

# Midterm Review 2 March

Hannah Lyness

# Vision



We can store images as arrays of values



It is difficult for a computer program to identify objects (Where's Waldo)

# Stereo Vision



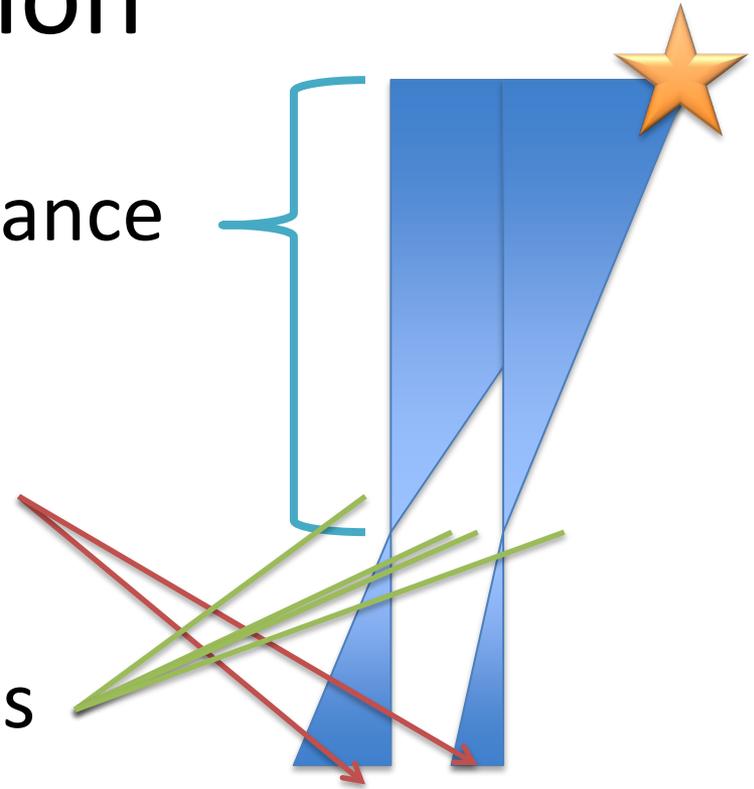
One perpendicular distance



Two separate pictures



Corresponding triangles



# Example Problem



Object is 20 pixels to the right center in left image, 10 pixels from center in the right image, we are at the exact same height as the objects.

Cameras are 4 in apart with distance between lens and image plane of 2 in.

Resolution is 20 ppi.

# Machine Learning



A Neural Net is just a series of weights



We used a neural net to classify Waldos from boxes

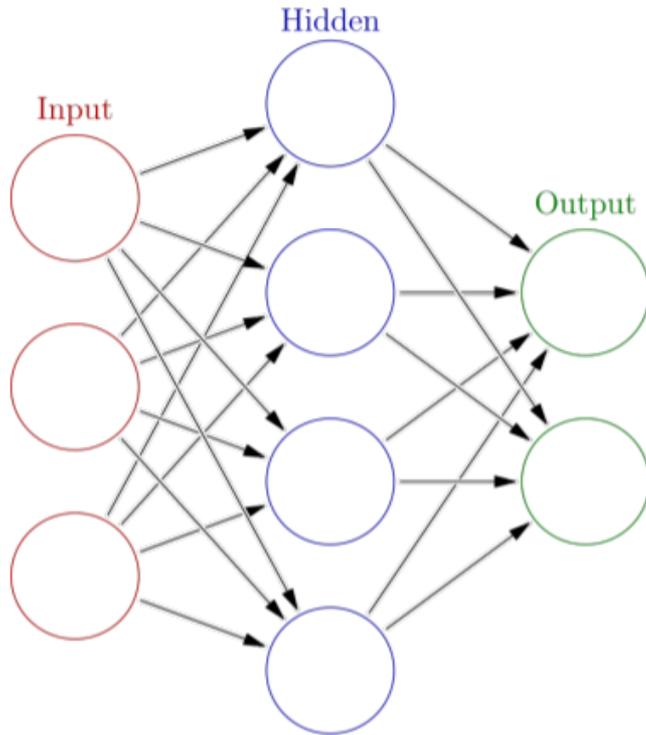


We used a neural net to learn desirable actions for a cartpole



Usual set up: training data, testing data.  
Train then validate.

# Neural Nets



A feed-forward artificial neural network is a computation graph which consists of neurons, each computing a linear combination of the previous layer composed with an activation function.

The final layer is considered the output layer.

If there are any layers of the network which are not the input or output layers, those are considered hidden layers, and the network is a “deep” neural network.

Note that the input layer does not have any weights associated with it.

# Neural Nets - Caveats

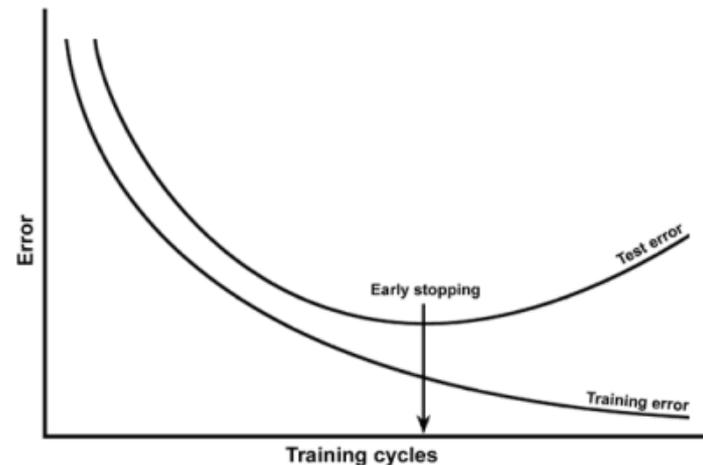
Black box



Requires a lot of data, time

(Image Source: <https://www.learnopencv.com/neural-networks-a-30000-feet-view-for-beginners/>)

Can overfit



# Machine Learning

- Parametric vs. Non-parametric Learning
- Supervised vs. Unsupervised Learning
- Reinforcement Learning: train an agent to map observations to actions
- Imitation Learning: train an agent to copy an expert

# Controls



PID is great



$K_p$  is like spring constant



$K_d$  is like damping coefficient



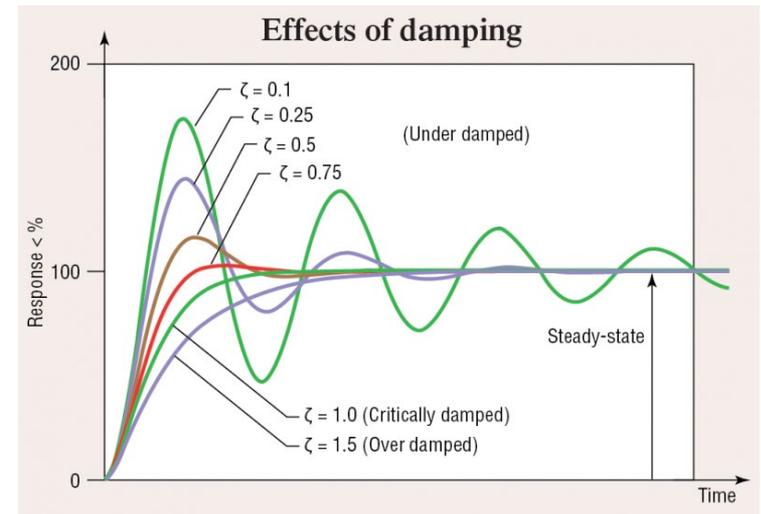
$K_i$  is like a friend that gently pulls you towards the correct path when you are converging on a different one

# Controls Examples

<https://youtu.be/IZPtFDXYQRU?t=1m>

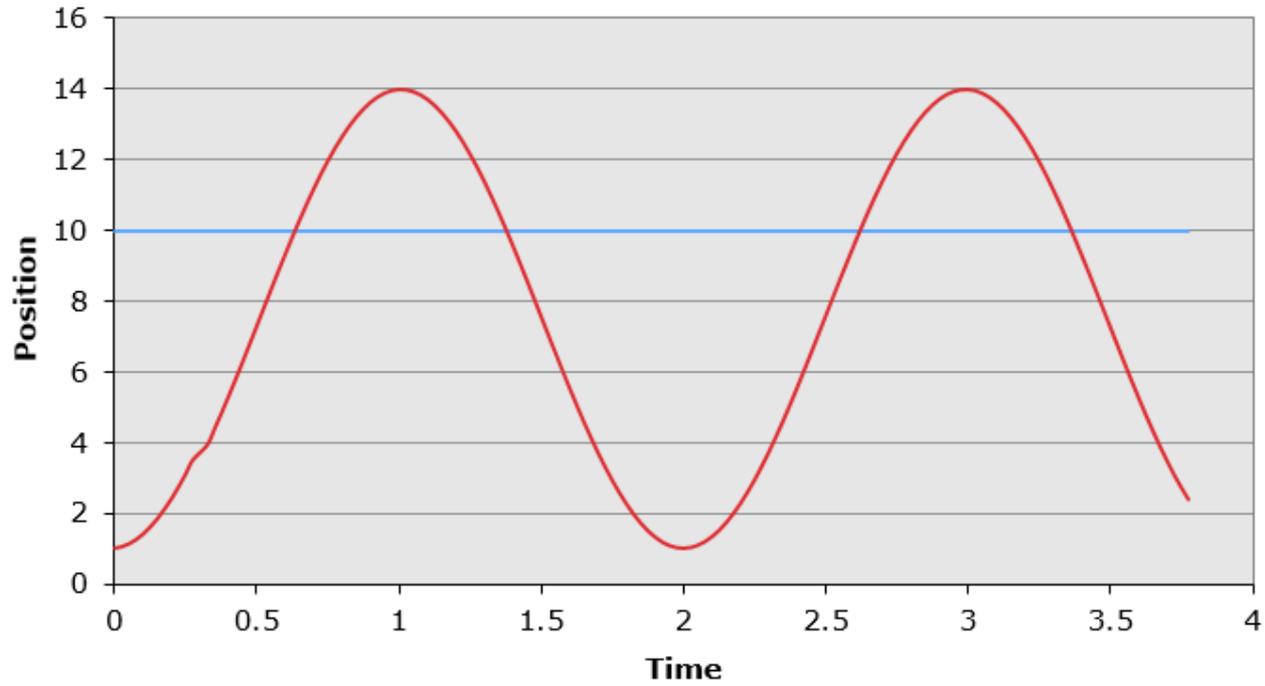
<https://youtu.be/sP1DzhT8Vzo?t=59>

<https://www.cs.cmu.edu/afs/cs/academic/class/15883-f15/lectures/cerebellum-controller/pid.xls>



# Controls Review

Control Motion



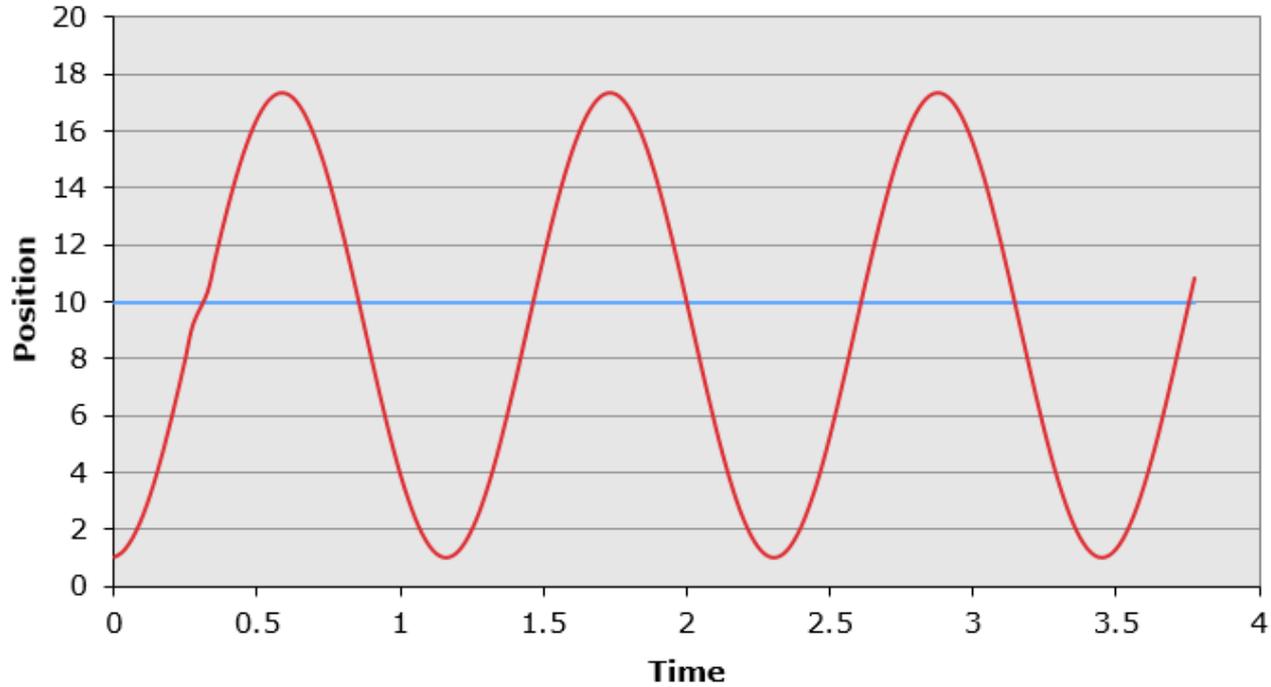
$$K_p = 1$$

$$K_d = 0$$

$$K_i = 0$$

# Controls Review

## Control Motion



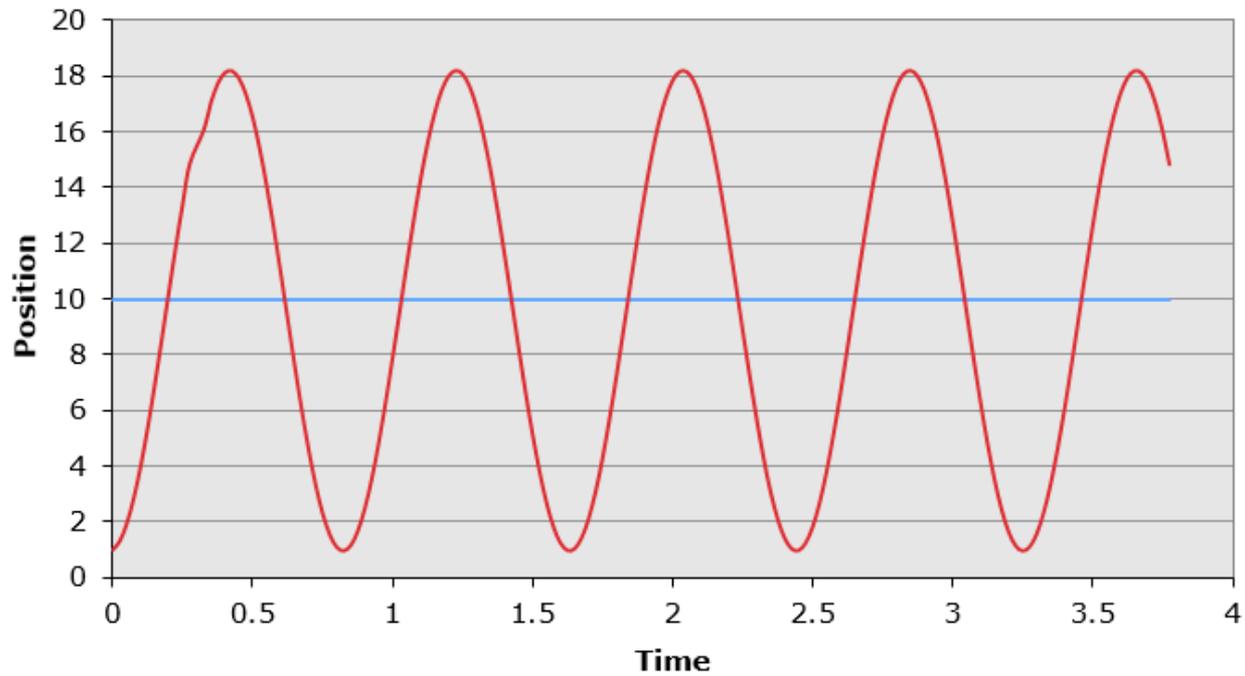
$$K_p = 3$$

$$K_d = 0$$

$$K_i = 0$$

# Controls Review

Control Motion

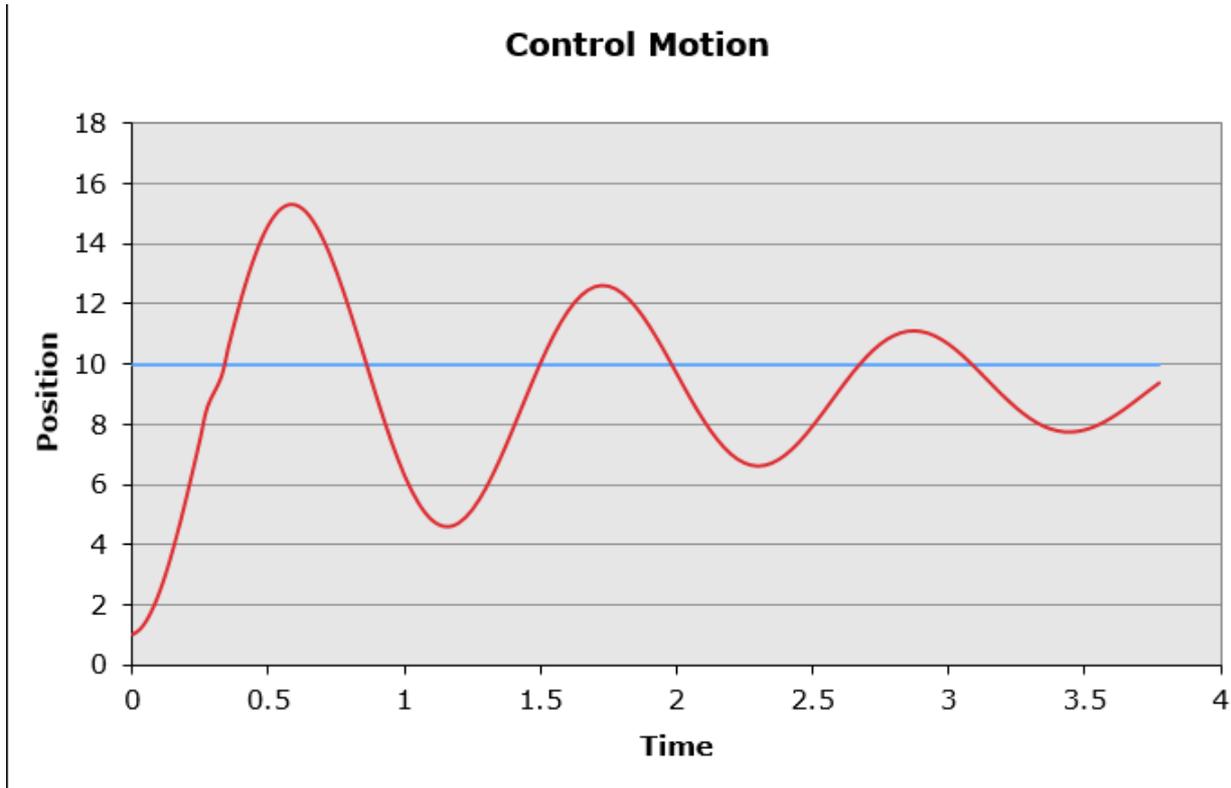


$$K_p = 6$$

$$K_d = 0$$

$$K_i = 0$$

# Controls Review



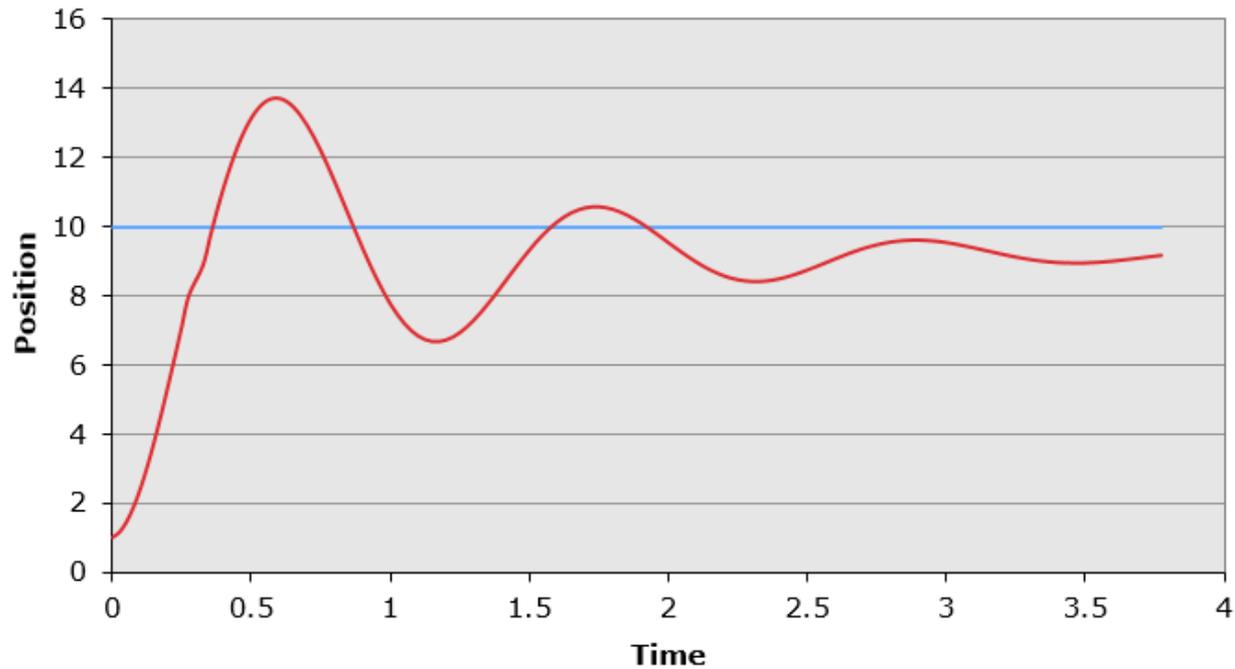
$$K_p = 3$$

$$K_d = 0.1$$

$$K_i = 0$$

# Controls Review

**Control Motion**

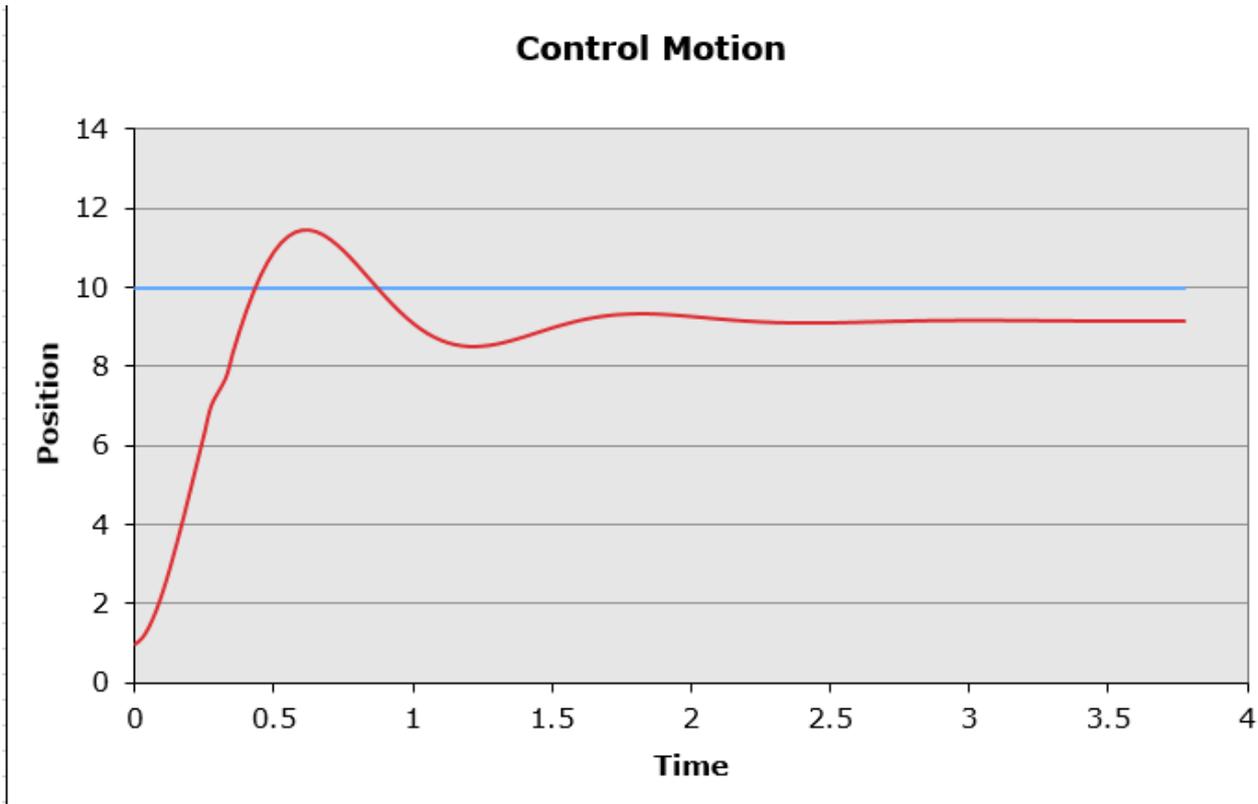


$$K_p = 3$$

$$K_d = 0.2$$

$$K_i = 0$$

# Controls Review



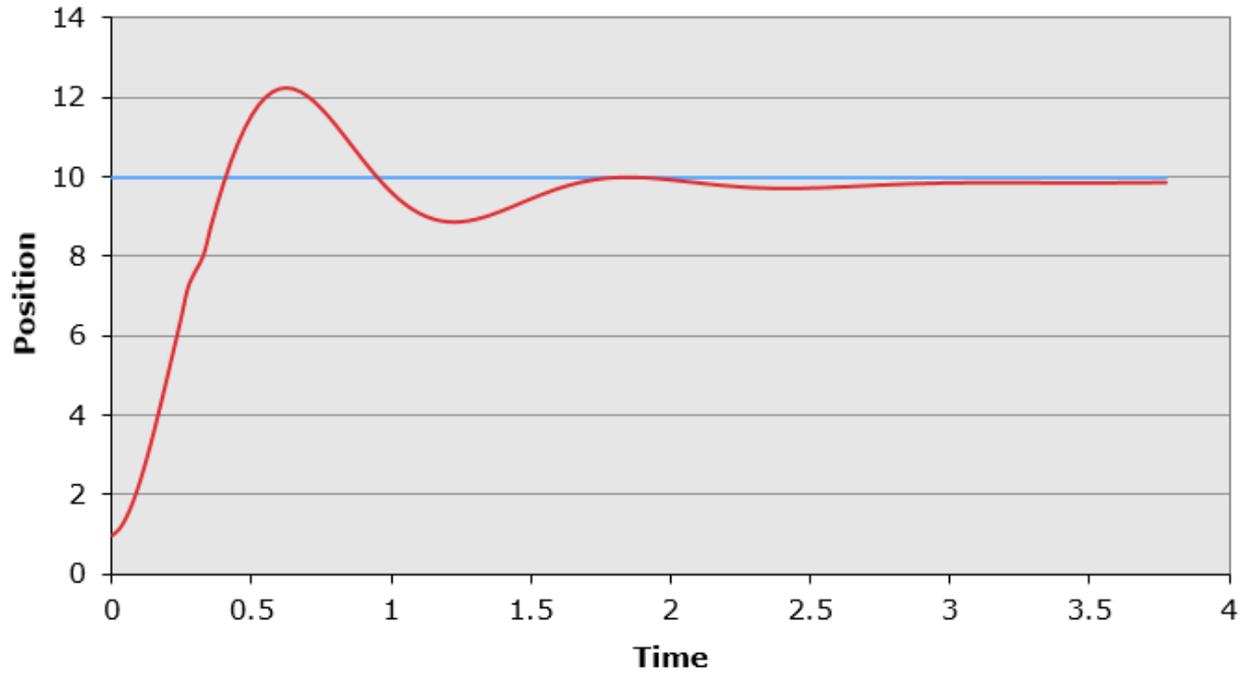
$$K_p = 3$$

$$K_d = 0.4$$

$$K_i = 0$$

# Controls Review

**Control Motion**



$$K_p = 3$$

$$K_d = 0.4$$

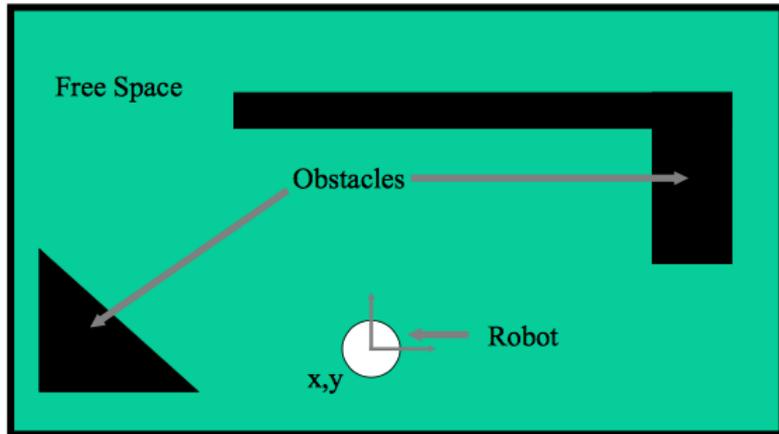
$$K_i = 1$$

# Path Planning

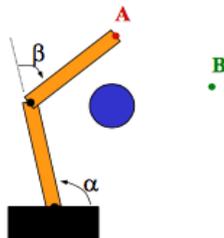
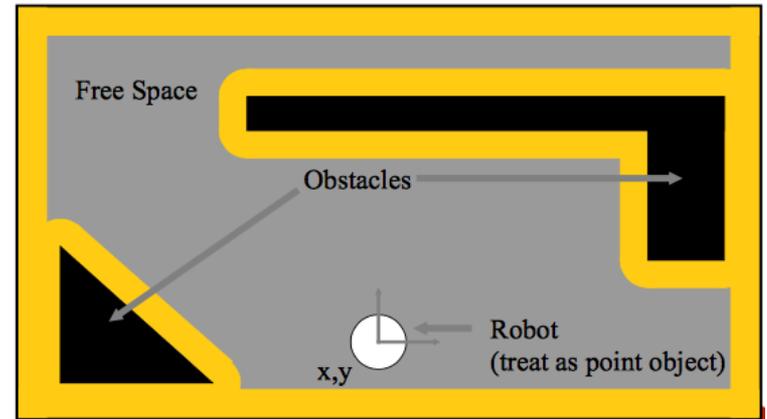


Free Space = Work space-Obstacles

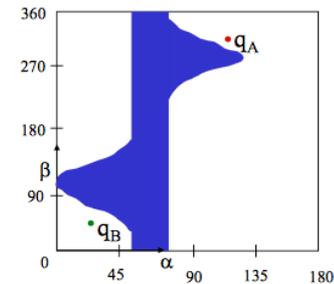
Configuration Space = Balloon world



Reference configuration



An obstacle in the robot's workspace



The C-space representation of this obstacle...

# Motion Planning



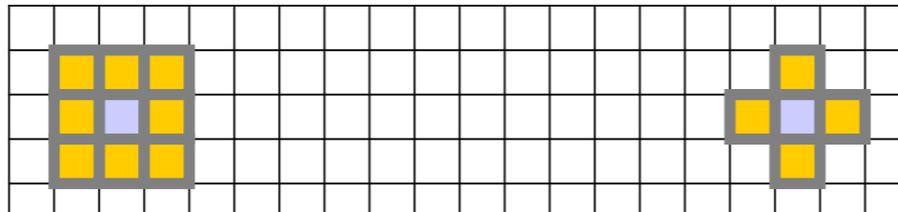
L1 is Manhattan Distance (Taxi Cab)

L2 is Euclidean Distance (Crow)



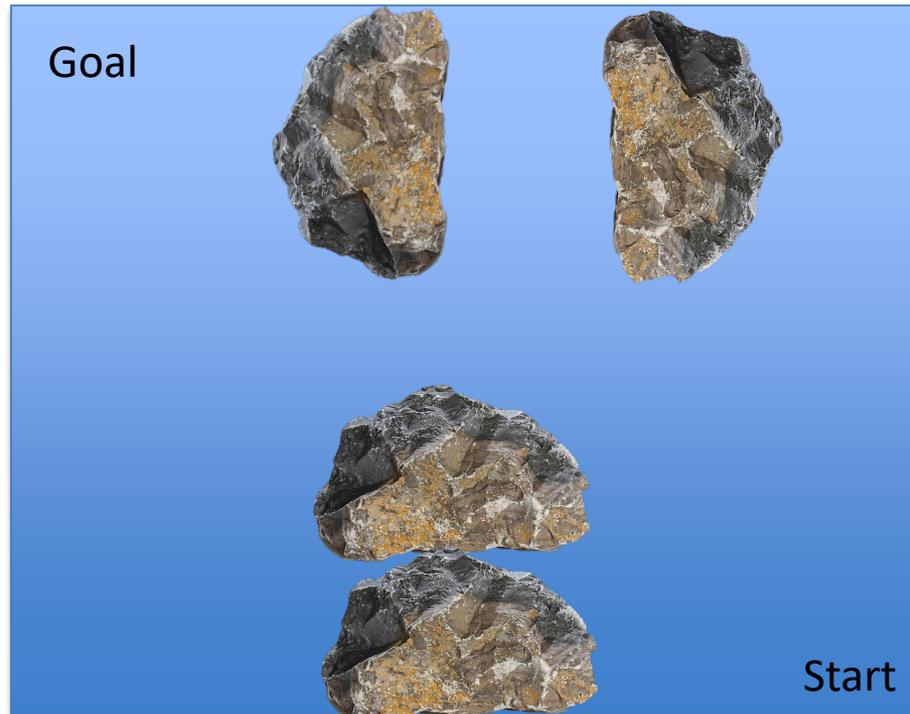
If our world is a grid, L1 is 4 point connectivity, L2 is 8

8-Point Connectivity • 4-Point Connectivity  
– (approximation of the L1 metric)



# Wavefront Example

We have an omnidirectional robot. We are looking to find the shortest path with respect to the L2 metric. Draw wavefronts on the image:



# Wavefront Planner

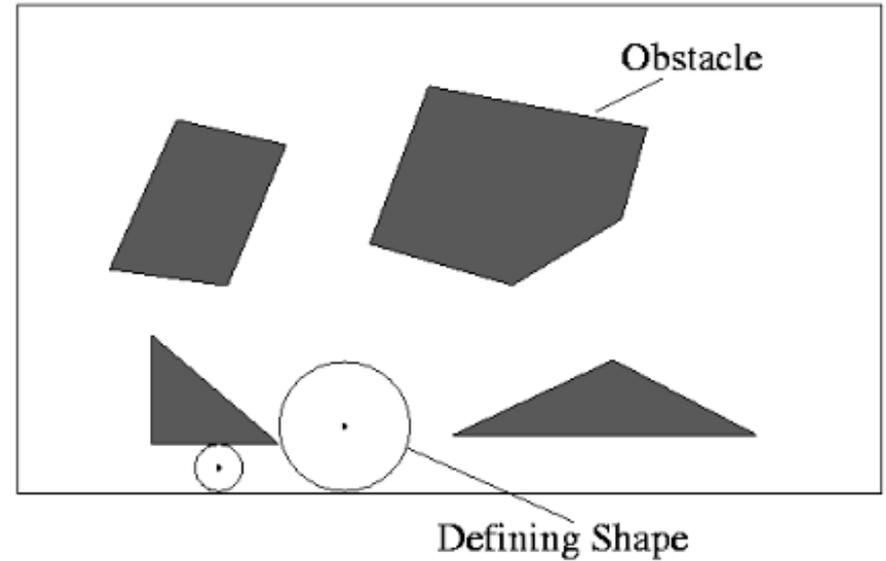
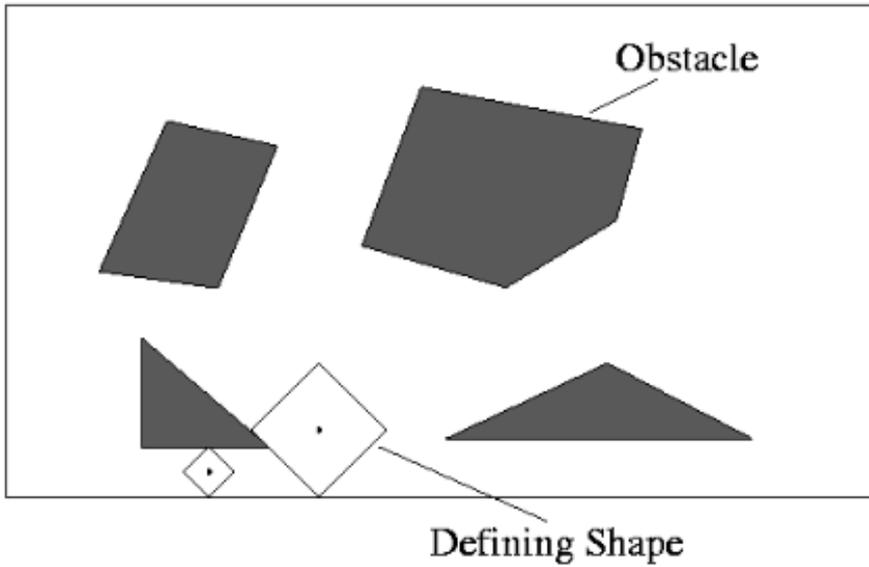


- Start from the goal



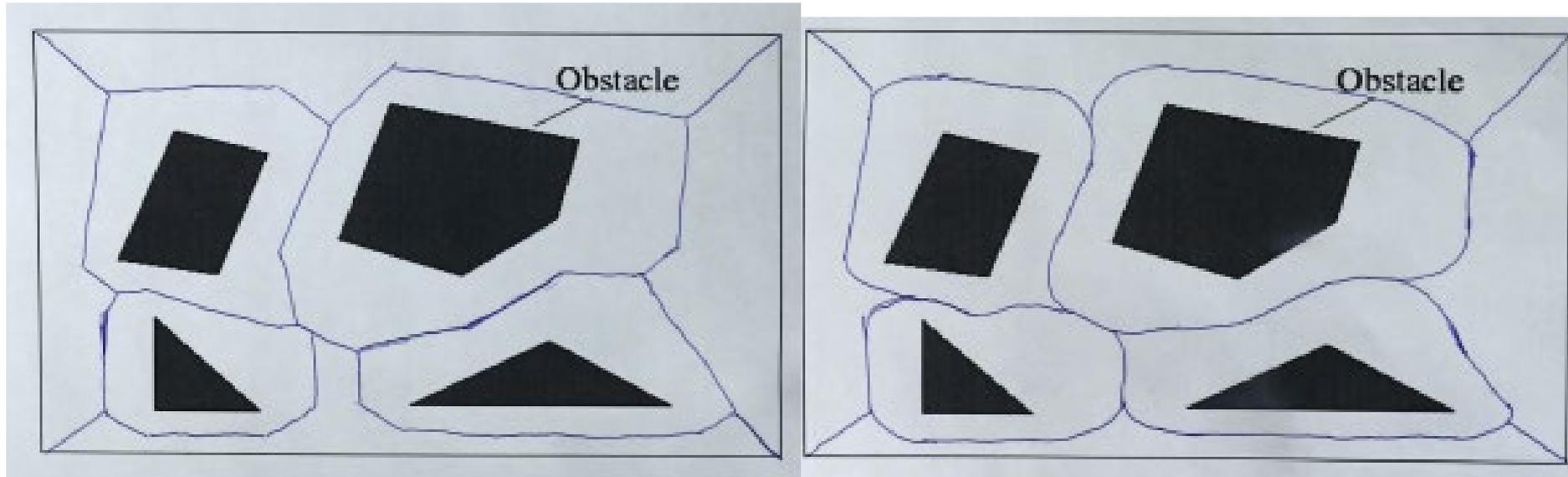
- When you are about to wrap around an obstacle, imagine a robot on the end of the previous wavefront

# Voronoi Example



For fun, also look up Delaunay Triangulation

# Voronoi Example



For fun, also look up Delaunay Triangulation

# Graph Search



Complete vs. not complete "There is no solution to this graph"

- if there are no **unexpanded** frontier nodes **then return** failure
- choose an **unexpanded** frontier node for expansion using **strategy**, and add it to the **expanded set**



Informed vs. uninformed



Uninformed



Informed

# Graph Search

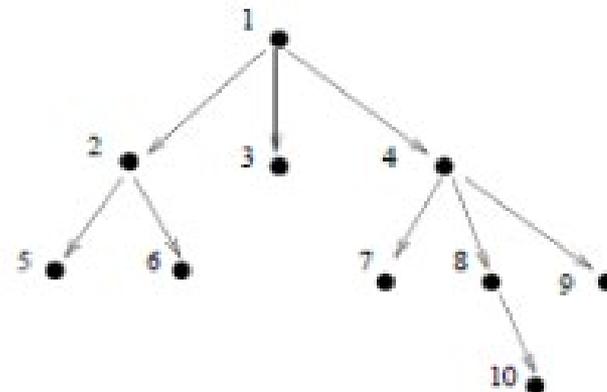
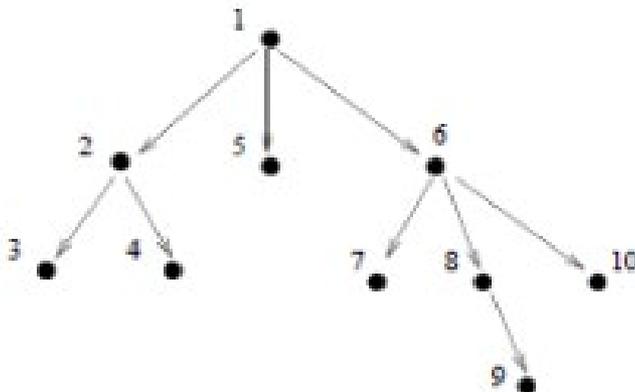


DFS: Waterslides



BFS: Spilled coffee

<http://www.how2examples.com/artificial-intelligence/tree-search>



# Graph Search



Greedy chooses lowest locally



A\* uses heuristics, which are approximations of future cost

# Heuristics



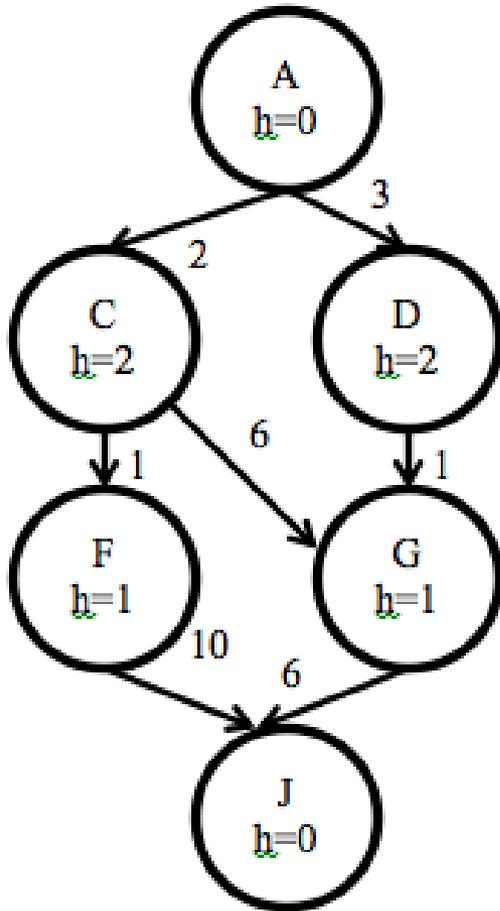
A\* graph search uses consistent heuristics (consistent is a stronger flavor of admissible) which you can think of as an optimistic lookout



“Yeah, our goal is definitely like 5 minutes away!!!”



# Example



Closed list: A(0)  
Open list: C (2+2)  
D(3+2)

# Localization



Without it, you are stuck on Lab 3

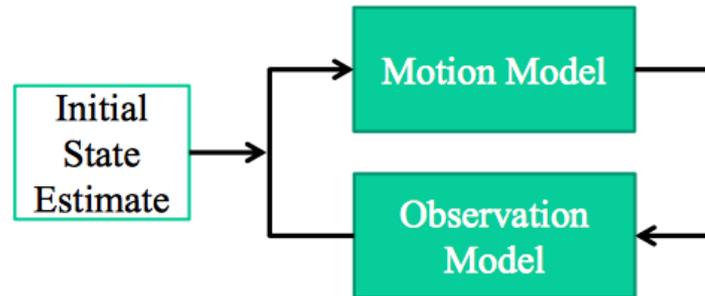


Everything adds noise



## Localization: Estimate State

- Move: Motion Model
- Observe: Observation Model

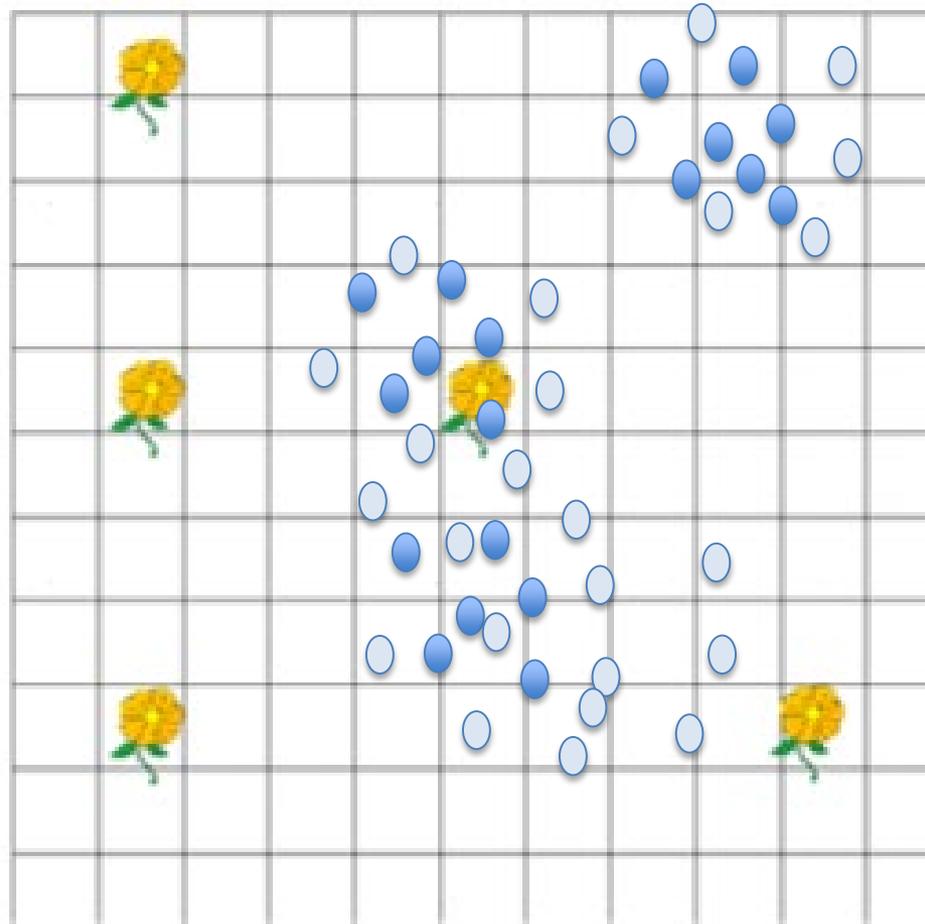


# 2D Example

Say we have some array of belief that we are building up.

How does our prediction change if we move 2 units left and 2 units up?

How does our prediction change if we detect a flower 1 unit left?



- Area of medium-high probability
- Area of medium-low probability

# Image credits

- <https://www.good.is/articles/bird-s-eye-view-the-most-comprehensive-aerial-shot-of-new-york-city>