ZEIT8009

Global Navigation Satellite Systems

Assignment 1

NINA AVERILL
z3531215

UNSW Canberra
Apr 2022

Contents

1 Introduction

The following report will analyse a linear Kalman filter applied to raw data from a moving robot. The dataset consists of calibrated Inertial Measurement Unit (IMU) (accelerometer, gyroscope) and geomagnetic field (magnetometer) readings from an Android phone. The data was collected by positioning a phone flat as a path is traversed through a building. The following Android model and sensor configuration is shown in Tables 1 and 2.

Table 1: Android Phone Configuration

Brand	Model	Android Version
OPPO	PBCM10	8.1.0

Table 2: Sensor Configuration

Model	Vendor	Resolution (m)	Power (W)	Maximum Range (m)
BMI160 Accelerometer	BOSCH	0.0023956299	0.18	39.22661
BMI160 Gyroscope	BOSCH	0.0010681152	0.9	34.906586
AK09911 Magnetometer	AKM	0.5996704	2.4	4911.9995

IMUs, consisting of accelerometer, gyroscopes and often magnetometers, are navigation devices used to calculate the position and orientation of a moving object without external references. The accelerometer measures the acceleration force along the x, y, z axes (including gravitational acceleration) in metres per second^2 (m/s^2), the gyroscope measures the rate of rotation around the x, y, z axes in radians per second (rad/s), and the magnetometer measures the geomagnetic field strength along the x, y, and z axes in microtesla (μT). All three sensors are calibrated but tend to exhibit significant noise and bias over time, with gyroscopes in particular often displaying unbounded drift. These noisy readings need to be corrected to derive the true position and orientation.

A series of waypoints were provided with the data that indicate the true (x, y) position over time and will be used the evaluate the position data output by the Kalman Filter. No other sources of truth were available and so the remaining state values will be evaluated in terms of the model uncertainty.

The raw data is processed to derive state information and the state is estimated using a linear Kalman Filter, a recursive filter that uses state space techniques to find refined estimates for the state. Though this algorithm is used extensively in literature, the model relies on the assumption that the process and measurement models are linear, and the process and measurement noise are additive Gaussian. The bias and drift associated with the raw readings mean that the process and error matrices are non-Gaussian. This problem would therefore be better represented by a non-linear Kalman Filter, such as Extended Kalman Filter (EKF) or Unscented Kalman Filter (UKF) and we expect the results of the Kalman Filter to be suboptimal.

The first half of the report will outline the state definition and Kalman algorithm used, while the second will evaluate it for its representation of the true values over time.

2 State Definition

2.1 State Vector

The model will track the position (x, y, z), velocity, $(\dot{x}, \dot{y}, \dot{z})$, acceleration $(\ddot{x}, \ddot{y}, \ddot{z})$, as well as the pose (ψ, θ, ϕ), rotation rate $(\dot{\psi}, \dot{\theta}, \dot{\phi})$ and the gyroscope bias $(\psi_\varepsilon, \theta_\varepsilon, \phi_\varepsilon)$. These values fully describe the position and orientation of the sensor over time. This gives us a state vector at time t of:

$$X_t = \begin{bmatrix} x_t \\ y_t \\ z_t \\ \dot{x}_t \\ \dot{y}_t \\ \dot{z}_t \\ \ddot{x}_t \\ \ddot{y}_t \\ \ddot{z}_t \\ \psi_t \\ \theta_t \\ \phi_t \\ \dot{\psi}_t \\ \dot{\theta}_t \\ \dot{\phi}_t \\ \psi_{\varepsilon t} \\ \theta_{\varepsilon,t} \\ \phi_{\varepsilon,t} \end{bmatrix} \tag{1}$$

2.2 State Transition

The model assume the acceleration of the robot over time to be linear, and uses this to derive estimates for the position and velocity of the robot, modelled by the kinematic equations:

$$\begin{cases} x_t = x_{t-1} + \Delta t \dot{x}_{t-1} + \frac{\Delta t^2}{2} \ddot{x}_{t-1} \\ y_t = y_{t-1} + \Delta t \dot{y}_{t-1} + \frac{\Delta t^2}{2} \ddot{y}_{t-1} \\ z_t = z_{t-1} + \Delta t \dot{z}_{t-1} + \frac{\Delta t^2}{2} \ddot{z}_{t-1} \\ \dot{x}_t = \dot{x}_{t-1} + \Delta t \ddot{x}_{t-1} \\ \dot{y}_t = \dot{y}_{t-1} + \Delta t \ddot{y}_{t-1} \\ \dot{z}_t = \dot{z}_{t-1} + \Delta t \ddot{z}_{t-1} \\ \ddot{x}_t = \ddot{x}_{t-1} \\ \ddot{y}_t = \ddot{y}_{t-1} \\ \ddot{z}_t = \ddot{z}_{t-1} \end{cases} \tag{2}$$

The gyroscope data can be modelled as:

$$\omega_t = \hat{\omega}_t + \omega_{\varepsilon,t} + n_g \tag{3}$$

Where ω is the measured angular velocity, $\hat{\omega}$ is true orientation, $\omega_{\varepsilon,t}$ is the gyroscope bias and n_g is white Gaussian noise.

Further assuming a linear angular rate of change, the measured angular velocity can be further written as:

$$\omega_t = \omega_{t-1} + \Delta t \dot{\omega}_{t-1} \tag{4}$$

We can therefore derive a representation of the current angular velocity as a function of the previous rate and bias:

$$\begin{aligned} \omega_t &= \omega_{t-1} + \Delta t \dot{\omega}_{t-1} && (5) \\ &= \omega_{t-1} + \Delta t \dot{\omega}_{t-1} - \omega_{\varepsilon,t-1} - n_g && (6) \\ & && (7) \end{aligned}$$

The state representation of orientation, rotation rate and gyroscope biases are therefore derived from the equations:

$$\begin{cases} \psi_t = \psi_{t-1} + \Delta t \dot{\psi}_{t-1} - \psi_{\varepsilon,t-1} \\ \theta_t = \theta_{t-1} + \Delta t \dot{\theta}_{t-1} - \theta_{\varepsilon,t-1} \\ \phi_t = \phi_{t-1} + \Delta t \dot{\phi}_{t-1} - \phi_{\varepsilon,t-1} \\ \dot{\psi}_t = \dot{\psi}_{t-1} \\ \dot{\theta}_t = \dot{\theta}_{t-1} \\ \dot{\phi}_t = \dot{\phi}_{t-1} \\ \psi_{\varepsilon,t} = \psi_{\varepsilon,t-1} \\ \theta_{\varepsilon,t} = \theta_{\varepsilon,t-1} \\ \phi_{\varepsilon,t} = \phi_{\varepsilon,t-1} \end{cases} \tag{8}$$

These equations, modelled in the state space, are used to generate the next predicted state vector.

In matrix form, this is represented as:

$$A = \begin{bmatrix}
1 & 0 & 0 & \Delta t & 0 & 0 & \frac{\Delta t^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{\Delta t^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{\Delta t^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

2.3 Measurements

For this model, linear acceleration, pose and rotation rate will be observed, represented by the matrix:

$$Y_t = \begin{bmatrix} \ddot{x}_t \\ \ddot{y}_t \\ \ddot{z}_t \\ \psi_t \\ \theta_t \\ \phi_t \\ \dot{\psi}_t \\ \dot{\theta}_t \\ \dot{\phi}_t \end{bmatrix} \tag{9}$$

However, these measurements are not available from the raw sensor readings, and must be derived.

2.3.1 Linear Acceleration

The accelerometer derives the readings by measuring the forces that are applied to the sensor itself. However, the force of gravity is always acting on the device such that the accelerometer will always measure a magnitude of g = 9.81 m/s^2 when stationary, acting in the direction of the Earth's centre of mass. To isolate the force of gravity, a low-pass filter is used, such that:

$$\mathbf{g_t} = \alpha \mathbf{g_{t-1}} + (1-\alpha) \begin{bmatrix} \hat{\ddot{x}}_t \\ \hat{\ddot{y}}_t \\ \hat{\ddot{z}}_t \end{bmatrix} \tag{10}$$

Where α is a weighting value such that $\alpha \in [0,1]$ and $\hat{\ddot{x}}_t, \hat{\ddot{y}}_t, \hat{\ddot{z}}_t$ are the raw accelerometer readings.

The contributing effects of gravity is then removed from the raw accelerometer data by applying a high pass filter:

$$\begin{bmatrix} \ddot{x}_t \\ \ddot{y}_t \\ \ddot{z}_t \end{bmatrix} = \begin{bmatrix} \hat{\ddot{x}}_t \\ \hat{\ddot{y}}_t \\ \hat{\ddot{z}}_t \end{bmatrix} - \mathbf{g_t} \tag{11}$$

This process removes erroneous forces from the sensor readings, such that only the active acceleration is retained. The results of this filtering process can be seen in Figure 1.

Figure 1: Comparison of raw and gravity-corrected acceleration over time

2.3.2 Euler Angles

Euler angles describe the orientation (pose) of a rigid body with respect to a reference coordinate frame and can be estimated from a combination of the accelerometer and magnetometer data. The estimation of roll and pitch from sensed acceleration assumes that there is no linear acceleration at the start, and so any normalised accelerometer measurement can be related to gravitational acceleration through a rotation matrix (Aranburu, 2018). The conversion from raw accelerometer readings to pose angle is shown in Equation 12 and 13.

$$\psi_t = \tan^{-1}\left(\frac{\hat{\hat{y}}_t}{\sqrt{\hat{\hat{x}}_t^2 + \hat{\hat{z}}_t}}\right) \tag{12}$$

$$\theta_t = \tan^{-1}\left(\frac{\hat{\hat{x}}_t}{\sqrt{\hat{\hat{y}}_t^2 + \hat{\hat{z}}_t}}\right) \tag{13}$$

These derived measurements are then used to transform the local magnetic field vectors into an estimate for the yaw angle. Magnetometer readings are defined in the body-fixed coordinate frame and so must undergo tilt compensation by rotating the sensor onto the horizontal plane (Jiang, 2018).

$$\zeta_x = \begin{bmatrix} \cos\theta_t \\ \sin\psi_t \cdot \sin\theta_t \\ \cos\psi_t \cdot \sin\theta_t \end{bmatrix} \begin{bmatrix} \hat{\zeta}_x \\ \hat{\zeta}_y \\ \hat{\zeta}_z \end{bmatrix}^T \tag{14}$$

$$\zeta_y = \begin{bmatrix} 0 \\ \cos\psi_t \\ \sin\psi_t \end{bmatrix} \begin{bmatrix} \hat{\zeta}_x \\ \hat{\zeta}_y \\ \hat{\zeta}_z \end{bmatrix}^T \tag{15}$$

Where $\hat{\zeta}_x, \hat{\zeta}_y, and \hat{\zeta}_z$ are the raw magnetometer readings in the body frame. The yaw estimate can then be calculated as:

$$\phi = \tan^{-1}\left(\frac{-\zeta_y}{\zeta_x}\right) \tag{16}$$

2.3.3 Euler Rates

The gyroscope measures the angular velocity in the body-fixed reference frame. These Euler rates can then be derived from these measurements through a rotation into the world frame (Mathworks, 2022).

$$\begin{bmatrix} \dot{\psi}_t \\ \dot{\theta}_t \\ \dot{\phi}_t \end{bmatrix} = \begin{bmatrix} 1 & \sin(\psi)\tan(\theta) & \cos(\psi)\tan(\theta) \\ 1 & \cos(\psi) & -\sin(\psi) \\ 1 & \frac{\sin(\psi)}{\cos(\theta)} & \frac{\cos(\psi)}{\cos(\theta)} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{17}$$

2.4 Control Input and Matrix

No control inputs are factored into the state estimation and will therefore be omitted from the following analysis.

2.5 Measurement Matrix

For this model, acceleration, pose and rotation rate are observed. This leads to the measurement matrix:

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{18}$$

2.6 Process Noise Uncertainty Matrix

Since the dynamic model doesn't include a control matrix, the process noise can be represented by the random variance in the accelerometer, gyroscope and magnetometer values. This variance is derived from the sensor covariance, generated from the set of measurements across the full state space. The errors in the state are also interdependent, as values are some elements represent the integration of others over time. This relationship can be captured by defining the matrix Q_s such that:

$$Q_s = \begin{bmatrix}
\sigma^2_{a,x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \sigma^2_{a,y} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \sigma^2_{a,z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \sigma^2_{a,x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \sigma^2_{a,y} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \sigma^2_{a,z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{a,x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{a,y} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{a,z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{g,x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{g,y} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{g,z} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{g,x} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{g,y} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{m,z} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{g,x} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{g,y} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \quad (19)$$

Where σ_a, σ_g and σ_m are the standard distribution of the accelerometer, gyroscope and magnetometer sensors respectively.

The interdependence of the state relationship can then be captured by Equation 20, thereby better representing the noise inherent in the dynamic model (Becker, 2022).

$$\mathbf{Q} = \mathbf{A}\mathbf{Q_s}\mathbf{A}^T \quad (20)$$

3 Kalman Filter Algorithm

3.1 Process

The Kalman Filter algorithm learns the dynamics of the state space through recursively minimising a quadratic cost function with respect to the estimated spaced and the correct space. This process consists of two primary phases: the predict and the update phase.

The predict phase uses knowledge of the state transition dynamics and the inherent noise characteristics to estimate the state and estimate uncertainty at time $t+1$. Assuming the system we are modelling is linear, the discrete model can be written as:

$$X_{t,p} = AX_{t-1} + w_t \tag{21}$$

Where $X_{t,p}$ is the state estimate and w_t is white Gaussian noise.

$$P_{t,p} = AP_{t-1}A^T + Q \tag{22}$$

The update phase is triggered by the input of a new observation and measurement uncertainty into the system and updates the state such that the uncertainty of future predictions is minimised. This update consists of the following equations:

$$K_t = \frac{P_{t,p}H^t}{HP_{t,p}H^T + R} \tag{23}$$

Where K_t is the Kalman gain, and R is the measurement uncertainty matrix.

$$X_t = X_{t,p} + K_t[Y_t - HX_t] \tag{24}$$

Where X_t is the estimated state space.

$$P_k = (I - K_t H)P_{t,p} \tag{25}$$

Where P_k is the estimated uncertainty and I is the identity matrix.

If the process inputs fully represent the state space, the Kalman gain and estimate uncertainty should converge towards zero over time.

3.2 Initial Conditions

The initial state estimate is given by the initial waypoint location and linear acceleration. All other values are estimated to be zero.

$$X_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 0 \\ 0 \\ 0 \\ \ddot{x}_t \\ \ddot{y}_t \\ \ddot{z}_t \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{26}$$

As the initial state is highly uncertain, the initial estimate uncertainty is set to an arbitrarily large value. In this case, the estimate uncertainty was represented by:

$$P_0 = I \cdot 500 \tag{27}$$

4 Evaluation

The Kalman Filter was modelled in python and run across numerous test data sets to evaluate the efficacy of this model. Overall, the model consistently showed convergence over time for a majority of the state element, with some notable exceptions. Comparing the observable state elements in Figures 2, 3, and 4, there is a significant decrease in the noise of the filtered data compared with the derived observations.

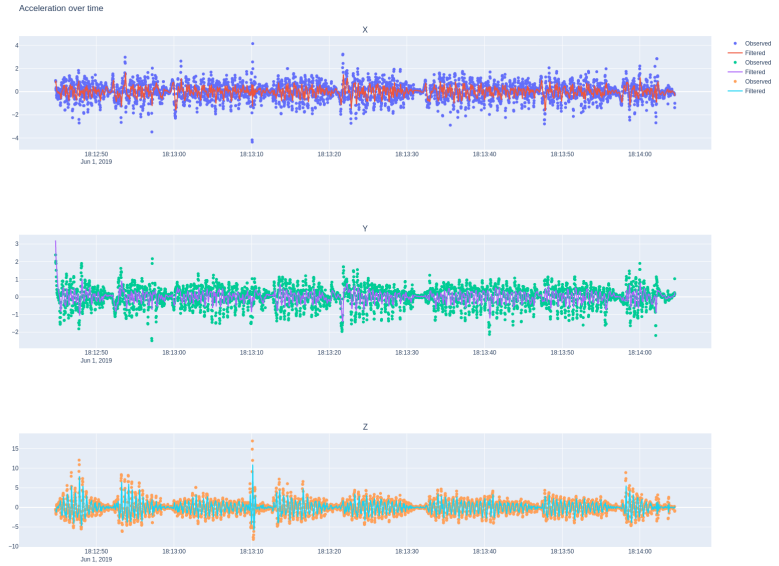

Figure 2: Comparison of raw and filtered acceleration over time

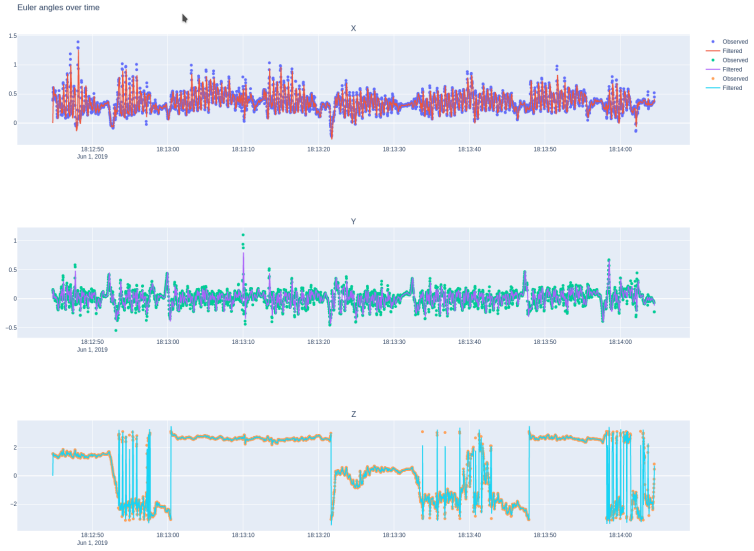

Figure 3: Comparison of raw and filtered Euler angles over time

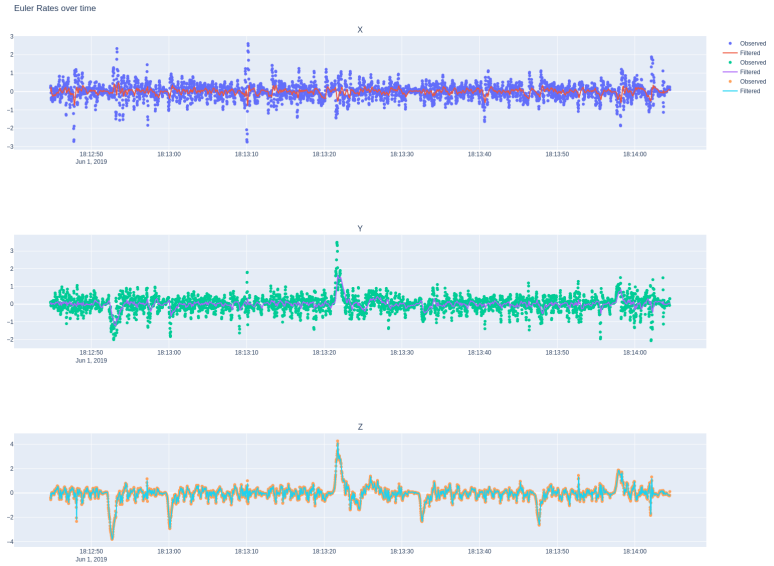

Figure 4: Comparison of raw and filtered Euler rates over time

Figure 5 displays the steady decrease in Kalmain gain for acceleration over time, suggesting that the model's predicted state became a more accurate representation of the true state than the noisy measurement data.

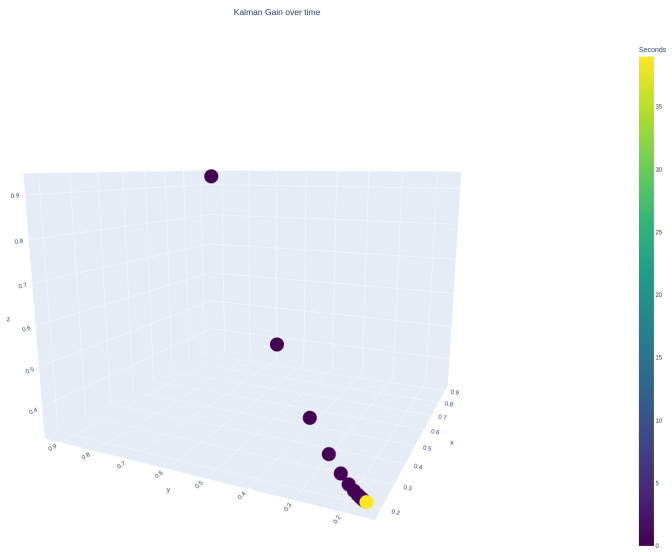

Figure 5: Kalmain Gain convergence over time

Therefore, the model can be shown to perform well for the state elements that can be validated against measurable data. However, Figure 6 displays the performance of the estimated position against the true value, derived from the list of waypoints. The model shows clear

divergence from the source truth, and fails to replicate trends the robot takes along the path. This degraded performance can clearly be seen in Figures 9 to 14, where the uncertainty displays an unbounded increase over time. This can likely be attributed to errors in the acceleration estimate compounding, as the model has no means with which to refine the predictions over time. An improvement to the model could therefore be the addition of GPS data, which would provide information about the current position of the vehicle in an external reference frame.

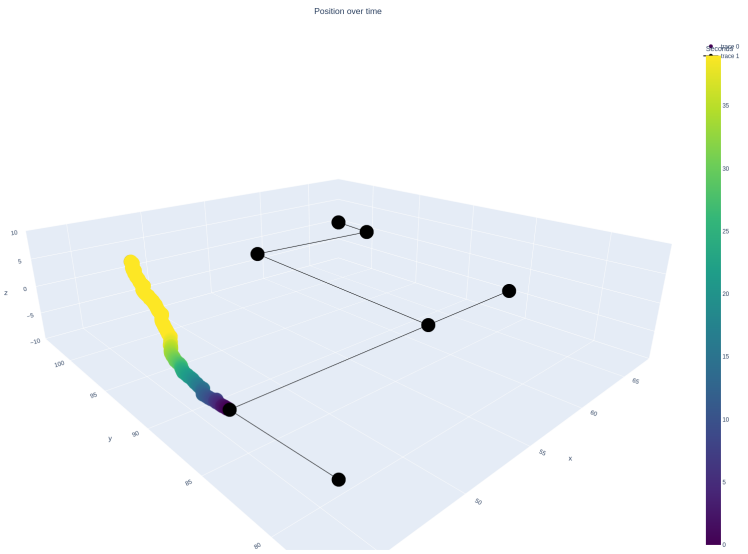

Figure 6: 3D Position over time

Figure 7 shows a similar divergence from the true value, but shows a closer approximation of the path and terminates near the final waypoint.

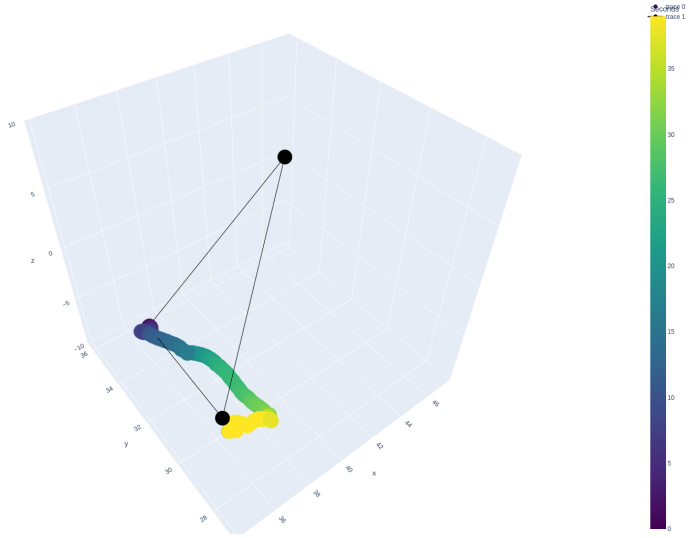

Figure 7: 3D Position over time for alternate data set

Another notable feature of the model is that the roll, pitch and yaw biases can be seen to converge over time, with roll and pitch rapidly converging towards zero. In comparison, though yaw bias does not exhibit unbounded uncertainty growth, the values oscillate around a non-zero value, indicating an unstable estimate function. This is further seen in Figure 3, where the estimates poorly track the noisy data. A possible explanation for this is its derivation from magnetometer readings, a sensor that is easily perturbed by electromagnetic disturbances from hard iron, electric current and indoor environments (Wu, 2019).

A key parameter of the model is the initial estimate uncertainty, as it is will affect the degree to which the Kalman Filter will weight the model prediction against the measurements. Too low an inital uncertainty, and the model can fail to converge to the true representation of the stare. In Figure 8, a low uncertainty for the uncertain initial estimate results in a diverging Kalman gain over time and unstable uncertainties for state estimates.

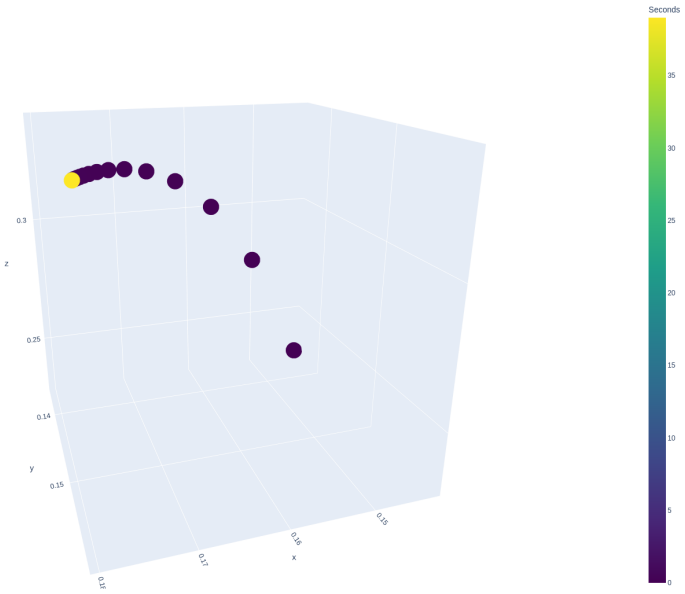

Figure 8: Kalman Gain divergence over time from low initial uncertainty

5 Conclusion

Though the linear Kalman filter is generally considered to be a poor estimator for non-linear state spaces, the model output reasonable estimates for most state elements despite the complexity of the system. The model clearly showed convergence towards a stable estimate for the state space that was a reasonable approximation of the noisy data. It is recommended that non-linear Kalman filter algorithms, such as EKFs or UKFs are considered, to identify if uncertainties in the position, velocity and yaw derivatives can be minimised.

6 List of Figures

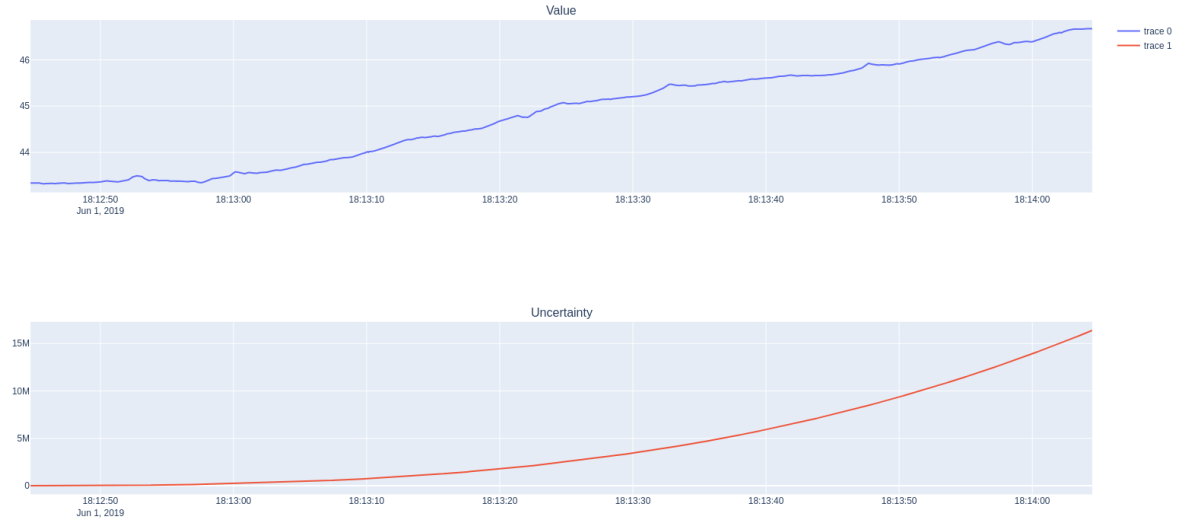

Figure 9: X Position state and uncertainty over time

Figure 10: Y Position state and uncertainty over time

Figure 11: Z Position state and uncertainty over time

Figure 12: X Velocity state and uncertainty over time

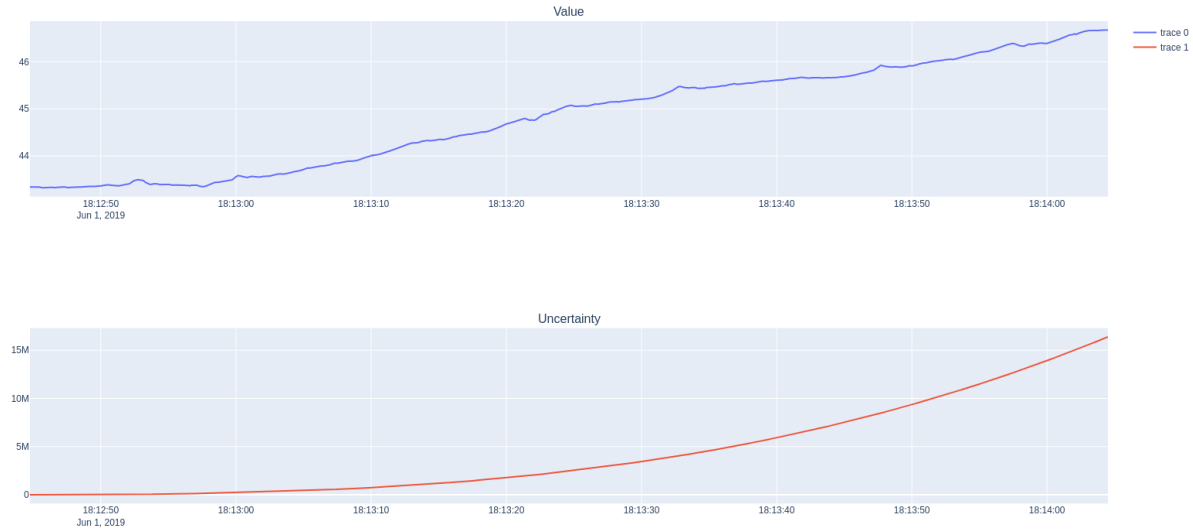

Figure 13: Y Velocity state and uncertainty over time

Figure 14: Z Velocity state and uncertainty over time

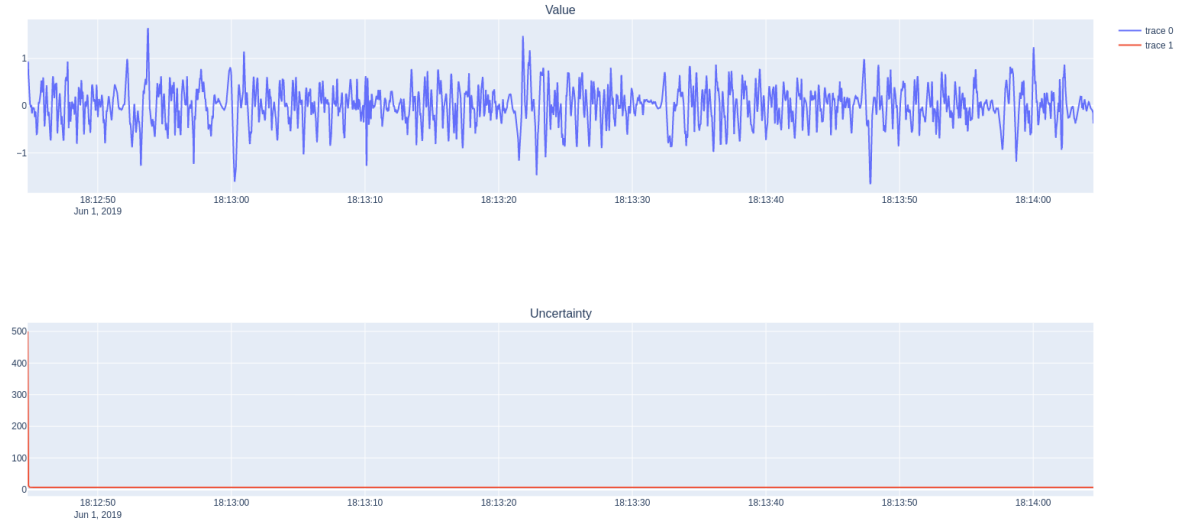

Figure 15: X Acceleration state and uncertainty over time

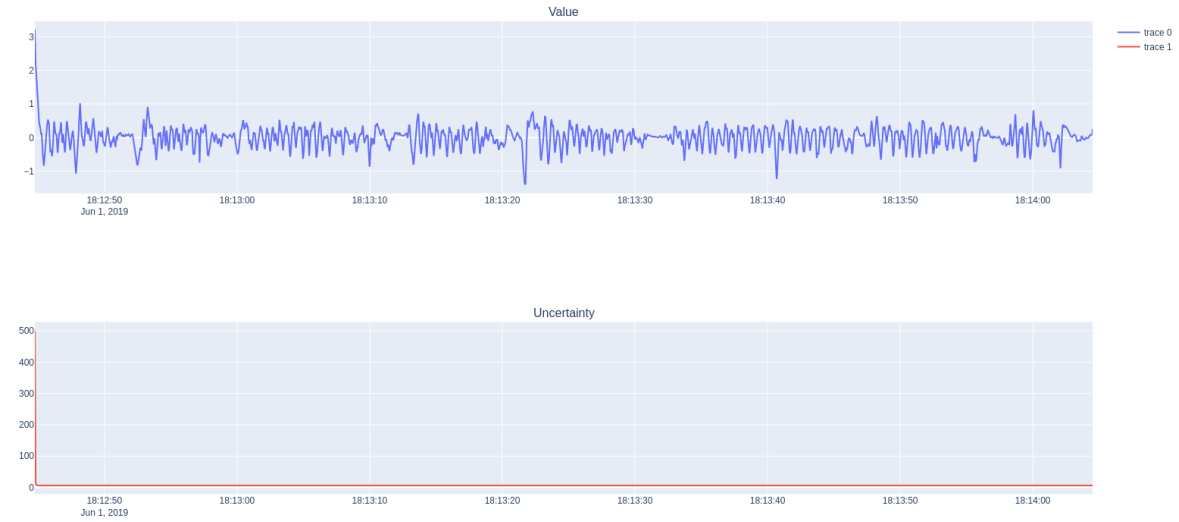

Figure 16: Y Acceleration state and uncertainty over time

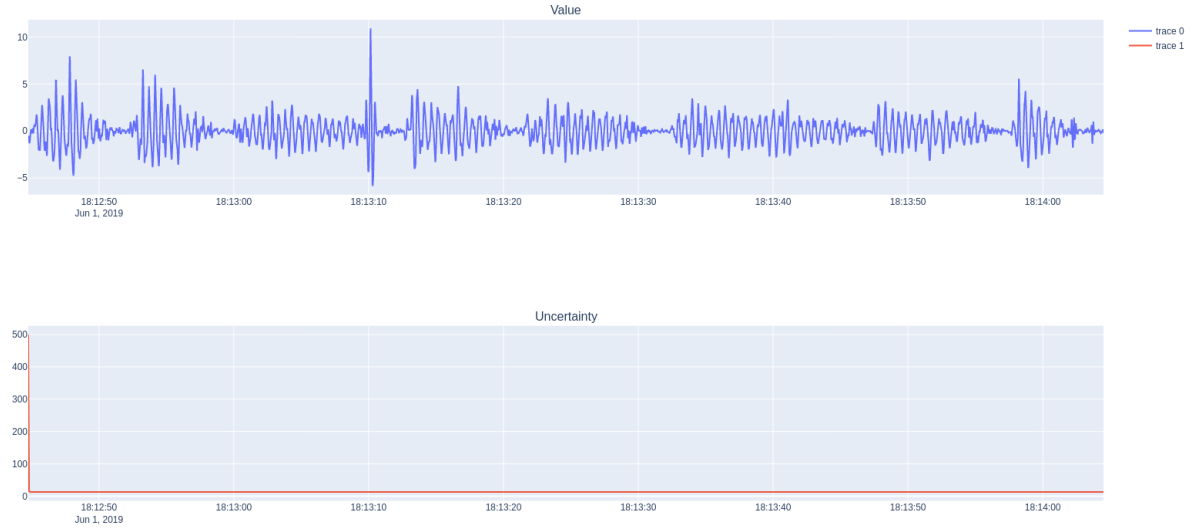

Figure 17: Z Acceleration state and uncertainty over time

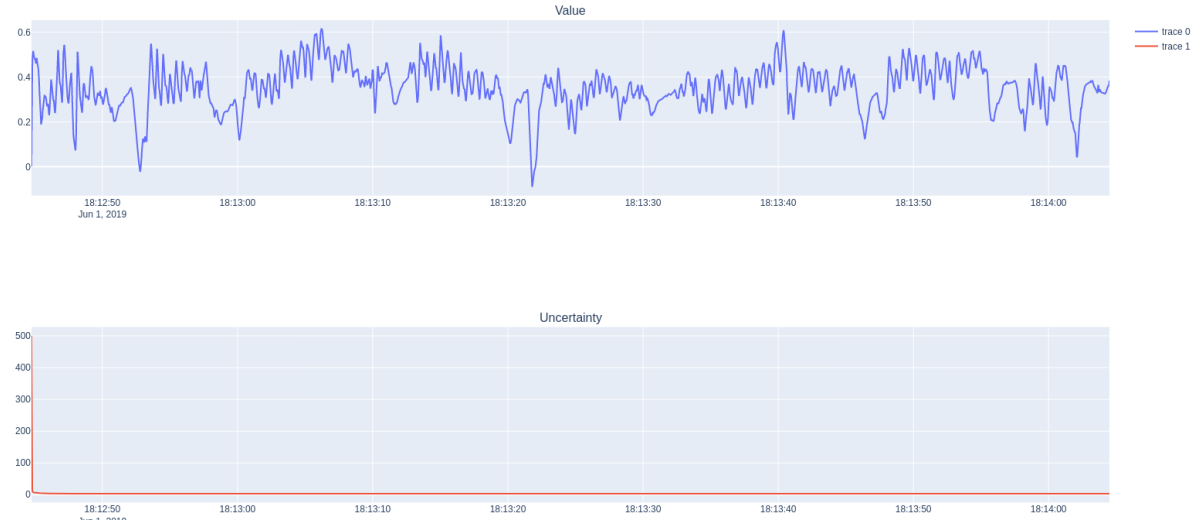

Figure 18: Roll state and uncertainty over time

Figure 19: Pitch state and uncertainty over time

Figure 20: Yaw state and uncertainty over time

Figure 21: Roll rate state and uncertainty over time

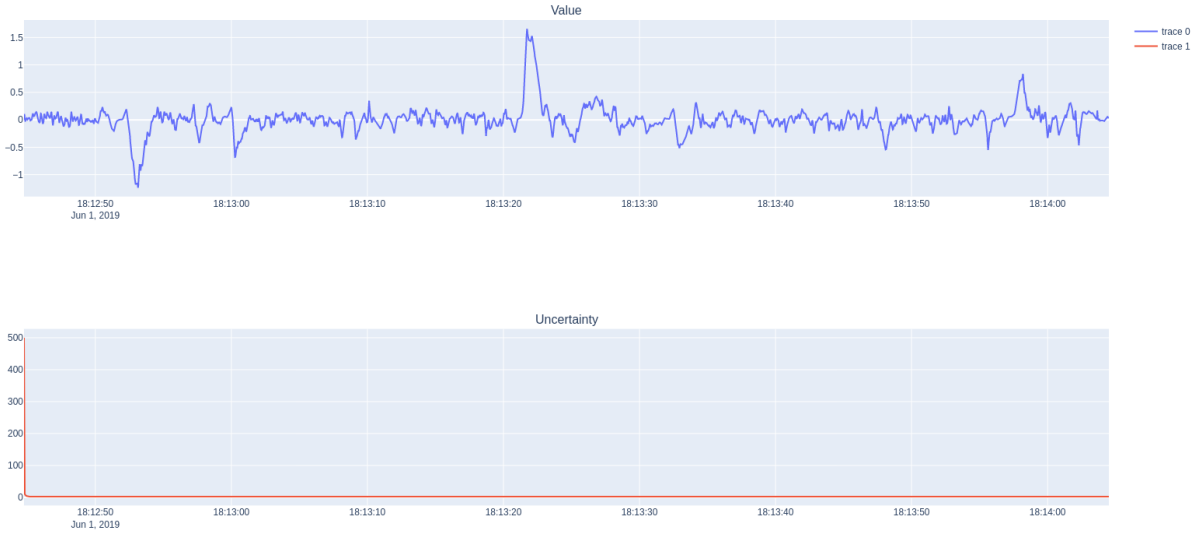

Figure 22: Pitch rate state and uncertainty over time

Figure 23: Yaw rate state and uncertainty over time

Figure 24: Roll bias state and uncertainty over time

Figure 25: Pitch bias state and uncertainty over time

Figure 26: Yaw bias state and uncertainty over time

7 Appendix

References

Aranburu, A. (2018). *Imu data processing to recognize activities of daily living with a smart headset*. University of California, Santa Cruz.

Becker, A. (2022). *Kalman filter: Constructing the process noise matrix q.*

Jiang, L. (2018, 05). *Yaw angle estimation from gyroscope and magnetometer based on kalman filter.*

Mathworks. (2022). *Custom variable mass 6dof (euler angles): Implement euler angle representation of six-degrees-of-freedom equations of motion of custom variable mass.*

Wu, J. (2019). Real-time magnetometer disturbance estimation via online nonlinear programming. *IEEE Sensors Journal*, *19*(12), 4405–4411.