# Report

# Project 1

Submitted by: Krishnesh Meratia

N12145709

## Abstract:

Extended Kalman filter is implemented for state estimation of a Micro Aerial Vehicle. The EKF uses the data from the IMU and Vicon as input and updates them online for the correct estimates of the states. In part 1 state estimation (pose & orientation) and part 2 state estimation (velocity) is done taking the input as body frame acceleration and angular velocity from the onboard IMU and measurement is given by the pose or velocity from the Vicon.

## Introduction:

Robots have to be able to accommodate the enormous uncertainty that exists in the physical world. Uncertainty are caused by sensors, actuator, robot software, internal models etc. State estimation addresses the problem of estimating quantities from sensor data that are not directly observable, but that can be inferred. To overcome uncertainty and accurate tracking of state we apply probabilistic robotics the idea of estimating state from sensor data, as it is important is several applications.

Probabilistic estimation techniques use a model for predicting aspects of the time behavior of a system (dynamic model) and a model of the sensor measurements (measurement model). Kalman filter is a tool for elegantly combining multisensory fusion, filtering, and motion prediction in a single fast and accurate framework. Since velocity and orientation estimation are non-linear problems, EKF is used.

A Euler Angle based formulation of the EKF is developed, where the angular velocity and acceleration are considered as control inputs corrupted by noise and bias. The non-linear measurement equations are formulated by rotating the reference angular velocity vector in the world-frame using the relation between the estimated orientation and angular velocities and the rotating the body frame angular velocity vector in the world frame using the estimated orientation matrix.

## Background:

The IMU consists of a tri-axial accelerometer, a tri-axial gyro, and a tri-axial magnetic sensor. The accelerometer measures the difference between the acceleration of the vehicle and gravitational acceleration. The gyroscope measures the angular velocity in the sensor frame. IMU frame is coincident with that of Micro Aerial Vehicle. The data we receive from IMU is Struct format

The gyroscope gives us a noisy estimate of angular velocity '$\omega_m$'

$\omega_m = \omega + b_g + n_g$

where $b_g$ = gyroscope bias $\dot{b}_g = n_{bg} \sim N(0, Q_g)$ and $n_g \sim N(0, \Sigma_g)$

The accelerometer gives us a noisy estimate of acceleration '$a_m$'

$a_m = R(q)^T(\ddot{p} - g) + b_a + n_a$

where $b_a$ = gyroscope bias $\dot{b}_a = n_{ba} \sim N(0, Q_a)$ and $n_a \sim N(0, \Sigma_a)$

The Vicon gives us velocity and position, orientation and velocity in world frame angular velocity in body frame. The Vicon data is stored in two matrix variables, time and Vicon. The time variable contains the timestamp while the Vicon variable contains the Vicon data in the following format:

$$[x, y, z, roll, pitch, yaw, v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T$$

Rotation matrices are used to change the vector from one frame to another
The Z-X-Y Euler angle parametrization of SO (3) is used for orientation
$q = [\phi \quad \theta \quad \varphi]^T = [\text{roll} \quad \text{pitch} \quad \text{yaw}]^T$

$R(q) = R_y(\varphi)\, R_x(\theta)\, R_z(\phi)$

R(q) =

$$\begin{bmatrix} \cos(\varphi)\cos(\theta) - \sin(\theta)\sin(\varphi)\sin(\phi) & -\cos(\phi)\sin(\varphi) & \cos(\varphi)\sin(\theta) + \cos(\theta)\sin(\phi)\sin(\varphi) \\ \cos(\theta)\sin(\varphi) + \cos(\varphi)\sin(\phi)\sin(\theta) & \cos(\phi)\cos(\varphi) & \sin(\varphi)\sin(\theta) - \cos(\varphi)\cos(\theta)\sin(\phi) \\ -\cos(\phi)\sin(\theta) & \sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix}$$

We use it to change the acceleration from the body frame of the IMU to world frame
$\ddot{p} = R(q)(a_m - b_a - n_a) + g$

For moving the angular velocity from IMU to the world frame:

$[\omega_x\ \omega_y\ \omega_z]^T = [\cos(\theta)\ 0\ \sin(\theta)]^T\, \dot{\phi}$
$[\omega_x\ \omega_y\ \omega_z]^T = [0\ 1\ 0]^T\, \dot{\theta}$
$[\omega_x\ \omega_y\ \omega_z]^T = [-\cos(\phi)\sin(\theta)\ \sin(\phi)\ \cos(\theta)\cos(\phi)]^T\, \dot{\varphi}$

$$\omega_m = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & -\cos(\phi)\sin(\theta) \\ 0 & 1 & \sin(\phi) \\ \sin(\theta) & 0 & \cos(\theta)\cos(\phi) \end{bmatrix} * \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix}$$

$\dot{q} = G(q)^{-1}(\omega_m - b_g - n_g)$

The data given to us has been resampled to reduce the rate to around 30 Hz.

# Filters

Bayes Filter:
Goal is to update the probability distribution of the robot pose using the realization of the control input and measurement. We use last pose of the robot which is probability
Prior = $p(x_0)$
Make a process model which uses the control input applied and last state to find the process
Process = $p(x_t | x_{t-1}, u_t)$
which is a conditional probability. And the measurement model as
Measurement = $p(z_t | x_t)$
which is a conditional probability. We use these to get the prediction step
Prediction step = $p(x_t | z_{t-1}, u_t) = \int p\,(x_y | x_{t-1}, u_t)\, p\,(x_{t-1} | z_{1:t-1}, u_{1:t})\, dx_{t-1}$
using the process model and the update step
Update step = $p(x_t | z_t, u_t) = \eta p(z_t | x_t) p(x_t | z_{1:t-1}, u_{1:t})$
using the prediction step and the measurement model.

Kalman Filter:
We use the same steps as in Bayes Filter but make a few assumptions, that the prior state of the robot is represented by a Gaussian distribution $p(x_0) \sim N(\mu_0, \Sigma_0)$. The process model $p(x_t | x_{t-1}, u_t)$ is linear with additive Gaussian white noise $x_t = A_t x_{t-1} + Bt\, u_t + n_t$ where $n_t \sim N(0, Q_t)$ and the measurement model is linear with additive Gaussian white noise $z_t = C_t x_t + v_t$ where $v_t \sim N(0, R_t)$. We use these and facts of probability to get the prediction step and update step

prior $p(x_{t-1} | z_{t-1}, u_{t-1}) \sim N(\mu_{t-1}, \Sigma_{t-1})$
prediction is $\bar{\mu}_t = A\,\mu_{t-1} + B\,u_t$ $\bar{\Sigma}_t = A\Sigma_{t-1}A^T + Q$
update is $\mu_t = \bar{\mu}_t + K_t(z_t - C\bar{\mu}_t)$ $\Sigma_t = \bar{\Sigma}_t - K_t C\bar{\Sigma}_t$ $K_t = \bar{\Sigma}_t C^T (C\,\bar{\Sigma}_t\,C^T + R)^{-1}$
where A, B and C are matrices found from the dynamical equations $\mu_{t-1}$is the mean of the previous state and $u_t$ is the control input $\Sigma_{t-1}$ is the covariance of the previous state and Q and Rare covariance of noise.

Extended Kalman Filter:
In this we need to linearize the dynamic model as it is non- linear, it’s is implemented below for the given problem
EKF linearizes a model that is nonlinear and updates the states
The states that we need to measure are:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} \text{positon} \\ \text{orientation} \\ \text{velocity} \\ \text{gyroscope bias} \\ \text{accelerometer bias} \end{bmatrix} \mathcal{E} R^{15}$$

To get the acceleration and angular velocity in world frame
The angular velocity in body frame is given by:

$$\omega_m = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & -\cos(\phi)\sin(\theta) \\ 0 & 1 & \sin(\phi) \\ \sin(\theta) & 0 & \cos(\theta)\cos(\phi) \end{bmatrix} * \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = G(q)\dot{q}$$

The gyroscope gives us a noisy estimate of angular velocity ‘$\omega_m$’
Angular velocity in world frame is:
$\dot{q} = G(q)^{-1}(\omega_m - b_g - n_g) = G(x_2)^{-1}(\omega_m - x_4 - n_g)$
The acceleration in body frame is given by:
$a_m = R(q)^T(\ddot{p} - g) + b_a + n_a$
where $b_a$ = gyroscope bias $\dot{b}_a = n_{ba} \sim N(0, Q_a)$ and $n_a \sim N(0, \Sigma_a)$
acceleration in world frame is:
$\ddot{p} = R(q)(a_m - b_a - n_a) + g = R(x_2)(a_m - x_5 - n_a)$

Implementation of EKF:
The prior state of the robot is represented by a Gaussian distribution
$p(x_0) \sim N(\mu_0, \Sigma_0)$

The continuous time process model is:
$\dot{x} = f(x, u, n)$

$$\dot{x} = \begin{bmatrix} \text{velocity} \\ \text{angular velocity} \\ \text{acceleration} \\ n_{bg} \\ n_{ba} \end{bmatrix} = \begin{bmatrix} x_3 \\ G(x_2)^{-1}(\omega_m - x_4 - n_g) \\ R(x_2)(a_m - x_5 - n_a) + g \\ n_{bg} \\ n_{ba} \end{bmatrix}$$

The process model is nonlinear
We linearize the dynamics about $x = \mu_{t-1}$, $u = u_t$, $n = 0$

$$\dot{x} \approx f(\mu_{t-1}, u_t, 0) + \frac{\partial f}{\partial x}\Big|_{(\mu_{t-1}, U_t, 0)}(x - \mu_{t-1}) + \frac{\partial f}{\partial u}\Big|_{(\mu_{t-1}, U_t, 0)}(u - u_t) + \frac{\partial f}{\partial n}\Big|_{(\mu_{t-1}, U_t, 0)}(n - 0)$$

Let:

$A_t = \frac{\partial f}{\partial x}\Big|_{(\mu_{t-1}, U_t, 0)}$ $B_t = \frac{\partial f}{\partial u}\Big|_{(\mu_{t-1}, U_t, 0)}$ $U_t = \frac{\partial f}{\partial n}\Big|_{(\mu_{t-1}, U_t, 0)}$

Linear dynamics is:
$\dot{x} \approx f(\mu_{t-1}, u_t, 0) + A_t (x - \mu_{t-1}) + B_t(u - u_t) + U_t(n-0)$

From discrete time to continuous time using one step Euler integration:
$x_t \approx x_{t-1} + f(x_{t-1}, u_t, n_t)(\delta t) = x_{t-1} + (\delta t)(f(\mu_{t-1}, u_t, 0) + A_t (x_{t-1} - \mu_{t-1}) + B_t(u_t - u_t) + U_t(n-0))$
$x_t \approx (I + \delta t\, A_t)\, x_{t-1} + \delta t\, U_t\, n + \delta t(f(\mu_{t-1}, u_t, 0) - A_t\, \mu_{t-1}$

Prediction Step:
$\overline{\mu_t} = \mu_{t-1} + (f(\mu_{t-1}, u_t, 0)$ $\quad \overline{\Sigma_t} = F_t \Sigma_{t-1} F_t^T + U_t \delta t Q_t U_t^T$ where $F_t = I + \delta t\, A_t$
Calculating $A_t$ and $U_t$

$$A_t = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_{15}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_{15}} \end{bmatrix} \mathcal{E} R^{15*15} \quad U_t = \begin{bmatrix} \frac{\partial f}{\partial n_1} & \cdots & \frac{\partial f}{\partial n_{12}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial n_1} & \cdots & \frac{\partial f}{\partial n_{12}} \end{bmatrix} \mathcal{E} R^{12*12}$$

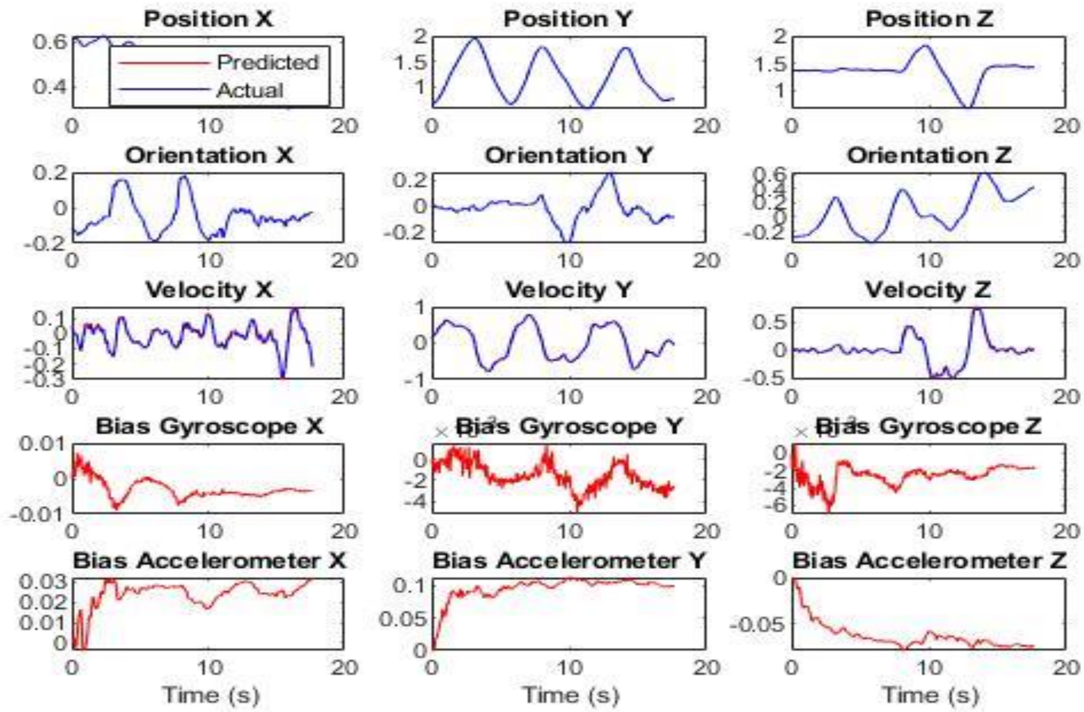
As the measurement model is linear the update step for Kalman Filter is used
Measurement Model:
$Z_t = C_t\, x_t + V_t$
Where $V_t$ is noise from Vicon $V_t \sim N(0, R_t)$
For Part 1:

$$Z = \begin{bmatrix} p \\ q \end{bmatrix} + V_t = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ \dot{p} \\ b_g \\ b_a \end{bmatrix} + V_t = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} positon \\ orientation \\ velocity \\ gyroscope\ bias \\ accelerometer\ bias \end{bmatrix}$$

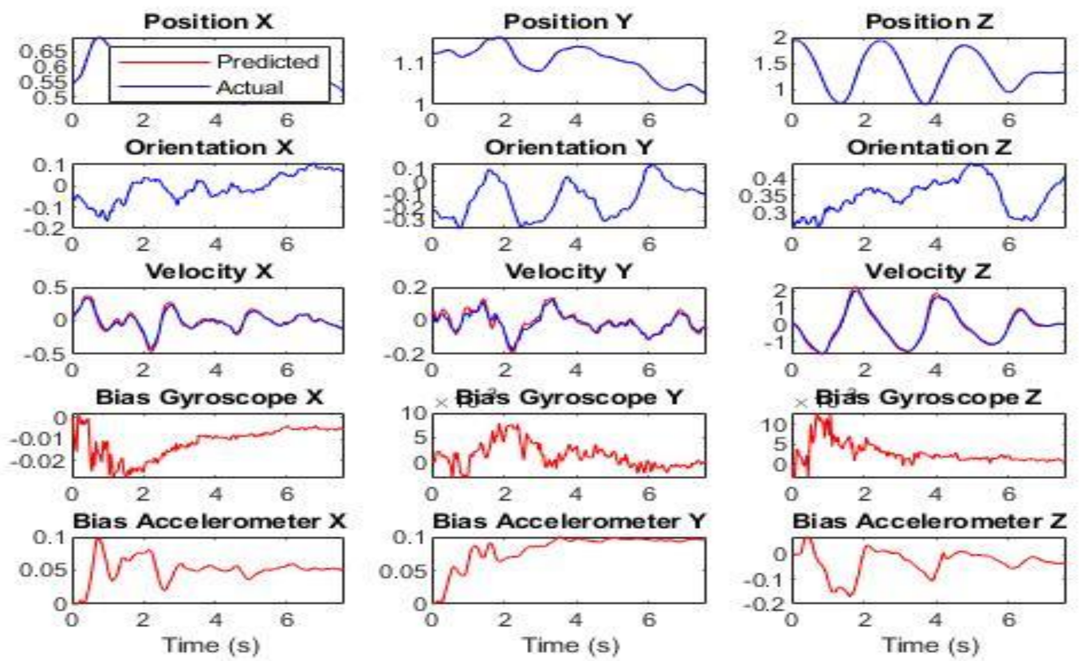
$$C_t = \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \end{bmatrix} \qquad R_t\ \mathcal{E}\ R^{6*6}$$

Update step:
$\mu_t = \overline{\mu_t} + K_t (z_t - C_t\, \overline{\mu_t})$
$\Sigma_t = \overline{\Sigma_t} - K_t\, C_t\, \overline{\Sigma_t}$
$K_t = \overline{\Sigma_t}\, C_t^T (C_t \overline{\Sigma_t}\, C_t^T + R_t)^{-1}$

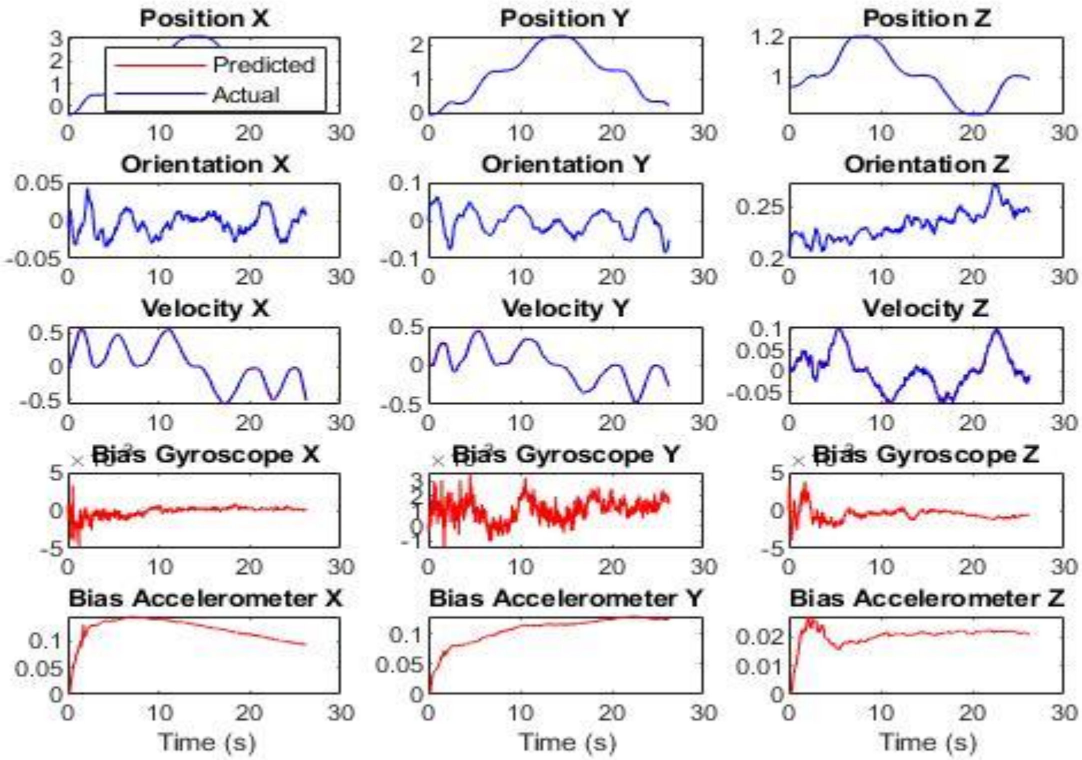
Results:
For Data set 1:




For Data set 4:



For Data set 9:




In part 1 the velocity is off because we are updating only the position and orientation.
For Part 2:

$$Z = [q] + V_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ \dot{p} \\ b_g \\ b_a \end{bmatrix} + V_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix} positon \\ orientation \\ velocity \\ gyroscope\ bias \\ accelerometer\ \text{bias} \end{bmatrix}$$

$$C_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathrm{I} & 0 & 0 \end{bmatrix} \qquad R_t \ \varepsilon\ R^{9*9}$$

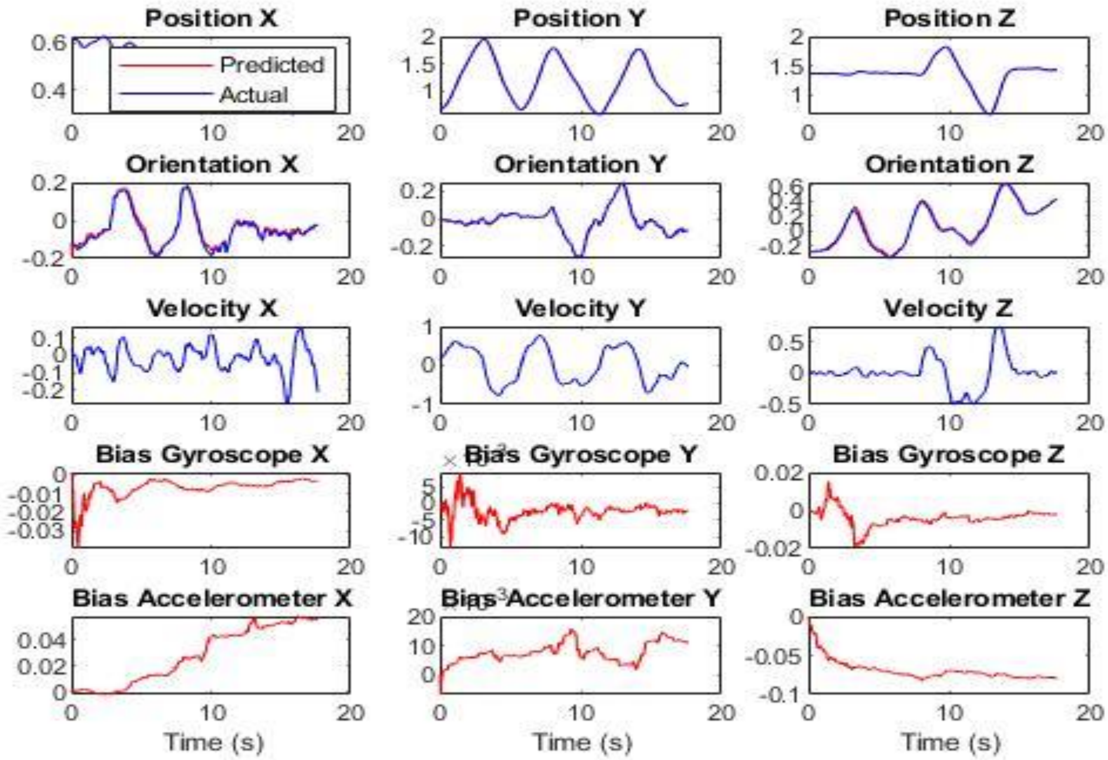
Update step:

$\mu_t = \bar{\mu}_t + K_t(z_t - C_t\,\bar{\mu}_t)$

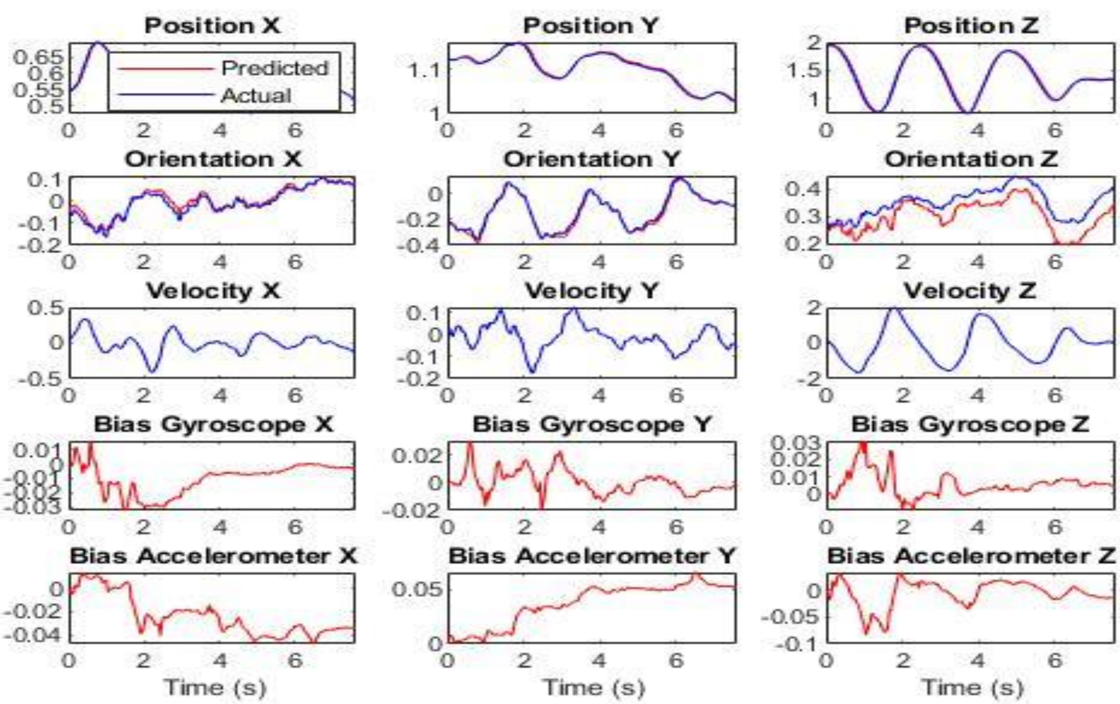
$\Sigma_t = \bar{\Sigma}_t - K_t\, C_t \bar{\Sigma}_t$

$K_t = \bar{\Sigma}_t\, C_t^T (C_t \bar{\Sigma}_t\, C_t^T + R_t)^{-1}$

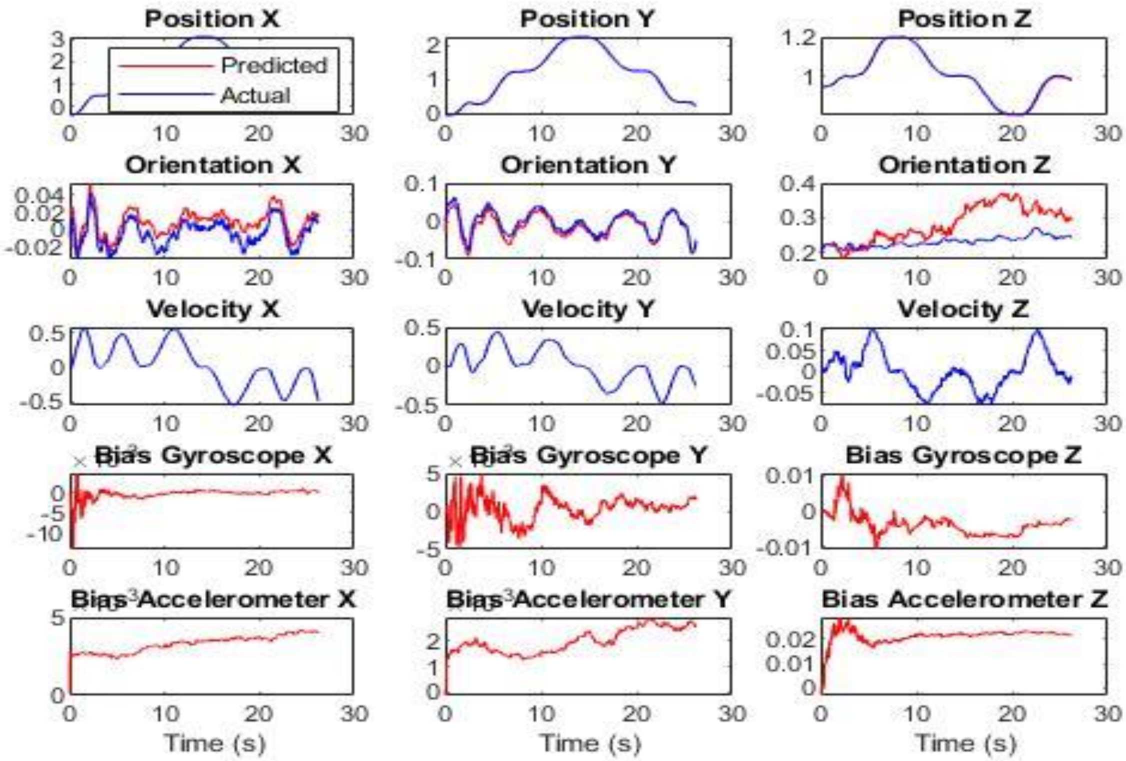
Results:
For Data set 1:




For Data set 4:



For Data set 9:




In part 2 position and orientation are off because we are updating only the velocity.