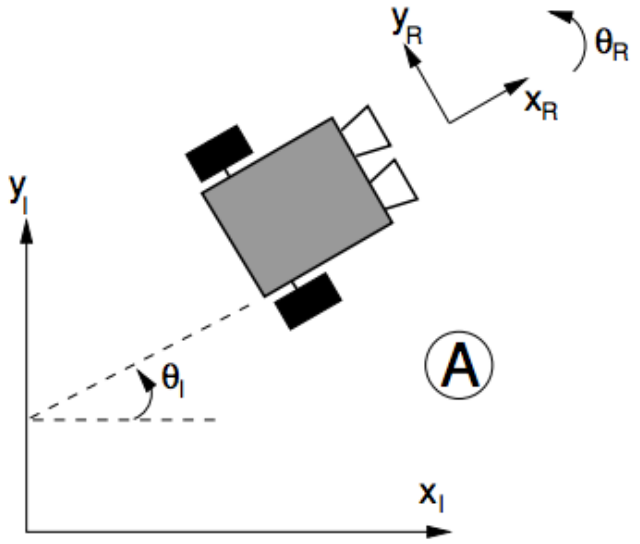
- Robot Position: $\xi_I = [x_I, y_I, \theta_I]^T$
- Mapping between frames

R vs. r
I vs l vs L vs l

$$\dot{\xi}_R = R(\theta)\,\dot{\xi}_I$$

$$\dot{\xi}_I = R(\theta)^{-1}\,\dot{\xi}_R$$

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$R(\theta)^{-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- l: distance from wheel to center of rotation
- r: radius of a wheel

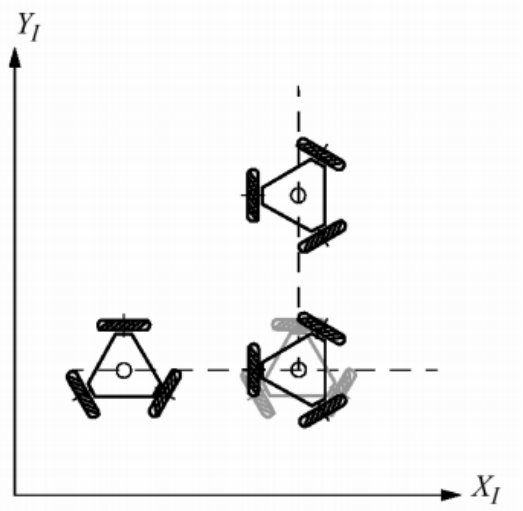
$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix} = \begin{bmatrix} \frac{r\dot{\varphi}_r}{2} + \frac{r\dot{\varphi}_l}{2} \\ 0 \\ \frac{r\dot{\varphi}_r}{2l} - \frac{r\dot{\varphi}_l}{2l} \end{bmatrix}$$

There is no ideal drive configuration that simultaneously maximizes stability, maneuverability, and controllability

Example: typically inverse correlation between controllability and maneuverability

Holonomic Robots

- Holonomic kinematic constraint can be expressed as explicit function of position variables only
- Non-holonomic constraint requires addition information
- Fixed/steered standard wheels impose non-holonomic constraints

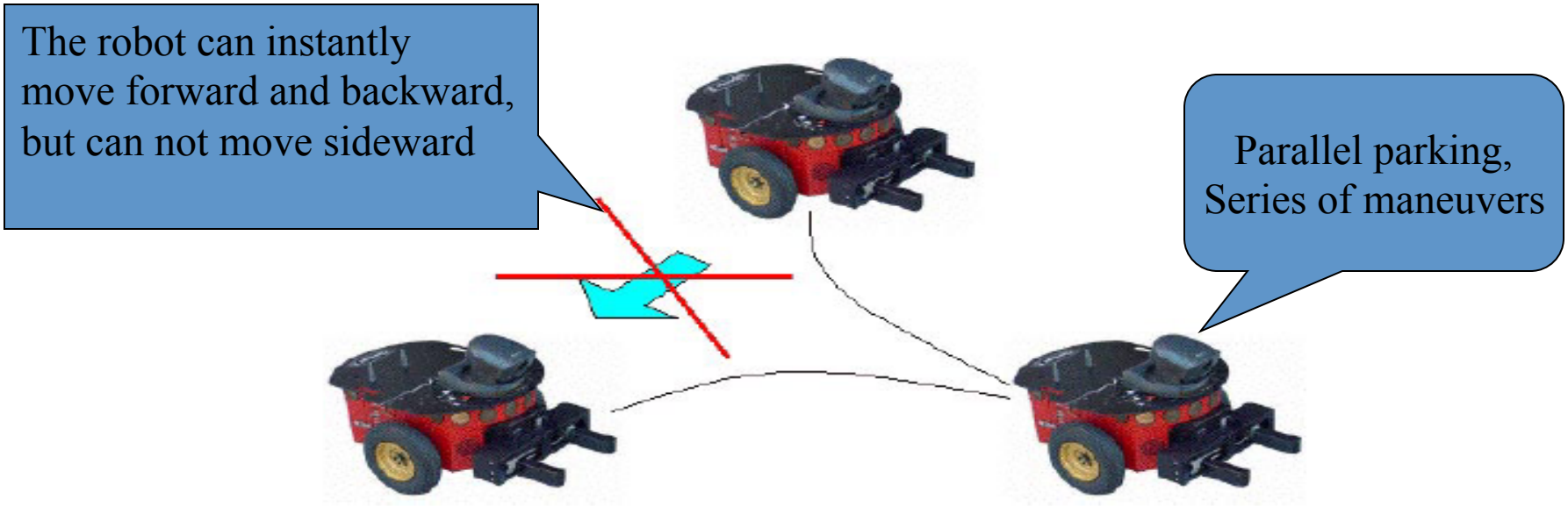


Non-holonomic constraint

A non-holonomic constraint is a constraint on the feasible **velocities** of a body

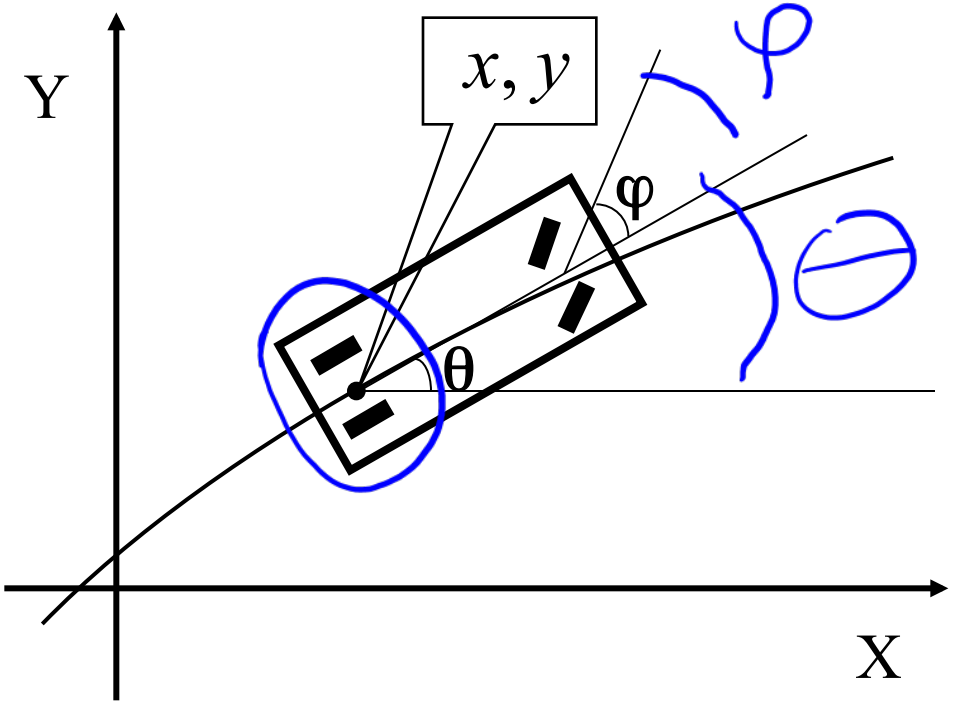
So what does that mean?

Your robot can move in some directions (forward and backward), but not others (sideward)



Kinematic model for car-like robot

- Control Input
- Driving type: Forward wheel drive



$$\{x, y, \theta, \varphi\}$$

$$\{u_1, u_2\}$$

u_1 : forward vel

u_2 : steering vel

$$\{\tau_1, \tau_2\}$$

Kinematic model for car-like robot

$$\dot{x} = u_1 \cos\theta$$

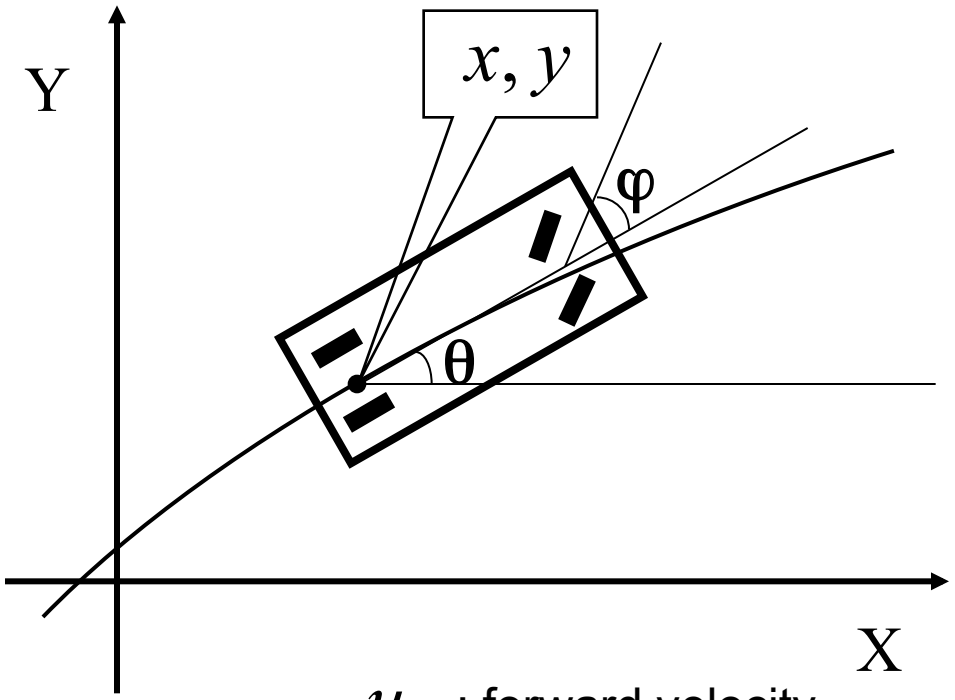
$$\dot{y} = u_1 \sin\theta$$

$$\dot{\theta} = \frac{u_1}{l} \tan\varphi$$

$$\dot{\varphi} = u_2$$

non-holonomic constraint:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0$$

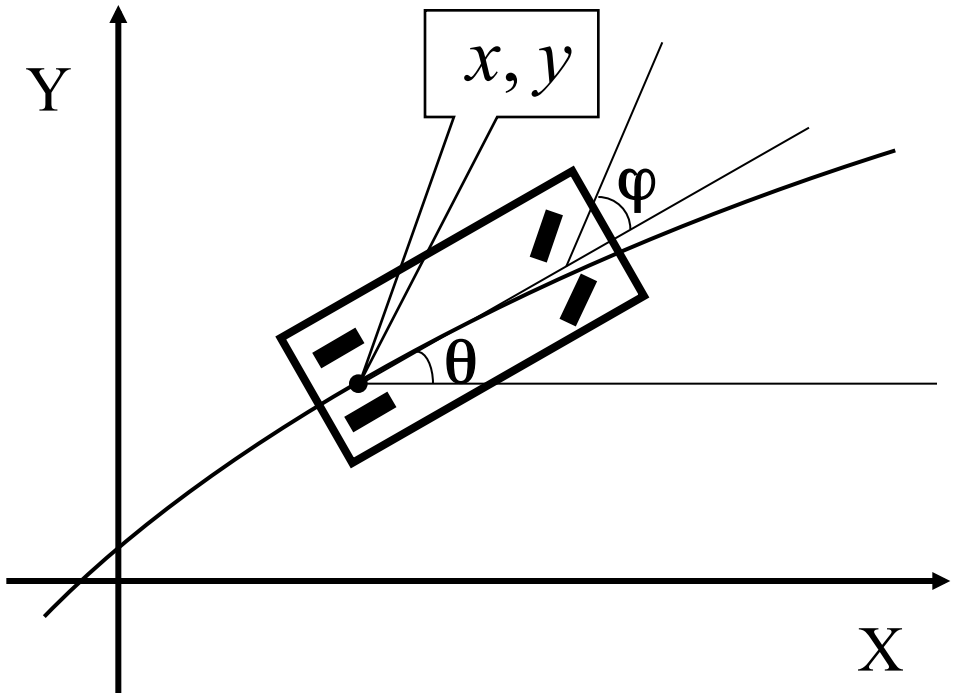


u_1 : forward velocity

u_2 : steering velocity

Dynamic Model

- Dynamic model



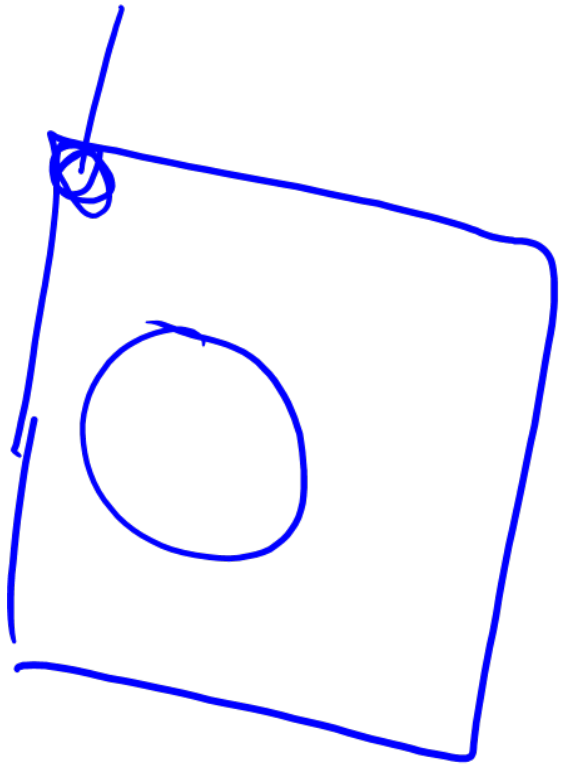
$$\begin{pmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} \sin\theta \\ \cos\theta \\ 0 \end{pmatrix} \lambda + \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

- Open loop control vs. Closed loop control

- Car’s cruse control

The World consists of...

- Obstacles
  - Places where the robot can't (shouldn't) go
- Free Space
  - Unoccupied space within the world
  - Robots "might" be able to go here
    - There may be unknown obstacles
    - The state may be unreachable

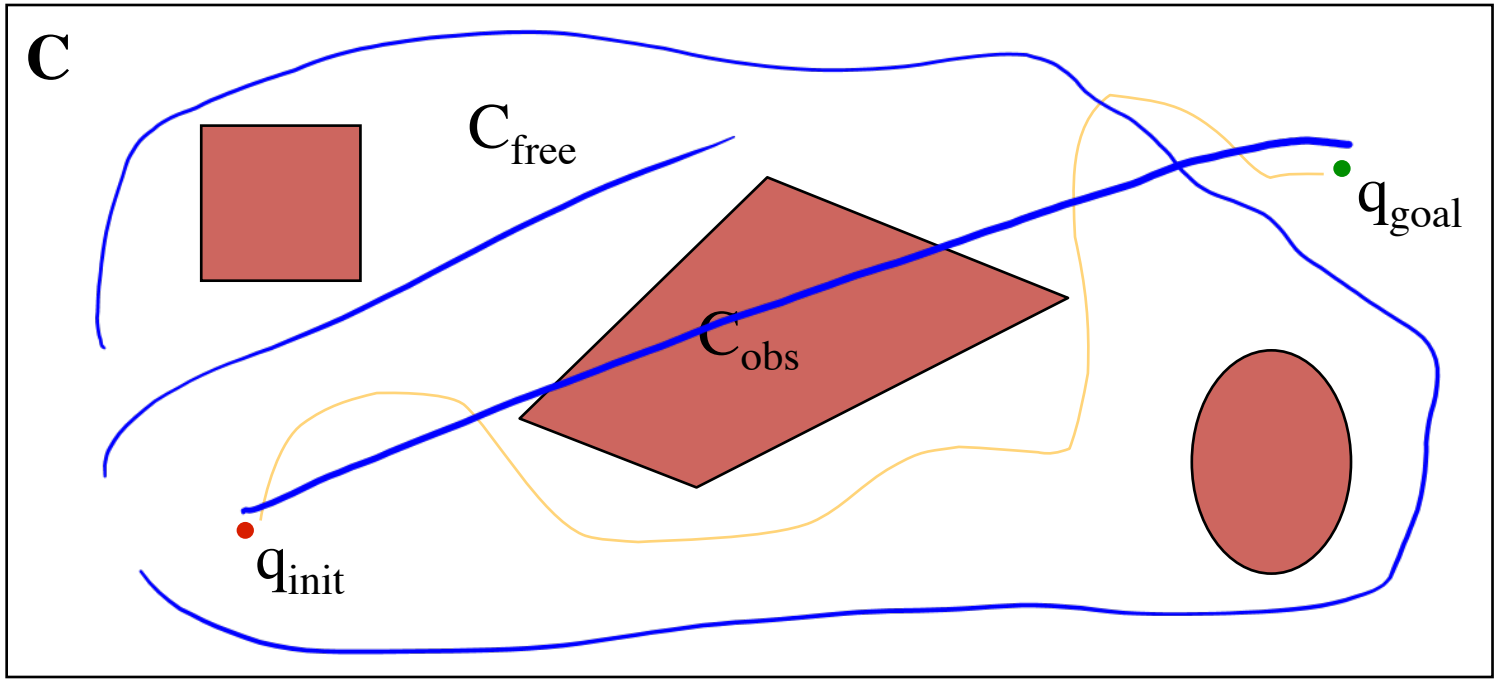


Configuration Space (C-Space)

- **Configuration Space:** the space of all possible robot configurations.
  - Data structure that allows us to represent occupied and free space in the environment

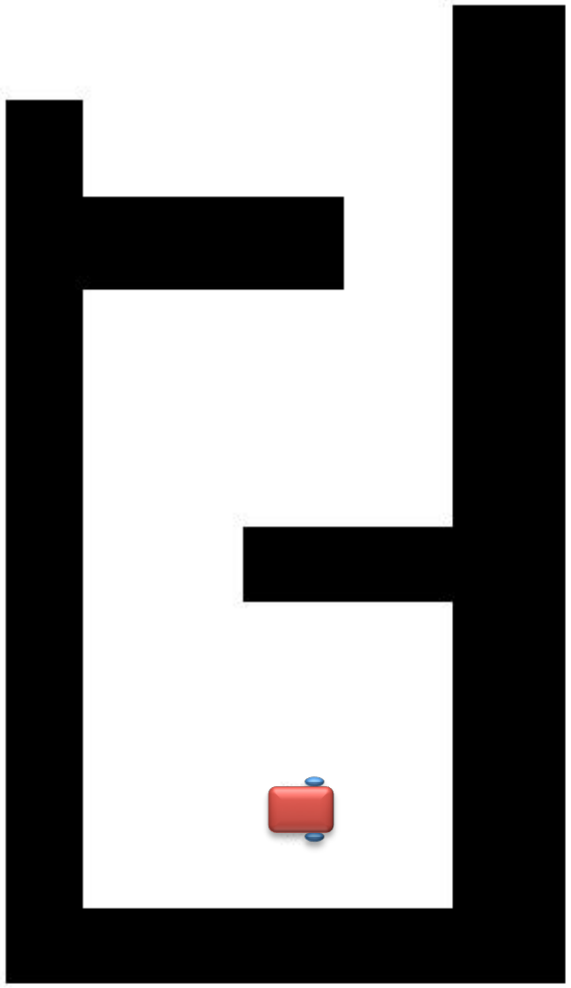
Configuration Space

Point robot (no constraints)

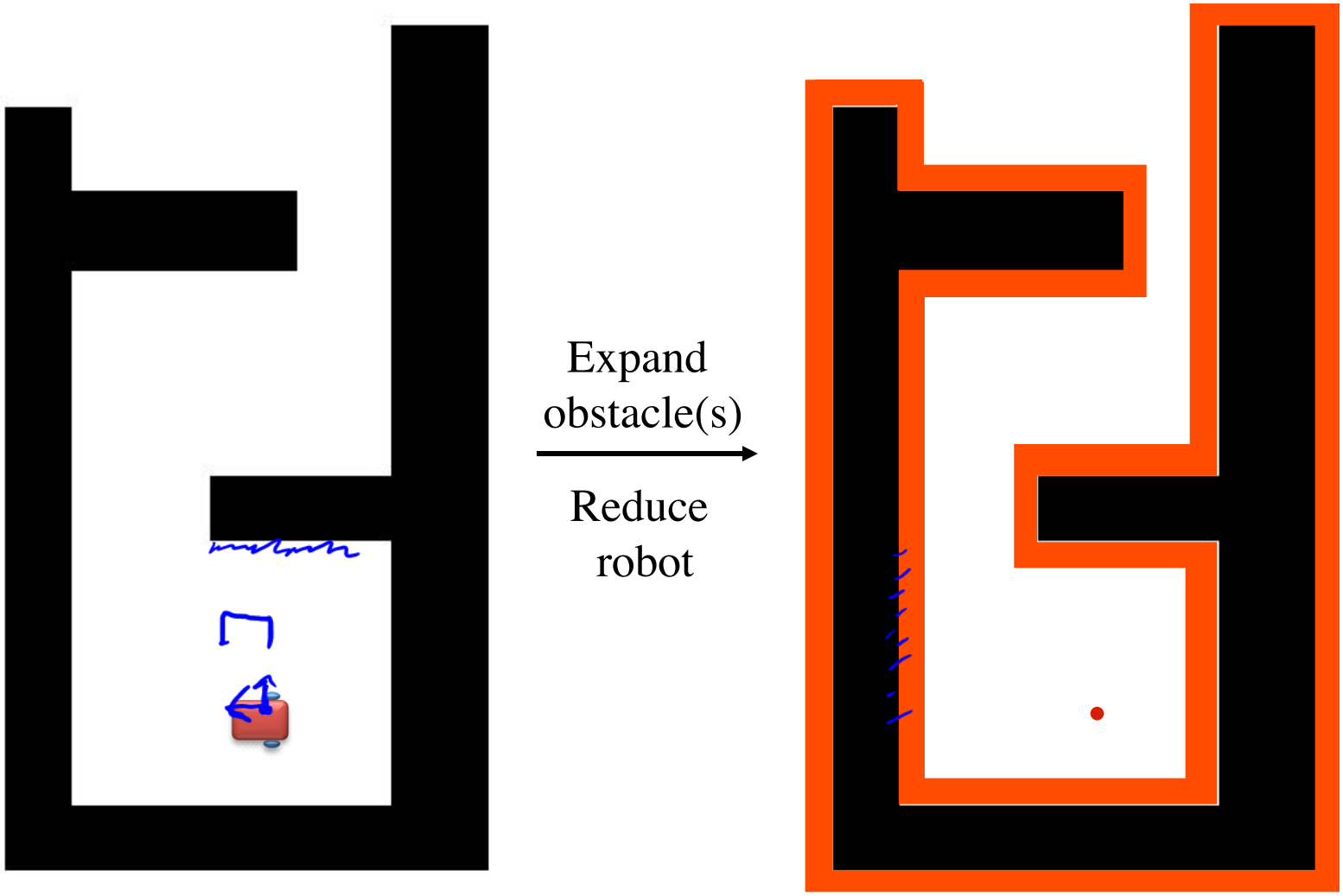


For a point robot moving in 2-D plane, C-space is R^2

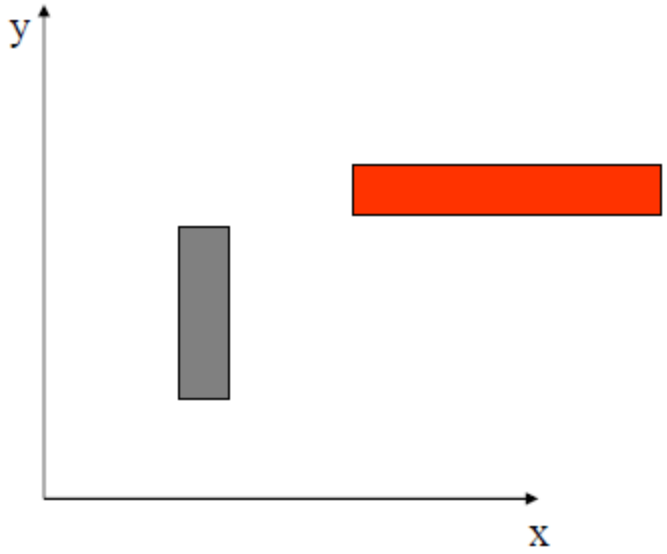
What if the robot is not a point?



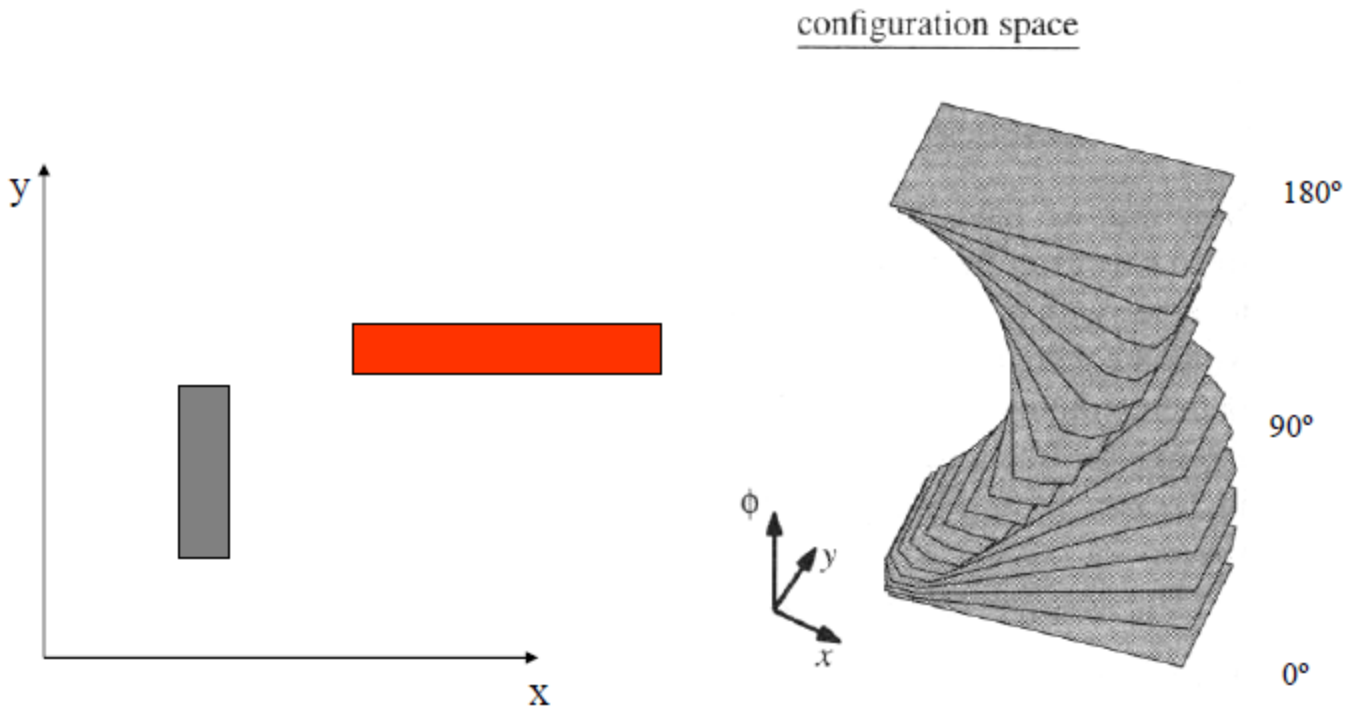
What if the robot is not a point?



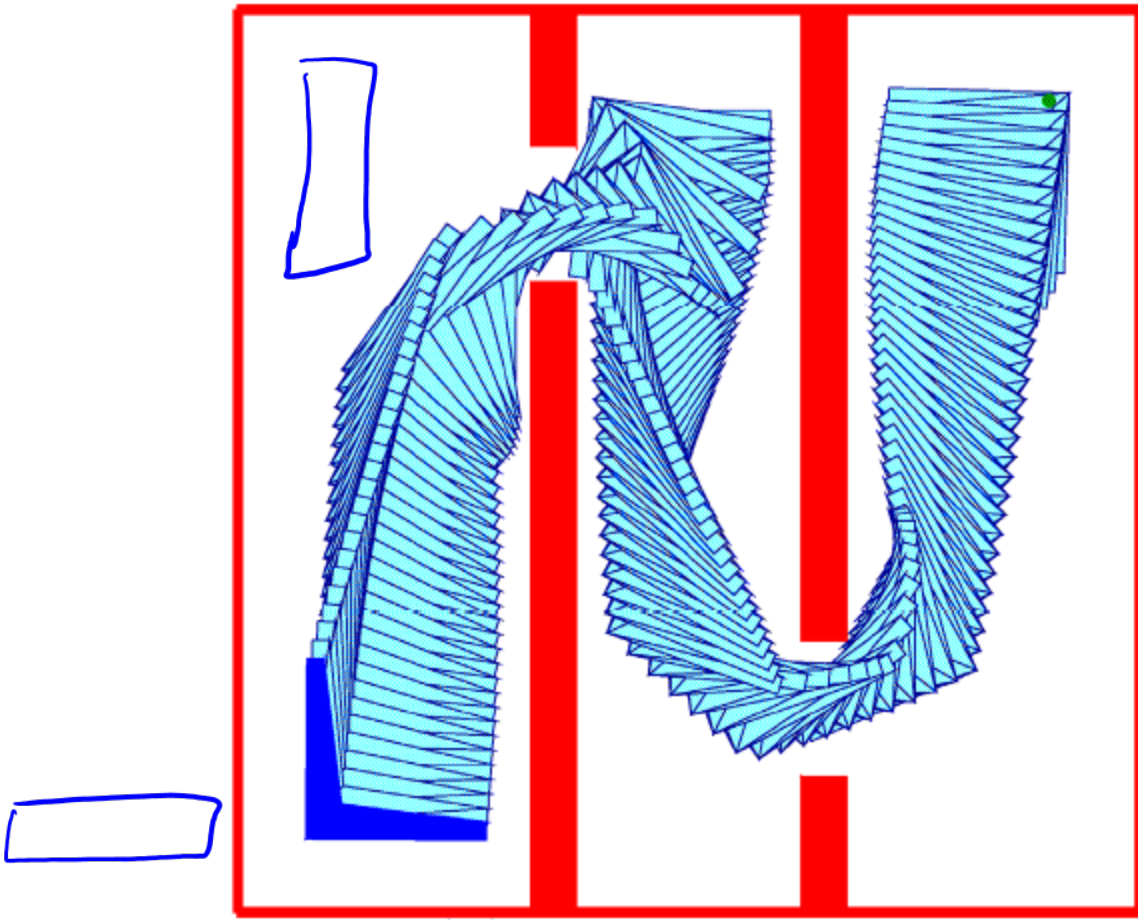
What if we want to preserve the angular component?



If we want to preserve the angular component…



Rigid Body Planning



Transfer in Reinforcement Learning via Shared Features: Konidaris, Scheidwasser, and Barto, 2012

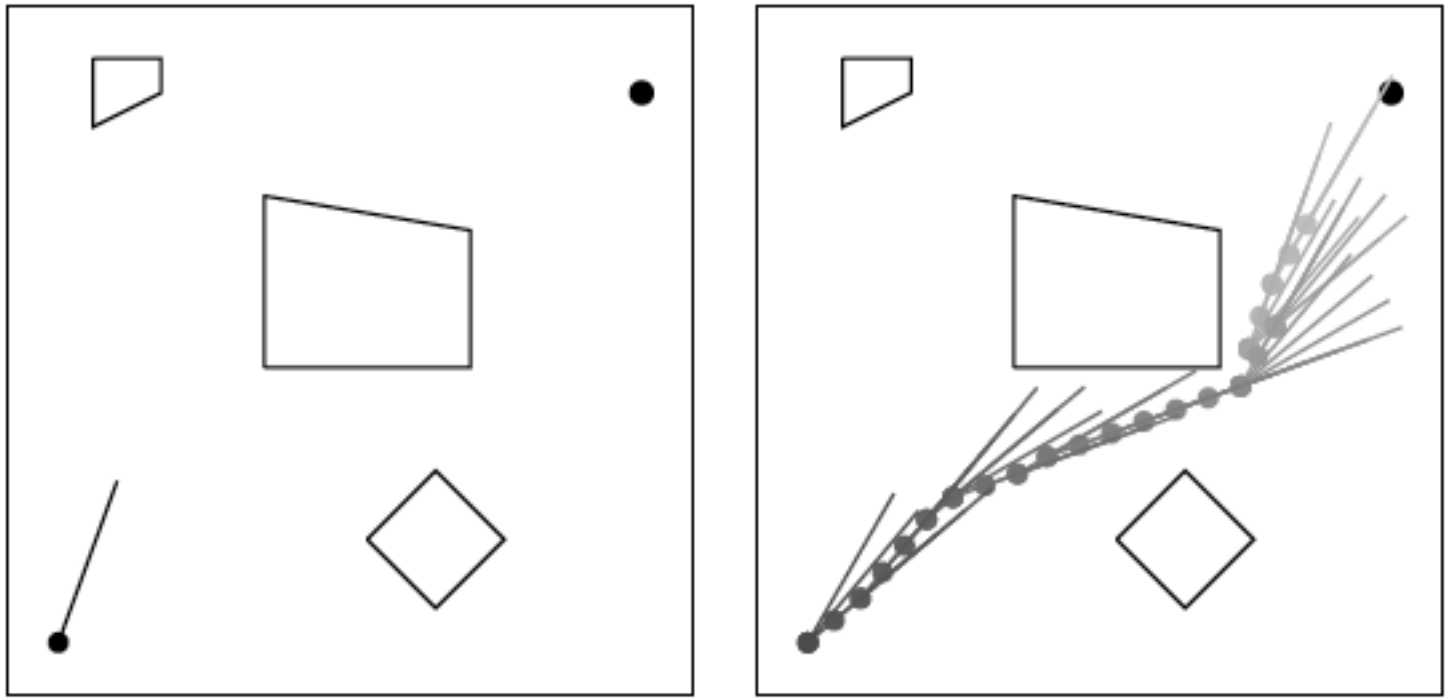


Figure 1: A 20x20 rod positioning task and one possible solution path.

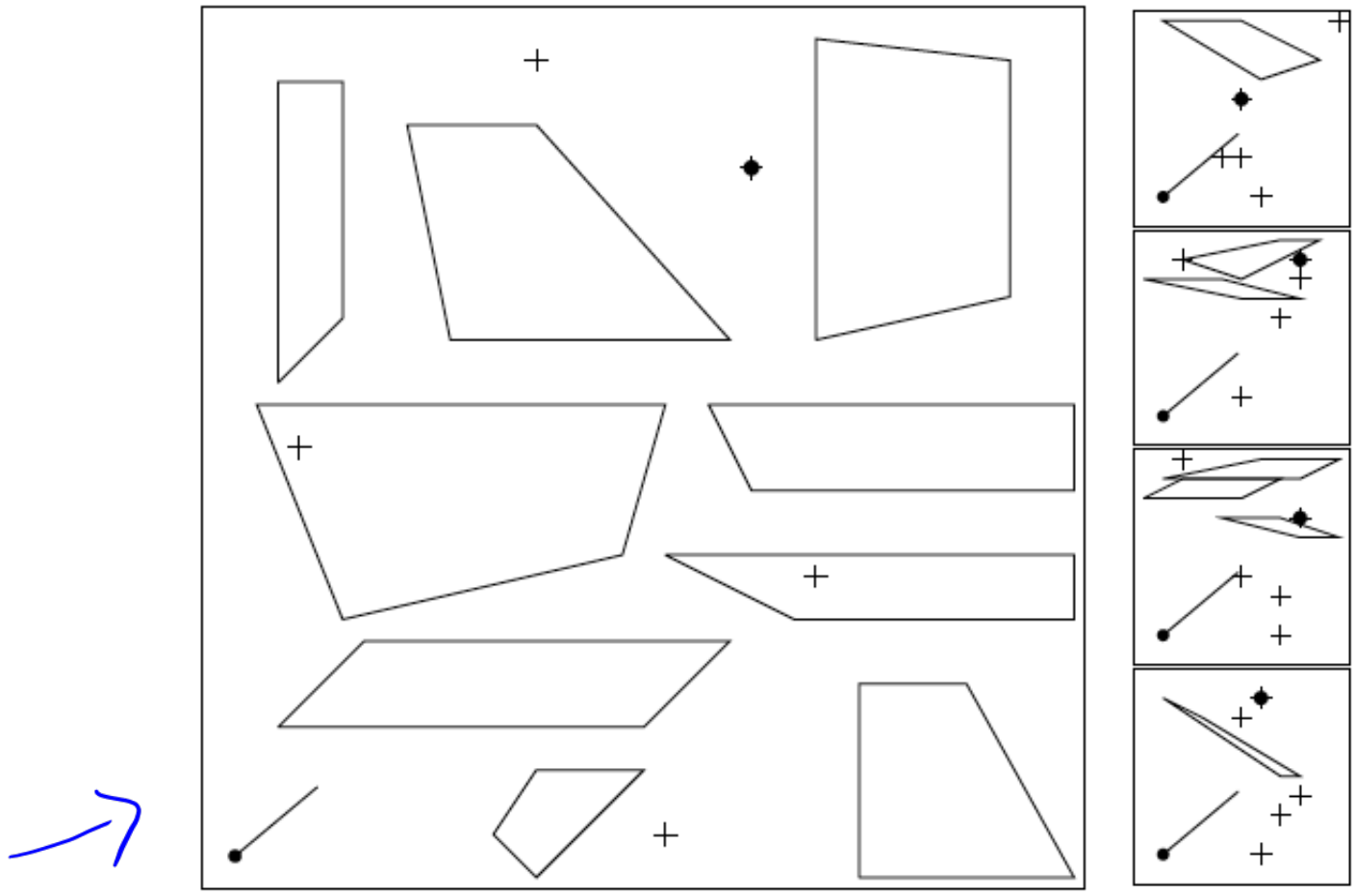
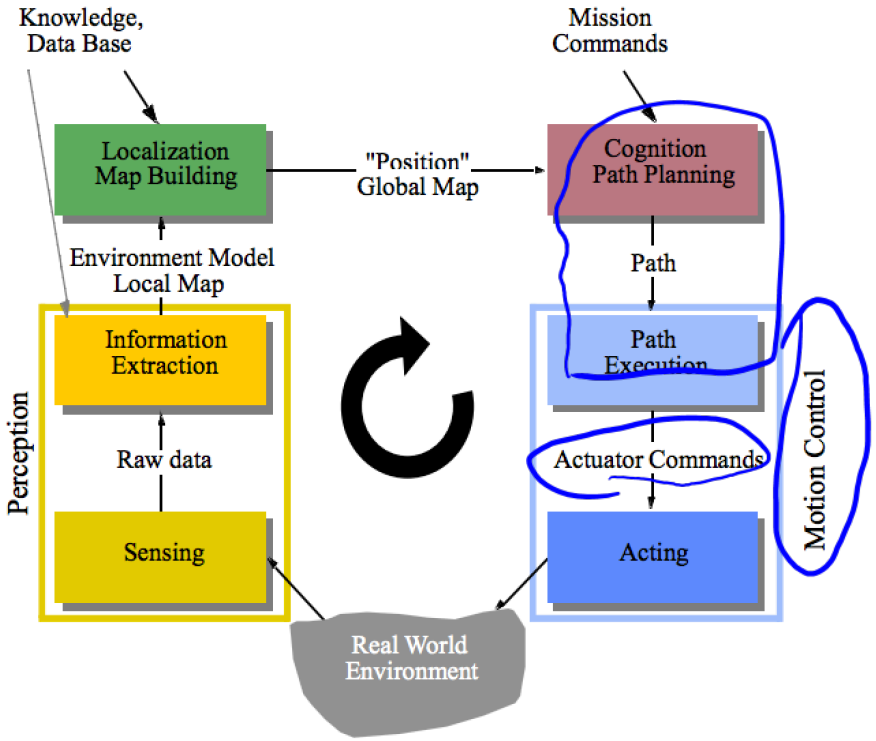


Figure 2: The homing experiment 40x40 test task (a) and four sample 10x10 training tasks (b). Beacon locations are shown as crosses, and the goal is shown as a large dot. Note that the first beacon is on the target in each task. The optimal solution for the test task requires 69 steps.

- Open loop control vs. Closed loop control

- Recap
  - Control Architectures
  - Sensors & Vision
  - Control & Kinematics

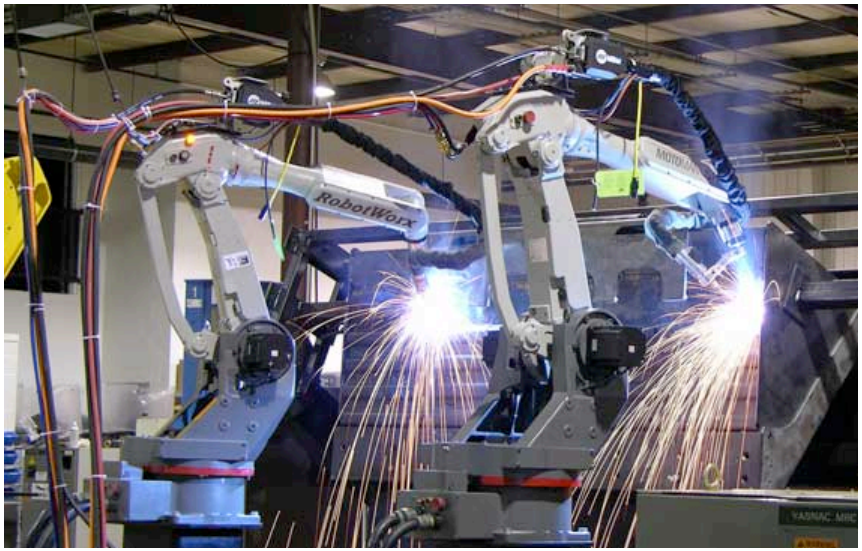


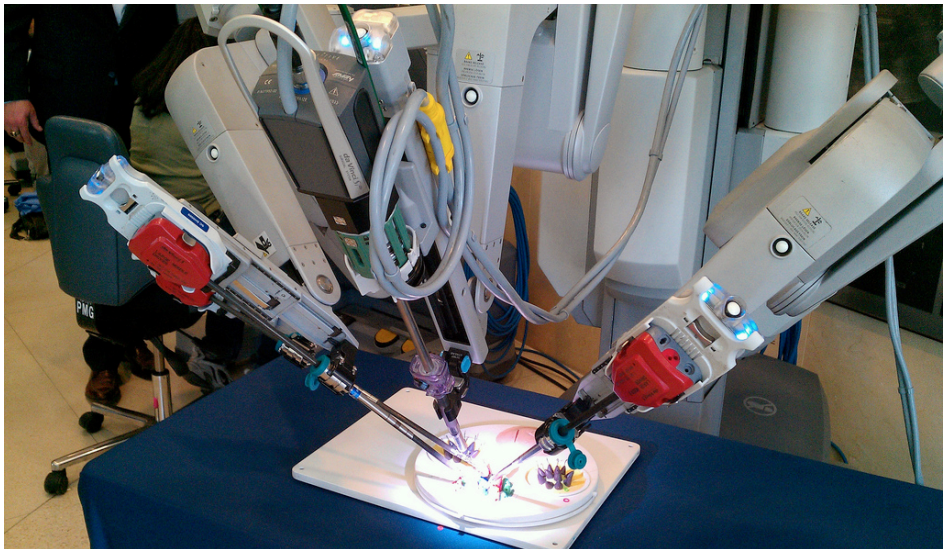
Implementation of a Gaussian process-based machine learning grasp predictor.

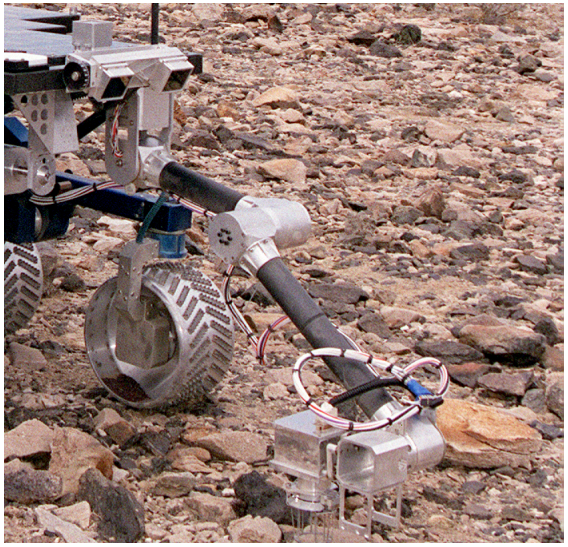
Goins, Carpenter, Wong, & Balasubramanian, 2015

With the goal of advancing the state of automatic robotic grasping, we present a novel approach that combines machine learning techniques and physical validation on a robotic platform to develop a comprehensive grasp predictor. After collecting a large grasp sample set (522 grasps), we first conduct a statistical analysis of the predictive ability of grasp quality metrics that are commonly used in the robotics literature. We then apply principal component analysis and Gaussian process (GP) algorithms on the grasp metrics that are discriminative to build a classifier, validate its performance, and compare the results to existing grasp planners. The key findings are as follows: (i) several of the existing grasp metrics are weak predictors of grasp quality when implemented on a robotic platform; (ii) the GP-based classifier significantly improves grasp prediction by combining multiple grasp metrics to increase true positive classification at low false positive rates; (iii) The GP classifier can be used generate new grasps to improve bad grasp samples by performing a local search to find neighboring grasps which have improved contact points and higher success rate.

Manipulators

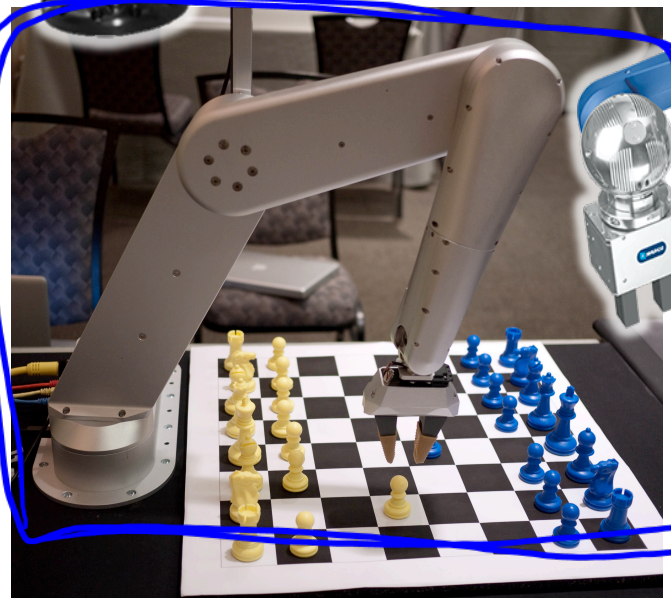




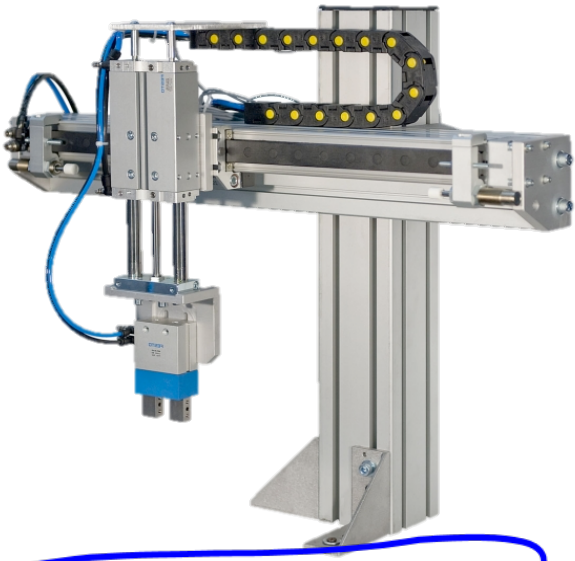


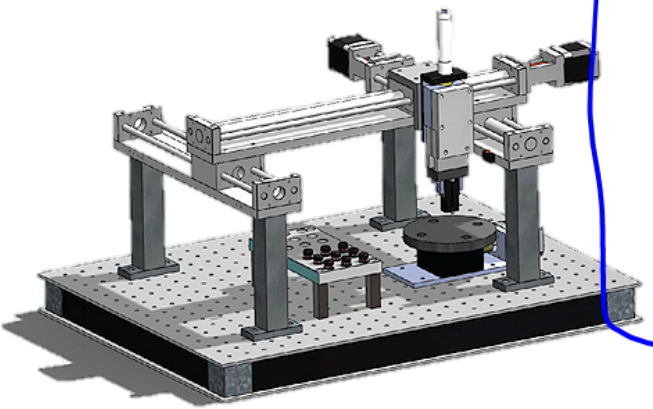
Manipulators

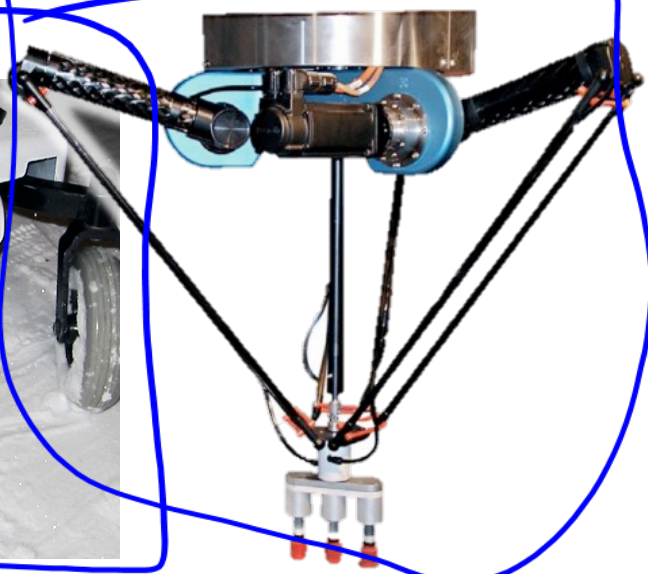



Manipulators

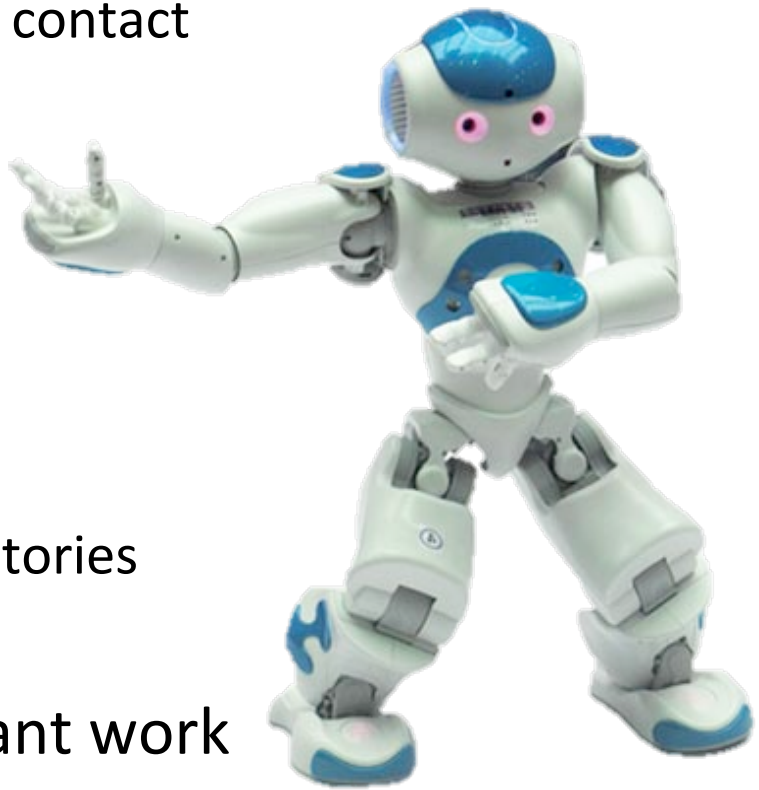






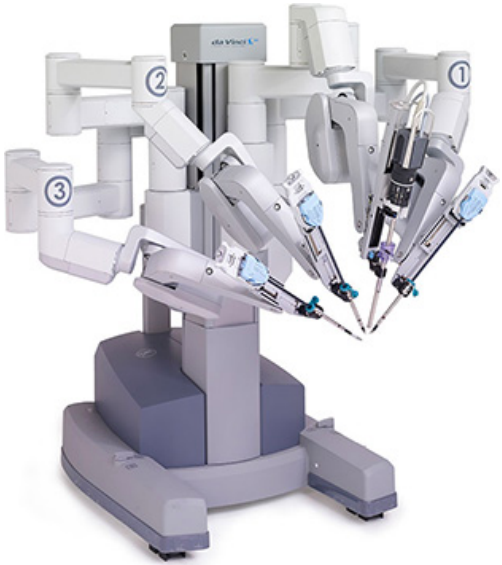
Manipulation

- What is a manipulator?
  - Manipulates something in the world
    - Physically alter the world through contact
    - As a primary goal
    - But not its own position

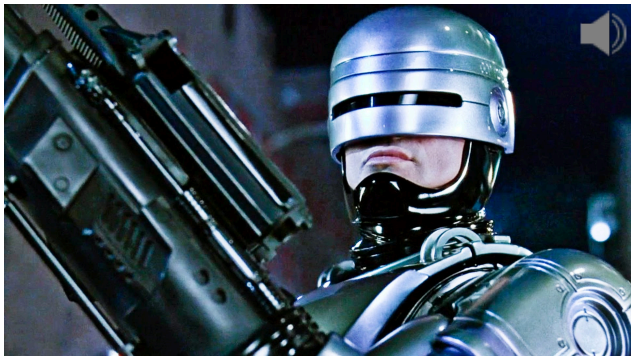


- When is this desirable?
  - Dangerous workspaces
    - Space; foundries; underwater; factories
  - Dirty
  - Dull Boring, repetitive, unpleasant work
  - Human-intractable workspaces
    - Too small; too big; too much precision needed

- Current
  - Industrial
    - Welding
    - Drilling
    - Attaching (screws, rivets)
    - Painting
    - Loading/unloading
  - Surgery
  - Space exploration
  - Chores
  - Patient care
  - Delivery



Davinci Robot



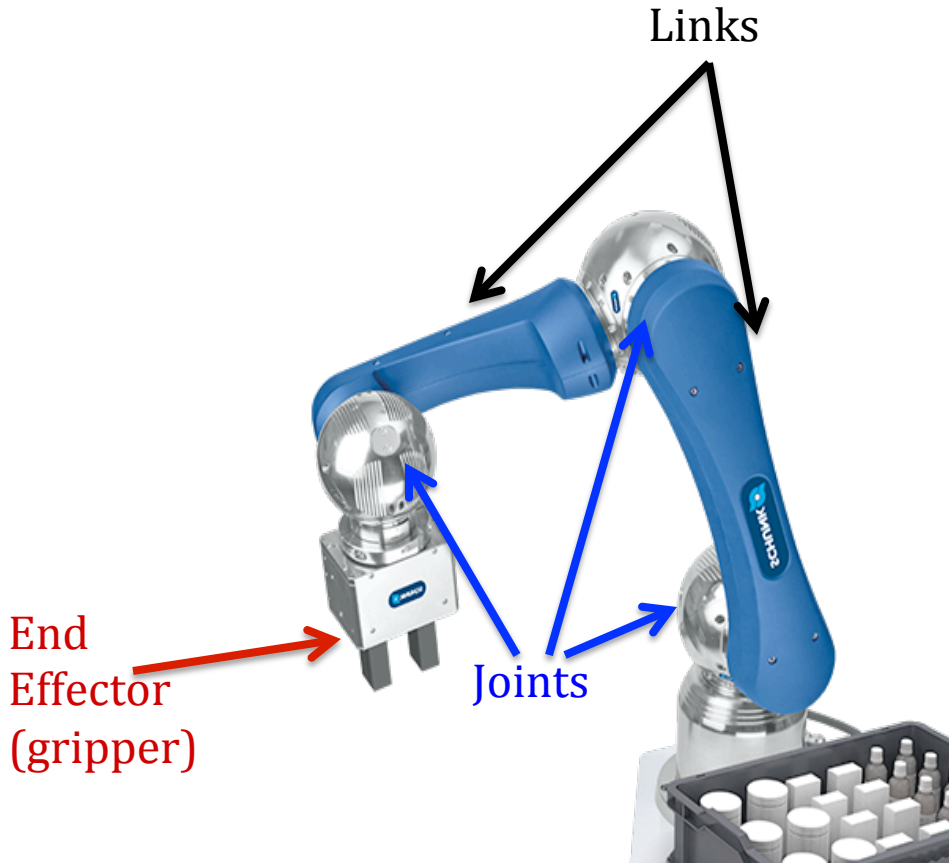
- Future
  - Elder care
  - Entertainment
  - Environment sampling
  - Compliant-material interactions (sewing)
  - Battery changing
  - Police work
- Plus: more chores, more patient care, more surgery, more space, &c. (but better)

Terminology

- Actuator
  - Generates motion or force; usually a motor
  - “Drive type”
- DoFs
- End Effector
  - Device at the end of an arm; interacts with environment
- Gripper
  - What it sounds like: a type of manipulator

Manipulator Robot

- Modeled as a chain of *rigid links* connected by *joints*
- Links: unjointed length of robot
- Joints: translational or rotational movement
  - Joints have DoFs
  - Sliding or jointed
- Manipulator / End Effectors
  - Grippers / Tools
  - Sensors

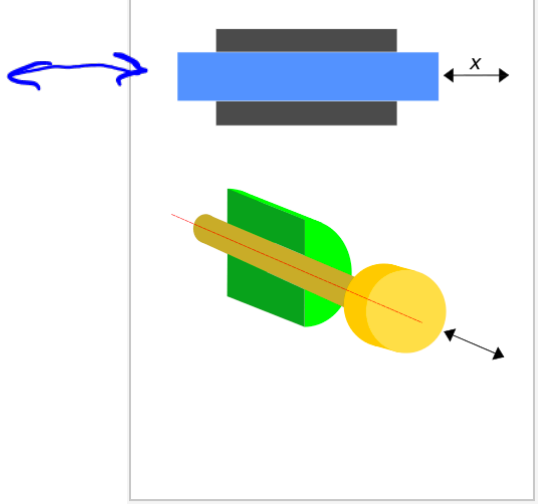


Characterization

- Types
  - By drive
  - By actuation
    - Tendons
    - Direct servoing
    - Underactuation
  - By motion
    - Prismatic (linear)
    - Revolute (rotational)
  - By Characteristics
    - Payload
    - Working area/radius

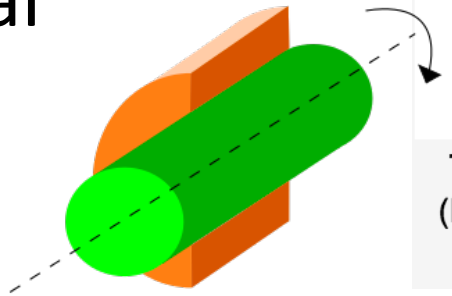
Joints

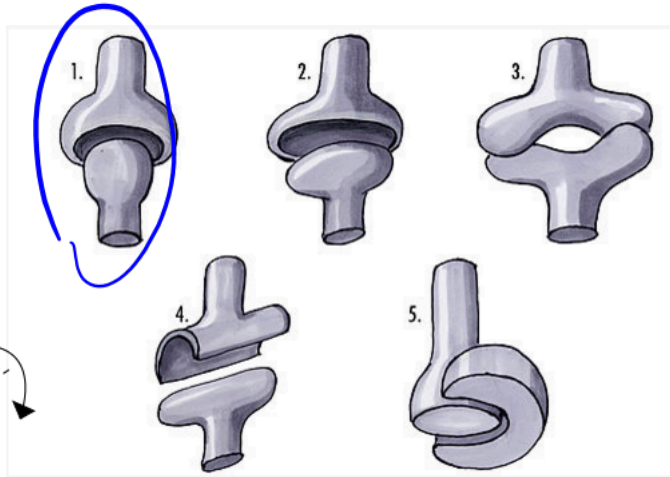
- Prismatic: sliding / translational



Prismatic joint seen in 2-dimensional form, noting that the joint may only move in one direction. Bottom shows a sample piston cylinder cutout which utilizes a prismatic joint.

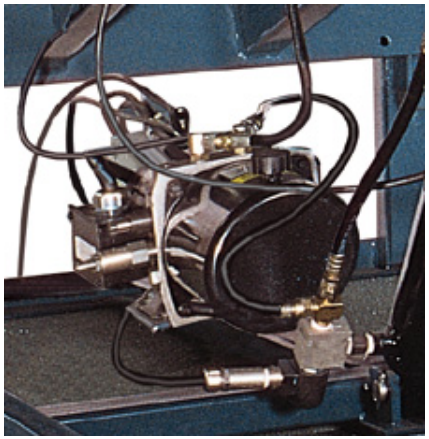
- Revolute: rotational





1: Ball and socket joint; **2**: Condyloid joint (Ellipsoid); **3**: Saddle joint; **4** Hinge joint; **5**: Pivot joint;

Actuators



Hydraulic Motor

Pneumatic Cylinder

Stepper Motor

Pneumatic Motor

DC Motor

Servo Motor

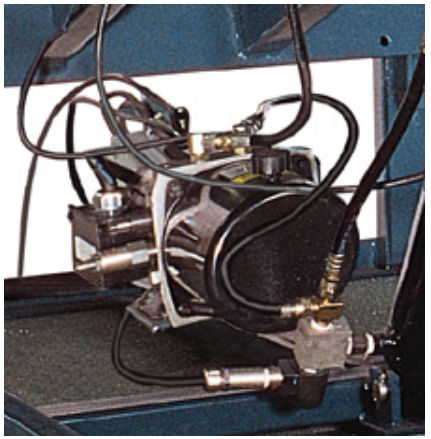
- DC vs. servo motor
  - https://www.youtube.com/watch?v=mSowgNx5u7I
- Pneumatic vs. Hydraulic
  - http://machinedesign.com/linear-motion/what-s-difference-between-pneumatic-hydraulic-and-electrical-actuators

When Do We Use…

- Hydraulic/pneumatic
  - Heavy loads, high speeds
  - Sometimes hard to control (esp. pneumatic)
  - Doesn't produce sparks

Pneumatic Motor

Pneumatic Cylinder



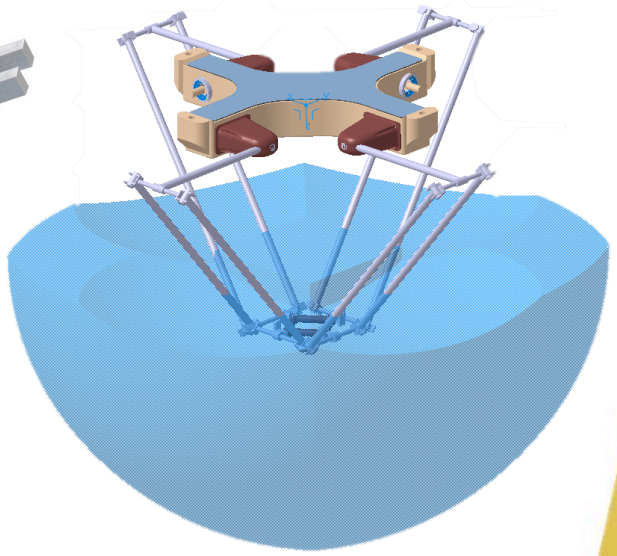
Hydraulic Motor

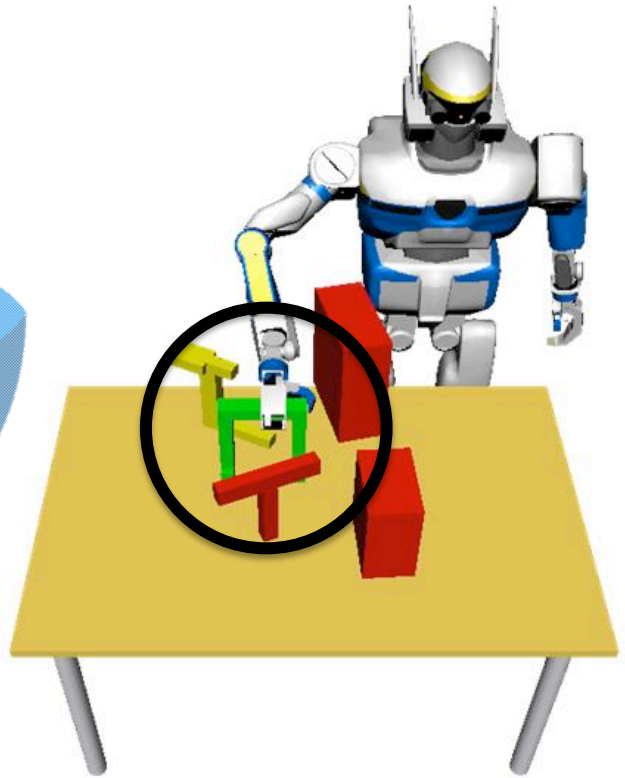
When Do We Use…

- Most common robotic actuators: combinations of different electro-mechanical devices
  - Stepper motor
    - Subdivides a rotation into # increment
    - Open Loop
  - Servo Motor
    - Subdivides a rotation arbitrarily
    - Closed Loop

Manipulation

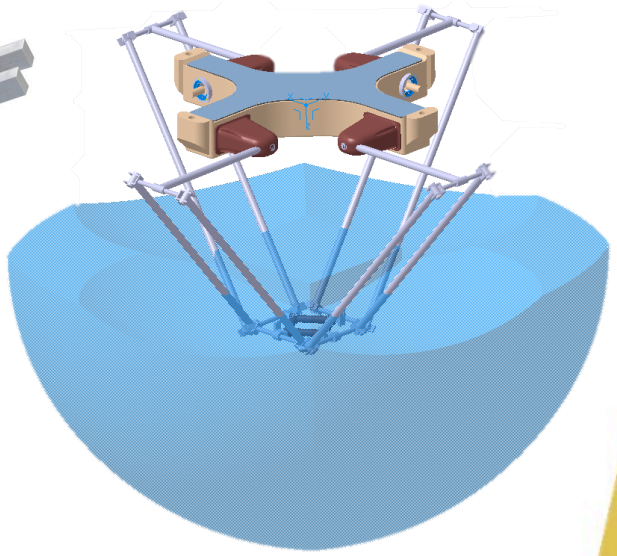
Configurations & Grasping

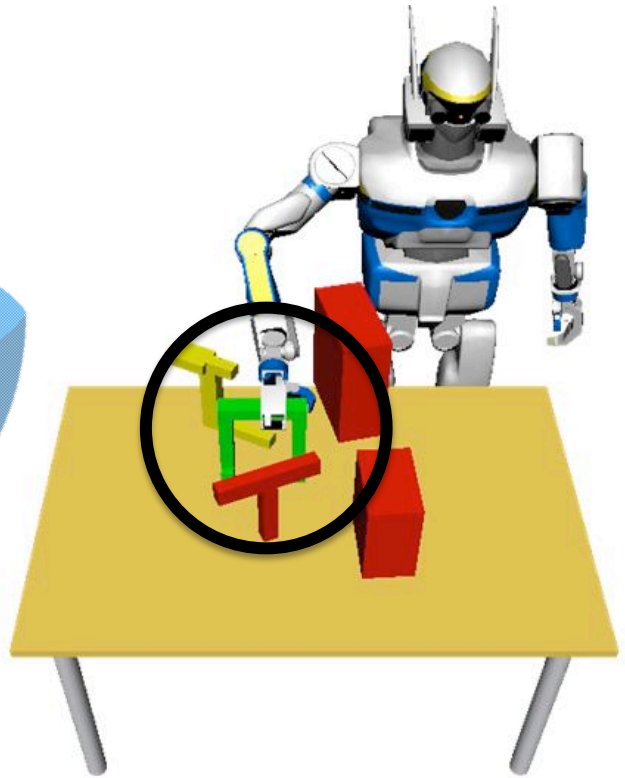




Manipulation

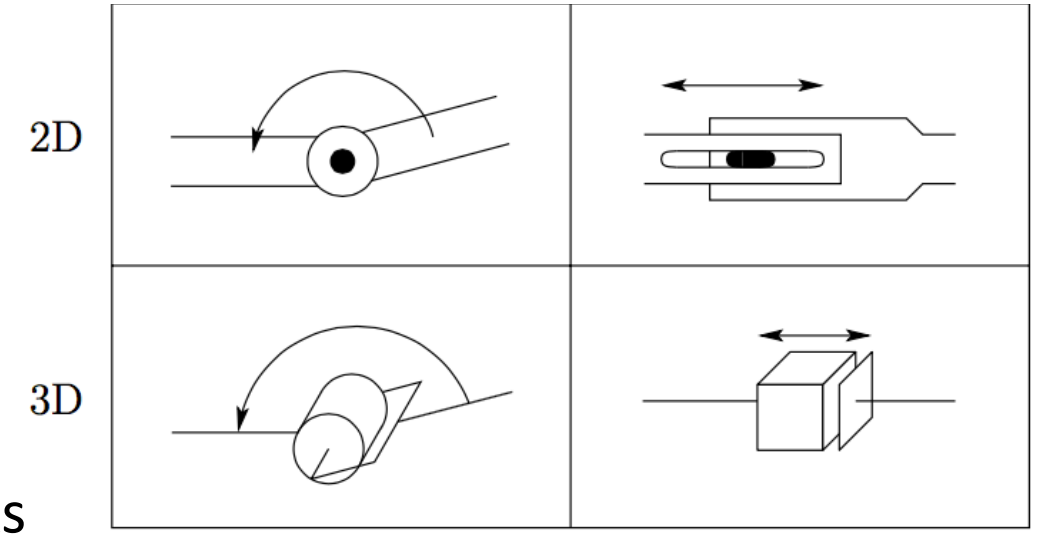
Configurations & Grasping





P(rismatic) & R(evolute)

- A kinematic chain of rigid links connected by joints
  - *"**Kinematics** is the branch of classical mechanics which describes the motion of objects and groups of objects."*
- Prismatic (denoted P)
  - Sliding / translational / linear; allows a linear relative motion between 2 links
- Revolute (denoted R)
  - Rotational; allows relative rotation between two links



Spong, Hutchinson, Vidyasagar. Robot Modeling and Control. 2006.

Joints: Denotation

- A joint represents a connection between two links
- Denotation of relative displacement between links
  - θ for revolute joint
  - d for prismatic joint
- Denotation of axis of motion
  - z_i between link i and link $i+1$
    - Axis of rotation of a revolute joint
    - Axis of translation of a prismatic joint

Spong, Hutchinson, Vidyasagar. Robot Modeling and Control. 2006.

Configuration Space

- **Configuration**
  - Specification of location of every point on manipulator.
- How can we specify?
  - Links are rigid
  - Base is (assumed to be) fixed
  - So if we know values for the joint variables
    - Angle for R joints (θ), offset for P joints (d)
  - We know everything!
- Manip. configuration ≡ a set of values for joint variables
- Set of all possible configurations is the ***configuration space.***

Spong, Hutchinson, Vidyasagar. Robot Modeling and Control. 2006.

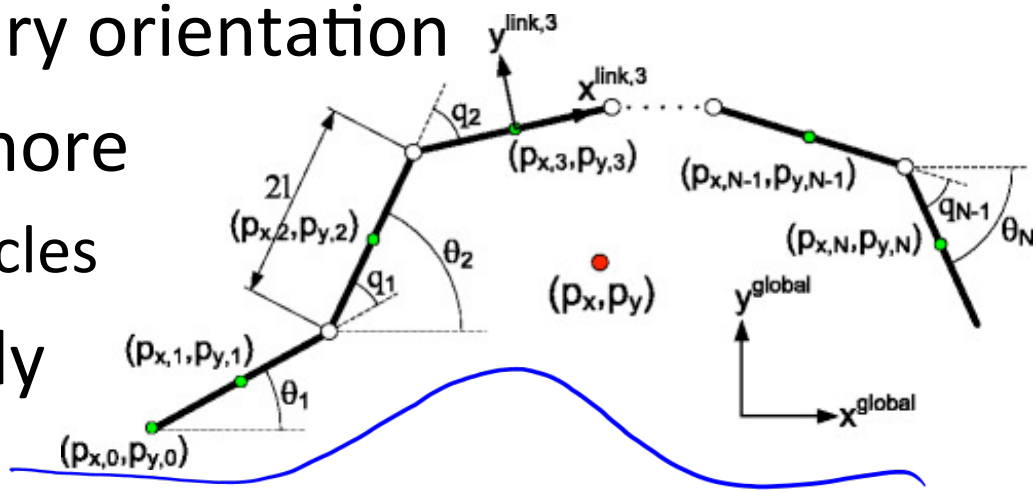
DoFs for Manipulation

- A system has n DoFs if exactly n parameters are required to completely specify the configuration.
- For a manipulator:
  - Configuration can be specified by n joint parameters, so
  - # of DoFs = dimension of the configuration space
  - So, # number of joints determines DoFs
- Rigid object in 3D space has six parameters
  - 3 positioning (x, y, z), 3 orientation (roll, pitch and yaw angles)
- DoFs < 6 ⇒ arm cannot reach every point in workspace with arbitrary orientation.

Spong, Hutchinson, Vidyasagar. Robot Modeling and Control. 2006.

Notes on DoFs

- DoFs < 6 ⇒ arm cannot reach every point in workspace with arbitrary orientation
- Sometimes you need more
  - E.g., dealing with obstacles
- DoFs > 6 is kinematically redundant

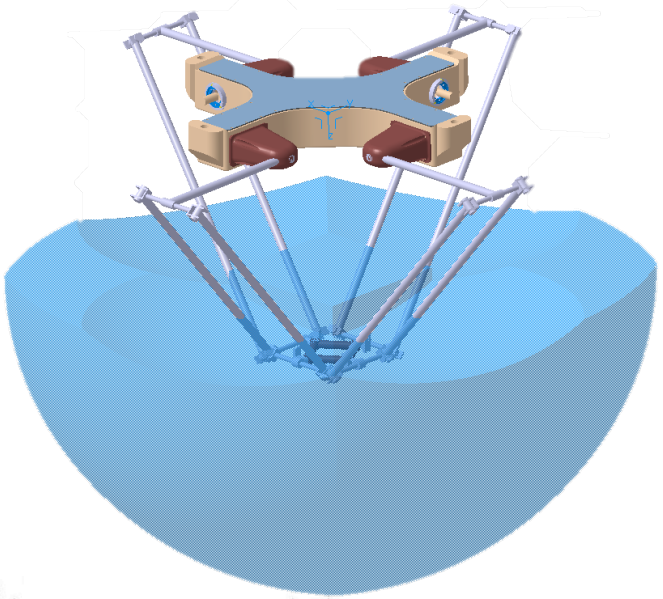


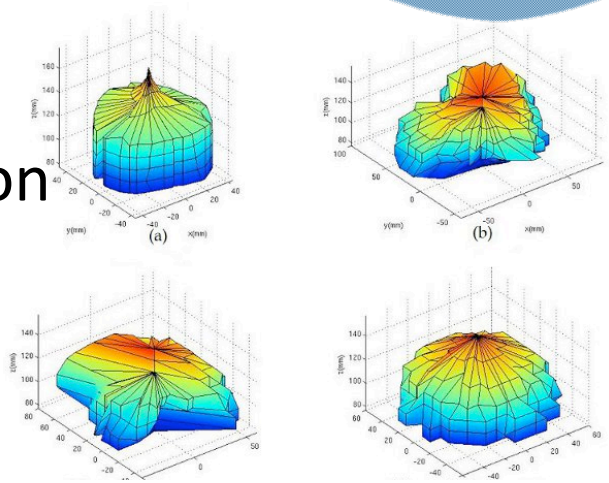
- Difficulty of control problem as # DoFs grows
  - Increases *rapidly* with the number of links
  - Every 2 links need a joint

Spong, Hutchinson, Vidyasagar. Robot Modeling and Control. 2006.
Ehsan Rezapour. Pettersen, Gravdahl, Liljebäck, Kelasidi. Robotics and Biomimetics. 2014.

Workspaces

- Configuration only provides geometry
- **Workspace**
  - Set of all possible positions of end effector
  - In practice, these can be complex
- **Dexterous workspace**
  - Set of points where end effector can be any orientation
  - Subset of workspace





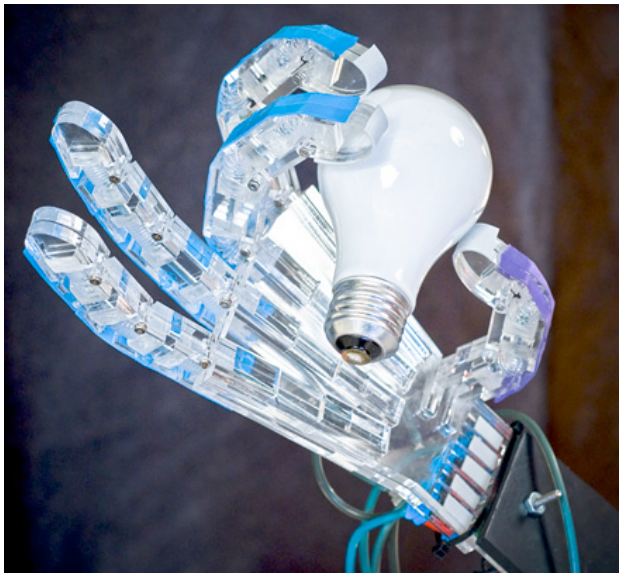
Spong, Hutchinson, Vidyasagar. Robot Modeling and Control. 2006.
engineerjau.wordpress.com/2013/07/07/on-the-basis-of-workspaces-of-robotic-manipulators-part-1

Grippers

- Four categories of robot grippers:
  - Impactive
    - Jaws or claws which physically grasp by direct impact upon the object

grasping

  - Ingressive
    - Pins, needles or hackles penetrate surface
      - Textile, carbon and fiberglass handling
  - Astrictive
    - Suction forces applied to surface
    - Vacuum, magneto- or electroadhesion
  - Kontugutive / Contigutive
    - Requiring direct contact for adhesion
    - Glue, surface tension or freezing



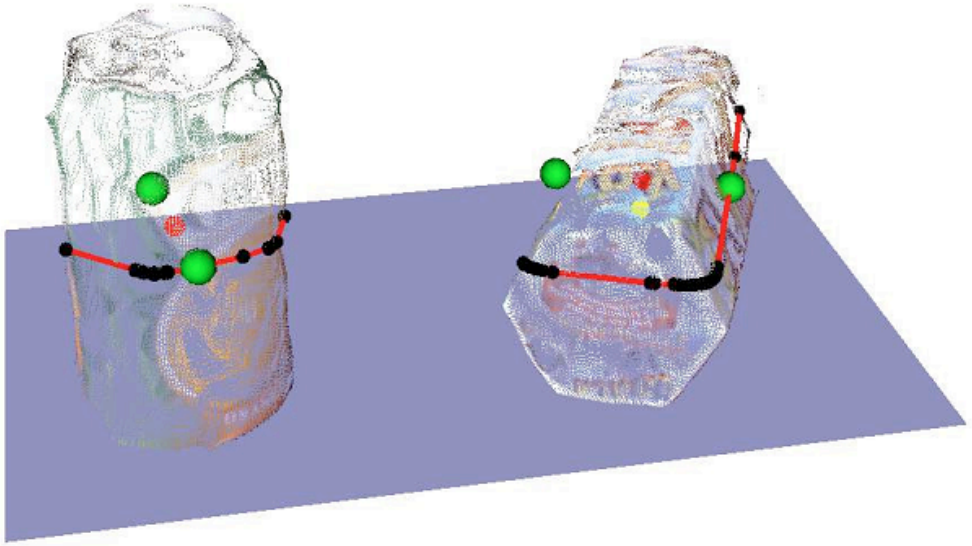
Monkman, Hesse, Steinmann, Schunk. Robot Grippers. 2007.
news.nationalgeographic.com/news/2009/05/090505-robot-hand-picture.html

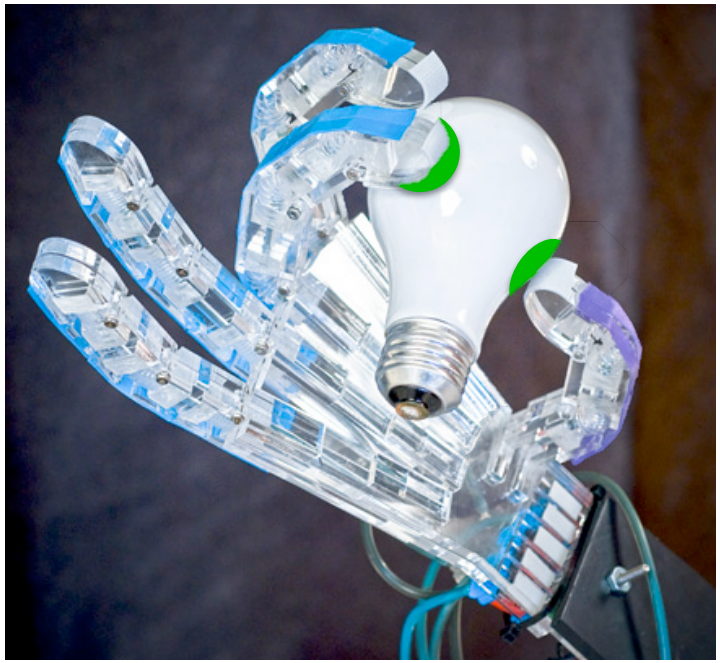
Universal Gripper: Video

- https://www.youtube.com/watch?v=0d4f8fEysf8
- https://www.youtube.com/watch?v=KBTrtQQKl7o

Grasps

- Grasp:
  - A set of contact points on an object's surface
  - Goal: constrain object's movement

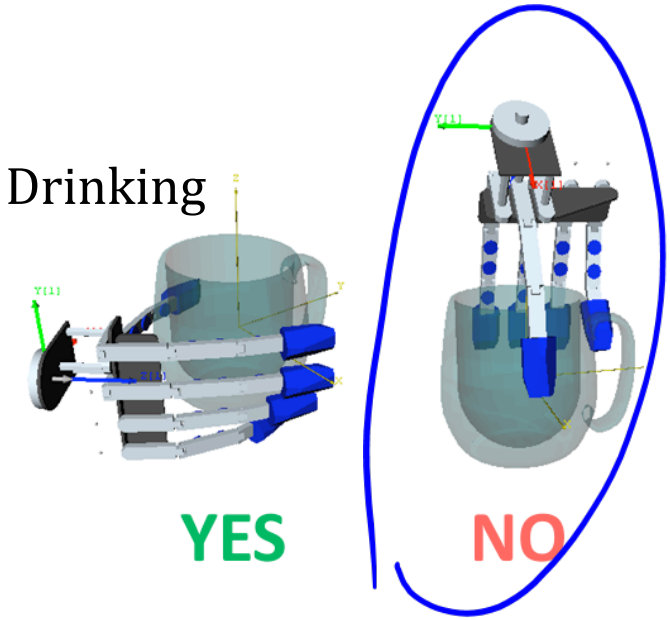




www.intechopen.com/books/robot-arms/robotic-grasping-of-unknown-objects1
news.nationalgeographic.com/news/2009/05/090505-robot-hand-picture.html

Grasps

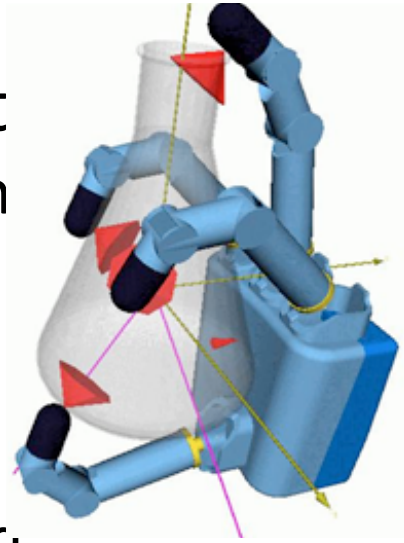
- Grasps vary by:
  - Hand (gripper)
  - Object being grasped
  - Type of motion desired
- For each hand or hand/object pair:
  - Where to grasp it?
  - How hard?
  - Then what?
- Additional constraints (e.g., don't spill)



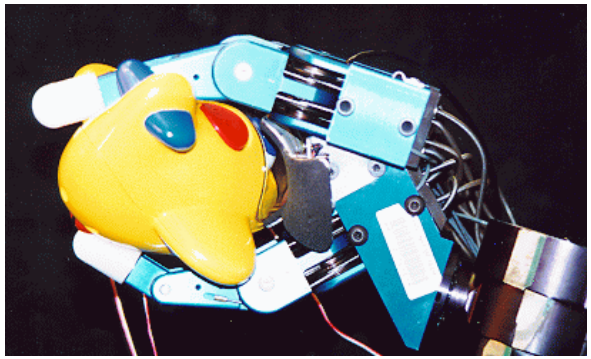
León, Morales, Sancho-Bru. Robot Grasping Foundations. 2013

The Grasping Problem

- Grasps are not obvious (easy to calculat
  - Any given object has arbitrary contact poin
  - Hand has geometry constraints, etc.
- Synthesized trial-and-error
  - For a hand/object pair:
  - Different grasp types planned and analyzed



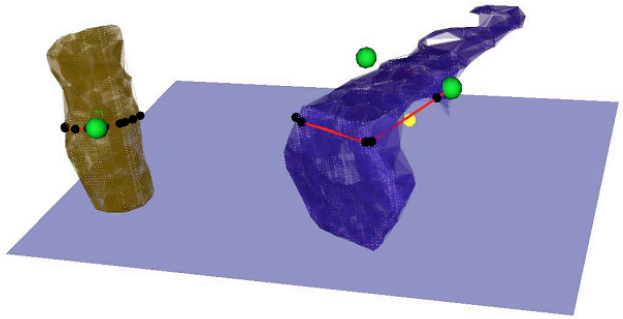
- Real trial and error

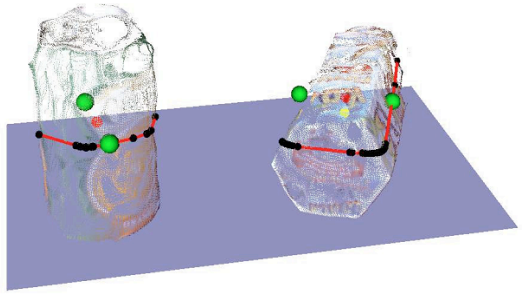


www.cs.columbia.edu/~cmatei/graspit/
www.programmingvision.com/research.html
www.cc.gatech.edu/gvu/people/faculty/nancy.pollard/grasp.html

Grasp Planning

- **Grasp synthesis:** Find suitable set of contacts, given
  - Object model
  - Constraints on allowable contacts
- **Grasp points** are determined
  - Mostly assume point **contacts**
  - Larger areas usually discretized
  - **Contact model** defines the force the manipulator exerts on contact areas
- **Grasp analysis**
  - Is that grasp stable?

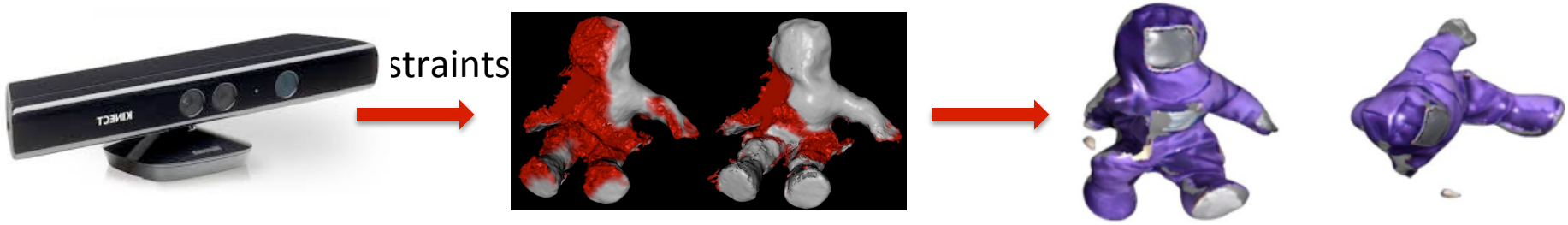




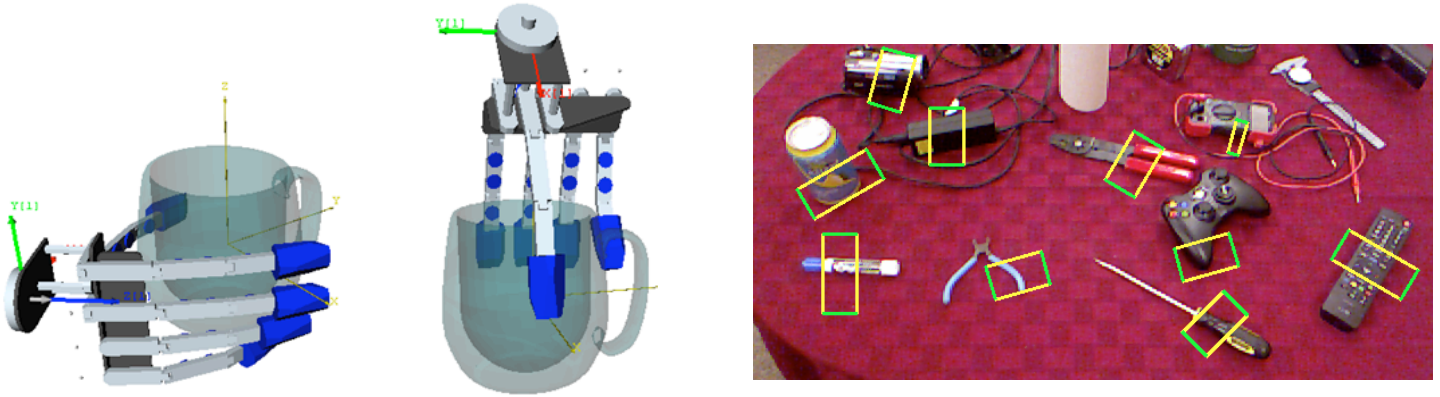
León, Morales, Sancho-Bru. Robot Grasping Foundations. 2013.
www.intechopen.com/books/robot-arms/robotic-grasping-of-unknown-objects1

Ongoing Research

- How do you get the object model?



- Background knowledge, mathematical modeling, …



www.madry.pro
www.cs.washington.edu/robotics/3d-in-hand

- https://www.youtube.com/watch?v=rJATi2ymAnY

Implementation of a Gaussian process-based machine learning grasp predictor.

Goins, Carpenter, Wong, & Balasubramanian, 2015

With the goal of advancing the state of automatic robotic grasping, we present a novel approach that combines machine learning techniques and physical validation on a robotic platform to develop a comprehensive grasp predictor. After collecting a large grasp sample set (522 grasps), we first conduct a statistical analysis of the predictive ability of grasp quality metrics that are commonly used in the robotics literature. We then apply principal component analysis and Gaussian process (GP) algorithms on the grasp metrics that are discriminative to build a classifier, validate its performance, and compare the results to existing grasp planners. The key findings are as follows: (i) several of the existing grasp metrics are weak predictors of grasp quality when implemented on a robotic platform; (ii) the GP-based classifier significantly improves grasp prediction by combining multiple grasp metrics to increase true positive classification at low false positive rates; (iii) The GP classifier can be used generate new grasps to improve bad grasp samples by performing a local search to find neighboring grasps which have improved contact points and higher success rate.