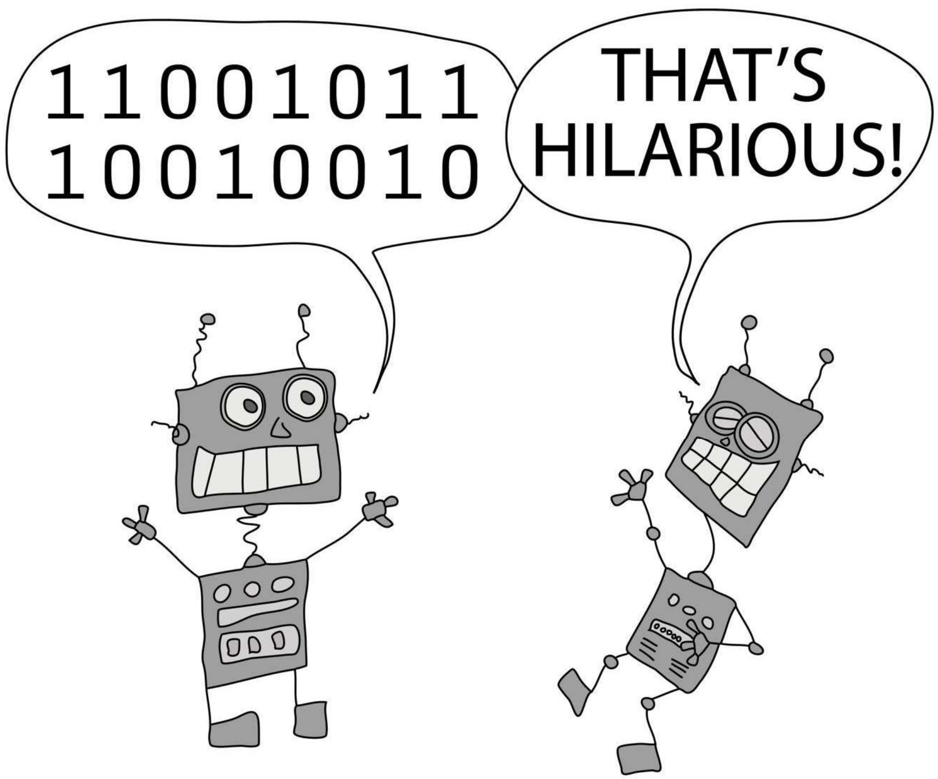
CS 249r: Special Topics in Edge Computing

Intro to Autonomous Systems / Robotics Wrap-Up

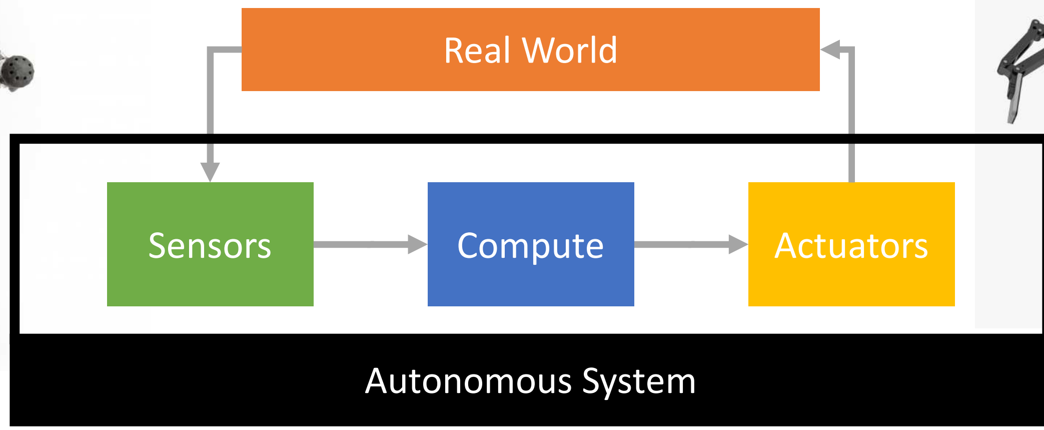


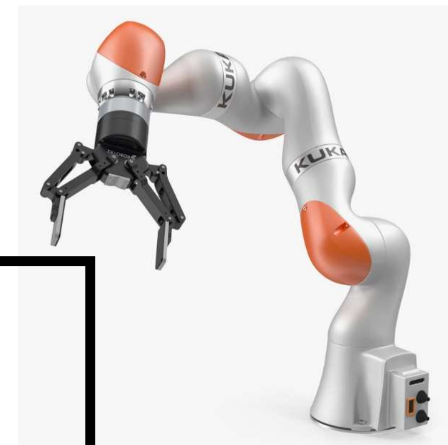
Brian Plancher
Fall 2019

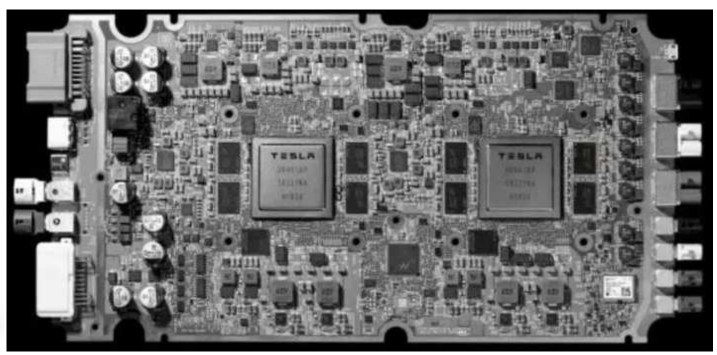
The goal for the next couple of lectures is to develop a high level understanding of:

1. What is an autonomous system
2. Key problems and constraints for autonomous systems
3. Some of the most important (classes of) algorithms in robotics
   A. The model based vs. model free tradeoff
   B. The online vs offline tradeoff
   C. The no free lunch theorem and the need for approximations
4. How computer systems / architecture design has and can play a role in improving autonomous systems

What do we mean by an Autonomous System?



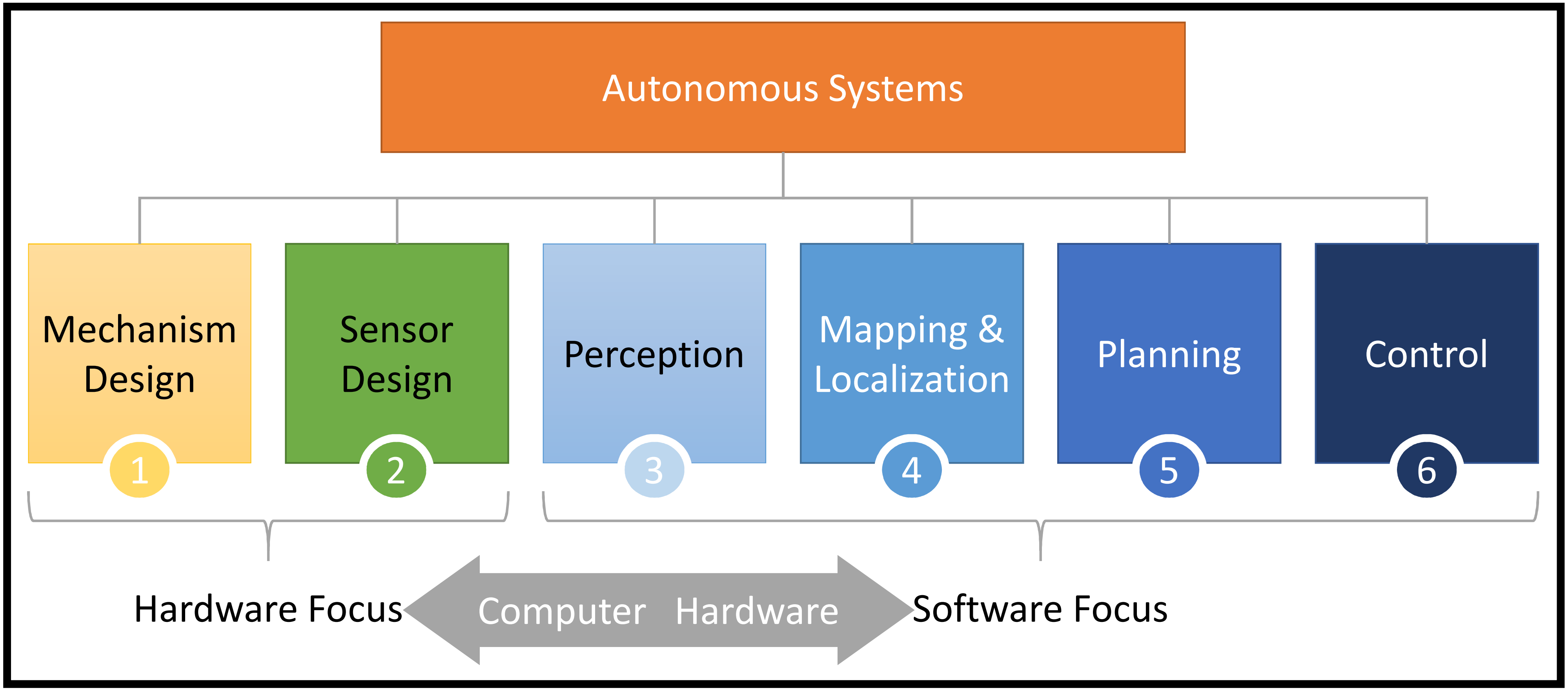




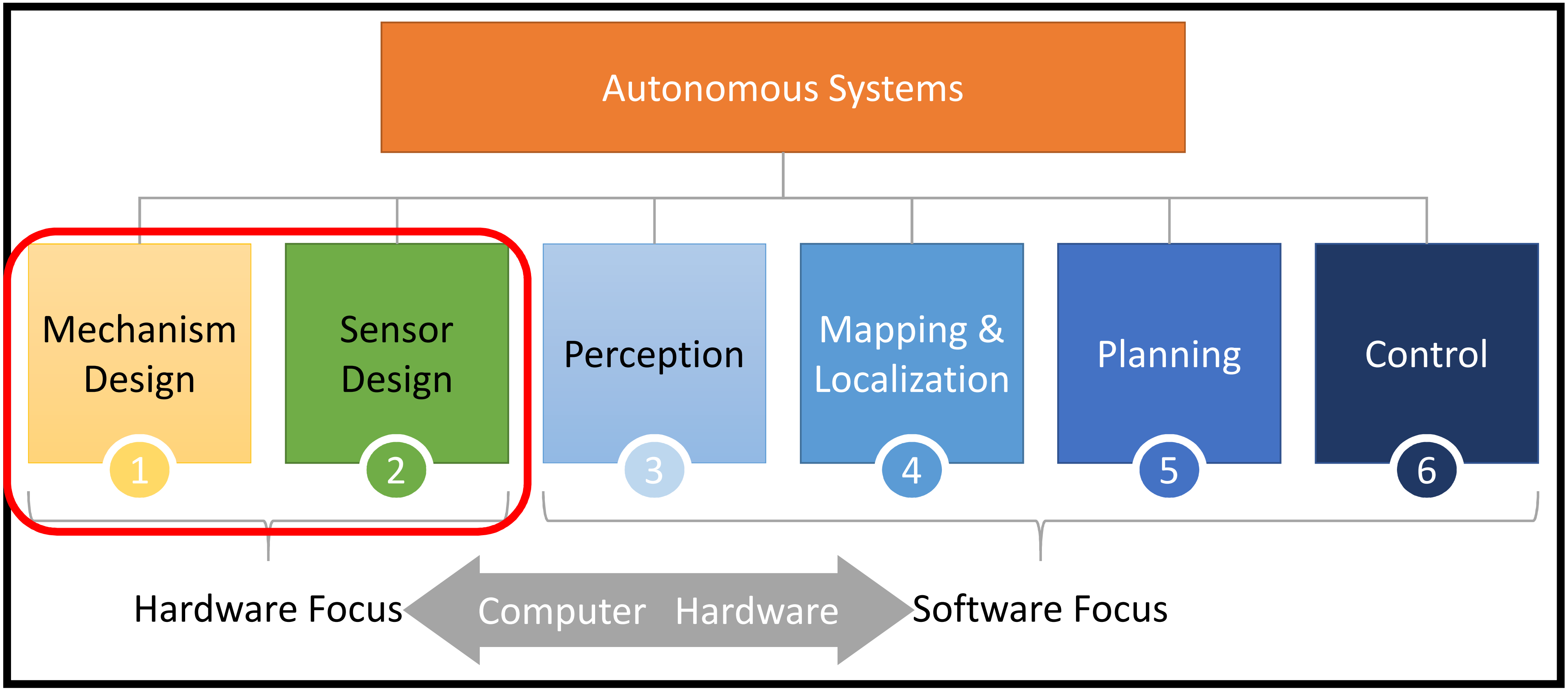
The goal for the next couple of lectures is to develop a high level understanding of:

1. What is an autonomous system
2. Key problems and constraints for autonomous systems
3. Some of the most important (classes of) algorithms in robotics
    A. The model based vs. model free tradeoff
    B. The online vs offline tradeoff
    C. The no free lunch theorem and the need for approximations
4. How computer systems / architecture design has and can play a role in improving autonomous systems

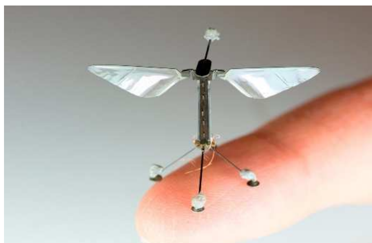
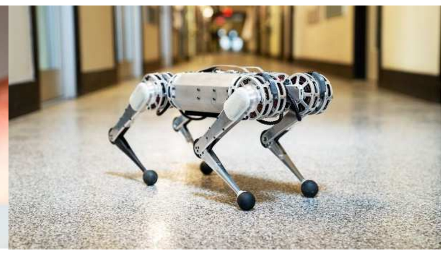
Autonomous Systems / Robotics is a BIG space



Autonomous Systems / Robotics is a BIG space



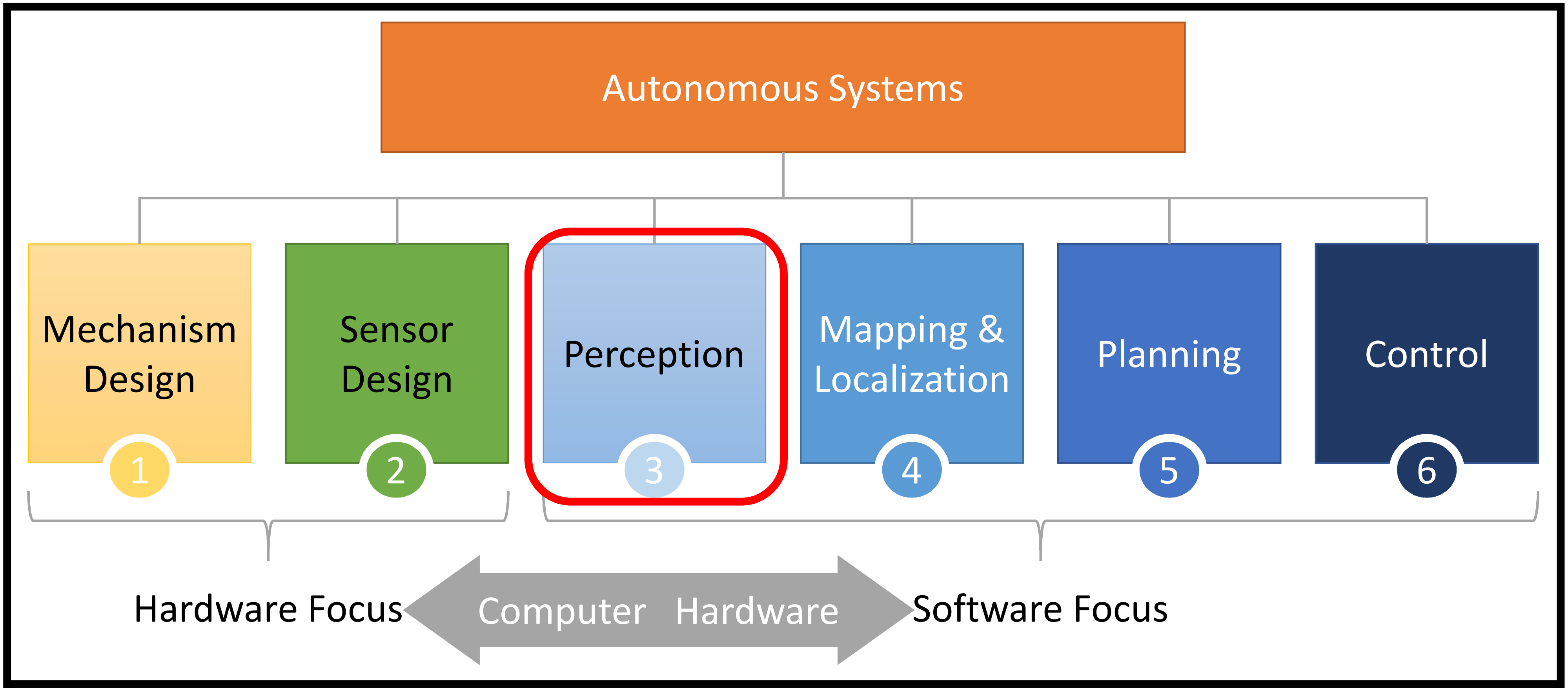
Key Takeaways:

1. When designing algorithms for robots you need to understand the physical capabilities of the robot and you (potentially) need to understand how to model its physical behaviors

2. Different kinds of systems will have different power, weight, and performance budgets for computer hardware

Autonomous Systems / Robotics is a BIG space

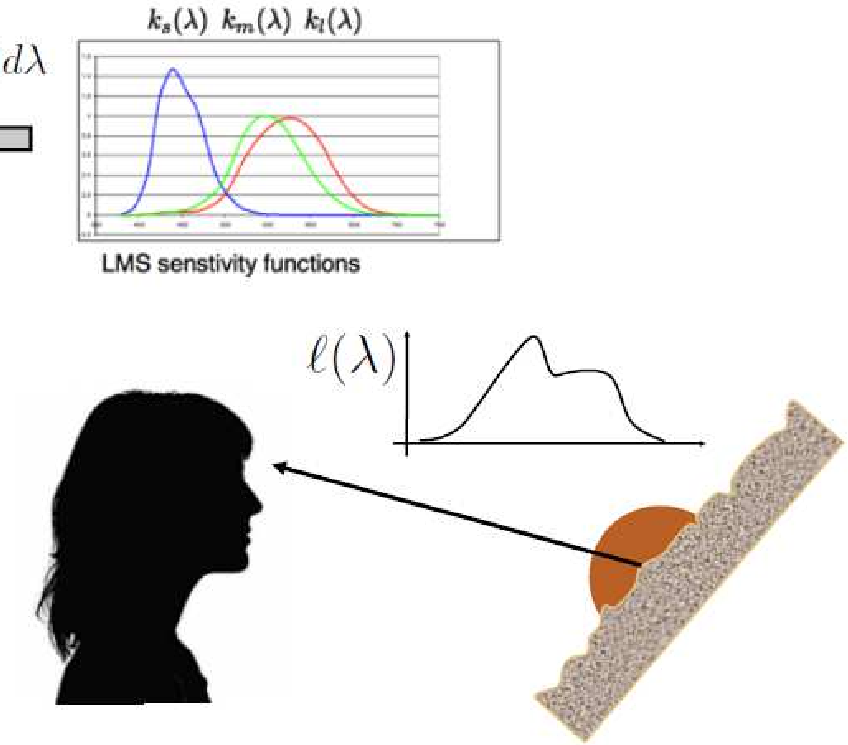


3 Computer Vision (and Perception in general) is hard

$$c_s = \int k_s(\lambda)\ell(\lambda)d\lambda$$

Retinal color

$$\mathbf{c}(\ell(\lambda)) = (c_s, c_m, c_l)$$



Perceived color

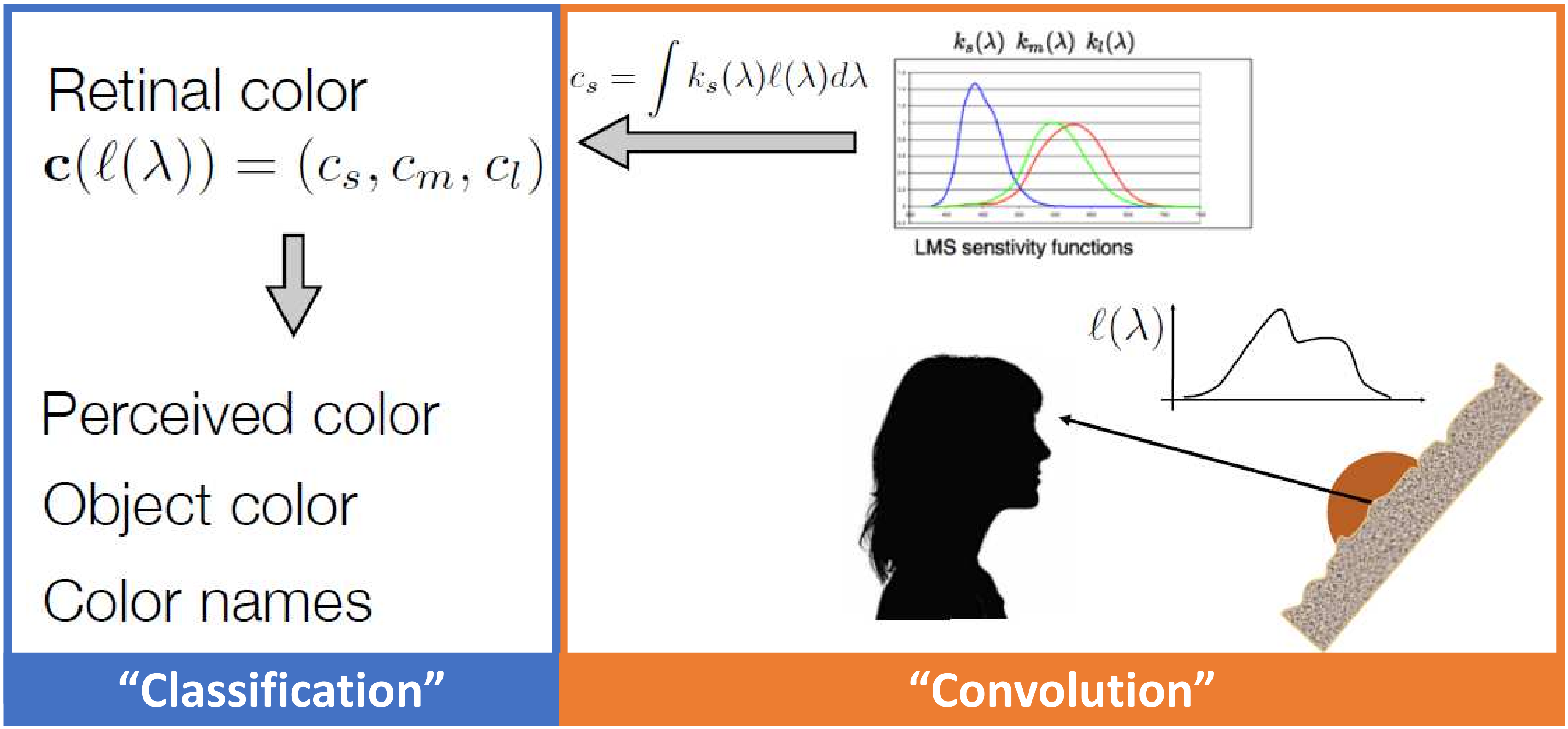
Object color

Color names

Slide Credit: Todd Zickler CS 283

3

CV/Perception is solved by modeling and approximating the classification of convolution



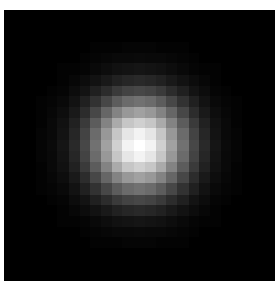
Slide Credit: Todd Zickler CS 283

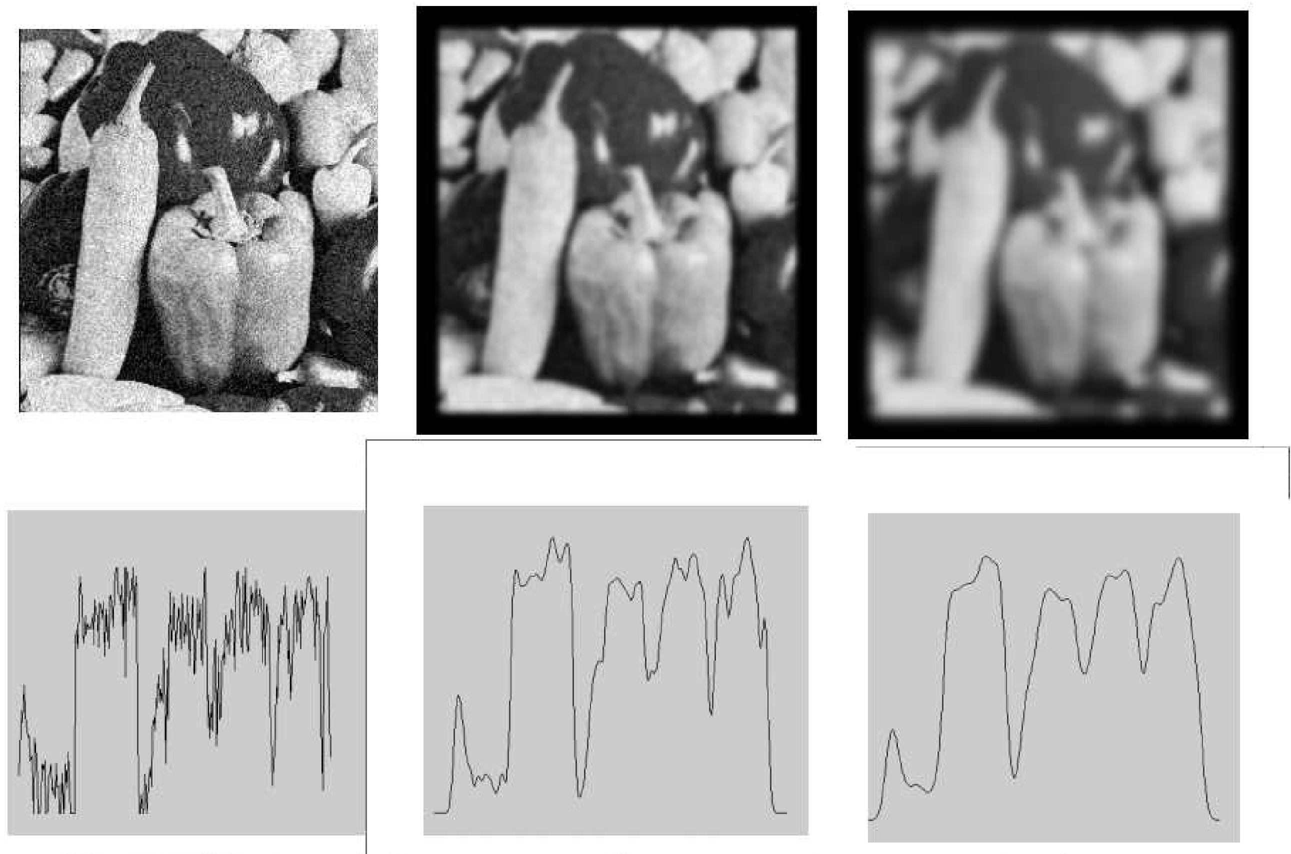
3 We approximate convolution using linear filters

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$



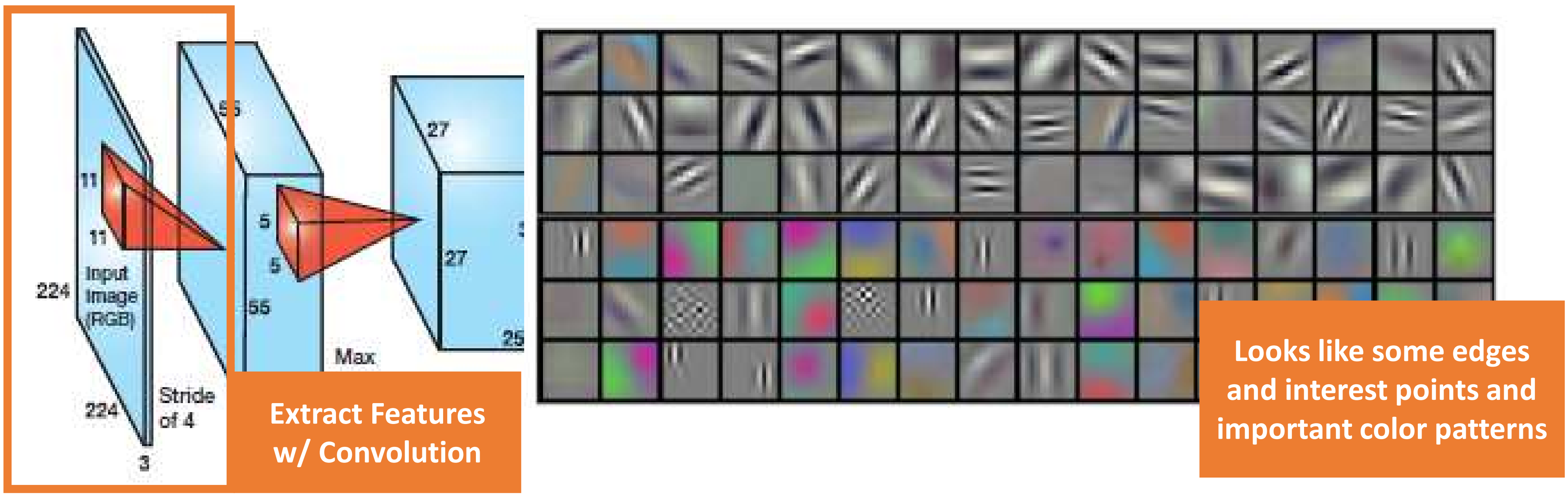


No smoothing $\sigma = 2$ $\sigma = 4$

3

Deep learning automates the design of filters, and the selection/combination of features for classification

AlexNet: the first widely successful application of deep learning

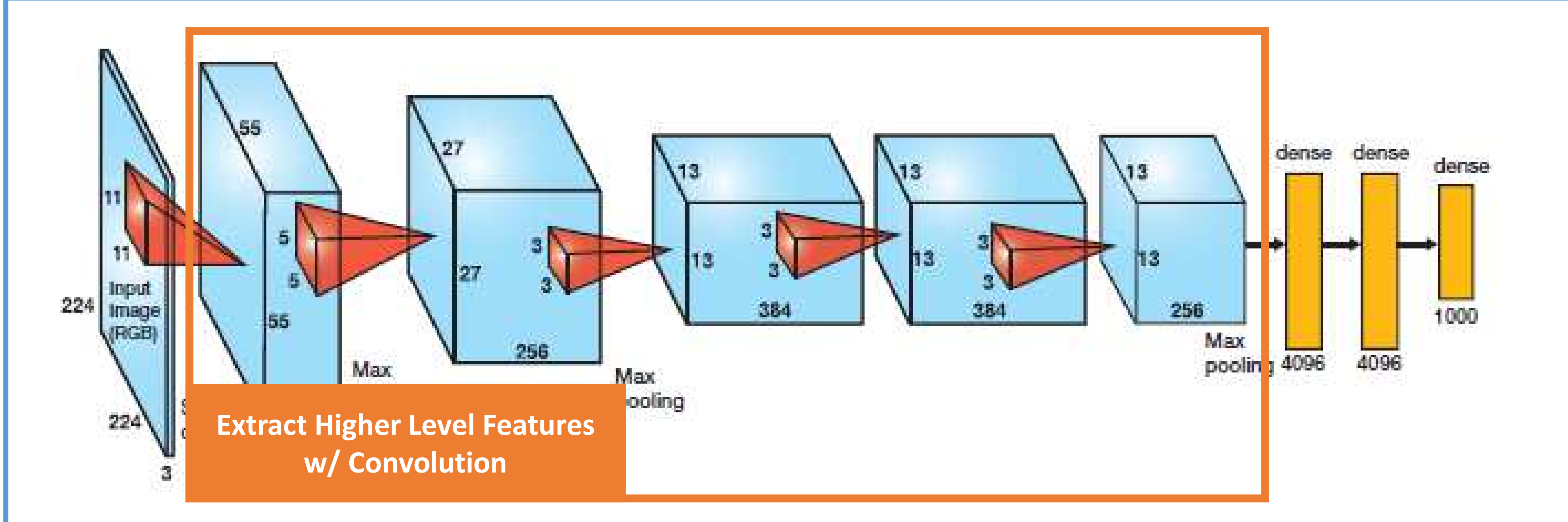


https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/

3

Deep learning automates the design of filters, and the selection/combination of features for classification

AlexNet: the first widely successful application of deep learning

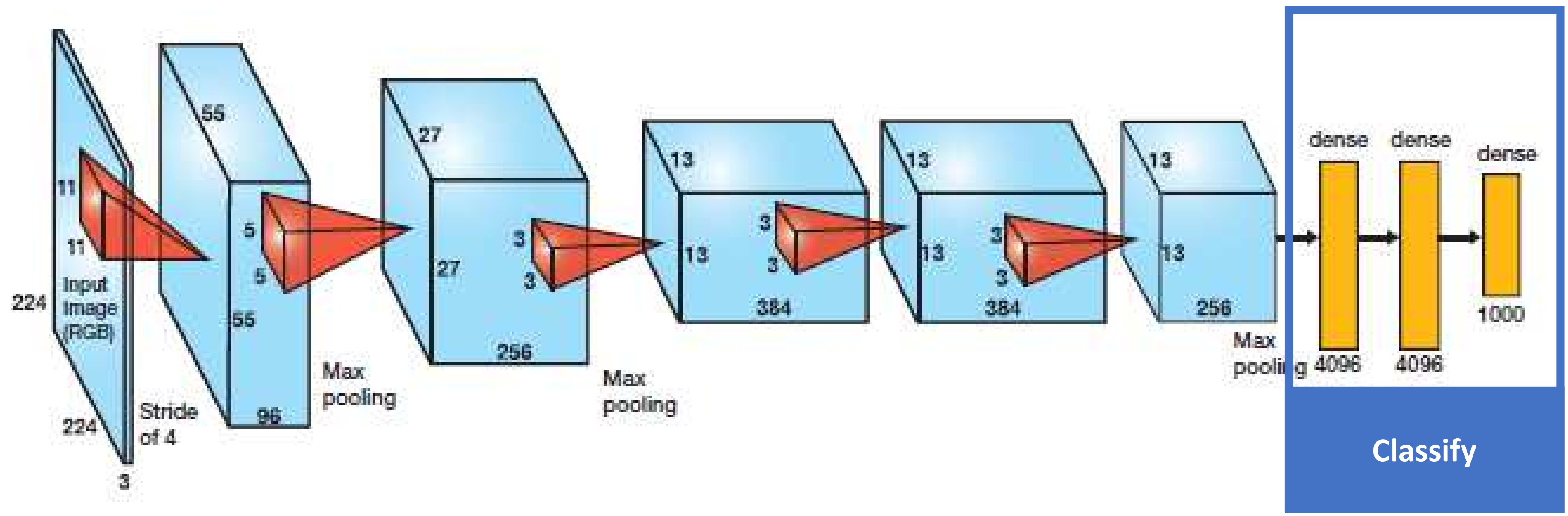


https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/

3

Deep learning automates the design of filters, and the selection/combination of features for classification

AlexNet: the first widely successful application of deep learning



https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/

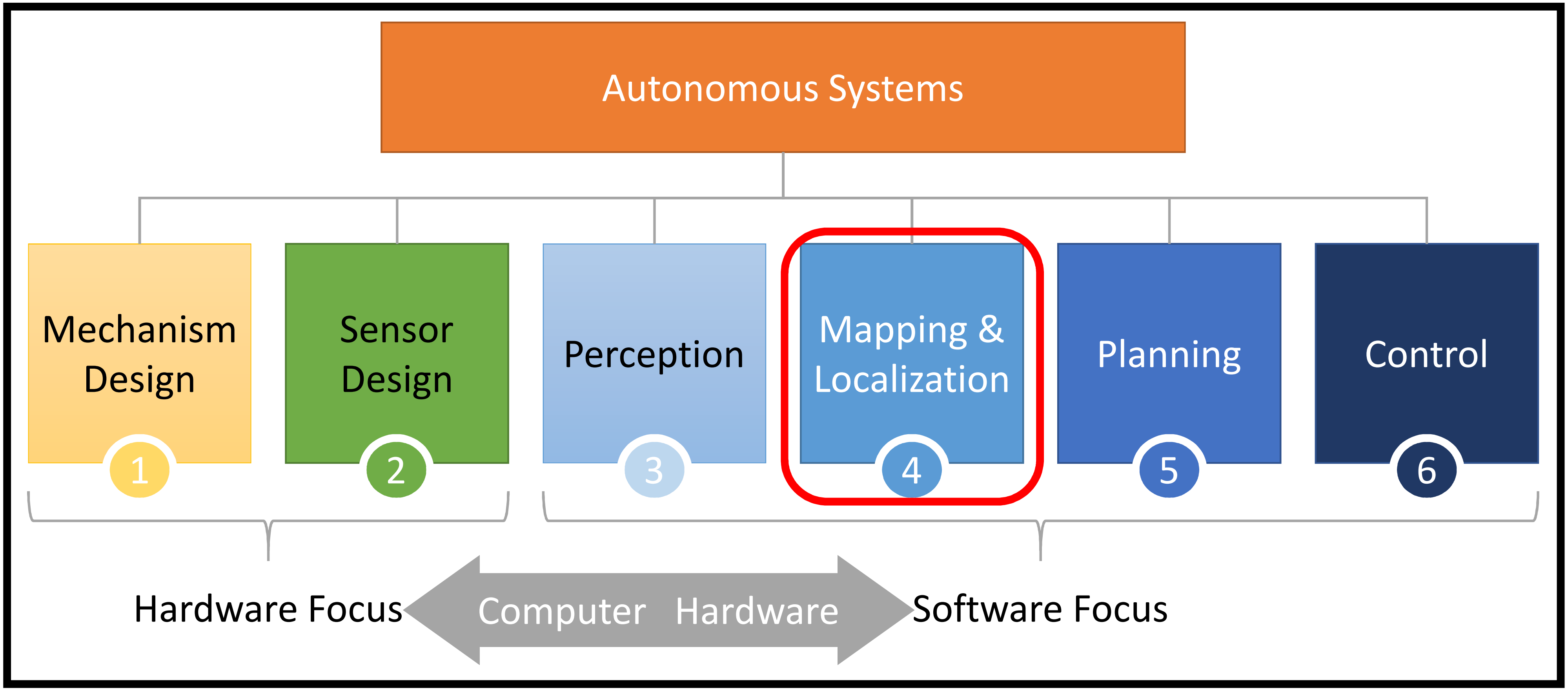
3

Deep learning automates the design of filters, and the selection/combination of features for classification

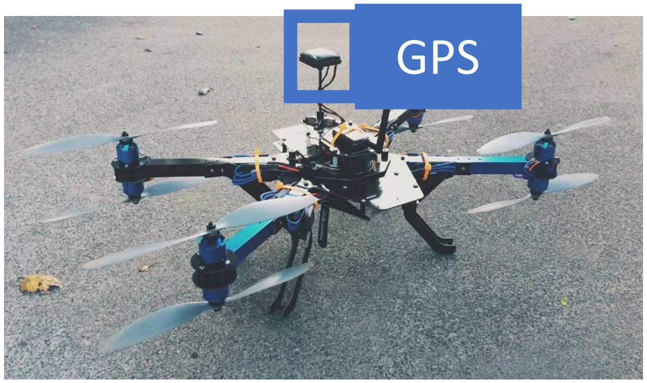
But watch our for adversarial attacks on the math!

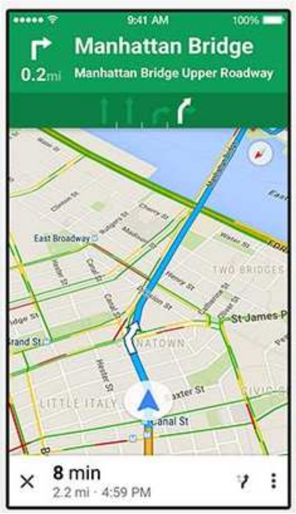
"No Free Lunch!"

Autonomous Systems / Robotics is a BIG space



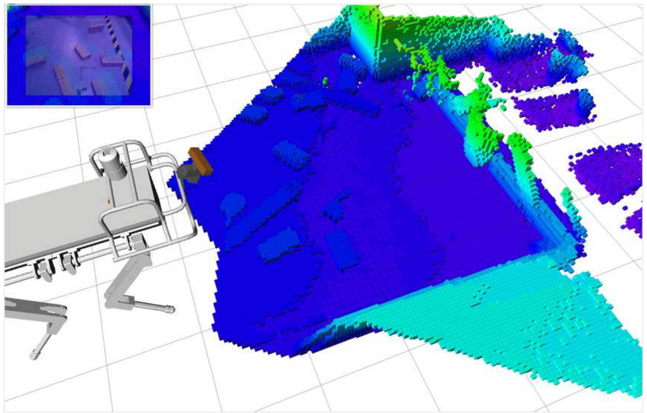
4 Mapping/Localization is hard

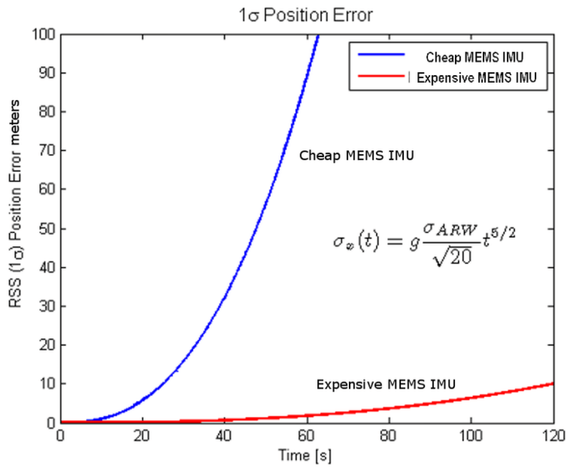




Three Problems

1. GPS is only accurate to O(10m)
2. GPS relies on already having a perfect map of the environment (unrealistic often)
3. Other sensor data is also quite noisy!





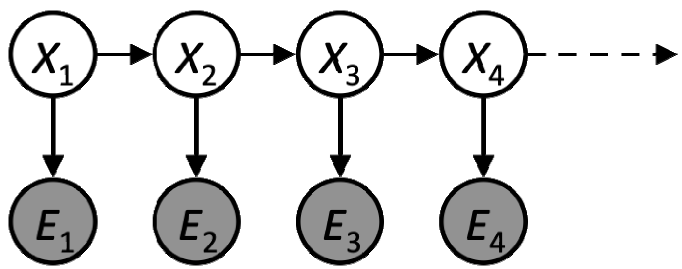
4 Mapping/Localization is solved by modeling the world as an HMM and using modeling and approximating to solve it

Track the **Belief State B_t** of the **state and landmarks**

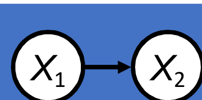
$$\boldsymbol{B_t = p(X_t | X_o, E_o \cdots E_{t-1})}$$

Hidden Markov Model (HMM)

States X update in time but we only observe the effects E

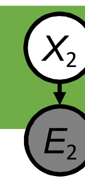


Time Update



$$P(x_{t+1} | x_0, e_1 \dots e_t) = \int P(x_t | x_0, e_1 \dots e_t) * P(x_{t+1} | x_t)$$

Evidence Update



$$P(x_{t+1} | x_0, e_1 \dots e_{t+1}) \propto \int P(x_{t+1} | x_0, e_1 \dots e_t) * P(e_{t+1} | x_{t+1})$$

4 Mapping/Localization is solved by modeling the world as an HMM and using modeling and approximating to solve it

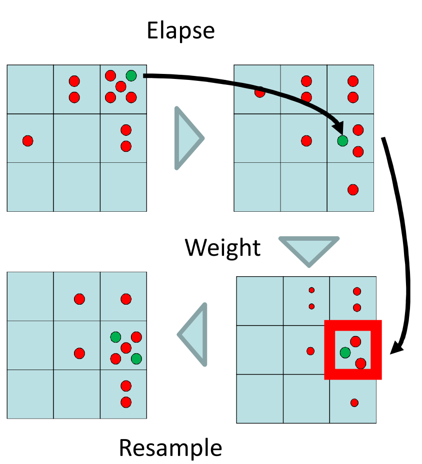
Time Update $X_1 \rightarrow X_2$

$$P(x_{t+1}|x_0, e_1 \dots e_t) = \int P(x_t|x_0, e_1 \dots e_t) * P(x_{t+1}|x_t)$$

Evidence Update $X_2 \rightarrow E_2$

$$P(x_{t+1}|x_0, e_1 \dots e_{t+1}) \propto \int P(x_{t+1}|x_0, e_1 \dots e_t) * P(e_{t+1}|x_{t+1})$$

Particle Filter



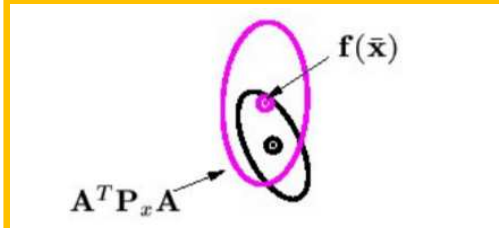
Approximate with Samples

Model With Gaussian

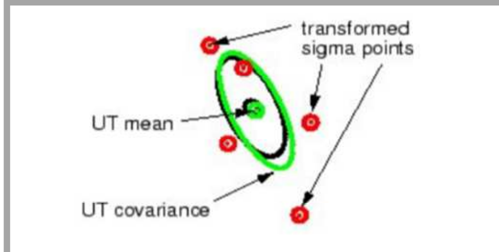
Approximate as Linear

Approximate with Samples

Extended Kalman Filter (EKF)



Unscented Kalman Filter - UKF



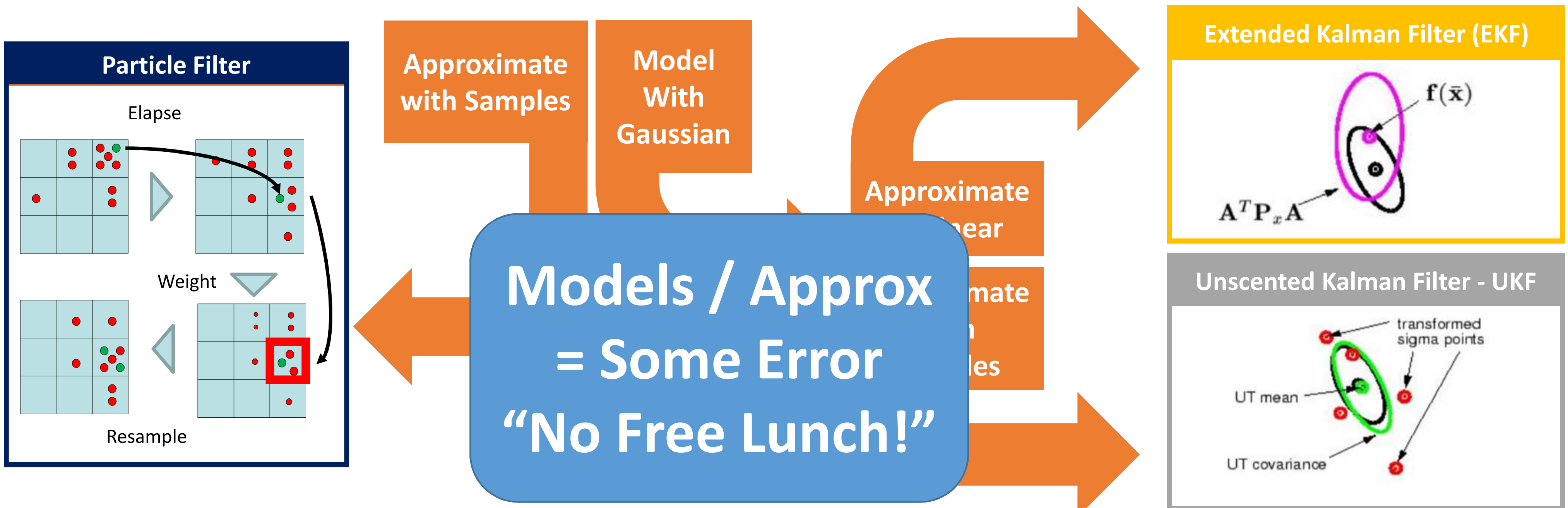
4 Mapping/Localization is solved by modeling the world as an HMM and using modeling and approximating to solve it

Time Update $X_1 \rightarrow X_2$

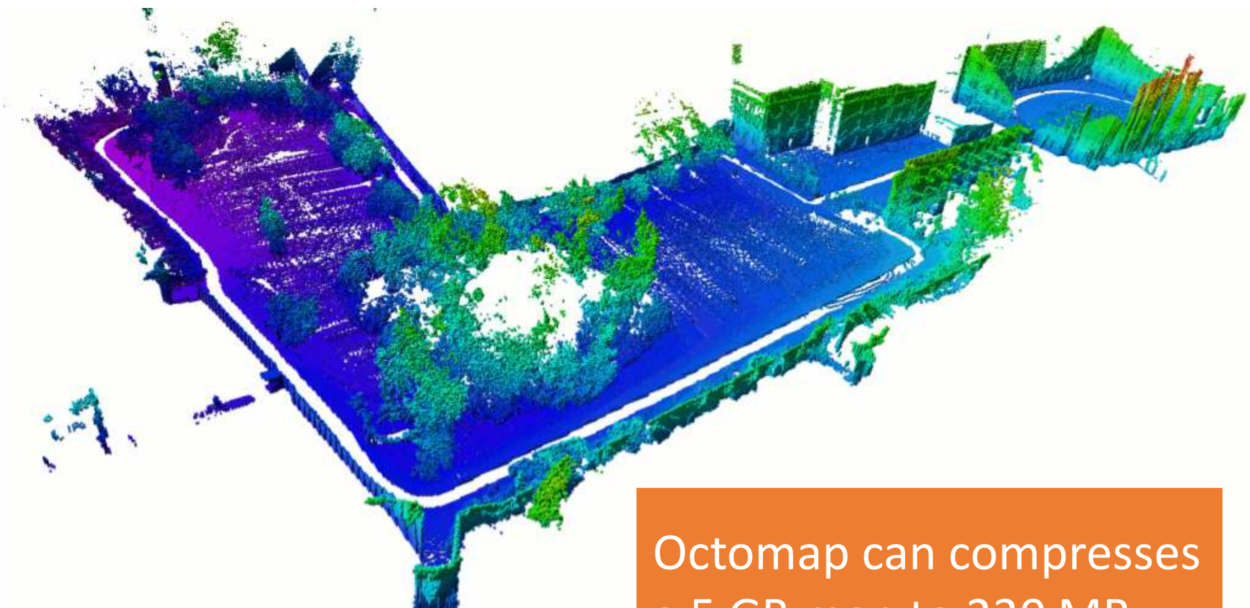
$$P(x_{t+1}|x_0, e_1 \ldots e_t) = \int P(x_t|x_0, e_1 \ldots e_t) * P(x_{t+1}|x_t)$$

Evidence Update $X_2 \rightarrow E_2$

$$P(x_{t+1}|x_0, e_1 \ldots e_{t+1}) \propto \int P(x_{t+1}|x_0, e_1 \ldots e_t) * P(e_{t+1}|x_{t+1})$$



4 Also we need to approximate the resolution of our maps and store them intelligently to fit them in memory



Octomap can compresses a 5 GB map to 230 MB

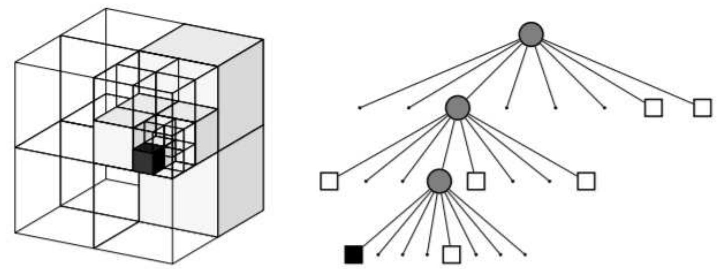


Fig. 2 Example of an octree storing free (shaded white) and occupied (black) cells. The volumetric model is shown on the left and the corresponding tree representation on the right.

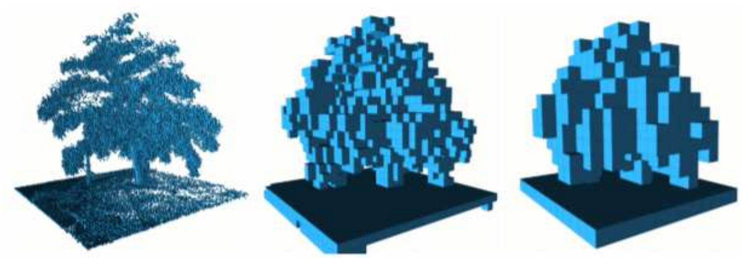
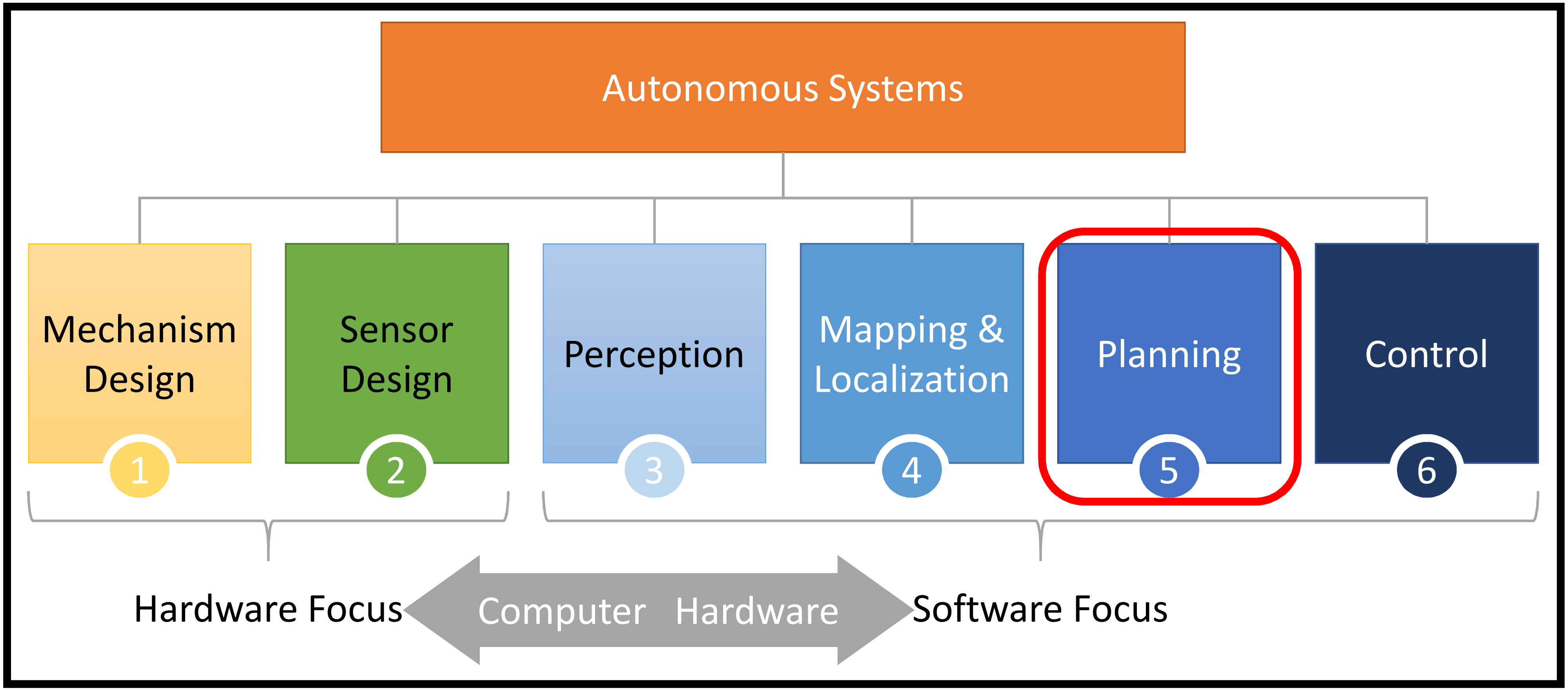


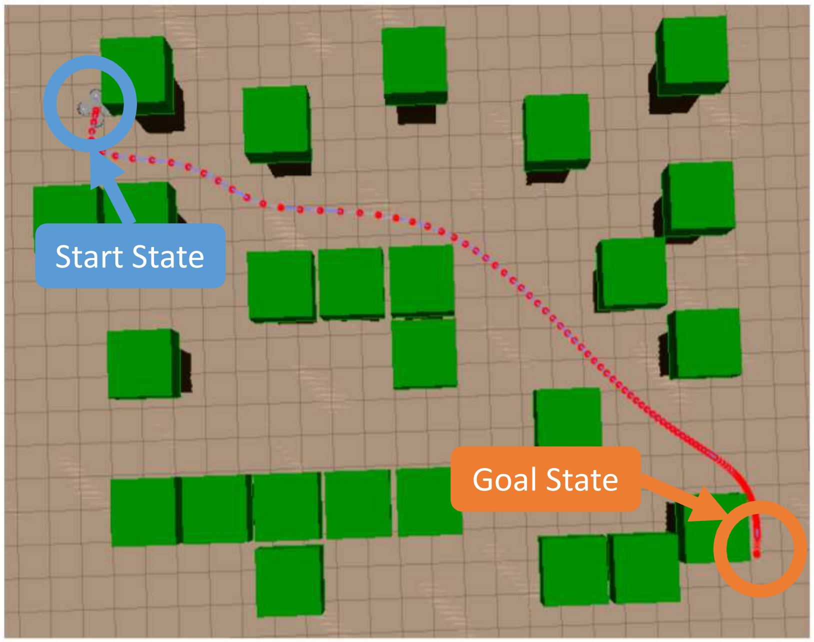
Fig. 3 By limiting the depth of a query, multiple resolutions of the same map can be obtained at any time. Occupied voxels are displayed in resolutions 0.08 m, 0.64 , and 1.28 m.

“Octomap” Hornung et. al. 2012

Autonomous Systems / Robotics is a BIG space



5 Planning (in Configuration Space) is hard



One approach is to ***discretize*** the statespace (grid it) and use graph search (A* = fast)

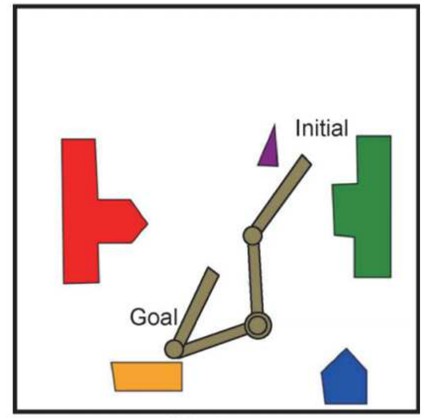
Another is to solve a global optimization problem:

$$\underset{s_0, a_0, \ldots, s_N, a_N}{\text{minimize}} \sum_{k=0}^{N} c(s_k, a_k)$$

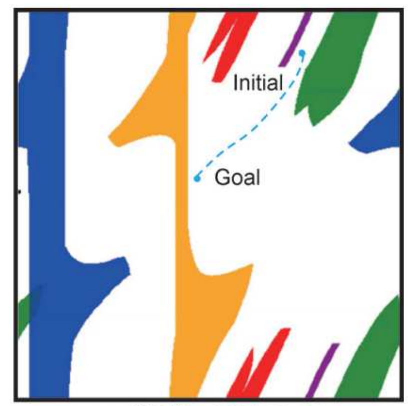
$$\text{subject to} \quad s_{k+1} = f(s_k, a_k)$$

$$s_N = s_{\text{goal}}$$

Complexity scales with $d^{|S| = |A|}$: **Curse of Dimensionality**



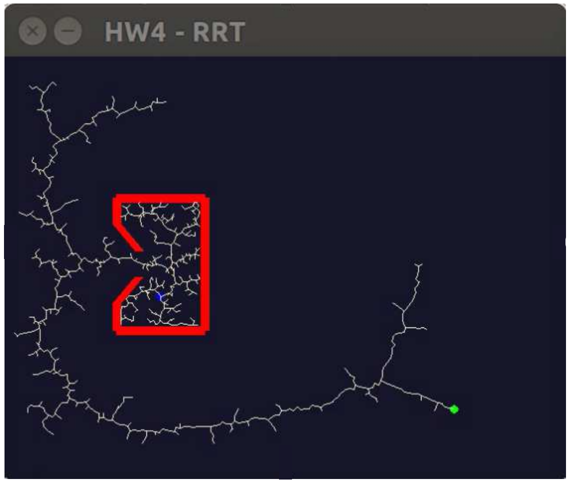
Workspace



Configuration space

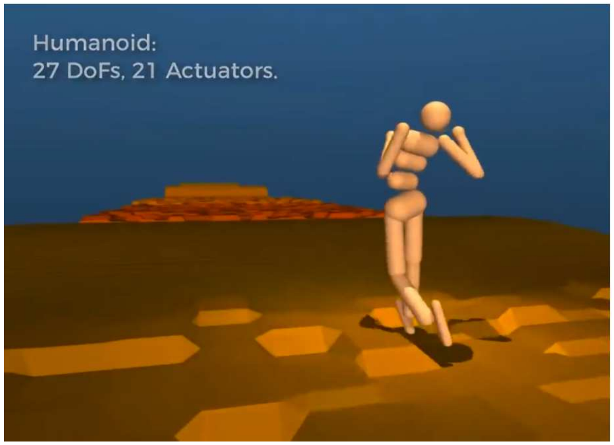
5

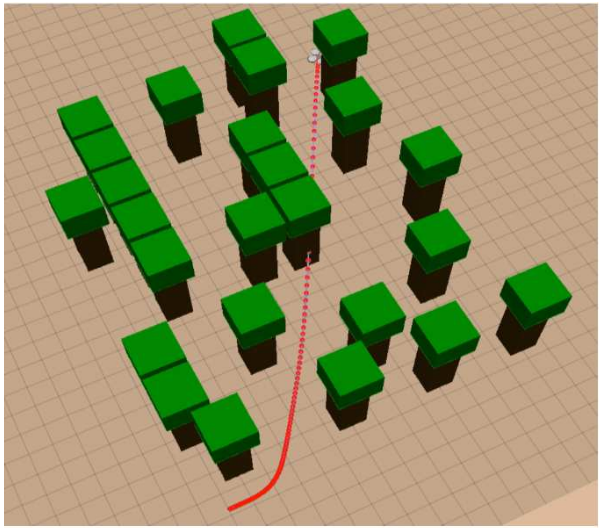
There are three main ways to approximately plan in Configuration Space



Random Search

Machine Learning

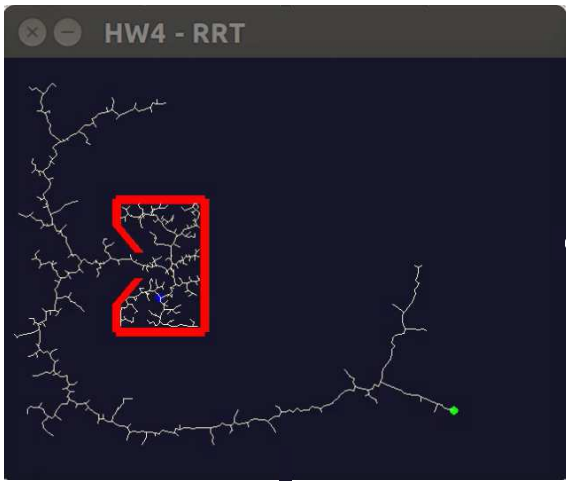




Local Search

We can approximately plan locally optimal plans in Configuration Space in three ways

5



Random Search

Online probabilistic completeness and optimality with RRT*

Offline/Online for "multi-query" with PRM

RRT

RRT*

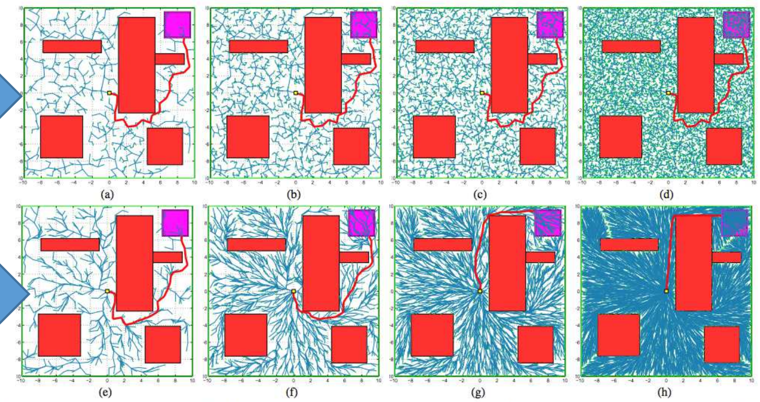
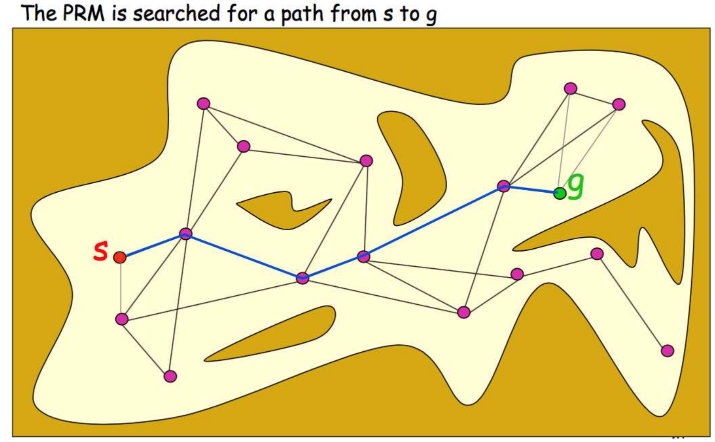
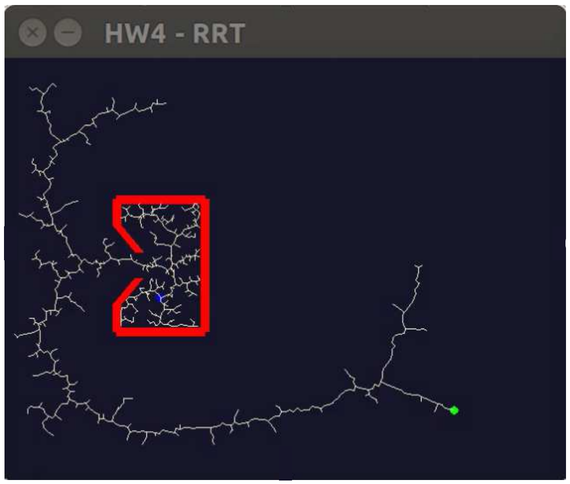


Fig. 1. A Comparison of the RRT* and RRT algorithms on a simulation example. The tree maintained by the RRT algorithm is shown in (a)-(d) in different stages, whereas that maintained by the RRT* algorithm is shown in (e)-(h). The tree snapshots (a), (e) are at 1000 iterations , (b), (f) at 2500 iterations, (c), (g) at 5000 iterations, and (d), (h) at 15,000 iterations. The goal regions are shown in magenta. The best paths that reach the target are highlighted with red.

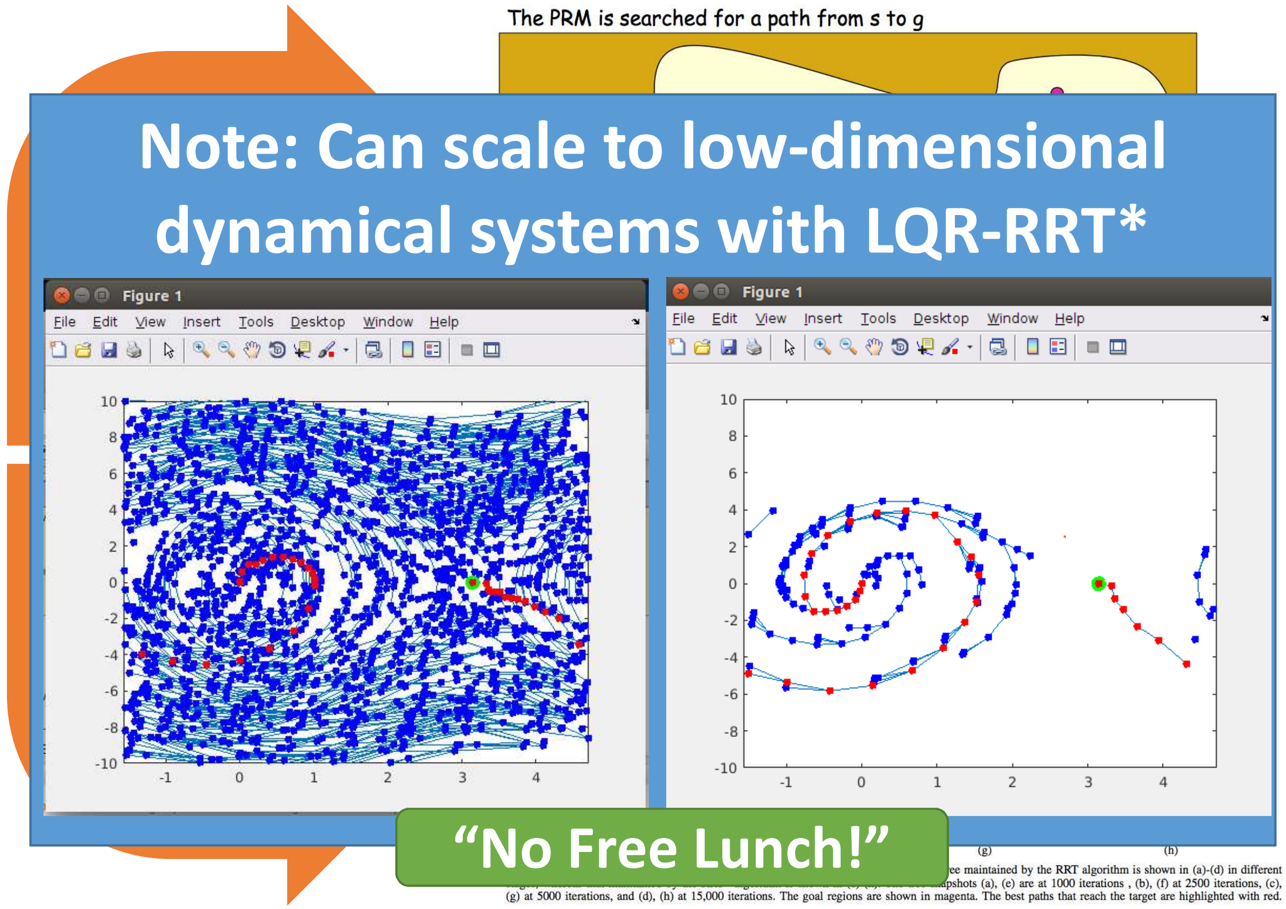
The PRM is searched for a path from s to g



5 We can approximately plan locally optimal plans in Configuration Space in three ways



Random Search

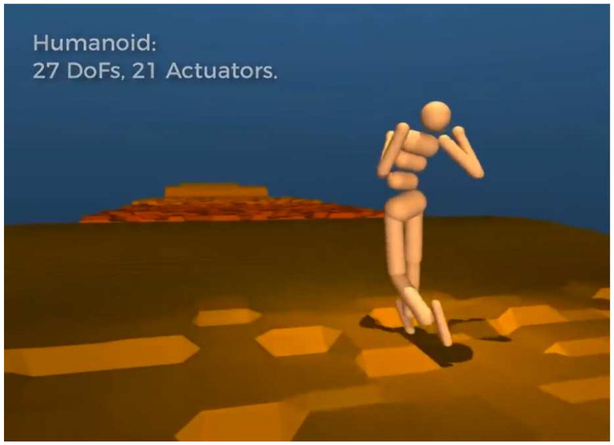


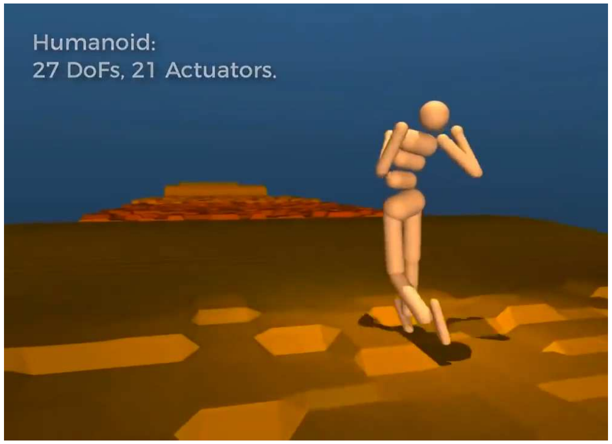
Note: Can scale to low-dimensional dynamical systems with LQR-RRT*

“No Free Lunch!”

5 We can approximately plan locally optimal plans in Configuration Space in three ways

Machine Learning





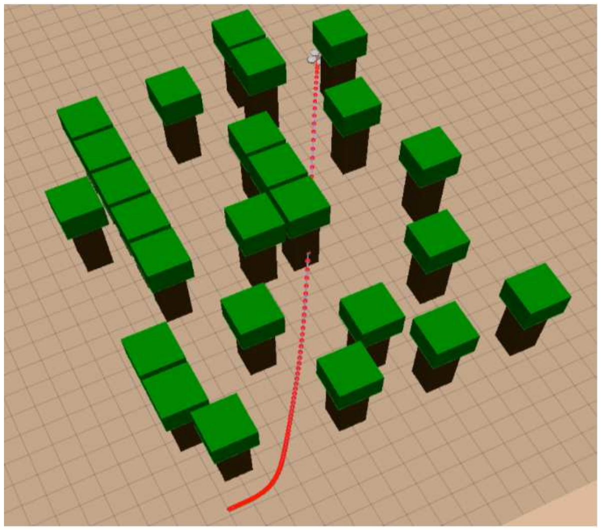
**My two cents:
Yes, and no free lunch!**

Needs to re-lean physics and suffers from sample complexity

In two weeks more on this!

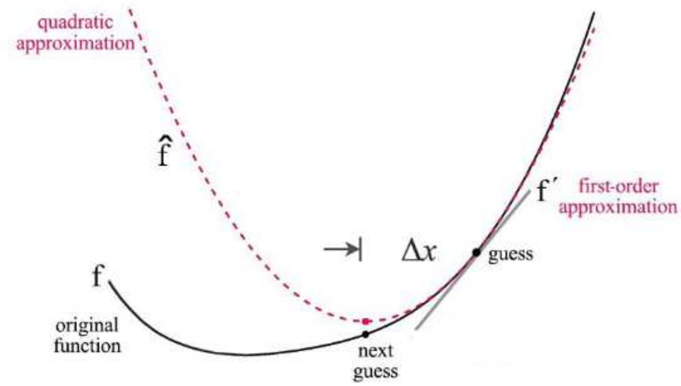
5

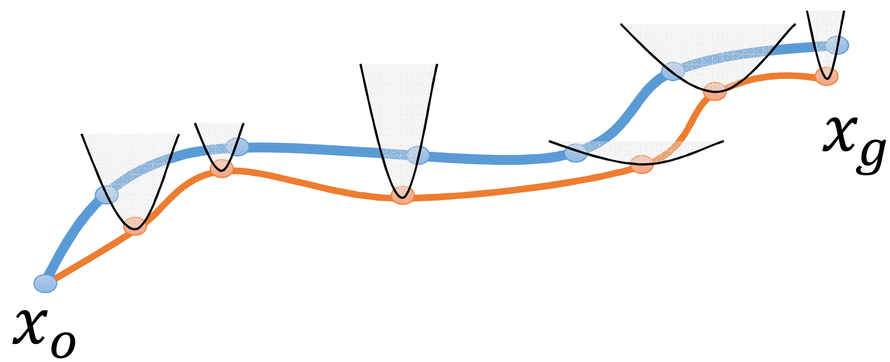
We can approximately plan locally optimal plans in Configuration Space in three ways



Local Search

Solve math locally with linear & quadratic approximations



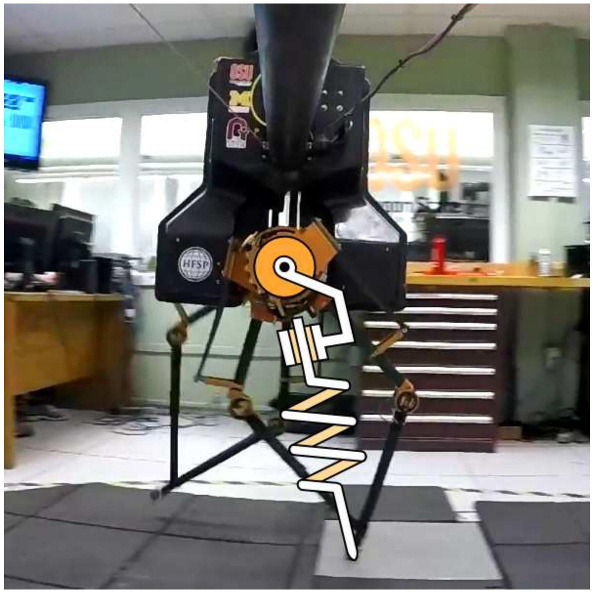


5

Practical Challenges for Trajectory Optimization: Not Complete, Not Robustness and Contact = No Free Lunch!

1. **Not complete** (aka no guaranteed solution) and often slow!
2. Solvers are **numerically sensitive**
3. Solutions are sensitive to initial trajectories and **perturbations**
4. The physics equations are fundamentally different when an object makes or breaks **contact** leading to a **combinatorial explosion**

One approach to avoid solving these large hard problems is to solve the problem by **combining simpler models** of the system although this leads to **conservative** behavior

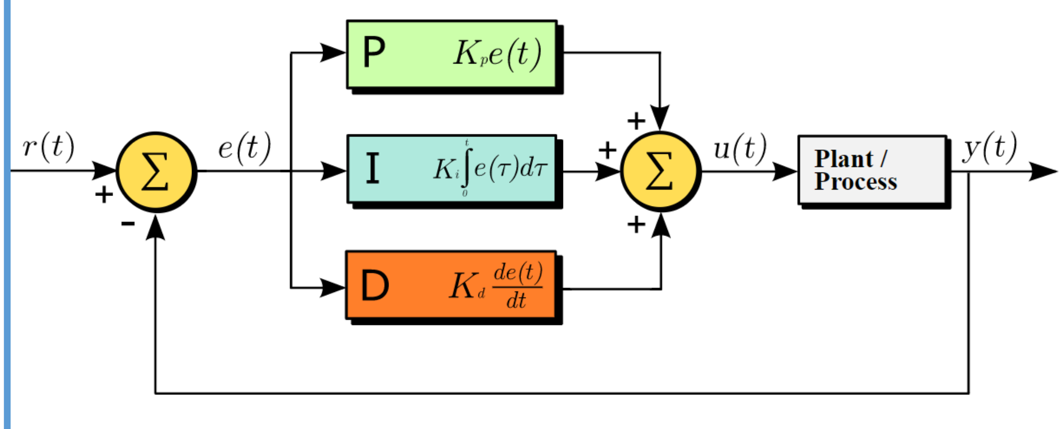


5 Control is hard (even for the experts)

6 We use feedback tracking controllers to run our plans in the real world (and handle the differences encountered)

Model as linear combination of errors and approximate gains

This is the canonical **PID controller**!



LQR: Quadratic Cost with Linear Dynamics

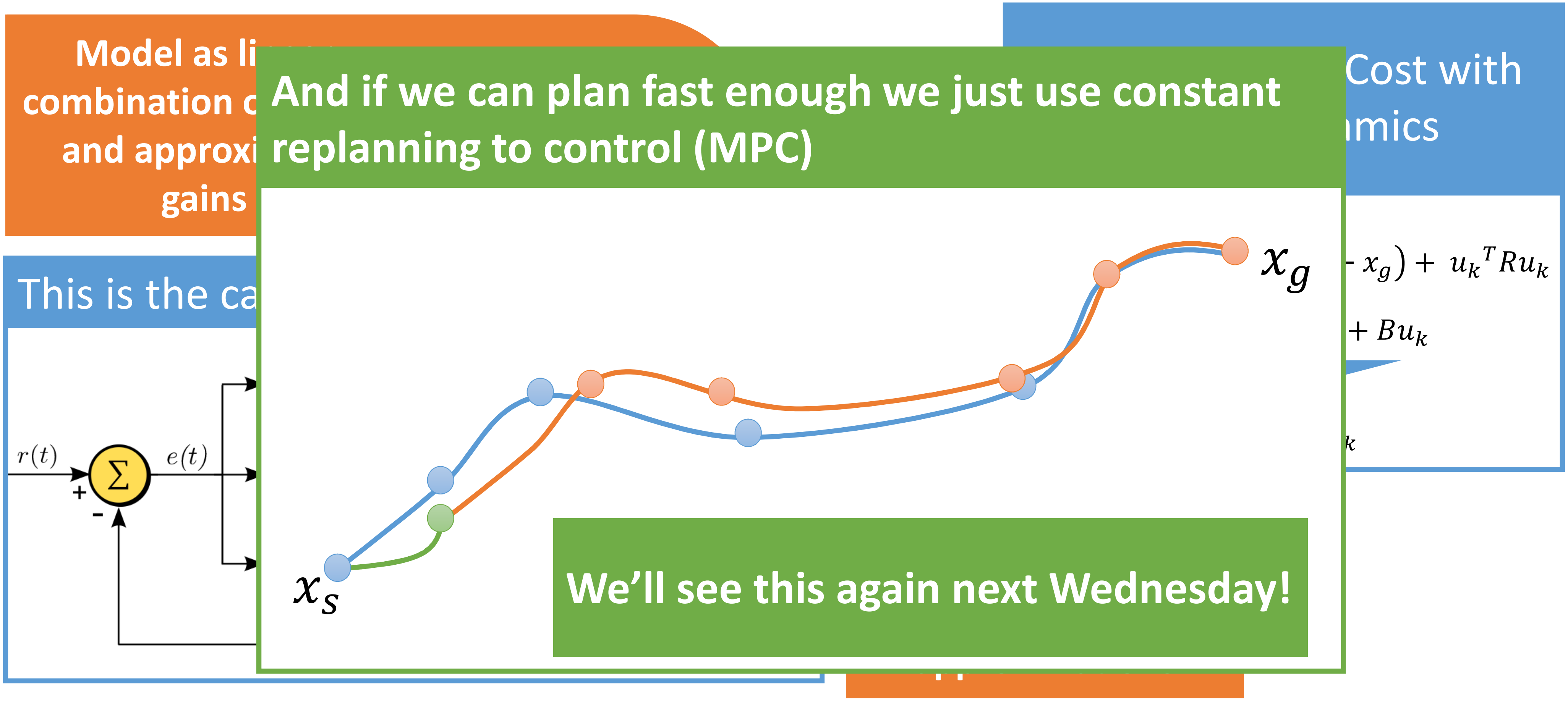
$$\min_{x,u} \sum_{k=0}^{N} (x_k - x_g)^T Q (x_k - x_g) + u_k{}^T R u_k$$

$$\text{s.t.} \quad x_{k+1} = A x_k + B u_k$$

$$u_k = -K_k x_k$$

Solve math locally with linear & quadratic approximations

6 We use feedback tracking controllers to run our plans in the real world (and handle the differences encountered)



6 Practical Challenges for Control: Contact

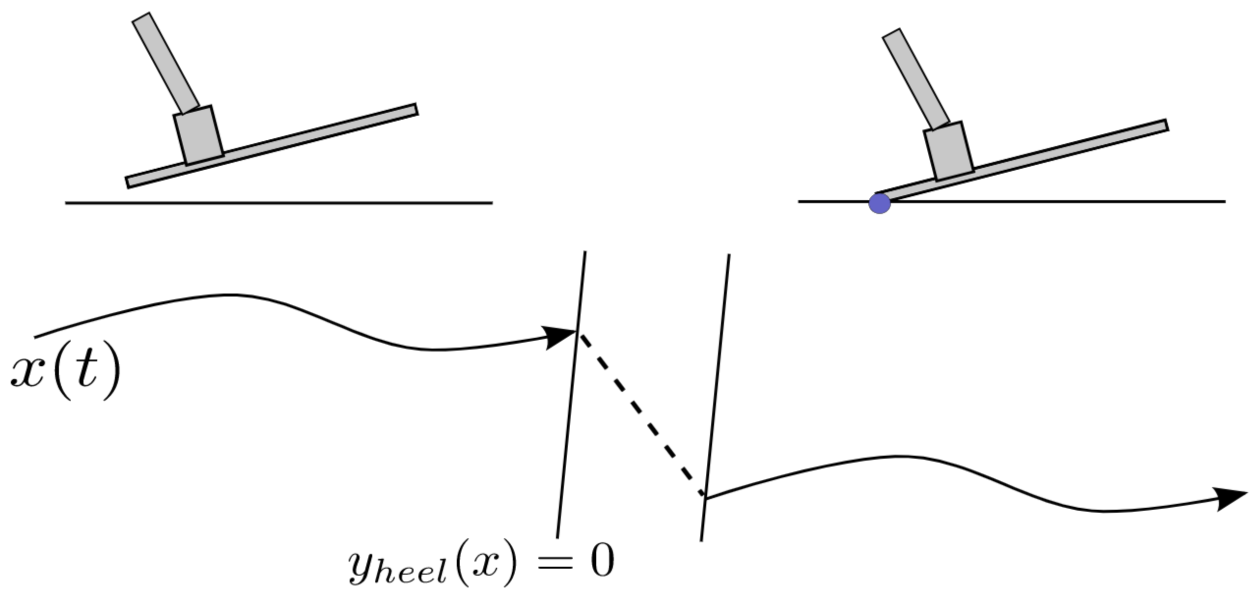


Figure 17.1 - Modeling contact as a hybrid system.

The goal for the next couple of lectures is to develop a high level understanding of:

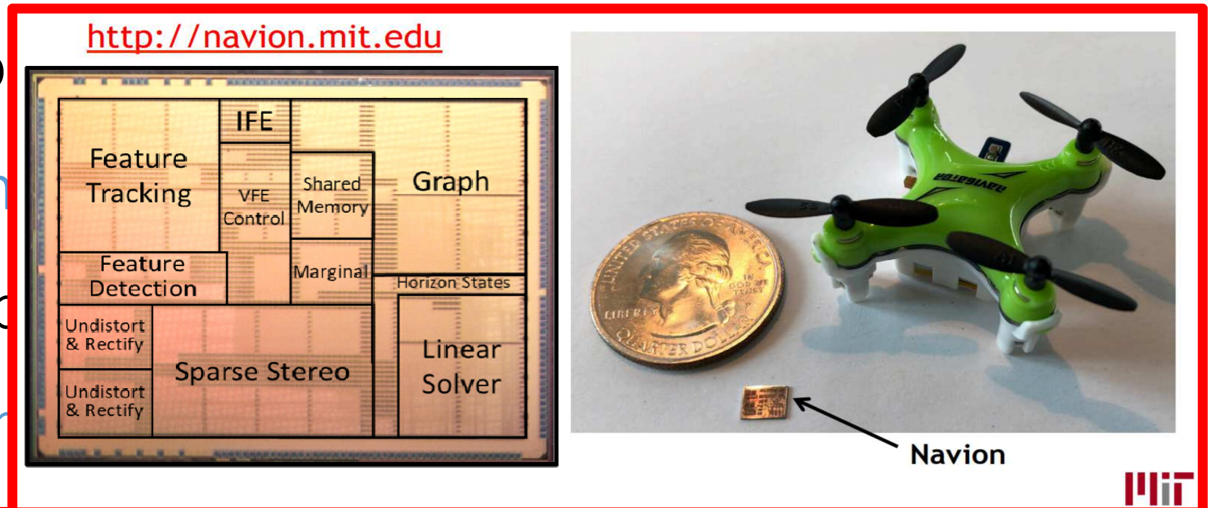
1. What is an autonomous system
2. Key problems and constraints for autonomous systems
3. Some of the most important (classes of) algorithms in robotics
   A. The model based vs. model free tradeoff
   B. The online vs offlin
   C. The no free lunch t … mations

This is what we will explore in all of the papers!

4. How computer systems / architecture design has and can play a role in improving autonomous systems

The goal for the next couple of lectures is to develop a high level understanding of:

1. What is an autonomo



2. Key problems and con
3. Some of the most imp
   A. The model based vs. r
   B. The online vs offlin
   C. The no free lunch t ... mations

This is what we will explore in all of the papers!

4. How computer systems / architecture design has and can play a role in improving autonomous systems

Your homework – get on HOTCRP

Email **Glenn Holloway:**
holloway@eecs.harvard.edu

He will send you a password (username is that email address) after which I can assign you access to review papers