New York University

Extended Kalman Filter for State Estimation of Robots

ROBOT LOCALISATION AND NAVIGATION

ROB-6213 Spring 2021

PROJECT-1

Professor: Giuseppe Loianno

Student Name: Rishi Eswar Varma Vegesna (N17479594)

Student Netid: rv2210

Student Name: - Rishi Eswar Varma Vegesna
Student Netid: - rv2210

INTRODUCTION:

The project was implementation of Extended Kalman filter for state estimation of a robot. The various state estimates given by the extended Kalman filter were position, velocity, orientation, gyroscopic bias and accelerometer bias. The given project consists of 2 parts Kalman filter part1 and Kalman filter part2. In Kalman filter part1 the measurement model update was given by Position and Orientation from Student Data(vicon). In the Kalman filter part2 the measurement model update was given by only velocity from student vicon data. The difference in this Measurement model update step causes change in C_t matrix values. The outline of the code for EKF was given and we need to update in Kalman Filter For loop, prediction step, and update step. These sections will be explained further in the report

KALMAN FILTER FUNCTION

In the project the main part was the Kalman filter .MAT file which is core of our project. This step is used to initialise the function, extract data from the student data given, calling Prediction step and update step functions. After calling the functions for update step and prediction step the current mean is saved in the savedstates to plot the values. And after saving the mean current to saved states the previous mean is also updated to current mean so that this value can be used as previous mean in the next iteration of the program. The equations used in the Kalman filter step are as follows

PROCESS MODEL AND PREDICTION STEP

This function calculates estimated mean and estimated covariance required for the update step. The inputs of this function are mean previous, angular velocity, previous covariance, acceleration, sample time. In the prediction step we need to calculate X (15x1) matrix and X_dot (15x1) matrix. The state matrix X is a 15x1 vector:

$$\mathbf{X= [\ p\ q\ p\_dot\ \ b_g\ b_a\ ]^T}$$

$$\mathbf{X=[\ x\ y\ z\ roll\ pitch\ yaw\ v_x\ v_y\ v_z\ b_{g1}\ b_{g2}\ b_{g3}\ b_{a1}\ b_{a2}\ b_{a3}]^T}$$

Whereas X_dot is

$$\mathbf{X\_dot= [\ X_3\ \ G(X_2)^{-1}R(w_m-X4-n_g)\ \ g+R(X_2)(a_m-X_5-n_a)\ \ n_{bg}\ \ n_{ba}]^T = f(\mu_{t-1},u_t,0)}$$

Where,

X3= [Vx Vy Vz]

$G(X_2)^{-1}=$ $\begin{pmatrix} \frac{\cos(\text{yaw})}{\cos(\text{pitch})} & \frac{\sin(\text{yaw})}{\cos(\text{pitch})} & 0 \\ -\sin(\text{yaw}) & \cos(\text{yaw}) & 0 \\ \frac{\cos(\text{yaw})\sin(\text{pitch})}{\cos(\text{pitch})} & \frac{\sin(\text{pitch})\sin(\text{yaw})}{\cos(\text{pitch})} & 1 \end{pmatrix}$

$W_m= [\ W_x\ W_y\ W_z]^T$

$a_m= [\ a_x\ a_y\ a_z]^T$

$X_{4=}$ [b_{g1} b_{g2} b_{g3}] ^{T}

X_5= [b_{a1} b_{a2} b_{a3}] ^{T}

In X_dot matrix we also have to calculate $G(X_2)$ and $G(X_2)^{-1}$ matrices. The G matrix depicts Euler angles of rotation matrix to Angular velocities whereas $G(X_2)^{-1}$ depicts angular velocities to Euler Rotation angles. Using X_dot and X we can calculate A_t by partial differentiating X_dot with X. To do this simply a jacobian function can be used inside the prediction step but using a Jacobian function will slow the processing speed of the program. To avoid this the jacobian matrix was calculated in a separate MATLAB live script and the output matrix A_t (15x15) was manually entered in the prediction step. After calculating A_t, V_t should be calculated V_t was a 15x15 matrix when X_dot was partially differentiated with respect to gaussian noises. To speed up the processing speed of the program V_t matrix was also calculate in a separate MATLAB script and the output matrix was manually entered into the prediction step. As the At and Vt were too complicated simplify function in MATLAB was used to Simplify the matrices. The A_t and V_t matrices were calculated as our system was non-linear so we have to Linearise the model first so the state variables A_t and U_t were calculated for linearisation of the model.

$$\mathbf{A_t} = \frac{\boldsymbol{df}(\boldsymbol{\mu t-1, ut, 0})}{\boldsymbol{dx}}$$

$$\mathbf{U_t} = \frac{\boldsymbol{df}(\boldsymbol{\mu t-1, ut, 0})}{\boldsymbol{dn}}$$

After calculating A_t we have to calculate F_t the formula for calculating F_t was:

$$\mathbf{F_t = I + dt * A_t}$$

Where $\mathbf{I}$ was an identity matrix of order 15*15, dt was sample time. Now Qd was Calculated, Q was an identity matrix of order 15* 15 identity matrix Q_d was given by the formula

$$\mathbf{Q_d = Q * dt}$$

F_t, U_t, and Q_d were calculated to discretise the model i.e., to convert the continuous step model into discrete system model. After calculating the state variables A_t, F_t, Q_d, Vt, and U_t they were used to calculate mean estimate and covariance estimate which were given as outputs of this function to Kalman filter function. The formulas for mean and covariance estimation were as follows

```
uEst=uPrev+(dt*X_dot)
covarEst= (Ft*covarPrev*(Ft')) + (Vt*Qd*(Vt'))
```

This Process model and update step were same for both Kalman filter part1 and Kalman filter part2

MEASUREMENT MODEL AND UPDATE STEP

This function is used to calculate current mean and current covariance. The inputs for the function were Measurements Z_t, estimated mean and estimated covariance from prediction step. Initially in the measurement model we have calculate Ct and W_t to linearise the system. They were given by the formulas:

$$\mathbf{C_t} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \end{bmatrix} + \boldsymbol{v} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \\ \boldsymbol{p_{dot}} \\ \boldsymbol{bg} \\ \boldsymbol{ba} \end{bmatrix} + \boldsymbol{v} = \boldsymbol{CX} + \boldsymbol{v}$$

In this case since measurement model depends on position and velocity the C_t matrix is identity matrix with 6 rows and 15 columns. After C_t we have to calculate R which is also an identity matrix multiplied by a scalar. This scalar value was a constant which depends upon how much we believe in our sensor measurements. By changing R value, the accuracy of the model in gyroscopic bias and accelerometer bias can be adjusted.

In Kalman filter part2 since the measurement model depends only on velocity the order of C_t matrix changes to an identity matrix with 3 rows and 15 columns. The formula for C_t matix in Kalman filter part2 was as follows

$$\mathbf{C_t} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \end{bmatrix} + \boldsymbol{v} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \\ \boldsymbol{p_{dot}} \\ \boldsymbol{bg} \\ \boldsymbol{ba} \end{bmatrix} + \boldsymbol{v} = \boldsymbol{CX} + \boldsymbol{v}$$

In both Kalman filter part1 and Kalman filter part2 after calculating C_t, Kalman gain is calculated in the function by using the following formula:

K_t= (covarEst*C_t')/((C_t*covarEst*C_t') +R)

Once Kalman gain is calculated the current mean and current covariance were calculated in the update step function as follows:

```
uCurr= uEst+ (Kt*(z_t - (Ct*uEst)));
covar_curr= covarEst- (Kt*Ct*covarEst);
```

Then this mean current and covariance current will be given as output to update step function in Kalman filter function. Then these outputs are saved in savedstates variable which was used to plot the data in graphical format.

PLOTS AND RESULTS

There were total of six plots 3 datasets for each Kalman filter. The plots show differences between the predicted model actual model. The plots for the datasets are as follows

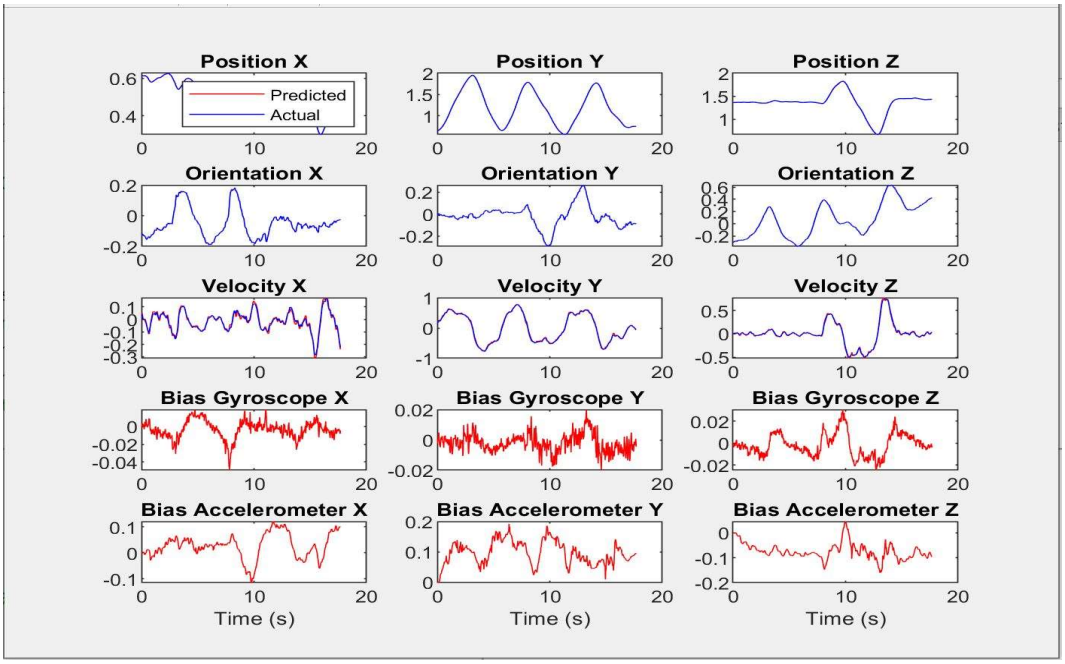

Figure 1: Kalman Filter Part1 Dataset 1

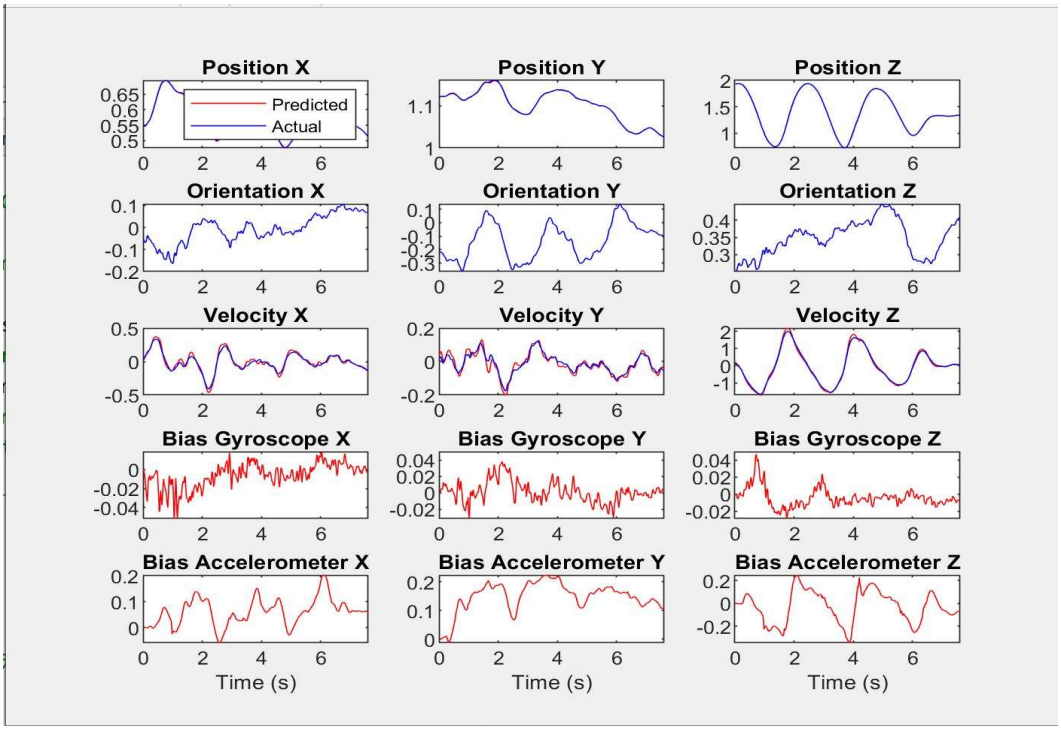

Figure 2: Kalman Filter Part1 Dataset 4

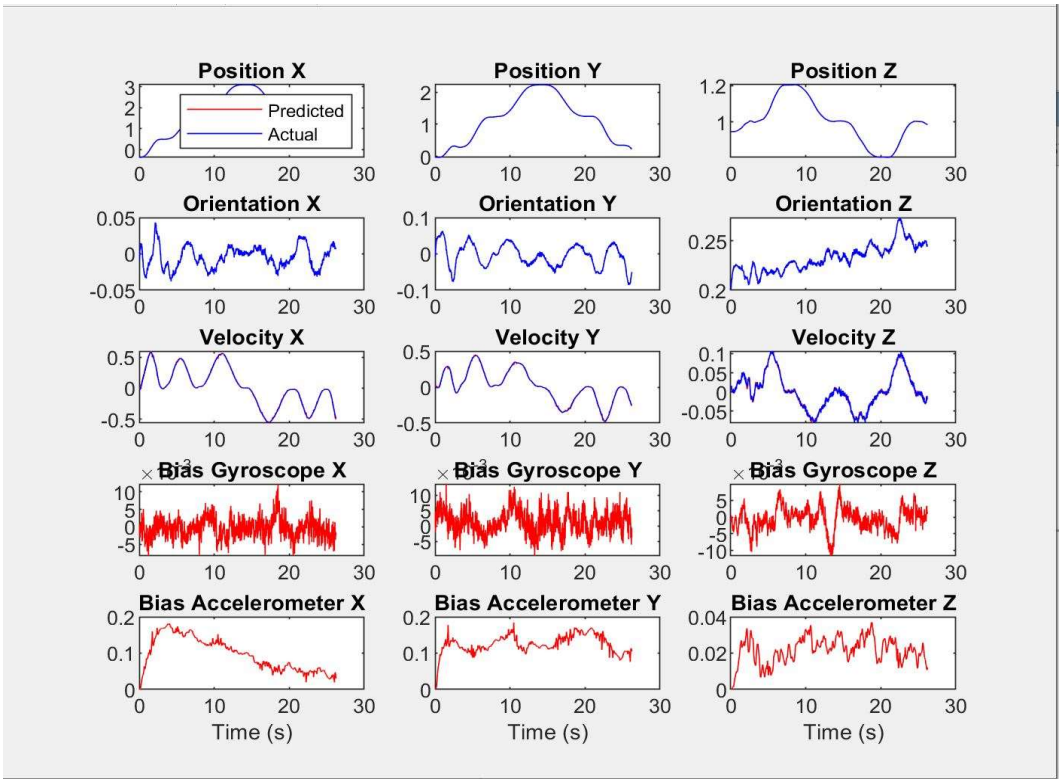

Figure 3: Kalman Filter Part1 Dataset 9

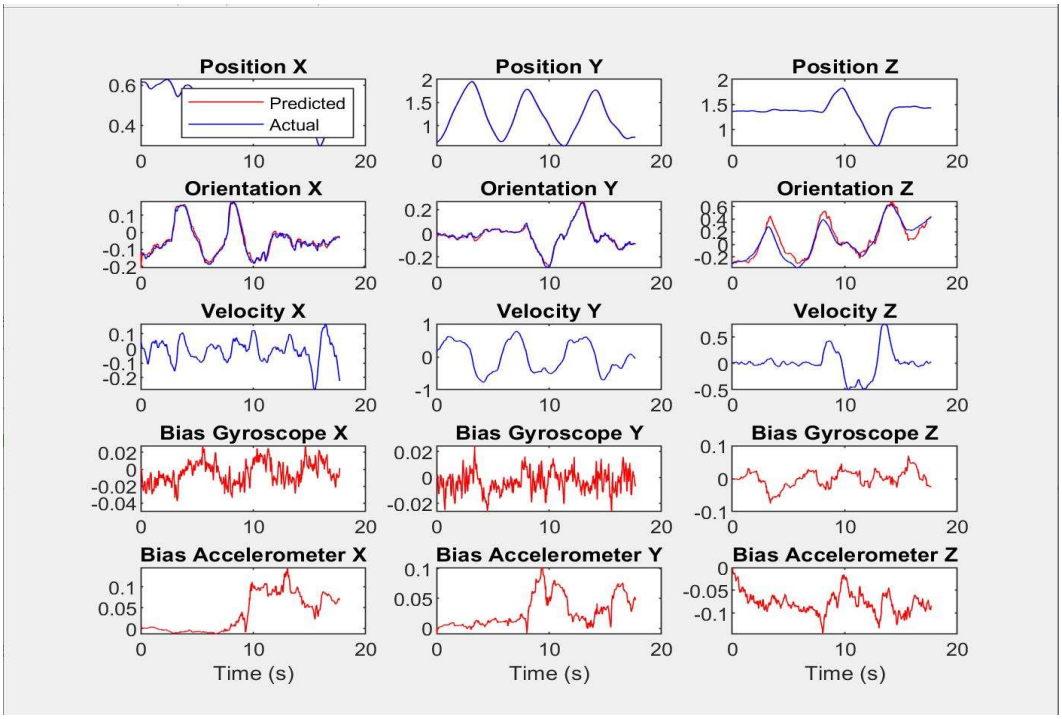

Figure 4: Kalman Filter Part 2 Dataset 1

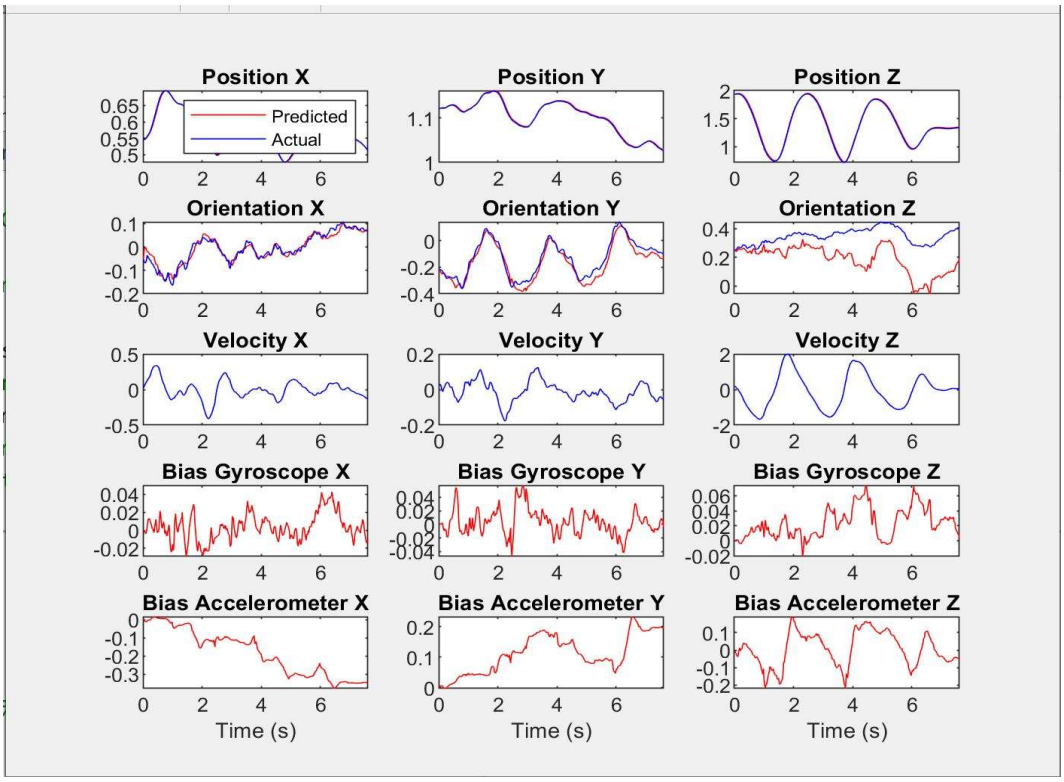

Figure 5: Kalman Filter Part 2 Dataset 4

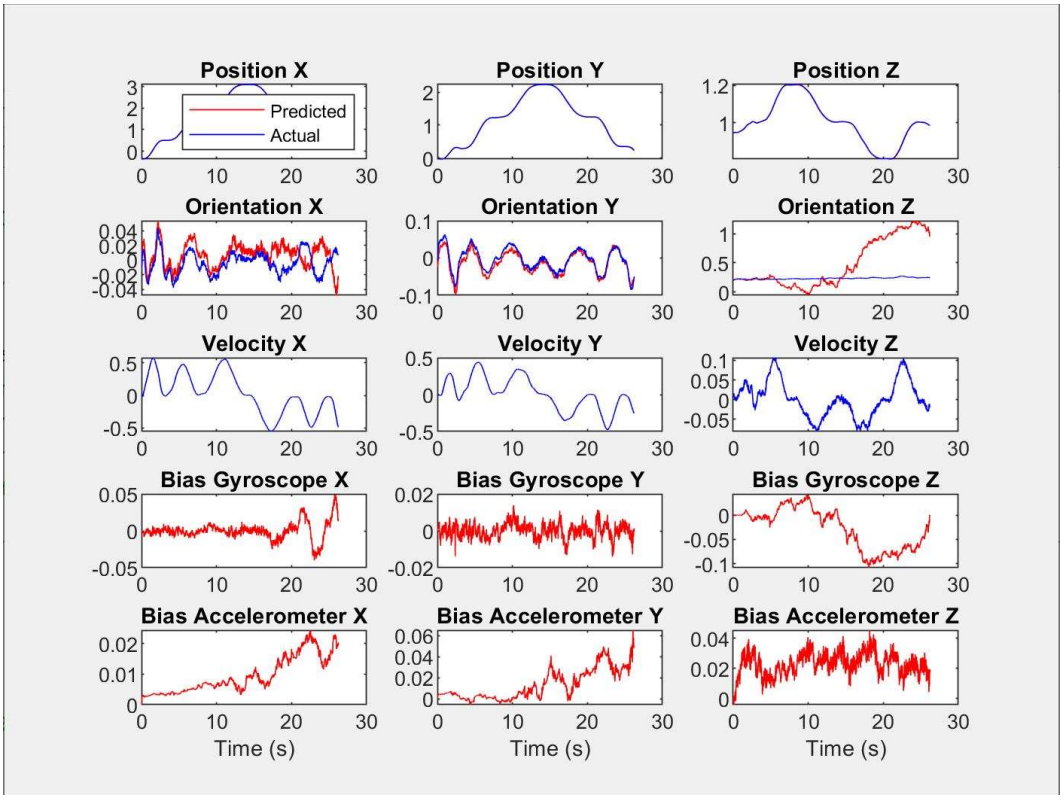

Figure 6:: Kalman Filter Part 2 Dataset 9

The plots of Kalman filter part1 were almost perfectly coinciding with actual plots the small differences in bias plots in part1 filter can be minimised by changing the value of R used in measurement model in update step. The plots of Kalman filter part2 were accurately coinciding with the actual plots except for orientation along Z-axis in dataset 4 and dataset 9. This is due to the fact that in Kalman filter part2 the measurement model only depends on velocity of the robot which results in slight inaccuracies along predicted orientations along X, Y, and Z-axis. The gyroscopic bias and accelerometer bias in both Kalman filter part 1 and Kalman filter part 2 can be made mode smooth by updating the values of R of measurement model present in Update step. The R value depicts how much we trust in the sensor readings given to the program. The larger the R value the lesser we depend on the sensor readings and vice versa. Overall, the predicted plots of the model along position, orientation, velocity, bias gyroscope, bias accelerometer were almost coinciding with the actual plots.

REFERENCES

1. Professor Giuseppe Loianno's Lecture notes
2. GRASP Lab report by Professor Vijay Kumar, university of Pennsylvania
3. Thrun S Burgard (2006), "Probabilistic Robotics", *Kybernetes*, Vol. 35 No. 7/8, pp
4. State Estimation for Robotics Chapter 3