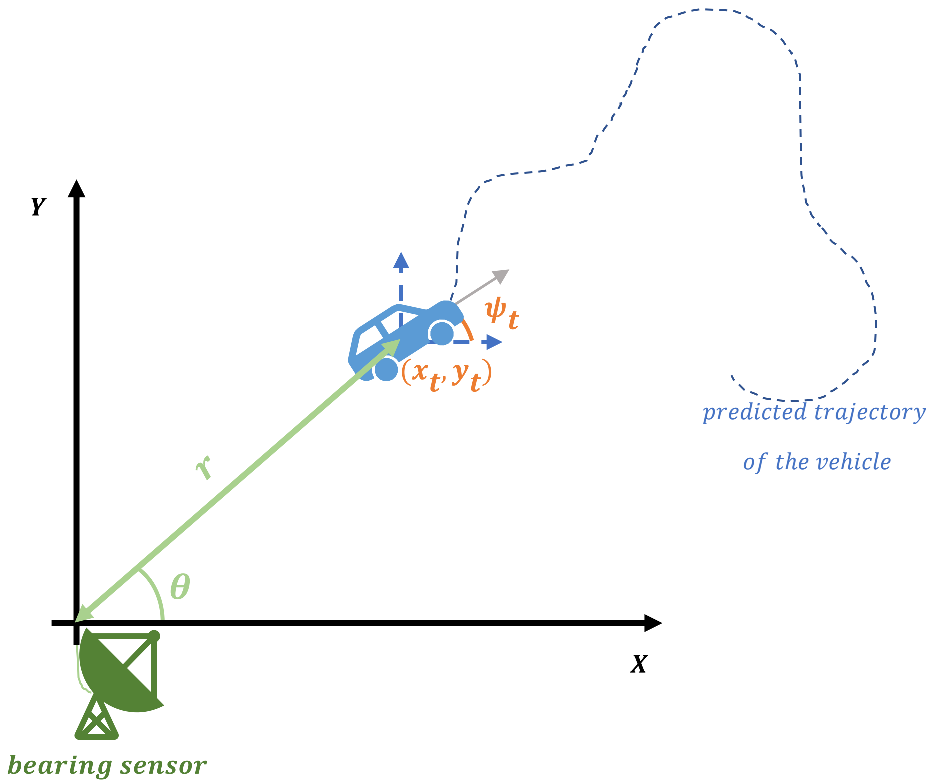
Optimal State Estimation

Practice 6. Extended Kalman filter

The Extended Kalman filter is a set of mathematical equations that provides an efficient computational estimate of the state of a process (e.g., the position and orientation of car) given a time-varying sequence of noisy measurements (e.g., distance, orientation of car etc) in nonlinear system. The filter is a popular mathematical estimator due to its efficiency and robustness. Our goal is to develop a simulation program with MATLAB to help develop the intuition and insight of novice users regarding the behavior of the Extended Kalman filter. We would have the ability to change various input parameters and then see how the Extended Kalman filter responds for a given set of noisy measurements.

[Practice 1: True state and measurement simulation for 2D vehicle]

The simulation targets are a dynamic state vector (*x, y,* ψ*(heading)*) of a vehicle on two-dimension space (*X and Y*), and a bearing sensor located at origin, which can measure the range r between sensor and the vehicle and the angle θ between the vehicle and *X* axis in world coordinate.



A nonlinear discrete vehicle motion model (dead reckoning, DR) with **velocity(V) and yawrate($\dot{\psi}$) inputs** can be represented as follow:

$$x_{t+1} = x_t + V \cdot dT \cdot cos(\psi_t)$$

$$y_{t+1} = y_t + V \cdot dT \cdot sin(\psi_t)$$

$$\psi_{t+1} = \psi_t + dT \cdot \dot{\psi}$$

$$(V = \sqrt{v_x^2 + v_y^2})$$

,where x_t is a X position of the vehicle at time t, y_t is a Y position of the vehicle at time t, ψ_t is the heading of the vehicle at time t, and dT is a sampling time. V is the speed of the vehicle and $\dot{\psi}$ is the yaw rate of the vehicle, which are given inputs of the vehicle model.

The discrete state space description of this system can be written as

$$\begin{bmatrix} x \\ y \\ \psi \end{bmatrix}_{t+1} = \begin{bmatrix} x_t + V \cdot dT \cdot cos(\psi_t) \\ y_t + V \cdot dT \cdot sin(\psi_t) \\ \psi_t + dT \cdot \dot{\psi} \end{bmatrix} = f(x, u, w, t)$$

The measurements are the range between the vehicle and the sensor, direction of the vehicle, therefore the measurement model can be described as follow:

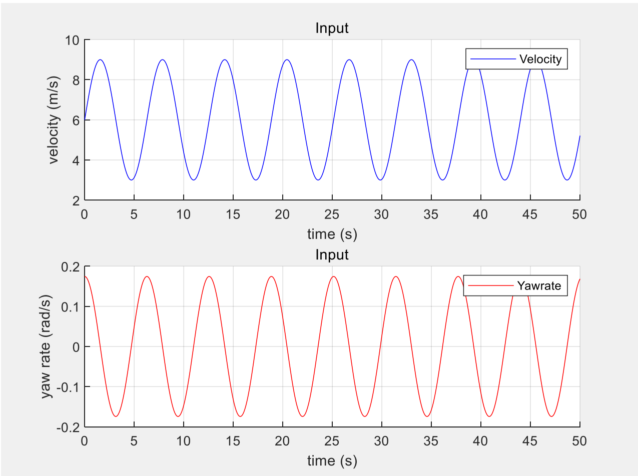
$$y_{t+1} = \begin{bmatrix} r \\ \theta \end{bmatrix}_{t+1} = \begin{bmatrix} \sqrt{r^2 + \theta^2} \\ tan^{-1}(y_t / x_t) \end{bmatrix} = \hbar(x, v, t)$$

We have the sampling time and initial information as below:

1) $\Delta t = 0.1$
2) initial position $x = 1m, y = 1m$
3) initial heading ψ = 45deg
4) initial velocity v_x= 3m/s, v_y =3m/s

[Practice 1-1]

Please simulate the x_t, y_t, ψ_t from time $t = 0$ to $t = 30$ using given velocity, yawrate data:



(input data)

Plot the simulation and submit the figure (each state and 2D vehicle position) for the simulation.

[Practice 1-2]

Please simulate y_t, which is a position sensor simulation data for the previous simulation. Each standard deviation of the sensor noise is **one meter** and **three degrees**. Plot the simulation results and save to m file. Submit the simulation figure and m. file.

(Hint: using gaussian random variable "normrnd" in Matlab)

[Practice 2: Kalman filter with Constant Velocity (CV) Model]

A constant velocity (CV) process model can be written as:

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{t+1} = \begin{bmatrix} 1 & 0 & dT & 0 \\ 0 & 1 & dT & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{t}$$

[Practice 2-1]

Estimate the state and the error between estimated state and true value using Extended Kalman filter for below cases:

- $x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, P_0 = \begin{bmatrix} 10000 & 0 & 0 & 0 \\ 0 & 10000 & 0 & 0 \\ 0 & 0 & 10000 & 0 \\ 0 & 0 & 0 & 10000 \end{bmatrix}.$

  Tuning the *Q*, *R* to get the best estimate! Submit the figures (x-axis is time and y-axes are $x_t, y_t, v_{x_t}, v_{y_t}$) for the estimated state and error.

[Practice 2-2]

Get the RMSE between estimated state and true value for each case.

[Practice 2-3]

Plot the P in three cases (x-axis is time and y-axes are P_{00}~P_{33}) for each case. Submit the figure for the P.

[Practice 3: Kalman filter with Dead Reckoning Model]

A state of Dead Reckoning model can be written as:

$$\begin{bmatrix} x \\ y \\ \psi \end{bmatrix}_{t+1} = \begin{bmatrix} x_t + V \cdot dT \cdot cos(\psi_t) \\ y_t + V \cdot dT \cdot sin(\psi_t) \\ \psi_t + dT \cdot \dot{\psi} \end{bmatrix}$$

[Practice 3-1]

Derive the process model for the Dead Reckoning model!

[Practice 3-2]

Plot the estimated state and the error between estimated state and true value (x-axis is time and y-axes are x_t, y_t, ψ_t) for each case. Submit the figures for the estimation for the best *Q, R*

[Practice 3-3] Get the RMSE between estimated state and true value for each case.

[Practice 3-4] Plot the P in three cases (x-axis is time and y-axes are P_{00}~P_{22}) for each case. Submit the figure for the P.

[Practice 3-5] Please describe the performance difference between CV and Dead Reckoning Kalman filter!