Midterm Review 26 Feb 2020

Hannah Lyness

Logistics

- Midterm Time: Monday 10:30AM
- Midterm Location: Normal Classroom
- You may use 2 sides of 1 8.5x11” paper written in your own hand
- Final value: 14% of final grade

Logistics (cont.)

- Test Coverage: Vision, Neural Nets, Controls, Learning for Controls, Reinforcement Learning, System engineering, Motion Planning, Graph Search, Localization, (blue indicates guest lecture)
- Test Non-Coverage: Syntax, style, proofs

Vision

Vision

We can store images as arrays of values

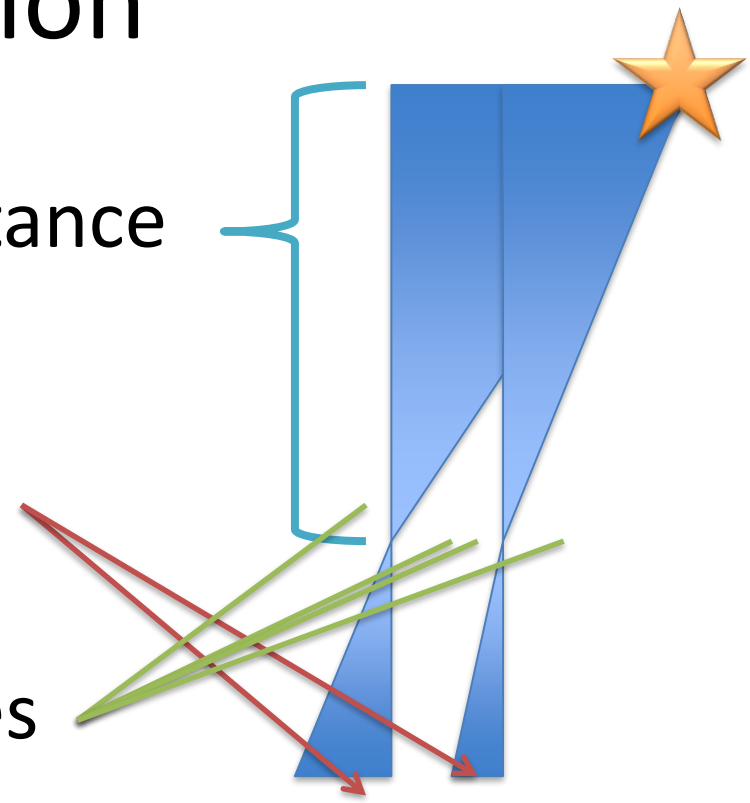
It is difficult for a computer program to identify objects (Where’s Waldo)

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.

Convolution

Convolution is like repeated elementwise multiplication and summation

$$g(x,y) = \omega * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s,t) f(x-s, y-t),$$

Choosing the filter or kernel determines what the convolution does to the image

Convolution

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

-1	-1	-1
2	2	2
-1	-1	-1

https://aishack.in/tutorials/image-convolution-examples/

Convolution

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

-1	-1	-1
2	2	2
-1	-1	-1

https://aishack.in/tutorials/image-convolution-examples/

Convolution

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

-1	-1	-1
2	2	2
-1	-1	-1

https://aishack.in/tutorials/image-convolution-examples/

Neural Nets

Neural Nets

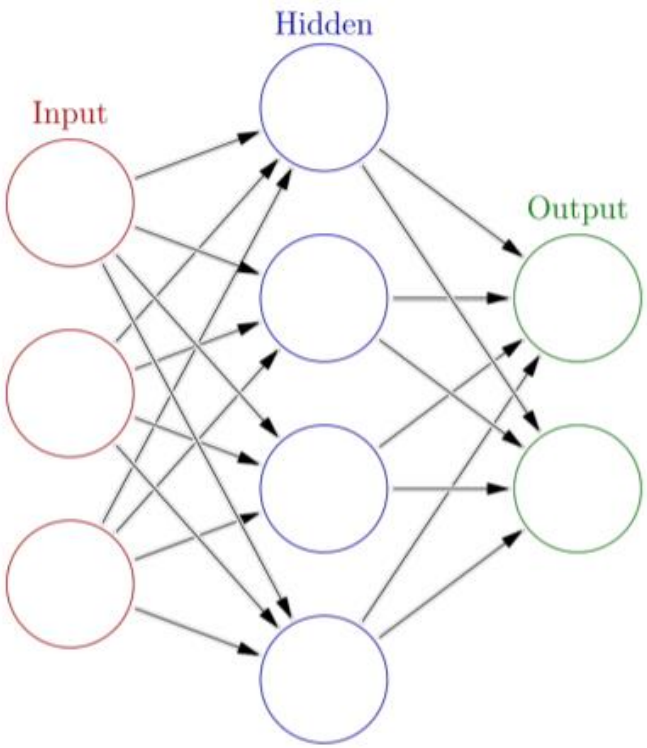


Image from wikipedia: https://commons.wikimedia.org/wiki/File:Colored_neural_network.svg

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.

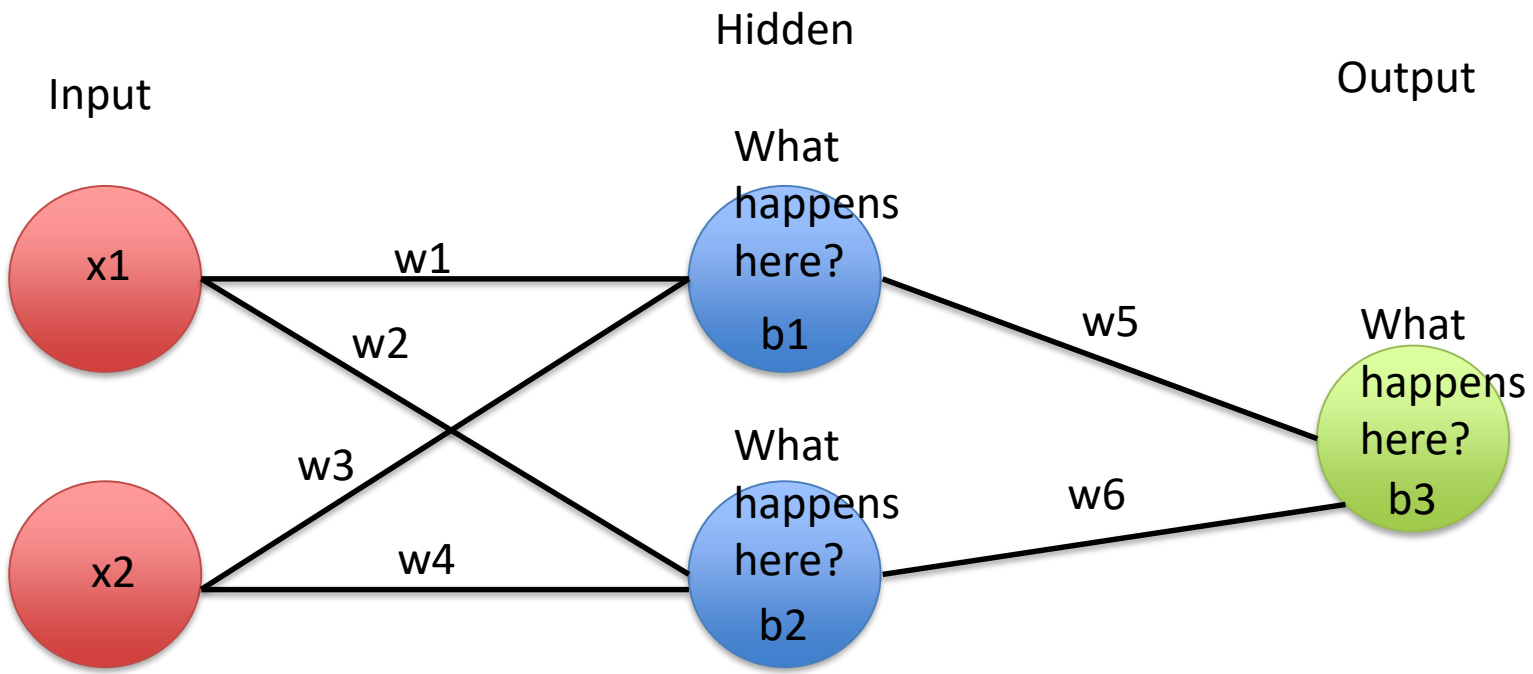
The final layer is considered the output layer.

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.

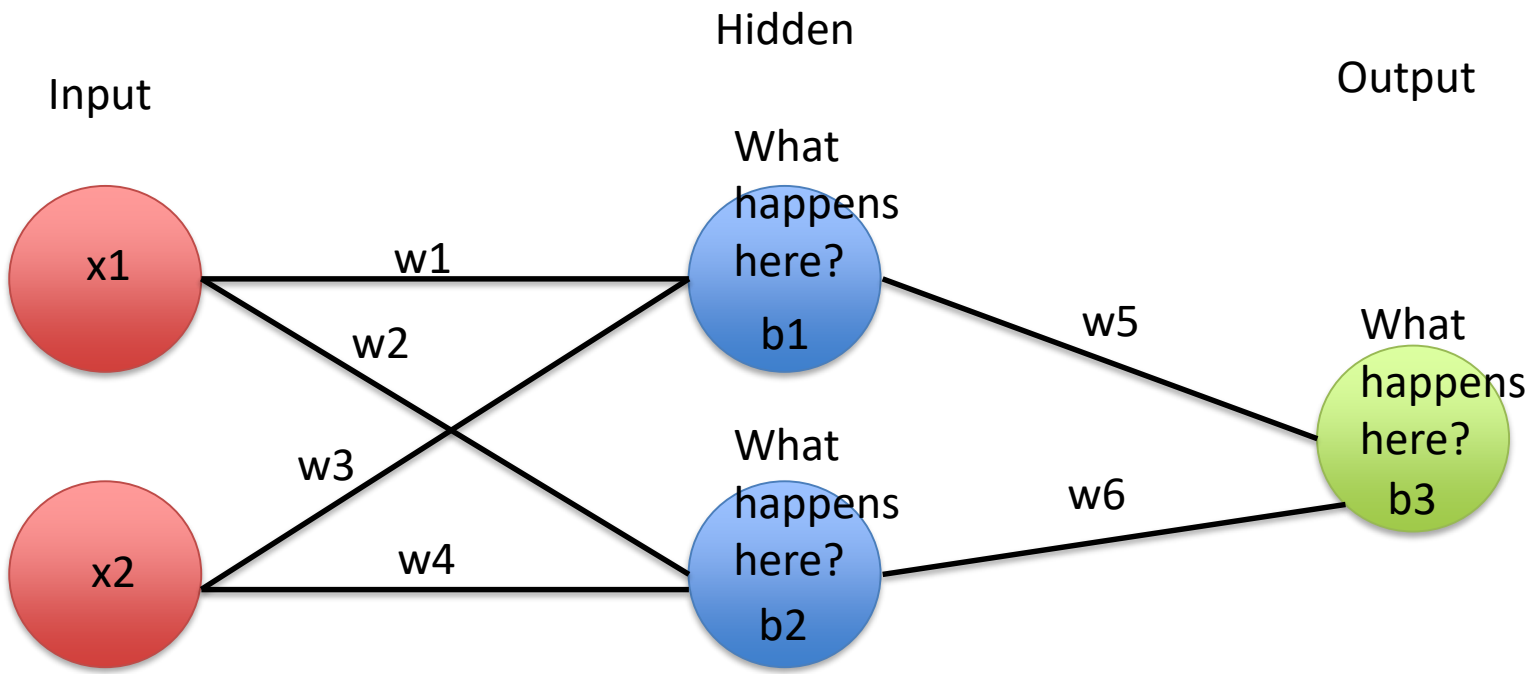
Neural Nets

A neural net is just a series of weights that we can tune



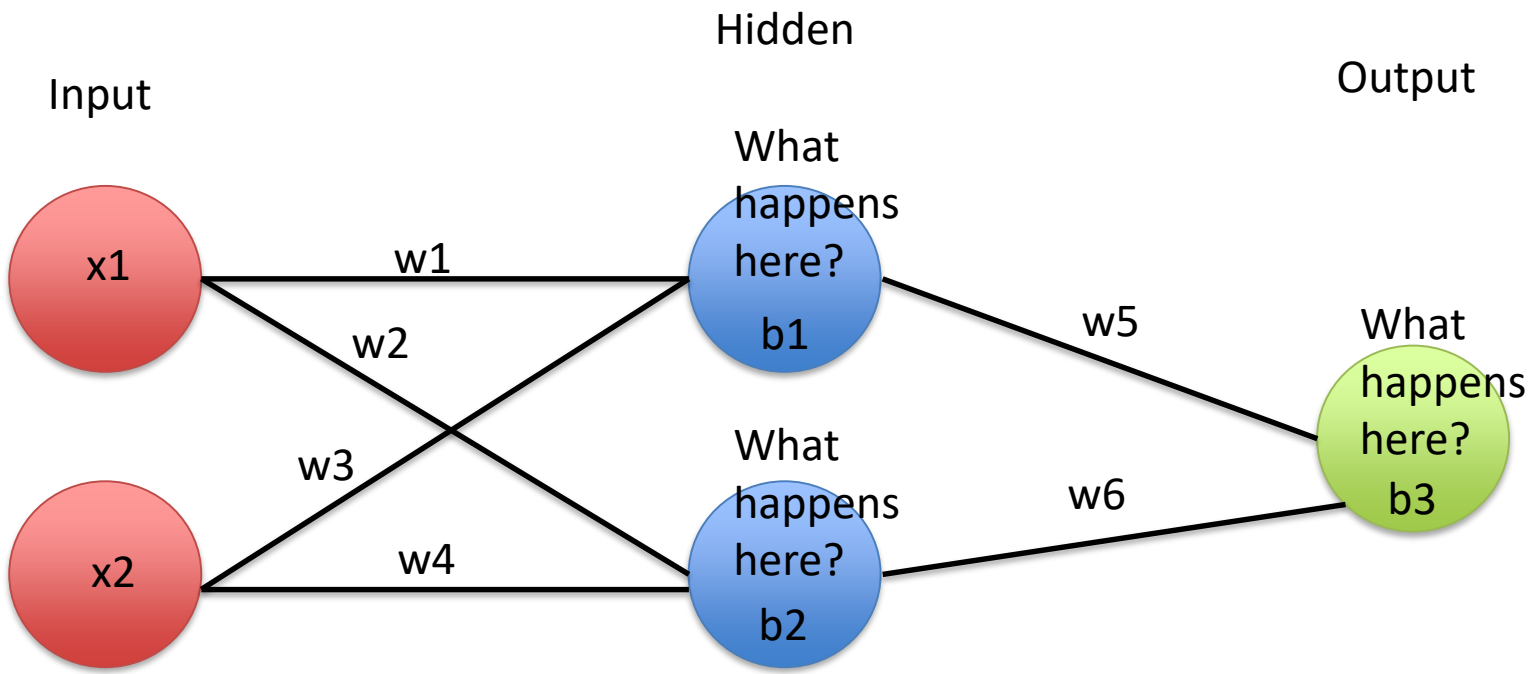
Neural Nets

We start with random weights and calculate the difference between what we got and what we want.



Neural Nets

We use backpropagation to calculate the gradient of each weight and then change each weight based on the average of the negative gradients.



Neural Nets

There are many prebuilt neural net tools.

We used a neural net to classify hedgehogs and porcupines

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

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

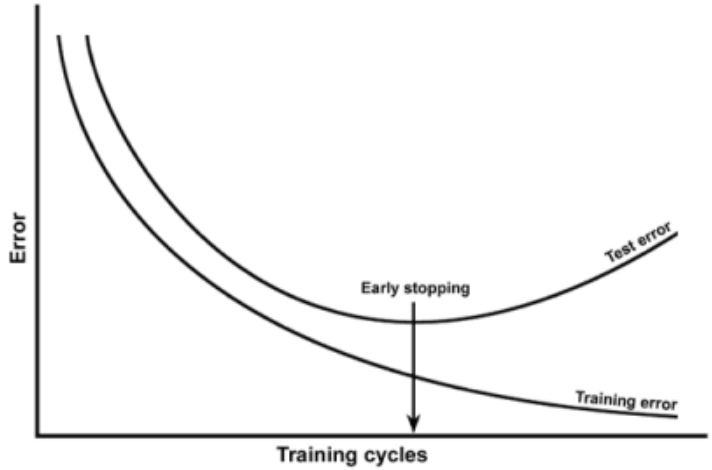
Neural Nets - Caveats

Black box

Requires a lot of data, time

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

Can overfit



Other Machine Learning Techniques

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

Controls

PID pros

PID cons

Controls

PID is great

Kp is like spring constant

Kd is like damping coefficient

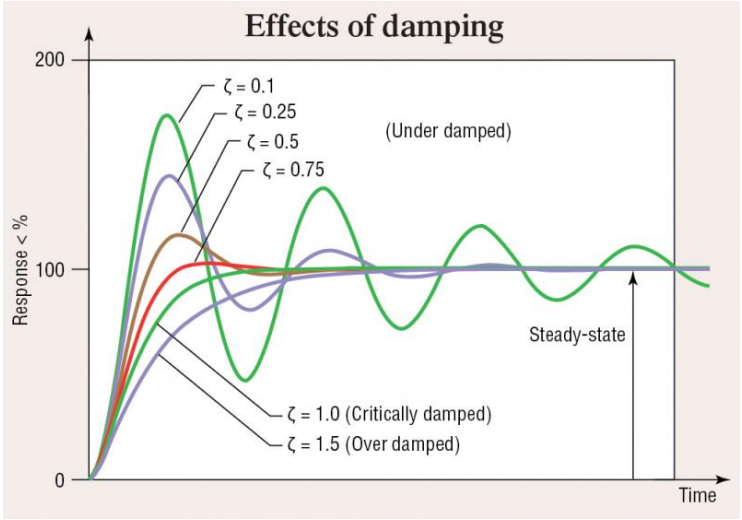
Ki is like a friend that gently pulls you towards the correct path when you are converging on a different one

Controls Examples

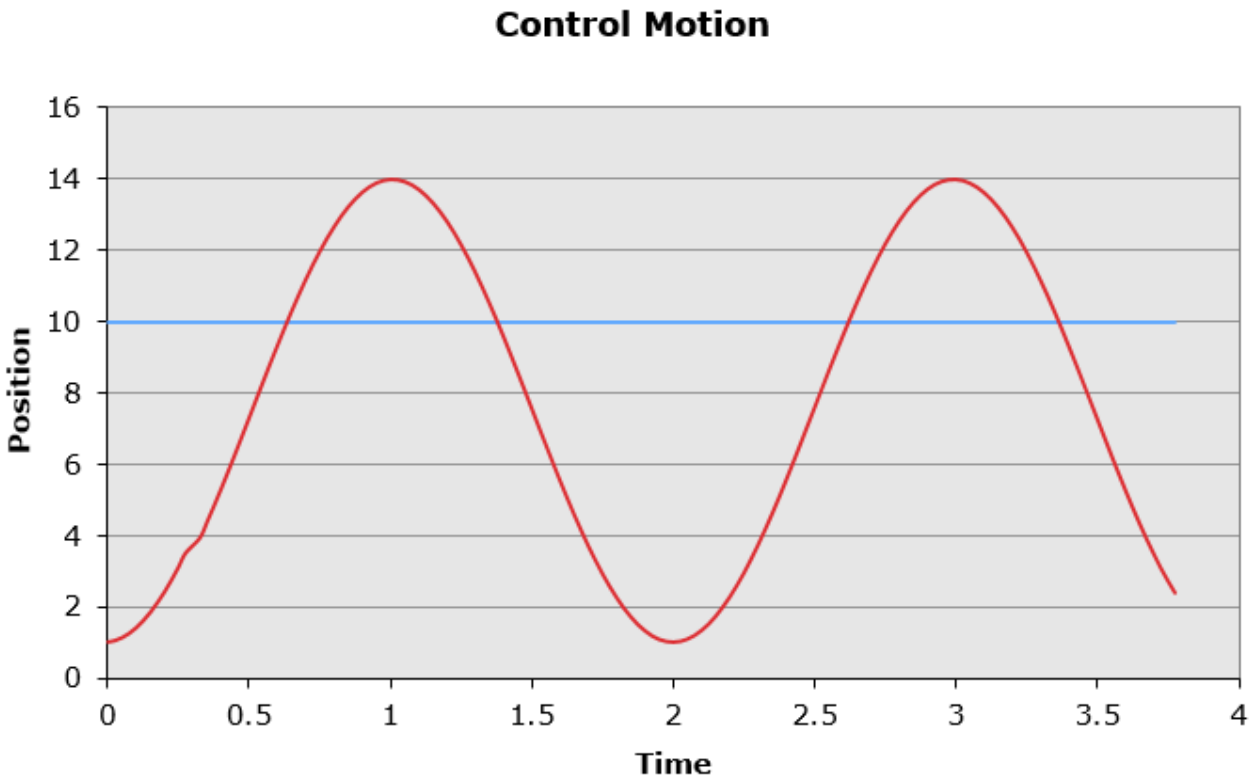
https://youtu.be/lZPtFDXYQRU?t=1m

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

https://www.cs.cmu.edu/~16311/current/schedule/pp/pid.xls



Controls Review

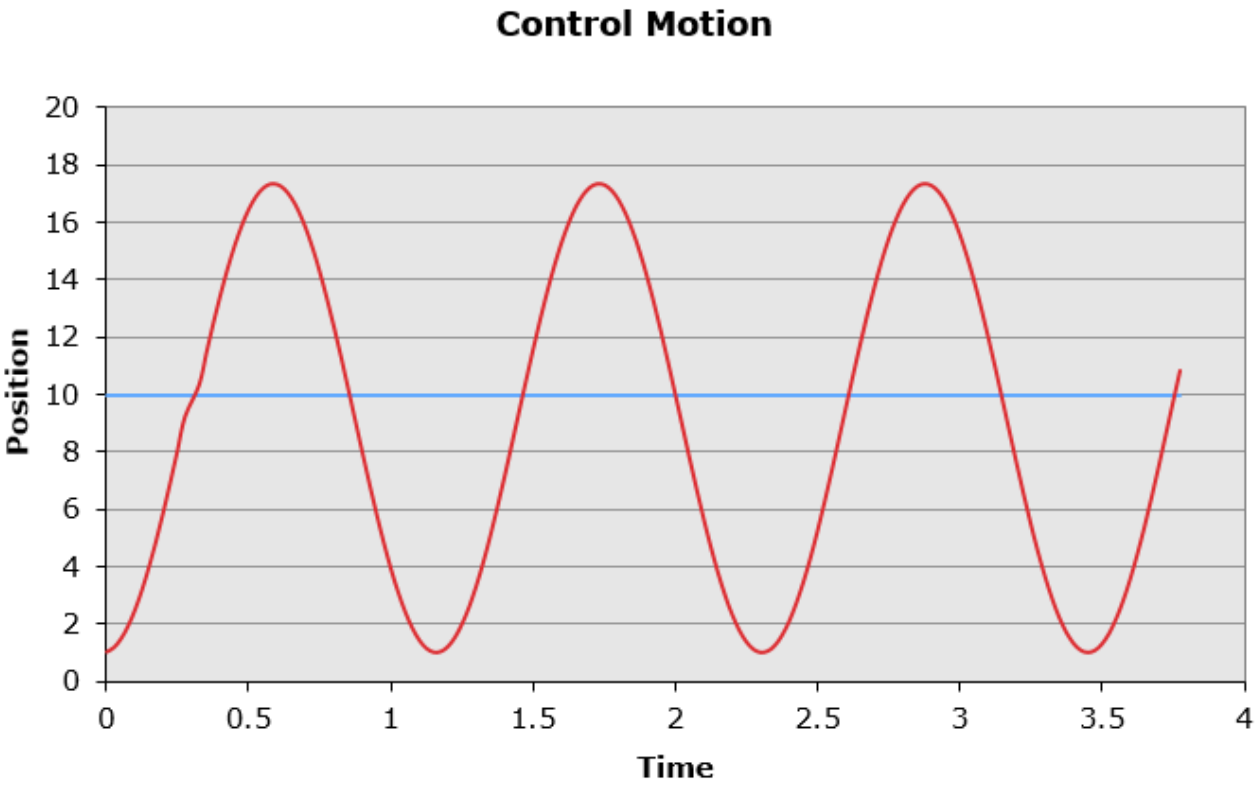


Kp = 1

Kd = 0

Ki = 0

Controls Review

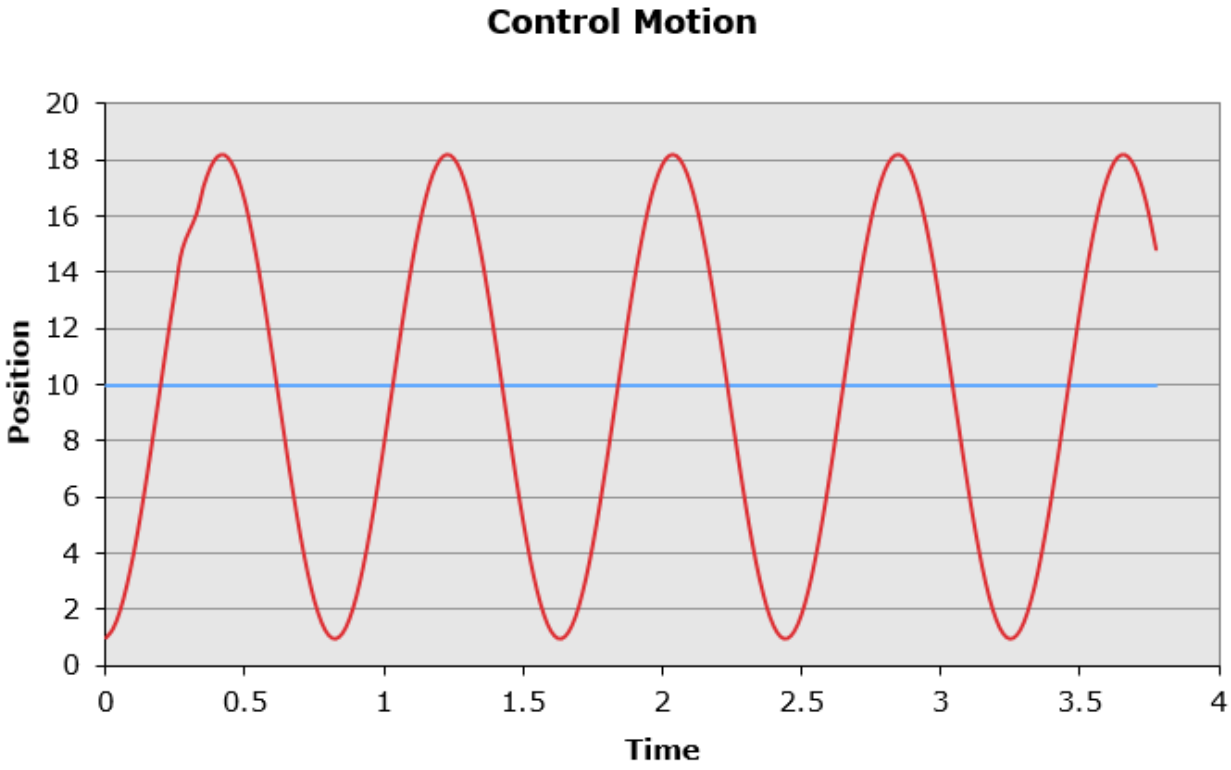


Kp = 3

Kd = 0

Ki = 0

Controls Review

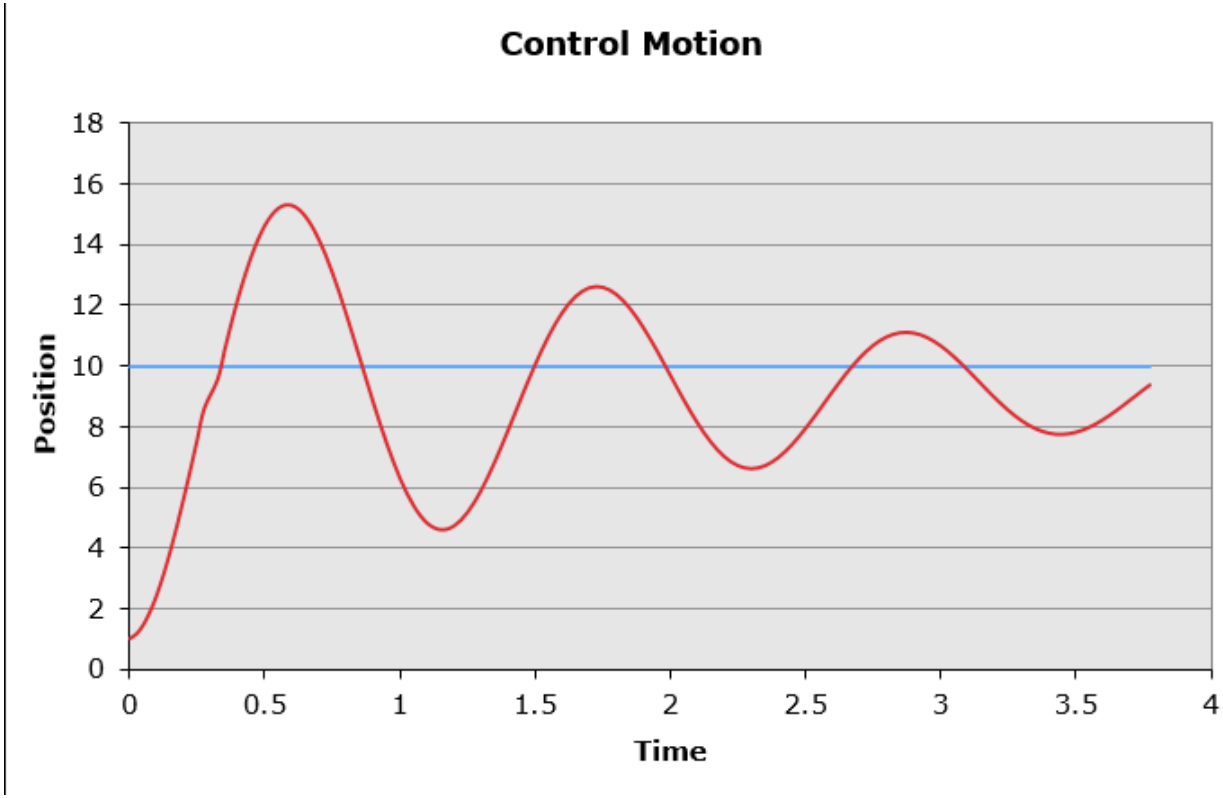


Kp = 6

Kd = 0

Ki = 0

Controls Review

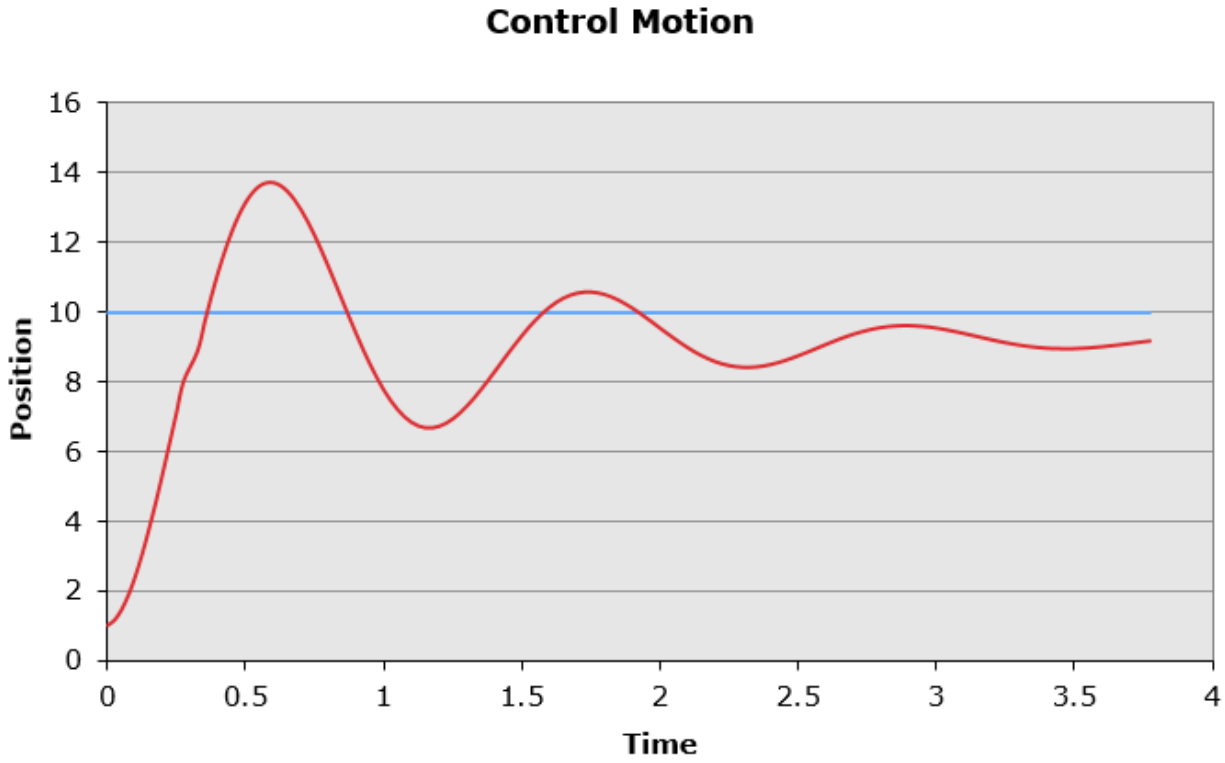


Kp = 3

Kd = 0.1

Ki = 0

Controls Review

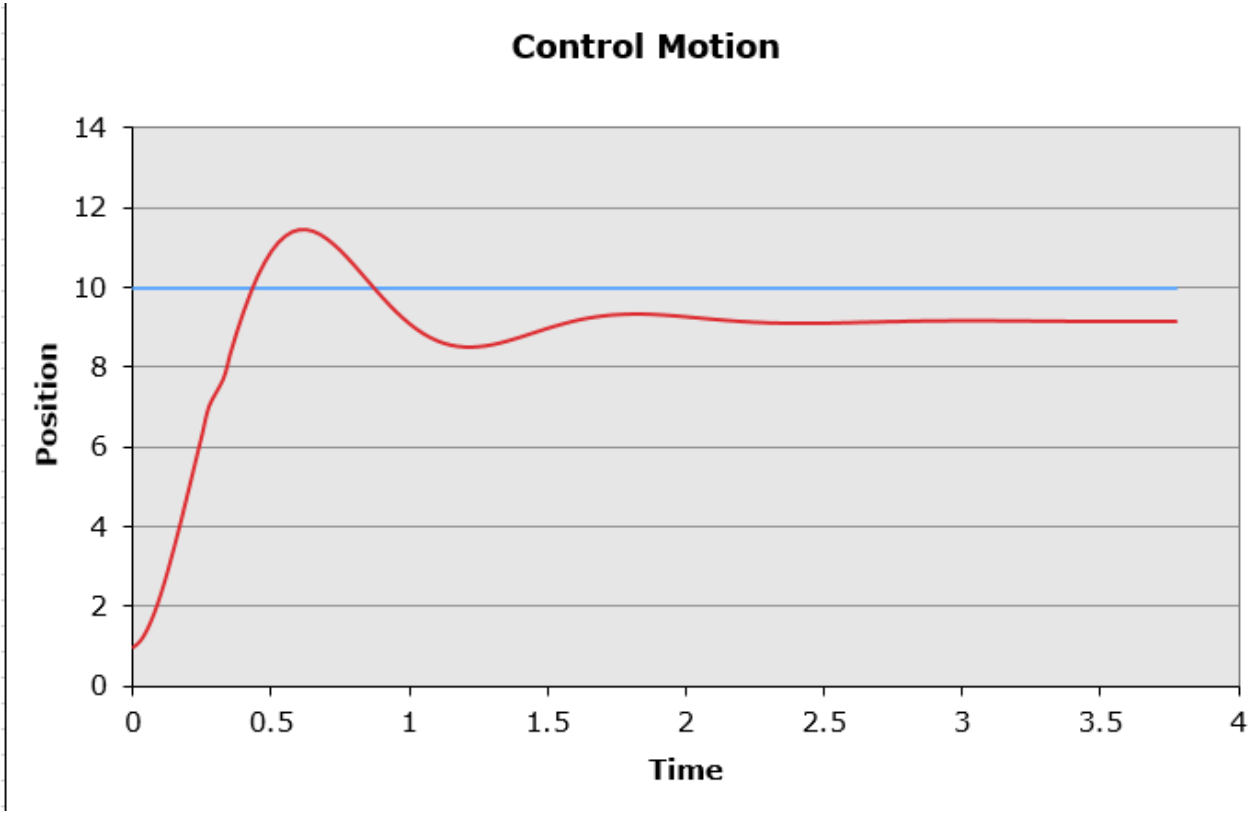


Kp = 3

Kd = 0.2

Ki = 0

Controls Review

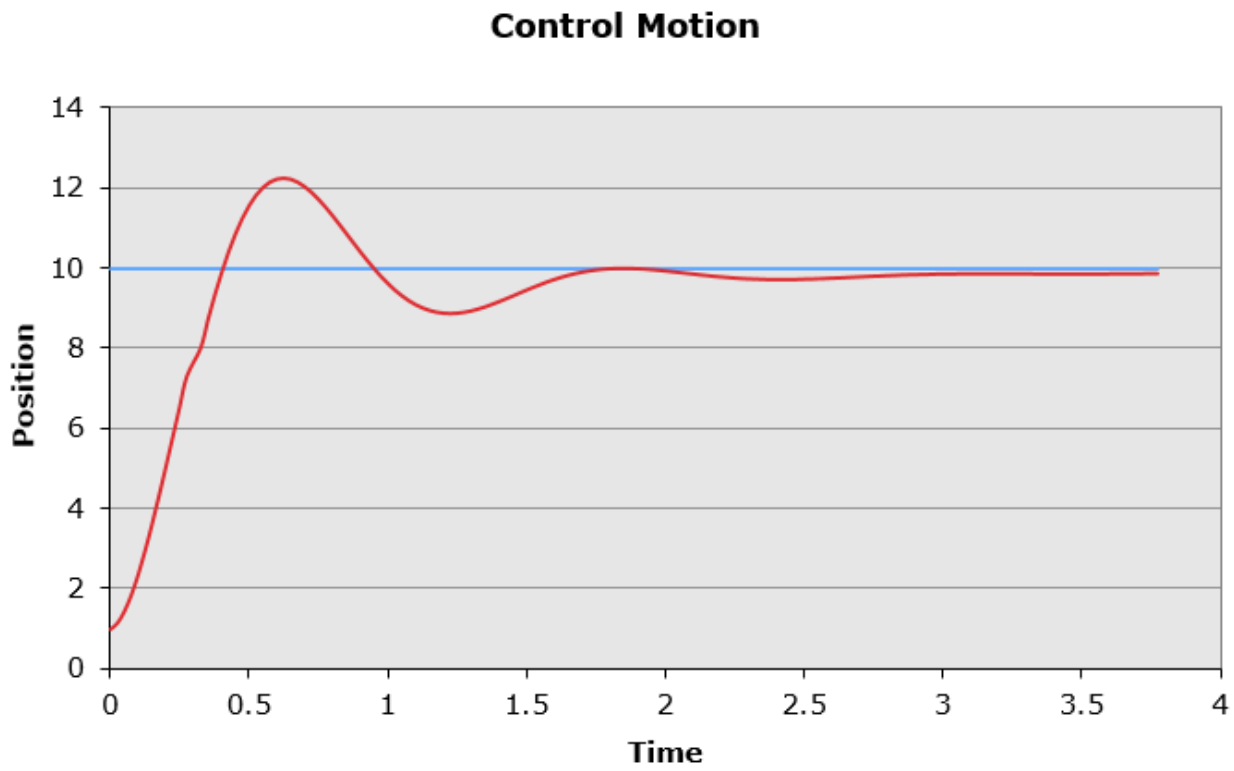


Kp = 3

Kd = 0.4

Ki = 0

Controls Review



Kp = 3

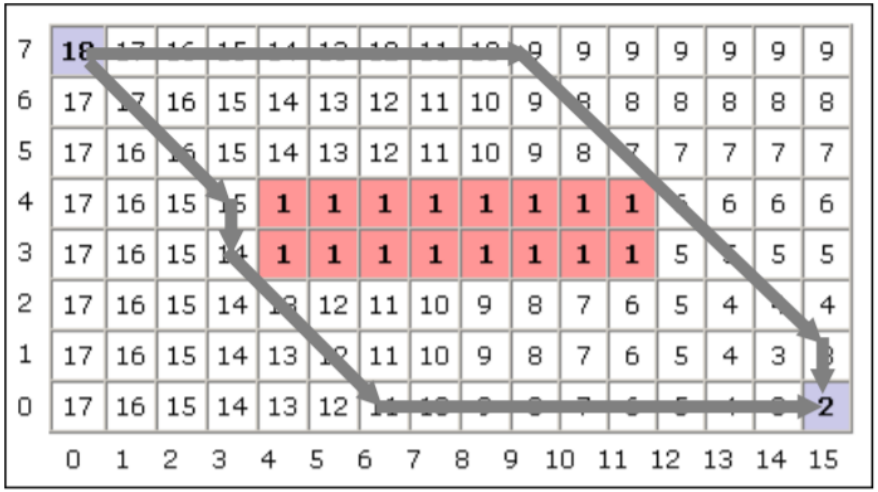
Kd = 0.4

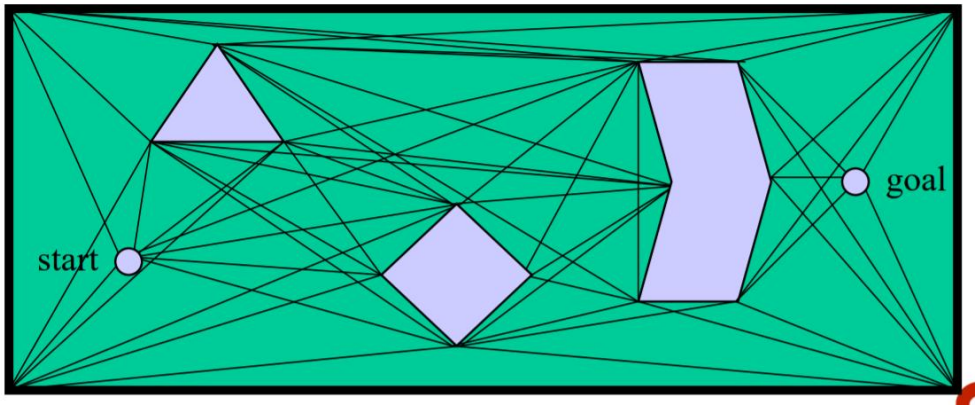
Ki = 1

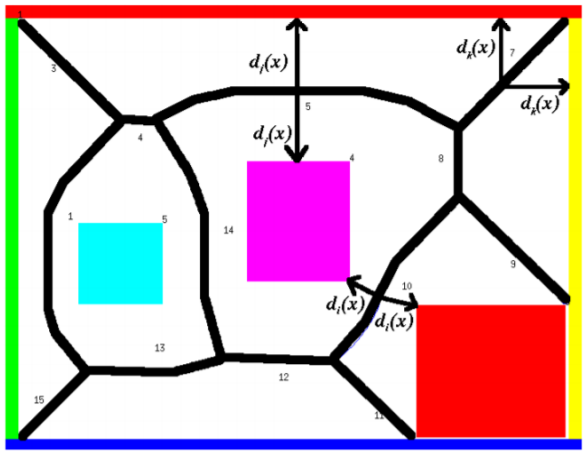
Path Planning

Path Planning

We can represent the world as a grid or with waypoints (roadmap)



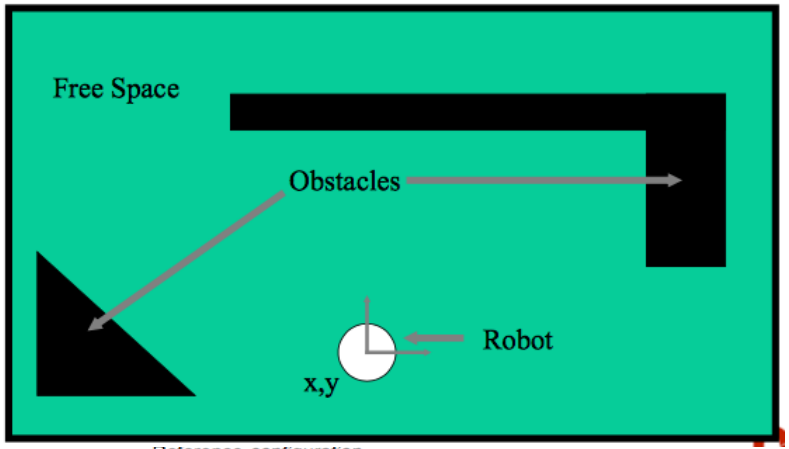




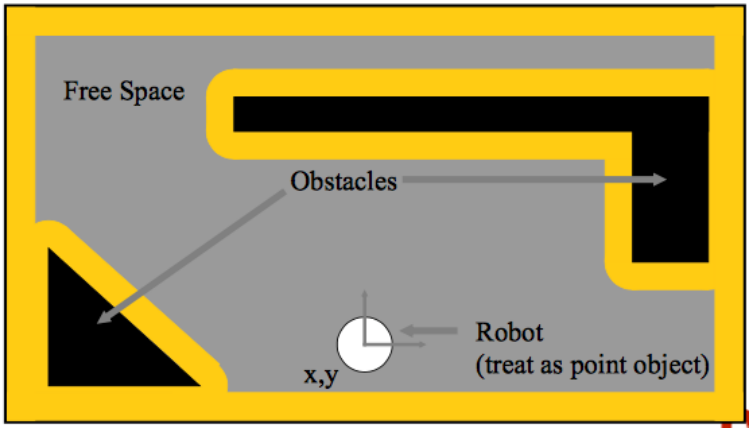
Path Planning

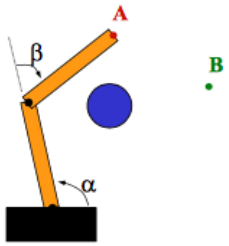
Free Space = Work space-Obstacles

Configuration Space = Balloon world

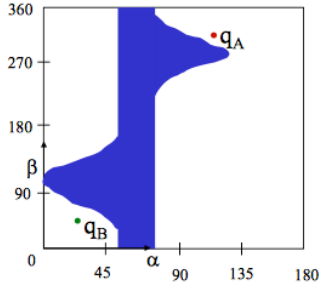


Reference *configuration*





An obstacle in the robot's workspace



The C-space representation of this obstacle…

Motion Planning

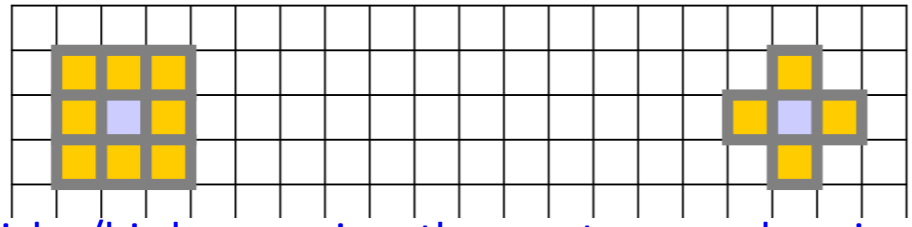
L1 is Manhattan Distance (Taxi Cab)

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

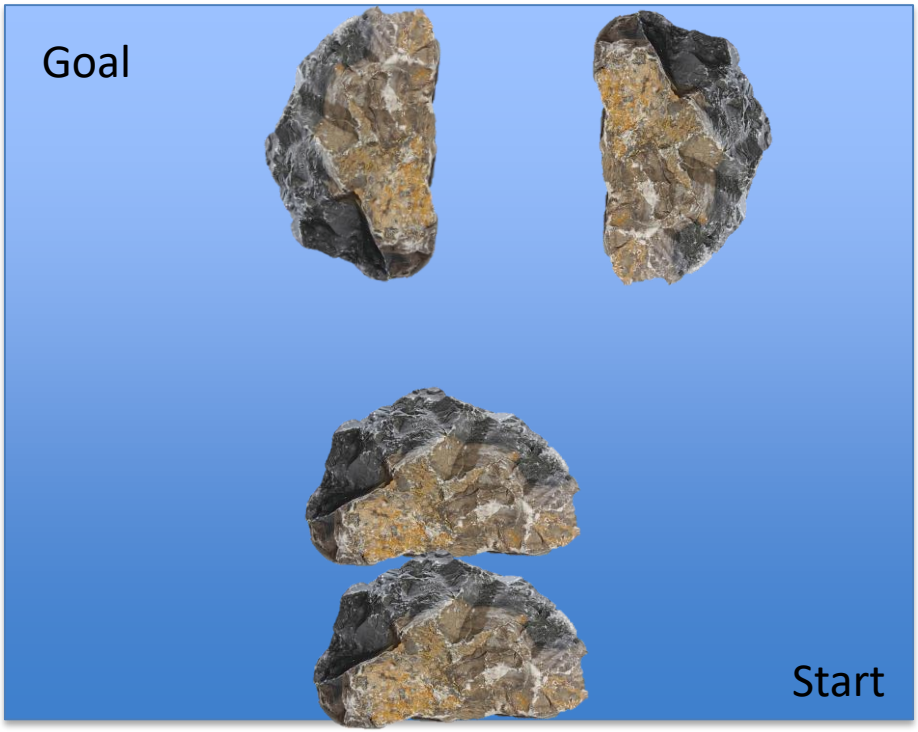
– *(approximation of the L1 metric)*



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

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

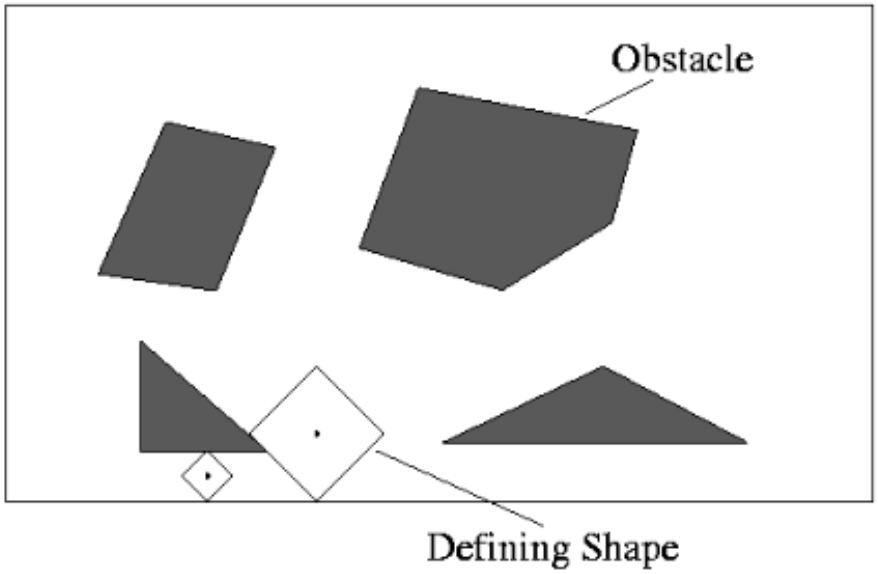
- When you are about to wrap around an obstacle, imagine a robot on the end of the previous wavefront

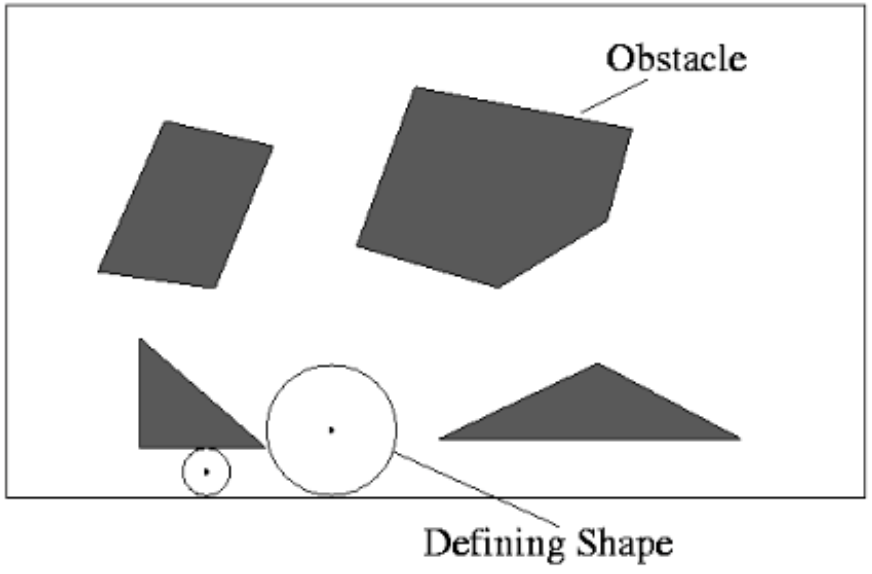
Roadmap Approaches

- Veronoi gives you max distance from obstacles https://www.youtube.com/watch?v=YBq_fzAOpVI

- Visibility gives you shortest path

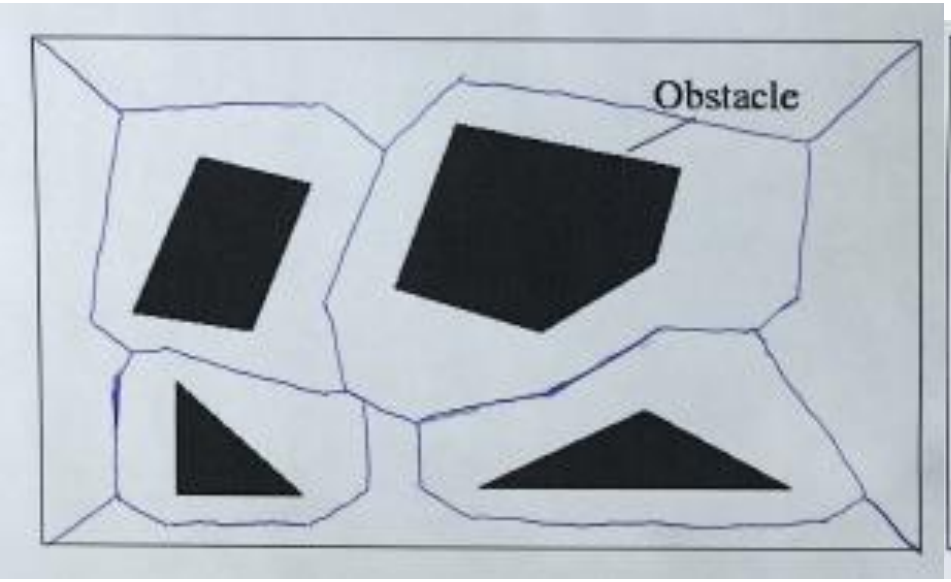
Voronoi Example

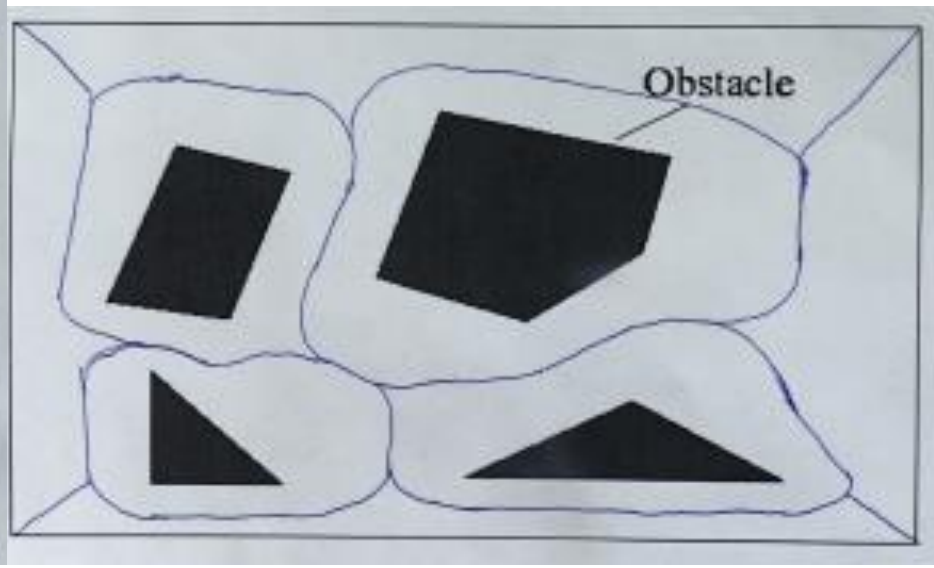




For fun, also look up Delaunay Triangulation

Voronoi Example





For fun, also look up Delaunay Triangularion

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

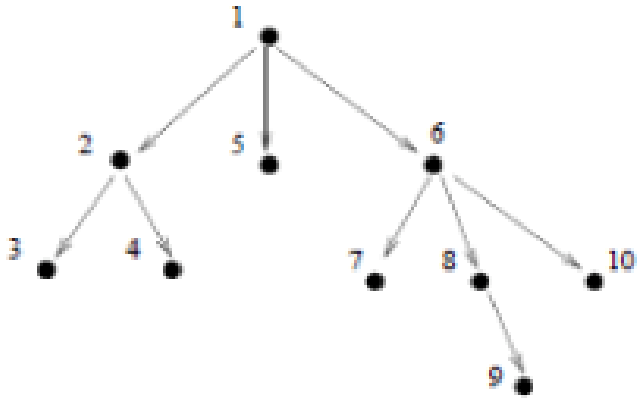
Images from https://www.cs.cmu.edu/afs/cs/Web/People/15780/slides/780s17-2.pdf

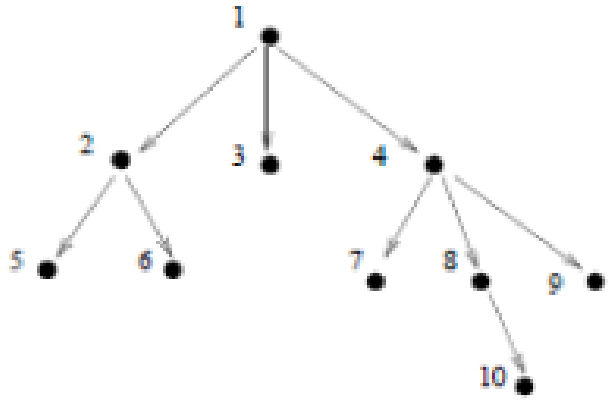
Graph Search

DFS: Waterslides

BFS: Spilled coffee

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





Graph Search

Greedy chooses lowest locally

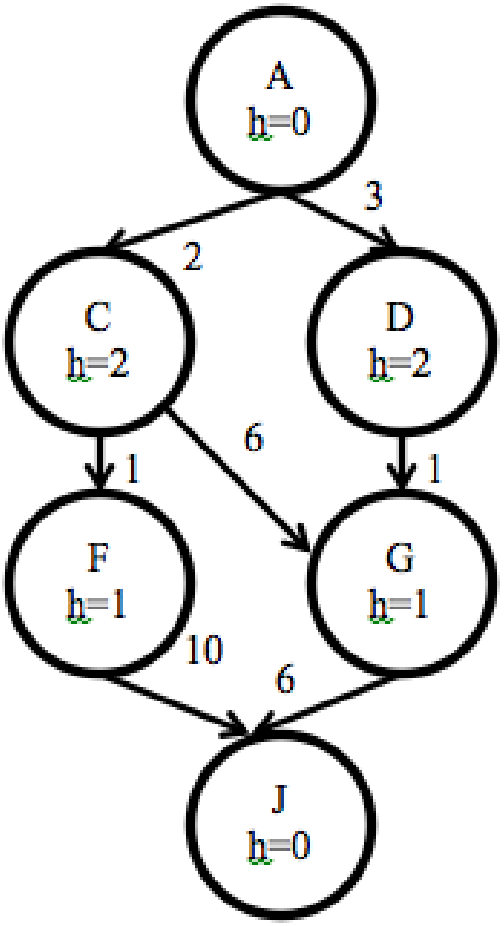
A* uses heuristics, which are approximations of future cost

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)

Image from https://www.cs.cmu.edu/afs/cs/Web/People/15780/slides/780s17-2.pdf

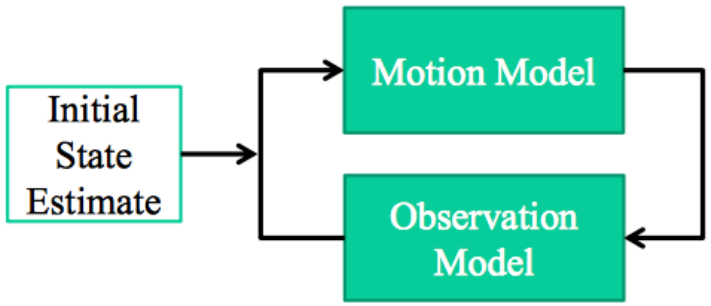
Localization

Without it, you are stuck on Lab 3

Everything adds noise

Localization: Estimate State

- Move: Motion Model
- Observe: Observation Model

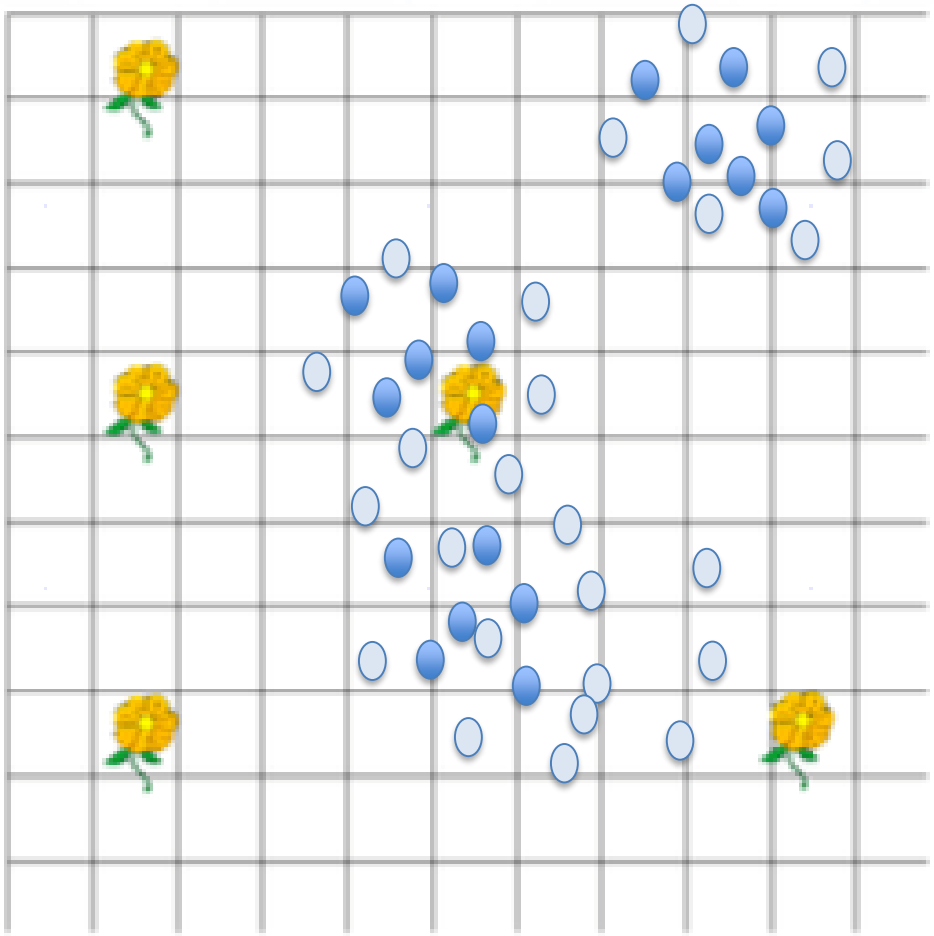


2D Example

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

Questions Posed By Students

List 3 functional and 3 non-functional components of a robot
Bug 1 is more breadth first and Bug 2 is more...
Draw the path that a robot would take using Bug 1, Draw the path that a robot would take using Bug 2