CSDS 419 Project Report

Noah Mollerstuen

Objective

The objective of this project is to explore the applications of probabilistic modelling to robotics. Specifically, I want to understand how probabilistic models can help a robot determine the state of its environment in the presence of conflicting measurements and uncertainties. I plan to develop a simulation of a two-dimensional mobile robot that will illustrate these principles.

Background

Many systems in robotics are dynamical, i.e. they change over time. For example, the position and velocity of the robot may change over time as the robot moves around and changes direction. The true state of the system (i.e. the true position of the robot) is often unknown, but various sensors may be used to gain information about the state of the system. Robots might use GPS, accelerometers, measure wheel rotations, ect. However, each of these measurements is uncertain. GPS may only be accurate to within 10 meters, and other measurements may drift over time. The problem then, is how to combine the various uncertain information available to the robot to optimally update its beliefs about the state of the system.

One common way to model how a system changes over time is a system of linear differential equations. A system of differential For example, for a robot moving in a straight line, with position x and velocity v:

$$x' = v$$
$$v' = 0$$

Kalman Filters

In my robotics work, I’ve often had a need to fuse together different types of sensor data to produce a single estimate of the robot’s state or the state of its environment. The proposed solution is almost invariably some kind of Kalman filter, but I’ve never really understood how they work. My first steps towards understanding have been following some online tutorials and reading the Barber chapter on continuous-state Markov models. In most of the tutorials, Kalman filters have been presented in a discrete context, and they seem to be somewhat simpler than Barber’s model, so I will likely first attempt to implement a Kalman filter in a discrete setting, but I am interested in returning to Barber’s model when I have a better fundamental understanding.

The basic principle behind a Kalman filter is twofold: first, apply state space equations to estimate the current state based on the previous state, and second, refine this belief based on evidence from sensors. These two stages are repeated each time a new sensor reading becomes available. The probabilities of the states are modelled by multivariate gaussians, because both operations are relatively simple to compute on a gaussian.

Resources

- Barber Ch. 24
- https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/

Todo

- Try implementing a a Kalman filter with pgmpy (or numpy alone)
- Investigate applying similar models to nonlinear systems (extended Kalman filters)