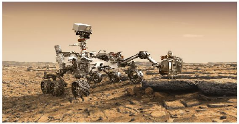
Introduction to Robotics

Week-1

Prof. Balaraman Ravindran

IIT-Madras

- Robotic systems perceive their environments through sensors(ultrasonic, infra-red, visual feed, etc.) and have some actuators (Wheels, mechanical arms, etc.) that they can use to manipulate their environment.



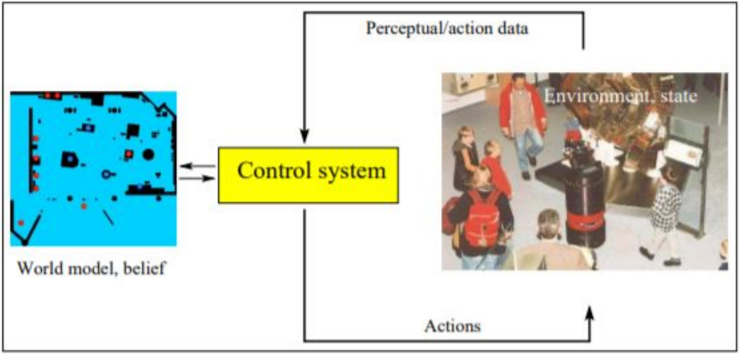
- Modern-day robotic systems operate in highly dynamic and unpredictable environments, while additional uncertainty might arise from unreliable sensor input, unreliable actuators, inaccuracies in internal computation, etc. A successful robotic system must robustly deal with this uncertainty.

- This course provides an in-depth introduction to probabilistic algorithms for robotics, representing uncertainty explicitly, and basing control decisions on probabilistic information.

Roadmap:

- Recursive State estimation
- Gaussian Filters
- Non-Parametric Filters
- Motion Models
- Mapping
- Localization
- Path Planning

- A core idea in probabilistic robotics is estimating state from sensor data. Typically, state variables are not directly measurable, and a robot has to rely on it's sensors to gather information, through interaction with the environment.



- **State ($x_0,\ x_1,\ x_2, ...,\ x_t$):**
  - Robot Pose / location in world
  - Configuration of Actuators
  - Surrounding object locations
  - location and velocity of other moving objects
  - Internal mechanics (Battery life, etc.)
  - etc.

  Complete states: We consider a state representation 'complete' if the future is independent of the history $\{(x_0, ..., x_{t-1}), (u_1, ..., u_{t-1}), (z_1, ..., z_{t-1})\}$ given the present state x_t. (Markov property)

- **Interaction**

  Measurement Data $(z_1, z_2, ..., z_t)$

  - Camera image
  - Ultrasonic sensor output
  - etc.

  Control Actions $(u_1, u_2, ..., u_t)$

  - Robot motion
  - Manipulation of objects
  - etc.

We Assume $x_0 + u_1 \rightarrow z_1$

Notation:

$x_{0:t} : x_0, x_1, ..., x_t$

$u_{1:t} : u_1, u_2, ..., u_t$

$z_{1:t} : z_1, z_2, ..., z_t$

The dynamical stochastic system of the robot and its environment are described by *state transition probabilities* and the *measurement probabilities*.

- **State Transition probability:**

  $p(x_t|x_{0:t-1},\ u_{1:t},\ z_{1:t-1})$

  Assuming the Markov property,

  $= p(x_t|x_{t-1},\ u_t)$
- **Measurement probability:**

  $p(z_t|x_{0:t},\ u_{1:t},\ z_{1:t-1})$

  Assuming the Markov property,

  $= p(z_t|x_t)$

Belief: As discussed, typically, state cannot be measured directly. A *Belief* reflects the robot's internal knowledge about the state of the environment.

Belief Distributions: A belief distribution assigns a probability to each possible state variable, conditioned on the available data.

$bel(x_t) = p(x_t | z_{1:t},\ u_{1:t})$

Occasionally, it will prove useful to calculate a posterior before incorporating z_t, just after executing the control u_t

$\overline{bel}(x_t) = p(x_t | z_{1:t-1},\ u_{1:t})$

We call this value the *'prediction'* as it does not incorporate the current measurement z_t.

- Calculating $bel(x_t)$ from $\overline{bel}(x_t)$ is called *correction* or the *measurement update*.

The Bayes Filter Algorithm

- This algorithm calculates the belief distribution from measurement and control data, recursively calculating $bel(x_t)$ from $bel(x_{t-1})$.
- There are two steps to this algorithm: *prediction* and measurement update.

Prediction:

$\overline{bel}(x_t) = p(x_t|z_{1:t-1},\ u_{1:t})$

Using $p(x) = \int p(x|y)p(y)dy$

$= \int p(x_t|x_{t-1}, z_{1:t-1}, u_{1:t})p(x_{t-1}|z_{1:t-1},\ u_{1:t-1})dx_{t-1}$

Assuming Markov Property,

$= \int p(x_t|x_{t-1},\ u_t)p(x_{t-1}|z_{1:t-1},\ u_{1:t-1})dx_{t-1}$

$= \int \boldsymbol{p(x_t|x_{t-1},\ u_t)bel(x_{t-1})dx_{t-1}}$

Measurement Update:

$bel(x_t) = p(x_t|z_{1:t},\ u_{1:t})$

Using Bayes rule $\{p(A|B) = \frac{p(B|A)p(A)}{p(B)}\}$,

$= \frac{p(z_t|x_t, z_{1:t-1}, u_{1:t})p(x_t|z_{1:t-1},\ u_{1:t})}{p(z_t|z_{1:t-1},\ u_{1:t})}$

$= \eta p(z_t|x_t,\ z_{1:t-1},\ u_{1:t})p(x_t|z_{1:t-1},\ u_{1:t})$

Assuming Markov Property,

$= \eta p(z_t|x_t)p(x_t|z_{1:t-1},\ u_{1:t}) = \boldsymbol{\eta p(z_t|x_t)\overline{bel}(x_t)}$

The final algorithm:

Algorithm Bayes_filter($bel(x_{t-1}), u_t, z_t$)**:**

  for all x_t do

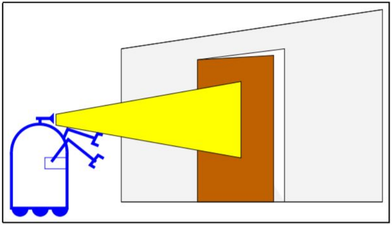
    $\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})\, dx$

    $bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$

  endfor

  return $bel(x_t)$

The Door World



- A door can be in two possible states **open** or **closed**.
- A robot estimates the state of a door using it's camera
- Only the robot can change the state of the door, using the action **push** to open a closed door.

An Illustration of the Bayes Filter Algorithm

We assume that the robot does not know the state of the door initially,

$$bel(X_0 = open) = 0.5$$
$$bel(X_0 = closed) = 0.5$$

Further assume that the robot's sensors are noisy,

$$p(Z_t = \text{sense-open} | X_t = open) = 0.6$$
$$p(Z_t = \text{sense-closed} | X_t = open) = 0.4$$

$$p(Z_t = \text{sense-open} | X_t = closed) = 0.2$$
$$p(Z_t = \text{sense-closed} | X_t = closed) = 0.8$$

Finally, assume that if an agent tries to open a door, there is a possibility of failure,

$$p(X_t = open | U_t = push,\ X_{t-1} = open) = 1$$
$$p(X_t = closed | U_t = push,\ X_{t-1} = open) = 0$$
$$p(X_t = open | U_t = push,\ X_{t-1} = closed) = 0.8$$
$$p(X_t = closed | U_t = push,\ X_{t-1} = closed) = 0,2$$

The agent can also choose to **do-nothing** in which case, the world does not change.

$$p(X_t = open | U_t = \text{do-nothing},\ X_{t-1} = open) = 1$$
$$p(X_t = closed | U_t = \text{do-nothing},\ X_{t-1} = open) = 0$$
$$p(X_t = open | U_t = \text{do-nothing},\ X_{t-1} = closed) = 0$$
$$p(X_t = closed | U_t = \text{do-nothing},\ X_{t-1} = closed) = 1$$

Bayes-filter$(bel(X_0), U_t = \text{do-nothing}, Z_t = \text{sense-open}))$

Prediction step: $\overline{bel}(x_1) = \sum_{x_0} p(x_1|u_1, x_0)bel(x_0)$

So,

$\overline{bel}(X_1 = open) =$

$p(X_1 = open|U_t = \text{do-nothing},\ X_0 = open)bel(X_0 = open)$

$+p(X_1 = open|U_t = \text{do-nothing},\ X_0 = closed)bel(X_0 = closed)$

$= (1 * 0.5 + 0 * 0.5) = \mathbf{0.5}$

Similarly,

$\overline{bel}(X_1 = closed) =$

$p(X_1 = closed|U_t = \text{do-nothing},\ X_0 = open)bel(X_0 = open)$

$+p(X_1 = closed|U_t = \text{do-nothing},\ X_0 = closed)bel(X_0 = closed)$

$= (0 * 0.5 + 1 * 0.5) = \mathbf{0.5}$

Measurement Update:

$bel(x_1) = \eta p(Z_1 = \text{sense-open}|x_1)\overline{bel}(x_1)$

So,

$bel(X_1 = open) =$
$\eta p(Z_1 = \text{sense-open}|X_1 = open)\overline{bel}(X_1 = open)$

$= \eta\, 0.6 * 0.5 = \eta\, 0.3$

Similarly,

$bel(X_1 = closed) =$
$\eta p(Z_1 = \text{sense-open}|X_1 = closed)\overline{bel}(X_1 = closed)$

$= \eta\, 0.2 * 0.5 = \eta\, 0.1$

The normalization constant can be calculated as:

$\eta = (0.3 + 0.1)^{-1} = 0.25$

Finally, we have:

$\boldsymbol{bel(X_1 = open) = 0.75}$

$\boldsymbol{bel(X_1 = closed) = 0.25}$

These values can be iterated for the next time step:

$\underline{\textbf{Bayes-filter}(bel(X_1), U_t = \text{push}, Z_t = \text{sense-open}))}$

Prediction step:

$\overline{bel}(X_2 = open) =$

$p(X_2 = open | U_t = \text{push},\ X_1 = open) bel(X_1 = open)$

$+ p(X_2 = open | U_t = \text{push},\ X_1 = closed) bel(X_1 = closed)$

$= (1 * 0.75 + 0.8 * 0.25) = \mathbf{0.95}$

Similarly,

$\overline{bel}(X_2 = closed) =$

$p(X_2 = closed | U_t = \text{push},\ X_1 = open) bel(X_1 = open)$

$+ p(X_2 = closed | U_t = \text{push},\ X_1 = closed) bel(X_1 = closed)$

$= (0 * 0.75 + 0.2 * 0.25) = \mathbf{0.05}$

Measurement Update:

$bel(X_2 = open) =$
$\eta p(Z_2 = \text{sense-open} | X_2 = open)\overline{bel}(X_2 = open)$

$= \eta\ 0.6 * 0.95 \approx \mathbf{0.98}$

Similarly,

$bel(X_2 = closed) =$
$\eta p(Z_2 = \text{sense-open} | X_2 = closed)\overline{bel}(X_2 = closed)$

$= \eta\ 0.2 * 0.05 \approx \mathbf{0.017}$