Navigation Project

Emilio Gigante

LM Ingegneria Robotica e dell'Automazione

Sistemi di Guida e Navigazione

Academic Year 2023/2024

Contents

1 Introduction

This project aims to design an integrated navigation system which can be applicable to an autonomous underwater vehicle which can navigate on the surface of water for short distances.
This project takes inspiration from a previous project developed for the Underwater Systems class of 2024 of University of Pisa in which I participated. That experiment used an AUV Zeno to simulate an obstacle avoidance mission. Given that we recorded data from the experiment, I retrieved those concerning the position and I selected some waypoints, which I slightly modified, to generate a trajectory.

2 Project specifications

I was assigned a code which describes the project specifications.
Code: **D - F - Q - aG - DAM - RO**
Here is the description of these specifications:

- **D**: Decoupled Kalman Filter, one filter for the attitude and one for position and velocity.
- **F**: Flat Earth model for navigation, simplification appropriate for short trajectories.
- **Q**: Quaternions have to be used to estimate the attitude.
- **aG**: The model of accelerometers I am going to use does not include bias instability, on the other hand the gyroscopes model does.
- **DAM**: The attitude filter will use the measurements coming from the accelerometers and a magnetometer.
- **RO**: The position and velocity filter will use the measurements of distances from some fixed points computed through an Ultra-Wideband sensor.

3 Scenario

The navigation scenario that was used to draft this report is defined through a trajectory generated using the Matlab function `waypointTrajectory` to which I provided the waypoints with the respective time of arrival and the sample rate, which was in this case chose as 100 Hz.

Time of arrival [s]	North [m]	East [m]	Down [m]
0.0	9.5	85.8	0.0
48.0	18.4	85.9	0.0
115.4	27.2	79.2	0.0
179.0	28.4	62.9	0.0
254.3	35.6	47.5	0.0
326.7	53.6	33.7	0.0
420.3	67.5	24.9	0.0
480.2	60.6	7.7	0.0

Table 1: List of waypoints used for trajectory generation.

From table 1 the scale of the mission can be acknowledged, making evident that the flat Earth model simplification is suitable for the purpose.
Here we can visualize the trajectory provided from the function.

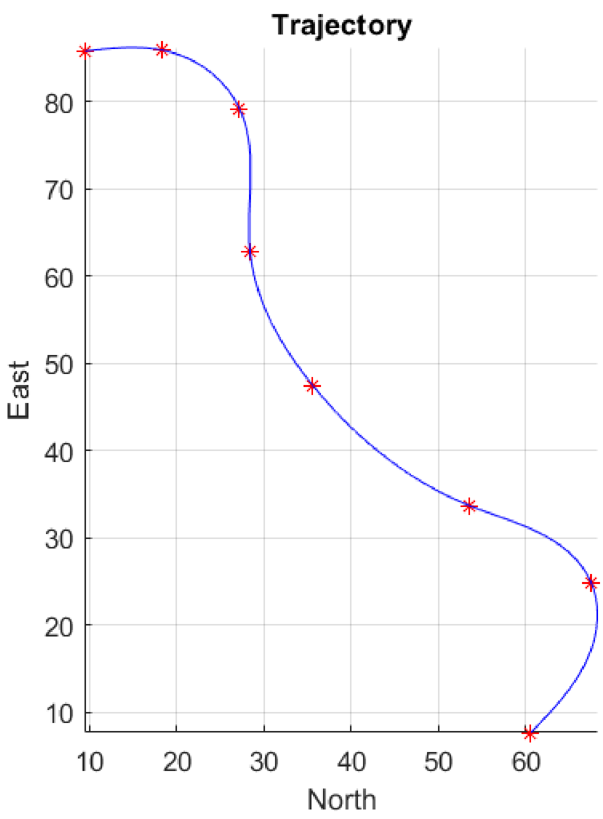

Figure 1: Navigation trajectory

3.1 True navigation variables

The function also provides the development over time of the following navigation variables:

- acceleration in Navigation frame $[m/s^2]$
- velocity in Navigation frame $[m/s]$
- position $[m]$
- angular velocity in Navigation frame $[rad/s]$
- attitude in form of quaternion

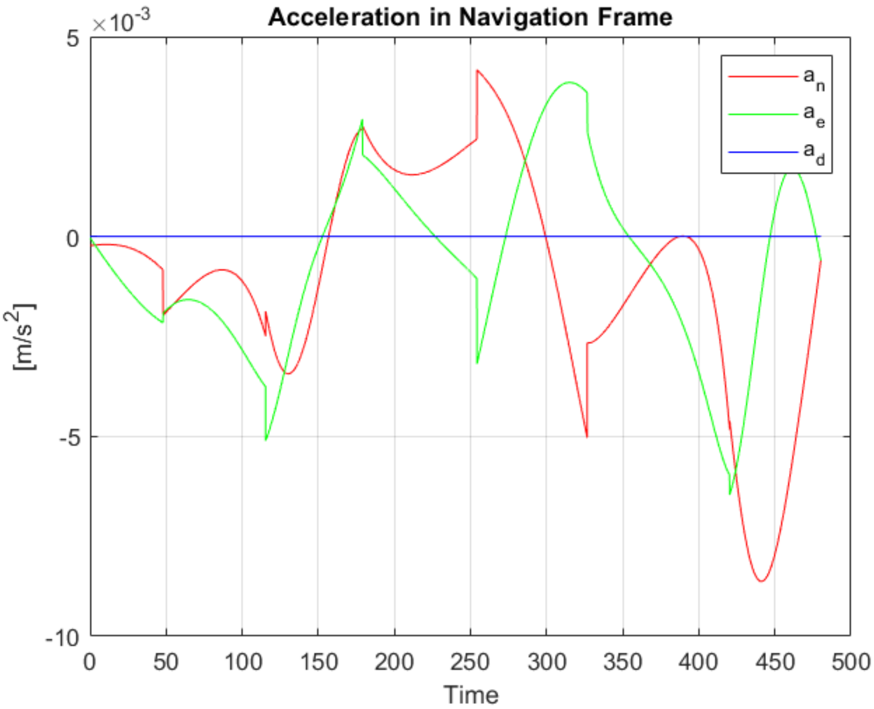

Figure 2: True acceleration in Navigation frame

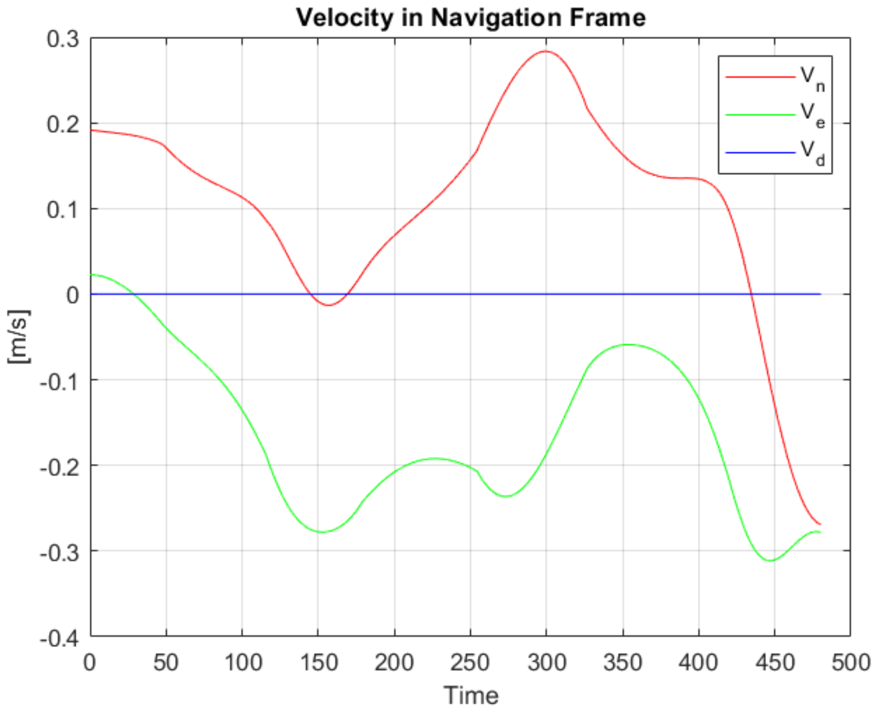

Figure 3: True velocity in Navigation frame

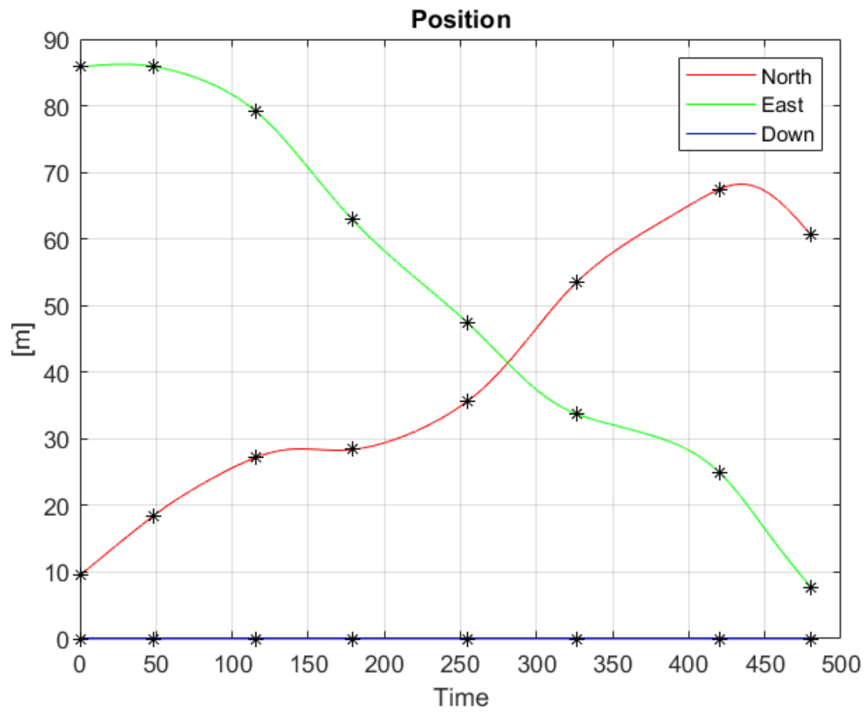

Figure 4: True position

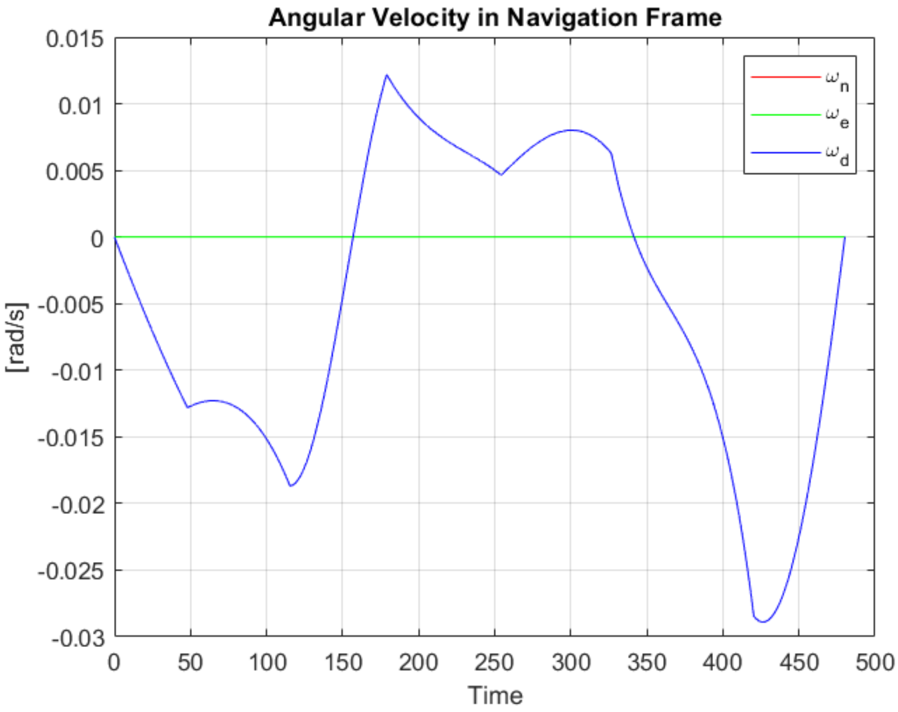

Figure 5: True angular velocity in Navigation frame

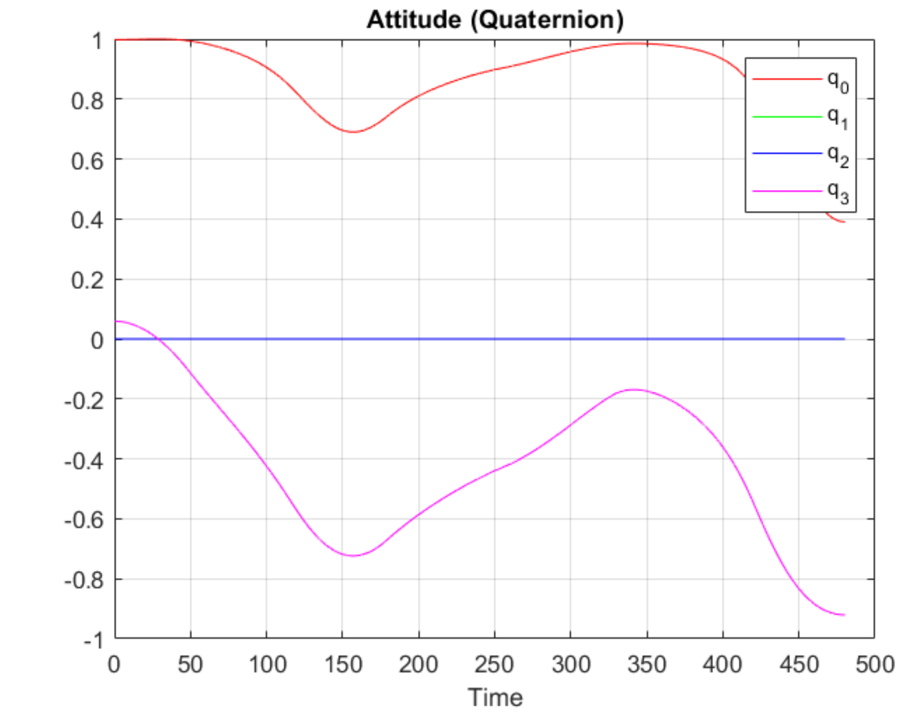

Figure 6: True attitude

The consistency with position, velocity and acceleration was hence verified recalculating position from both velocity and acceleration.

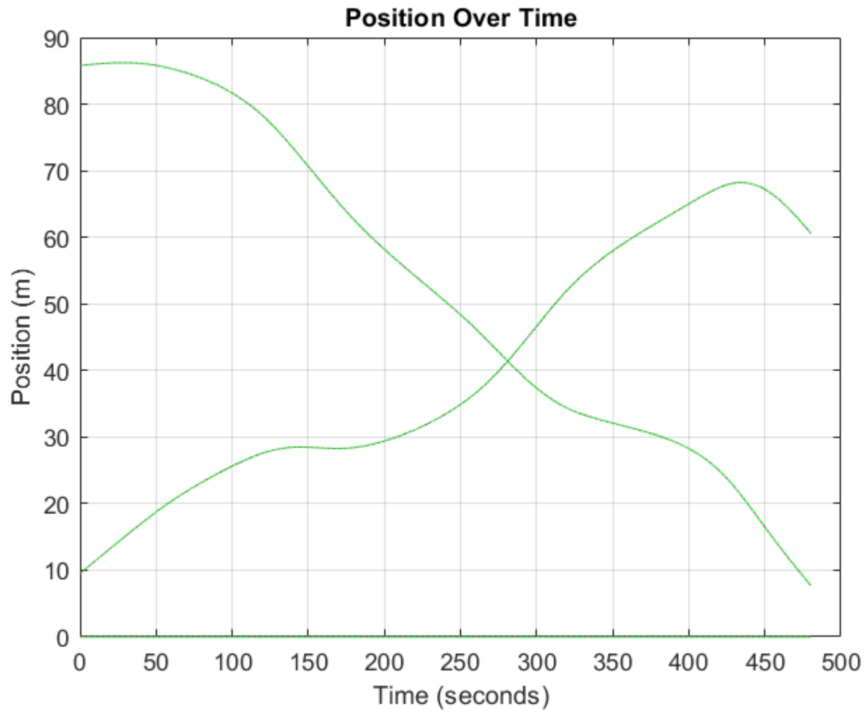

Figure 7: position-velocity-acceleration consistency

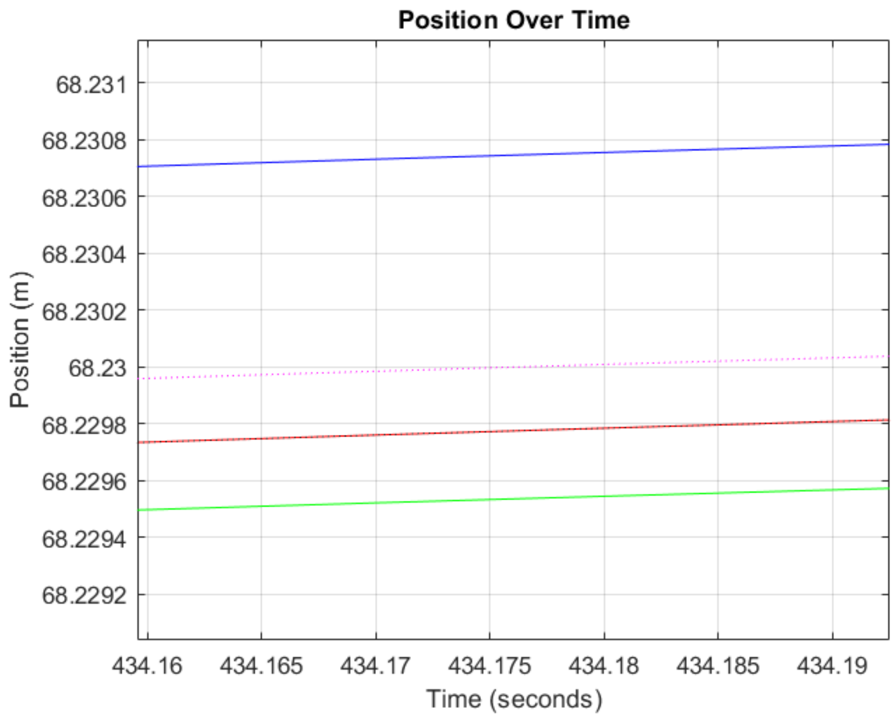

Figure 8: position-velocity-acceleration consistency

3.2 Acceleration filtering

In order to preserve the actuators of the vehicle is useful to smooth the acceleration seen in figure 2. In this case a low pass filter with 1 Hz bandwidth was used, resulting in a smoother acceleration, from which velocity and position have been derived from, these filtered variables are the ones that will be used in the following sections.

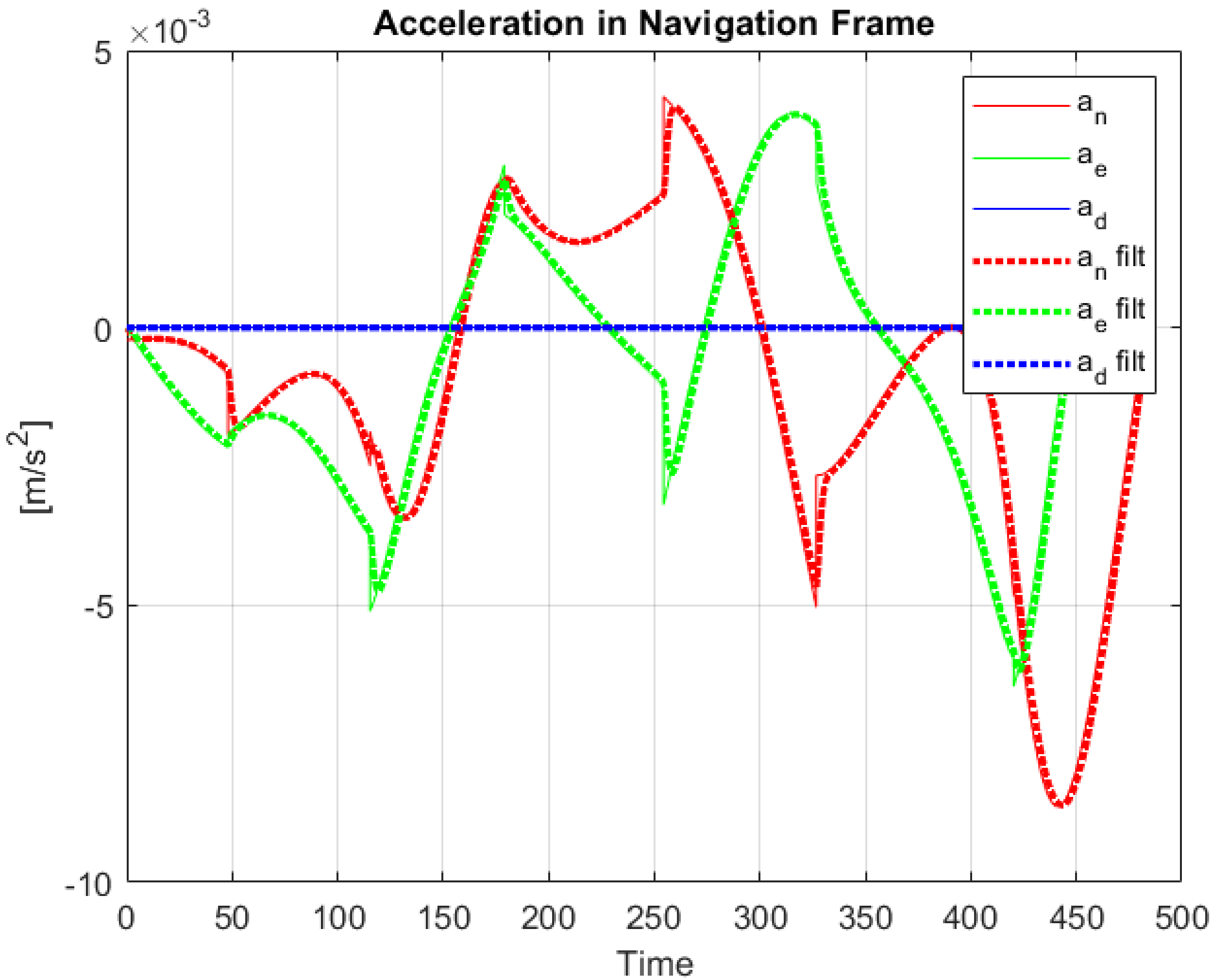

Figure 9: Filtered acceleration in Navigation frame

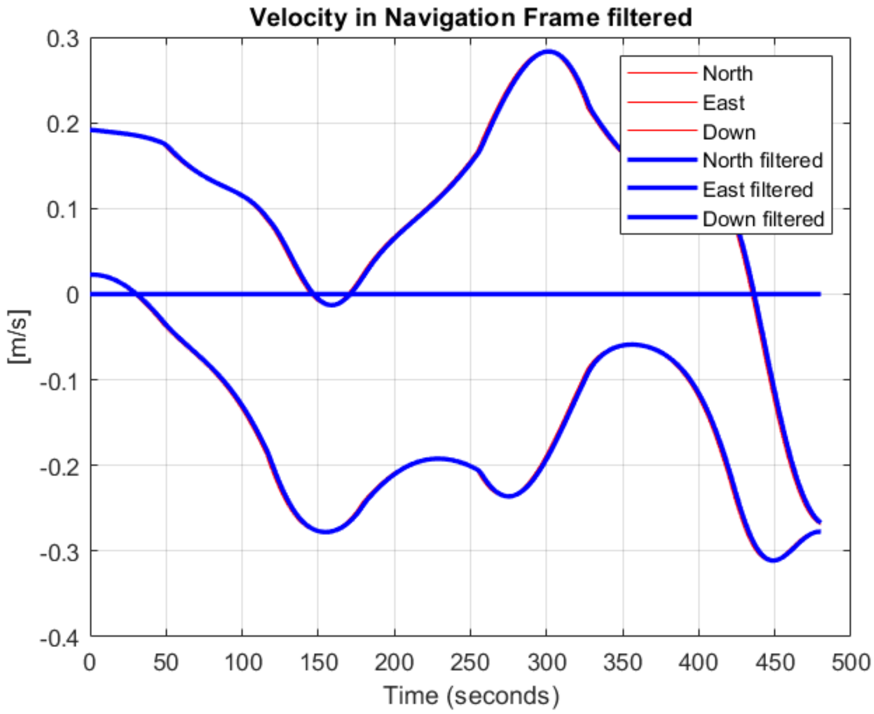

Figure 10: Filtered velocity in Navigation frame

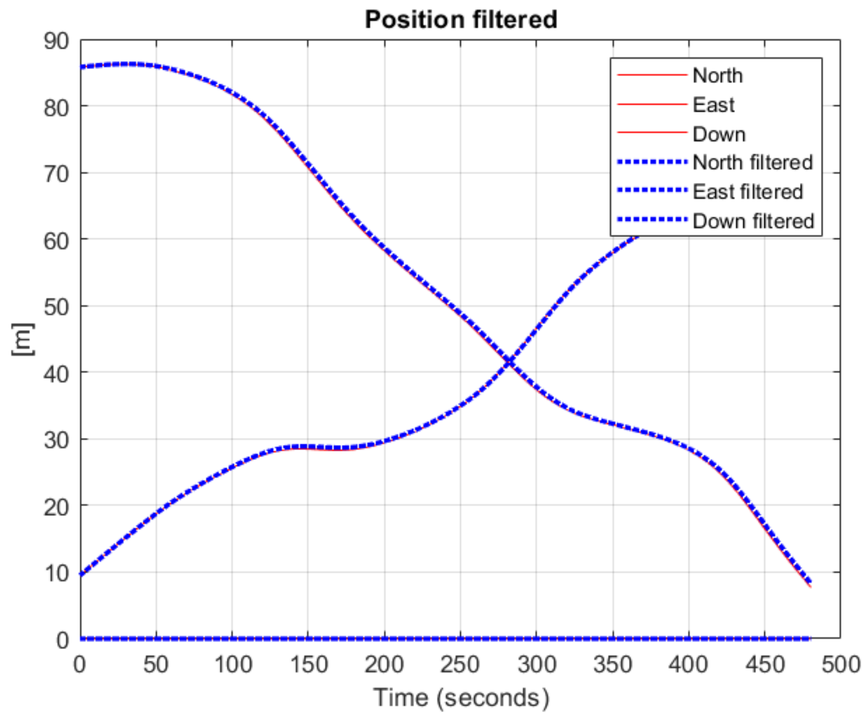

Figure 11: Filtered position

4 Sensors

I assume there is a strapdown platform on board which contains all the sensors.
There is a description of how the sensor used for the navigation system were modeled:

- **Accelerometers**: I assume I have 3 accelerometers, each one have its sensitive axis on a Body frame axis of the vehicle, for simplification the measurements of each accelerometer are going to be the elements of a 3×1 vector.
  This vector is modeled subtracting the gravity force g_n to the true acceleration in Navigation frame acc_n in order to obtain the specific force in Navigation frame f_n, then this is rotated through the rotation matrix from Navigation to Body frame C_n^b which is calculated from the true attitude, this way we obtain a clean specific force in Body frame, then, according to a Analog Devices ADXL354 datasheet (frequency of work it is assumed to be $100\,Hz$) is added the noise, modeled as white, with a variance of $1.384 \times 10^{-6}\,(m/s^2)^2$ on every component. This way we have a noisy f_b which serves as the output.
  Bias instability is not considered in the model.
  Constant $\boldsymbol{g}_n = [0 \quad 0 \quad 9.8022]^T\,m/s^2$ is obtained from the WGS-84 model at latitude 40.61043876911262 and longitude 9.756915512868472 and at the sea level $(0\,m)$, which is assumed to be the origin of the Navigation frame.

- **Gyroscopes**: I assume I have 3 gyroscopes, each one have its sensitive axis on a Body frame axis of the vehicle, for simplification the measurements of each gyroscope are going to be the elements of a 3×1 vector.
  The model of this vector uses the true angular velocity in Navigation frame ω_n and rotates it to the Body frame to ω_b similarly to how was done previously with the accelerometers. Then according to a Bosch BMG250 datasheet working at $100\,Hz$ it adds a constant bias with the value of $0.5°/s$ $(0.0087\,rad/s)$ and a white noise with a variance of $5.97 \times 10^{-6}\,(rad/s)^2$ on every component.

- **Magnetometer**: This model uses the magnetic field vector in Navigation frame $\boldsymbol{B}_n = [25.35 \quad 1.28 \quad 38.32]^T\,\mu T$ obtained through the WMM2020 model using the coordinate of the Navigation frame mentioned before, it rotates it to Body frame as previously described, and adds a white noise on every component which has a variance of $0.08\,(\mu T)^2$ according to a Honeywell HMC5883L datasheet working at $50\,Hz$.

- **Ultra-Wideband sensor**: The model assumes that there are 6 antennas in the following position of the Navigation frame in meters:

$$pa_1 = [0 \quad 0 \quad 0] \quad pa_2 = [70 \quad 2 \quad 1]$$

$$pa_3 = [72 \quad 50 \quad 0] \quad pa_4 = [68 \quad 100 \quad 1]$$

$$pa_5 = [2 \quad 102 \quad 0] \quad pa_6 = [-2 \quad 48 \quad 1]$$

  These antenna transmit to the sensor on board of the vehicle which can compute the distances from each one of the antennas.
  The model computes the distances from the true position of the vehicle and then adds on every distance a white noise with a variance of $0.01\,m^2$ according to a Decawave DW1000 datasheet, working sub-sampled at $100\,Hz$.

5 Attitude estimation

In order to obtain an attitude estimation an extended Kalman filter is used.
This section describes how the filter was modeled.

- **state**: The state is composed by the quaternion $\boldsymbol{q} = [q_0 \quad q_1 \quad q_2 \quad q_3]$ and the gyroscopes bias $\boldsymbol{\omega}_{bias}$, so the state is

$$\boldsymbol{x} = [q_0 \quad q_1 \quad q_2 \quad q_3 \quad \omega_{bias,x} \quad \omega_{bias,y} \quad \omega_{bias,z}]^T$$

- **input**: the inputs of the systems are the measurements of the gyroscopes [4]

$$\boldsymbol{\omega}_m = [\omega_{m,x} \quad \omega_{m,y} \quad \omega_{m,z}]^T$$

- **process noise**: I consider the noise of the process to be attributed just to the the gyroscopes measurements.

$$\boldsymbol{\omega}_m = \bar{\boldsymbol{\omega}}_m + \boldsymbol{w}_{\omega_m}$$

- **measurements**: The measurements available are the ones from the accelerometers [4] and the magnetometer [4]

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_{acc} \\ \boldsymbol{y}_{mag} \end{bmatrix}$$

- **sample time**: sample time of the simulation $\Delta t = 0.01s$

Simple mechanization of the attitude does not include the bias model so its equations are equivalent to the dynamics of the quaternion

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{\omega}_m \quad [1]$$

Which I discretized for a better comparison with the EKF results, so the equations are

$$\boldsymbol{q}_{k+1|k} = \boldsymbol{q}_{k|k} \otimes \boldsymbol{q}\{(\boldsymbol{\omega}_{m,k}\Delta t\} \quad [2]$$

The evolution of the attitude resulting from simple mechanization (in roll, pitch and yaw form) with respect to the true values is shown in figure 12.

5.1 Prediction

For the filter we have the same equations but we have to consider the bias

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes (\boldsymbol{\omega}_m - \boldsymbol{\omega}_{bias}) \\ \dot{\boldsymbol{\omega}}_{bias} = 0 \end{bmatrix}$$

The filter model requires the discretization of the equations

$$\boldsymbol{x}_{k+1|k} = \begin{bmatrix} \boldsymbol{q}_{k+1|k} = \boldsymbol{q}_{k|k} \otimes \boldsymbol{q}\{(\boldsymbol{\omega}_{m,k} - \omega_{bias,k})\Delta t\} \\ \boldsymbol{\omega}_{bias,k+1|k} = \boldsymbol{\omega}_{bias,k} \end{bmatrix} = \boldsymbol{f}_k$$

the state is updated through the discretized dynamics, but in order to update the P matrix it's required to derive the $\boldsymbol{f}_k$ to get the matrices F and D.

$$F_k = \left.\frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{x}_k}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} \qquad D_k = \left.\frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{w}_k}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} \quad [3]$$

The Q matrix was in this case taken as proportional with an empirical tuning parameter to the variance of the gyroscopes white noise

$$Q = a \times \sigma^2_{gyro} = 5 \times 5.97 \times 10^{-6} = 2.98 \times 10^{-5}$$

So the prediction formulas are

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{f}_k \qquad P_{k+1|k} = F_k P_{k|k} F_k^T + D_k Q D_k^T$$

[1] ' $\otimes$ ' indicates the quaternion product.
[2] $q\{\boldsymbol{v}\} = [\cos v/2 \quad \hat{\boldsymbol{v}} \times \sin v/2]$
[3] Given the complexity of the quaternion composition it was necessary to create this Jacobians matrices through symbolic calculus in Matlab.

5.2 Correction

As briefly mentioned before the filter receives the measurements from the accelerometers and the magnetometer, the model of the measurements is

$$\boldsymbol{h}_k = \begin{bmatrix} -C_n^b\{\boldsymbol{q}_{k|k-1}\}\boldsymbol{g}_n \\ C_n^b\{\boldsymbol{q}_{k|k-1}\}\boldsymbol{B}_n \end{bmatrix}$$

It's worth highlighting that this model of the accelerometers is suitable just because the accelerations of the vehicle are significantly lower with respect to gravity acceleration.
The rotation matrix from Navigation to Body C_n^b is calculated from the quaternion as

$$C_n^b = (q_w^2 - \boldsymbol{q}_v^T\boldsymbol{q}_v)I_3 + 2\boldsymbol{q}_v\boldsymbol{q}_v^T + 2q_w[\boldsymbol{q}_v]_\times$$

$$where \quad \boldsymbol{q} = \begin{bmatrix} qw \\ \boldsymbol{q}_v \end{bmatrix} \quad and \quad [\boldsymbol{a}]_\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}$$

Then we need to define the measurements noise matrix

$$R = \begin{bmatrix} \sigma_{acc}^2\, I_3 & 0_{3\times3} \\ 0_{3\times3} & \sigma_{mag}^2\, I_3 \end{bmatrix} = \begin{bmatrix} 1.384\times10^{-6}\, I_3 & 0_{3\times3} \\ 0_{3\times3} & 0.08\, I_3 \end{bmatrix}$$

At this point we proceed calculating the elements necessary for the correction

$$\boldsymbol{e} = \boldsymbol{y}_k - \boldsymbol{h}_k$$

$$H_k = \left.\frac{\partial \boldsymbol{h}_k}{\partial \boldsymbol{x}_k}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} [4] \qquad S_k = H_k P_{k|k-1} H_k^T + R \qquad L_k = P_{k|k-1} H_k^T S_k^{-1}$$

So the actual correction is

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + L_k\boldsymbol{e}_k$$

$$P_{k|k} = (I_6 - L_kH_k)P_{k|k-1}(I_6 - L_kH_k)^T + L_kRL_k^T$$

5.3 Initial conditions

I assume to know the average of initial orientation of the vehicle exactly but with a standard deviation of 0.0175, and I impose the initial bias average to being null given that I have no information about it, and with a $1°/s$ ($0.0175\, rad/s$) standard deviation, so in formulas

$$\boldsymbol{x}_0 = [0.9983 \quad 0 \quad 0 \quad 0.0590 \quad 0 \quad 0 \quad 0]^T$$

$$P_0 = (0.0175)^2 I_6$$

5.4 Osservability

I made a rough osservability check buildind a pseudo osservability matrix as

$$\tilde{\mathcal{O}} = \begin{bmatrix} H \\ H\,F \end{bmatrix}$$

Through symbolic calculus (it is shown in the files provided for the simulation) we can see that we have full rank of the pseudo osservability matrix, so we expect to have good results, and since we have redundant measure (since in the model just C_n^b is important) we are going to be able to estimate bias properly too.

[4] Given the complexity of the matrix in relation to the state it was necessary to create this Jacobians matrices through symbolic calculus in Matlab.

5.5 Estimation results

The effectiveness of the filter results evident if we compare the estimated quaternion with the simple mechanization laws of the attitude which does not take into consideration the bias correction. On the figure 12 we can see the evolution of roll, pitch and yaw according to their true value, their mechanized value and their estimated value. On figure 13 we can see the difference between the error of simple mechanization and of estimated attitude with respect to the true attitude, on the second graphic it is highlighted the mean squared error (MSE) for the estimated attitude.

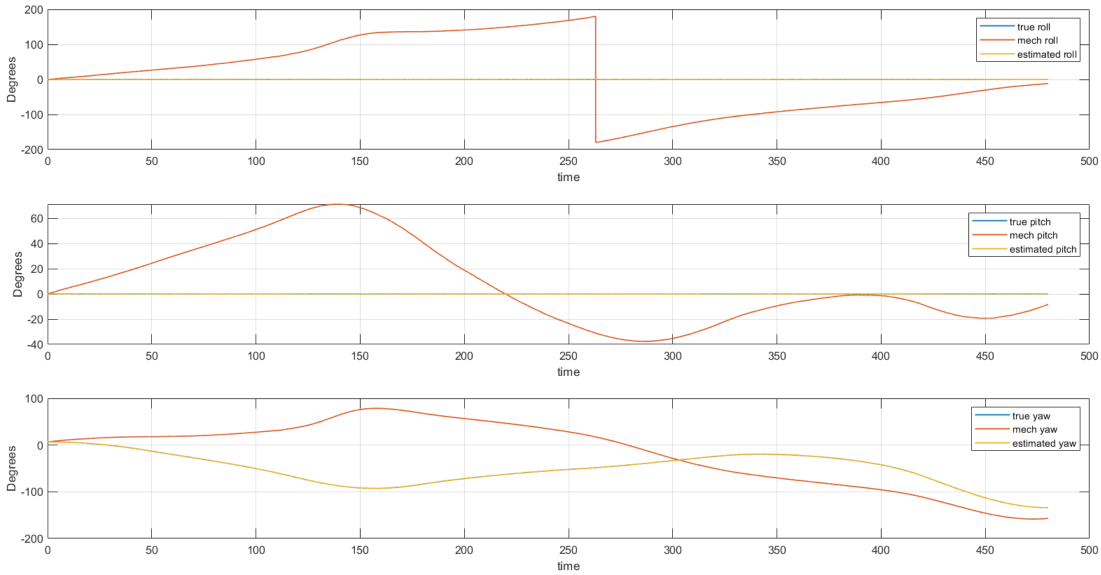

Figure 12: Attitude mechanization and estimation in comparison

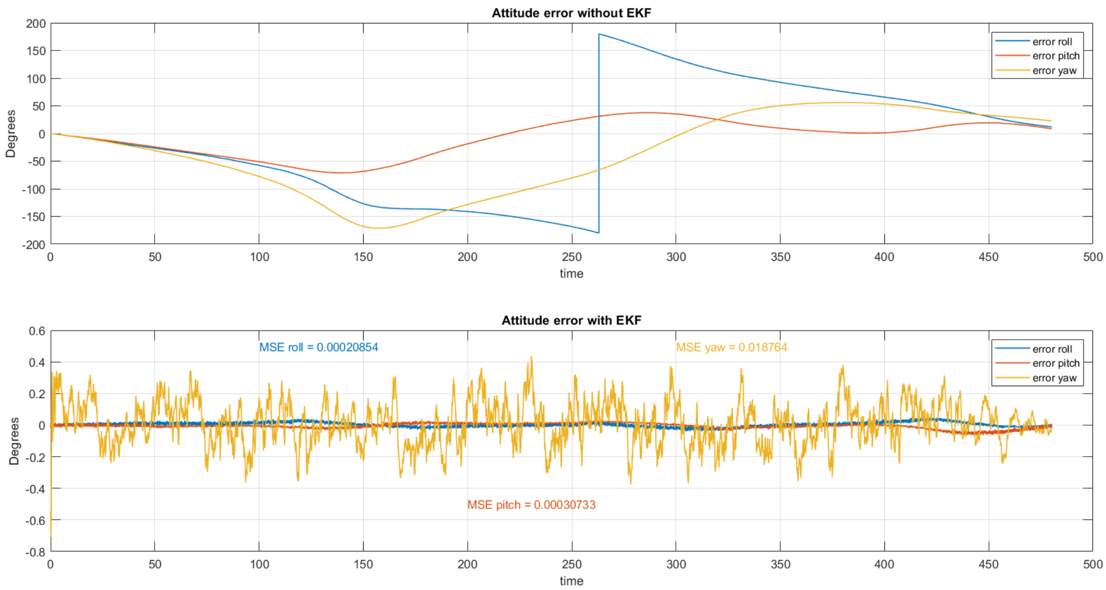

Figure 13: Attitude error from mechanization and estimation in comparison

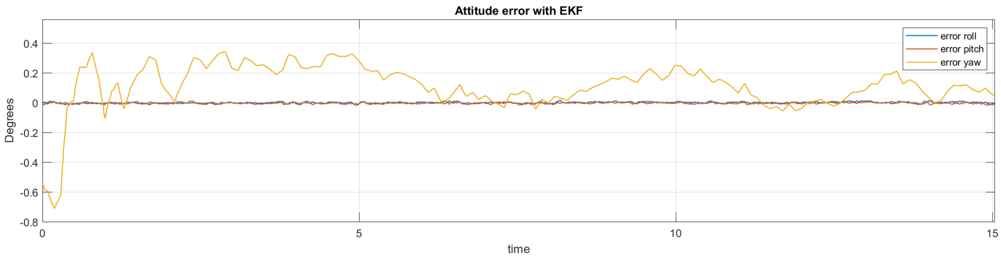

Figure 14: Attitude transitory, lasts around 1 s.

It's also useful to see that the bias was properly estimated

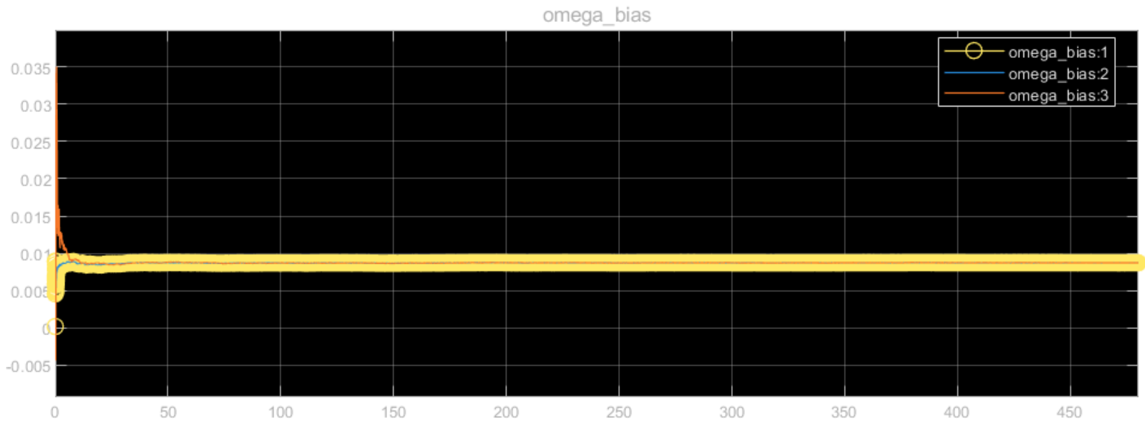

Figure 15: ω_{bias} estimation [m/s]

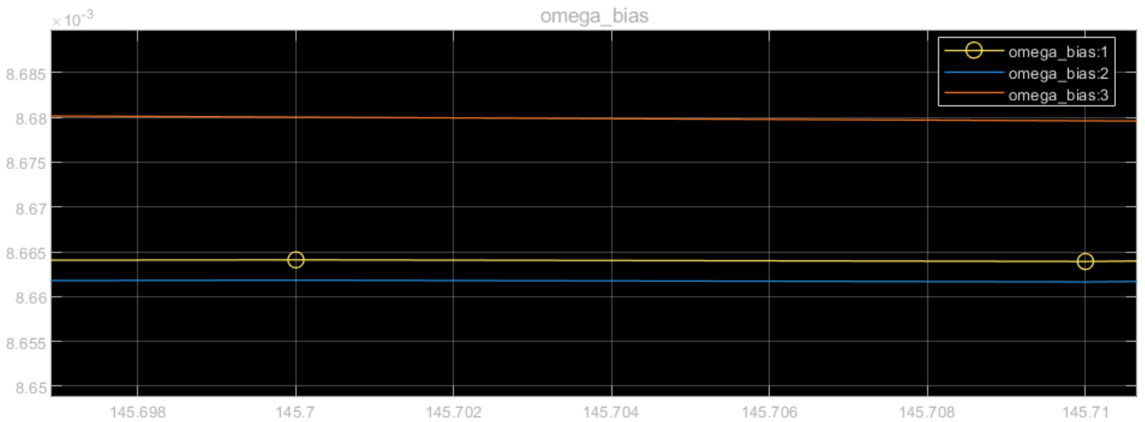

Figure 16: ω_{bias} estimation up close [m/s]

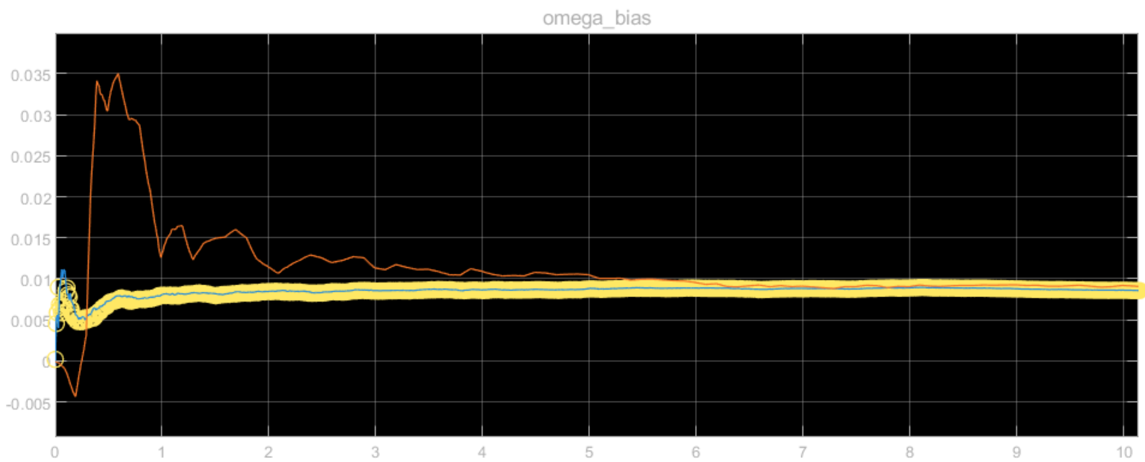

Figure 17: ω_{bias} transitory, lasts around 6 s.

Here is the computation of the mean squared error of the state with respect to the true values in order to have the quantification of the entity of the correction

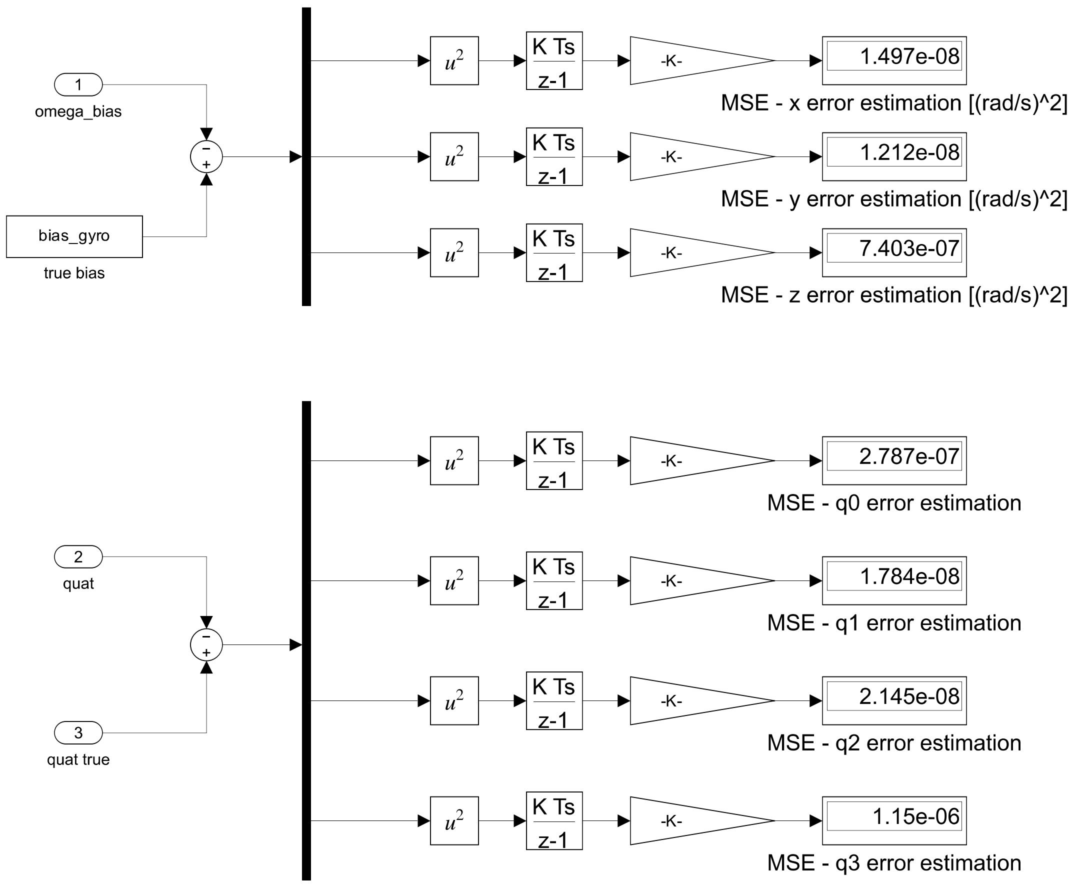

Figure 18: Mean Squared Error of the attitude and the bias

It's also important to verify that the filter work results in a better estimation with respect to the measurements. So here is the comparison between the measurements and the equivalent value obtained with the sensor model (except for the noise) but with the true and estimated attitude, this is done both for the magnetometer and the accelerometers.

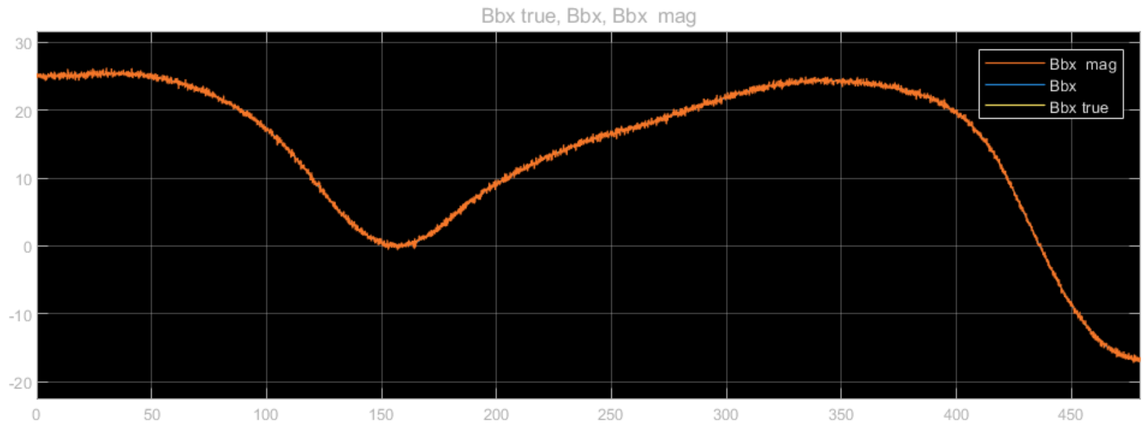

Figure 19: $B_{b,x}$ comparison $[\mu T]$.

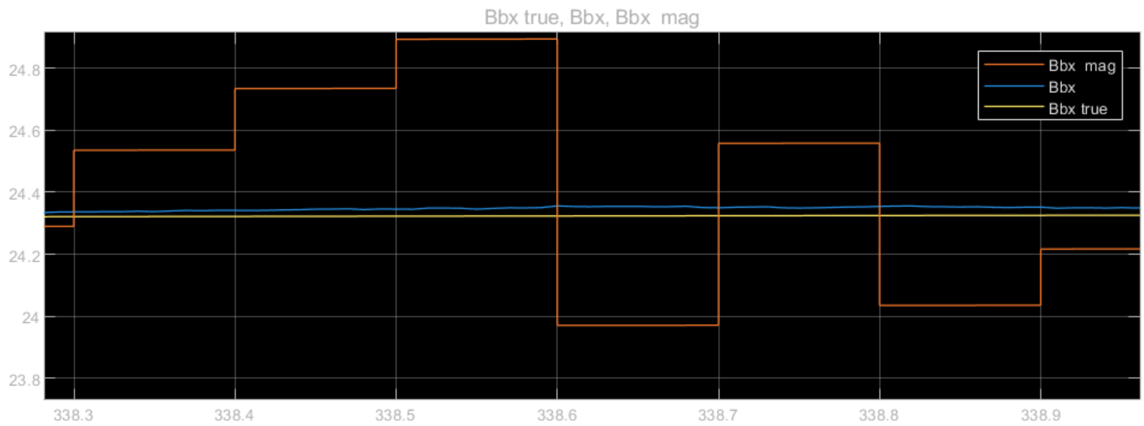

Figure 20: $B_{b,x}$ comparison up close $[\mu T]$.

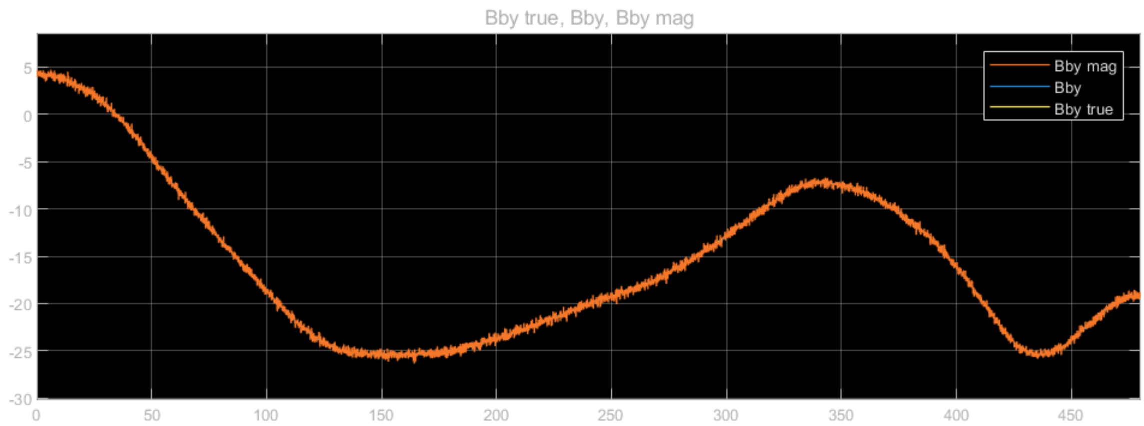

Figure 21: $B_{b,y}$ comparison $[\mu T]$.

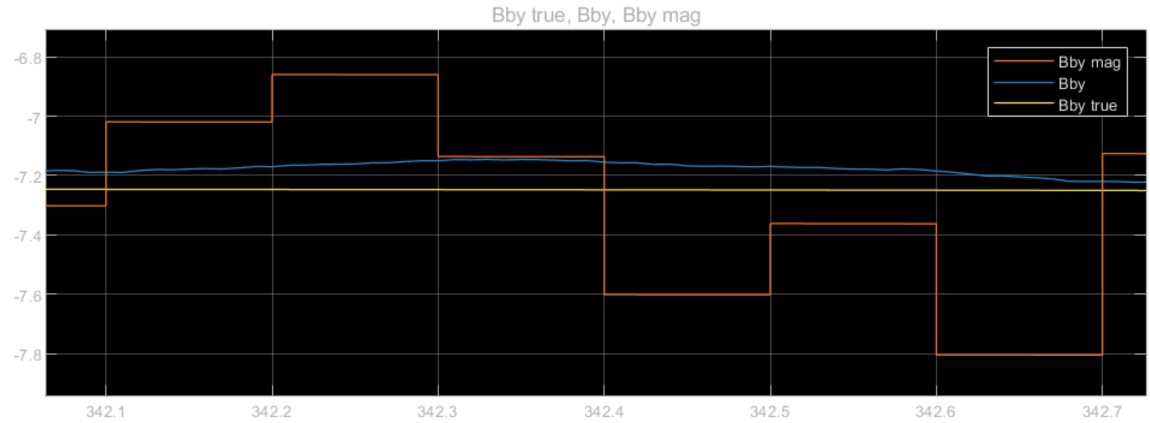

Figure 22: $B_{b,y}$ comparison up close $[\mu T]$.

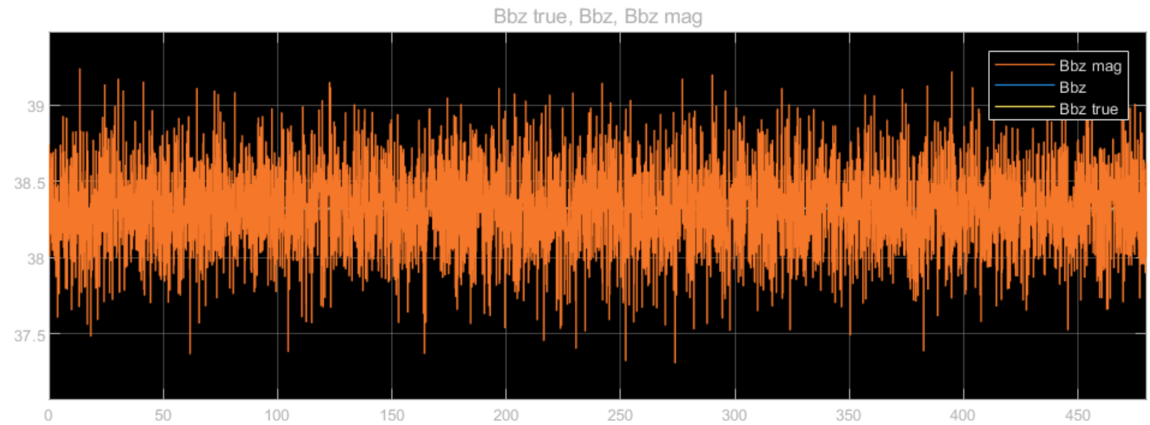

Figure 23: $B_{b,z}$ comparison $[\mu T]$.

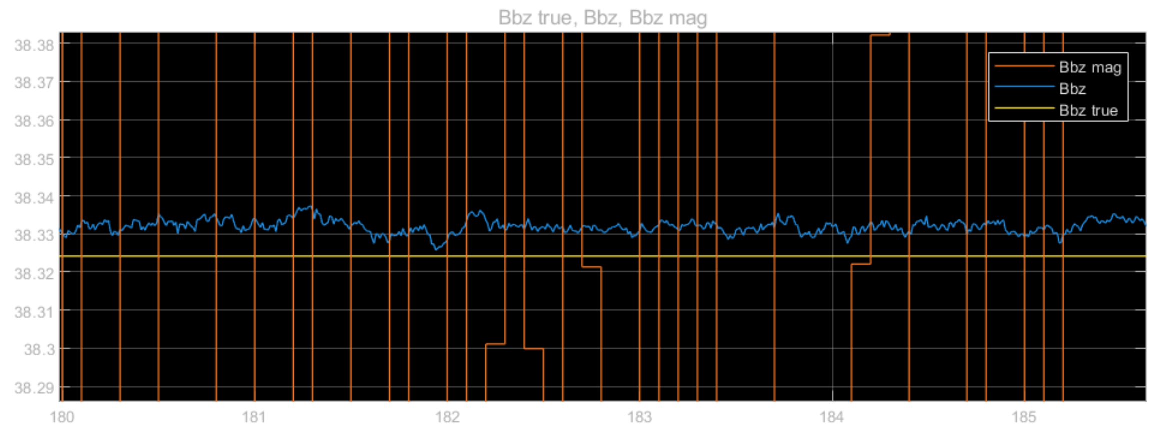

Figure 24: $B_{b,z}$ comparison up close $[\mu T]$.

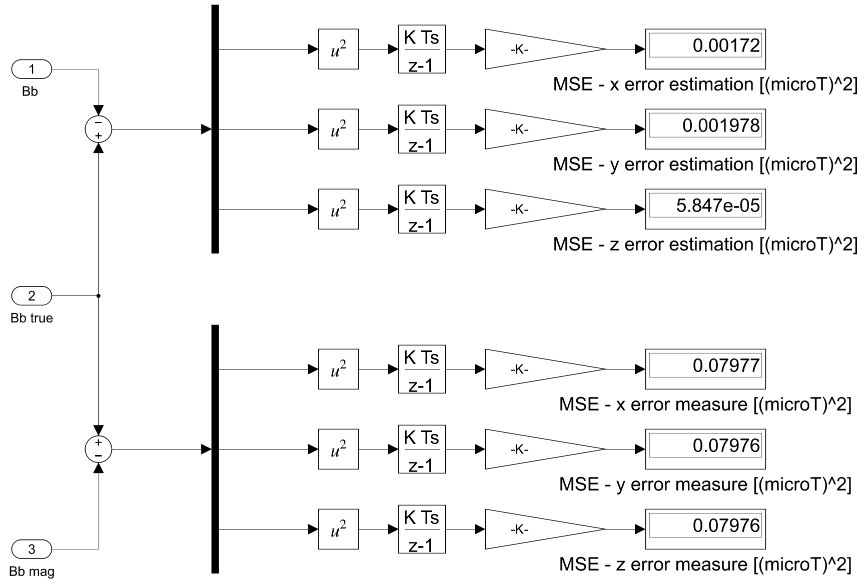

Figure 25: We can see how the MSE of the estimation dropped to around 2% of the MSE of the measurement for the first 2 components and to around 0.08% for the third .

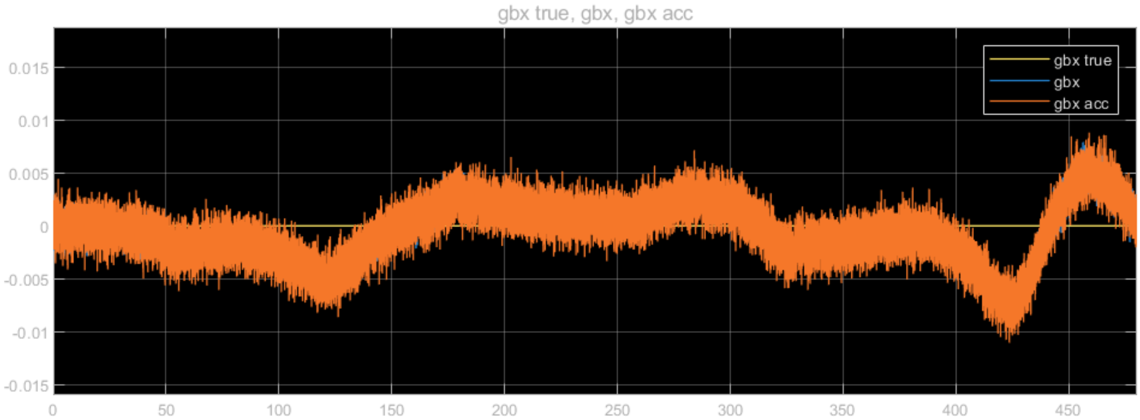

Figure 26: $g_{b,x}$ comparison $[m/s^2]$.

Figure 27: $g_{b,x}$ comparison up close $[m/s^2]$.

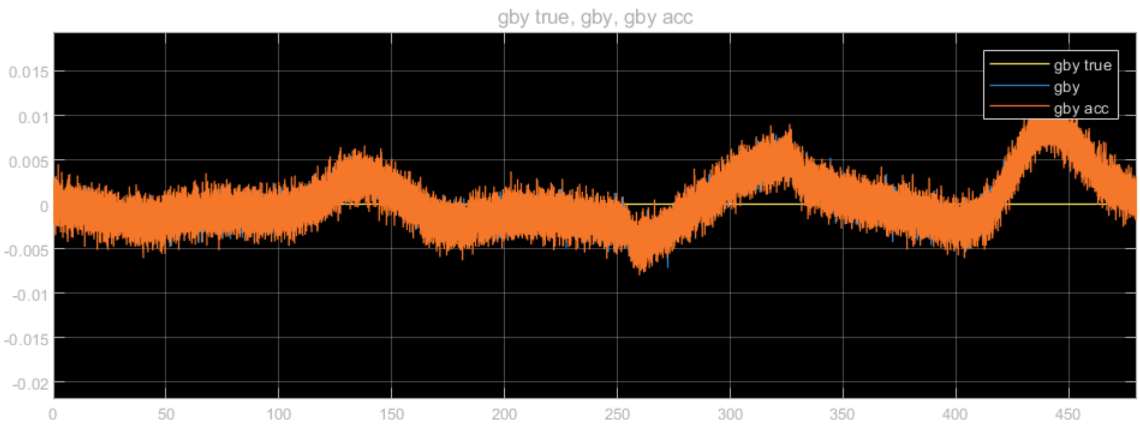

Figure 28: $g_{b,y}$ comparison $[m/s^2]$.

Figure 29: $g_{b,y}$ comparison up close $[m/s^2]$.

Figure 30: $g_{b,z}$ comparison $[m/s^2]$.

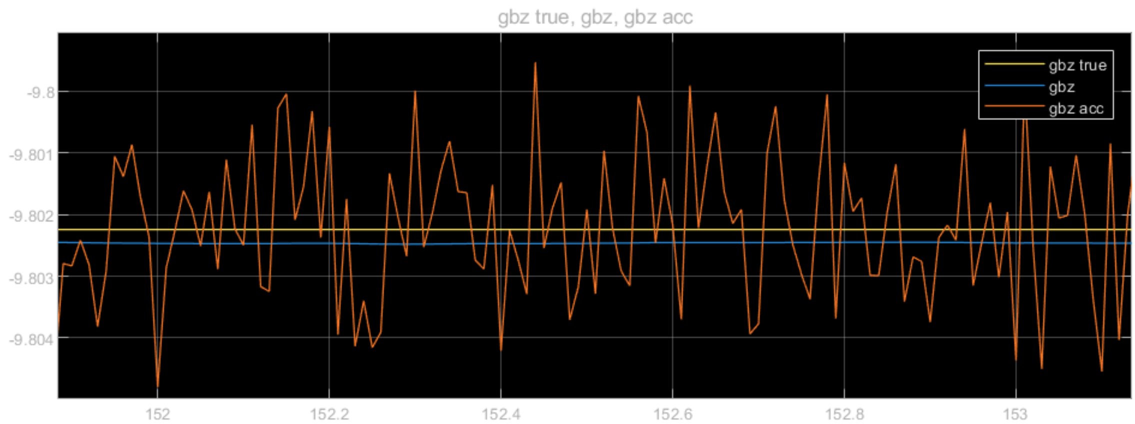

Figure 31: $g_{b,z}$ comparison up close $[m/s^2]$.

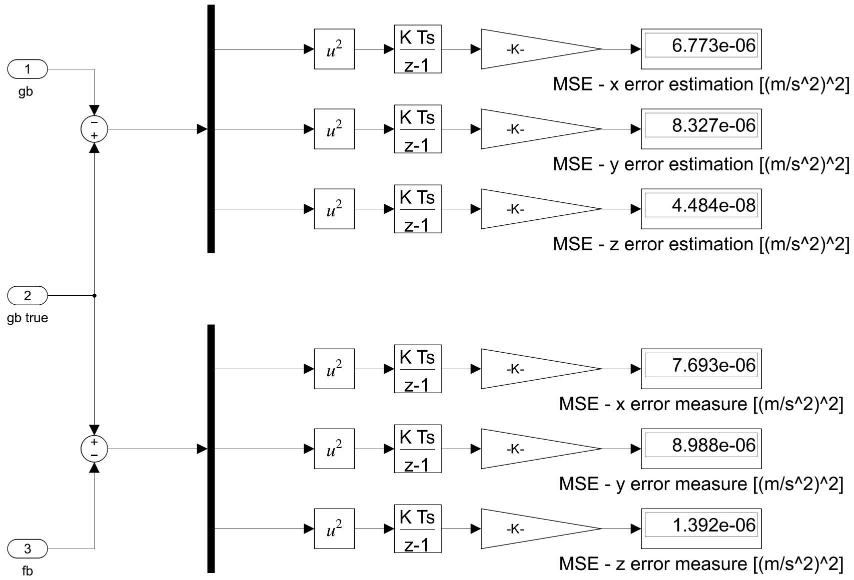

Figure 32: Due to the imperfection of the model the MSE of the estimation just slightly decreased to around 88% of the MSE of the measurement for the first component and to around 92% for the second, on the other hand, advantaged by the greater value, the tird component MSE dropped to around 3% of the MSE of the measurement.

6 Position and Velocity estimation

Now that we made sure that we have a good attitude estimation we can use that to build an extended Kalman filter for position and velocity estimation.
Here is the description of the filter model.

- **state**: The state is composed by the position $\boldsymbol{p} = [p_N \quad p_E \quad p_D]^T$ and the velocity in Navigation frame (the subscript indicating the frame will from now on be withheld) $\boldsymbol{v} = [v_N \quad v_E \quad v_D]^T$. So the state is
$$\boldsymbol{x} = [p_N \quad p_E \quad p_D \quad v_N \quad v_E \quad v_D]^T$$
- **input**: the inputs of the systems are the measurements of the accelerometers [4]
$$\boldsymbol{f}_m = [f_{m,x} \quad f_{m,y} \quad f_{m,z}]^T$$
and the rotation matrix from Body to Navigation C_b^n computed from the attitude estimated by the attitude filter.
- **process noise**: I consider the noise of the process to be attributed just to the the accelerometers measurements.
$$\boldsymbol{f}_m = \bar{\boldsymbol{f}}_m + \boldsymbol{w}_{f_m}$$
- **measurements**: The measurements available are the distances from the antennas computed by the Ultra-Wideband sensor. [4]
$$\boldsymbol{y} = [d_1 \quad d_2 \quad d_3 \quad d_4 \quad d_5 \quad d_6]^T$$
- **sample time**: sample time of the simulation $\Delta t = 0.01s$

Since we are using a flat Earth model the mechanization equations of position and velocity are very simple
$$\dot{\boldsymbol{p}} = \boldsymbol{v}$$
$$\dot{\boldsymbol{v}} = C_b^n \boldsymbol{f}_m + \boldsymbol{g}_n$$
Which I discretized for better comparison with the filter results
$$\boldsymbol{p}_{k+1|k} = \boldsymbol{p}_{k|k} + \boldsymbol{v}_{k|k}\Delta t$$
$$\boldsymbol{v}_{k+1|k} = \boldsymbol{v}_{k|k} + \left(C_b^n \boldsymbol{f}_m + \boldsymbol{g}_n\right)\Delta t$$
The evolution of the mechanized position with respect to its true value is shown in figure 33, same goes for velocity in figure 36.

6.1 Prediction

Resuming the discretized mechanization equations
$$\boldsymbol{p}_{k+1|k} = \boldsymbol{p}_{k|k} + \boldsymbol{v}_{k|k}\Delta t = \boldsymbol{f}_{1,k}$$
$$\boldsymbol{v}_{k+1|k} = \boldsymbol{v}_{k|k} + \left(C_b^n \boldsymbol{f}_m + \boldsymbol{g}_n\right)\Delta t = \boldsymbol{f}_{2,k}$$
These equations leads also to the simple Jacobian matrices necessary for P updating
$$F_k = \left.\frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{x}_k}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \begin{bmatrix} I_3 & I_3\Delta t \\ 0_{3\times 3} & I_3 \end{bmatrix} \qquad D_k = \left.\frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{w}_k}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \begin{bmatrix} 0_{3\times 3} \\ C_b^n \Delta t \end{bmatrix}$$
The process noise matrix Q (in this case a scalar) was chosen proportional to the variance of the accelerometers white noise with an empirical tuning parameter a
$$Q = a \times \sigma_{acc}^2 = 10^4 \times 1.386 \times 10^{-6} = 0.01386$$
So, finally, the prediction equations are
$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{f}_k \qquad P_{k+1|k} = F_k P_{k|k} F_k^T + D_k Q D_k^T$$

6.2 Correction

The measurement can easily be modeled with the geometric distance between the position at the given step and each of the 6 antennas.
Indicating the i-th antenna's position as $\boldsymbol{pa}_i = [pa_{i,N} \quad pa_{i,E} \quad pa_{i,D}]^T$

$$\boldsymbol{h}_{i,k} = \sqrt{(p_{N,k|k-1} - pa_{i,N})^2 + (p_{E,k|k-1} - pa_{i,E})^2 + (p_{D,k|k-1} - pa_{i,D})^2} \quad for \quad i = 1,2,3,4,5,6$$

After that we obtain the Jacobian matrix H

$$H_{i,k} = \left.\frac{\partial \boldsymbol{h}_{i,k}}{\partial \boldsymbol{x}_k}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} = \begin{bmatrix} \frac{p_{N,k|k-1} - pa_{i,N}}{h_{i,k|k-1}} & \frac{p_{E,k|k-1} - pa_{i,E}}{h_{i,k|k-1}} & \frac{p_{D,k|k-1} - pa_{i,D}}{h_{i,k|k-1}} & 0 & 0 & 0 \end{bmatrix} \quad for \quad i = 1, ..., 6$$

At this point we proceed calculating the other elements necessary for the correction

$$\boldsymbol{e} = \boldsymbol{y}_k - \boldsymbol{h}_k \qquad S_k = H_k P_{k|k-1} H_k^T + R \qquad L_k = P_{k|k-1} H_k^T S_k^{-1}$$

So the actual correction is

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + L_k \boldsymbol{e}_k$$

$$P_{k|k} = (I_6 - L_k H_k) P_{k|k-1} (I_6 - L_k H_k)^T + L_k R L_k^T$$

6.3 Initial conditions

I assume to know exactly the average of the initial position, also given that is my first waypoint, but with a standard deviation of 1 m. For the velocity I make the assumption that the vehicle is still at first so I assume an initial null average but with a 0.5 m/s standard deviation.

$$\boldsymbol{x}_0 = [9.5 \quad 85.8 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$P_0 = \begin{bmatrix} I_3 & 0_{3\times3} \\ 0_{3\times3} & 0.25\, I_3 \end{bmatrix}$$

6.4 Osservability

I made a rough osservability check buildind a pseudo osservability matrix as

$$\tilde{\mathcal{O}} = \begin{bmatrix} H \\ H\,F \end{bmatrix}$$

Through symbolic calculus (it is shown in the files provided for the simulation) we can see that we have full rank of the pseudo osservability matrix, so even though we do not have direct measurements of velocity we expect good results.

6.5 Estimation Results

In order to understand the work that this filter does I am now going to show the comparison between the 6 variables (position and velocity components) estimated by the filter and the simple mechanization which takes in input the same C_b^n matrix which is the one computed from the attitude filter estimation (the mecanized variable, though, have the exact initial conditions for both position and velocity).
We can see the velocity correction works even though we do not have direct measurements.
Figure 33 shows the components comparison, while figure 34 shows errors with respect to the true values. Likewise with velocity in figures 36 and 37.

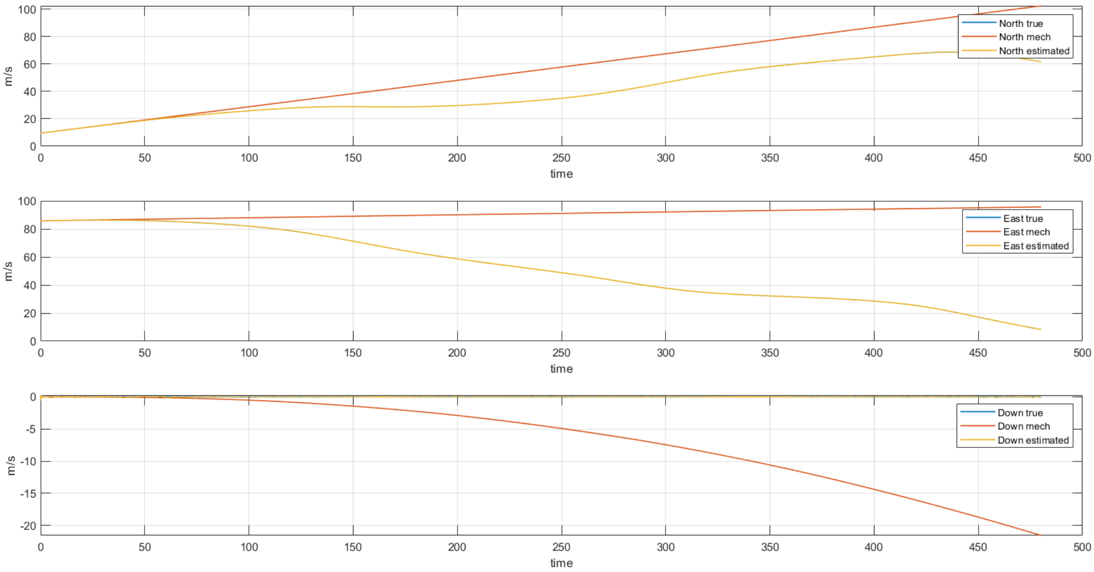

Figure 33: Position from mechanization and estimation in comparison

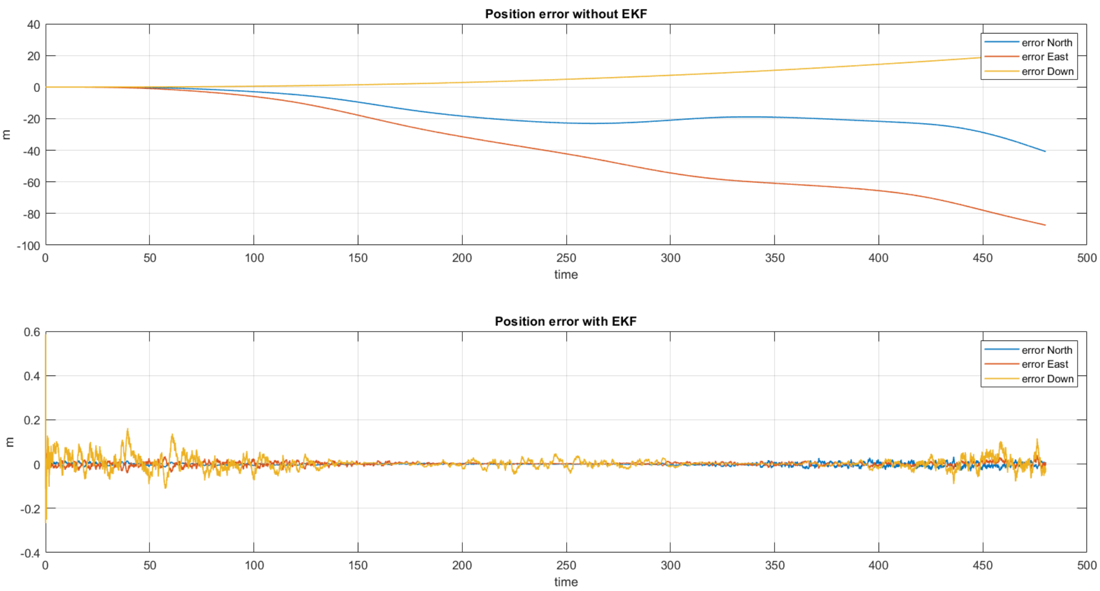

Figure 34: Position error from mechanization and estimation in comparison

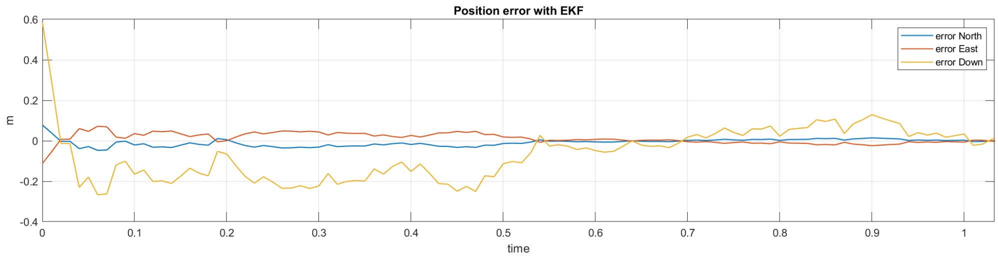

Figure 35: Position transitory, lasts around 0.6 s.

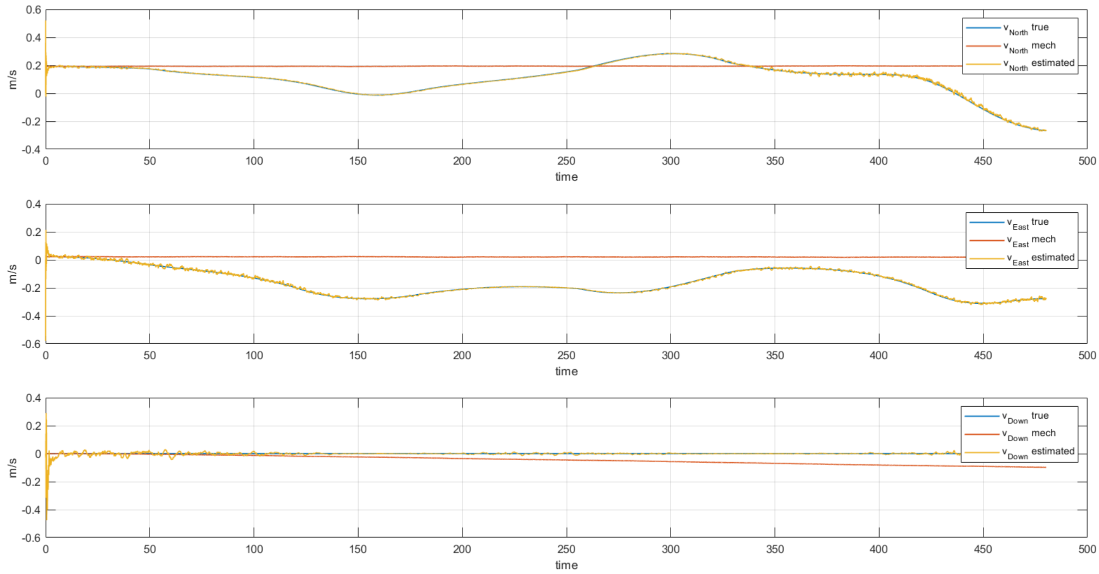

Figure 36: Velocity from mechanization and estimation in comparison

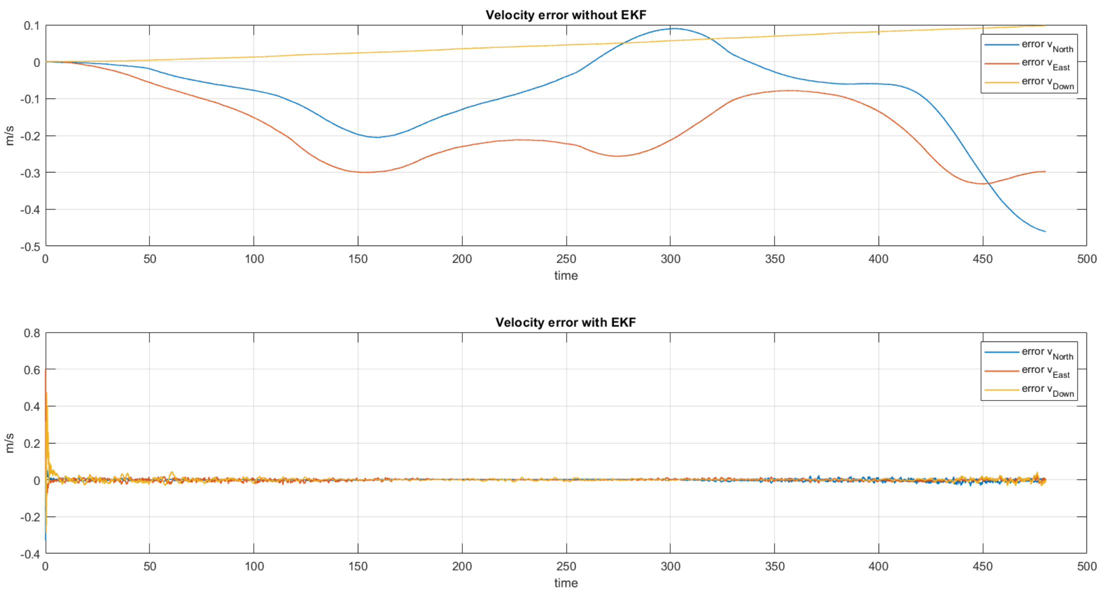

Figure 37: Velocity error from mechanization and estimation in comparison

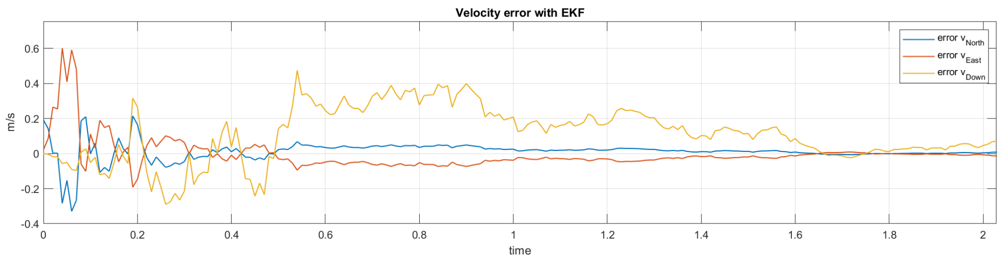

Figure 38: Velocity transitory, lasts around 1.8 s.

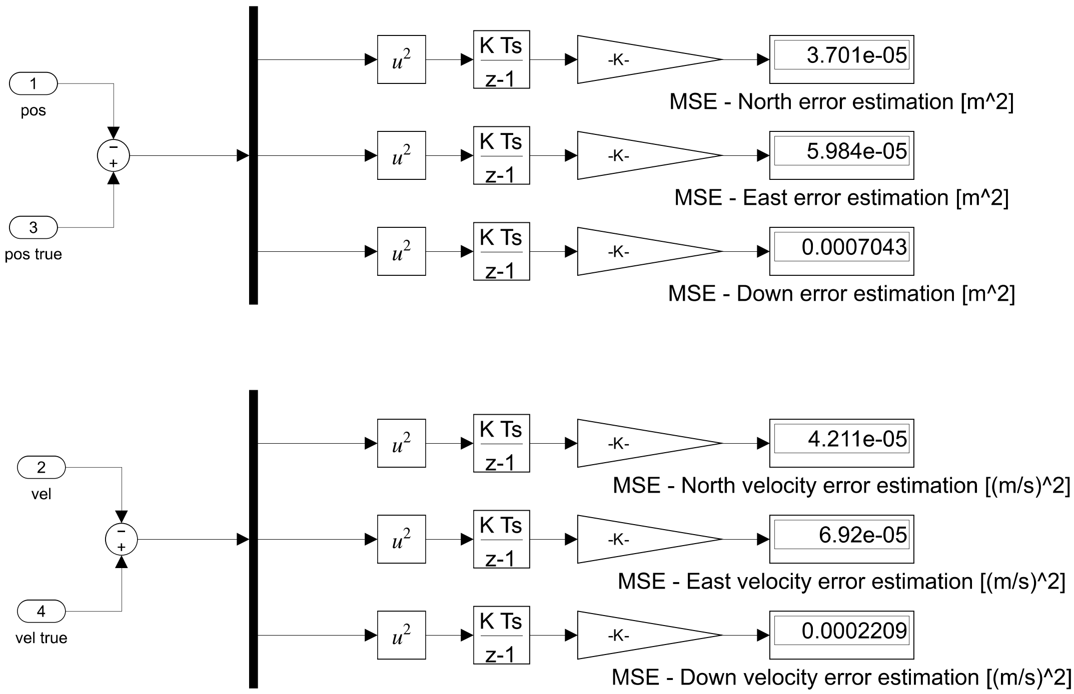

Figure 39: Here we have the mean squared error with respect of their true value of position and velocity. We can notice that the MSE on Down direction it is higher, this is due to probably due to the shorter distances from the antennas in that direction.

It is also important to highlight how the improvement of the estimation with respect to the measurements. So, considering for simplicity just the distance computed from point 6 [4] (It is easy to select another distance from the Simulink simulation), this is a comparison between the measurement of the sensor and the distance computed with the sensor model (except fo the noise) but using the estimated position and the true position.

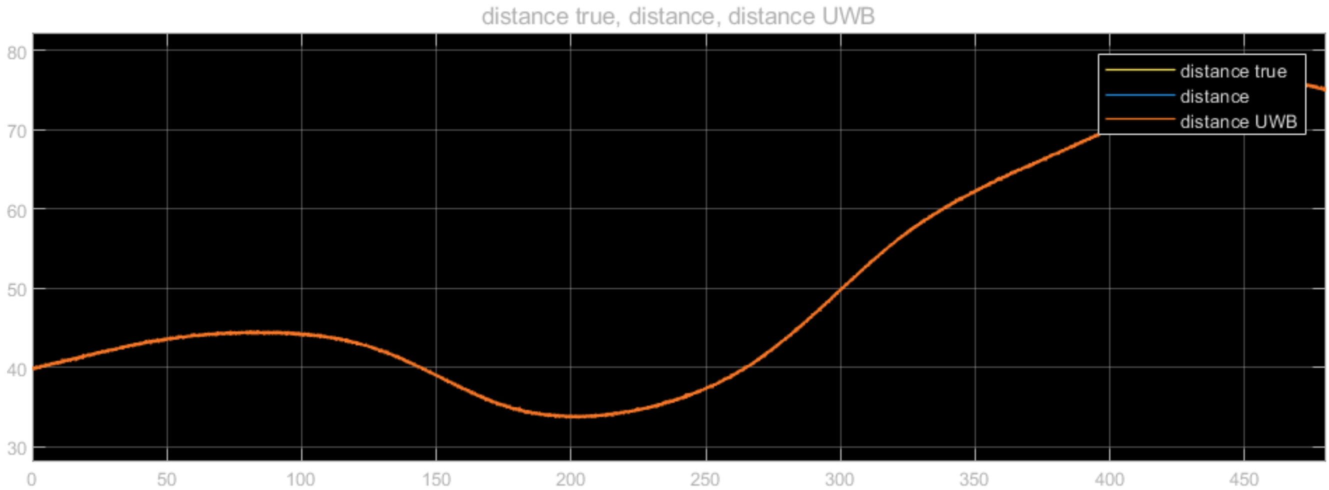

Figure 40: Distance comparison

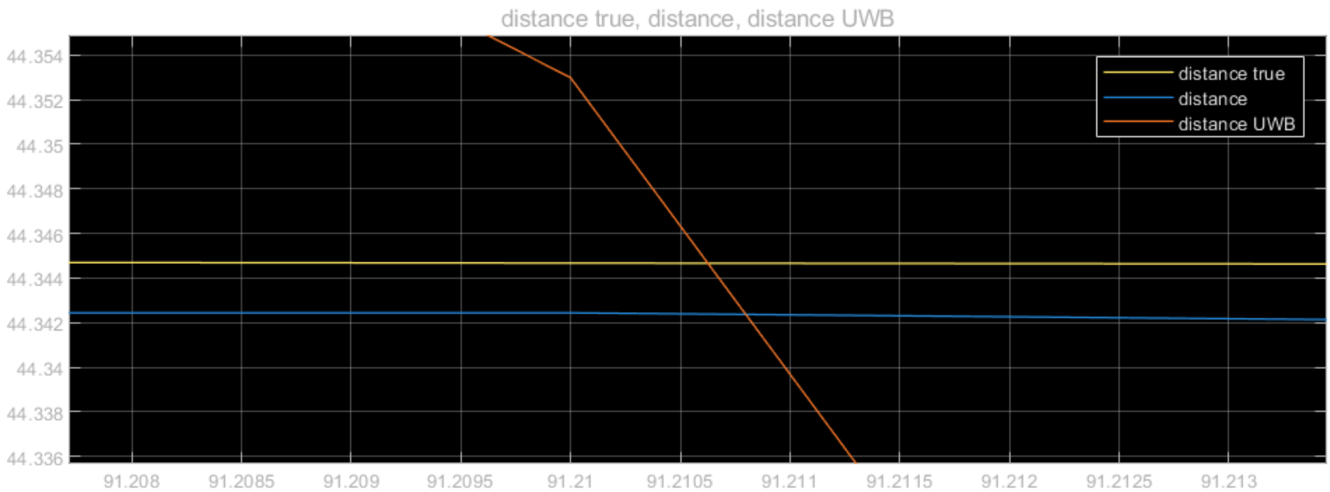

Figure 41: Distance comparison up close

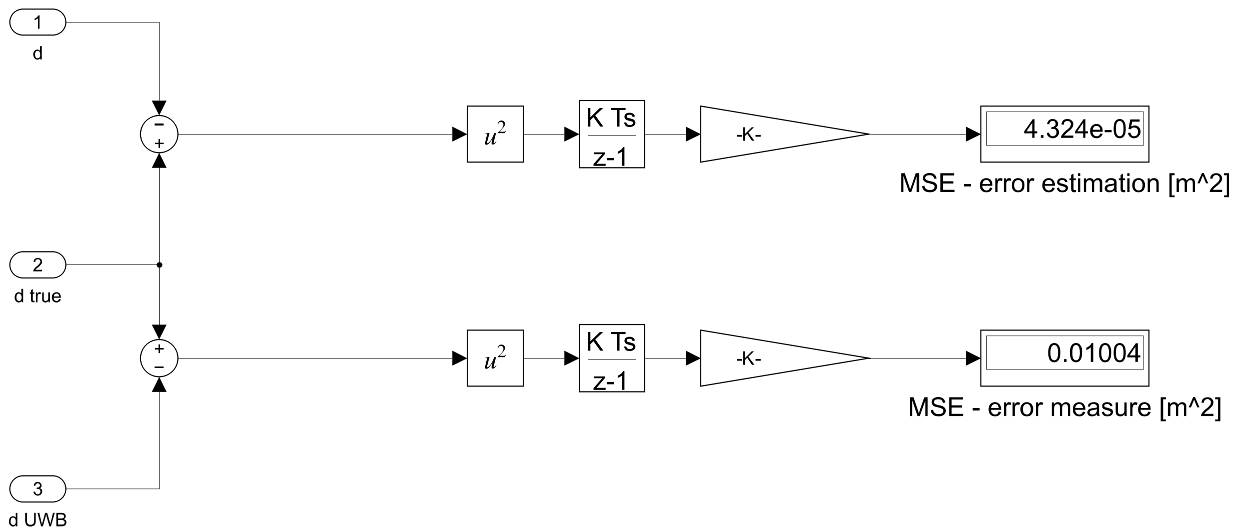

Figure 42: We can see how the mean squared error of the estimated distance drops to around 0.4% of the the MSE of the measurement.

7 Conclusions

I can eventually affirm that for this specific navigation scenario the navigation systems proved to be reasonably effective (except for a very brief transitory) since during the whole navigation it maintains an average error with respect to true values of approximately:

- 0.02° for roll and pitch and 0.15° for the yaw.
- $0.008\,m$ for North and East position and $0.03\,m$ for Down position.
- $0.008\,m/s$ for North and East velocity and $0.02\,m/s$ for Down position.

Further improvements could be done to the systems changing the position and/or velocity sensors in order to enable the navigation to happen underwater.
Finally I am also going to show the Simulink scheme through which major part of the simulations were performed.

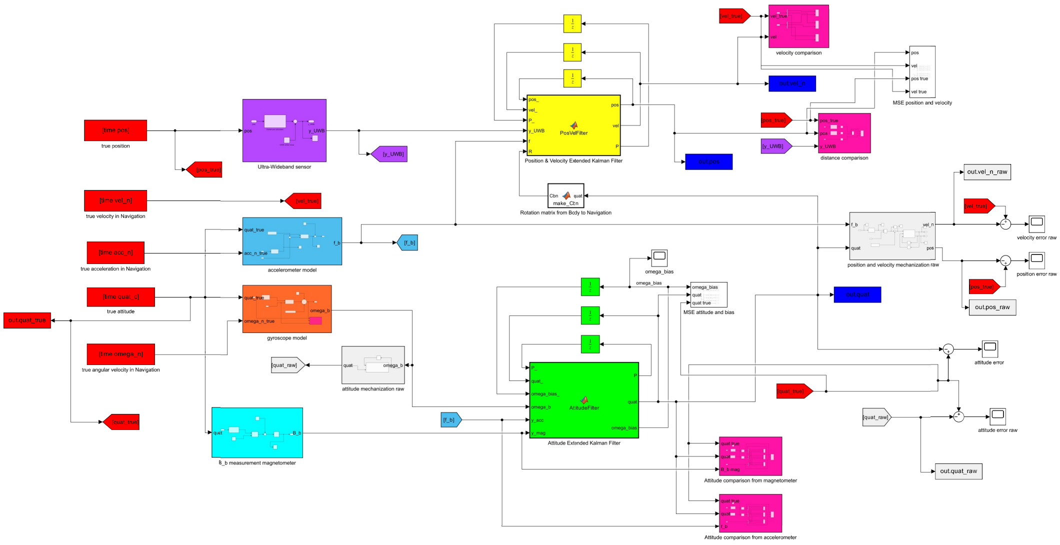

Figure 43: Simulink scheme of the navigation system.