A scenic landscape photograph serves as the background for the entire page. In the foreground, a small, weathered stone cairn sits on a rocky outcrop. Beyond it, a range of mountains is visible under a clear blue sky.

A Python Implementation of the Direct Strapdown Method for Airborne Gravimetry

Synthesis project by:
Christian Solgaard
Spring 2023

A Python Implementation of the Direct Strapdown Method for Airborne Gravimetry

Author

Christian Solgaard - s183821

Msc in Earth and Space Physics and Engineering
Spring 2023

Supervisor:

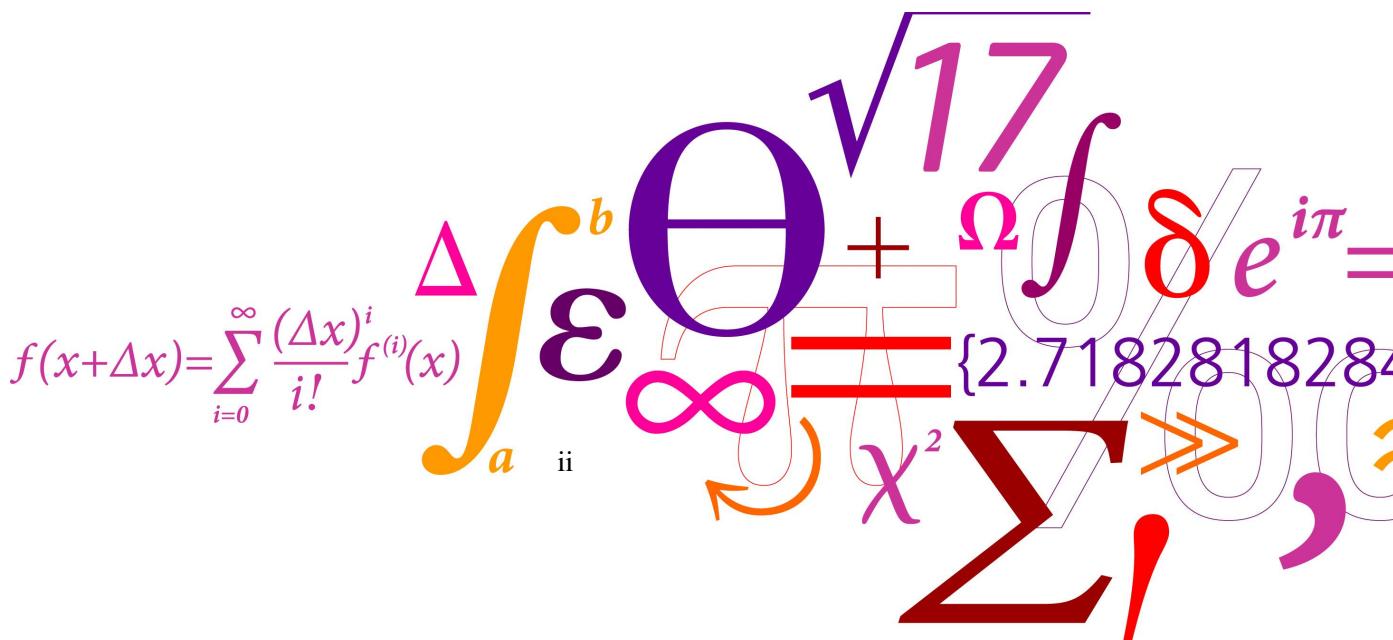
Tim Enzlberger Jensen
Geodesy and Earth Observation, Researcher
DTU Space

Abstract: In this study, an implementation of the direct strapdown method, for computing gravity disturbance, based on GNSS position and specific force measurements from an IMU is developed and written in Python. A pipeline from data-loading subroutines to the specific computations routines was written, such that the full analysis can be performed using the package. As a validation of the algorithm, two case studies were undertaken, these surveys were conducted in the same area over Denmark. The result between two waypoints (forward and backwards) was compared and analyzed. The computed gravity disturbance was compared to results based upon the indirect method. Through the validation process it was found that the implemented method, was unstable. Providing some results that with a high frequency diverted from the long wavelength tendency. Here a correlation with the physical movement of the airplane was found via an analysis based upon the attitude controls.¹

¹Frontpage Picture: Christian Solgaard, Sisimiut 2022.

Contents

1	Introduction	1
2	Description of the Problem	2
2.1	The Direct Strapdown Method	2
2.2	Data Description	3
3	Methodology	5
3.1	Computation of GNSS position derived acceleration	5
3.2	Conversion from body b -frame to navigation n -frame	5
3.3	Computation of the Eötvös correction (δg_{eot}^n)	7
3.4	Filtration of Signals	8
3.5	Computation Gravity Disturbance δg^n and normal gravity γ^n	8
3.6	Bias and Drift Determination	8
4	Case Study	10
4.1	DK2022 - FlightID 285	11
4.2	Roskilde16 - FlightID 116	14
5	Discussion	15
5.1	Attitude analysis	15
6	Conclusion	18
6.1	Further Work	18
A	Roskilde16 - FlightID 116 Results	21
B	Software Description	23
B.1	Direct Strapdown	23
B.2	Data Loader	30



1 | Introduction

Strapdown gravimetry is a term covering the geodetic technique, where an initial navigation/measurement unit (INS/IMU) is used in combination with a GNSS solution for position, to determine gravity estimates, when performing airborne surveying. [Becker, 2016] The Strapdown gravimetry method differs from the more traditional way of performing airborne gravimetry surveys, as no horizontally stabilized platform is needed for the stabilization of the gravimeter. Instead the IMU is fastened directly to the airplane or helicopter. This means a reduction in volume and weight of the system, giving new opportunities for performing surveys [Vyazmin et al., 2021, Enzlberger, 2018, Glennie et al., 2000]. The data processing of strapdown gravimetry, can be divided into two methodologies. The indirect and the direct method.

The main difference between the two methods, is that in the direct method, the gravity estimates is computed via GNSS derived accelerations and the difference to the specific force measurements from the IMU. In the indirect method, all IMU and GNSS measurements is integrated using a single Kalman filter, afterwards a separation of the specific force measurements and the gravity can be done using the GNSS position. [Jekeli and Garcia, 1997, Johann et al., 2019].

The aim for this project is an implementation of the direct method. The written software is based upon the algorithm described by [Johann et al., 2019]. In section 2 the algorithm and used data is described. Each of the individual sub-routines is described in theory in section 3 and the practical python implementation in Appendix B. For a validation of the computed gravity disturbance a comparison is made with the results of the same case studies computed via the indirect method, which is provided by Tim E. Jensen from DTU SPACE. [Jensen and Forsberg, 2018]

2 | Description of the Problem

The scope of work can be defined in two directions. Firstly the implementation of the direct strapdown method in Python. Secondly an implementation of a routine in Python to read and convert the hexadecimal encoded raw IMU data. This report is focused around the implementation of the direct strapdown method. Therefore, for more information about the IMU loading routine, see Appendix B.2.1.

2.1 The Direct Strapdown Method

As described in the introduction section 1, the main goal for this project is an implementation of the direct strapdown method. This essentially means that the gravity disturbance defined by equation 1, is computed via the method outlined in Figure 2.1

$$\delta g^n = \ddot{r}^n - f^n + (2\Omega_{ie}^n + \Omega_{en}^n) \cdot \dot{r}^n - \gamma^n \quad (1)$$

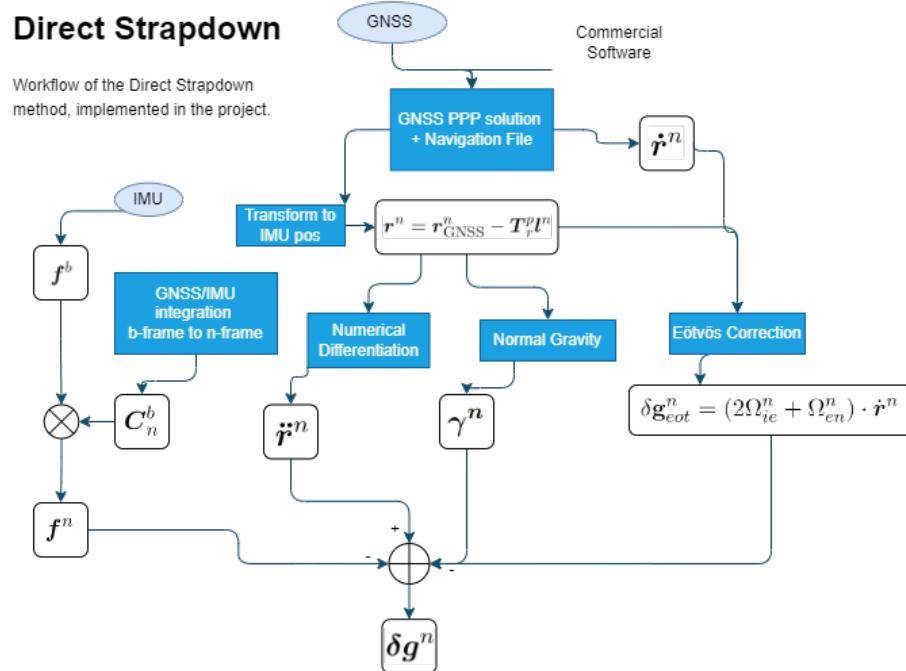


Figure 2.1: Flow chart of the direct method, implemented in the project. The structure is based upon the implementation done by [Johann et al., 2019]

The direct strapdown method, which is outlined in Figure 2.1 can be described as a "cascade" method. [Becker, 2016]. This means that the final product is computed via a series of smaller subroutines which are implemented to calculate the different quantities of equation 1. The method is based upon a direct

comparison of the specific force measured by the IMU and the GNSS derived accelerations. [Johann et al., 2019]. For this scope of work, a commercial software is used for the GNSS/INS integration to derive the navigation file and the Precise Point Position (PPP) solution. The quantities needed to derive the gravity disturbances, are GNSS derived acceleration \ddot{r}^n , a conversion of the specific force from body frame to navigation frame $f^b \rightarrow f^n$, a computation of the Eötvös correction due to the relative movement of the airplane and the Earth, finally a computation of the normal gravity in the navigation frame. A further description for the individual subroutines is outlined in section 3.

2.2 Data Description

As described in the problem description, two different types of datasets is needed to perform the analysis. The Output from the IMAR unit, here referenced as IMU data and the output from the GNSS receiver. For this scope of work the GNSS precise point precision solution (PPP) and navigation file, is computed using commercial software.

In Figure 2.2 and 2.3 the GNSS- and IMU-data for one of the case studies DK2022 of FlightID 285 can be seen. For the example of DK2022 the GNSS solution is obtained at 1Hz.

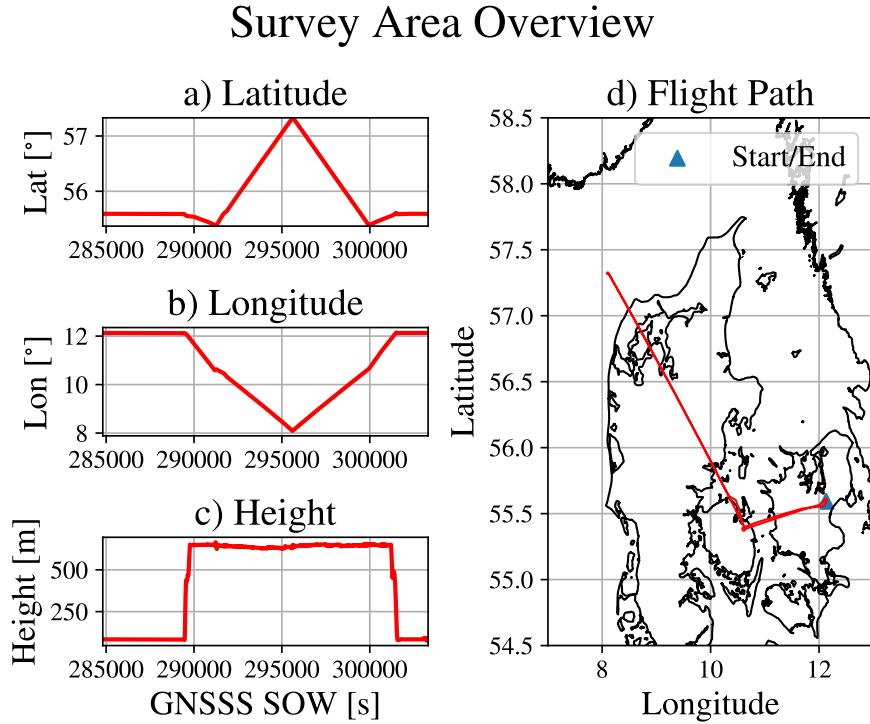


Figure 2.2: FlightID 285 GNSS PPP solution information: a) The latitude in degrees [$^\circ$] vs. Time epoch [s]. b) The longitude in degrees [$^\circ$] vs. Time epoch [s]. c) The height in meters [m] above the reference ellipsoid vs. Time epoch [s]. d) An overview of the survey lines for FlightID 285.

The primary data drawn from the GNSS derived PPP solution and navigation file, are the coordinates

in a geodetic coordinate system, the time epochs, the three-component velocity vector, and the attitude controls (roll, pitch and yaw). In Figure 2.2 the PPP solution for FlightID 285 is shown. The time period (Seconds Of Week) SOW, from the GNSS is used as the reference time frame, for which the navigation data and IMU data are interpolated upon. This leads to the final result being in reference to the GNSS time epochs.

The IMU data is included in the raw format, which is the IMAR unit measures in the three component reference system body frame (b-frame). The sample frequency of the IMAR unit is at 300Hz. Therefore, a lowpass filter is implemented to mitigate the high frequency noise affecting the specific force measurements. Both the original down component and a lowpass filtered version can be seen in Figure 2.3. The sampling rate used is 120 s, resulting in a smoothing of the original signal. An extensive data-loader has been written in Python, to read and decode the IMAR output files. For further information on this, see Appendix B.2.1. [Johann et al., 2019, Enzlberger, 2018]

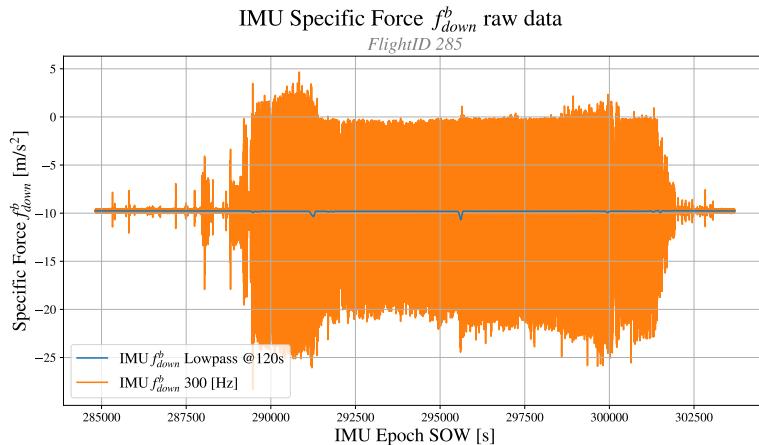


Figure 2.3: The down component of the specific force at 300 Hz (original) and the down component with a multi stage lowpass butterworth filter sampled at 120 s. The y-axis is given in m/s^2 and the x-axis is the IMU time epochs given in Seconds Of Week (SOW) [s]

The IMU used by DTU SPACE, is the iNAT-RQH-400x model provided by iMAR. This model has a high data rate of 300Hz and a high resolution of 0.000'3 degree in the attitude controls (roll, pitch and yaw). Further information about the instrument is found in [IMAR, 2015]. In addition to the IMU from iMAR, a temperature stabilization unit is also used to avoid thermal driven bias in the specific force measurements. [Simav et al., 2020] This product is the ICD iTempStap-AddOn also provided by iMAR. For further information on this instrument see [IMAR, 2018].

3 | Methodology

In the following section, the theoretical aspects of the direct strapdown method are described. A short introduction to the implementation of each sub routine is also included. For a more detailed code review, see Appendix [B].

3.1 Computation of GNSS position derived acceleration

As described from the GNSS receiver the PPP solution is obtained of the position vector \mathbf{r}_{GNSS}^n . For this scope of work, two different methods have been implemented to calculate the acceleration $\ddot{\mathbf{r}}_{D,GNSS}^n$. The first method works by using interpolating using spline functions and then calculating the 2. order derivative. The second method utilizes discrete differences. The GNSS epoch series are used to estimate an equally spaced time vector, used as a basis for interpolation of the PPP solution height estimate. The result of the interpolation is then used to compute the vertical accelerations. [Johann et al., 2019, Enzlberger, 2018]

3.2 Conversion from body *b*-frame to navigation *n*-frame

As explained in the introduction, the method of the direct strapdown utilizes both an IMU and the GNSS PPP solution and navigation file information. Before the two different subsystems can be compared and analyzed, a transformation of reference frame is needed. The GNSS solution \mathbf{r}_{GNSS}^n is given in the navigation frame, denoted by the superscript "n", and the IMU specific force measurements \mathbf{f}_{IMU}^b is given in the body reference frame, in this work denoted by the superscript "b". See Figure 3.1 for a illustration of the relations between the navigation and body frame. [Enzlberger, 2018]

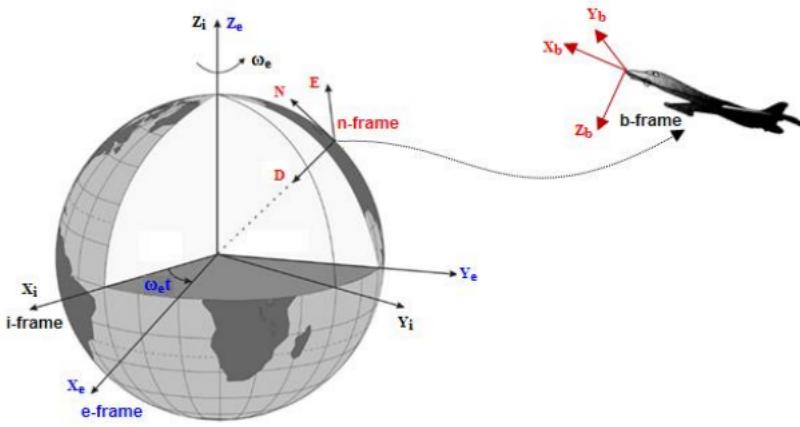


Figure 3.1: A graphical interpretation of the reference frames relation to one another. In this work the body *b*-frame and the navigation *n*-frame is used. [Nourmohammadi and Keighobadi, 2018]

The local-level frame (LLF) or navigation frame (n), is used to represent a vehicle or an objects attitude and velocity when moving in relation to the earth, either on the surface or at a low altitude. The origin of the frame is at the center of the sensor/system of interest and is traditionally oriented around an orthogonal right handed system, with the y-axis towards true north, the x-axis towards east and the z-axis pointing up perpendicular to the reference ellipsoid. This constellation is commonly referred to as the ENU (East-North-Up) frame. however for this project an alternative version is utilizes using the NED (North-East-Down) configuration, meaning that the z-axis is pointing inwards perpendicular to the reference ellipsoid following the plumb-line. [Noureldin et al., 2013, Enzlberger, 2023].

As seen in Figure 3.1 the body frame b -frame, is a representation of the local movement/attitude of an object. The origin is commonly at the center of gravity of the vehicle or measurement unit, which in this project is the IMAR unit. For this project, the axis is aligned with the airplane such that the y-axis is pointing straight forward, describing the roll of the system. The x-axis points towards the transverse direction describing the pitch of the system and the z-axis finalizes the right hand coordinate system pointing towards the vertical, this axis describes the yaw of the system. [Noureldin et al., 2013].

As the output gravity disturbance is aimed to be given in the navigation frame, a reference frame transformation is needed. This is done by converting the specific force measurements from the IMU from b -frame to n -frame, as given by:

$$\mathbf{f}^n = \mathbf{C}_b^n \mathbf{f}^b \quad (2)$$

Where \mathbf{C}_b^n is the rotation matrix, based upon the Euler angles roll, pitch and yaw. \mathbf{f}^n and \mathbf{f}^b are the specific force measurements, in the n -frame and b -frame respectively. The rotation matrix is defined as:

$$\mathbf{C}_b^n = \begin{bmatrix} \cos(p) \cos(y) & \cos(p) \sin(y) & -\sin(p) \\ \sin(r) \sin(p) \cos(y) - \cos(r) \sin(y) & \sin(r) \sin(p) \sin(y) + \cos(r) \cos(y) & \sin(r) \cos(p) \\ \cos(r) \sin(p) \cos(y) + \sin(r) \sin(y) & \cos(r) \sin(p) \sin(y) - \sin(r) \cos(y) & \cos(r) \cos(p) \end{bmatrix} \quad (3)$$

Where p denotes the pitch, y the yaw and r the roll. The attitude is obtained via the navigation file. For a more detailed description of the implementation in Python see Appendix B.1.3 [Enzlberger, 2018]

3.2.1 Position transformation to IMU location

As the survey setup is based upon both the IMU system and a GNSS receiver, a position translation between the GNSS receiver location and IMU location is needed. To do this, the location of the GNSS receiver in relation to the IMU is needed. In this project, the position is given in the n -frame. Having the GNSS position vector $\mathbf{r}_{\text{GNSS}}^n$, the position transformation is given as:

$$\mathbf{r}^n = \mathbf{r}_{\text{GNSS}}^n - \mathbf{T}_r^p \mathbf{l}^n \quad (4)$$

Where \mathbf{r}^n is the GNSS derived PPP solution given in the IMU location of the setup, \mathbf{l}^n is the relative position of the GNSS receiver in relation to the IMU, and \mathbf{T}_r^p is the transformation matrix from Cartesian

to geodetic coordinates defined by: [Johann et al., 2019]

$$T_r^p = \begin{bmatrix} \frac{1}{R_n+h} & 0 & 0 \\ 0 & \frac{1}{(R_e+h)\cos(\phi)} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (5)$$

Where h is the ellipsoidal height and R_n and R_e is curvature of the Earth defined by:

$$R_n = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \quad (6)$$

And

$$R_e = R_n \frac{1 - e^2}{1 - e^2 \sin^2 \phi} \quad (7)$$

Where a is the semi-major axis of Earth, $a = 6378137.0[\text{m}]$ and e^2 is the squared value of Earth's eccentricity given at: $e^2 = 6.69437999014 \cdot 10^{-3}$. For a more detailed description of the implementation in Python see Appendix B.1.1 [Clynch, 2002, Enzberger, 2018]

3.3 Computation of the Eötvös correction (δg_{eot}^n)

As the gravity disturbance is computed based upon measurements taken while in motion, a correction based on the rotation of the earth and the movement of the IMU/GNSS relative to Earth. This is done by computation of the Eötvös correction, which is defined by: [Wei and Schwarz, 1998, Titterton and Weston, 2011]

$$\delta g_{eot}^n = (2\Omega_{ie}^n + \Omega_{en}^n) \cdot \dot{\mathbf{r}}^n \quad (8)$$

Where Ω_{ie}^n is the skew-symmetric matrix of the Earth's rotation rate $\Omega_{ie}^n = [\omega_{ie}^n \times]$, with earth's scalar rotation rate $\omega_{ie} = 7,292115 \cdot 10^{-5} \text{ s}^{-1}$.

$$\Omega_{ie}^n = \omega_{ie} \begin{bmatrix} 0 & \sin(\phi) & 0 \\ -\sin(\phi) & 0 & -\cos(\phi) \\ 0 & \cos(\phi) & 0 \end{bmatrix} \quad (9)$$

ϕ is the geodetic latitude given at each GNSS epoch. Ω_{en}^n is the skew-symmetric matrix of the transport rate, defining the relative rotation between the moving navigation frame and the earth fixed frame. It is defined by $\Omega_{en}^n = [\omega_{en}^n \times]$:

$$\Omega_{en}^n = \begin{bmatrix} 0 & \frac{\dot{r}_E \tan(\phi)}{R_e+h} & -\frac{\dot{r}_N}{R_n+h} \\ -\frac{\dot{r}_E \tan(\phi)}{R_e+h} & 0 & -\frac{\dot{r}_E}{R_e+h} \\ \frac{\dot{r}_N}{R_n+h} & \frac{\dot{r}_E}{R_e+h} & 0 \end{bmatrix} \quad (10)$$

Where ϕ is the geodetic latitude at each GNSS epoch, the $\dot{\mathbf{r}}_E$ and $\dot{\mathbf{r}}_N$ is the velocity components given in the North-East-Down frame, based upon the GNSS positioning vector \mathbf{r}^n . For this project the navigation file from the GNSS receiver was used, with the implementation of interpolation to the GNSS positioning

epochs.

With this, the Eötvös correction can be computed for each GNSS epoch in the survey. For the details on the implementation in Python see Appendix B.1.4 [Johann et al., 2019, W. Müller, 2012, Lynch, 2002]

3.4 Filtration of Signals

As described briefly in the data description section 2.2, the input IMU signal is low pass filtered using a multi-stage Butterworth lowpass filter. The purpose is to remove high frequency noise from the input signals. The same method is used for the navigation file and GNSS position file. As described in [Johann et al., 2019] and the final result of the gravity disturbance is also lowpass filtered again to decrease the high frequency noise which could have mitigated through the first filtration. The implemented Butterworth filter is based upon an implementation made by Tim E. Jensen. For the details of the implementation in Python see Appendix B.1.5.

3.5 Computation Gravity Disturbance δg^n and normal gravity γ^n

The final step in the direct method before corrections, is the determination of the gravity disturbance result based on the measurement from the survey. The gravity disturbance δg^n is defined as the difference between the gravity vector \mathbf{g}^n and the normal gravity vector $\boldsymbol{\gamma}^n$. Note that in this case, it is defined in the n -frame: [Kwon and Jekeli, 2001, W. Müller, 2012, Lowrie, 2007]

$$\delta \mathbf{g}^n = \mathbf{g}^n - \boldsymbol{\gamma}^n \quad (11)$$

Where the gravity vector \mathbf{g}^n is computed based upon the specific force from the IMU, the acceleration from the GNSS receiver, and the Eötvös correction by:

$$\mathbf{g}^n = \ddot{\mathbf{r}}^n - \mathbf{f}^n + \delta \mathbf{g}_{eot}^n \quad (12)$$

This means that only the normal gravity $\boldsymbol{\gamma}^n$ is not yet determined. A full description of the determination of this is out of the scope of this work. Instead, see [Enzilberger, 2023]. Note that for this project, only the normal gravity in the navigation reference frame is used, however a full implementation was produced in python for later use. For the details on this implementation see Appendix B.1.6

3.6 Bias and Drift Determination

Having obtained the down component of the gravity disturbance δg_D^n , a drift and bias correction can be introduced. This correction is performed as accelerometer measurements can be subject to significant bias and drift over time. The correction is applied based upon the calculated gravity disturbance as in [Johann et al., 2019]. The correction is defined as: [Glennie et al., 2000]

$$\kappa(t) = \kappa_1 + \frac{t - t_1}{t_2 - t_1} (\kappa_2 - \kappa_1) \quad (13)$$

Where κ_1 and κ_2 are the reference corrections from the start and the end of the survey. t_1 and t_2 are the start and end times. t is the current GNSS epoch. For the cases analyzed in this project, no reference gravity $g_{ref,i}$ is available. This is omitted by determining the reference correction based upon the gravity disturbance in the stationary periods before and after the survey flight. Here it is assumed that the difference in the parking location of the airplane is negligible due to the relative small variation of the gravity disturbance field, as long as the difference is on the scale of meters. The determination of the stationary period is based on the magnitude of the velocity $\dot{\mathbf{r}}_{GNSS}^n$, coming from the associated navigation file. A threshold value of 0.001 [m/s²] is chosen and a minimum of 100 epochs is used for determining a mean value of the gravity disturbance. Utilizing this information, a reference correction is computed using the following:

$$\kappa_i = \mu(\delta g_{D,i}) - \mu(\delta g_{D,ref,1,2}) \quad (14)$$

Where κ_i is the correction for the current GNSS epoch and $\mu(\delta g_{D,i})$ is the mean of the down component of the gravity disturbance at the start or end stationary periods over a minimum of 100 GNSS epochs. $\mu(\delta g_{D,ref,1,2})$ is the mean of both the start and end periods. As no reference gravity is available during flight either, only the linear drift/bias can be computed. The final correction is applied to the gravity disturbance estimate by:

$$\delta g_{D,corr}^n = \delta g_D^n - \kappa(t) \quad (15)$$

The implementation of the method is described in further detail in Appendix B.1.10.

4 | Case Study

To test and verify the implementation of the direct strapdown method two similar surveys have been used. The two cases are referred to as DK2022 or FlightID 285 for the main case and Roskilde16 or FlightID 116 for the secondary analysis set. Both these surveys has been conducted using a smaller airplane and the same IMAR unit. In Figure 4.1, the survey lines for both surveys can be seen. These two cases have been used, as they provide a common ground for verification of the method, as both surveys have been conducted between the same set of waypoints in the north-west direction. This means that a forward and backward pass can be computed, which will be elaborated further later in the case section. Further results for both cases are available via Tim E. Jensen and DTU Space, where the indirect method has been used to compute the gravity disturbances.

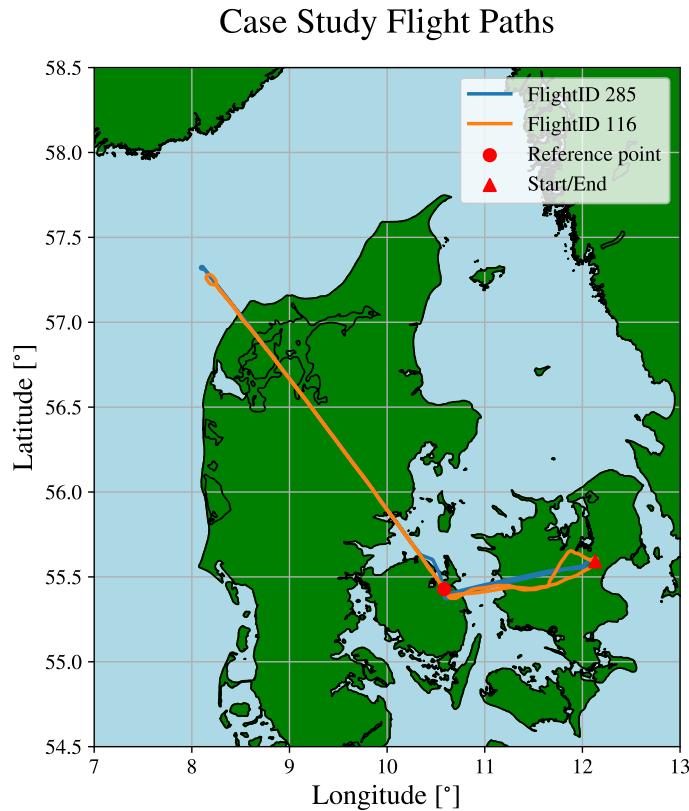


Figure 4.1: Overview of the two case studies incorporated in the scope of work. FlightID 285 is depicted as the blue line, FlightID 116 is depicted as the orange line, the red circle is the reference point at Kølstrup Forsamlingshus, used for verification of the method and the red triangle is the starting and ending position at Roskilde airport. All information is given in accordance to the WGS84 coordinate system.

4.1 DK2022 - FlightID 285

The first of the two case studies is the DK2022 or FlightID 285 survey. This survey was conducted as a test flight of the equipment in 2022 over the area of Denmark. Starting point is the Roskilde Airport, and a set of waypoints for verification are placed just outside the city of Odense and Northeast of Hanstholm in the northern part of Jutland see Figure 4.1. An overview of the more specific flightlines and forward and backwards pass's can be seen in Figure 4.2 below.

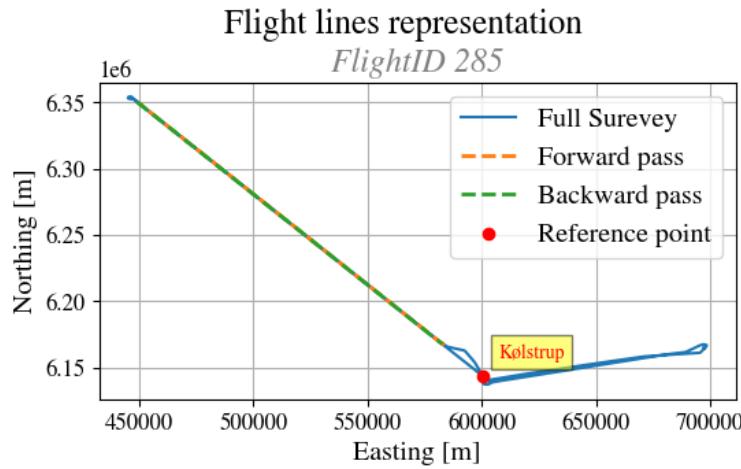


Figure 4.2: Survey lines for FlightID 285. The geodetic coordinates has been converted to UTM32 providing the x-axis in the Easting direction given in [m] and the y-axis the Northing direction given in [m]. Blue line is the original full survey. Yellow striped line is the forward pass. Green striped line is the backward pass and the red dot, is the reference location of Kølstrup Forsamlingshus. [SDFI, 2017]

The survey was flown with the IMAR IMU unit sampling at 300Hz and the GNSS receiver at 1Hz, the spatial position of the IMAR unit to the GNSS receiver is measured in the n -frame as presented in section 3.2 and has the values:

$$l^n = \begin{bmatrix} 1.570 \\ 0.170 \\ -1.470 \end{bmatrix} m \quad (16)$$

For the full survey length, the IMU data vector contains 5662914 datapoints, sampled at 300Hz. The GNSS data vector contains 18872 datapoints and the navigation file contains 17997 datapoints, where both the GNSS and navigation information is sampled at 1Hz. Before the direct strapdown method can be initialized a lowpass filtration is needed for the IMU data, as described in Section 2.2. The data is lowpass filtered at 120 s, to avoid high frequency errors mitigating through the analysis.

Through the methodology explained in Figure 2.1, an estimate of the gravity disturbance is computed. This result is obtained for the full time series of GNSS time epochs, however only the time epochs between the two waypoints are relevant for this scope of work. A subset of the resulting gravity disturbance is shown in Figure 4.3 as a function of time epoch.

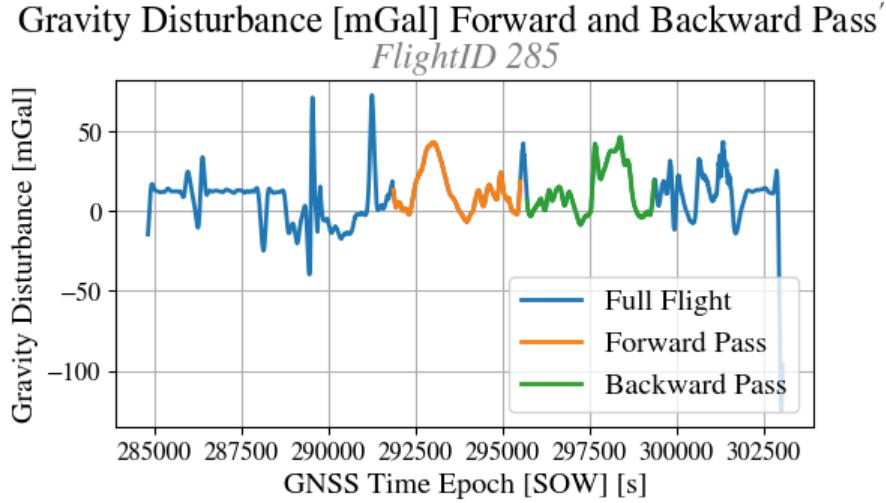


Figure 4.3: The Gravity Disturbance [mGal] calculated with the Direct Strapdown method for the FlightID 285. The Gravity Disturbance is given vs. the GNSS time epochs [SOW] [s]. For reference for later results, the Forward and Backward pass's has been highlighted with orange and green line respectively.

Shown in Figure 4.3, are both the Full flight line and the forward and backward passes of the verification subset of the timeseries. The determination of the length of the verification passes is out of the scope of this report. It is, however based upon the distance from the reference point at Kølstrup forsamlingshus and the stability in the attitude controls from the navigation file. The decision of the reference point is based upon a linear continuation of the verification lines in the south-west direction.

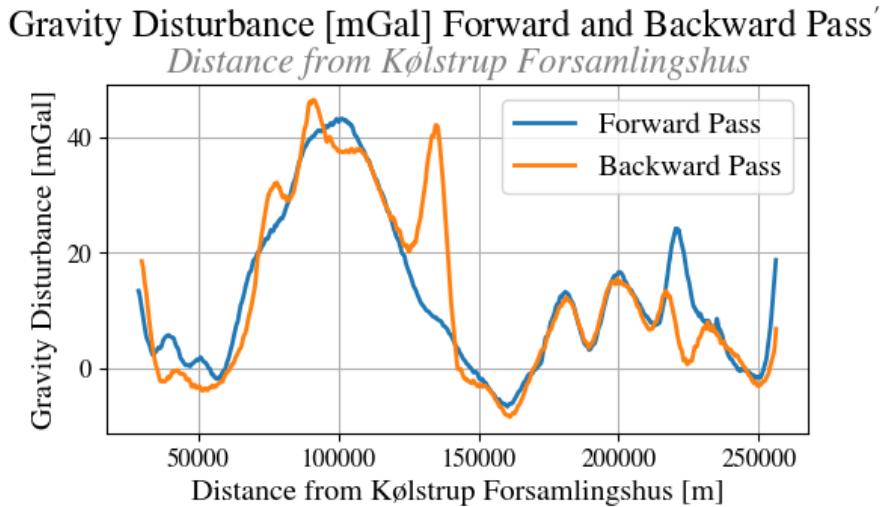


Figure 4.4: Results of the verification process for FlightID 285. Blue line is the forward pass and the orange line the backward pass. Both signals is the gravity disturbance [mGal] given vs. the distance in [m] from the reference point. The backward pass has been flipped 180°] around the y-axis.

The result shown in Figure 4.3 has been subtracted by 140 [mGal] for better comparability with the result from the indirect method. For this scope of work the computed gravity disturbance is not the final product, as the aim for the project is the implementation of the direct strapdown method. To verify the implementation of the method, the verification subset or verification lines are used. In Figure 4.4 the gravity disturbance for the forward and backward passes are shown as a function of the distance to the reference point, placed at Kølstrup forsamlingshus (see Figure 4.2). The blue line is the forward pass and the orange line the backward pass. As previously explained, the general output of the gravity disturbance is not the significant result, however it is worth noting that the large amplitude signal between 50km - 150km from the reference point in both signals, is probably a product of the Silkeborg Gravity High [Strykowski, 1999]. It can be observed that the backward pass diverges significantly from the forward pass in high frequency periods. The Root Mean Square Error (RMSE) is determined for the residual between the two passes, which is found to be RMSE = 7.12mGal [Forsberg and Olesen, 2010, John R. Taylor, 1997]. This is much higher than what is found in [Johann et al., 2019]. The large RMSE value means that the implementation is not stable as it needs to be. To further analyze this result, the computed gravity disturbance using the indirect method is used as a reference. [Data originates from Tim E. Jensen at DTU Space]. A comparison of the two results can be seen in Figure 4.5.

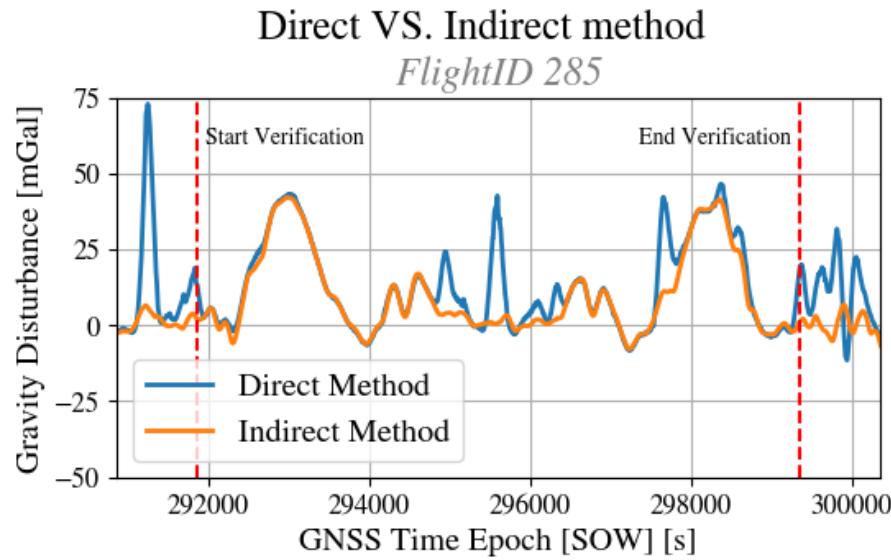


Figure 4.5: Results of the direct strapdown method and the indirect method provided by Tim E. Jensen, DTU. The result of the indirect method has been interpolated upon the GNSS time epoch used in direct estimation. The blue line is the direct method and orange line the indirect method. 140 [mGal] has been subtracted from the direct method result, for a better comparison. Vertical red dotted lines indicate the start and end of the verification subset lines.

In Figure 4.5 the results from the direct and indirect method are compared. The results from the indirect method have been interpolated on to the GNSS time epoch used for the direct method. The two vertical red dotted lines indicate the start and end of the verification subset of the timeseries, note that the point

at which the plane turned around is still incorporated. The difference between the two results is most significant in the second half of the signal, indicating the backward pass. High frequency residuals between the two signals are observed. However it is also observed that the indirect method is subject to deviations in the forward and backward pass verification. This is most significant around the Silkeborg gravity high.

Further analysis of the possible causes for the instability of the results is found in the section 5.

4.2 Roskilde16 - FlightID 116

The second of the two case studies is the Roskilde16 or FlightID 116 survey. This survey was conducted in 2016 in the same area as the first case study, see Figure 4.1. Small variations in the survey lines are to be expected, but the two surveys are flown following the same set of waypoints. The specific flight lines with forward and backward passes are shown in Figure A.1 in Appendix A. Only a short description of the results for this survey is showcased in this section, as the survey proved troublesome and did not yield enough useful information to be included here. Yet some of the results are included in the discussion, therefore the inclusion of the survey.

Overall the survey was conducted using an IMAR IMU unit sampling at 300Hz in the *b*-frame like for case one. Note that the IMU data for this survey has not been through a thermal correction, as described by [Johann et al., 2019, Becker et al., 2015], which the IMU data for case one was. This should result in a drift or bias trend in the timeseries. The GNSS PPP solution is obtained in 2Hz and the navigation file is sampled at 1Hz. As for case one the difference between the location of the IMU and GNSS receiver is described in the *n*-frame, and was measured at:

$$\mathbf{l}^n = \begin{bmatrix} -0.345 \\ -0.323 \\ -0.653 \end{bmatrix} m \quad (17)$$

Working through the direct strapdown yields the resulting verification line comparison shown in Figure A.2 in Appendix A. Here, it is observed that the resulting gravity disturbance for the survey is subject to high frequency noise, but more or less follows the same long frequency signal also observed in the first case study. The RMSE value for the residual between the two verification lines is determined to RMSE = 34.44mGal [Forsberg and Olesen, 2010], which is roughly a factor 5 higher than for the first case. A further discussion of possible causes of this unstable result may be found in the discussion in section 5.

5 | Discussion

The following section includes a discussion of the results from the two case studies. The main focus for this will be on the FlightID 285 due to the problems with FlightID 116 as explained in section 4.2. The main focus of the discussion is around probable causes of the instability in the results of the direct strapdown method. The gravity disturbance computed based upon the indirect method is used as a reference solution.

5.1 Attitude analysis

The results from FlightID 285, shown in Figure 4.3, indicated that high frequency errors are mitigating through the implementation of the direct strapdown method. The origin of these high frequency disturbances is difficult to conclude. However, an analysis of the attitude controls for the airplane may provide an idea of where the highest variance is rooted. In Figure 5.1, the attitude controls, roll, pitch and yaw for FlightID 285 are plotted after an interpolation to the GNSS time epoch, used throughout the project analysis.

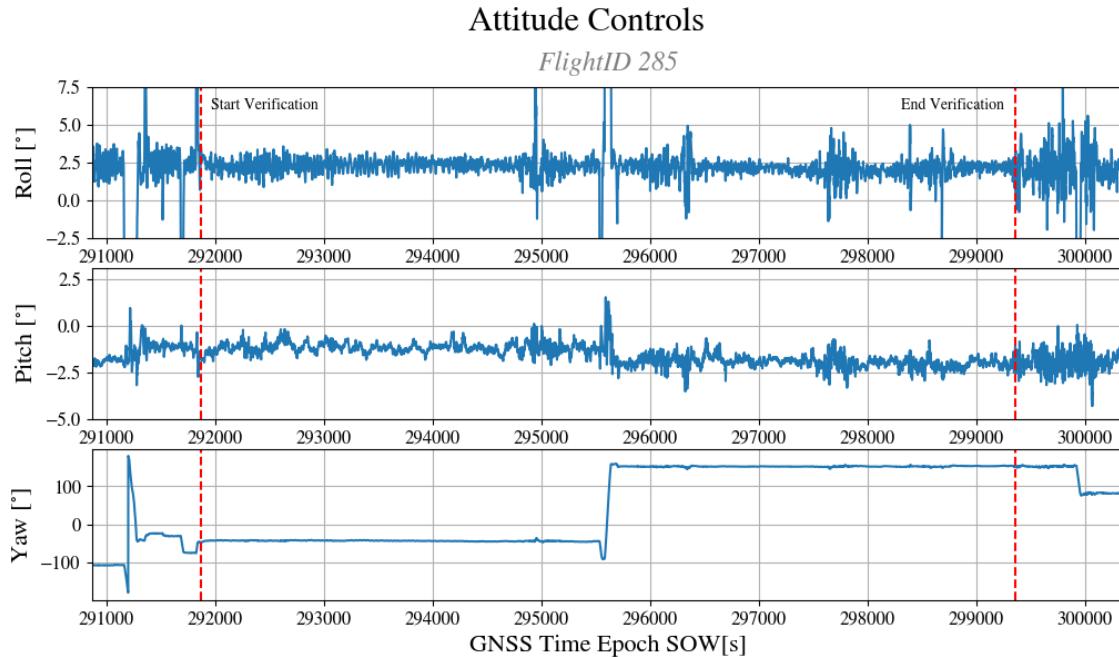


Figure 5.1: Attitude controls from the navigation file, interpolated upon to the GNSS Time epochs. Top: Roll angle, middle: Pitch angle, and bottom: Yaw. The timeframe is around the subset used for the verification indicated by the vertical start and end verification lines. Note that the point at which the plane turned around is included in the timeseries.

As described in [Johann et al., 2019] one correction based on the yaw is the heading error of the plane.

has not been implemented, however, this should not cause the high frequency deviations seen in the results. More interesting is the physical aspect of the pitch and roll angles, in relation to the method in general. Assume that the airplane is subject to a high frequency vibration with a relative stable periodic behavior, due to the engines of the plane. These vibrations would probably not mitigate through to the sampled attitude controls, but would be visible in the IMU readings. Furthermore, assume the airplane/helicopter is subject to wind gust or turbulence. This would greatly impact the attitude of the moving subject. However, it is not certain that the sampling rate of the GNSS receiver is sufficiently high to sample the impact, resulting in a loss of information, which is clear in the IMU readings instead. The implementation of the direct method in this project, has been on the basis of the scalar gravimetry where only the downward component of the gravity disturbance has been computed. This means that a fast change in roll or pitch would significantly change the orientation of the n -frame, such that the estimation of the downward component is affected by an unknown deflection of the vertical.

A statistical overview of the attitude controls for both case studies is shown in Table 5.1 below.

Flight ID 285		
	Roll [°]	Pitch [°]
μ	2.34	-1.60
σ	2.45	0.53
σ^2	6.00	0.28

FlightID 116		
	Roll [°]	Pitch [°]
μ	5.72	7.55
σ	1.23	0.50
σ^2	1.52	0.25

Table 5.1: Mean, standard deviation and variance of the attitude controls for FlightID 285 and 116. The statistical analysis is based only upon the time epochs used in the verification process.

The yaw is excluded from the table, as the statistical properties of a heading angle were deemed unimportant/inconsequential for/to the results. For both case studies it can be observed that the roll is the attribute subjected to the highest degree of variation. It can interestingly also be observed that the mean of the roll in both cases is not 0 degrees, as one could assume. This could be due to the GNSS receiver not being completely vertical. From the results found for FlightID 116, it could also be assumed that, the attitude controls are more unstable compared to FlightID 285. Yet, this is shown not to be the case.

As shown in Table 5.1, the roll attribute is the most unstable of the attitude controls, motivating a further analysis. As in section 4.1, the results for the gravity disturbance based upon the indirect method is used to evaluate the results for the direct method. In Figure 5.2 the difference between the direct and indirect methods, is plotted on the same timeline as the roll degree.

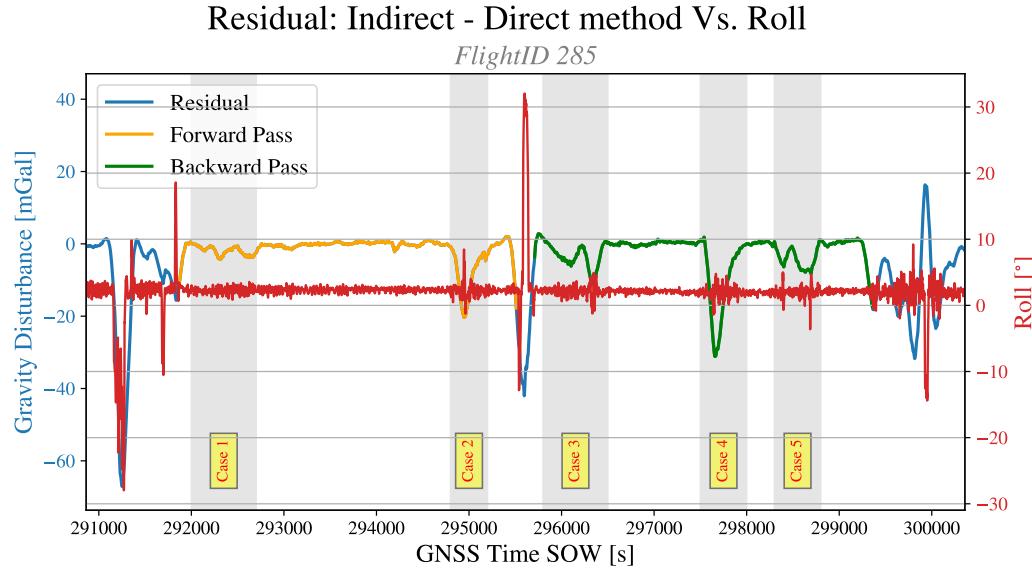


Figure 5.2: The residual between the gravity disturbance computed based upon the direct and indirect method for FlightID 285. The indirect estimations have been interpolated upon to the GNSS time epoch used in the implementation for the direct method. Red signal depicts the Roll angle [$^{\circ}$] vs GNSS time epoch. Highlighted grey columns indicate the 1-5 cases where the residual diverges significantly from 0 mGal, inside the verification timeframe.

Highlighted in Figure 5.2 are case 1-5, which indicate when the residual between the direct and indirect estimated gravity disturbances significantly diverge from 0 mGal inside the timeframe of the verification subset. It is observed that these cases coincide with an increase in the variance of the roll angle. Therefore, a clear indication is evident that the current implementation of the direct method is not sufficient for correcting for high frequency changes in the airplane attitude.

Returning to the FlightID 116 results, the values in Table 5.1 may be further scrutinized and compared to figure A.3 in appendix A. As previously discussed, nothing in the attitude controls indicate that the high frequency errors that we see mitigating through to the result. It is therefore unlikely that the problem is rooted in these, instead if we look at the implementation of the software. As the GNSS receiver has a sampling rate of 2Hz compared to the 1Hz for FlightID 285, this has proven to create problems with high frequency errors throughout the analysis. Furthermore, the raw IMU data is not as stable in amplitude as for FlightID 285. This could also lead to a GNSS driven bias in the final product. This is, however, speculation in the quality and robustness of the implementation. Further analysis is necessary for validation. This report has focused on the implementation of a scalar gravimetric computation of the downward component of the gravity disturbance. As the other components have been excluded from the implementation, useful information may be lost. The attitude effects discussed previously are an example of this. If the method included a derivation of all three components of the gravity disturbance, then an analysis of the deflection of the vertical could be performed, which could then be used to correct the resulting product.

6 | Conclusion

The main objective of this project was to write an implementation in Python of the direct strapdown method, for calculating gravity disturbance based IMU and GNSS data. Later on, this was expanded to also include a implementation of a routine for reading and converting the raw hexadecimal encoded IMAR IMU .bat files. This was done to create an alignment of the necessary routines from taking the data from A to B where B is the end result. The result of this dual project has been a complete implementation of a routine to read and convert the raw IMU data, as well as a simple implementation of the direct strapdown method.

The determination of the downward component of the gravity disturbance was of highest interest, therefore the implementation was written to compute this. This lead to different problems with unstable results in high turbulent surveying, as described in section 5.1. Furthermore, it was found that the implementation is rather unstable when it comes to sampling rates outside of 1Hz and 300Hz for the GNSS receiver and IMU respectively.

An analysis of the attitude controls showed that the fast changes in the roll were correlated with some of the high frequency errors that were found in the result for FlightID 285. However, as only one result was obtained it is not possible to conclude a direct correlation. For this, a more substantial analysis would be necessary. This could, however, solve some other problems with the implementation as well. In the following subsection a short overview of the possible work yet to be undertaken is listed.

6.1 Further Work

The following section is a list of possible ways to further work with the implementation and testing of the work presented in this report.

1. An inclusion of tie values to correct offsets between lines. Tie value correction is a common practice in both airborne gravimetry and magnetic surveys [Johann et al., 2019, Hinze et al., 2013, Hwang et al., 2006]. As the case studies used in this scope of work were fairly simple with no overlaps, tie lines were not flown/tie line corrected was not implemented, which when used in practice is a necessity.
2. A full implementation of all three gravity disturbance components. This should be a fairly quick change to the implementation and it would increase the possibility of analysis.
3. An analysis and solution for the instability against changes in input sampling rates, that proved a difficulty with the FlightID 116 survey. This would also mean determining an alternative way of filtering the attitude controls, and the high variance input IMU components.
4. An implementation of quaternions as a routine to transform from different reference frames, which could lead to a reduction in computation power. [Enzlberger, 2018]

To further reduce the computation power and therefore speed up the computation, would be an implementation in C or another low level language like Fortran. Lastly, a full implementation as a software product, where the routine is run through executable(s) files, or inputs to a command prompt.

References

- Becker, D. (2016). Advanced Calibration Methods for Strapdown Airborne Gravimetry.
- Becker, D., Nielsen, J. E., Ayres-Sampaio, D., Forsberg, R., Becker, M., & Bastos, L. (2015). Drift reduction in strapdown airborne gravimetry using a simple thermal correction. *Journal of Geodesy*, 89(11), 1133–1144. <https://doi.org/10.1007/s00190-015-0839-8>
- Clynch, J. R. (2002). Radius of the Earth - Radii Used in Geodesy.
- Enzlberger, T. (2018). *General rights Airborne Strapdown Gravity Measurements for Geodesy and Geophysics* (tech. rep.).
- Enzlberger, T. (2023). *General rights Airborne Strapdown Gravity Measurements for Geodesy and Geophysics* (tech. rep.).
- Forsberg, R., & Olesen, A. V. (2010). Airborne gravity field determination. *Sciences of Geodesy - I: Advances and Future Directions*, 83–104. [https://doi.org/10.1007/978-3-642-11741-1__3/COVER](https://doi.org/10.1007/978-3-642-11741-1__3)
- Glennie, C. L., Schwarz, K. P., Bruton, A. M., Forsberg, R., Olesen, A. V., & Kms, K. K. (2000). A Comparison of Stable Platform and Strapdown Airborne Gravity.
- Hinze, W. J., Frese, R. R. B. V., & Saad, A. H. (2013). *Gravity and Magnetic Exploration* (tech. rep.). Cambridge University Press. New York.
- Hwang, C., Hsiao, Y. S., & Shih, H. C. (2006). Data reduction in scalar airborne gravimetry: Theory, software and case study in Taiwan. *Computers and Geosciences*, 32(10), 1573–1584. <https://doi.org/10.1016/J.CAGEO.2006.02.015>
- IMAR. (2015). *Technical Data of iNAT-RQH-400x* (tech. rep.). www.imar-navigation.de
- IMAR. (2018). *ICD TempStab-AddOn* (tech. rep.). www.imar-navigation.de
- Jekeli, C., & Garcia, R. (1997). GPS phase accelerations for moving-base vector gravimetry. *Journal of Geodesy*, 71(10), 630–639. <https://doi.org/10.1007/S001900050130/METRICS>
- Jensen, T. E., & Forsberg, R. (2018). Helicopter test of a strapdown airborne gravimetry system. *Sensors (Switzerland)*, 18(9). <https://doi.org/10.3390/s18093121>
- Johann, F., Becker, D., Becker, M., Forsberg, R., & Kadir, M. (2019). The Direct Method in Strapdown Airborne Gravimetry – a Review.
- John R. Taylor. (1997). *An Introduction To Error Analysis The study if Uncertainties in Physical Measurements* (Second Edition). University Science Books.
- Kwon, J. H., & Jekeli, C. (2001). *A new approach for airborne vector gravimetry using GPS/INS* (tech. rep.).
- Lowrie, W. (2007). *Fundamentals of Geophysics* (Vol. Second Edition). Cambridge University Press. <http://ebooks.cambridge.org/ebook.jsf?bid=CBO9780511807107>

- Noureldin, A., Karamat, T. B., & Georgy, J. (2013). *Fundamentals of inertial navigation, satellite-based positioning and their integration*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-30466-8>
- Nourmohammadi, H., & Keighobadi, J. (2018). Fuzzy adaptive integration scheme for low-cost SIN-S/GPS navigation system. *Mechanical Systems and Signal Processing*, 99, 434–449. <https://doi.org/10.1016/j.ymssp.2017.06.030>
- SDFI. (2017). UTM/ETRS89: Den primærekortprojektion i DanmarkGeodætisk. *The Geomatics Notes Series*.
- Simav, M., Becker, D., Yildiz, H., & Hoss, M. (2020). Impact of temperature stabilization on the strapdown airborne gravimetry: a case study in Central Turkey. *Journal of Geodesy*, 94(4). <https://doi.org/10.1007/s00190-020-01369-5>
- Strykowski, G. (1999). *Silkeborg Gravity High Revisited: Horizontal Extension of the Sbource and its Uniqueness* (tech. rep. No. 4).
- Titterton, D., & Weston, J. (2011). Basic principles of strapdown inertial navigation systems. *Strapdown Inertial Navigation Technology*, 17–58. https://doi.org/10.1049/PBRA017E{_}CH3
- Vyazmin, V., Golovan, A., & Bolotin, Y. (2021). *New Strapdown Airborne Gravimetry Algorithms: Testing with Real Flight Data* (tech. rep.).
- W. Müller, T. (2012). *Geodesy* (4th, ed). de Gruyter.
- Wei, M., & Schwarz, K. P. (1998). Flight test results from a strapdown airborne gravity system. *Journal of Geodesy*, 72(6), 323–332. <https://doi.org/10.1007/S001900050171/METRICS>

A | Roskilde16 - FlightID 116 Results

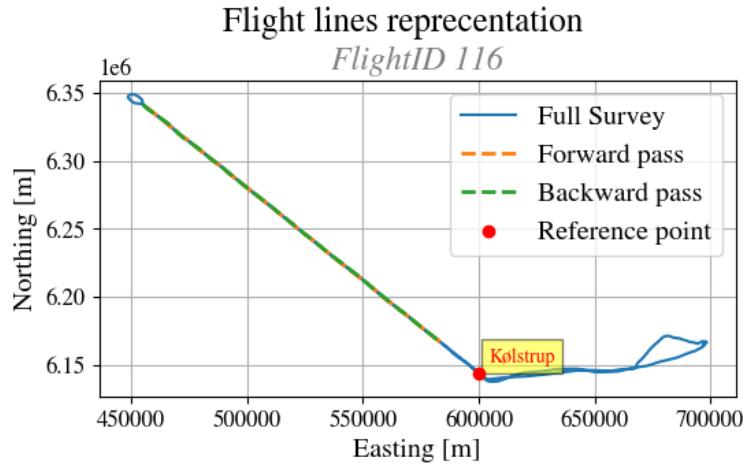


Figure A.1: Survey lines for FlightID 116. The geodetic coordinates has been converted to UTM32 providing the x-axis in the Easting direction given in [m] and the y-axis the Northing direction given in [m]. Blue line is the original full survey. Yellow striped line is the forward pass. Green striped line is the backward pass and the red dot, is the reference location of Kølstrup Forsamlingshus. [SDFI, 2017]

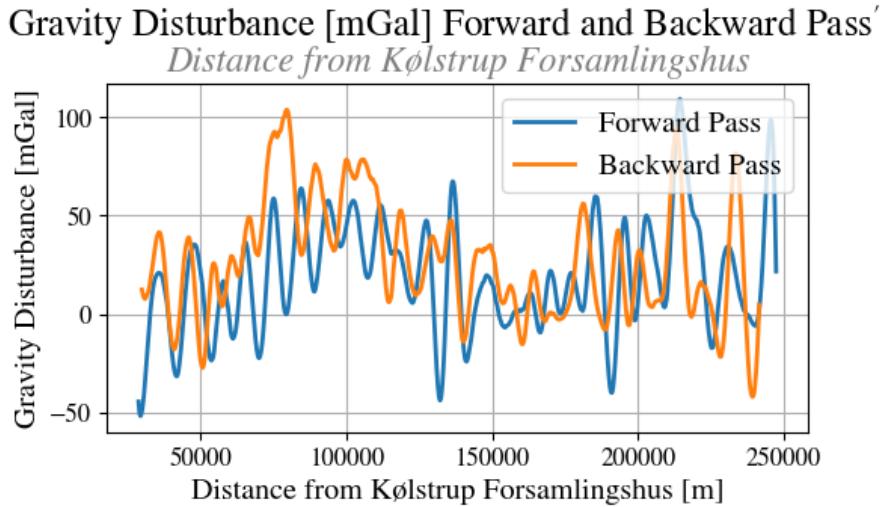


Figure A.2: Results of the verification process for FlightID 116. Blue line is the forward pass and the orange line the backward pass. Both signals is the gravity disturbance [mGal] given vs. the distance in [m] from the reference point. The backward pass has been flipped 180[°] around the y-axis. The Forward and Backward pass is further based upon the verification lines from the FlightID 285 analysis.

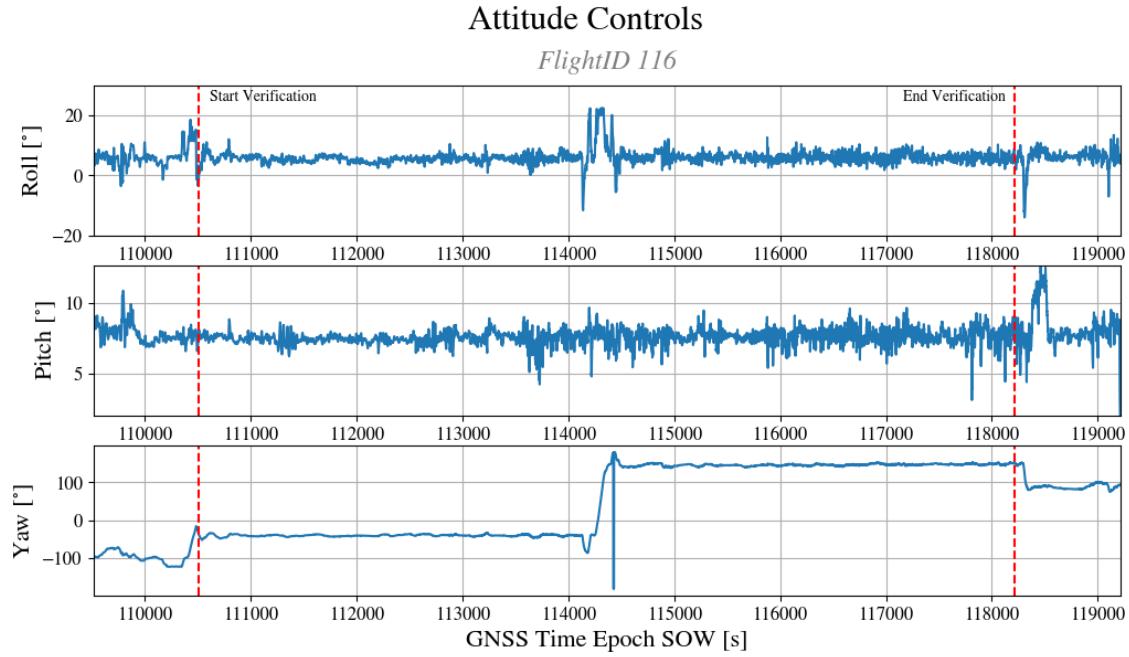


Figure A.3: Attitude controls from the navigation file, interpolated upon to the GNSS Time epochs. Top: Roll angle, middle: Pitch angle and bottom: Yaw. The timeframe is around the subset used for the verification indicated by the vertical start and end verification lines. Note that the turn around is included in the timeseries.

B | Software Description

The following Appendix contains the documentation for each of the functions used to implement the Direct Strapdown method. These functions can be found in the GitHub repositories: <https://github.com/CSolgaard/Syntese>, the data for the case studies is not a part of this GitHub repositories. Appendix B.1.xx contains the descriptions of the source functions used in the method. Appendix B.2.xx contains the dataloader functions written to load and convert the raw IMAR data and the preprocessed GNSS and navigation data.

B.1 Direct Strapdown

This subsection of appendix B contains the docstrings and information about calling and working with the scripts that has been written and implemented in the scope of this work. The full code can be found the the Github Respiratory in the "src" subfolder. All functions is implemented in the python file "Direct_Strapdown.py".

B.1.1 pos_translate_v1()

```
1      """
2          Procedure to translate position according to lever arm
3          This is a python implementation of the Matlab code written by
4              Tim Jensen
5              (DTU) 24/11/20
6
7
8      Input:
9          lat            Nx1 array of latitude (GNSS position) [deg]
10         lon            Nx1 array of longitude (GNSS position) [deg]
11         h              Nx1 array of ellipsoidal height (GNSS position)
12             [m]
13         roll           Nx1 array of bank angle [deg]
14         pitch          Nx1 array of elevation angle [deg]
15         yaw            Nx1 array of heading angle [deg]
16         lever_arm     3x1 array of lever arm along body-axes (IMU→
17             GNSS) [m]
18
19      Output:
```

```
18     olat      Nx1 array of corrected latitude (IMU position) [  
19         deg]  
20     olon      Nx1 array of corrected longitude (IMU position)  
21         [deg]  
22     oh        Nx1 array of corrected height (IMU position) [  
23         deg]  
24
```

```
23     Author: Christian Solgaard (DTU – Master student) 30/01/2023  
24     """
```

B.1.2 gnss_accelerations_v1()

```
1     """  
2     Function that derives vertical accelerations from a time series  
3         of  
4         heights. This function is based upon the Matlab implementation  
5         written by  
6         Tim E. Jensen 27/09–2019
```

```
7  
8     Input:  
9     time    Nx1 array with time stamps [s]  
10    h       Nx1 array with heights [m]  
11    method   String can be either:  
12        splines  
13        difference  
14  
15     Optional input:  
16     tout     Mx1 vector denoting time stamps with evaluated  
17         accelerations  
18         [s]  
19  
20     Output:  
21     tout     Mx1 array with time stamps [s]  
22     acc      Mx1 array with accelerations [m/s^2]
```

22
23

24 Author: Christian Solgaard (DTU – Master student) 01/02–2023
25 """

B.1.3 b2n_v1()

```
1       """
2       Procedure that rotates a vector from body-frame to navigation-
3       frame
4       This is a python implementation of the Matlab code written by
5       Tim Jensen
6       (DTU) 24/11/20
```

```
6
7       Input:
8           time     Nx1 array of time stamps
9           bacc     Nx3 array of observations in body-frame
10          att      Nx3 array of attitude (bank, elevation, heading) [deg]
11
12       Output:
13          nacc     Nx3 array of observations in navigation-frame
```

16 Author: Christian Solgaard (DTU – Master student) 30/01/2023
17 """

B.1.4 transport_rate_v2()

```
1       """
2       Procedure to compute transport rate
3       This is a python implementation of the Matlab code written by
4       Tim Jensen
5       (DTU) 24/11/20
```

6

```
7      Input:  
8          time    Nx1 array of time stamps  
9          vel     Nx3 array of velocity (north ,east ,down) [m/s ]  
10         pos    Nx3 array of position (lat ,lon ,h) [deg ,deg ,m]  
11  
12      Output:  
13          tacc   Nx3 array of computed transport rate (north ,east ,down)  
14             ) [m/s^2]  
15
```

```
16      Author: Christian Solgaard (DTU – Master student) 31/01/2023  
17      """
```

B.1.5 but2_v2()

```
1      """  
2      2nd order multi-stage forward/backward butterworth filter.  
3      This is a python implementation of the Matlab code by Tim  
4          Jensen (DTU) 29/05/2020,  
5      Which is further based on the code by Rene Forsberg , Nov. 1997  
6
```

```
7      Input:  
8          data    Vector of data series  
9          stage   Number of iterations , 1 iteration = forwar+backward  
10         run  
11         ftc     Filter time constant [s]  
12         dt      Sample interval [s]  
13  
14      Output:  
15          fdata   Vector of filtered data series
```

```
18      Author: Christian Solgaard (DTU – Master student) 27/02–2023  
19      """
```

B.1.6 normal_gravity_precise_v1()

```
1      """
2      normal_gravity_precise — routine that performs an exact
3          computation of
4          the normal gravity vector according to WGS84. Input are
5          geodetic
6          coordinates (WGS84) and output is along the north , east and
7          down axes of
8          a local coordinate frame (n-frame). The down axis is defined as
9          being
10         perpendicular to the reference ellipsoid.
11
12
13     Input:
14         lat      Geodetic latitude [deg]
15         lon      Geodetic longitude [deg]
16         h        Ellipsoidal height [m]
17
18     Optional input:
19         lf      Integer value specifying resolving axes frame:
20                 0: Ellipsoidal—harmonic (u,beta,gamma)
21                 1: Rectangular (x,y,z)
22                 2: Spherical (r,lat,lon)
23                 3: Geodetic (north,east,down) <— DEFAULT
24
25     Output:
26         g      MATLAB structure with 3 grids corresponding to the
27             three
28                 components of the specified reference frame [m/s^2]
```

30 Author: Christian Solgaard (DTU – Master student) 02/02–2023

31

"""

B.1.7 interpolate_DS()

1

"""

2

Interpolation function , based on the scipy package

3

Inputs :

4

var1 : ... (np.array)

5

var2 : ... (np.array)

6

var3 : ... (np.array)

7

kind : ... (str)

8

bounds_error : ... (str)

9

fill_value : ... (str)

10

11

12

Author: Christian Solgaard (DTU – Master student) 02/02–2023

13

"""

B.1.8 movmean()

1

"""

2

A simple implementation of a lowpass filter .

3

The lowpass filter is based on a moving mean method , computed
using

4

a kernel window and convolution .

5

6

Input :

7

array : data [N x 1] , (np.array)

8

window_size : size of moving kernal , (int)

10

11

Output :

12

moving_averages : data [N–window_size x 1] , (np.array)

13

14

15

Author: Christian Solgaard (DTU – Master student) 14/03–2023

16

"""

B.1.9 cutoff_bound()

```
1      """
2      Procedure to clean and remove divergence effects from numerical
3      differentiation , originating from acceleration calculation from
4      GNSS:
5
6
7      Input:
8          arr: np.array () , [1 , N]
9
10     Output:
11        arr: np.array () , [1 , N-x]
12        where x, is length of the removed values .
13        x: int , number of elements removed from array .
14
15
16     Author: Christian Solgaard (DTU – Master student) 03/03–2023
17     """

```

B.1.10 bias_drift_corr()

```
1      """
2      Procedure to calculate the bias and drift correction of
3      gravity disturbance , derived using the Direcht Strapdown method
4
5
6
7      Input:
8          dg: np.array () , Gravity disturbance array
9          time: np.array () , Time vector [SOW] , (gnss_time)
10         vel: np.array () , Velocity (scalar) , [N,1] , gnss_time
11
12     Output:
13         dg_corr: np.array () , Corrected gravity disturbance .
14
15
16
17     Author: Christian Solgaard (DTU – Master student) 03/03–2023
18     """

```

B.2 Data Loader

This subsection of appendix B, is related to the functions written for reading and converting data from the IMAR IMU, GNSS and navigation file. The full code can be found the the Github Respiratory in the "src" subfolder. All functions is implemented in the python file "IMU_load.py"

B.2.1 readIMAR()

```
1      """
2      Procedure to read and convert the hexadecimal formatted data
3          from the
4      IMAR IMU output file. The converted file , is located at the
5          origin path of the .dat file .
6
7      The readIMAR function uses two different subroutines :
8          readIMAR_Header()
9      which reads and saves important information from the header of
10         the .dat
11         file , and readIMAR_Data() which loads the data based upon the
12             information
13             transferred form the readIMAR_Header() function.
```

```
11     Input :
12         ifile      Path and name of the .dat file
13
14     Optional input arguments :
15         hfunc      Input argument related to optional functions ("echo
16                     ")
16         harg      Argument related to the optional function ("on "/" off
17                     ")
17
18     Output :
19         data      dataframe with columns: "time", "bacc1", "bacc2", "
20                         bacc3".
20         .pkl file: Datafile in the pkl file format , special fileformat
21             for python
21                 Located at the original datafile location.
```

```
24     Author: Christian Solgaard (DTU – Master student) 17/02–2023
25     """

```

B.2.2 load_gnss()

```
1     """
2     Loading function for the ppp gnss file , in .txt format
3
4
5     Input:
6     ifile :           Path and name of file ___.txt
7
8     Output:
9     gnss :           Pandas dataframe format containing the
10    columns:
11    ["lat", "lon", "h", "time"]
12
13    Author: Christian Solgaard (DTU – Master student) 17/02–2023
14    """

```

B.2.3 load_nav()

```
1     """
2     Loading function for the navigation file , in .txt format
3
4
5     Input:
6     ifile :           Path and name of file ___.txt
7
8     Output:
9     nav :            Pandas dataframe format containing the
10    columns:
11    ["lat", "lon", "h", "vn", "ve", "vd", "roll", "
12      pitch", "yaw", "time"]
13     header :         Header information .
14
15    Author: Christian Solgaard (DTU – Master student) 17/02–2023
16    """

```