Localization

Howie Choset

Adapted from slides by Humphrey Hu, Trevor Decker, and Brad Neuman

Last updated: 27 February 2017 HDL

Localization

- General robotic task
  - "Where am I?"
- Techniques generalize to <u>many</u> estimation tasks
  - System parameter estimation
  - Noisy signal smoothing
  - Weather system modeling

Localization Problem Definition

- Goal: Estimate **state** given a history of **observations** and **actions**
  - **State:** Information sufficient to predict observations
  - **Observation:** Information derived from state
  - **Actions:** Inputs that affect the state

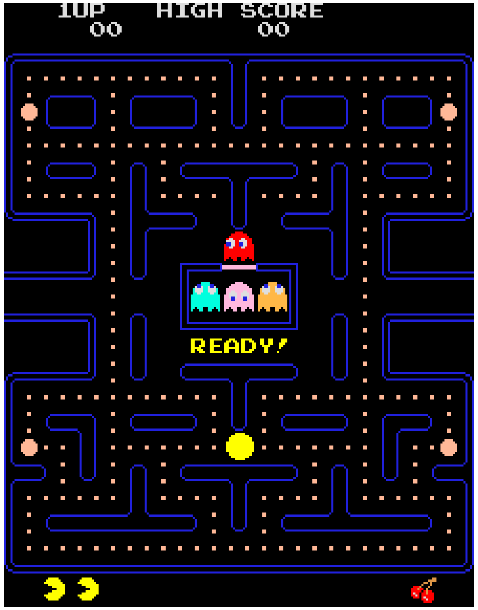
- State is important, but what is it?

State

- Any parameterization to describe our system
- Examples
  - Car / Planar Robot
    - Minimal (x, y, Θ)
    - Could be (x, y, Θ, temperature, time of day, Google stock, favorite color, etc.)
  - Consider slot car
    - Only 1D problem!

Action / Control Input

- "Things we can do"
- Can be discrete set:
  - Pacman left, right, up, down
- Or continuous inputs:
  - Helicopter throttles



Observation

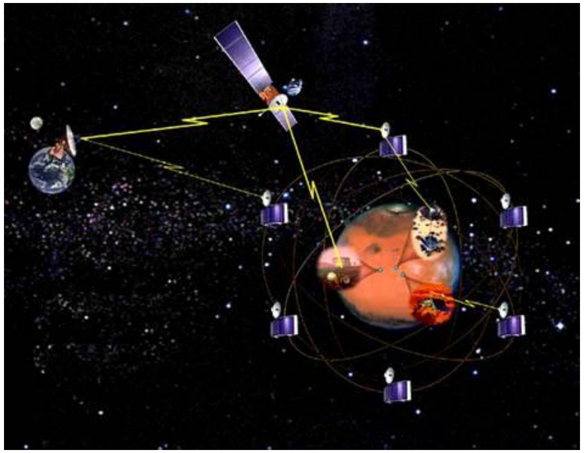
- Information derive from a sensor
- Examples
  - Time (from a clock)
  - Temperature (thermoater)
  - Encoder
- Good vs. Bad (really strong vs. weak correlations)
  - ie. Temperature at a city for estimating a weather system

Absolute vs. Relative

- Absolute Localization
  - Defines state relative to a common (usually fixed) reference frame
  - Useful for coordination, navigation, etc.
- Relative Localization
  - Defines state relative to a local (non-shared) reference frame
  - Useful for exploration, displacement estimation

Why not GPS/Vicon?

- Signal-denied environments
- Insufficient performance
  - Accuracy
  - Bandwidth (update rate)
  - Bias
  - Receiver size



Why not odometry?

- Uncertainty and error accumulation!
  - Unmodeled environmental factors
  - Integration errors
  - Modeling errors
- Bottom line:

Uncertainty is a part of life.

We have to deal with it!

Localization Problem Definition

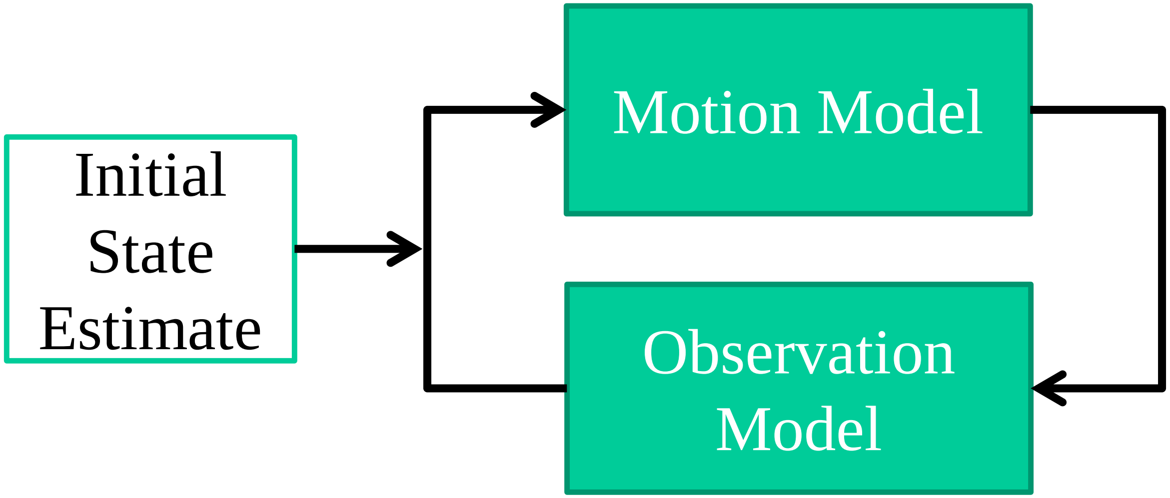
- Goal: Estimate state given a history of observations and actions
  - **State:** Information sufficient to predict observations
  - **Observation:** Information derived from state
  - **Actions:** Inputs that affect the state

Localization Problem Definition

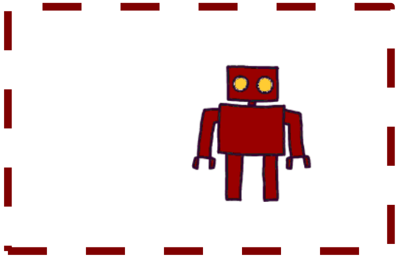
- Goal: Estimate state given a history of *noisy* observations and *noisy* actions
  - **State:** Information sufficient to predict observations
  - **Observation:** Information derived from state
  - **Actions:** Inputs that affect the state

Localization: Estimate State

- Move: Motion Model
- Observe: Observation Model



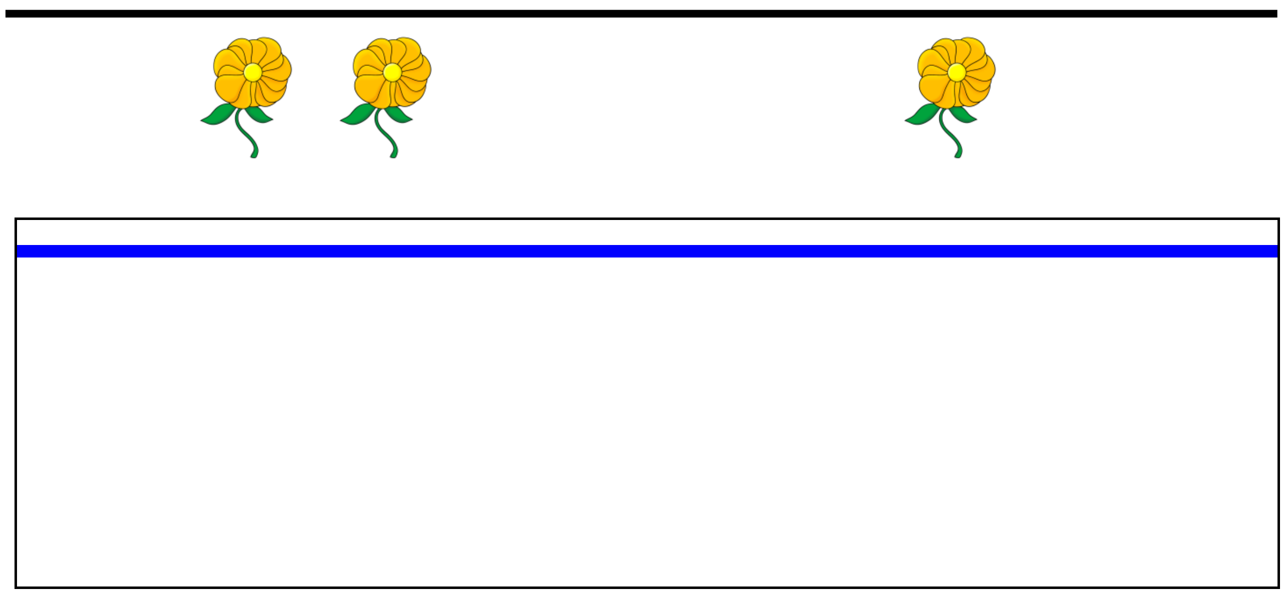
Example



- Moving only in one dimension
- Known map of flower garden
- Simple flower detector
    - Beeps when you are in front of a flower
    - Gaussian distribution of a flower given a beep

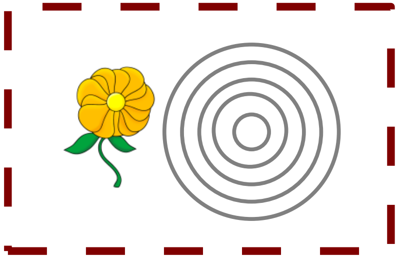
Example

- Initially, no idea where we are



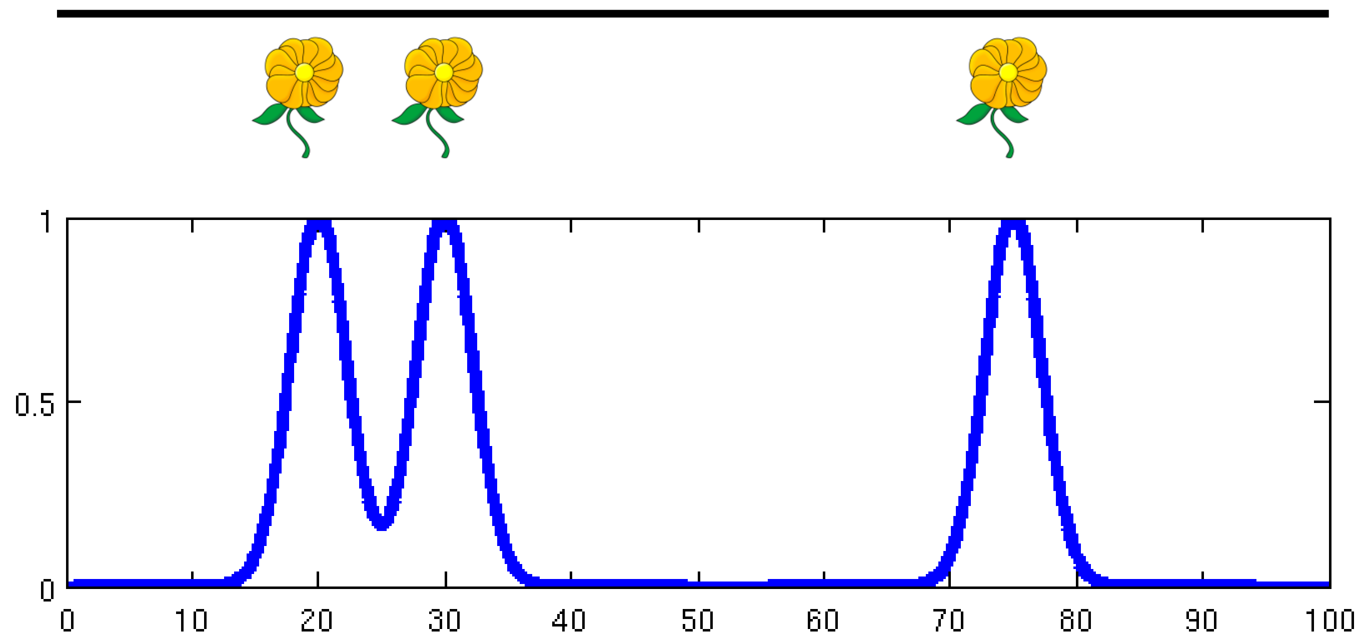
Example

- First observation update



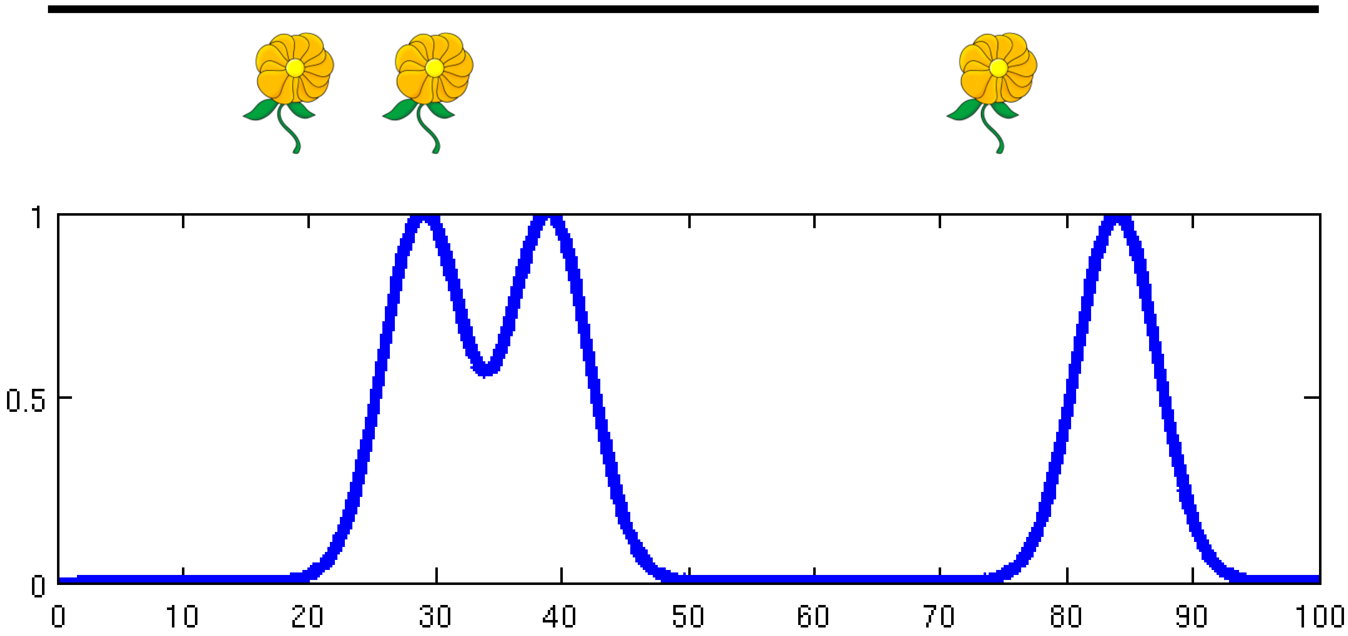
Example

- New belief about location



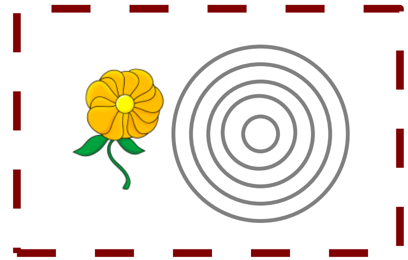
Example

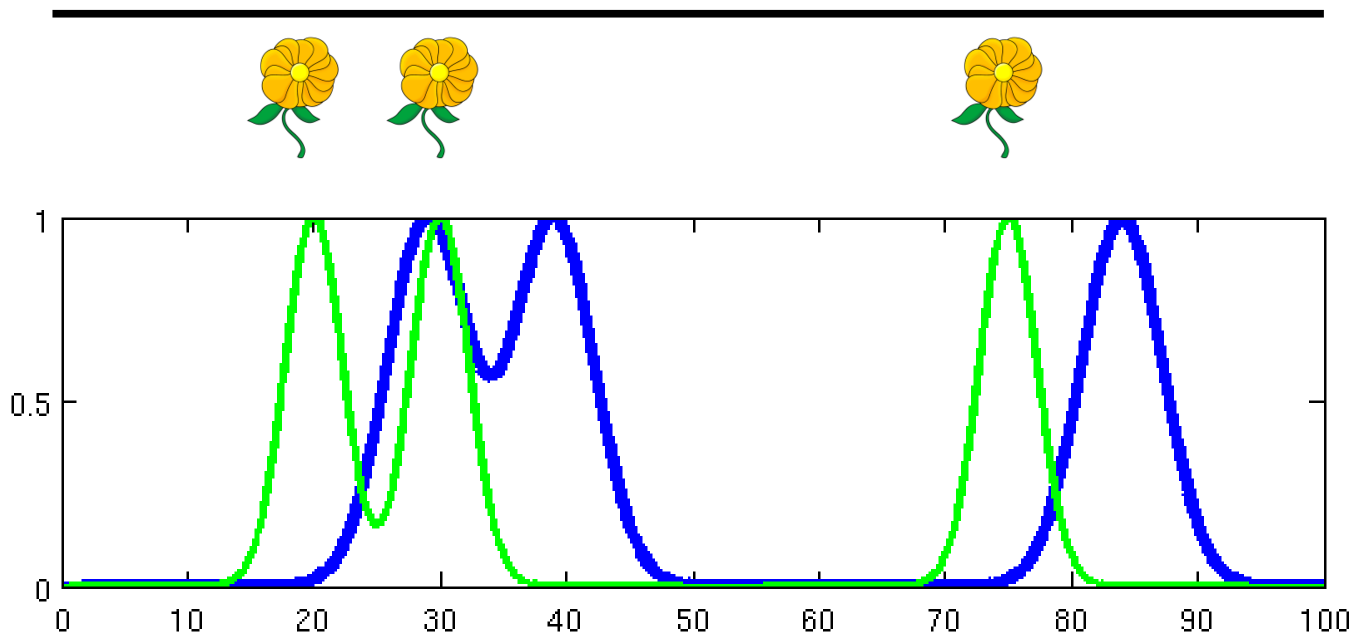
- Robot moves, motion update



Example

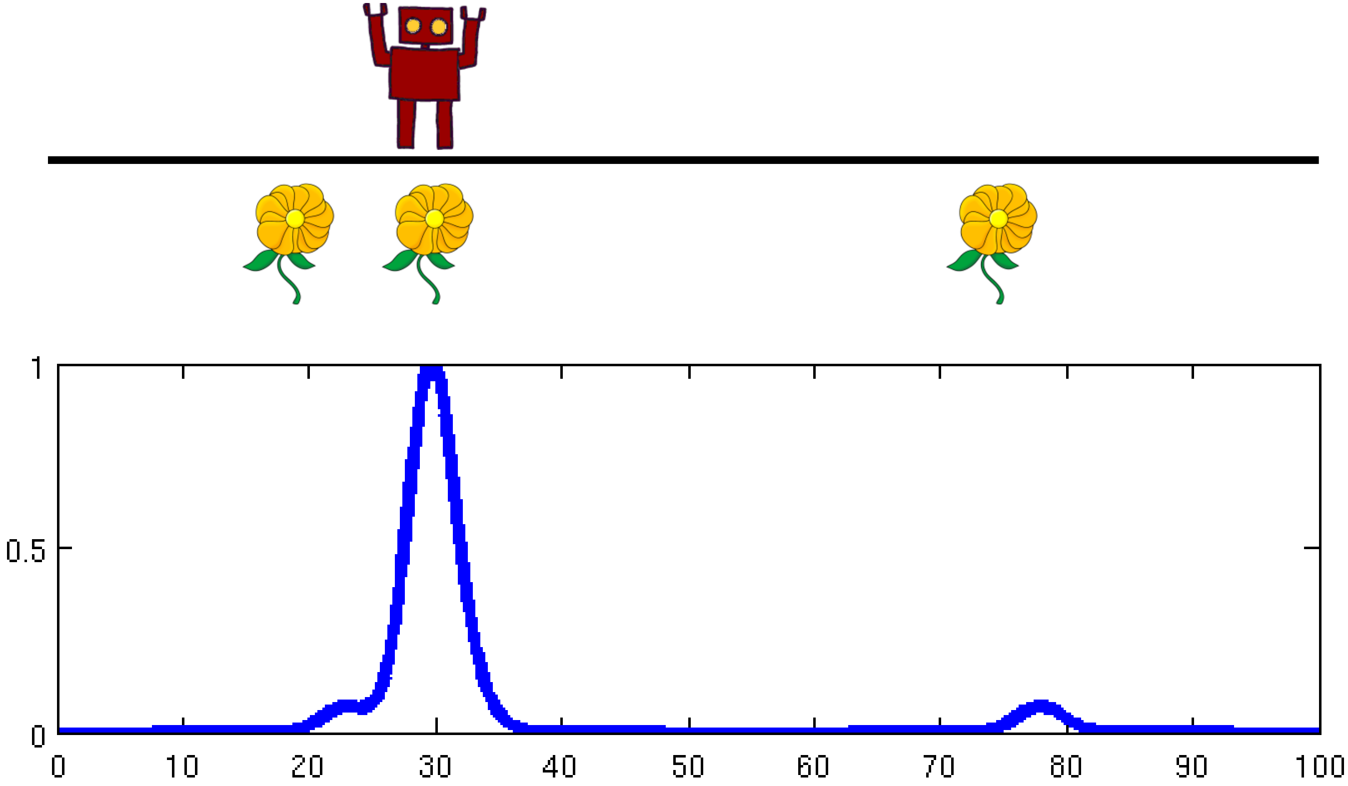
- Observation update





Example

- Final Belief



Differential Drive Motion Model

- **State**: (x, y, Θ) *SE2* pose
- **Actions**:
  - – Drive forward, angle --OR--
  - – Drive wheel 1, drive wheel 2
- **Measurements**: Wheel rotation ticks

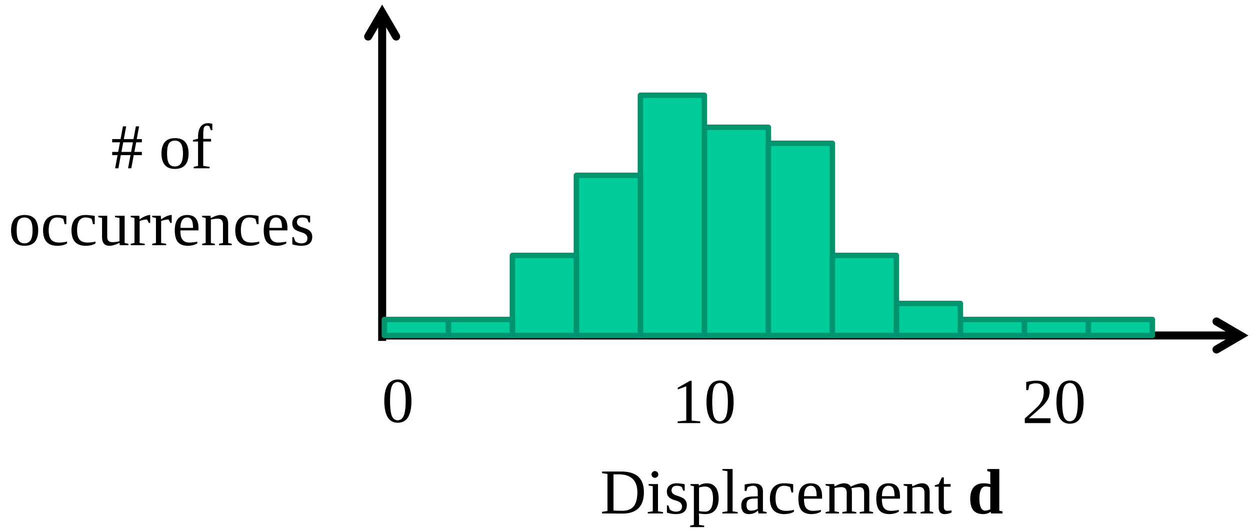
2-wheeled Lego Robot

Differential Drive Example

- Run the same trajectory many times
  - They're all different!
  - Why?

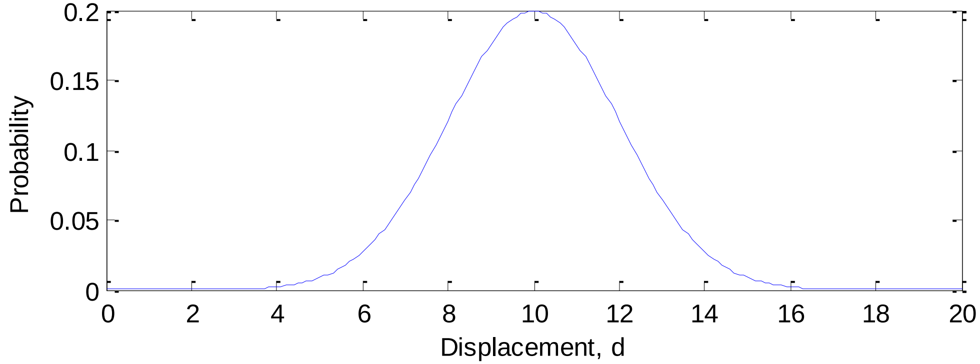
Differential Drive Example

- Run a 10 cm straight trajectory many times
- Look at the results as a distribution



Differential Dive Example

- Run a 10 cm straight trajectory many times
- Look at the results as a distribution



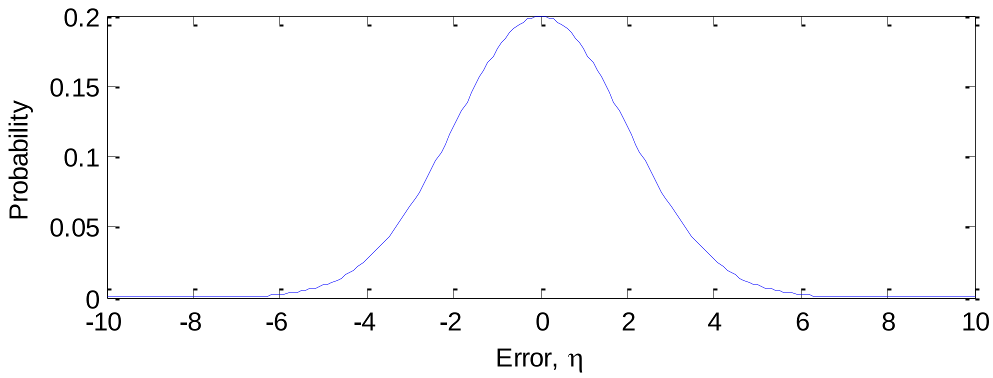
$$\eta \sim \mathcal{N}(\mu, \sigma^2)$$

$$x^{t+1} = f(x^t) + \eta$$

Differential Drive Example

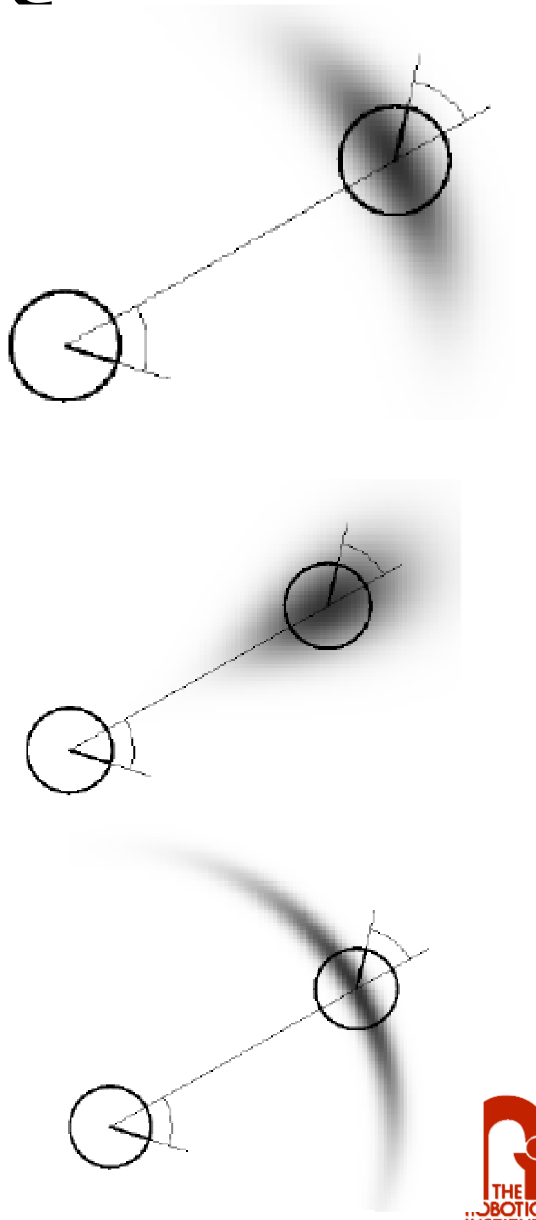
- Subtract out model contribution to determine noise component

$$x^{t+1} = f(x^t, u^t) + \eta^t$$



Odometry Example

- Differential drive robot experiences uncertainty in distance traveled and heading
  - Produces a "banana" distribution
  - Hard to model!

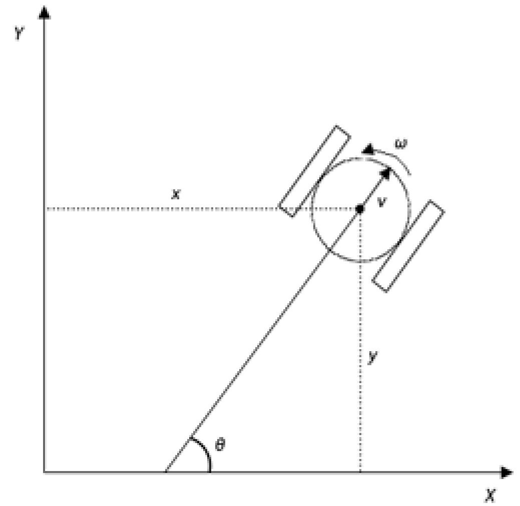


Differential Drive Sensor Model

- Recall our odometry equations:

$$v_l = \frac{\text{Current Encoder Ticks (left motor)} - \text{Encoder Ticks saved from previous loop (left motor)}}{\text{Time elapsed since we last polled the encoders}} \frac{\pi}{180}$$

$$v_r = \frac{\text{Current Encoder Ticks (right motor)} - \text{Encoder Ticks saved from previous loop (right motor)}}{\text{Time elapsed since we last polled the encoders}} \frac{\pi}{180}$$



observations

$$V_l = v_l R$$
$$V_r = v_r R$$
$$v = \frac{V_r + V_l}{2}$$
$$\omega = \frac{V_r - V_l}{L}$$

effect on state

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \end{pmatrix}$$

$$k_{00} = v\cos(\theta_{n-1})$$
$$k_{01} = v\sin(\theta_{n-1})$$
$$k_{02} = \omega$$

$$k_{10} = v\cos\left(\theta_{n-1} + \tfrac{t}{2}k_{02}\right)$$
$$k_{11} = v\sin\left(\theta_{n-1} + \tfrac{t}{2}k_{02}\right)$$
$$k_{12} = \omega$$

$$k_{20} = v\cos\left(\theta_{n-1} + \tfrac{t}{2}k_{12}\right)$$
$$k_{21} = v\sin\left(\theta_{n-1} + \tfrac{t}{2}k_{12}\right)$$
$$k_{22} = \omega$$

$$k_{30} = v\cos(\theta_{n-1} + t\,k_{22})$$
$$k_{31} = v\sin(\theta_{n-1} + t\,k_{22})$$
$$k_{32} = \omega$$

$$\begin{pmatrix} x_n \\ y_n \\ \theta_n \end{pmatrix} = \begin{pmatrix} x_{n-1} \\ y_{n-1} \\ \theta_{n-1} \end{pmatrix} + \frac{t}{6}\begin{pmatrix} k_{00} + 2(k_{10} + k_{20}) + k_{30} \\ k_{01} + 2(k_{11} + k_{21}) + k_{31} \\ k_{02} + 2(k_{12} + k_{22}) + k_{32} \end{pmatrix}$$

“initial” state

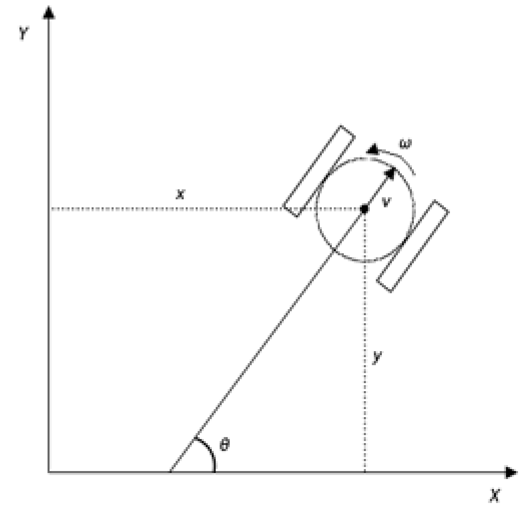
Motion

Differential Drive ~~Sensor~~ Model

- Recall our odometry equations:

$$v_l = \frac{\text{Current Encoder Ticks (left motor)} - \text{Encoder Ticks saved from previous loop (left motor)}}{\text{Time elapsed since we last polled the encoders}} \frac{\pi}{180}$$

$$v_r = \frac{\text{Current Encoder Ticks (right motor)} - \text{Encoder Ticks saved from previous loop (right motor)}}{\text{Time elapsed since we last polled the encoders}} \frac{\pi}{180}$$



observations

$$V_l = v_l R$$
$$V_r = v_r R$$
$$v = \frac{V_r + V_l}{2}$$
$$\omega = \frac{V_r - V_l}{L}$$

effect on state

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \end{pmatrix}$$

$$k_{00} = v\cos(\theta_{n-1})$$
$$k_{01} = v\sin(\theta_{n-1})$$
$$k_{02} = \omega$$

$$k_{10} = v\cos\left(\theta_{n-1} + \frac{t}{2} k_{02}\right)$$
$$k_{11} = v\sin\left(\theta_{n-1} + \frac{t}{2} k_{02}\right)$$
$$k_{12} = \omega$$

$$k_{20} = v\cos\left(\theta_{n-1} + \frac{t}{2} k_{12}\right)$$
$$k_{21} = v\sin\left(\theta_{n-1} + \frac{t}{2} k_{12}\right)$$
$$k_{22} = \omega$$

$$k_{30} = v\cos(\theta_{n-1} + t\, k_{22})$$
$$k_{31} = v\sin(\theta_{n-1} + t\, k_{22})$$
$$k_{32} = \omega$$

$$\begin{pmatrix} x_n \\ y_n \\ \theta_n \end{pmatrix} = \begin{pmatrix} x_{n-1} \\ y_{n-1} \\ \theta_{n-1} \end{pmatrix} + \frac{t}{6} \begin{pmatrix} k_{00} + 2(k_{10} + k_{20}) + k_{30} \\ k_{01} + 2(k_{11} + k_{21}) + k_{31} \\ k_{02} + 2(k_{12} + k_{22}) + k_{32} \end{pmatrix}$$

“initial” state

A Brief Overview of

PROBABILITY

Discrete Probability Distribution

- Let X be the value of a die roll
- X is unknown (a Random Variable)
- P(X = v) means "Probability that we sample X and it equals v"

v	P(X=v)
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

Discrete Probability Distribution

- Let X be the value of a die roll
- X is unknown (a Random Variable)
- P(X = v) means "Probability that we sample X and it equals v"

v	P(X=v)
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

Sums to 1

Discrete Probability Distribution

- This time, X is a weighted die
- This is a different distribution for the same variable

v	P(X=v)
1	0.1
2	0.1
3	0.1
4	0.2
5	0.25
6	0.25

Discrete Probability Distribution

- Consider a sum of dice

And = x   Or = +

v	P(X1=v)
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

v	P(X2=v)
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

v	P(X1 + X2 = v)
2	P(X1 = 1) * P(X2 = 1)
3	P(X1 = 1) * P(X2 = 2) + P(X1 = 2) * P(X2 = 1)
4	P(X1 = 1) * P(X2 = 3) + …
5	
6	
7	
8	
9	
10	
11	
12	

Discrete Probability Distribution

- Can we separate the probabilities?
- P(X2 = hot | X1 = summer) is high; P(X2 = hot | X1 = winter) is low
- P(X2 = v | X1 = 1) means "Probability that roll 2 is v if roll 1 is 1"
  - Independent variables

v	P(X2=v | X1 = 1)
1	1/6
2	1/6
3	1/6
4	1/6
5	1/6
6	1/6

Discrete Probability Distribution

- Using conditional probabilities allows us to easily combine:

$$P(x, y) = \sum_x \sum_y P(x|y)P(y)$$

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

Bayes theorem

Putting it Together

- At time step 0:
  1. Robot takes an action

$$x^1 = f(x^0, u^0) + \eta^0$$

  2. Robot makes an observation

$$z^1, h(x^1)$$

Distribution of measurement at time step 1:

Putting it Together

- At time step 0:
  1. Robot takes an action

$$x^1 = f(x^0, u^0) + \eta^0$$

  2. Robot makes an observation

$$z^1, h(x^1)$$

Distribution of measurement at time step 1:

$p(x^1 \mid u^0, z^1) =$

Putting it Together

- At time step 0:
  1. Robot takes an action

$$x^1 = f(x^0, u^0) + \eta^0$$

  2. Robot makes an observation

$$z^1, h(x^1)$$

Distribution of position at time step 1:

$$p(x^1 \mid u^0, z^1) \propto \underbrace{p(z^1 \mid x^1)}_{\text{Observation Model}} \; \underbrace{p(x^1 \mid x^0, u^0)}_{\text{Transition Model}} \; \underbrace{p(x^0)}_{\text{Prior}}$$

Observation Model   Transition Model   Prior

Putting it Together

- At time step 1:
  1. Robot takes an action

$$x^2 = f(x^1, u^1) + \eta^1$$

  2. Robot makes an observation

$$z^2, h(x^2)$$

Distribution of position at time step 1:

Putting it Together

- At time step 1:

  1. Robot takes an action

$$x^2 = f(x^1, u^1) + \eta^1$$

  2. Robot makes an observation

$$z^2, h(x^2)$$

Distribution of position at time step 1:

$p(x^2 \mid u^0, z^1, u^1, z^2) =$

Putting it Together

- At time step 1:
  1. Robot takes an action

$$x^2 = f(x^1, u^1) + \eta^1$$

  2. Robot makes an observation

$$z^2, h(x^2)$$

Distribution of position at time step 1:

$$p(x^2 \mid u^0, z^1, u^1, z^2)$$

$$\propto \underbrace{p(z^2 \mid x^2)\, p(x^2 \mid x^1, u^1)}_{\text{New trans. \& obs. model}}\ \underbrace{p(z^1 \mid x^1)\, p(x^1 \mid x^0, u^0)\, p(x^0)}_{\text{Previous result}}$$

New trans. & obs. model — Previous result

Putting it Together

- At time step 1:
  1. Robot takes an action

$$x^2 = f(x^1, u^1) + \eta^1$$

  2. Robot makes an observation

$$z^2, h(x^2)$$

Distribution of position at time step 1:

$$p(x^2 \mid u^0, z^1, u^1, z^2)$$
$$\propto p(z^2 \mid x^2)\, p(x^2 \mid x^1, u^1)\, p(x^1 \mid u^0, z^1)$$

New trans. & obs. model — Previous result

Recursive Inference

- At time step t:
  1. Robot takes an action

$$x^t = f(x^{t-1}, u^{t-1}) + \eta^{t-1}$$

  2. Robot makes an observation

$$z^t, h(x^t)$$

Distribution of position at time step 1:

$p(x^t \mid u^0, \ldots, u^t, z^1, \ldots, z^t)$

$\propto p(z^t \mid x^t)\, p(x^t \mid x^{t-1}, u^{t-1}) p(x^{t-1} \mid u^0, \ldots, u^{t-1}, z^1, \ldots, z^{t-1})$

New trans. & obs. model — Previous result

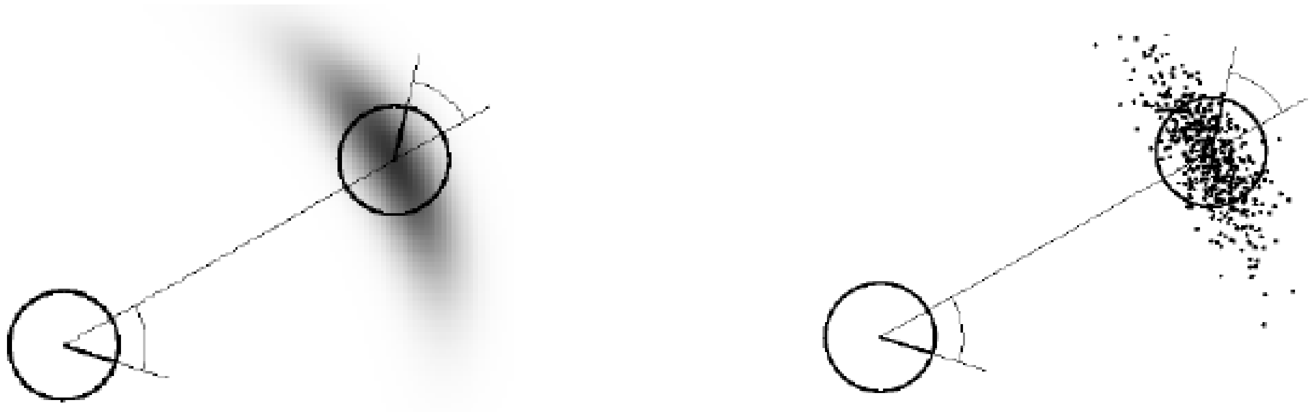
Filtering Algorithm for Localization

- For each possible location:
  - Apply motion model
- For each possible location:
  - Apply observation model
- Loop forever

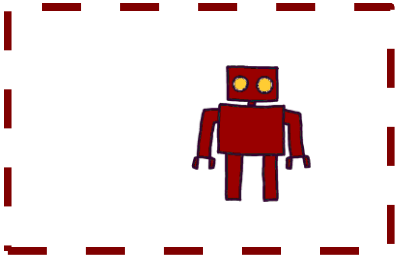
Filtering Algorithm for Localization

- For each possible location:
  - Apply motion model
- For each possible location:
  - Apply observation model
- Loop forever
-
- **Too slow!**
- Let's use discrete hypotheses instead

Sampling From the Motion Model



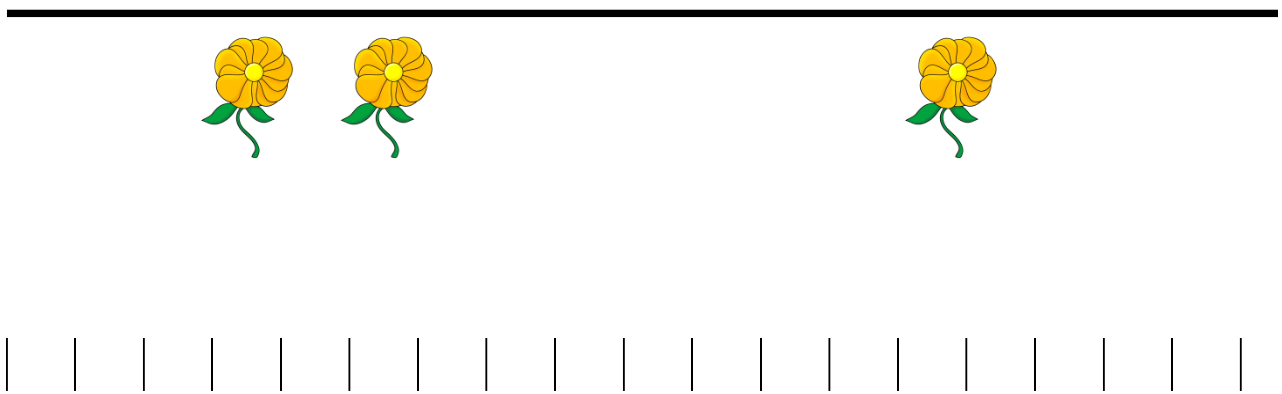
Example Revisited



- Same flower-happy robot
- Same map
- This time, track samples (particles)

Example Revisited

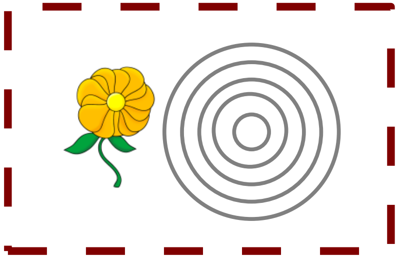
- Initially, no idea where we are

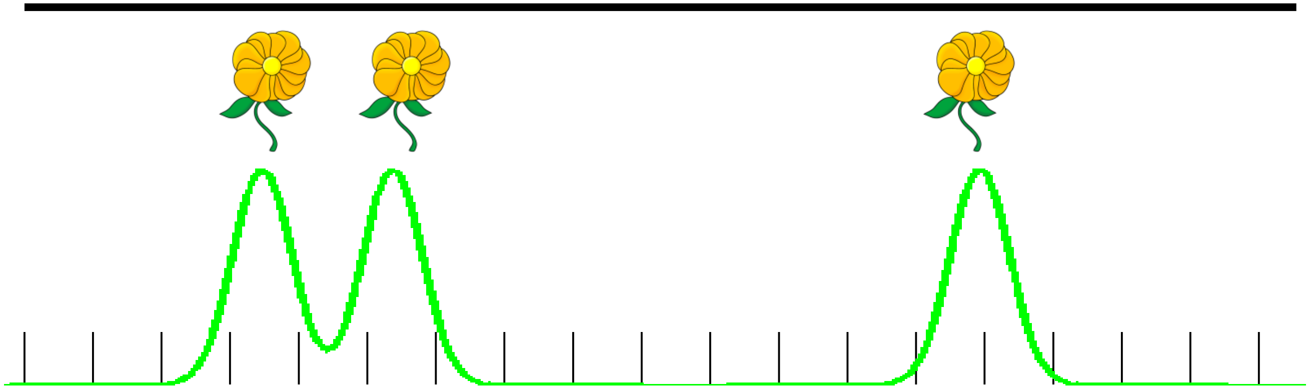


The particles. Height represents the weight (probability or confidence) of a given particle

Example

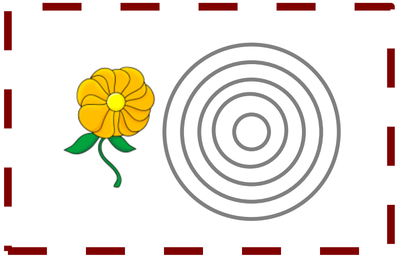
- First observation update

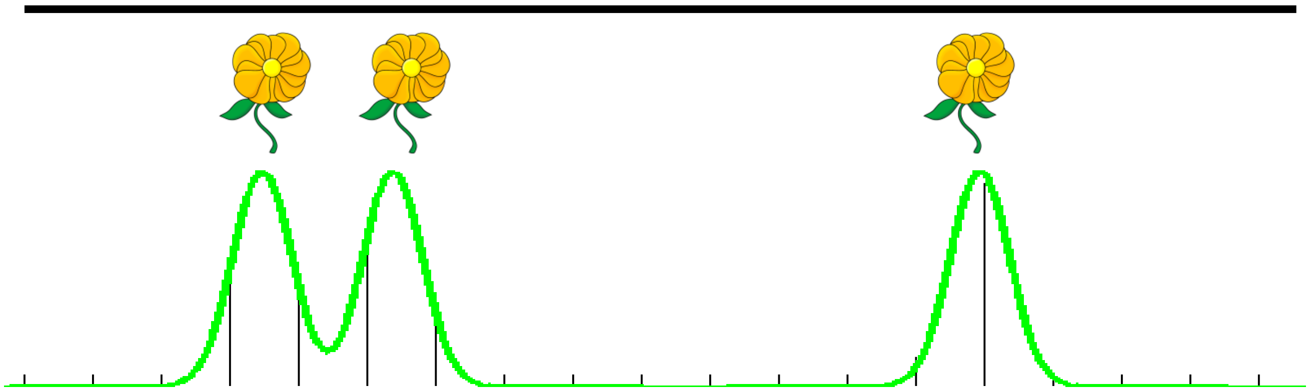




Example

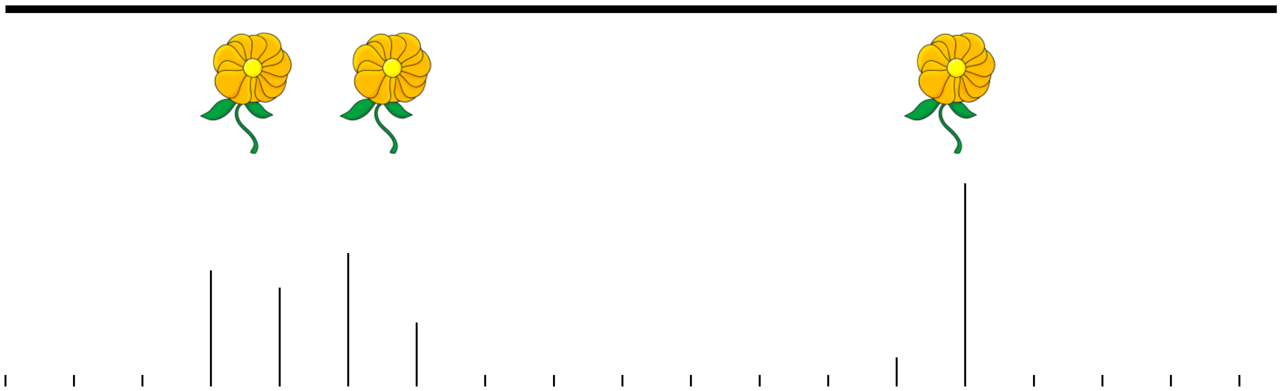
- First observation update
- Evaluate model at particles





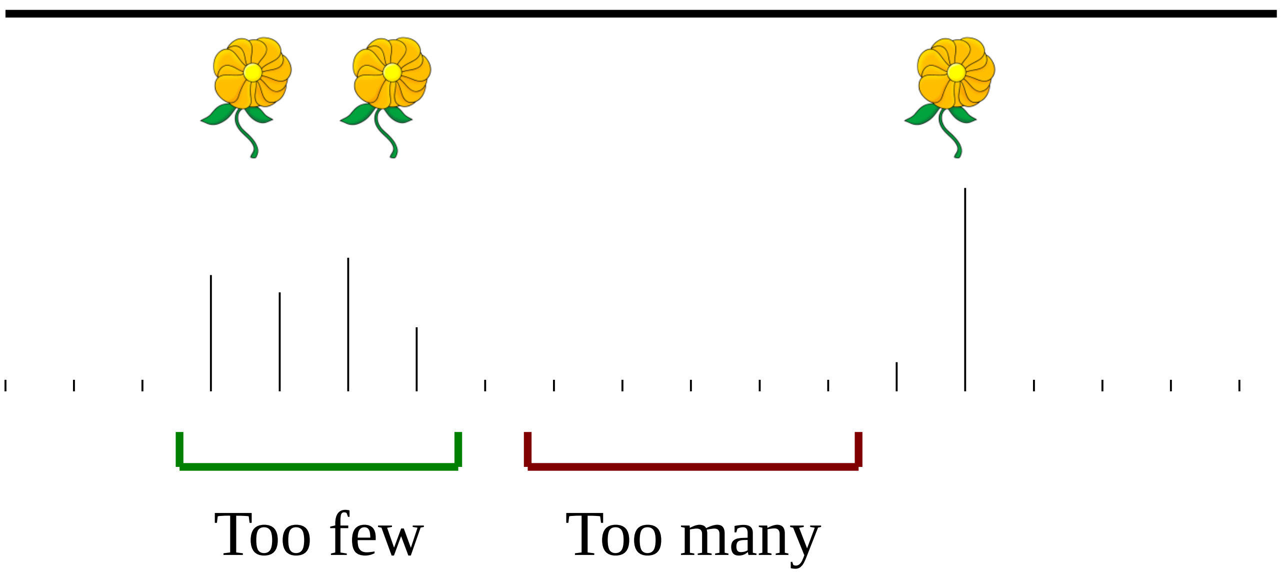
Example

- New belief



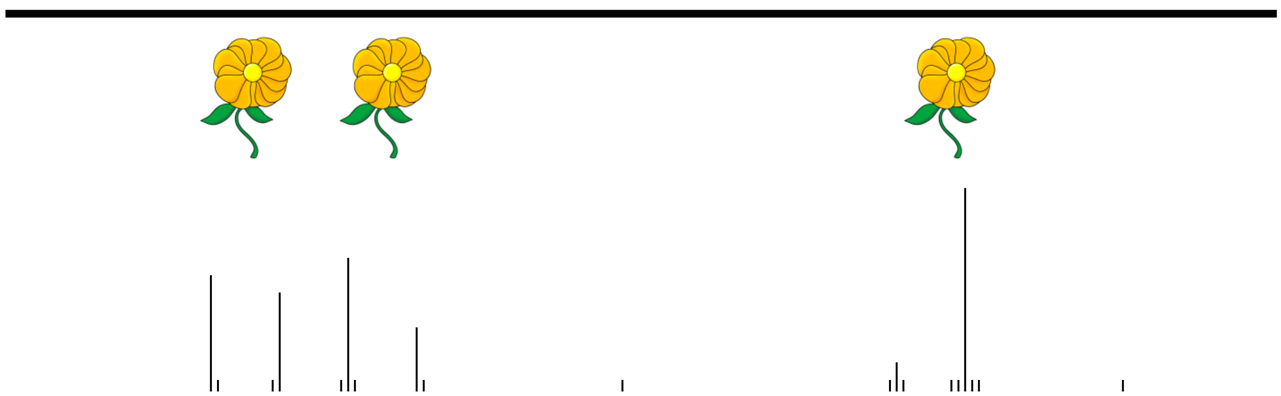
Example

- New belief



Example

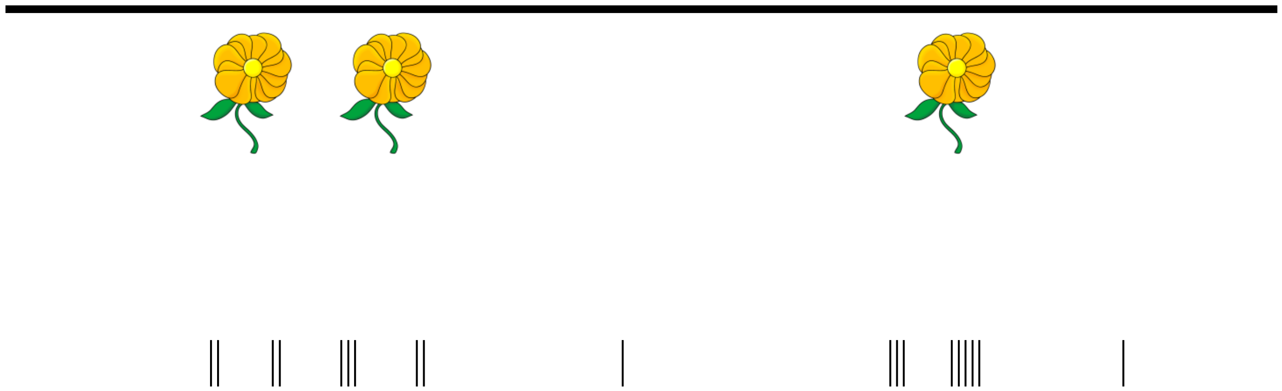
- Resample particles



Higher weight particles get more particles allocated near them during the resample

Example

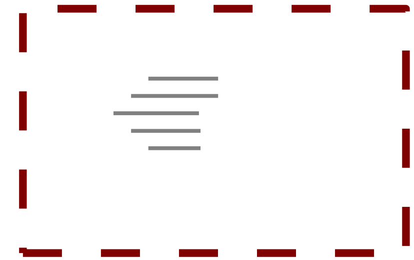
- Reset weights

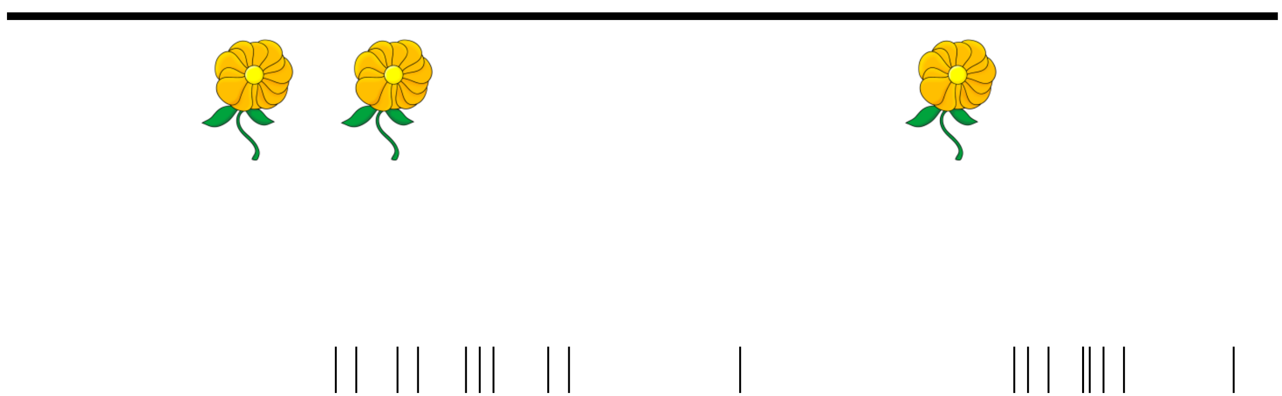


Density of particles is related to weight of particles previously

Example

- Robot moves, motion update

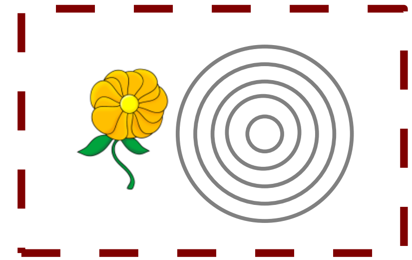


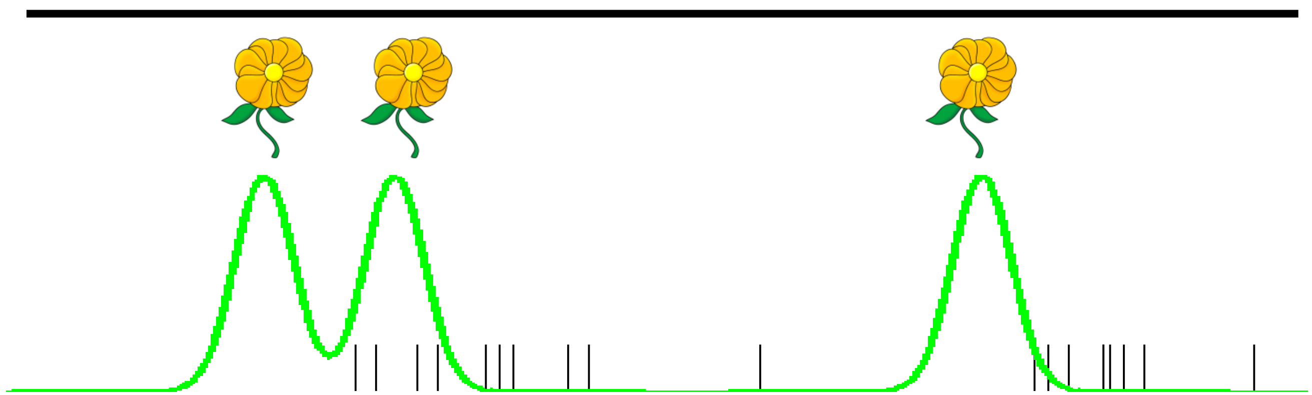


Particles spread out

Example

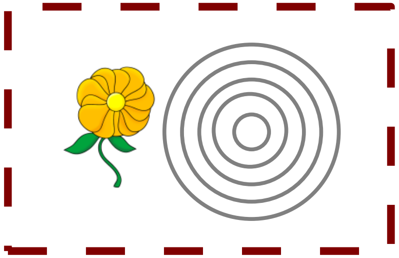
- Observation update

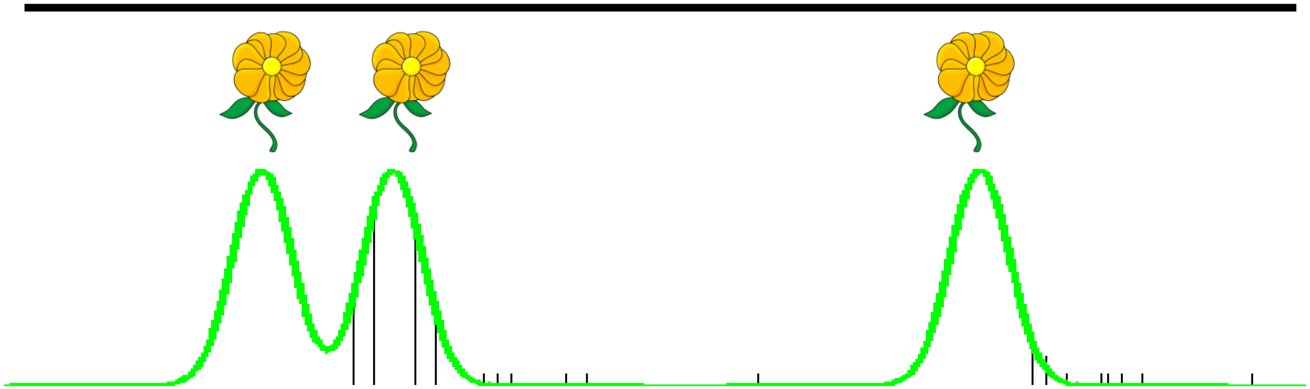




Example

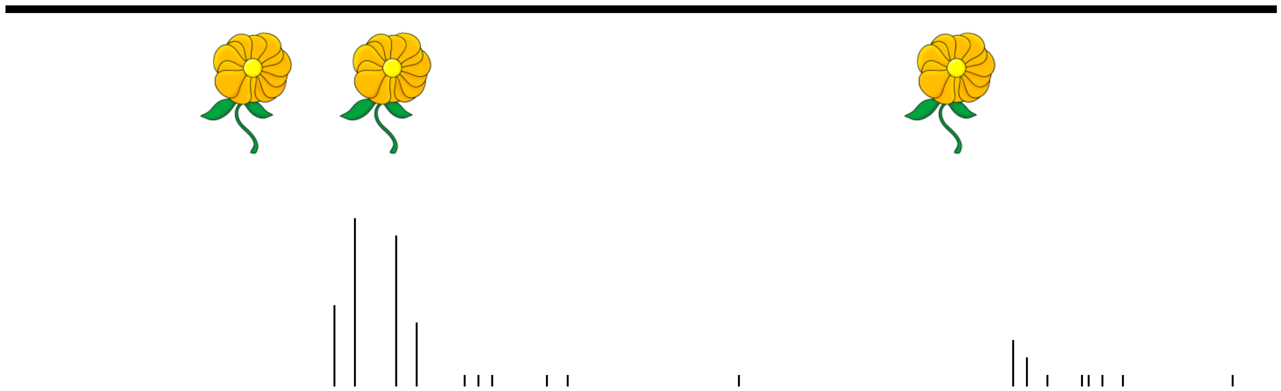
- Observation update



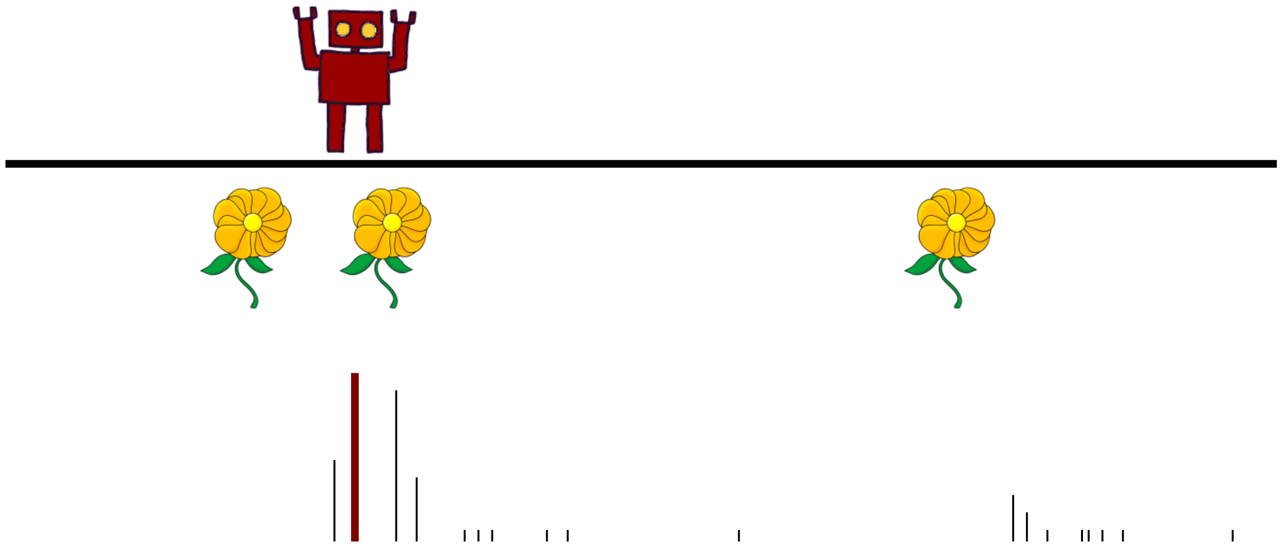


Example

- Estimate is best particle



Example



Discrete Bayes Filter

Algorithm 1: Discrete Bayes Filter Transition Update

Data: Discretized state grid X

N-D probability map m

Action u

Copy $mPrev \leftarrow m$;

for *All states* $x \in X$ **do**

$m[x] \leftarrow \sum_{y \in X} p(x|y,u) \cdot mPrev[y]$;

end

Normalize m;

Discrete Bayes Filter

Algorithm 2: Discrete Bayes Filter Measurement Update

Data:
Discretized state grid X
Landmark locations L
N-D probability map m
Observation z
Copy $mPrev \leftarrow m$;
for *All states* $x \in X$ **do**
    $m[x] \leftarrow \sum_{l \in L} p(z|x, l) \cdot mPrev[x]$;
end
Normalize m;