ROB-6213: Project 1

Name: Puda Zhao
studentID: N13625214
netID: pz2078

Part 1

Prediction Step

This part is in function pred_step in file pred_step.m:

Step1. Process Model

From input $\boldsymbol{\mu_{t-1}}$, $\boldsymbol{\omega}$, $\boldsymbol{a}$, we could build the process model with function G(q) and R(q)

Here we use Z-Y-X Euler Angle to Rotation Matrix.

$$G(q)^{-1} = \begin{bmatrix} cos\varphi/cos\theta & sin\varphi/cos\theta & 0 \\ -sin\varphi & cos\varphi & 0 \\ cos\varphi sin\theta/cos\theta & sin\theta sin\varphi/cos\theta & 1 \end{bmatrix}$$

$$R(q) = \begin{bmatrix} cos\theta cos\varphi & sin\psi sin\theta cos\varphi - cos\psi sin\varphi & cos\psi sin\theta cos\varphi + sin\psi sin\theta \\ cos\theta sin\varphi & sin\psi sin\theta sin\varphi + cos\psi cos\varphi & cos\psi sin\theta sin\varphi - sin\psi cos\theta \\ -sin\varphi & sin\psi cos\theta & cos\psi cos\theta \end{bmatrix}$$

$$\dot{x} = \begin{bmatrix} x_4 \\ G(x_2)^{-1}(\omega_m - x_4 - n_g) \\ R(q)(a_m - x_5 - n_a) \\ n_{bg} \\ n_{ba} \end{bmatrix}$$

Besides, it's also necessary to set n_g, n_a, n_{bg}, n_{ba}. Here I define $n \sim N(0, \sigma^2 = 0.01)$

Step2. Linearization

The hardest part in Extended Kalman Filter.

To get A_t and U_t, I set 'syms' in MATLAB to get the expression. The results are showed in MATLAB.

Step3. Discretization and Prediction

This part is simple to do. Follow these equations:

$$F_t = I + \delta_t A_t$$
$$V_t = U_t$$
$$Q_d = Q\delta_t$$

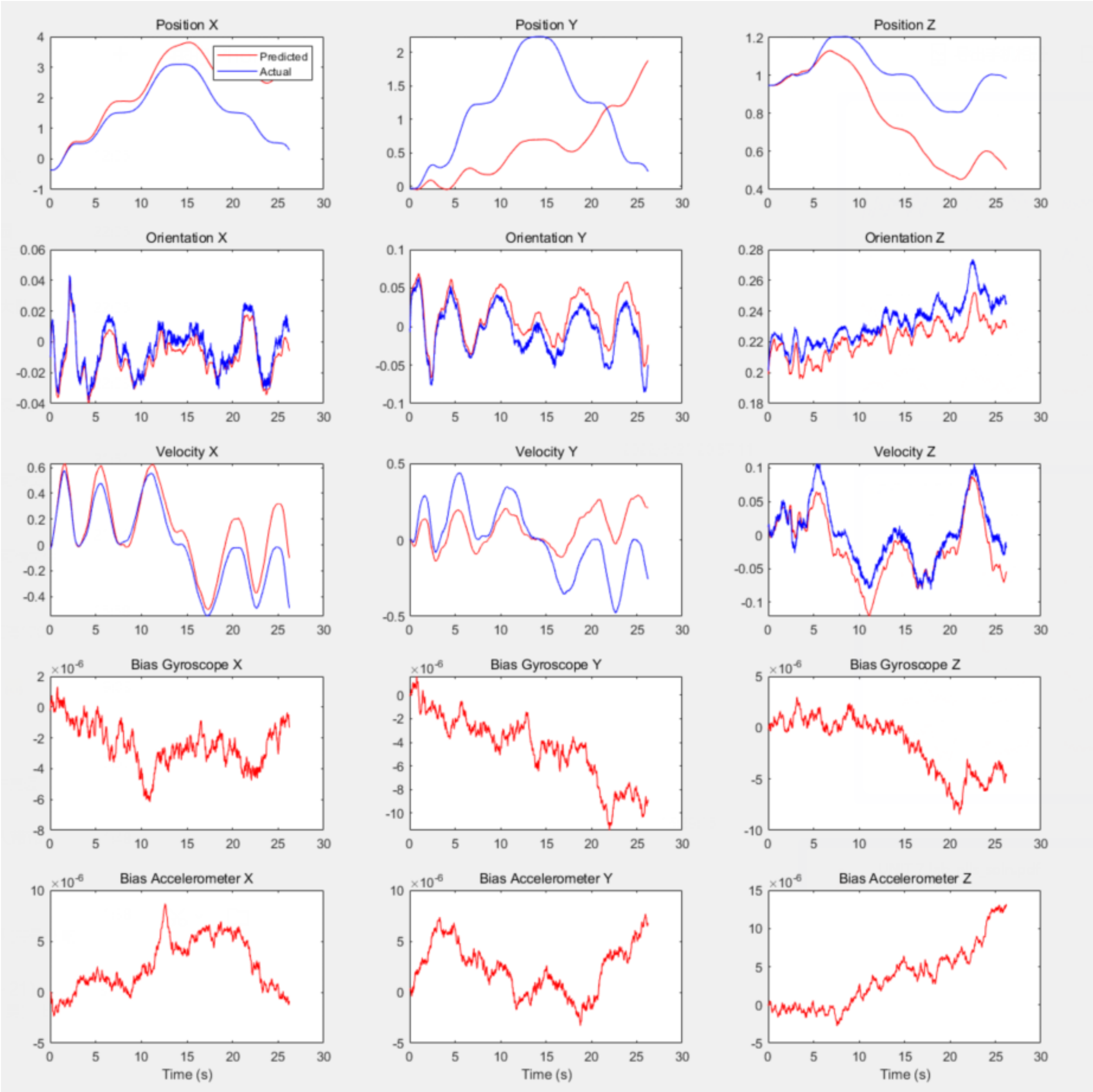
After this, we can do prediction:

$$\bar{\mu}_t = \mu_{t-1} + \delta_t f(\mu_{t-1}, u_t, 0)$$
$$\bar{\Sigma}_t = F_t \Sigma_{t-1} F_t^{\,T} + V_t Q_d V_t^{\,T}$$

If we **only use prediction part** to plot the result, it's like this:

You can see that the results are close but not accurate. So, we need update step to get better results.

The Kalman Gain K_t will decide how much we should believe between the estimate result and measurement result.



Update Step

This part is in function upd_step in file upd_step.m:

Step1. Assumptions and Linearization

The process is also not complex.

Because our Observation Model is simple. We can directly get that

$$C_t \text{ is a } 6 \times 15 \text{ identity matrix } W_t \text{ is a } 6 \times 6 \text{ identity matrix.}$$

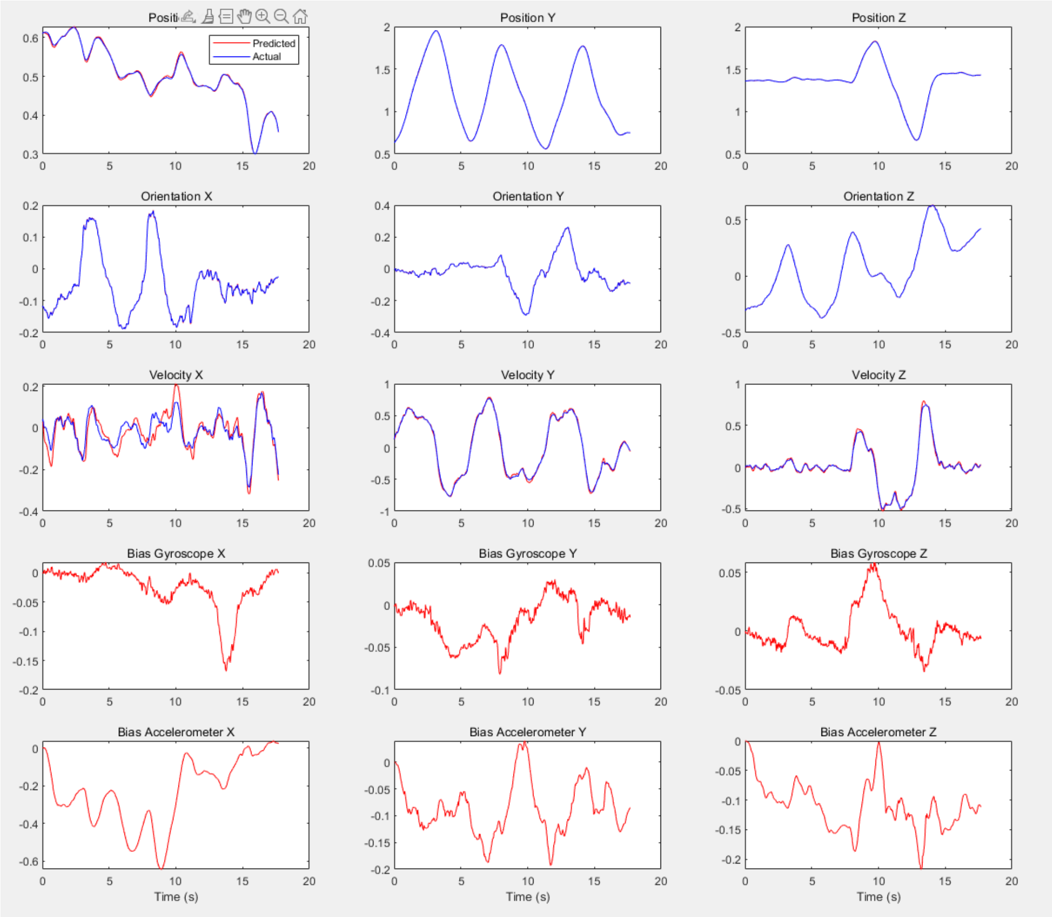
Step2. Update Step

$$K_t = \bar{\Sigma}_t {C_t}^T (C_t \bar{\Sigma}_t {C_t}^T + W_t R_t {W_t}^T)$$
$$\mu_t = \bar{\mu}_t + K_t C_t \bar{\Sigma}_t$$
$$\Sigma_t = \bar{\Sigma}_t - K_t (z_t - g(\bar{\mu}_t, 0))$$

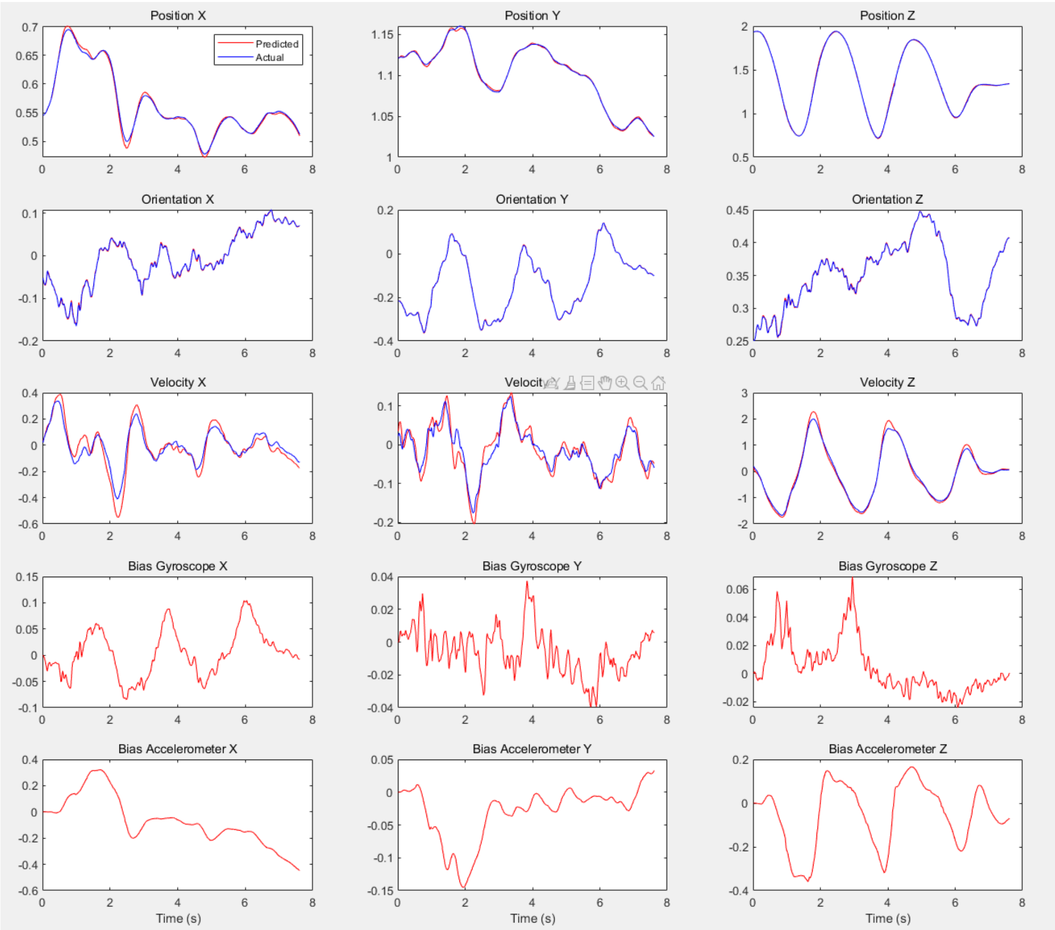
K_t is the Kalman-gain. The Kalman-gain is the weight given to the measurements and current-state estimate, and can be "tuned" to achieve a particular performance. With a high-gain, the filter places more weight on the most recent measurements, and thus conforms to them more responsively.

Plotting results:

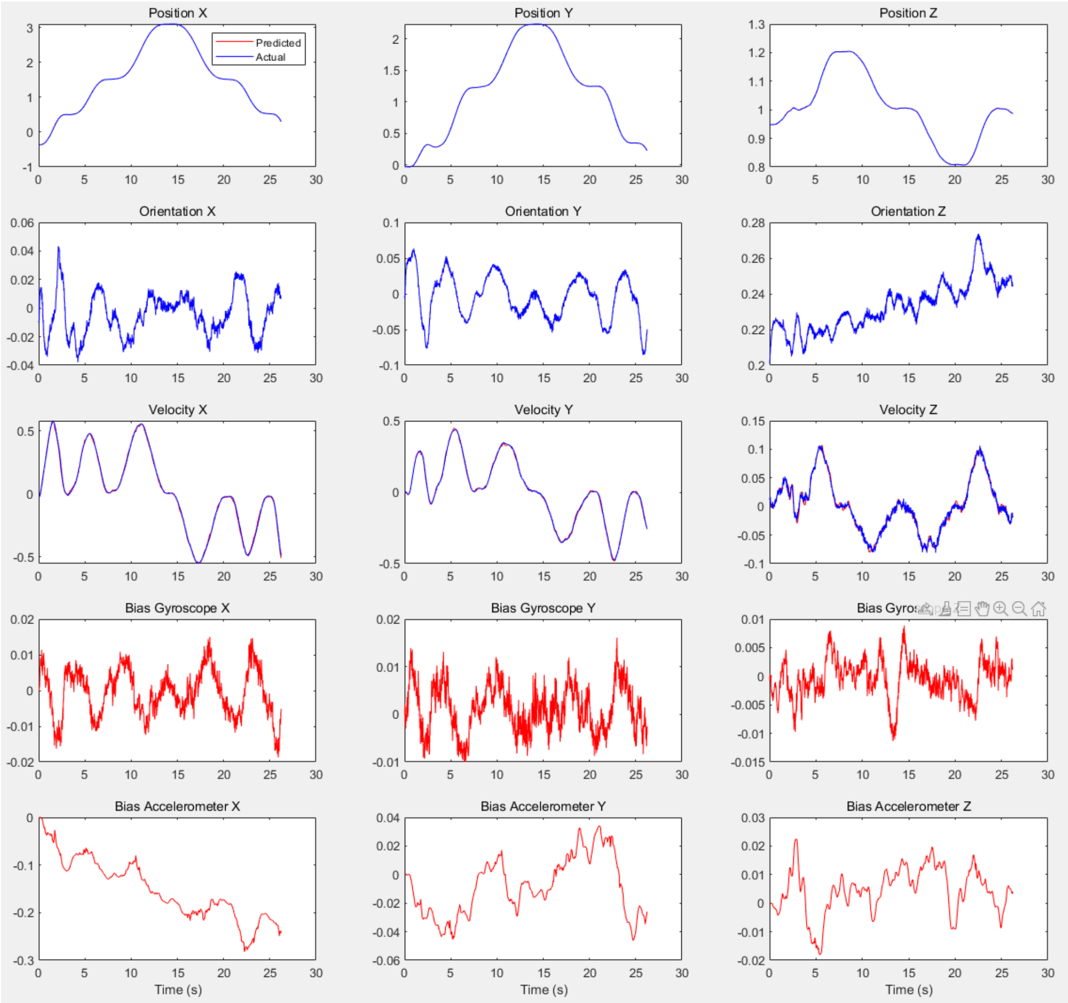
Datasheet 1:



Datasheet 4:



Datasheet 9:



Position X
Predicted
Actual
Position Y
Position Z
Orientation X
Orientation Y
Orientation Z
Velocity X
Velocity Y
Velocity Z
Bias Gyroscope X
Bias Gyroscope Y
Bias Accelerometer X
Bias Accelerometer Y
Bias Accelerometer Z
Time (s)
Time (s)
Time (s)

Part 2

Prediction Step

This part is the same with part1.

Update Step

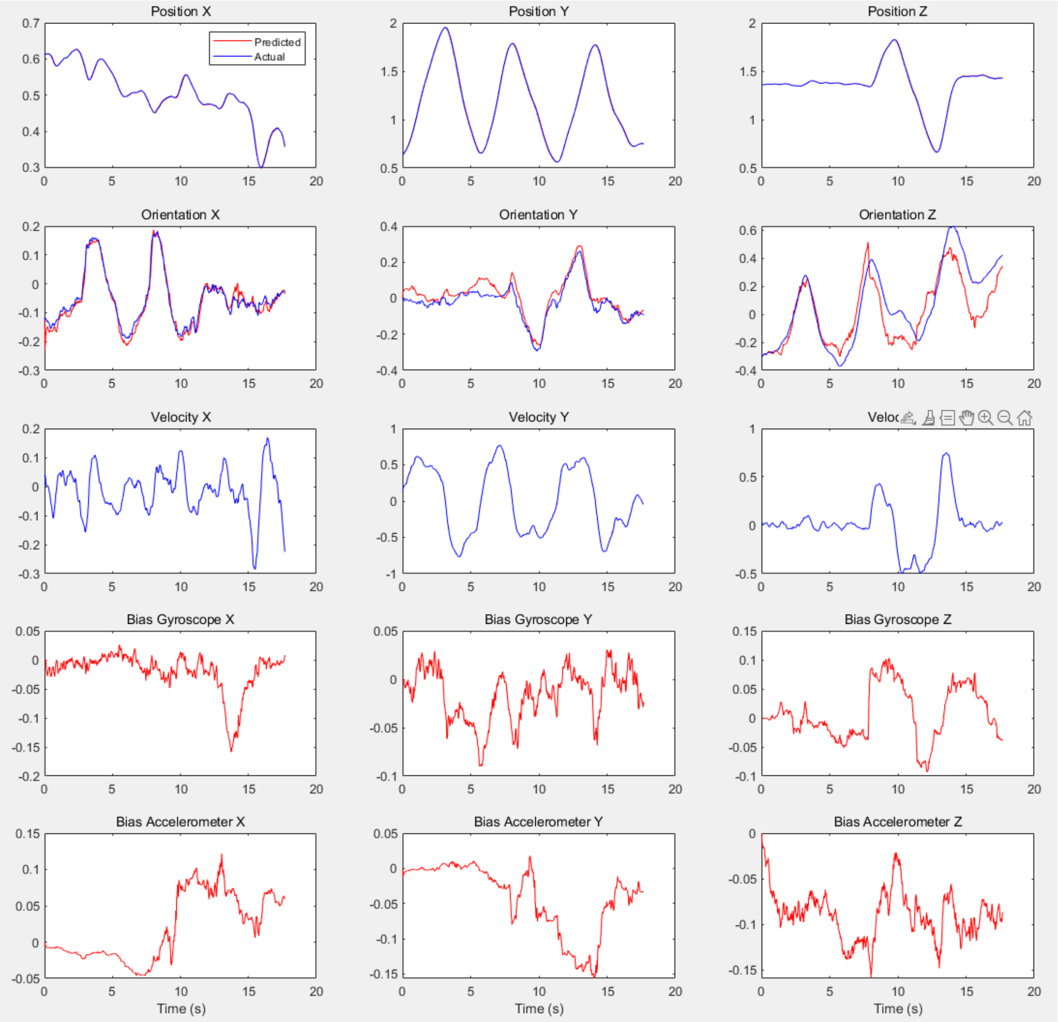
Because this time, we use observation models for the velocity. So, we need to modify C_t.

$$C_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

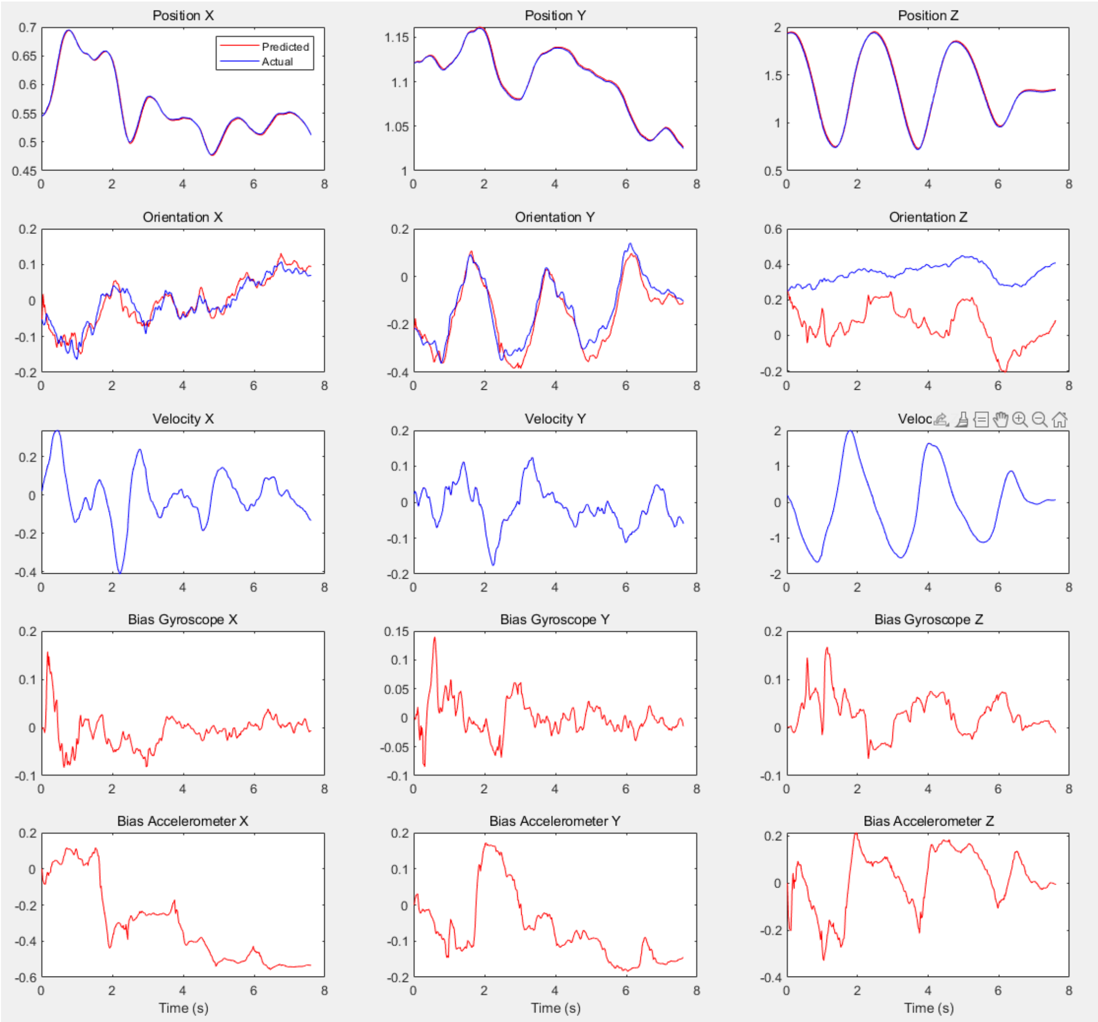
Right this time, W_t *becomes a* 3×3 *identity matrix*.

Plotting results:

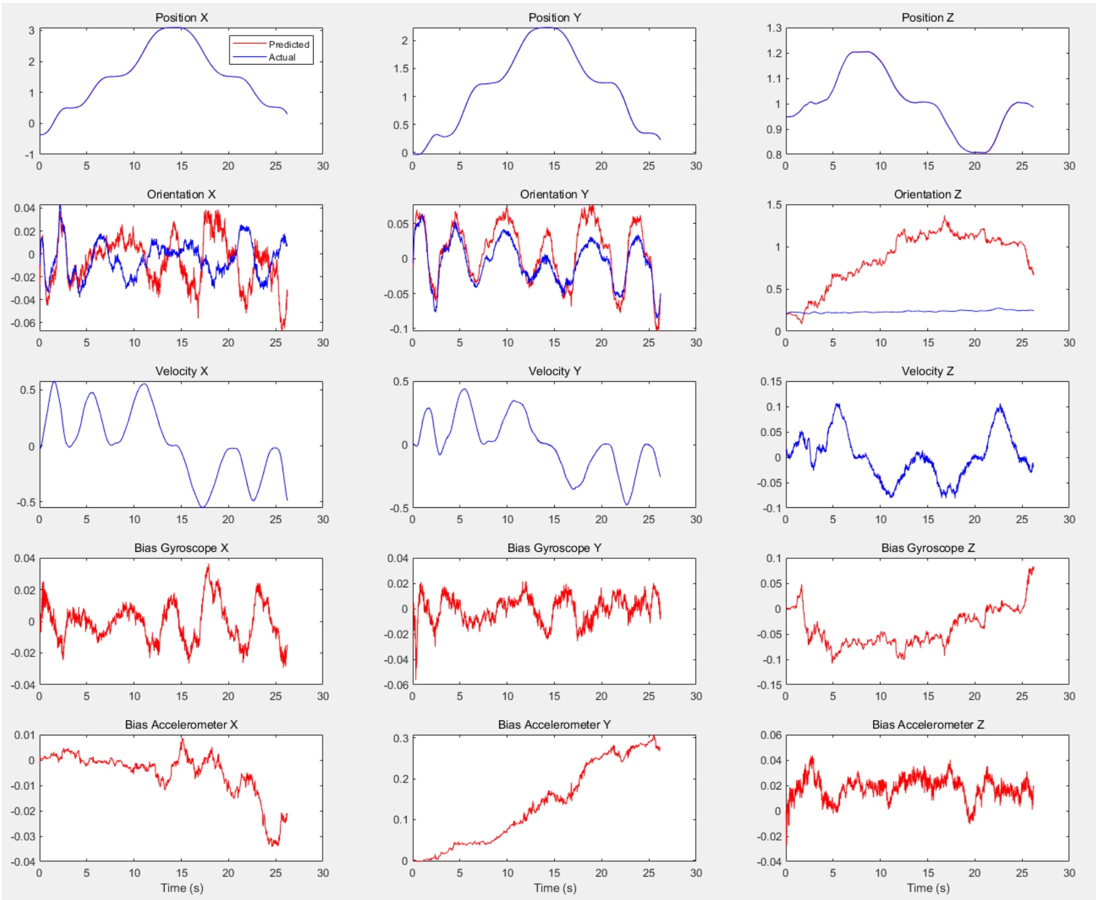
Datasheet 1:



Datasheet 4:



Datasheet 9:



Conclusion

In part1, we can see that the curve overlap is very high, and in part2 we see the orientation part, the predicted curve and the actual curve are quite different. This is because the observation model we used in part 2 only contains the linear velocity and cannot correct the direction, so the plotted prediction image is only the result of the prediction step, and the error is large.