Turtlebot EKF and Control

Alonzo Lopez

March 2021

1 System Description

1.1 State

$$X = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \tag{1}$$

1.2 Actions

We can send commands in the form of ROS Twist messages containing desired linear and angular velocity.

$$X = \begin{bmatrix} v & \omega \end{bmatrix}^T \tag{2}$$

where $v = \dot{x}$ and $\omega = \dot{\theta}$. Note that though we can send other velocities (e.g. v_y and v_z and angular rates about x and y), those velocity components are ignored and only the two components, v_x and angular velocity about z, or ω are extracted to command the forward velocity of the robot and the angular turning velocity, respectively. To simplify the notation going forward, we use v instead of v_x. The velocity commands are then split across the two wheels of the differential drive wheel base by the ROS Differential Drive Controller.

1.3 Dynamics

To describe the dynamic motion of the Turtlebot, we use the simple Unicycle model (also see LaValle's chapter on differential models) The continuous-time equations of motion are

$$\begin{aligned} \dot{x} &= v\ cos(\theta) \\ \dot{y} &= v\ sin(\theta) \\ \dot{\theta} &= \omega \end{aligned} \tag{3}$$

The discretized dynamics are

$$\begin{aligned} x_{k+1} &= x_k + v_k\ cos(\theta_k)\ dt \\ y_{k+1} &= y_k + v_k\ sin(\theta_k)\ dt \\ \theta_{k+1} &= \theta_k + \omega_k\ dt \end{aligned} \tag{4}$$

It's important to note that we assume our control action for $u = \begin{bmatrix} v & \omega \end{bmatrix}^T$ takes effect immediately as opposed to a more realistic model that incorporates acceleration.

$$X_{k+1} = f(X_k) + g(X_k, u_k) \tag{5}$$

1.4 Sensors

2 Experiment

2.1 Truth in Simulation