Report

May 5, 2021

1 EKF Assignment

1.1 Prithvi Poddar (17191)

In the following assignment I have used the previously built RRTstar class to generate the path from start to goal, nvigating betweent the obstacles.

1.2 Path following

The RRTstar algorithm givrs us waypoints along the path to be followed. Since the objective of this assignment is the application of EKF, I chode to stick with simple go-to-goal behaviour, taking the agent from one waypoint to another.
Thus, in the animation, the agent slows down as it reaches a waypoint and then speeds up again.

1.3 Simulating the environment

In the code, I have maintined 2 sets of observations,
1. The real coordinates of the agent 2. The agent's belief of its coordinates

In the animations, the black triangle represents the true coordinates of the agent and the red triangle represents the belief of the agent with the ellipse around the red triangle representing the uncertinity in the belief.

1.4 Extended Kalman Filter

Given the motion model of the agent:

$$x_t = g(u_t, x_{t-1}) + \epsilon_t$$

Where ϵ_t is a drawn from a Normal distribution with $mean = 0$ and $covariance = R_t$
And the observation model:

$$z_t = h(x_t) + \delta_t$$

Where δ_t is a drawn from a Normal distribution with $mean = 0$ and $covariance = Q_t$
The EKF algorithm can be written as follows:

EKF $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$:
1. $\overline{\mu_t} = g(u_t, \mu_{t-1})$
2. $\overline{\Sigma_t} = G_t \Sigma_{t-1} G_t^T + R_t$
3. $K_t = \overline{\Sigma_t} H_t^T (H_t \overline{\Sigma_t} H_t^T + Q_t)^{-1}$
4. $\mu_t = \overline{\mu_t} + K_t(z_t - h(\overline{\mu_t}))$
5. $\Sigma_t = (I - K_t H_t)\overline{\Sigma_t}$
6. Return μ_t, Σ_t

1.5 Motion model

For this assignment, I have considered a unicycle model with the following motion model:

$$
\begin{aligned}
x_t &= x_{t-1} + v\cos(\phi_{t-1})dt \\
y_t &= y_{t-1} + v\sin(\phi_{t-1})dt \\
\phi_t &= \phi_{t-1} + \omega dt
\end{aligned}
$$

And the control input being :

$$u_t = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

We can write the jacobian of the motion model as:

$$G_t = \begin{bmatrix} \frac{\partial x_t}{\partial x_{t-1}} & \frac{\partial x_t}{\partial y_{t-1}} & \frac{\partial x_t}{\partial \phi_{t-1}} \\ \frac{\partial y_t}{\partial x_{t-1}} & \frac{\partial y_t}{\partial y_{t-1}} & \frac{\partial y_t}{\partial \phi_{t-1}} \\ \frac{\partial \phi_t}{\partial x_{t-1}} & \frac{\partial \phi_t}{\partial y_{t-1}} & \frac{\partial \phi_t}{\partial \phi_{t-1}} \end{bmatrix}$$

$$G_t = \begin{bmatrix} 1 & 0 & -v\sin(\phi_{t-1})dt \\ 0 & 1 & v\cos(\phi_{t-1})dt \\ 0 & 0 & 1 \end{bmatrix}$$

1.6 Observation model

We assume a sensor that returns the coordinates of the obstacle with respect to the position of the agent.
Given that (x, y, ϕ) are the coordinates of the agent and (x_o, y_o) are the coordintes of the obstacle in the world coordinate, then we ca calculate the coordinates of the obstacle w.r.t the agent (x', y') with the following equations:

$$
\begin{aligned}
x' &= (x_o - x_t)\cos(\phi_t) + (y_o - y_t)\sin(\phi_t) \\
y' &= (y_o - y_t)\cos(\phi_t) - (x_o - x_t)\sin(\phi_t)
\end{aligned}
$$

We can write the jacobian of the observation as follows:

$$H_t = \begin{bmatrix} \frac{\partial x'}{\partial x_t} & \frac{\partial x'}{\partial y_t} & \frac{\partial x'}{\partial \phi_t} \\ \frac{\partial y'}{\partial x_t} & \frac{\partial y'}{\partial y_t} & \frac{\partial y'}{\partial \phi_t} \end{bmatrix}$$

$$H_t = \begin{bmatrix} -\cos(\phi_t) & -\sin(\phi_t) & (x_t - x_o)\sin(\phi_t) + (y_o - y_t)\cos(\phi_t) \\ \sin(\phi_t) & -\cos(\phi_t) & (x_t - x_o)\cos(\phi_t) - (y_t - y_o)\sin(\phi_t) \end{bmatrix}$$