunnelilasereita ole. Jos ohjelmassa yritettäisiin käyttää struct-tietorakenteen jäseniä, joita ei ole olemassa, ohjelman suoritus keskeytyisi.

Tätä varten voidaan käyttää Matlabin funktiota isfield(S,field). Funktio tarvitsee kaksi argumenttia. Funktio antaa paluuarvon 1, jos jäsen sisältyy tietorakenteeseen ja 0 jos se ei sisälly. Funktion argumenteissa S:n sijalle laitetaan tutkittava struct-tietorakenne ja field:n paikalle tiedusteltava jäsen. Funktiossa parse_xml yhdistetään if-lause ja funktio isfield tunnelilaserien tarkistamiseksi seuraavasti, "if isfield(unedited_tunnel_line, "LaserTable") == 1". Ehdon täytyessä tallennetaan tarvittavat tiedot. Paluuarvon tunnel_line_info tiedot tallennetaan edellä esiteltyjä menetelmiä käyttäen. Funktio parse_xml tallentaa XML-tiedostosta vain ohjelmassa myöhemmin tarvittavat tiedot. Tulevaisuudessa mahdollisesti tulee tarvetta tiedolle, jota ei ole tallennettu. Tässä tapauksessa funktiota parse_xml on helppo muuttaa sisällyttämään halutut tiedot.

Funktion parse_xml toiseen paluuarvoon curve_table_array tallennetaan kaarretaulukko taulukko (array) muotoon. XML-tiedoston parsimisen jälkeen kaarretaulukko on tallennettu struct-tietorakenteessa solutaulukkona (cell array). Taulukossa jokaisessa solussa on struct-tietorakenne, joka sisältää kyseisen kaarretaulukon pisteen tiedot. Tiedoissa on xyz-koordinaatit, kaarretaulukkopisteen tunnistenumero, paalunumero, kallistuskulma ja vapaaehtoinen kommentti. Solutaulukossa on kaarretaulukon pisteiden verran soluja.

Solutaulukosta tiedot siirretään paluuarvoon curve_table_array. Tämä toteutetaan käyttämällä toistorakennetta, tarkemmin sanottuna for-silmukkaa. Kyseinen

silmukka toistaa sisältönsä määritellyn määrän verran. Kuvassa 23 on kaarretaulukon tallentamisen toteuttava ohjelmaosuus.

<code>curve_table_array = zeros(length(unedited_curve_table),5); % Preallocate the array</code>	1
<pre> % Add curve points to the data structures for i = 1:length(unedited_curve_table) temp_line_point = unedited_curve_table{i}; peg_number = str2double(temp_line_point.IRu_colonu_PegNumber.Text); peg.x_coord = str2double(temp_line_point.IRu_colonu_PointCoord.IRu_colonu_PointX.Text); peg.y_coord = str2double(temp_line_point.IRu_colonu_PointCoord.IRu_colonu_PointY.Text); peg.z_coord = str2double(temp_line_point.IRu_colonu_PointCoord.IRu_colonu_PointZ.Text); peg.camber_angle = str2double(temp_line_point.IRu_colonu_PointG.Text); curve_table_array(i,:) = [peg_number peg.x_coord peg.y_coord peg.z_coord peg.camber_angle]; end </pre>	2

KUVA 23. Kaarretaulukon tallentaminen

Toistorakenteen sisällä taulukon koon muuttaminen huonontaa ohjelman suorituskykyä. Tässä tapauksessa tiedetään ennen toistorakenteen käyttöä taulukon tuleva koko. Sen takia taulukko voidaan allokoida etukäteen. Allokointi näkyy kuvan 23 kohdassa 1. Tässä taulukkoon allokoidaan yhtä monta riviä kuin kaarretaulukossa on pisteitä. Jokaisella rivillä on 5 saraketta. Jokaiseen taulukon alkioon sijoitetaan luku 0 Matlabin funktiolla zeros. Allokoinnin seurauksena toistorakenteen sisällä taulukon kokoa ei kasvateta. Ainoastaan taulukon arvoja muutetaan, jonka seurauksena suorituskyky on parempi.

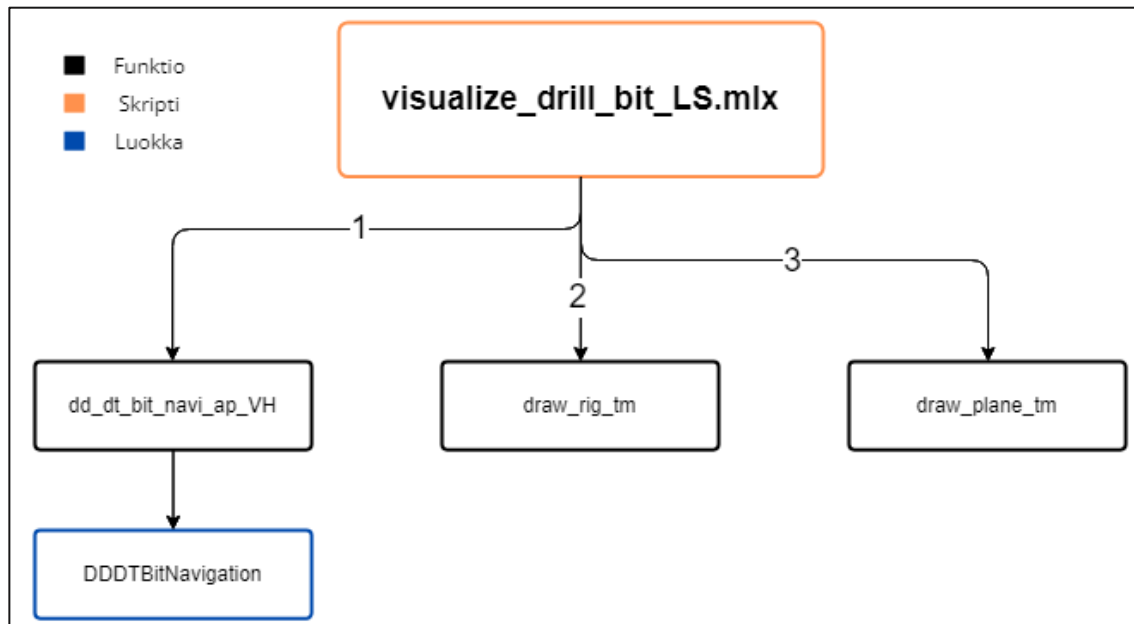
Kuvan 23 kohdassa 2 sijaitsee toistorakenne. Muuttuja `i` asetetaan arvoon 1 ja se kasvaa yhdellä aina kun toistorakenne on suoritettu kerran. Toistorakenteen sisällä ensin tallennetaan väliaikaisesti muuttujiin halutut tiedot, esim. komennolla `peg_number = str2double(temp_line_point.IRu_colonu_PegNumber.Text)` tallennetaan paalunumero väliaikaiseen muuttujaan. Väliaikainen tallennus on tehty, jotta koodin luettavuus on parempi. Toistorakenteen viimeisellä rivillä komennolla `curve_table_array(i,:) = [peg_number peg.x_coord peg.y_coord peg.z_coord peg.camber_angle]` tallennetaan paluuarvoon `curve_table_array` käsiteltävän kaarretaulukkopisteen tiedot. Komennossa muuttuja `i` määrää taulukon rivin sijainnin. Toistorakenteen päätyttyä on koko kaarretaulukko tallennettu muotoon, jota Sandvikin navigointifunktiot käyttävät.

5.4 Porakruununavigoinnin visualisointi

Porakruununavigoinnin visualisointi toteutetaan livescript-tiedostossa visualize_drill_bit. Ohjelma käynnistetään Matlabin yläreunasta live editor välilehden run painikkeella. Tällöin livescript-tiedoston ohjelma käynnistyy. Visualisoinnissa tärkeä ominaisuus on nähdä miten parametrien muutokset vaikuttavat visualisoinnin lopputulokseen. Tässä tulee yksi livescript-tiedoston vahvuus esiin. Ohjelmasta voidaan määritellä osuudet, jotka suoritetaan uudestaan syöttökentän arvon muuttuessa. Seurauksena ohjelmaa ei tarvitse käynnistää uudestaan eri arvoilla, vaan syötettyjen arvojen muuttuessa tulokset nähdään heti.

Ohjelman rakenne

Ohjelman toiminta perustuu aikaisemmin esitettyjen funktioiden ja navigointifunktioiden käyttämiseen. Näiden välillä livescript-tiedostossa toteutetaan tarvittavia operaatioita, jotta käytettävillä funktioilla saavutetaan haluttu lopputulos. Näitä ovat mm. parametrien muuttuessa edellisten visualisointien poisto, matriisin piirto näkyviin ja kuvaajaan liittyvät muutoseikat.



KUVA 24. Porakruununavigoinnin visualisoinnin ohjelmarakenne

Kuvassa 24 näkyy ohjelman rakenne. Ensin kutsutaan funktiota dd_dt_bit_navi_ap_VH, jolle annetaan argumentteina käyttäjän syöttämät parametrit. Tämä funktio toimii ns. välikätenä ohjelman ja luokan DDDTBitNavigation

välillä. Luokan sisällä on funktiot, jotka suorittavat varsinaisen navigaatiotapahtuman. Funktion `dd_dt_bit_navi_ap_VH` paluuarvona on muunnosmatriisi poralaitteen koordinaatistosta poraussuunnitelman koordinaatistoon. Matriisia käytetään piirtämään navigointitaso ja poralaite samaan koordinaatistoon.

Seuraavaksi ohjelma kutsuu funktiota `draw_rig_tm`. Porakruununavigoinnissa käytetään koordinaatistona poralaitteen koordinaatistoa, jolloin poralaitetta ei tarvitse muuntaa toiseen koordinaatistoon. Viimeiseksi kutsutaan funktiota `draw_plane_tm`, jolle argumenttina annetaan aikaisemmin saatu muunnosmatriisi.

Käyttöliittymä

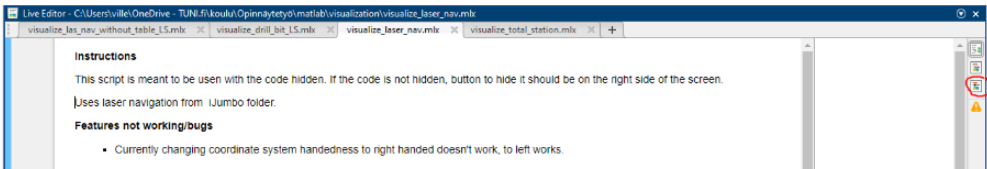
Lopputuloksena saadaan kuvassa 25 näkyvä käyttöliittymä. Tässä huomattavaa on, että käyttöliittymässä itsessään ei esiinny ohjelmakoodia, jota on esitelty aikaisemmissa funktioissa. Käyttöliittymässä esiintyy vain visualisoinnin käyttöön vaadittavat syöttökentät parametrien muuttamiseen. Käyttöliittymässä esiintyvät parametrit vaikuttavat navigoinnin lopputulokseen. Tuloksena saatu muunnosmatriisi luonnollisesti vaikuttaa visualisointiin, mutta opinnäytetyön kannalta ei ole oleellista tietää millä tavalla parametrien vaikutukset lasketaan.

Visualization of drill bit navigation.
Instructions etc. at the bottom.

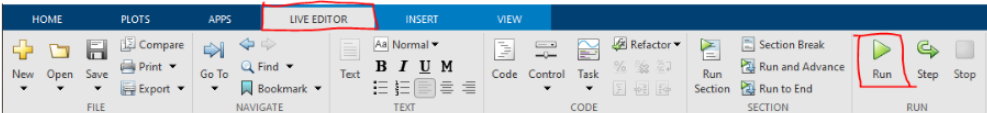
Rig lateral
Rig longitudinal
Nav point x (DP) Nav point z (DP)
Ramp angle
Nav point x (RIG) Nav point y (RIG) Nav point z (RIG)
Target angle U rig
 Filled navigation plane

Important things to know
This program does not contain any kind of user input validation. If the inputs are not proper, the program will stop and output an error.

Instructions
This script is meant to be used with the code hidden. If the code is not hidden, button to hide it should be on the right side of the screen (picture 1)



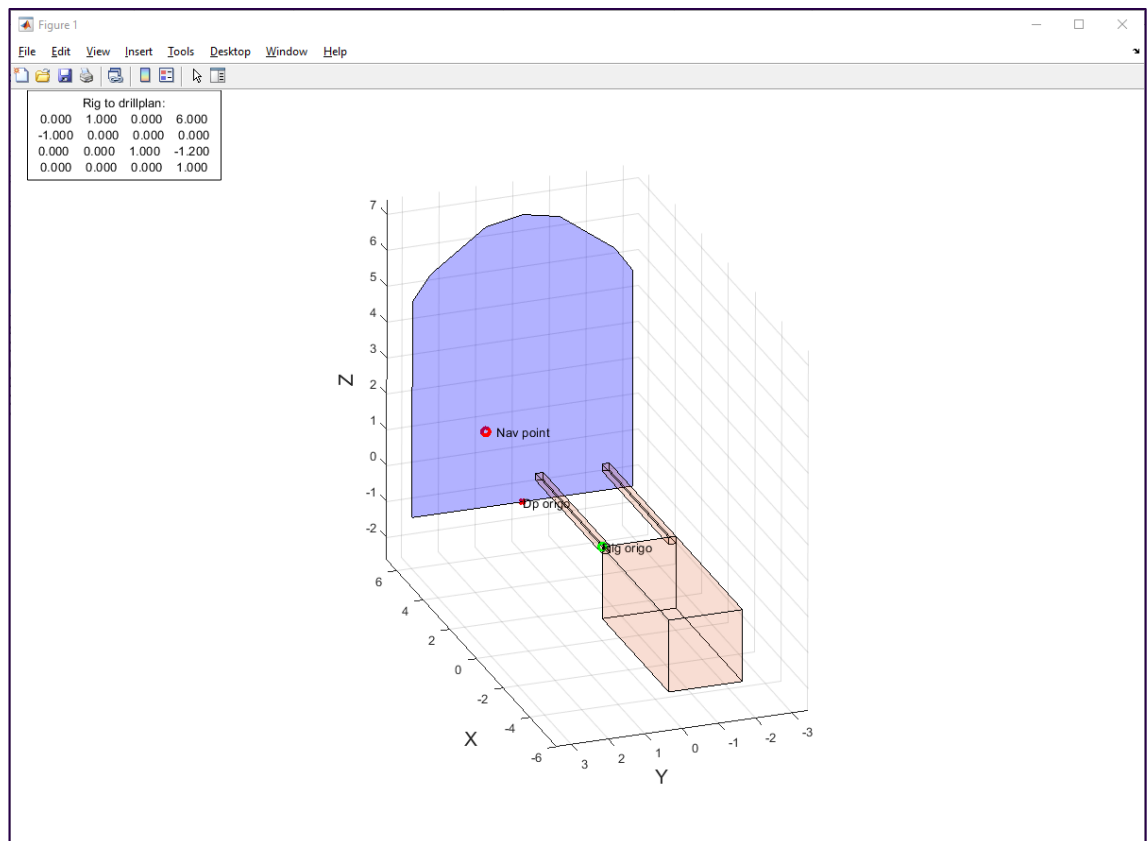
Picture 1. Hiding the code.



KUVA 25. Porakruununavigoinnin käyttöliittymä

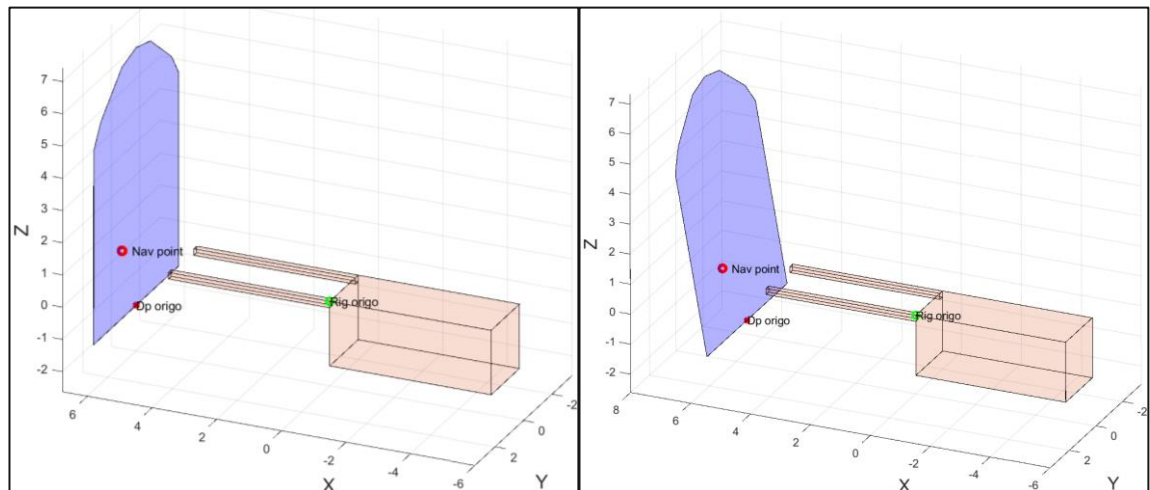
Visualisointinäkymä

Ohjelman käynnistämisen seurauksena aukeaa Matlabin kuvaajaikkuna (kuva 26). Kuvaaja on 3D-koordinaatisto, johon visualisointi toteutetaan. Livescriptissä tehdyt muutokset näkyvät kuvaajassa reaaliajassa. Kuvaajaa voidaan kiertää, zoomata ja siirtää käyttäjän toimesta. Kuvaajassa näkyy navigointipiste, jonka avulla muunnos koordinaatistojen välillä lasketaan, molempien koordinaatistojen origot ja käytettävä muunnosmatriisi.



KUVA 26. Porakruununavigoinnin näkymä

Kuvassa 27 nähdään miten visualisointi reagoi syöttöparametreja muuttaessa. Vasemmassa kuvaajassa parametri poralaitteen kallistus (kuva 25: rig longitudinal) on arvossa 0 astetta ja oikeassa kuvaajassa parametri saa arvon 10 astetta. Näin voidaan testata ja nähdä erilaisten parametriarvojen vaikutukset reaaliajassa. Tässä täytyy muistaa, että koordinaatisto on poralaitteen koordinaatisto, jolloin navigaatiotasoa reagoi muutoksiin poralaitteen pysyessä paikallaan.



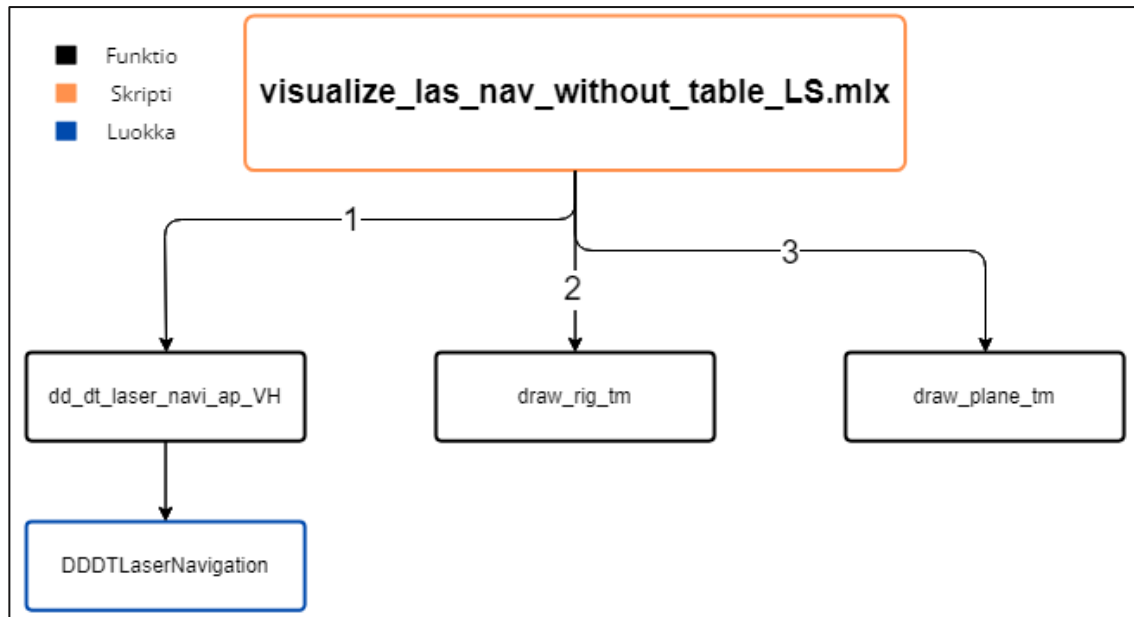
KUVA 27. Porakruununavigoinnin parametrimuutoksen vaikutus

5.5 Lasernavigoinnin visualisointi ilman kaarretaulukkoa

Lasernavigointi ilman kaarretaulukkoa ja porakruununavigointi ovat hyvin samankaltaisia visualisoinnin näkökulmasta. Molemmissa on vain kaksi koordinaatistoa ja tämän seurauksena vain yksi muunnosmatriisi.

Ohjelman rakenne

Kuvasta 28 nähdään ohjelman rakenteen olevan melkein identtinen porakruununavigoinnin visualisoinnin kanssa. Yksi eroavaisuuksista ohjelman rakenteessa on navigointifunktio ja luokka, joita käytetään. Kutsuttava navigointifunktio on nimeltään `dd_dt_laser_navi_ap_VH` ja se kutsuu luokkaa `DDDTLaserNavigation`. Funktio palauttaa muunnosmatriisin poralaitteen koordinaatistosta poraussuunnitelman koordinaatistoon.



KUVA 28. Lasernavigoinnin visualisoinnin ohjelmarakenne ilman kaarretaulukkoa

Käyttöliittymä

Ohjelman rakenteen ollessa samanlainen porakruununavigoinnin visualisoinnin kanssa käyttöliittymässä on isoja eroja. Lasernavigoinnin funktiot käyttävät huomattavasti enemmän parametreja, jolloin syötettävien arvojen määrä kasvaa käyttöliittymässä. Kuvassa 29 näkyy visualisoinnin käyttöliittymä. Opinnäytetyön kannalta ei ole oleellista ymmärtää kaikkien parametrien vaikutusta navigointitahtumaan.

Visualization of laser navigation without the use of curve table.
Instructions etc. at the bottom.

Navigation point input type

Nav point x (DP) Nav point z (DP)

Laser U Angle Laser V Angle

Lasera x (DP) Lasera y (DP) Lasera z (DP)

Laserb x (DP) Laserb y (DP) Laserb z (DP)

Rig Lateral Rig Longitudal

Nav point x (RIG) Nav point y (RIG) Nav point z (RIG)

Feed UG Feed VG

Feed U Rig Feed V Rig

Feed Roll Rig

Aim Offset Aim Offset Aim Offset

Filled navigation plane

KUVA 29. Lasernavigoinnin käyttöliittymä ilman kaarretaulukkoa

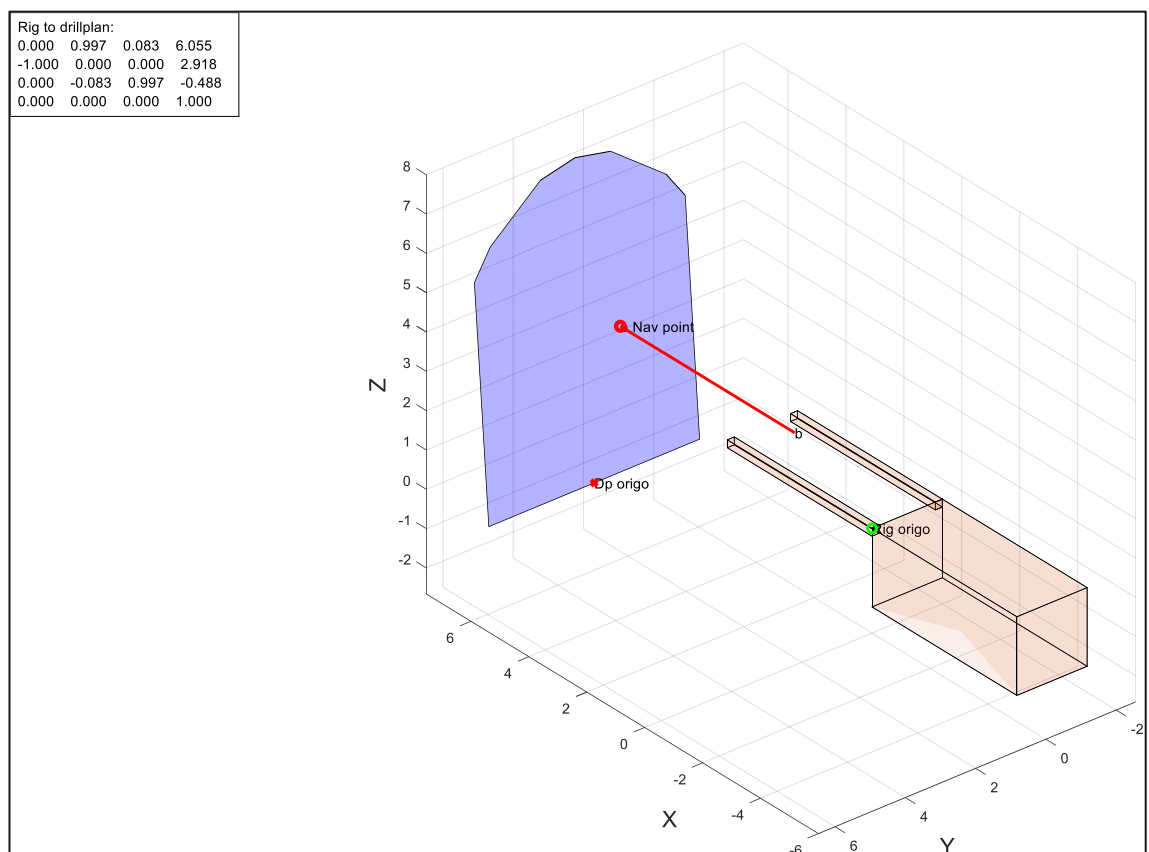
Lasernavigoinnissa on kaksi eri vaihtoehtoa navigointipisteen syötölle poraus suunnitelman koordinaatistossa. Voidaan syöttää pisteen koordinaatit poraus suunnitelman koordinaatistossa (kuva 29: Nav point x/z (DP)) ja laserin kulmat

(kuva 29: Laser U/V angle). Toisena vaihtoehtona on syöttää laserin 2 pistettä (kuva 29: Lasera/Laserb x/y/z (DP)), joiden kautta laser kulkee. Toisessa vaihtoehdossa navigaatiofunktio laskee laserin ja navigointitason leikkauspisteen sekä leikkauskulmat. Leikkauspistettä käytetään navigaatiopisteenä toisessa vaihtoehdossa.

Käyttöliittymässä on pudotusvalikko (kuva 29: Navigation point input type), josta valitaan käytettävä tapa. Sen jälkeen livescript-tiedostossa huomioidaan tämä valinta lasernavigaatiofunktioita kutsuessa.

Visualisointinäköymä

Lasernavigoinnin visualisoinnissa aukeaa identtinen näköymä porauskruununavigoinnin visualisointiin verratessa. Lasernavigoinnin visualisoinnissa on lisäksi piirretty käytetty laser, joka näkyy kuvassa 30.

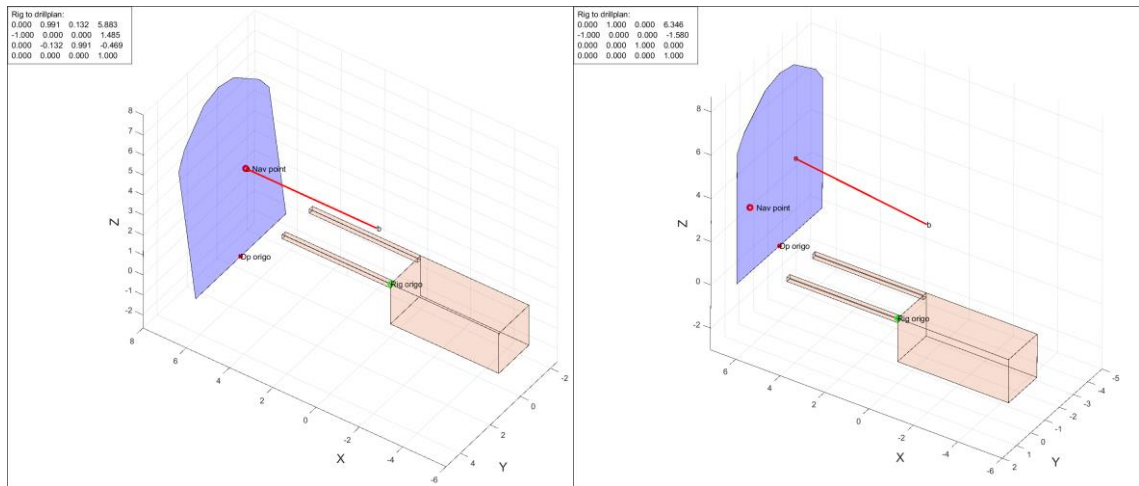


KUVA 30. Lasernavigoinnin näköymä ilman kaarretaulukkoa

Kuvasta 31 nähdään miten navigointipisteen syöttötapa vaikuttaa navigoinnin lopputulokseen. Vasemmassa käytetään laserin ja navigointitason leikkauspistettä navigointipisteenä, oikeassa navigointipiste ja laserkulmat syötetään käsin.

Oikeanpuolisessa visualisoinnissa laserin piirto on turha. Laserilla ei ole vaikutusta navigointiin ja sen pisteitä muuttamalla ei vaikuteta navigointitapahtumaan ollenkaan. Tulevaisuudessa visualisointiin voisi lisätä ominaisuuden, jolla laserin piirron saisi halutessaan pois.

Kuvasta 31 nähdään myös navigointipisteen syöttötävän vaikutus muunnosmatriiseihin. Ilman visualisointia navigointisyöttötävän erot nähtäisiin vain näissä muunnosmatriiseissa. Matriiseista on kuitenkin hankala tulkita parametrimuutoksien vaikutusta navigointitapahtumaan. Sen sijaan visualisoinnin tuloksista on hyvin helppo nähdä muutokset parametrien muuttuessa.



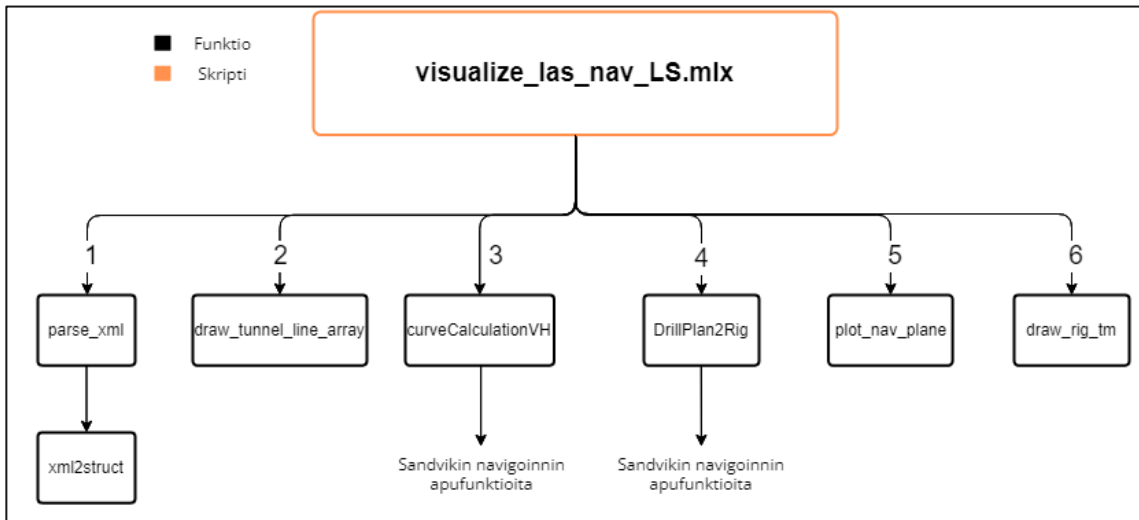
KUVA 31. Navigointipisteen syöttötävän ero

5.6 Lasernavigoinnin visualisointi kaarretaulukon kanssa

Lasernavigoinnin visualisointi kaarretaulukon kanssa on toteutettu edellisten tapaan Matlabin livescript-tiedostolla. Navigoinnissa on mukana tunnelilinja ja sen sisältämä kaarretaulukko. Poralaitteen koordinaatiston ja poraussuunnitelman koordinaatiston lisäksi mukana on nyt projektikoordinaatisto. Aiemmassa kahdessa tapauksessa visualisointi tapahtui poralaitteen koordinaatistossa. Nyt visualisointi tapahtuu projektin koordinaatistossa. Tämän seurauksena sekä navigointitaso että poralaite pitää muuntaa projektikoordinaatistoon.

Ohjelman rakenne

Lasernavigoinnin visualisoinnin ohjelmarakenne on selkeästi monimutkaisempi kuin aikaisemmissa tapauksissa. Kuvasta 32 nähdään kutsuttavien funktioiden määrän kasvaneen kaksinkertaiseksi. Tässä navigointimenetelmässä ei ole käytetty luokkia, vaan kaikki toiminnot on toteutettu funktioina.



KUVA 32. Lasernavigoinnin visualisoinnin ohjelmarakenne

Ensin kutsutaan funktiota `parse_xml`, jonka paluuarvoina saadaan kaarretaulukko ja tunnelilinjan tiedot, joita tarvitaan seuraavissa funktioissa. Funktiolla `draw_tunnel_line_array` piirretään kaarretaulukon pisteet kuvaajaan. Funktio `curveCalculationVH` palauttaa muunnosmatriisin projektikoordinaatistosta poraussuunnitelman koordinaatistoon. Funktio `DrillPlan2Rig` palauttaa muunnosmatriisin poraussuunnitelmasta poralaitteeseen. Näiden kahden muunnosmatriisin avulla pystytään sujuvasti liikkumaan kaikkien 3 koordinaatiston välillä. Funktiot `curveCalculationVH` ja `DrillPlan2Rig` käyttävät useita muita navigaatioon liittyviä alafunktioita navigoinnin toteuttamiseksi. Navigointien alafunktiot eivät ole oleellisia opinnäytetyön kannalta. Funktio `plot_nav_plane` piirtää navigointitason kuvaajaan ja `draw_rig_tm` piirtää poralaitteen kuvaajaan.

Käyttöliittymä

Kuvassa 33 näkyvä käyttöliittymä muistuttaa hyvin paljon aikaisempia. Se sisältää tarvittavat syötteen parametrien muunnoksiin. Lisänä on otettu käyttöön liu-

kusäätimet porauksen alkupisteen eli paalunumeron (kuva 33: start point) ja katkon pituuden säädölle (kuva 33: Depth). Arvot voi edelleen halutessaan syöttää kirjoittamalla.

Lisäksi navigointimenetelmässä on kaksi tapaa antaa tunnelilaserien sijainnit (kuva 33: Laser input type). Arvot voidaan lukea suoraan XML-tiedostosta tai antaa projektikoordinaatiston pisteinä. Jos syöttötapana on XML-tiedosto, mutta tiedosto ei sisällä tunnelilasereita ohjelma avaa kuvan 34 mukaisen virheviestin laserien puuttumisesta. Ohjelman suoritus keskeytyy kunnes käyttäjä painaa virheviestissä näkyvää OK painiketta. Opinnäytetyön kannalta ei ole oleellista ymmärtää kaikkien parametrien vaikutusta navigointitapahtumaan.

Visualization of laser navigation with curve table.
Instructions etc. at the bottom.

Laser input type

Lasera x (Project) Lasera y (Project) Lasera z (Project)

Laserb x (Project) Laserb y (Project) Laserb z (Project)

start point Depth

Filled navigation plane

TargetY (RIG)

TargetX (RIG)

TargetZ (RIG)

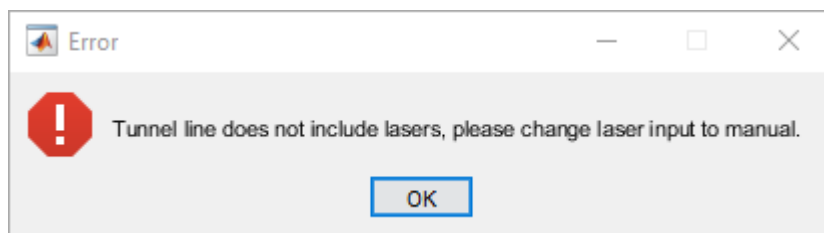
TargetU

TargetV

RigSwing

RigTilt

KUVA 33. Lasernavigoinnin visualisoinnin käyttöliittymä

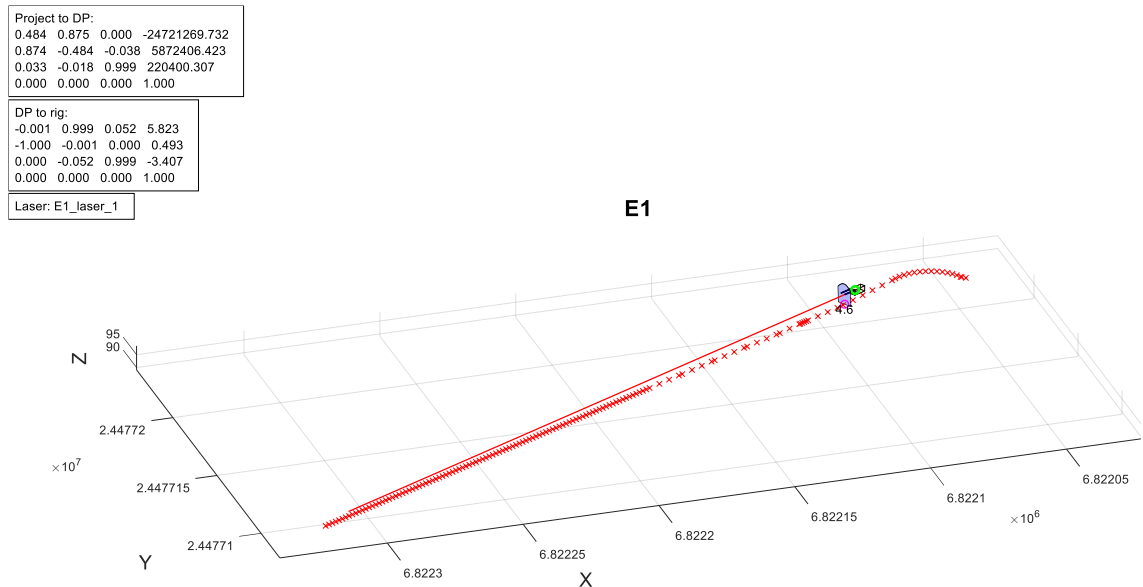


KUVA 34. Virheviesti laserien puutteesta

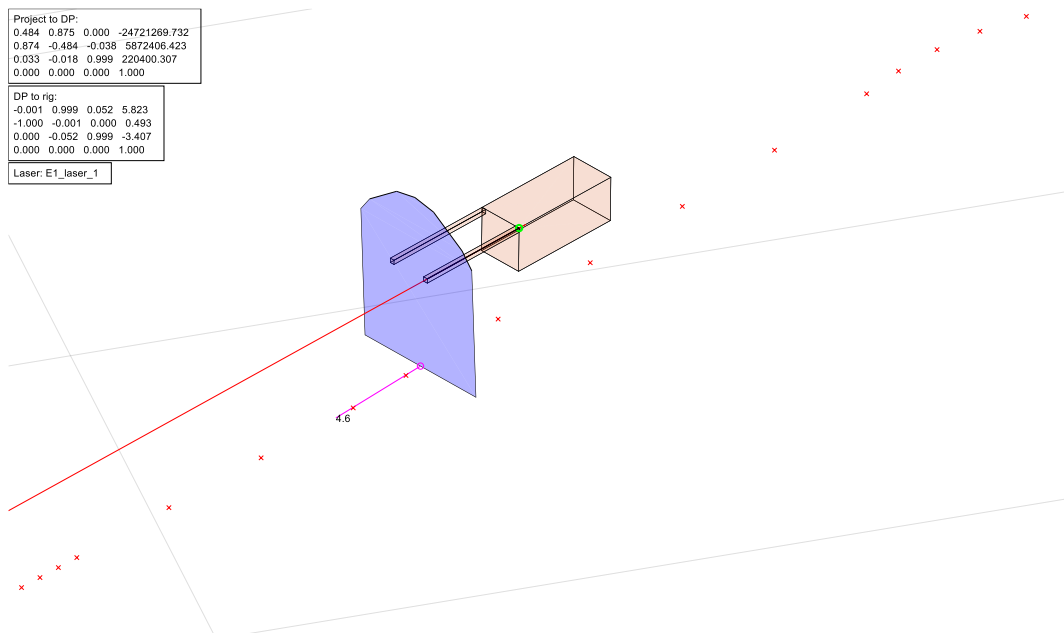
Visualisointinäköymä

Olettaen parametrien olevan sallituissa rajoissa tiedoston valitsemisen jälkeen aukeaa kuvan 35 näköinen kuvaaja. Tunnelilinjan pituudesta ja pisteiden määrästä johtuen yksityiskohdat eivät ole selkeitä. Kuvaajaa voidaan zoomata lähem-

mäs, jolloin saadaan kuvan 36 mukainen tilanne. Tästä kuvasta nähdään jo paremmin yksityiskohtia. Kuvaajassa esiintyy punaisina rasteina kaarretaulukon pisteet, navigointitaso, poralaite, käytetty tunnelilaseri, liilalla katko, katkon pituus numerona ja vihreällä ympyrällä poralaitteen origon sijainti. Kuvaajassa on myös esillä molemmat muunnosmatriisit ja tunnelilaserin nimi, jos laseri on valittu XML-tiedostosta.



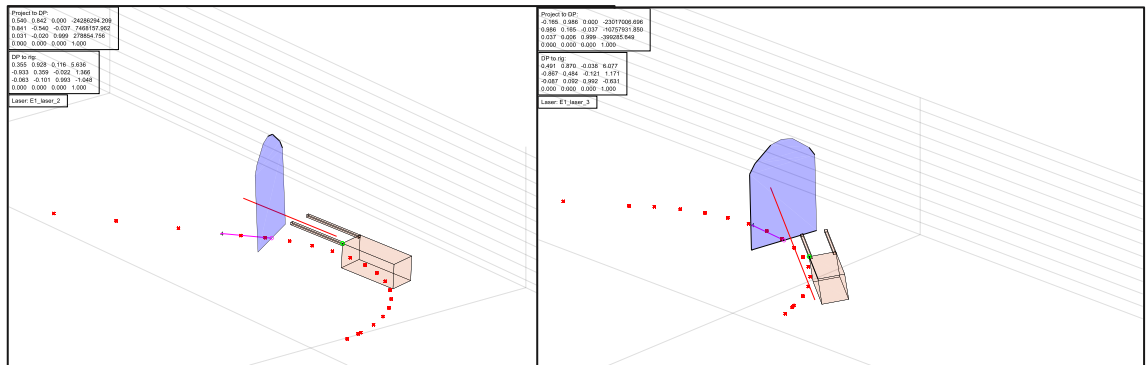
KUVA 35. Lasernavigoinnin näkymä



KUVA 36. Lasernavigoinnin näkymä zoomattuna

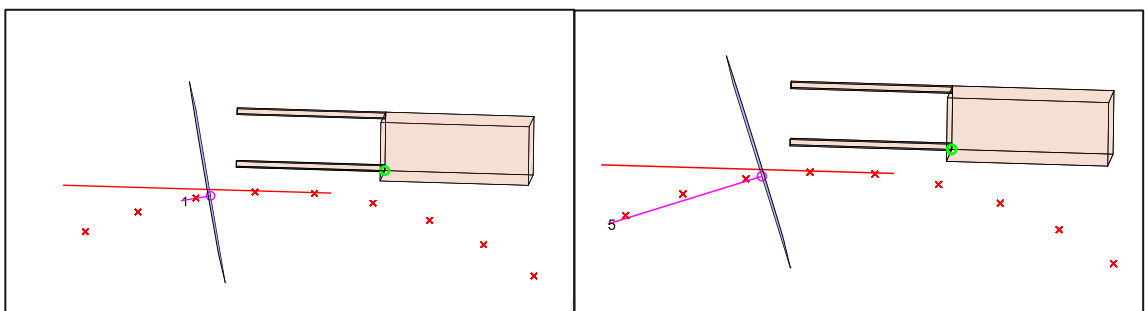
Esimerkissä käytetyllä tunnelilinjalta on kolme tunnelilaseria. Jokaiselle tunnelilaserille on määritelty paalunumeroväli, jolla kyseistä laseria käytetään. Visualisointiohjelma etsii oikean laserin annetulle paalunumerolle ja vaihtaa sen käytettäväksi laseriksi. Kuvasta 35 nähdään tunnelilinjan olevan pitkä suora, jolla on mutka toisessa päässä. Kuvassa poralaitte sijaitsee suoralla ja koko suoran ajan käytössä on kuvassakin näkyvä tunnelilaseri.

Kuvasta 37 nähdään mitä mutkan kohdalla tapahtuu. Käyttöliittymältä annetaan haluttu katkon aloituspiste (kuva 33: start point), jonka perusteella navigointitaso sijoitetaan. Tässä paalunumero on annettu mutkan kohdalta, jolloin tunnelilaseri vaihtuu pois kuvan 35 laserista. Kuvassa 37 vaihtuu myös tunnelilaseri kertaalleen mutkan aikana.



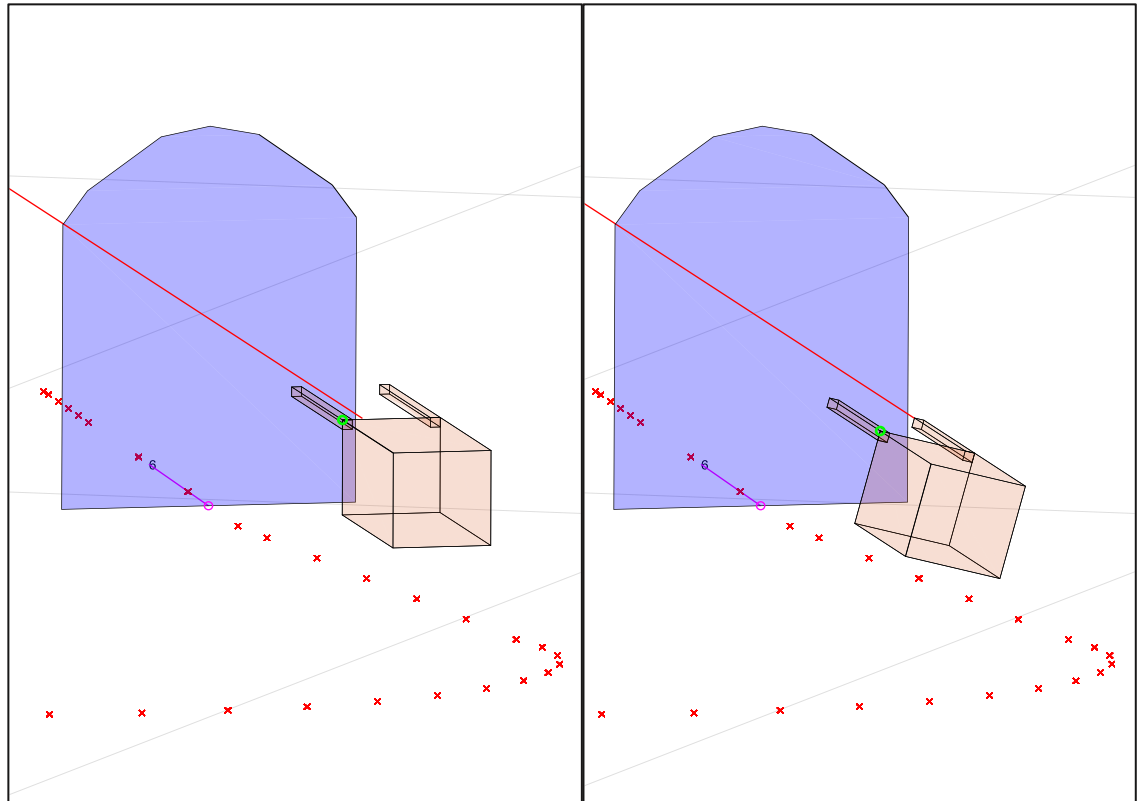
KUVA 37. Lasernavigointi mutkassa

Kuvasta 38 nähdään katkon pituuden (kuva 33: Depth) vaikutus navigointiin. Kuvasta ero on hankalampi hahmottaa, mutta kannattaa kiinnittää huomiota navigointitason ja poralaitteen puomien väliseen etäisyyteen. Työkalua käyttäessä erot havaitsee helpommin, koska parametria muuttaessa visualisointi muuttuu reaaliajassa. Katkon pituus vaikuttaa navigointitason suuntaan. Isommalla katkon pituudella navigointitason ns. oikaisee enemmän mutkan kautta.



KUVA 38. Katkon pituuden vaikutus navigointitasoon (ylhäältä)

Kuvasta 39 nähdään poralaitteen kallistusarvon (kuva 33: RigSwing) vaikutus. Oikeassa käyttötapauksessa poralaitteen eri kallistuskulmat luettaisiin suoraan poralaitteen antureilta, mutta tässä arvot syötetään itse. Kuvassa vasemmalla kallistuksen arvo on 0 astetta ja oikealla se on muutettu arvoon 15 astetta.



KUVA 39. Poralaitteen kallistusarvon vaikutus

5.7 Takymetrinavigoinnin visualisointi

Takymetrinavigoinnin visualisointi on samankaltainen edellisen lasernavigoinnin kanssa. Molemmissa käytetään kaarretaulukkoa ja visualisointi esitellään projektiikoordinaatistossa.

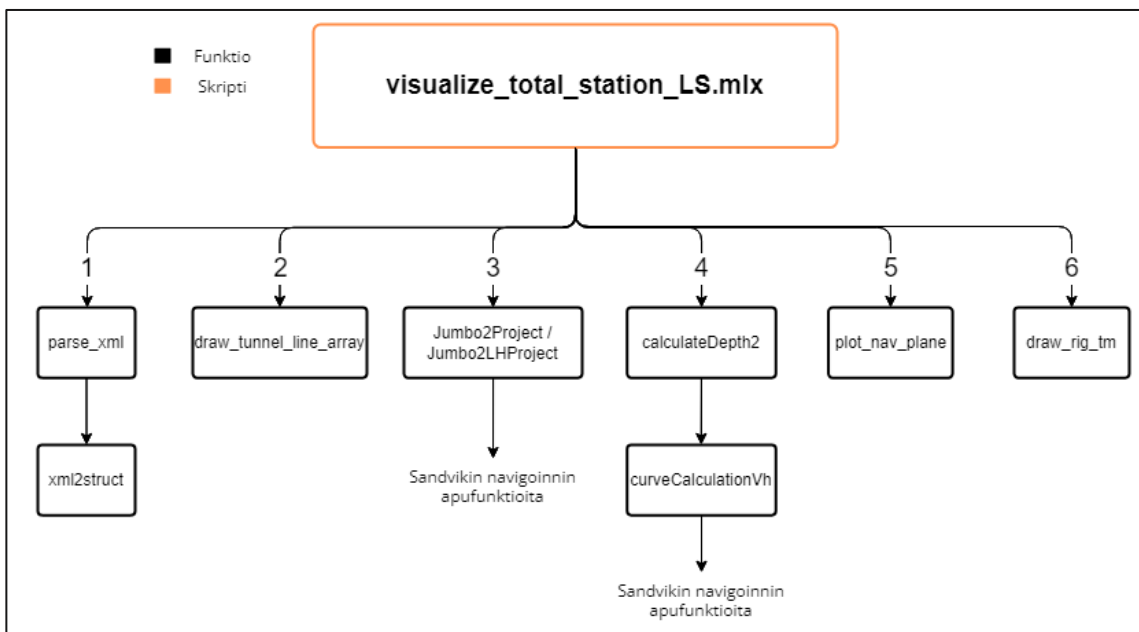
Ohjelman rakenne

Kuvassa 40 näkyvässä ohjelman rakenteessa on muutamia eroavaisuuksia verrattuna edelliseen lasernavigointiin. XML-tiedoston parsiminen (kuva 40: parse_xml) ja tunnelilinjan piirto (kuva 40: draw_tunnel_line_array) tehdään edelleen samalla tavalla. Ensimmäinen ero tulee poralaitteen sijainnin määrittelyssä. Lasernavigaatioissa määriteltiin poralaitteen sijainti

poraussuunnitelmaan nähden ja saatiin tuloksena muunnosmatriisi poraussuunnitelmasta poralaitteeseen.

Takymetrinavigoinnissa sen sijaan käytetään projektikoordinaatiston kätisyydestä riippuen funktiota Jumbo2Project (oikeankätinen) tai Jumbo2LHProject (vasemmankätinen). Funktio palauttaa muunnosmatriisin poralaitteesta projektikoordinaatistoon.

Seuraavaksi käytetään funktiota calculateDepth2. Takymetrinavigoinnissa ei anneta suoraan paalunumeroa kuten lasernavigoinnissa, vaan paalunumero lasketaan annetusta navigointipisteestä. Funktiolle annetaan navigointipiste projektikoordinaateissa. Funktio laskee navigointipisteestä paalunumeron ja kutsuu lasernavigoinnissakin käytettyä funktiota curveCalculationVh. Tuloksena saadaan muunnosmatriisi projektikoordinaatistosta poraussuunnitelman koordinaatistoon. Funktio plot_nav_plane piirtää navigointitason kuvaajaan ja draw_rig_tm piirtää poralaitteen kuvaajaan.



KUVA 40. Takymetrinavigoinnin visualisoinnin ohjelmarakenne

Käyttöliittymä

Takymetrinavigoinnin käyttöliittymä seuraa samaa periaatetta edellisten navigointimenetelmien käyttöliittymien kanssa. Tarvittavia parametreja varten käyttöliittymässä on syötekentät. Poiketen lasernavigoinnista paalunumeroa ei syötetä

käyttöliittymästä. Sen sijaan syötetään navigointipiste. Navigointipisteen voi syöttää poralaitteen koordinaateissa (kuva 41: Navigation point x/y/z (Rig)), jolloin se simuloi navigointipisteen osoittamista poralaitteen puomilla. Toinen vaihtoehto on antaa navigointipiste projektikoordinaatistossa (kuva 41: Navigation point x/y/z (Pro)), joka simuloi navigointipisteen mittaamista takymetrilla. Navigointipisteen syöttötavan valinta on toteutettu pudotusvalikolla (kuva 41: In which coordinates do you want to give navigation point?). Jos navigointipiste osoitetaan puomilla, tarvitsee se muuttaa projektikoordinaatistoon ennen käyttöä. Tässä tapauksessa tarvitaan muunnosmatriisi poralaitteen koordinaateista projektin koordinaatteihin ennen funktion calculateDepth2 kutsumista. Jos navigointipiste annettaisiin aina projektikoordinaateissa, voitaisiin kuvassa 40 näkyvien vaiheiden kolme ja neljä järjestystä vaihtaa.

Visualization of total station navigation with curve table.
Instructions etc. at the bottom.

Depth

Filled navigation plane

Front prism x (Rig) Front prism y (Rig) 0
Rear prism x (Rig) Rear prism y (Rig) Rear prism z (Rig)

Front prism x (Project) Front prism y (Project) Front prism z (Project)
Rear prism x (Project) Rear prism y (Project) Rear prism z (Project)

TargetV
RigSwing

In which coordinates do you want to give navigation point?

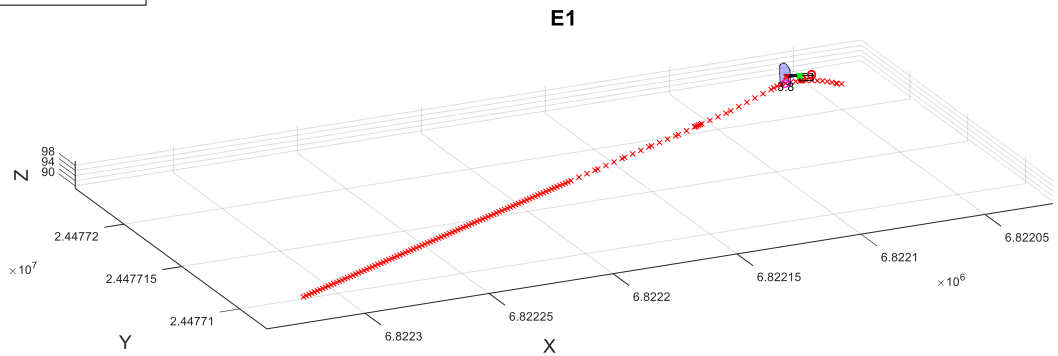
Navigation point x (Rig) Navigation point y (Rig) Navigation point z (Rig)
Navigation point x (Pro) Navigation point y (Pro) Navigation point z (Pro)

KUVA 41. Takymetrinavigoinnin visualisoinnin käyttöliittymä

Visualisointinäköymä

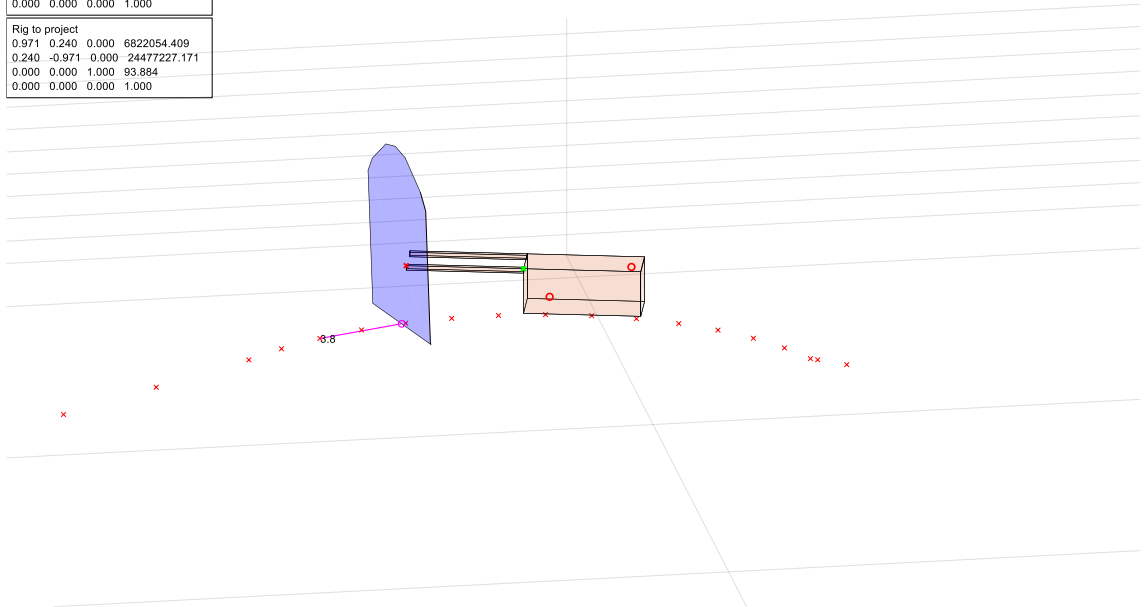
Kuvassa 42 näkyy takymetrinavigoinnin visualisoinnin näköymä. Esimerkissä on käytössä sama tunnelilinja kuin edellisessä lasernavigoinnissa. Näköymä onkin nopeasti katsottuna melkein identtinen. Zoomatusta kuvasta 43 nähdään kuitenkin tarkemmin kahden visualisoinnin erot. Takymetrinavigoinnin visualisaatiossa ei piirretä laseria, jota ei käytetä. Sen sijaan piirretään kaksi navigointipristmaa näkyviin punaisina ympyröinä. Lisäksi navigointipiste on merkattu punaisella rastilla navigointitasossa.

Project to DP:				
0.256	0.967	0.000	-25408089.374	
0.966	-0.256	-0.037	-322627.282	
0.036	-0.009	0.999	-11979.920	
0.000	0.000	0.000	1.000	
Rig to project				
0.971	0.240	0.000	6822054.409	
0.240	-0.971	0.000	24477227.171	
0.000	0.000	1.000	93.884	
0.000	0.000	0.000	1.000	



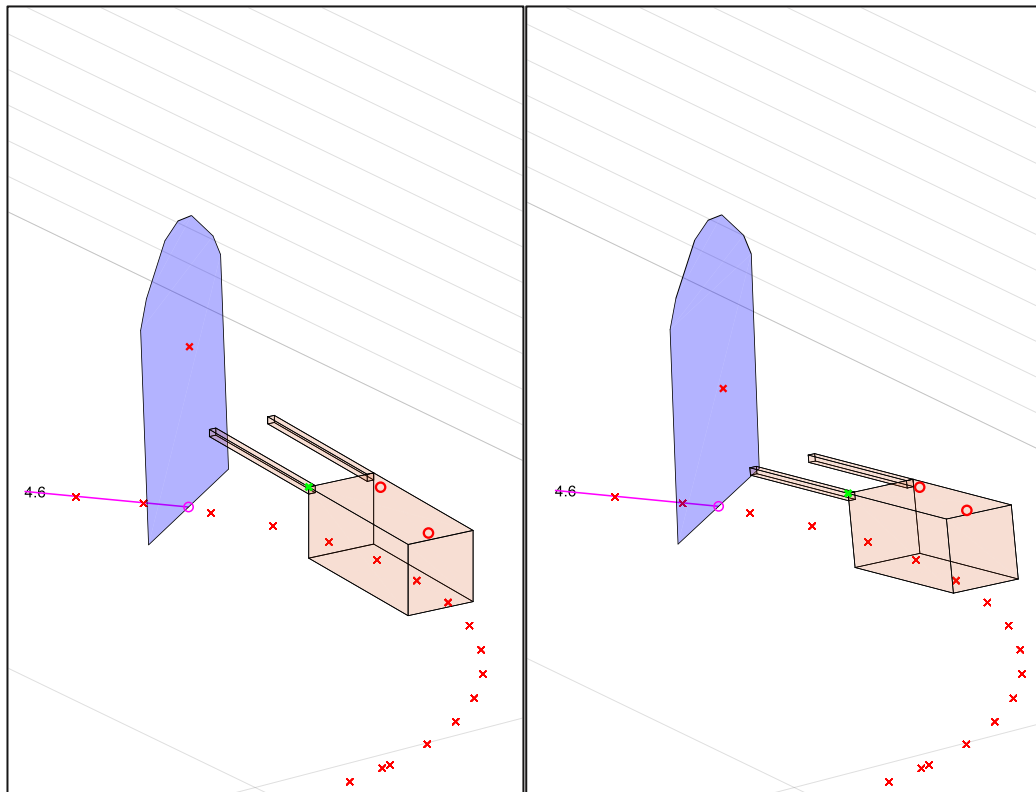
KUVA 42. Takymetrinavigoinnin näkymä

Project to DP:				
0.256	0.967	0.000	-25408089.374	
0.966	-0.256	-0.037	-322627.282	
0.036	-0.009	0.999	-11979.920	
0.000	0.000	0.000	1.000	
Rig to project				
0.971	0.240	0.000	6822054.409	
0.240	-0.971	0.000	24477227.171	
0.000	0.000	1.000	93.884	
0.000	0.000	0.000	1.000	



KUVA 43. Takymetrinavigoinnin näkymä zoomattuna

Kuvassa 44 näkyy poralaitteen kulman vaikutus visualisointiin. Vasemmassa tapauksessa poralaitteen kulma-arvo (kuva 41: RigSwing) on 0 astetta ja oikealla se on muutettu arvoon -10 astetta. Seurauksena huomataan myös navigointipisteen muuttuvan.



KUVA 44. Poralaitteen kulma-arvon vaikutus visualisointiin

6 POHDINTA

Opinnäytetyön tavoitteena oli maanalaisten poralaitteiden käyttämien navigointimenetelmien visualisointityökalu yritykselle Sandvik Mining and Construction Oy. Työn tavoitteena oli luoda visualisointityökalu neljään yrityksessä käytettyyn maanalaiseen navigointimenetelmään.

Sandvikilla oli olemassa navigointiin käytettävät Matlab-koodit, joiden tuloksia työkalulla täytyi pystyä visualisoimaan. Työkalua oli tarkoitus käyttää mainittujen koodien jatkokehitykseen. Erityisesti työkalulla haluttiin toteuttaa koodien laadunvarmistus ennen koodien ja algoritmien pääsyä tuotantoon. Työkalun kohderyhmänä oli siis ohjelmistokehittäjät, jotka työskentelevät navigointimenetelmien kanssa.

Voitiin olettaa jatkokäyttäjän tuntevan työkalun kehitykseen käytetyn ympäristön. Tämä helpotti visualisoinnin suunnittelua. Opinnäytetyössä pystyttiin keskittymään pelkästään visualisoinnin tuottamiseen. Jos navigointimenetelmät ja työympäristö olisivat tuntemattomia kohderyhmälle, olisi tämä täytynyt ottaa huomioon työkalua suunnitellessa. Tämä olisi tarkoittanut ylimääräistä ohjeistusta ja erityisesti käyttäjän syötteen tarkistusta.

Ajatuksena oli luoda yksi työkalu, josta pystyttäisiin valitsemaan käytetty navigointimenetelmä ja visualisoimaan se. Tämä ajatus kuitenkin jäi taka-alalle hyvin nopeasti. Tästä ei olisi saavutettu suurta hyötyä, mutta se olisi monimutkaistanut ohjelmointia. Sen sijaan luotiin jokaiselle navigointimenetelmälle oma työkalunsa. Työkalut käyttävät silti yhteisiä funktioita ja toimintoja.

Opinnäytetyön alussa lähdettiin luomaan työkalua Matlab skripti-pohjaisena. Tämä toimi hyvin alussa. Työkalun käytön kannalta oli tärkeää kyetä muuttamaan navigointimenetelmien parametreja kesken ohjelman suorituksen. Tutustuttiin erilaisiin keinoihin saavuttaa tämä Matlabin avulla, mutta tähän kului kohtuuttomasti aikaa. Tuloksena saatiin kuitenkin ensimmäinen versio, jossa parametreja kykeni muuttamaan kesken ohjelman suorituksen. Tämä oli kankea käyttää ja eikä sillä saavutettu kaikkia haluttuja toimintoja.

Ratkaisuna luovuttiin skriptistä ja siirryttiin käyttämään Matlabin livescript-tiedostoja. Vaihdoilla saavutettiin samat asiat kuin aikaisemmin käytetyllä skriptillä ja vielä huomattavasti helpommin. Lisäksi livescriptistä löydettiin opinnäytetyön kannalta hyödyllisiä ominaisuuksia. Tiedostoon voidaan kirjoittamaan ohjelmakoodia ja normaalia tekstiä sekaisin. Seurauksena työkaluun itsessään kyettiin kirjoittamaan mm. käyttöohjeet kuvien kanssa ja muu mahdollinen haluttu informaatio.

Opinnäytetyön kannalta livescriptien suurin ominaisuus löytyi kuitenkin sen sisään rakennetuista ohjauselementeistä. Näihin sisältyy mm. syöttökenttiä, liukuohjaimia ja pudotusvalikoita. Näissä ohjaimissa on toiminto, jolla pystytään ohjaimen arvon muuttuessa suorittamaan haluttu ohjelmanosuus uudestaan. Tällä pystyttiin saavuttamaan helposti työkalussa haluttu parametrien muutos kesken ohjelman suorituksen.

Opinnäytetyön tuloksena luotiin neljä Matlab livescript-tiedostoa, yksi kutakin navigointimenetelmää varten. Tiedosto visualisoi tulokset erilliseen 3D-kuvaajaikkunaan. Kuvaajaa voidaan zoomaata, pyörittää ja liikuttaa. Lisäksi navigaation parametreja voi muuttaa kesken ohjelman suorituksen ja visualisointi reagoi tähän reaaliajassa.

Opinnäytetyön tulos täyttää asetetut tavoitteet, mutta se ei ole täydellinen. Ensinnäkin työkalussa on hyvin vähän syötteen tarkistusta. Tämän seurauksena käytetyt navigaatiofunktiot voivat antaa virheilmoituksen ja lopettaa suorituksen kesken. Virheilmoitukset vaihtelevat laajasti ja ne voivat olla hankalasti tulkittavia. Oletuksena kuitenkin oli, että työkalua käyttäisivät asian hyvin tuntevat ihmiset, jolloin todettiin syötteen tarkastamisen olevan vaihtoehtoista tässä vaiheessa. Tämä on kuitenkin yksi mahdollinen kehityksen kohde tulevaisuudessa.

Visualisoinnissa poralaitetta esittää todellisuudesta paljon yksinkertaistettu malli. Tämän voisi halutessaan vaihtaa 3D-malliin oikeasta poralaitteesta. Muutos toisi visualisointiin tiettyä ammattimaisuutta ja olisi tietty huomattavasti lähempänä oikeaa maailmaa.

Opinnäytetyön tuloksena tuotettu työkalu on nyt Sandvikin ohjelmistokehittäjien käytössä. Kuten aikaisemmin mainittiin työkalun käyttäjät osaavat käyttää myös Matlabia. Heillä on hyvät lähtökohdat ylläpitää ja kehittää työkalua. Viimeisenä kehityskohteenä mainitaan työkalun suorituskyvyn parantaminen. Melko varmasti kokeneempi Matlab-käyttäjä kykenisi osan toiminnoista toteuttamaan tehokkaammin. Toisaalta paremmasta suorituskyvystä ei välttämättä saataisi suurta hyötyä työkalun käyttöympäristössä.

LÄHTEET

Boyd, S. ja Lieven V. 2018. Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares. Cambridge: Cambridge University Press.

Carter, R. 2019. Interoperability: Essential for Building the Mine of the Future. Engineering and Mining Journal 220 (10): 46-48, 50.

Gilat, A. 2014. Matlab: An Introduction with applications. 5. painos. New York: Wiley.

Kennie, T. ja Petrie, G. 2014. Engineering Surveying Technology. 1. painos. Boca Raton, Yhdysvallat: CRC Press.

Maidl, B. Markus T. ja Ulrich M. 2013. Handbook of Tunnel Engineering I: Structures and Methods. Berliini, Saksa: Ernst & Sohn.

Maidl, B. Markus T. ja Ulrich M. 2014. Handbook of tunnel engineering II: basics and additional services for design and construction. Berliini, Saksa: Ernst & Sohn.

McBain, J. ja Markus T. 2012. Software architecture for condition monitoring of mobile underground mining machinery: A framework extensible to intelligent signal processing and analysis. 2012 IEEE Conference on Prognostics and Health Management, 1–12.

Mathworks. ei pvm. Enter statements in Command Window - Matlab & Simulink - MathWorks Nordic. Viitattu 17. maaliskuuta 2022. https://se.mathworks.com/help/matlab/matlab_env/enter-statements-in-command-window.html.

Mathworks. ei pvm. Structure Arrays - Matlab & Simulink - MathWorks Nordic. Viitattu 5. maaliskuuta 2022. https://se.mathworks.com/help/matlab/matlab_prog/create-a-structure-array.html.

O'reilly, J. ja David, M. 2018. A drill rig positioning and drill rod alignment system. Patentti AU2016308671 (A1).

Sandvik AB. ei pvm. Business Areas. Sandvik Group. Viitattu 13. helmikuuta 2022a. <https://www.home.sandvik/en/about-us/business-areas/>.

Sandvik AB. ei pvm. Income Statement. Viitattu 13. helmikuuta 2022b. <https://www.home.sandvik/en/investors/sandvik-in-figures/>.

Sandvik Mining and Construction. ei pvm. Functional Specification - Navigation. Ei julkinen.

Sandvik Mining and Construction Oy. 2015. Understanding underground brochure.

Sandvik Mining and Rock Technology. ei pvm. Tietoja meistä. Viitattu 13. helmikuuta 2022. <https://www.rocktechnology.sandvik/fi/tietoja-meist%C3%A4/>.

Sandvik Mining and Rock Technology. ei pvm. Test Mine. Viitattu 4. maaliskuuta 2022b. <https://www.rocktechnology.sandvik/en/innovation-in-mining/test-mine/>.

Tamrock Oy (Oy Tampella Ab) — Osakehistoria. ei pvm. Viitattu 4. maaliskuuta 2022. <https://www.porssitieto.fi/poistuneet/tamrock.shtml>.

Turk, I. 2019. Practical Matlab: With Modeling, Simulation, and Processing Projects. Berkeley, Yhdysvallat: Apress L. P.

Ware, C. 2012. Information Visualization: Perception for Design. Boston, Yhdysvallat: Morgan Kaufmann.

Zahneisen, B. ja Ernst, T. 2016. Homogeneous Coordinates in Motion Correction. Magnetic Resonance in Medicine 75 (1): 274–79.

Zou, D. 2017. Theory and Technology of Rock Excavation for Civil Engineering, Singapore: Metallurgical Industry Press.