Lecture 05
Virtual Sensors & Localization Principles
Instructor: Jingjin Yu
Outline

Sensor mapping

Perfect virtual (abstract) sensors
⇒ Opposite extremes: the dummy sensor and the hyper sensor
⇒ Not as extreme sensors
⇒ Virtual depth sensors
⇒ Comparing the strength of sensors

Some important distance measurement and localization principles
⇒ Triangulation
⇒ GPS and trilateration
Sensor Modeling

A sensor can be modeled using a mapping $h$:

$$h: X \rightarrow Y$$

$X$: state space

$Y$: observation space

$h$: sensor mapping

We will use this to characterize various sensors.

Here, we will not discuss too much on individual sensor performance, e.g., accuracy.

But remember generally sensors have uncertainties.
Mapping for Common Sensors (I)

Absolute wheel encoder

\( X \): wheel location, \( S^1 \)

\( Y \): \( \{0,1\}^4 \)

\( h : S^1 \rightarrow \{0,1\}^4 \)

Infinite resolution: \( h : S^1 \rightarrow S^1 \)

Gyro

\( X : \mathbb{R}^3 \times SO(3) \)

\( Y : SO(3) \)

This is all possible robot position and orientation

This is all possible sensor output values

Similar for compass

\( h : \mathbb{R}^2 \times S^1 \rightarrow S^1 \)
Mapping for Common Sensors (II)

The mapping can be complex when history is involved

Relative wheel encoder

\[ h \text{ maps } \Delta x \times \Delta t \text{ to } \omega \text{ (rotational speed, } \omega = \frac{\Delta x}{r \Delta t} \) \]

\[ \Rightarrow \text{Simple thing to do: choose the right state} \]

\[ h: X_\omega \rightarrow Y_\omega \]

\[ X_\omega : \text{rotational speed (} \mathbb{R} \) \]

\[ Y_\omega : \text{measured rotational speed (} \mathbb{R} \) \]

\[ \Rightarrow \text{But this makes it trivial} \]

Same is true for certain other sensors, e.g. accelerometers

Then why use sensing mapping for modeling?

\[ \Rightarrow \text{For studying computation related issues} \]

\[ \Rightarrow \text{Virtual sensors} \]
Perfect Virtual Sensors

For a sensor mapping $h: P \rightarrow Y$, let’s fix $P$ for the discussion.

A perfect virtual sensor is one that slices $P$ into pieces, i.e.,

Different mapping $h$ does the slicing differently.

We will look at the preimage of $h$ for a given $y \in Y$

$$h^{-1}(y) = \{ p \in P \mid h(p) = y \}$$

Basically, a perfect sensor does not producing overlapping preimages.

⇒ This makes the conceptual study easier to do.

⇒ Most real sensors do not behave exactly like this.

Images: Mobile Robotics by LaValle
Opposite Extremes

The “dummy” sensor: the sensor that does nothing

\[ h: P \to \{0\} \]
\[ \forall p \in P, h(p) = 0 \]

In practice, this could be a broken sensor

The “hyper” sensor: sensor that is the most accurate

\[ Y = P \]
\[ h: P \to Y, p \mapsto p \]

What are the preimages?

\[ \text{Dummy sensor, } h^{-1}(y) = P \]
\[ \text{Hyper sensor, } h^{-1}(y) = y \]

Both sensors are absurd

\[ \text{But useful as extreme cases} \]
Not As Extreme Examples

Projection sensor

\[ P = X_1 \times X_2 \times \cdots \times X_k \]

For \( p = (x_1, \ldots, x_k) \in X \), \( h(x_1, \ldots, x_k) = x_1 \)

What is \( Y \)?

\[ Y = X_1 \]

That is, \( h \) defines a hyper sensor in a subspace \( X_1 \)

Given \( y \in Y \), what is \( h^{-1}(y) \)?

Have we seen projection sensors before?

Yes, gyro

And compass: \( h(x, y, \theta) = \theta \)

What is \( h^{-1}(\theta) \)?
Virtual Depth Sensors (I)

For mobile robots to navigate among obstacles

\[ X = \mathbb{R}^2 \times S^1 \times \mathcal{F} \]

\[ \mathcal{F} \]: the space of free spaces

\[ F \in \mathcal{F} \] is the current free space

\[ \Rightarrow \text{Suppose the robot has a sonar mounted in the front} \]

\[ \Rightarrow \text{The robot can measure the distance to the obstacle} \]

\[ \Rightarrow \text{What is the sensor mapping?} \]

\[ Y = \mathbb{R}, \text{ a distance} \]

\[ h(x, y, \theta, F) = \|(x, y) - b(p)\| \]

\[ \Rightarrow \text{This is a simple direction depth sensor} \]

\[ \Rightarrow \text{What if the sensor is at a different angle } \phi \? \]

\[ h_{\phi}(x, y, \theta, F) = \|(x, y) - b(p, \phi)\| \]

\[ \Rightarrow \text{For } k \text{ directions } \phi_1, \ldots, \phi_k \]

\[ h(x, y, \theta, F) = (h_{\phi_1}, \ldots, h_{\phi_k}) \]
Virtual Depth Sensors (II)

Omnidirectional depth sensors

\[ \text{For } k \text{ directions } \phi_1, \ldots, \phi_k \]
\[ h(x, y, \theta, F) = (h_{\phi_1}, \ldots, h_{\phi_k}) \]

Do you recall an example?

Sick lasers: \( k = 1080 \)

When \( k \) is big, this approaches an omnidirectional depth sensor

It may have cutoffs

Boundary distance sensor

What if we want to measure the distance to the nearest wall?

\[ h_{bd}(p) = \min_{0 \leq \phi < 2\pi} h_{\phi} \]

Proximity sensor

Proximity sensor is a special boundary sensor

\[ h_{\varepsilon}(p) = \begin{cases} 1, & h_{bd}(p) \leq \varepsilon \\ 0, & \text{otherwise} \end{cases} \]
Comparing Sensor Strengths

Why study perfect virtual sensors?

⇒ To compare the strength of different sensors
⇒ Dummy versus hyper – which is more powerful?
  ⇒ It depends! If I don’t use the sensor, it doesn’t matter
  ⇒ If I need to use it, then dummy is useless
⇒ $h_{bd}$ versus $k$-directional?
  ⇒ Both are not as strong as $h_{omni}$
  ⇒ Not comparable with each other

Sensor dominance

⇒ A perfect virtual sensor $h_1 : X \rightarrow Y_1$ is “better” than $h_2 : X \rightarrow Y_2$ if

$$\forall y_1 \in Y_1, \exists y_2 \in Y_2 \text{ s.t. } h_1^{-1}(y_1) \subset h_2^{-1}(y_2)$$

⇒ That is, $h_1$ can “simulate” $h_2$
⇒ But, this is not always beneficial!!!
Measuring Distance with Triangulation

Triangulation is an ancient technique

⇒ Known for at least 1700 years (Pei Xiu)

Very simply principle

![Diagram of triangulation principle](image)

But very useful!

⇒ Sick laser

⇒ Kinect (RGBD)

Images: Wikipedia and Mobile Robotics by LaValle
The Global Positioning System (GPS) is a space-based navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.

GPS is a U.S. owned system. Many other countries have their own:

- GLONASS (Russia, about the same time)
- Galileo (EU), BeiDou (China)
- IRNSS (Indian, regional), Quasi-Zenith (Japan, regional)
A Brief History of GPS

1960s – conceptualization in U.S. military
1978 – the launch of the first GPS satellite
1989 – The introduction of the first hand-held GPS receiver
1992 – Used in Operation Desert Storm
1996 – President Clinton decided to make GPS free for civilian use

Great! There are many things I would not be able to do without GPS!
Driving to NBHS, track my runs, outdoor exploration, ...
Segments of GPS

GPS has three segments

- **Space segment (satellites):**
  - 24 GPS satellites needed, in six obits, 4 each
  - About 20,200 kilometers altitude
  - Ensures at least 4 satellites are visible anywhere on earth
  - Currently 31 GPS satellites for redundancy

- **Control segment (stations):**
  - Controls the satellites
  - Make sure they work well

- **User segment (receivers):**

Image source: Garmin, gps.gov
How does Global Positioning System Work?

⇒ The principle is called **trilateration**: determining absolute or relative location of points by **measurement of distance**.

⇒ A two-dimensional example

⇒ GPS is three-dimensional, so 4 satellites!

Image source: moorefamrsbg.org
Many Additional Technologies

- GPS is certainly rocket science
- Satellites also highly complex
  - Must have a very accurate clock to encode its signal
  - Uses Einstein’s special and general theory of relativity
  - Special theory of relativity: clocks on faster moving objects are slow
    - For GPS satellites moving very fast, ~7 microseconds slower
  - General theory of relativity: clocks closer to massive objects are slower
    - Clocks on earth are ~45 microseconds slower
  - A total of ~38 microseconds difference
  - GPS must have clocks at nanosecond accuracy

- Even with these, error on the ground can be about 30 meters
  - Something called Kalman filter is used to reduce the error to about 1-5 meters