Robot Localization and Navigation Project 3

Ali Rasteh

April 25, 2024

Problem Definition

In this project, we are going to combine measurements from both a vision sensor on the drone in form of pose and velocity estimations along with IMU velocity measurements. Just similar to Project 1, the chosen orientation representation for this project is ZYX Euler parameterization, and the experiments are done in two parts. In the first part, the position and orientation from the onboard Apriltag tracker are used as measurements for an Unscented Kalman Filter fed with IMU measurements as input to a motion kinematic model. In the second part of the project, the measurement is just the velocity of the drone computed from the onboard vision sensor.

Problem Formulation

Process Model

The following state vector is used as the state in system dynamics for the purpose of this project.

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \dot{\mathbf{p}} \\ \mathbf{b}_g \\ \mathbf{b}_{\boldsymbol{a}} \end{bmatrix} = \begin{bmatrix} \text{position} \\ \text{orientation} \\ \text{linear velocity} \\ \text{gyroscope bias} \\ \text{accelerometer bias} \end{bmatrix} \in \mathbf{R}^{15} \tag{1}$$

As mentioned in the course slides, the following formulation is used as the system dynamics equations for both part 1 and part 2 in this project.

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_3 \\ \mathbf{G}\left(\mathbf{x}_2\right)^{-1}\left(\boldsymbol{\omega}_m - \mathbf{x}_4 - \mathbf{n}_g\right) \\ \mathbf{g} + R\left(\mathbf{x}_2\right)\left(\mathbf{a}_m - \mathbf{x}_5 - \mathbf{n}_a\right) \\ \mathbf{n}_{bg} \\ \mathbf{n}_{ba} \end{bmatrix} = f(x, u, n) \tag{2}$$

where, $\mathbf{n}_a$ and $\mathbf{n}_g$ represent the accelerometer and gyroscope noise terms and $\mathbf{n}_{ba}$ and $\mathbf{n}_{bg}$ represent the dynamics of the accelerometer and gyroscope biases. As mentioned in the project the noise covariance matrix or noise standard deviation values should be tuned in a way that the estimation works well with all 3 datasets. However, in the real-world scenario, these Gaussian noise variances should be provided by the manufacturer of the sensors and used in the dynamics. However, in this project, we did according to the requirements and tuned the covariance matrix and our defined values could be observed in "pred_step.m". Furthermore, the accelerometer and gyroscope measurements are considered as the inputs to the process model and are indicated by $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ in Eq.2. Still $\mathbf{G}\left(\mathbf{x}_2\right)$ and $\mathbf{R}\left(\mathbf{x}_2\right)$ should be defined in the process equations. The $\mathbf{R}\left(\mathbf{x}_2\right)$ is the rotation matrix corresponding to $\mathbf{x}_2$ or the orientation of the system which are the Euler angles along Z, Y, and X axis. The $\mathbf{R}$ matrix for the ZYX rotation could be derived by multiplying the basic rotation matrices along each of the axes as follows.

$${}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}}(\psi, \theta, \phi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & -sin(\phi) \\ 0 & sin(\phi) & cos(\phi) \end{bmatrix} \tag{3}$$

where $\mathcal{W}$ represents the world inertial frame and $\mathcal{B}$ is the body frame fixed to the IMU. In this project, the IMU frame coincides with the robot body frame, and no more rotation is needed. A rotation matrix ${}^{A}\mathbf{R}_{B}$ rotates the frame A to align with frame B. So in Eq. 3 we first rotate the inertial frame around its z-axis for ψ radians and

then around the y-axis of the resulting frame for θ radian and finally around the x-axis of the result for ϕ radians. In our codes, we use the "eul2rotm_zyx and" MATLAB builtin "eul2rotm" functions to do rotations.
The $\mathbf{G}(\psi, \theta, \phi)$ matrix maps the rate of the Euler parameters $\dot{\psi}, \dot{\theta}, \dot{\phi}$ to the body angular velocity of frame $\mathcal{B}$ with respect to frame $\mathcal{W}$ and expressed in frame $\mathcal{W}$, ${}^{\mathcal{W}}\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{W}}$.

$$ {}^{\mathcal{W}}\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{W}} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \mathbf{G}(\psi, \theta, \phi) \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{4} $$

The expression for this mapping is derived based on the law of adding angular velocities. For the ZYX convention in this project, it can be derived as follows:

$$ \begin{aligned} \mathbf{G}(\psi, \theta, \phi) = & \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \dot{\psi} \\ & + \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \dot{\theta} \\ & + \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \dot{\phi} \end{aligned} \tag{5} $$

The Eq. 5 we first express the rotation rate around the z-axis, $\dot{\psi}$ in the world frame, then we express the rotation rate around the y-axis, $\dot{\theta}$ in the world frame, and finally we express the rotation rate around the x-axis, $\dot{\phi}$ in the world frame. We sum all of these rotation rates expressed in the world frame to find the complete transformation matrix. The resulting expression is then computed as follows:

$$ \mathbf{G}(\psi, \theta, \phi) = \begin{bmatrix} cos(\theta)cos(\psi) & -sin(\psi) & 0 \\ cos(\theta)sin(\psi) & cos(\psi) & 0 \\ -sin(\theta) & 0 & 1 \end{bmatrix} \tag{6} $$

It's very important to note that the gyroscope, measures the body angular velocity as expressed in the body frame, ${}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}}^{\mathcal{W}}$, while Eq.4 expresses the Euler parameter rates to the body angular rate expressed in the world frame. So we should pre-multiply $\mathbf{G}(\psi, \theta, \phi)$ by the inverse of $\mathbf{R}$ shown in Eq. 3 to find the right and final transformation matrix. So the final $\mathbf{G}$ matrix is as follows and appears the same in our codes.

$$ \mathbf{G}(\psi, \theta, \phi) = \begin{bmatrix} 1 & 0 & -sin(\theta) \\ 0 & cos(\phi) & sin(\phi)cos(\theta) \\ 0 & -sin(\phi) & cos(\phi)cos(\theta) \end{bmatrix} \tag{7} $$

Measurement Model for Part1

In the first part of this project, the measurements are assumed to be the position and orientation of the robot reported by the Vicon system. These measurements are directly accessible in the states, so the measurement model is simply as follows. It simply extracts the first 6 items in systems states which correspond to the position and orientation of the drone.

$$ \begin{aligned} &\mathbf{C} = \begin{bmatrix} \mathbf{I}_{6\times 6} & \mathbf{0}_{6\times 9} \end{bmatrix}, \\ &\mathbf{y} = \mathbf{C}\mathbf{x} + v \end{aligned} \tag{8} $$

Measurement Model for Part2

In the second part of the project, the measurement is just the velocity of the drone computed from the onboard vision sensor. The difficulty in this section is that the provided velocities by the camera are in its local coordinate frame and must be mapped to the IMU body frame before fusion. We achieve this through the adjoint transform.

Computing the Adjoint Using Camera measurements

In mapping the body-frame camera linear velocity to the IMU frame, the angular velocity of the platform needs to be taken into account. In this section, I use the angular velocity from the optical flow to formulate the mapping:

$$ {}^{c}\mathbf{v}_{c}^{w} = \mathbf{R}_{b}^{c}\mathbf{R}_{b}^{w}\left(\mathbf{x}_2\right)^{\top}\mathbf{x}_3 - \mathbf{R}_{b}^{c}S\left(\mathbf{r}_{bc}^{b}\right)\mathbf{R}_{c}^{b}({}^{c}\boldsymbol{\omega}_{c}^{w} + \mathbf{w}_{\omega}) + \mathbf{w}_{v} \tag{9} $$

Where $\mathbf{R}_{b}^{c}$ and $\mathbf{r}_{b}^{c}$ are the extrinsic transformation between the camera and the IMU, $S(.)$ is the skew-symmetric matrix representation of the cross product and $\mathbf{w}_{v}, \mathbf{w}_{\omega}$ are linear and angular velocity noise terms. Furthermore, $\mathbf{R}_{b}^{w}\left(\mathbf{x}_2\right)$ is the rotation matrix parameterized by the rotation part of the state $(\mathbf{x}_2)$.

Computing the Adjoint Using Gyroscope measurements

We can modify the measurement model as follows to use the more robust angular velocity sensor which is the gyroscope. We should also consider the bias drift of the gyroscope.

$$ {}^{c}\mathbf{v}_{c}^{w} = \mathbf{R}_{b}^{c}\mathbf{R}_{b}^{w}\left(\mathbf{x}_2\right)^{\top}\mathbf{x}_3 - \mathbf{R}_{b}^{c}S\left(\mathbf{r}_{bc}^{b}\right)\left({}^{b}\boldsymbol{\omega}_{b}^{w} + \mathbf{w}_{gyro} - \mathbf{x}_4\right) + \mathbf{w}_{v} \tag{10} $$

Note that the gyroscope bias is removed through its estimated value in the state $(\mathbf{x}_4)$.

Filter Equations

To deal with the nonlinearities, in this project we resort to the UKF way of propagating uncertainties. As described in the course slides, the filter operates in two steps:

UKF Transform

The goal of the transform is to propagate a random variable through a nonlinear mapping and determine the distribution of the output variable. For a nonlinear system with non-additive noise terms, we should augment the state as $\mathbf{x}_{aug} = [\mathbf{x}^{\top}, \mathbf{n}^{\top}]^{\top}$:

$$ X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{aug})\ \boldsymbol{\mu} = \begin{bmatrix}\boldsymbol{\mu}_x \\ \mathbf{0}\end{bmatrix},\ \boldsymbol{\Sigma}_{aug} = \begin{bmatrix}\boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}\end{bmatrix} \tag{11} $$

Denoting n, n_q as the dimensionality of state and noise, the augmented state has a dimension equal to the sum of the two and is defined as $n' = n + n_q$. For the augmented states defined above we then generate $2n'+1$ sigma points as follows:

$$ \mathcal{X}_{aug}^{(0)} = \boldsymbol{\mu}_{aug}, \quad \mathcal{X}_{aug}^{(i)} = \boldsymbol{\mu}_{aug} \pm \sqrt{n' + \lambda'}\left[\sqrt{\boldsymbol{\Sigma}_{aug}}\right]_i \tag{12} $$

where $\lambda = \alpha^2(n' + k) - n'$ determines the spread of the sigma points through α, k. As stated in the course slides, typical values for these parameters for the case of a Gaussian distributed random variable are $\alpha = 0.001, k = 1$. Then, we feed these sigma points through the nonlinearity as follows:

$$ \mathcal{Y}^{(i)} = h\left(\mathcal{X}_{aug}^{(i),x}, \mathcal{X}_{aug}^{(i),q}\right) \quad i = 0, \dots 2n' \tag{13} $$

where $\mathcal{X}_{aug}^{(i),x}$ denotes the state section of the the sigma point $\mathcal{X}^{(i)}$ while $\mathcal{X}_{aug}^{(i),q}$ represents the noise section.

Finally, we use the mapped sigma points $\dagger^{(i)}$ to compute the corresponding mean and covariance as follows:

$$ \boldsymbol{m}_Y = \sum_{i=0}^{2n'} W_i^{(m)'} y^{(i)} \quad W_0^{(m)\prime} = \frac{\lambda'}{n' + \lambda'} \quad W_i^{(m)'} = \frac{1}{2\left(n' + \lambda'\right)} \quad i = 1, \dots 2n' \tag{14} $$

$$ \boldsymbol{S}_Y = \sum_{i=0}^{2n'} W_i^{(c)\prime}\left(\mathcal{Y}^{(i)} - \boldsymbol{m}_Y\right)\left(\mathcal{Y}^{(i)} - \boldsymbol{m}_Y\right)^{T} \quad W_0^{(c)\prime} = \frac{\lambda'}{n' + \lambda'} + \left(1 - \alpha^2 + \beta\right) \quad W_i^{(c)\prime} = \frac{1}{2\left(n' + \lambda'\right)} \tag{15} $$

Furthermore, the cross-covariance of $\mathbf{X}$ and $\mathbf{Y}$ may be computed as follows:

$$\boldsymbol{C}_U = \sum_{i=0}^{2n'} W_i^{(c)\prime} \left(\mathcal{X}^{(i),x} - \mathbf{m}_X\right) \left(\mathcal{Y}^{(i)} - \boldsymbol{m}_Y\right)^T \tag{16}$$

Prediction

In the prediction step, the latest posterior is propagated in time through the measurements from our introspective modality. In contrast to the EKF that uses the Jacobian of the motion and measurement model to propagate uncertainties, here we employ the unscented transform to achieve this goal. As described in the previous section, first we generate the augmented sigma points and propagate them through the system motion model:

$$\chi_t^{(i)} = f\left(\chi_{\text{aug},t-1}^{(i),x}, u_t, \chi_{aug,t-1}^{(i),q}\right) \quad i = 0, \dots 2n' \tag{17}$$

Here $\mathbf{u}_t$ encodes the IMU measurements into a single vector. Then, we compute the prediction mean and covariance using Eq.14, Eq.15 to get $\bar{\boldsymbol{\Sigma}}$ $\bar{\boldsymbol{\mu}}$. The bar on the state mean and covariance indicate that they are predicted values from interoceptive measurements up to the current moment and the exteroceptive measurements up to the previous timestep. Note that in discretizing the dynamics, we multiply the noise covariance values by the sampling time Δt, and the bias dynamics are propagated as $\mathbf{x}_{bias,t} = \mathbf{x}_{bias,t-1} + \mathbf{n}_{bias,d}$ where $\mathbf{n}_{bias,d} \sim \mathcal{N}(\mathbf{0}, \Delta t \mathbf{n}_b)$.

Update

Part1

The next step in the Kalman filter is to update our prediction based on the observations received from our measurements. By doing the update step, our sense of the system states statistically approaches the actual values. For the first part of the project, the measurement model is linear and the update procedure is the same as the case for the EKF:

$$\begin{aligned}
\boldsymbol{\mu}_t &= \bar{\boldsymbol{\mu}}_t + \mathbf{K}_t \left(\mathbf{z}_t - g\left(\bar{\boldsymbol{\mu}}_t, 0\right)\right) \\
\boldsymbol{\Sigma}_t &= \bar{\boldsymbol{\Sigma}}_t - \mathbf{K}_t \mathbf{C} \bar{\boldsymbol{\Sigma}}_t \\
\mathbf{K}_t &= \bar{\boldsymbol{\Sigma}}_t \mathbf{C}^T \left(\mathbf{C} \bar{\boldsymbol{\Sigma}}_t \mathbf{C}^T + \mathbf{R}\right)^{-1}
\end{aligned} \tag{18}$$

where $\mathbf{z}_t$ and $\mathbf{R}$ are the measurement value and corresponding noise estimates. Intuitively, the Kalman gain $\mathbf{K}_t$ at each step is determined by how uncertain the filter is about the measurement ($\mathbf{R}$) and the prediction from the previous step ($\bar{\boldsymbol{\Sigma}}_t$).

Part2

The measurement model for the second part is no longer linear (Eq.9 and Eq.10) and we should again use the unconnected transformation in order to manipulate the uncertainties. Specifically, similar to the procedure we took for the prediction step, for either measurement model we create an augmented state defined as the stack of the state and noise variables and propagate it through the nonlinear measurements model as follows:

$$\mathcal{Z}_t^{(i)} = g\left(\chi_{aug,t}^{(i),x}, \chi_{aug,t}^{(i),v}\right) \quad i = 0, \dots 2n'' \tag{19}$$

Using these sigma points, we then compute the measurement mean $\mathbf{m}_Z$, covariance $\mathbf{S}$, and cross-covariance $\mathbf{C}$ using equations Eq.14,15, and 16. Finally, the updated equations are formulated as follows:

$$\begin{aligned}
\boldsymbol{\mu}_t &= \overline{\boldsymbol{\mu}}_t + \boldsymbol{K}_t \left(\mathbf{z}_t - \mathbf{m}_Z\right) \\
\boldsymbol{\Sigma}_t &= \overline{\boldsymbol{\Sigma}}_t - \boldsymbol{K}_t \boldsymbol{S}_t \boldsymbol{K}_t^T \\
\boldsymbol{K}_t &= \boldsymbol{C}_t \boldsymbol{S}_t^{-1}
\end{aligned} \tag{20}$$

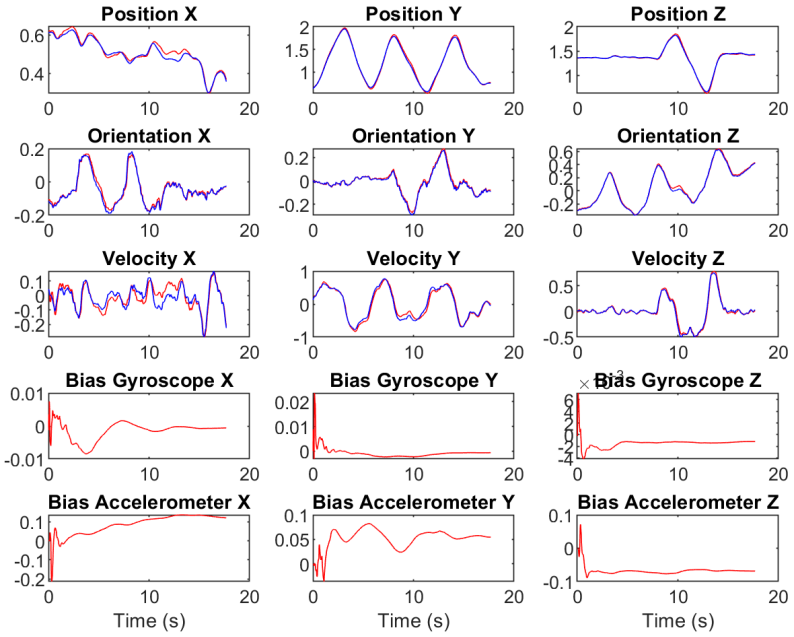

Figure 1: Results of part 1 for dataset 1 where the UKF is updated by the pose and position measurements from the aprigTag tracker. The blue curves show the actual values of states while the red curves show the predicted values. The UKF estimator is clearly converging to the actual values using measurements.

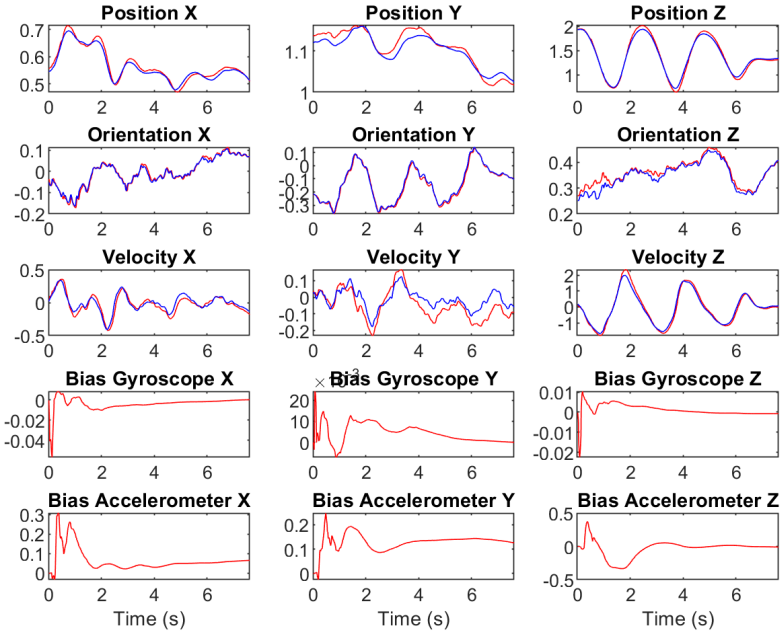

Figure 2: Results of part 1 for dataset 4 where the UKF is updated by the pose and position measurements from the aprigTag tracker. The blue curves show the actual values of states while the red curves show the predicted values. The UKF estimator is clearly converging to the actual values using measurements.

Project Results

Part1

The results of the first part of the project are shown in Fig.1 to 2. As shown in the figures the actual and predicted values of the system states are close to each other which shows the system is observable. It was predictable as in this

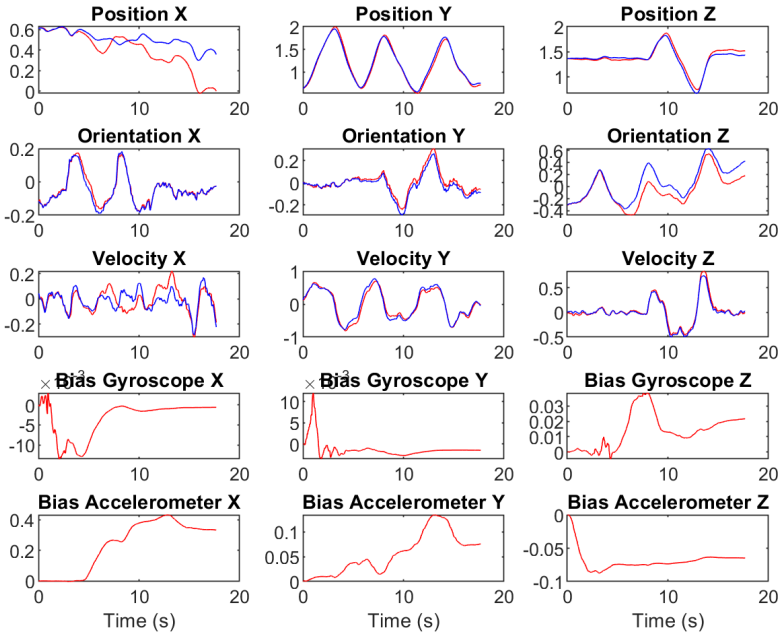

Figure 3: Results of part 2 for dataset 1 where the UKF is updated by velocity measurements obtained from the optical flow. The blue curves show the actual values of states while the red curves show the predicted values. The UKF estimator is clearly converging to the actual values using measurements.

part we have measurements from both the translation and orientation of the drone. Also, it can be inferred that the fluctuation in IMU sensor biases is small and not sharp in time which makes it possible for the filter to use the IMU measurements effectively.

Part2

In the second part of the project, the filter is updated only using the measured velocity by the optical flow and mapped to the world-frame velocity through measurement models Eq.9, 10. This scenario is more challenging to handle as the inference of the full pose through linear velocity requires the trajectory to be suitably excited.

Computing the Adjoint Using Camera

Fig.3 to 4 shows the result of fusing the IMU data with linear velocity obtained from the optical flow system (Eq.10). In this section, the body angular rate for computing the adjoint map is also taken from the optical flow sensor. As it can be seen, as opposed to the previous part, the estimation accuracy is not as high because the fusion of the onboard sensor is carried out in the world frame while both sensors measure their quantities in the local-body frame and do so only through the linear velocity as the measurement. It seems that this system has observability issues and yaw drift which is clear from the plots.

Computing the Adjoint Using Gyroscope

Fig.5 to 6 shows the result of fusing the IMU data with linear velocity obtained from the optical flow system and mapped to the body frame through the angular velocity measured by the gyroscope (Eq.9). As can be seen, since the gyroscope provides a more accurate and less noisy estimate of the angular velocity, the drift is not as bad as in the previous section. However, the latent root of it is still in place and we observe slight drifts throughout time. Finally, similar to the previous part, the bias terms are mostly stable for both measurement models. The fluctuations at some points are mandatory for finding the right orientation states as they are the only degrees of freedom in the filter that can modify the output of the IMU to match the observations.

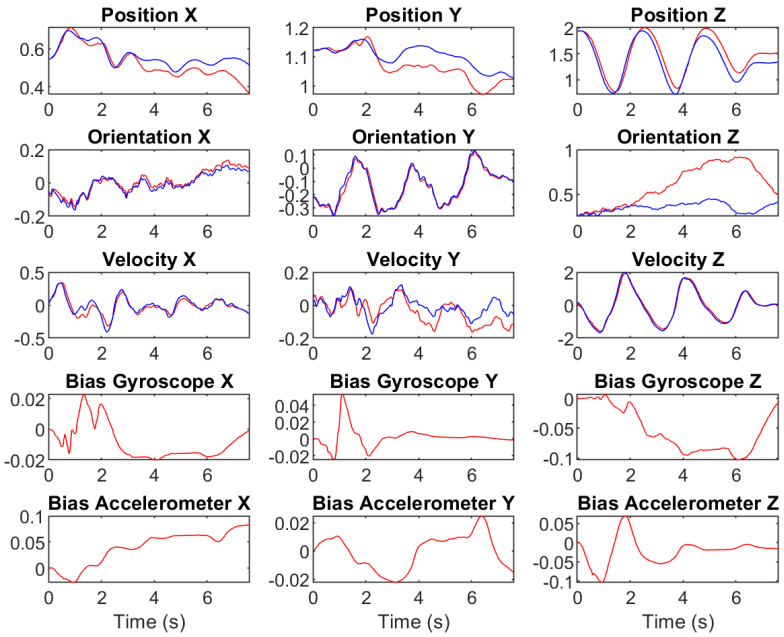

Figure 4: Results of part 2 for dataset 4 where the UKF is updated by velocity measurements obtained from the optical flow. The blue curves show the actual values of states while the red curves show the predicted values. The UKF estimator is clearly converging to the actual values using measurements.

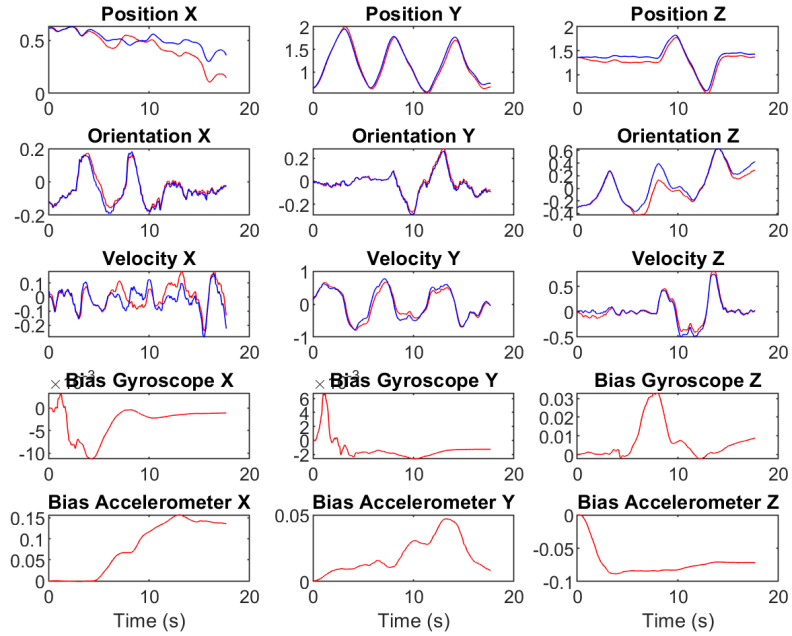

Figure 5: Results of part 2 for dataset 1 where the UKF is updated by velocity measurements obtained from the optical flow and mapped to the body frame through the gyroscope measurements. The blue curves show the actual values of states while the red curves show the predicted values. The UKF estimator is clearly converging to the actual values using measurements.

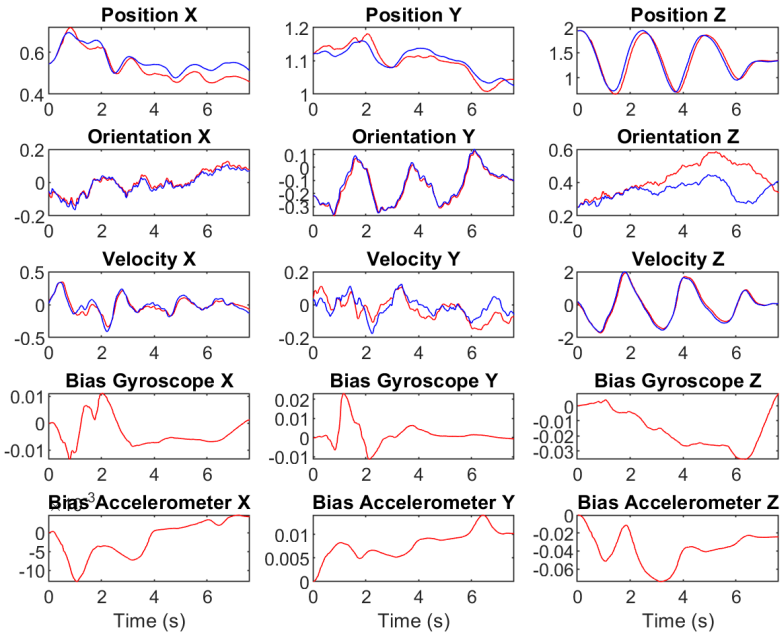

Figure 6: Results of part 2 for dataset 4 where the UKF is updated by velocity measurements obtained from the optical flow and mapped to the body frame through the gyroscope measurements. The blue curves show the actual values of states while the red curves show the predicted values. The UKF estimator is clearly converging to the actual values using measurements.