0. I did this homework all by myself.
   Github link is :

2.

System State $= S = \begin{bmatrix} position\ of\ car\ at\ x\ direction \\ position\ of\ car\ at\ y\ direction \\ angle\ of\ car \\ motor\ angular\ velocity \\ bias \end{bmatrix} = \begin{bmatrix} x \\ y \\ theta \\ v \\ bias \end{bmatrix}$

Input of motor angular velocity $= in = \begin{bmatrix} velocity\ of\ left\ motor \\ velocity\ of\ right\ motor \end{bmatrix} = \begin{bmatrix} v0 \\ v1 \end{bmatrix}$

Thus system model is:

$$S = f(x, in) + N(0, Q)$$

Dynamic system is:

$$\begin{bmatrix} x \\ y \\ theta \\ v \\ bias \end{bmatrix} = \begin{bmatrix} x \\ y \\ theta \\ v \\ bias \end{bmatrix} + \begin{bmatrix} \cos(theta) * (v0 + v1) * radius\ of\ wheel/2 \\ \sin(theta) * (v0 + v1) * radius\ of\ wheel/2 \\ (v0 - v1) * radius\ of\ wheel/time \\ v \\ bias \end{bmatrix}$$

Measurement state:

$$M = \begin{bmatrix} distance\ to\ front\ line \\ distance\ to\ right\ line \\ angle\ of\ car \\ bias \end{bmatrix} = \begin{bmatrix} df \\ dr \\ angle \\ bias \end{bmatrix}$$

Measurement model is :

$$M = H(state) + noise\ part(W)$$

H function can be classified to 4 situations basing on geometric relationship.

1. Angle in (0,90)

$$H = \begin{bmatrix} (length\ of\ board - x) * \cos(angle) + (width\ of\ board - y) * \sin(angle) \\ (length\ of\ board - x) * \sin(angle) + y * \cos(angle) \\ angle\ of\ car \\ bias \end{bmatrix}$$

2. Angle in (90,180)

$$H = \begin{bmatrix} (width\ of\ board - y) * \sin(angle) + x * \cos(angle) \\ (length\ of\ board - x) * \sin(angle) + (width\ of\ board - y) * \cos(angle) \\ angle\ of\ car \\ bias \end{bmatrix}$$

3. Angle in (180,270)

$$H = \begin{bmatrix} -x * \cos(angle) - y * \sin(angle) \\ -(width\ of\ board - y) * \cos(angle) - x * \sin(angle) \\ angle\ of\ car \\ bias \end{bmatrix}$$

4. Angle in (270,360)

$$H = \begin{bmatrix} (length\ of\ board - x) * \cos(angle) - y * \sin(angle) \\ y * \cos(angle) - x * \sin(angle) \\ angle\ of\ car \\ bias \end{bmatrix}$$

Since Kalman filter only work on linear situation, we apply non-linear stochastic difference equations for system model. By using extended Kalman filter, nonlinear terms are assumed being converted to linear term by partial differential method.

Time update:

$$x_k = f(x_{k-1}, in)$$

$$state\_covariance_{ahead} = \frac{\partial f(x,in)}{\partial x} \times state\_covariance_{post} \times (\frac{\partial f(x,in)}{\partial x})^T + \frac{\partial f(x,in)}{\partial in} \times input\_covariance_{post} \times (\frac{\partial f(x,in)}{\partial in})^T$$

Measurement update:

$$Kalman\ gain = state\_covariance_{ahead} \times \left(\frac{\partial H(state)}{\partial state}\right)^T (\frac{\partial H(state)}{\partial state} \times state_{covariance_{ahead}} \times \left(\frac{\partial H(state)}{\partial state}\right)^T + R)^{-1}$$

$$x_k = x_{k-1} + Kalman\ gain \times (distance\ measuremtn\ by\ sensor - true\ motion\ distance)$$

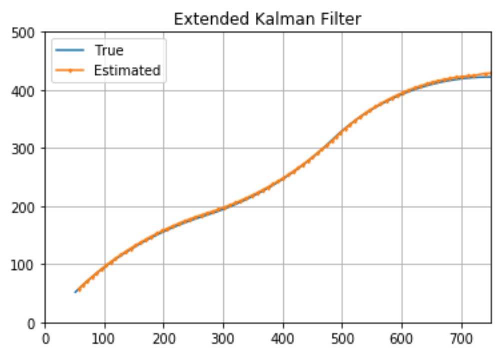
$$state\_covariance_{ahead} = (I - Kalman\ gain \times \frac{\partial H(state)}{\partial state})\ state\_covariance_{post}$$

Keep time update and measurement update loop and the EKF works.

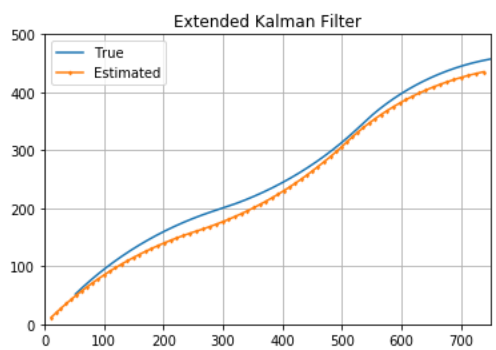
Results:

We choose (left speed, right speed) = (1,1.2) then (1.2, 1) then (1, 1.3) and set initial state at (50,50) with angle pi/4.

Know the initial state:



Not know the initial state:



The results show that EKF perform well with known initial state. There is some error when initial state is unknown. Error approximation can affect the result at some extent. Thus, it is better to do enough times of experiments to get exact standard deviation. More sophisticated model includes other parameters can also enhance the ability of car in real world.