Lecture 1 - Intro to Robot Mapping

- **Robot** - device that has a means by which it moves through the environment, typically has sensors on-board.
- **Mapping** - developing a model of the environment

Related Terms

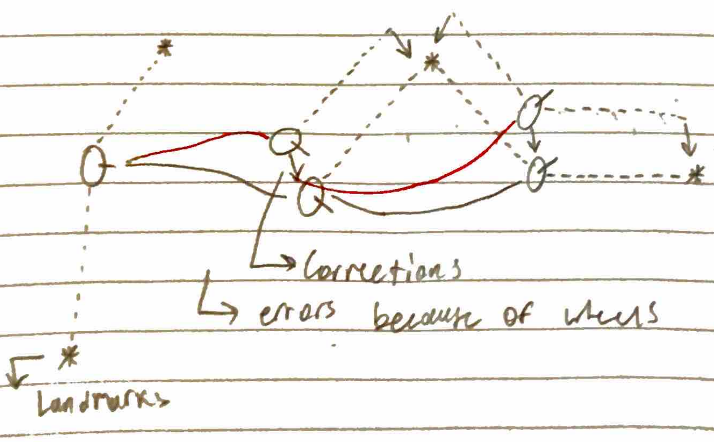
- State Estimation
  - ↳ estimate state of "world"
  - ↳ where things are
- Localization
  - ↳ an application of state estimation
  - ↳ finding position & orientation of device
- Mapping
  - ↳ estimate model of "world" using sensors
  - ↳ often know where sensor is
- SLAM
  - ↳ when you don't have sensor location you need to do "simultaneous localization and mapping"

benefit from SLAM, not in this course

- Navigation
  - ↳ steps to go through to get somewhere
- Motion Planning
  - ↳ can go beyond navigation
  - ↳ moving parts

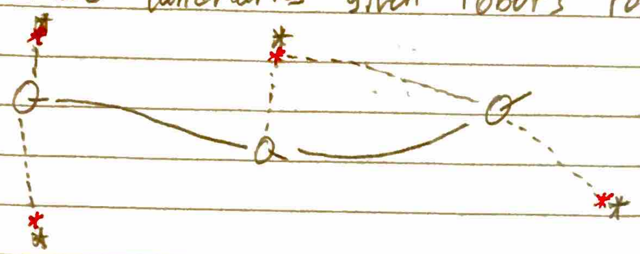
Localization Example - have a map

- estimate robot's poses given landmarks



Mapping Example - given location

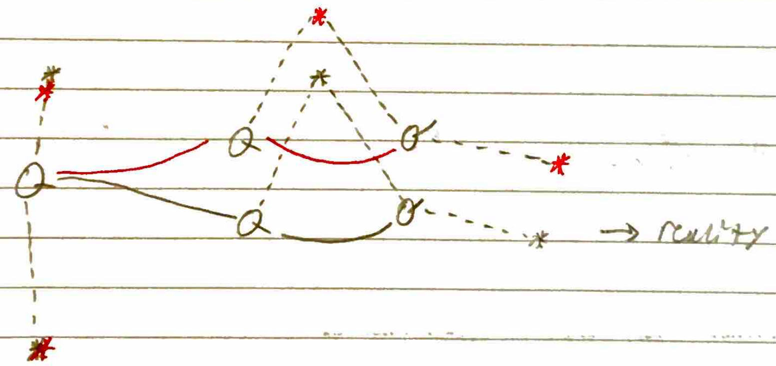
- estimate landmarks given robot's poses



↳ error because of sensors

SLAM Problem

- estimate robot's poses & landmarks simultaneously



- Note: Map estimate accuracy is worse here (than prev. diagram) since it depends on location accuracy

Chicken/egg problem:

- map needed for localization
- pose estimate is needed for mapping

→ odometry is feedback from sensors after giving a command & executing it

Defining the SLAM Problem

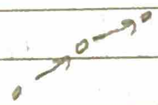
Givens:

- Robot's controls
  $u_1 : T = \{u_1, u_2, u_3, \dots u_T\}$
- Observations
  $z_1 : T = \{z_1, z_2, z_3, \dots z_T\}$

These are not free of err so we need to use probabalistic techniques

Want:

- Map of environment
  m
- Path of the robot
  $x_0 : T = \{x_0, x_1, x_2, \dots x_T\}$
  ↳ starts from 0; 1 more element than # of commands



↳ 3 poses & 2 commands

A Probabilistic World

- estimate robot's path & the map

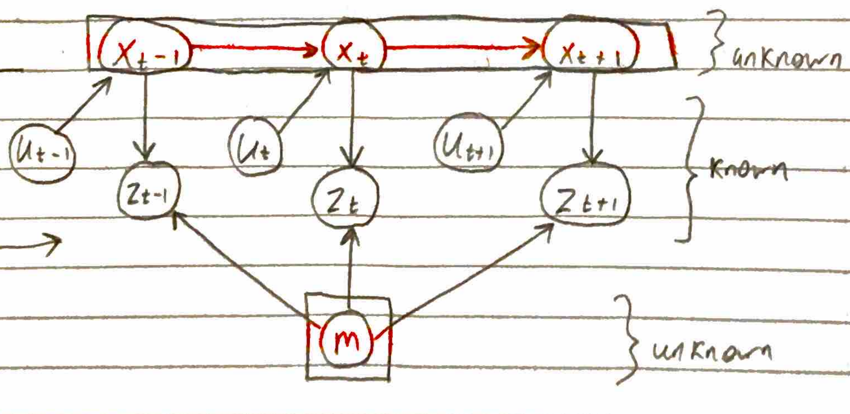
can do this using different techniques ← $\boxed{\text{estimate this given this}}$ → what this course is about

$$\underbrace{P}_{\text{Probability distribution}}(\overbrace{\underbrace{x_0 : T}_{\text{path}}, \underbrace{m}_{\text{map}}}^{\text{estimate this}} \mid \overbrace{\underbrace{z_1 : T}_{\text{Observations}}, \underbrace{u_1 : T}_{\text{Controls}}}^{\text{given this}})$$

Graphically,

- arrow means "influences"

Know as "Full SLAM" →



Full SLAM vs. Online SLAM

- Full SLAM estimates entire path

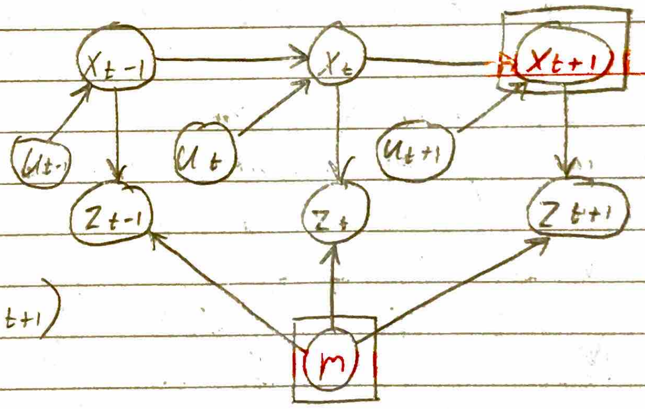
$$P(x_{0:T}, m \mid z_{1:T}, u_{1:T})$$

- Online SLAM seeks to recover only most recent pose

$$P(x_t, m \mid z_{1:t}, u_{1:t})$$

↳ most real-world applications

Graphically (online)



$P(x_{t+1}, m \mid z_{1:t+1}, u_{1:t+1})$

Online SLAM

- online SLAM means marginalizing out the prev. poses
↳ can be done through integration.

$$P(x_t, m \mid z_{1:t}, u_{1:t}) = \int_{x_0} \cdots \int_{x_{t-1}} P(x_{0:t}, m \mid z_{1:t}, u_{1:t})\, dx_{t-1} \ldots dx_0$$

because ; $\boxed{P(A, B) = P(A) - \int_B P(A, B)\, dB}$

→ at every point in time, one of these integrals is solved

Why is SLAM Hard?

1
- robot path & map both unknown
- map & pose estimates are correlated
  ↳ depend on eachother
- as you travel through the world, uncertainty of knowing where you are adds to uncertainty of sensor readings
  ↳ this keeps accumulating as you travel

Note; when you observe the same land mark more than once (from different positions), the uncertainty of where that land mark is (map) decreases, which then decreases uncertainty of previous positions.

2
- the mapping between observations & the map is unknown
  ↳ picking the wrong data association can have catastrophic consequences (divergence)

↳ in the real world, since it is impossible to track all possible data associations, we assume to have perfect correspondence
↳ there are some techniques to assume a certain # of wrong associations

Active vs Passive SLAM

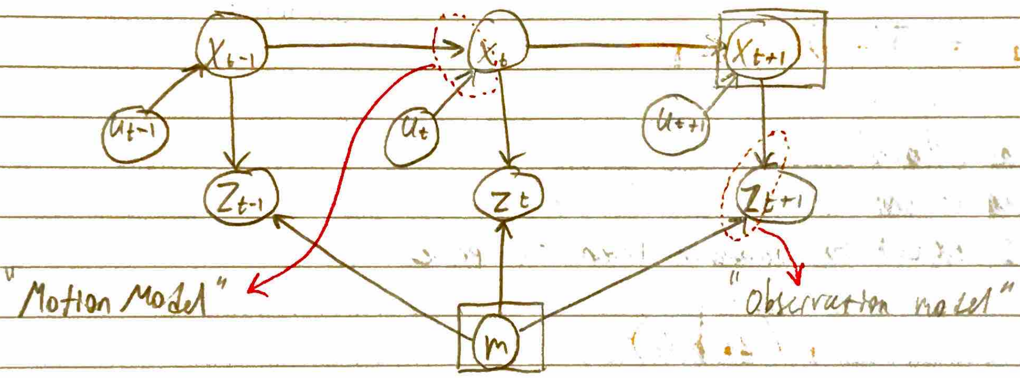
Active; robot drives it'self to decide where to go to explore properly
Passive; robot controlled by joystick
↳ mostly looking at passive in this course

Three Main Paradigms

1) Kalman Filter
2) Partical Filter
3) Graph-Based

} all of these use the below two models

Motion Model & Observation Model



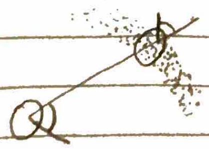
Motion Model

- describes relative motion of the model
  ↳ given I know where I was & what command will be given, I can know where I will be

$$P(x_t \mid x_{t-1}, u_t)$$

eg. Gaussian Model

eg. Non-Gaussian Model



$h = \sqrt{x^2 + y^2} \quad \rightarrow x^2 = (\Delta x)^2$

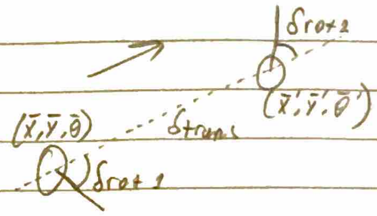
Standard Odometry Model - Motion

- Robot moves from $(\bar{x}, \bar{y}, \bar{\theta})$ to $(\bar{x}', \bar{y}', \bar{\theta}')$
- Odometry info $u = (\delta_{rot1}, \delta_{trans}, \delta_{rot2})$

$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$

$$\delta_{rot1} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$

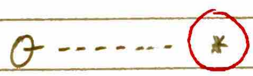


Observation Model

- relates measurments with robot's pose
- what do I expect to observe given the pose

$$P(z_t \mid x_t)$$

- eg. Gaussian Model



- eg Non-Gaussian Model