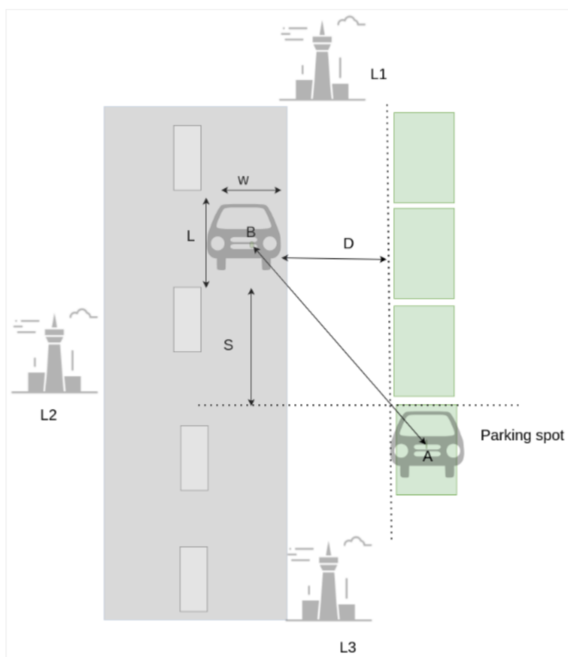
Assignment 3

Ivan Efremov

November 2022

1 Task description



So in our task we have to move robot from point B to A and localize it using sensors data and motion model.

Problem Description

- There are three landmarks: L1 (5, 30), L2 (5, -30), and L3 (-5, 0), which can be seen by the sensor attached to car. Sensor reading are obtained in the following way

$$\underbrace{\begin{bmatrix} r_t^i \\ \theta_t^i \end{bmatrix}}_{z_t^i} = \underbrace{\begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ atan2(m_{j,y} - y, m_{j,x} - x) - \theta \end{pmatrix}}_{h(x_{t,j,m})} + N(0, R) \qquad (1)$$

, where $m_{j,x}, m_{j,y}$ denotes the coordinates of jth landmark detection at time t. The white noise of each sensor reading , the optimal robot current location estimation, and the vehicle heading angle are given by $R = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_r^2 \end{bmatrix}$, $\mathbf{x}_{t,x}^- = x, \mathbf{x}_{t,y}^- = y$, and θ, respectively.

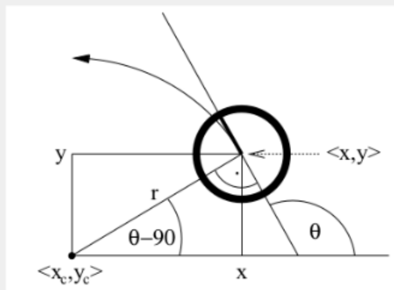
Problem Description

- The car has to incorporate sensor reading to improve its state estimation, i.e., use EKF localization, to navigate from position B (0, 0) to A. Location A has to be obtained using the given information, where lateral and longitudinal displacement is given by D and S, respectively
- You can make assumptions about this robot vehicle parameters, including width W and length L

2 Theory

2.1 Motion model

VELOCITY MOTION MODEL (NOISE-FREE)



- The center of the circle is at

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} x - \frac{v}{\omega} sin(\theta) \\ y + \frac{v}{\omega} cos(\theta) \end{bmatrix}$$

- After δt time, ideal robot will be at $\mathbf{x}_{t+1} = \begin{bmatrix} x_{t+1} & y_{t+1} & \theta_{t+1} \end{bmatrix}$

$$= \begin{bmatrix} x_c + \frac{v}{\omega} sin(\theta_t + \omega\delta t) \\ y_c - \frac{v}{\omega} cos(\theta_t + \omega\delta t) \\ \theta_t + \omega\delta t \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} -\frac{v}{\omega} sin(\theta_t) + \frac{v}{\omega} sin(\theta_t + \omega\delta t) \\ \frac{v}{\omega} cos(\theta_t) - \frac{v}{\omega} cos(\theta_t + \omega\delta t) \\ \omega\delta t \end{bmatrix}$$

Figure 1: Motion model in case of rotation with angular velocity.

```
# In case angular velocity equal to 0
f = sympy.Matrix([
    [x + dt*vt*sympy.cos(theta)],
    [y + dt*vt*sympy.sin(theta)],
    [theta]
])
```

Figure 2: Movement model in case of movement along a straight line.

I decided to use Dubins path planning to design my control. For this task I used an LSR Dubins path.

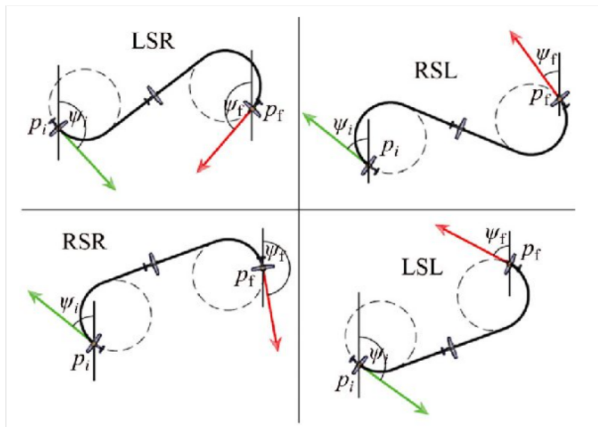
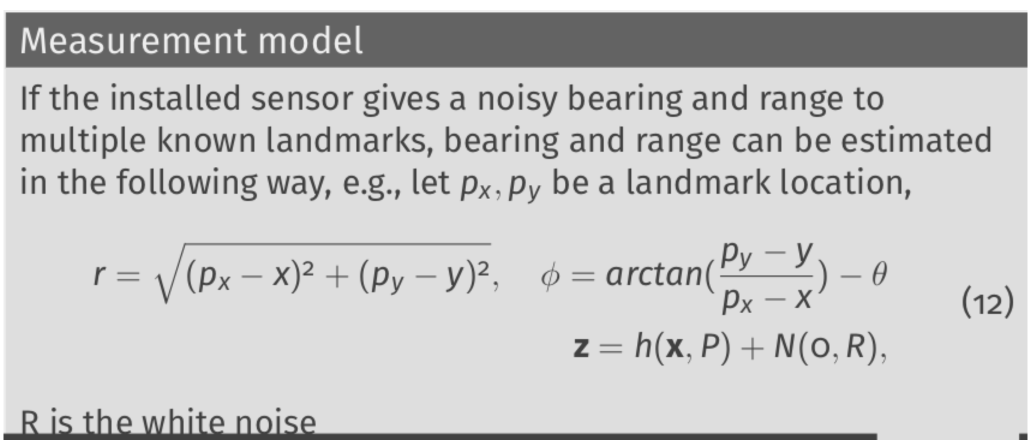


Figure 3: Examples of Dubins pathes.

2.2 Measurement model

The first part of the measurement model description was in the problem description.



Measurement model

If the installed sensor gives a noisy bearing and range to multiple known landmarks, bearing and range can be estimated in the following way, e.g., let p_x, p_y be a landmark location,

$$r = \sqrt{(p_x - x)^2 + (p_y - y)^2}, \quad \phi = arctan(\frac{p_y - y}{p_x - x}) - \theta \qquad (12)$$

$$\mathbf{z} = h(\mathbf{x}, P) + N(0, R),$$

R is the white noise

Figure 4: Measurement model description.

3 Results

The code for this assignment you can find in my GitHub repository.

Algorithm is implemented according to the theory. Results you can see in figure 5.

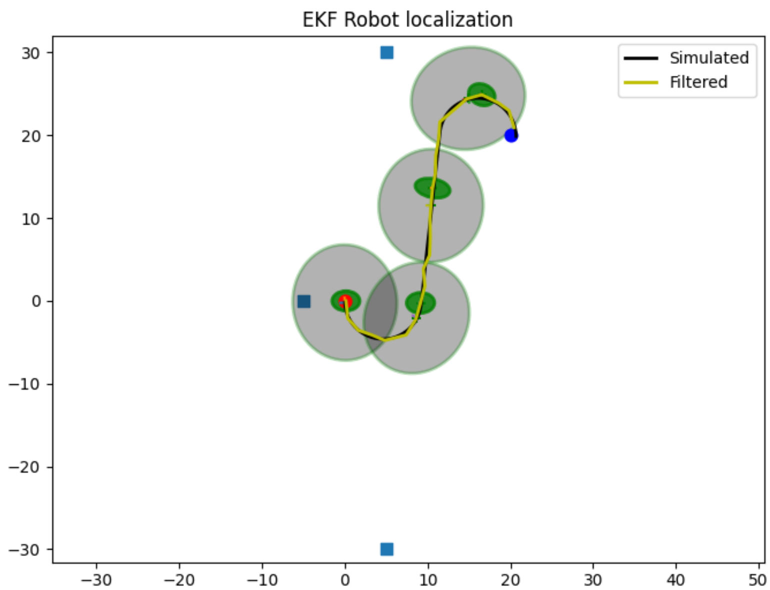


Figure 5: EKF robot localization

Green and gray ellipses are uncertainty estimation over the time. Blue squares are landmarks. The black line is the ideal simulated Dubins path and the yellow line is the EKF filtered trajectory that takes into account model prediction and sensor measurement data. As we can see EKF filter works quite good.