Terms Clarification

The following three terms are equivalent:

1. State Estimator
2. Perception
3. Sensor Function

State Estimation

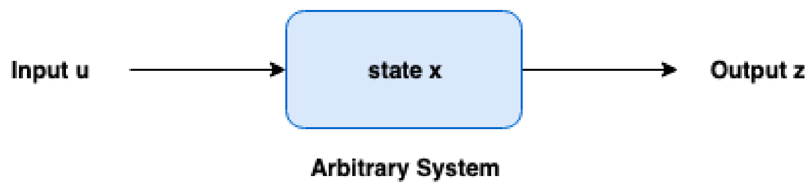

Figure 1: Arbitrary System Block Diagram

State update dynamics

$$x' = f(x, u, t)$$

Sensor update

$$z' = h(x, u, t)$$

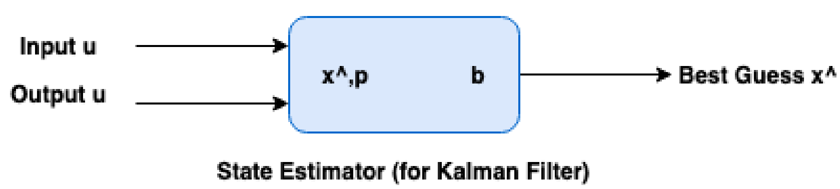

Figure 2: State Estimator Block Diagram

Figure out state using sensor inputs and know inputs to state

What is the state of the state estimator?

$$Input\ b' = f(b, u, z, t)$$
$$Output\ \hat{x} = h(b, u, z, t)$$

This output is our "best guess state estimation"

Best Guess at x ($\hat{x}$): defined in terms of probability distribution of x, three approaches can be taken as shown in Figure 3.

1. $\hat{x}_{max\ likelihood}$ $(mode)$
2. $\hat{x}_{min-max}$ $(median)$
3. $\hat{x}_{min\ variance}$ $(mean)$

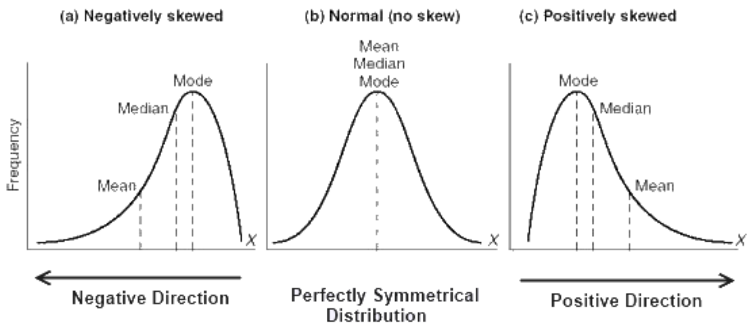

Figure 3: 3 Approaches to find Best Guess

State of State Estimator

$$Belief(x') = P(x = x')$$

This gives a PDF of x

P is the probability that our best guess for x (x') is equal to the actual value of x, given over all possible values of x'

Example: 1-D point

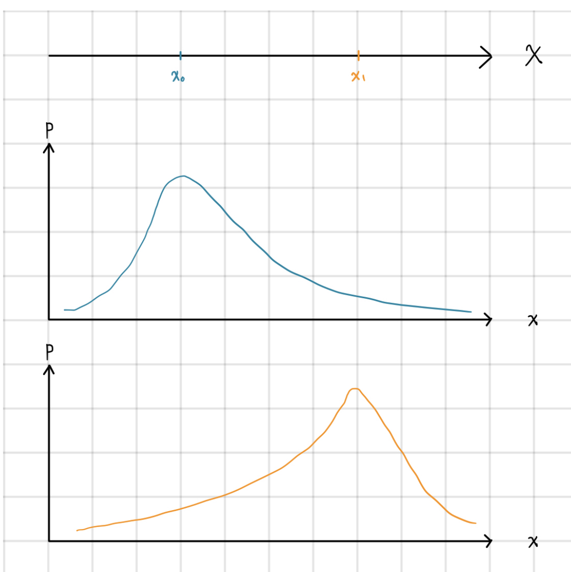

Figure 4: Belief State Example

In order to apply a Kalman filter, the system must:

1. Be linear.
2. Have Gaussian noise and Gaussian uncertainty.

Therefore, all possible beliefs will also follow a Gaussian distribution.

To fully define a Gaussian distribution, we only need to numbers:

1. Mean.
2. Variance.

Discrete State

Example:

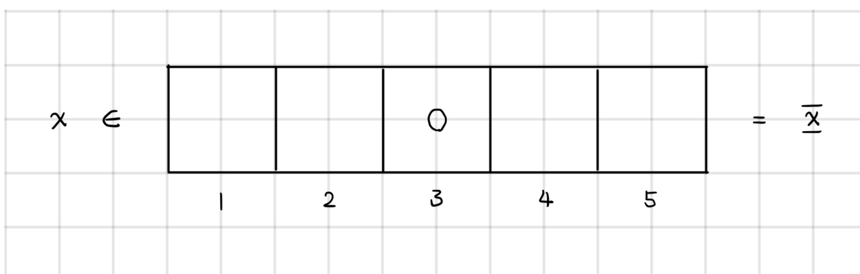

Figure 5: Discrete State Example

$$x[t+1] = x[t] + u[t] + d[t]$$
$$noise : u \in -1, 0, 1$$

To quantify the noise, we need to give a PDF of the noise, specified over u.

An example PDF is:

$$d = \begin{cases} -1, & \text{up to } 10\% \\ 0, & \text{up to } 80\% \\ 1, & \text{up to } 10\% \end{cases}$$

This example PDF implies that the noise will lead to a reading that is 1 to the left 10% of the time, a reading of the particle being unchanged 80% of the time, and a reading that is 1 to the right 10% of the time.

How to describe the belief state:

$$Bel(x) = [p(x=1), p(x=2), p(x=3), p(x=4), p(x=5)]$$

The belief state is described by the probabilities of each state.

The particle starts in the middle,

$$Bel_0(x) = (0, 0, 1, 0, 0), u(0) = 0$$

$$Bel_1(x) = (0, 0.1, 0.8, 0.1, 0), u(1) = 0$$

$$Bel_2(x) = (0.01, 0.16, 0.66, 0.16, 0.01)$$

Note: in $Bel_2(x)$, to find values $(x_1, x_2, x_3, x_4, x_5)$:

$$x_1 = P(1) * 0.8 + P(2) * 0.1 = 0 * 0.8 + 0.1 * 0.1 = 0.01$$

$$x_2 = P(1) * 0.1 + P(2) * 0.8 + P(3) * 0.1 = 0 * 0.1 + 0.1 * 0.8 + 0.8 * 0.1 = 0.16$$

$$x_3 = P(2) * 0.1 + P(3) * 0.8 + P(4) * 0.1 = 0.1 * 0.1 + 0.8 * 0.8 + 0.1 * 0.1 = 0.66$$

$$x_4 = P(3) * 0.1 + P(4) * 0.8 + P(5) * 0.1 = 0.8 * 0.1 + 0.1 * 0.8 + 0 * 0.1 = 0.16$$

$$x_5 = P(4) * 0.1 + P(5) * 0.8 = 0.1 * 0.1 + 0 * 0.8 = 0.01$$

Bayesian Filter

Belief state update with input u(t).

$$Bel_{t+1}(x) = f(Bel_t(x), u(t))$$

$$Bel_{t+1}(x) = \sum_{x' \in \underline{\overline{x}}} Bel_t(x') \cdot P(x_{t+1} = x | x_t = x', u)$$

$$z[t] = x[t] + n[t]$$

Note: The noise, n[t], does not change where we are, just what information we know about where we are.

$$Bel_1^+(x) = P(x|z) = \frac{P(z|x)P(x)}{P(z)}$$

$$Where\ P(x) = Bel_1^-(x)$$

How to apply Bayesian filter for finite states to infinite number of states?
Make a lot of samples, and run dynamics update on each sample, as shown in Figure 6.

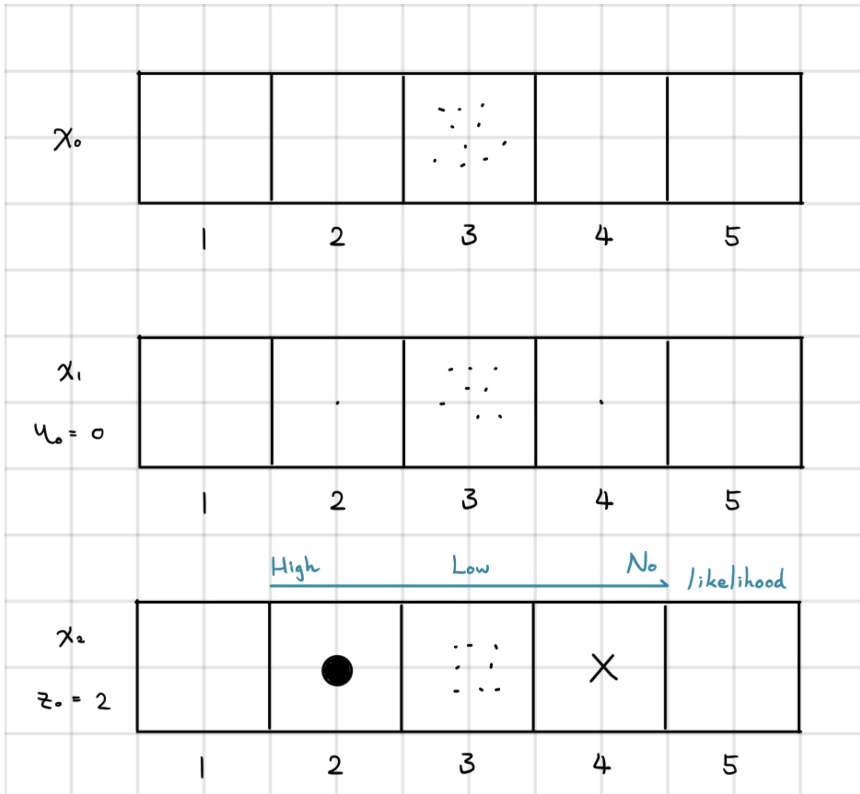

Figure 6: Bayesian Filter Example

Particle Filter

Steps for implementing a basic, minimum, particle filter.

1. Spawn N particles according to the knowledge of initial state and with equal probability.
2. Run dynamics update to each particles' state.
3. Update the measurement to particle probability.
4. Loop between steps 2 and 3 each time.

Note:

1. The Bayesian filter requires state storage and operations, which grows with size.
2. In a particle filter, N dictates the run time.
3. Problems with particle filters:
   (a) Loss of particles.
   (b) Run the risk of too little particles left.
   (c) Particle loss means loss of information.

References

1. Mehta, Ankur. "Design of Robotic Systems I." EE183DA. University of California Los Angeles, Los Angeles, California. 24 Jan. 2019. Lecture.
2. Asada, and Harry. "Lecture Notes." MIT Open CourseWare. Massachusetts Institute of Technology, n.d. Web. 31 Jan. 2019.