UNIVERSITY OF SOUTH CAROLINA

CSCE 790 MACHINE LEARNING IN ROBOTICS

A brief intro to robots

Spring 2018

Presentation: Marios Xanthidis

Ioannis Rekleitis

Robot Sensors

- Sensors are devices that can sense and measure physical properties of the environment,
  - e.g. temperature, luminance, resistance to touch, weight, size, etc.
  - The key phenomenon is transduction
    - Transduction (engineering) is a process that converts one type of energy to another
- They deliver *low-level* information about the environment the robot is working in.
  - Return an incomplete description of the world.

Robot Sensors

- This information is **noisy** (imprecise).
- Cannot be modelled completely:
  - Reading = *f*(env) where *f* is the model of the sensor
  - Finding the inverse:
    - ill posed problem (solution not uniquely defined)
    - collapsing of dimensionality leads to ambiguity

Types of sensor

- General classification:
  - **active** *versus* **passive**
    - Active: emit energy in environment
      - More robust, less efficient
    - Passive: passively receive energy from env.
      - Less intrusive, but depends on env. e.g. light for camera
    - Example: stereo vision versus range finder.
  - **contact** *versus* **non-contact**

Sensors

- **Proprioceptive Sensors**

  (monitor state of robot)
  - IMU (accels & gyros)
  - Wheel encoders
  - Doppler radar …

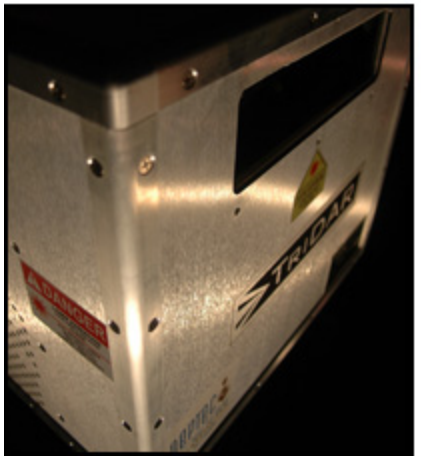
- **Exteroceptive Sensors**

  (monitor environment)
  - Cameras (single, stereo, omni, FLIR …)
  - Laser scanner
  - MW radar
  - Sonar
  - Tactile…

Types of sensor

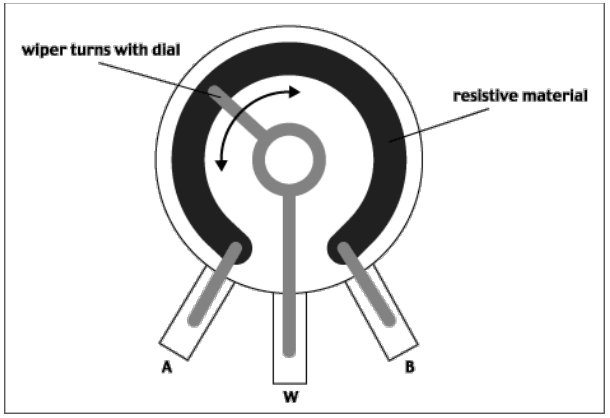
Specific examples

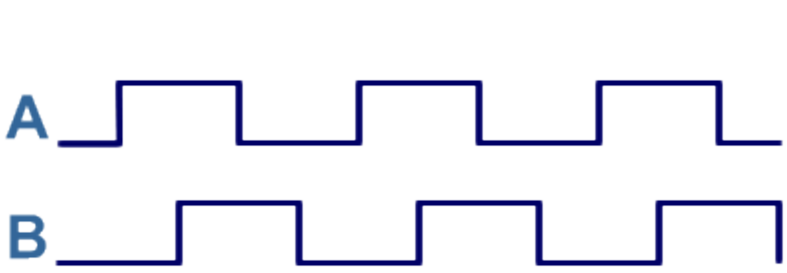
- tactile
- close-range proximity
- angular position
- infrared
- Sonar
- laser (various types)
- radar
- compasses, gyroscopes
- Force
- GPS
- vision

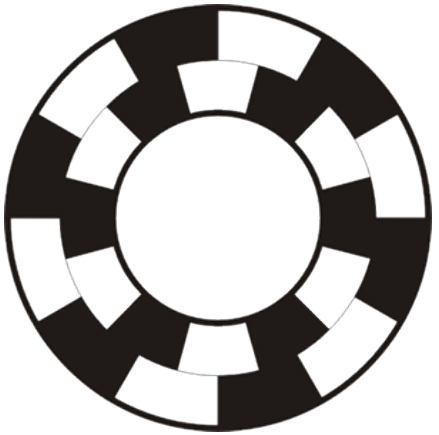


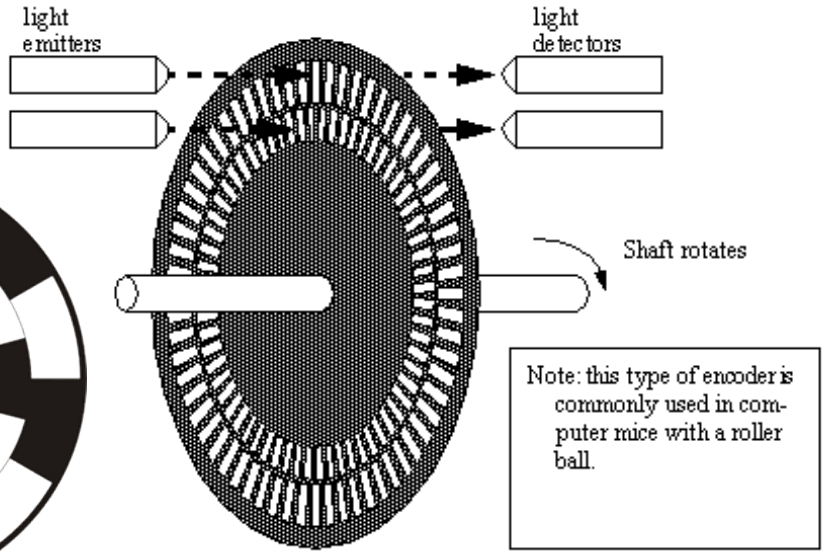
Angular Position: Rotary Encoder

- Potentiometer
  - Used in the Servo on the boebots
- Optical Disks (Relative)
  - Counting the slots
  - Direction by having pars of emitters/receivers out of phase: Quadrature decoding
  - Can spin very fast: 500 kHz









What is an actuator?

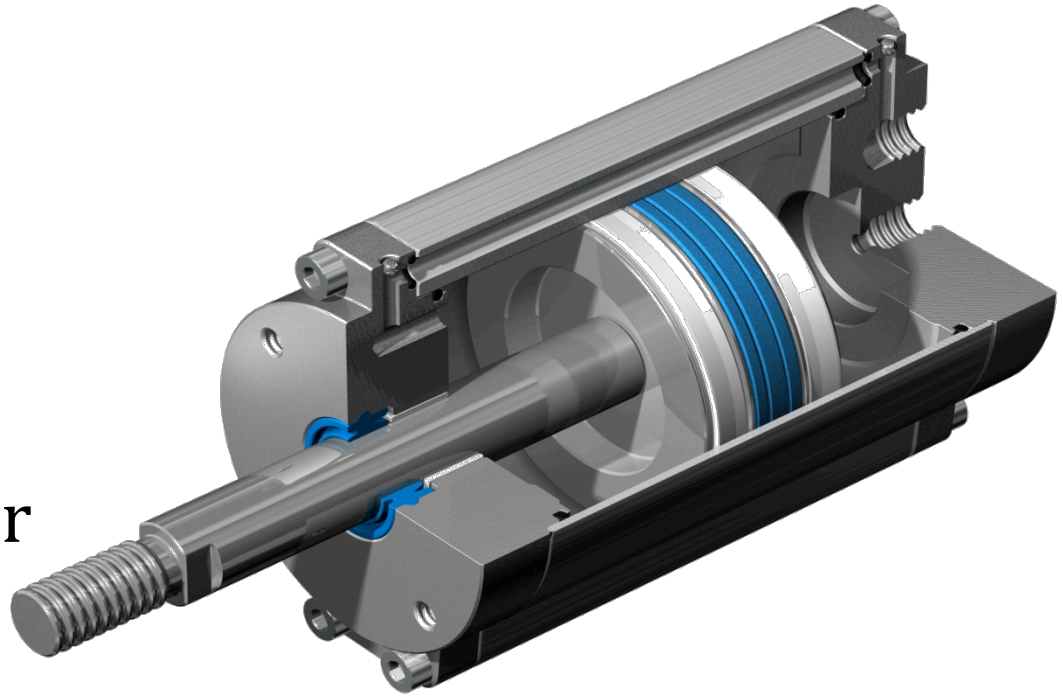
- Device for moving or controlling a system.
- “Robot Muscles”

Hydraulic Actuators

- Pros:
  - Powerful
  - Fast
  - Stiff
- Cons
  - Messy
  - Maintenance
  - External Pump

Pneumatic Actuators

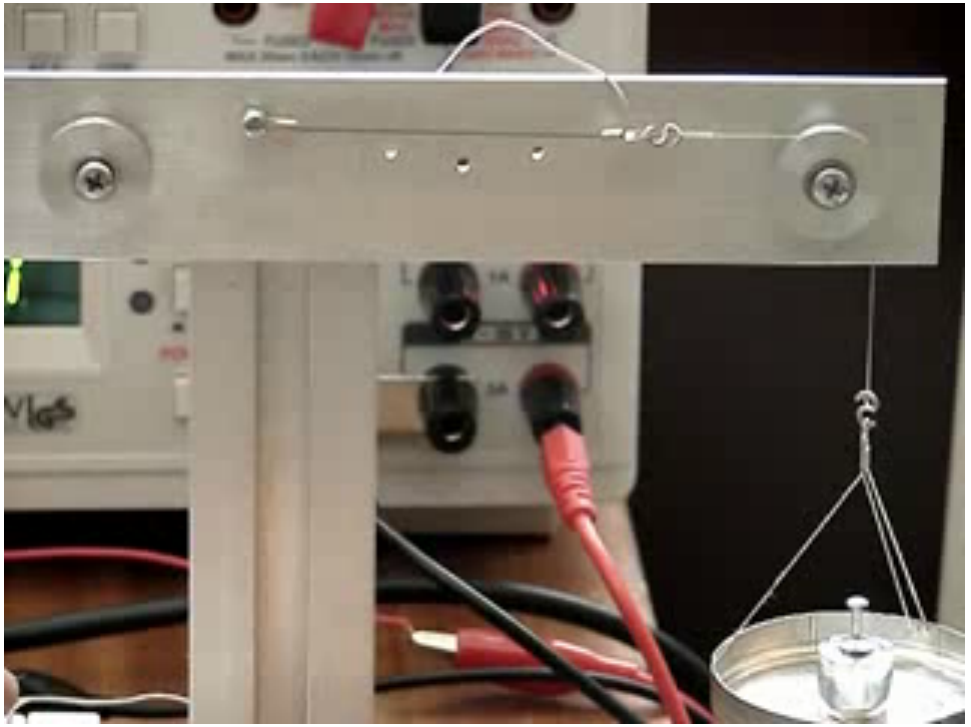
- Pros:
  - Powerful
  - Cheap
- Cons
  - Soft/Compliant
  - External Compressor



Shape Memory Alloy Actuators

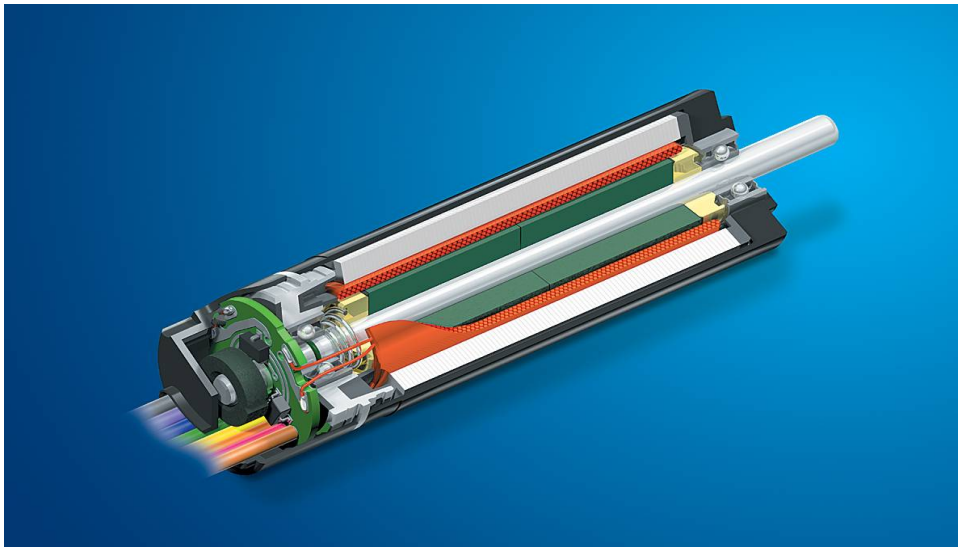
- Works by warming and cooling Nitinol wires.

- Pros:
  - Light
  - Powerful
- Cons:
  - Slow (cooling)



Electric Actuators

- Pros
  - Better position precision
  - Well understood
  - No separate power source
  - Cheap
- Cons
  - Heavy
  - Weaker/slower than hydraulics
  - Cooling issue

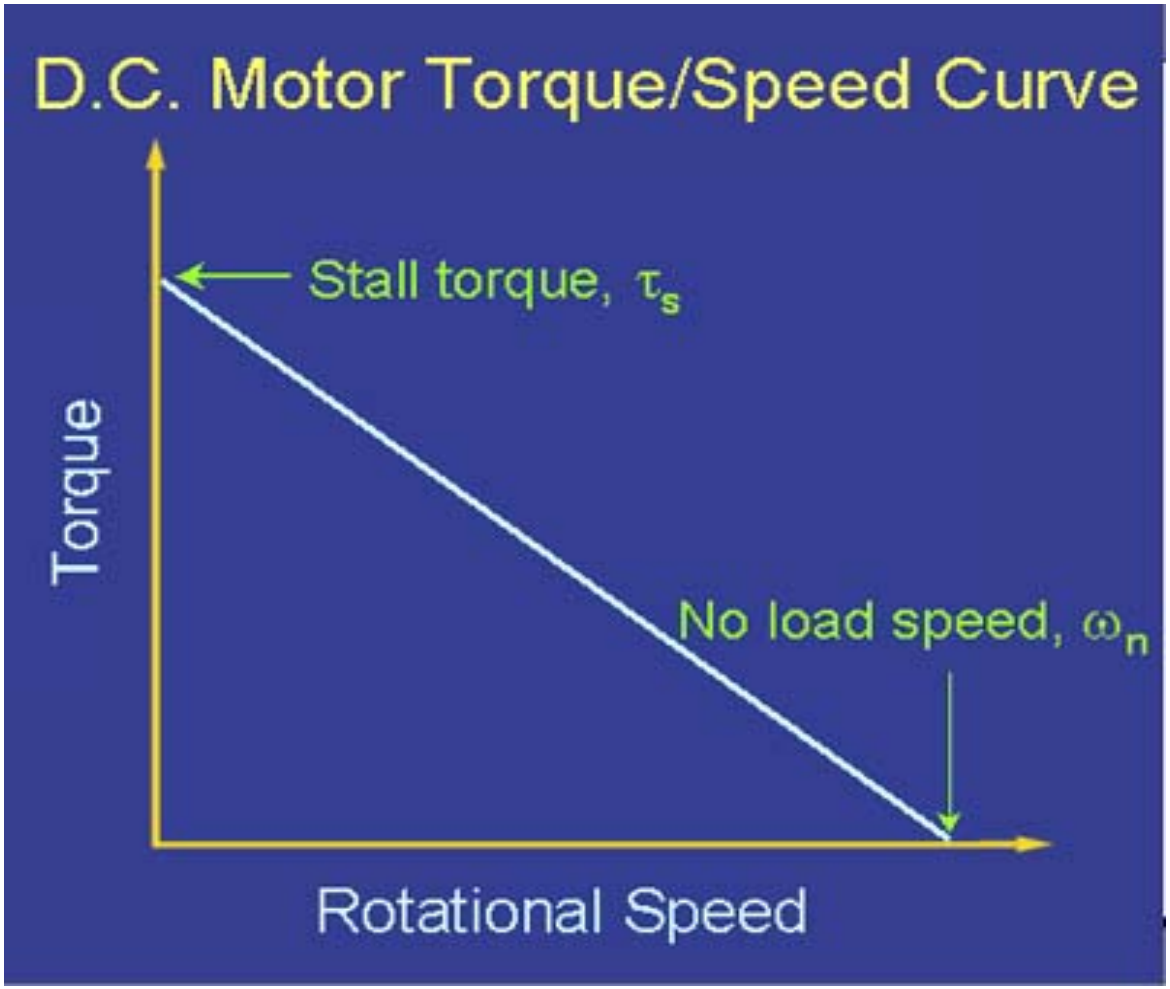


DC motor torque τ

τ = torque

I = current

$$\tau \propto I$$



Vehicle Locomotion

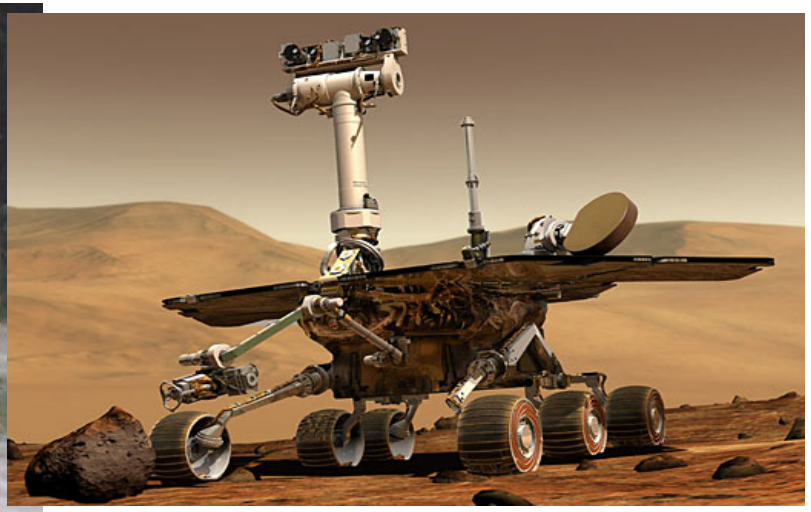
- Objective: convert desire to move A→B into an actual motion:
  - How to arrange actuators (mechanical design)
  - actuator output ←→ Incremental motion: *Forward kinematics* and *inverse kinematics*

Vehicle Locomotion

- *Forward Kinematics:*
  - (actuators actions) → pose
- *Inverse Kinematics (inverse-K):*
  - pose → (actuators actions)

$$\text{pose} = \{x, y, \theta\}$$

Wheeled Vehicles

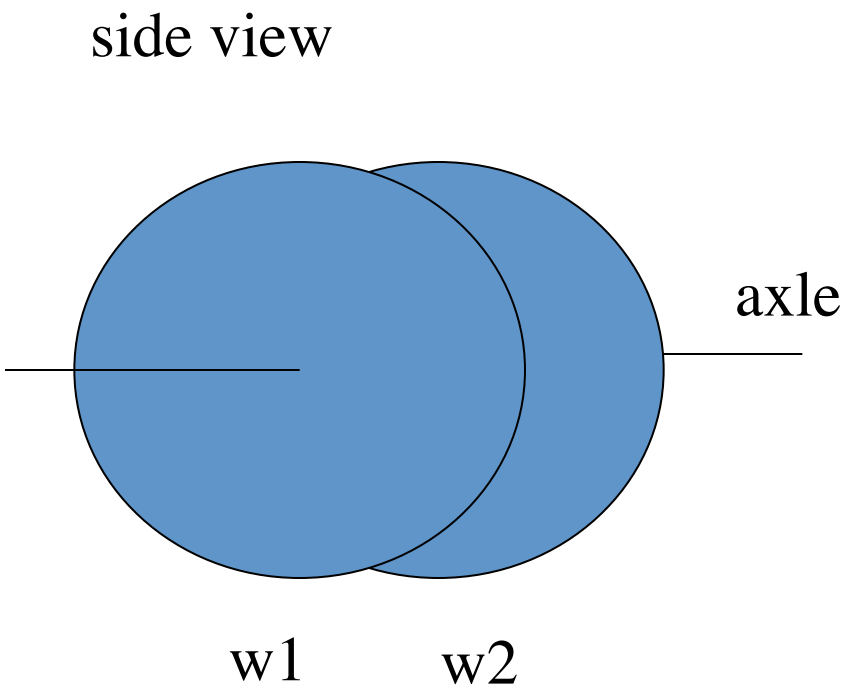


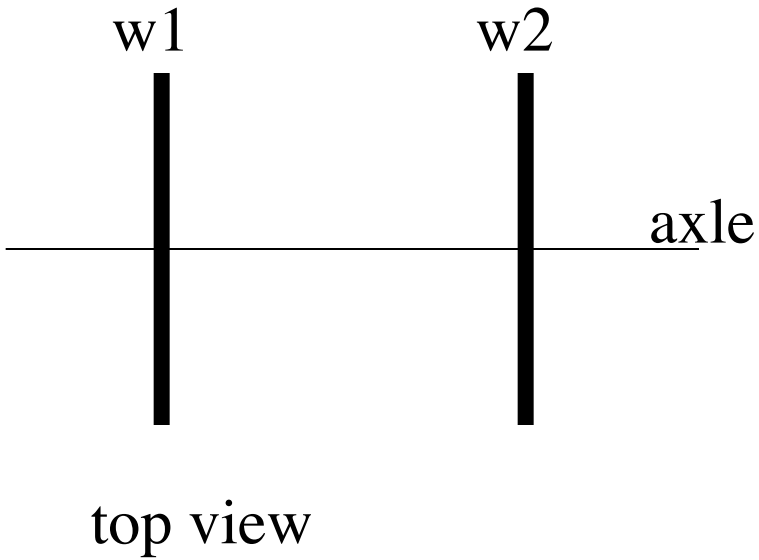
Differential drive

Basic design:

- 2 circular wheels
- infinitely thin
- same diameter
- mounted along a common axis
- vehicle body is irrelevant (in theory).

Idealized differential drive



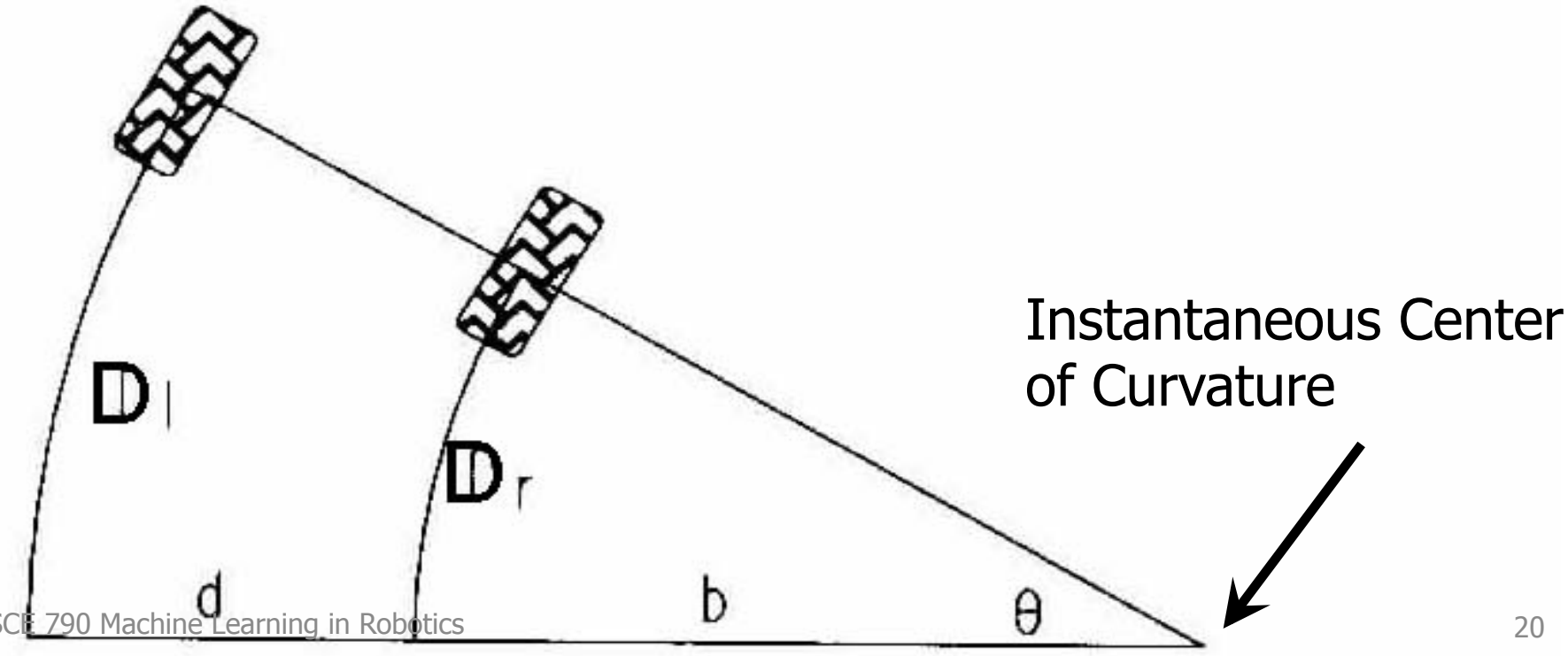


Differential Drive Intuition

- Drive straight ahead?
- Turn in place?
- (these are questions of *kinematics*)

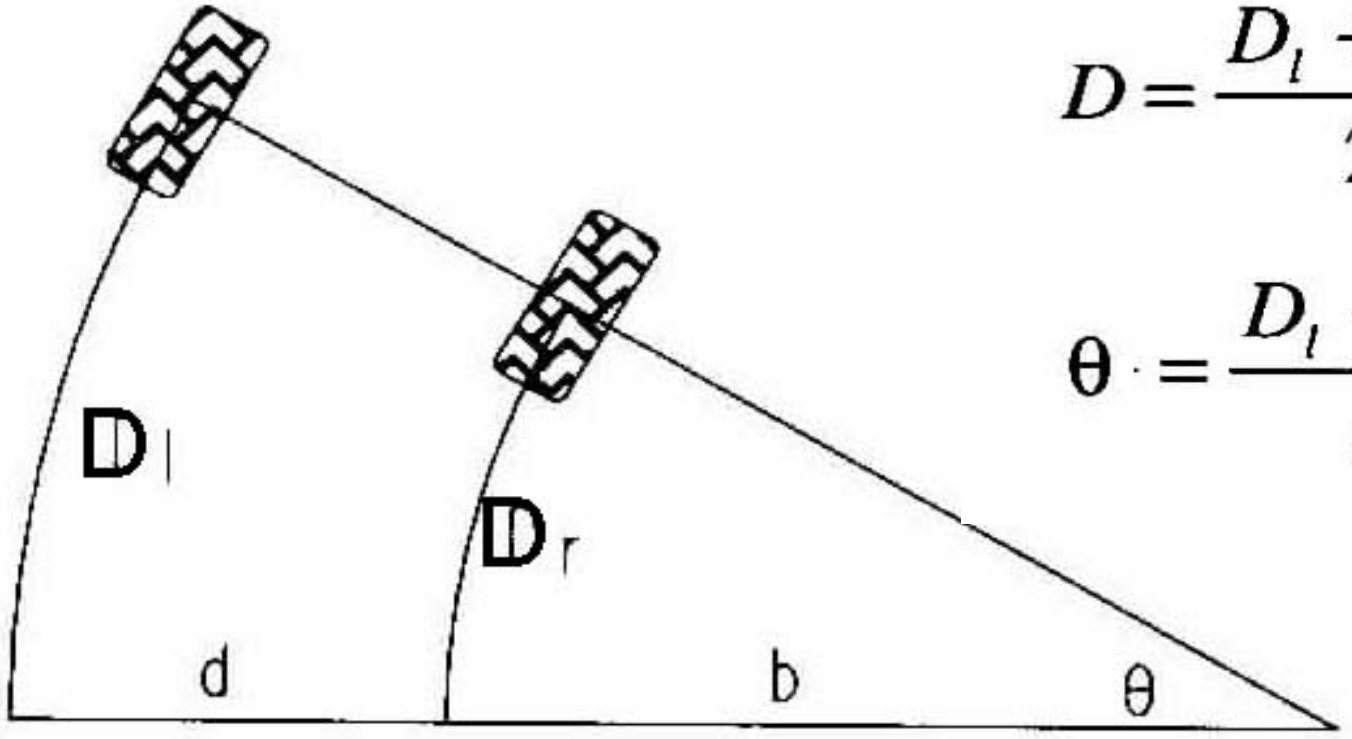
Differential Drive Observation

- Vehicle rotation can be described relative to an axis running though the two wheels.



Forward Kinematics of Differential Drive

- Wheel rotation by angle ϕ_1, ϕ_2
- Distance of wheel motion $D_i = \phi_i r$



$$D = \frac{D_l + D_r}{2}$$

$$\theta = \frac{D_l - D_r}{d}$$

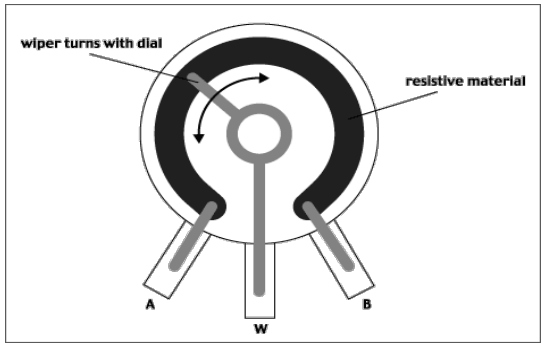
Walking Robots: RHex: Tripod Gait

Actuators + Sensors

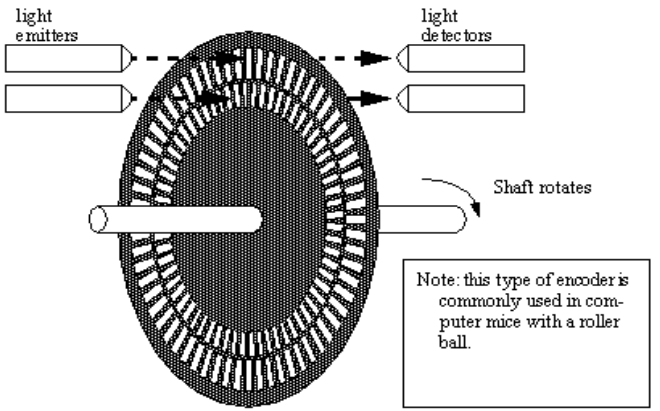
- Presented actuators and sensors, e.g.:
  - Electrical motor
  - Angular position encoder
- How to get the best precision/performance?

Electric Motor

Potentiometer

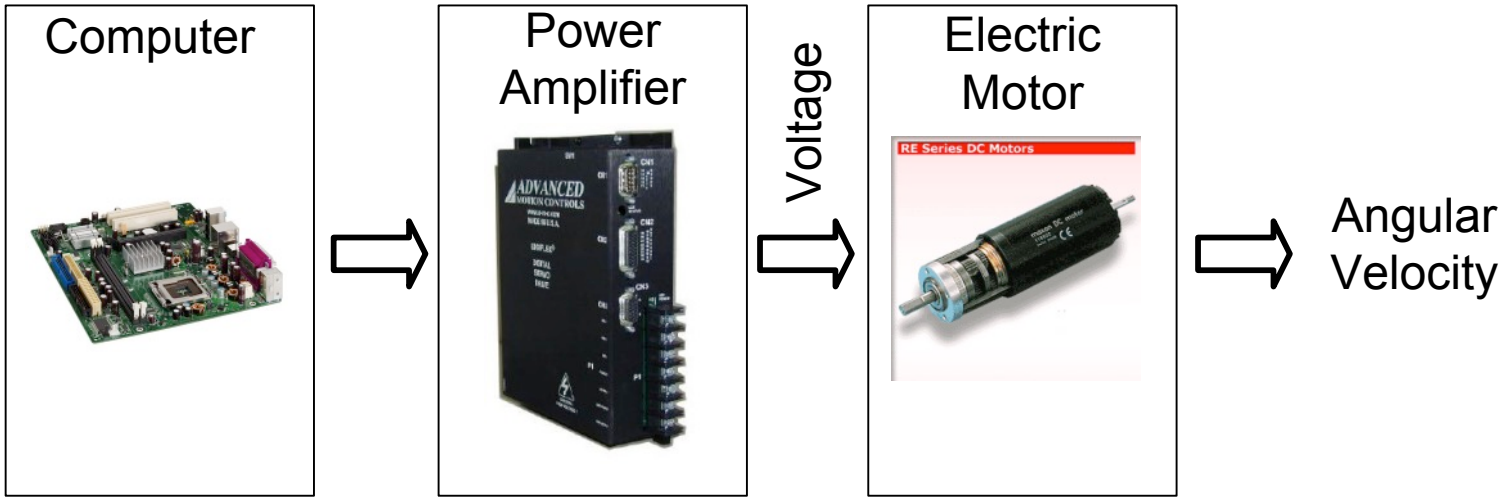


Optical Encoder



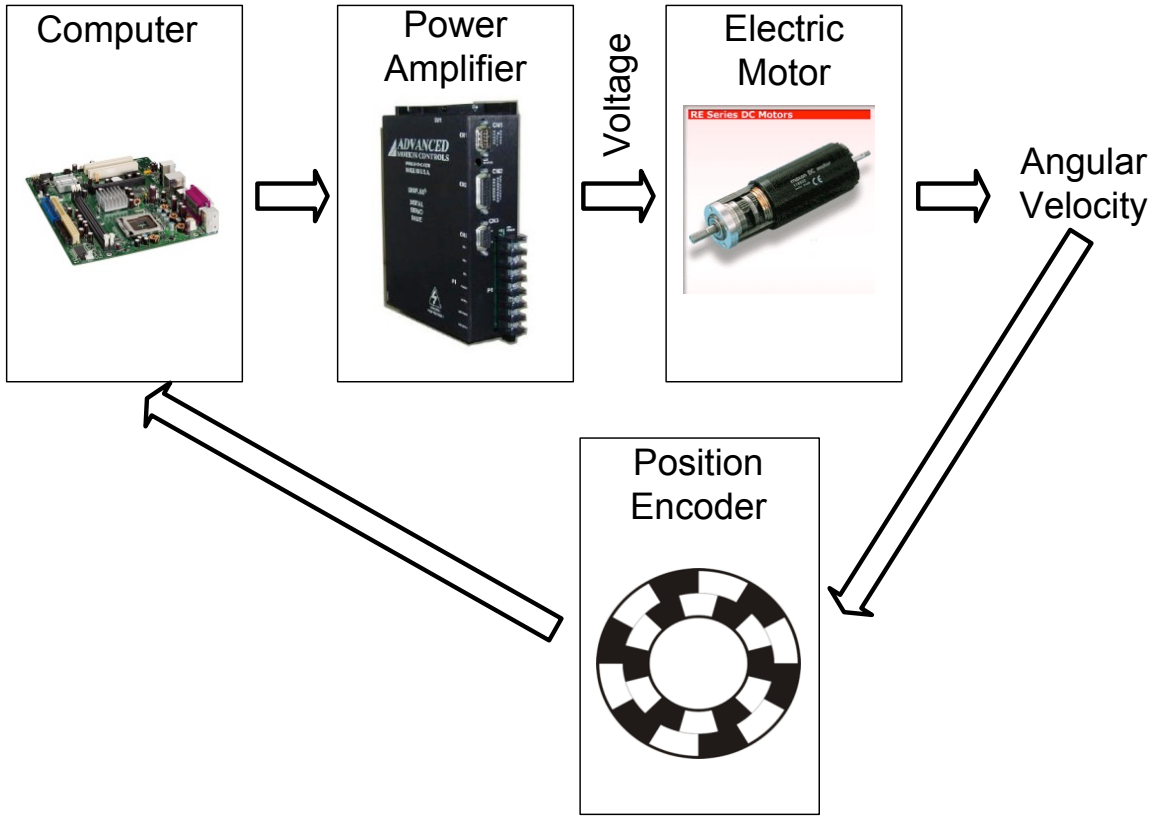
Open-loop

- Goal: move a differential drive robot
  - spin motors at a given angular velocity
- How: apply a fixed voltage to it,
  - and never check to see if it is rotating properly…



Closing the loop

- Let's measure the actual angular velocities.
- Compensate for changes in load by *feeding back* some information.



Components of Feedback Systems

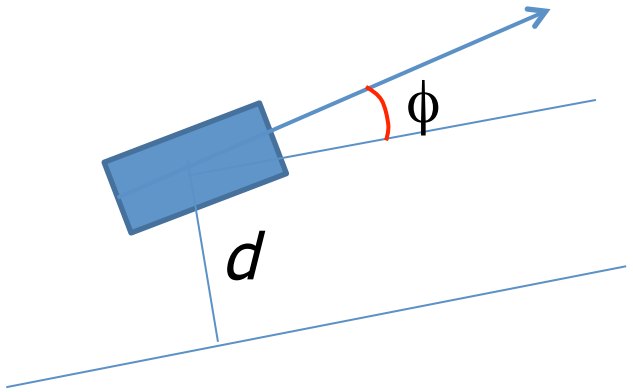
- Power amplification
  - Neural signal power (μW) vs. muscle power output (tens of W)
  - Means it is an active system, as opposed to passive.
- Actuator
- Feedback
  - measurement (sensor)
- Error signal
- Controller

Effect of Controller Functions

- Proportional Action
  - Simplest Controller Function
- Integral Action
  - Eliminates steady-state error
  - Can cause oscillations
- Derivative Action ("rate control")
  - Effective in transient periods
  - Provides faster response (higher sensitivity)
  - Never used alone

Line Following

- Consider a 2D robot with pose (x, y, θ) following a line. The robot is moving with constant speed v, and we control the angular velocity ω *(turnrate)*.

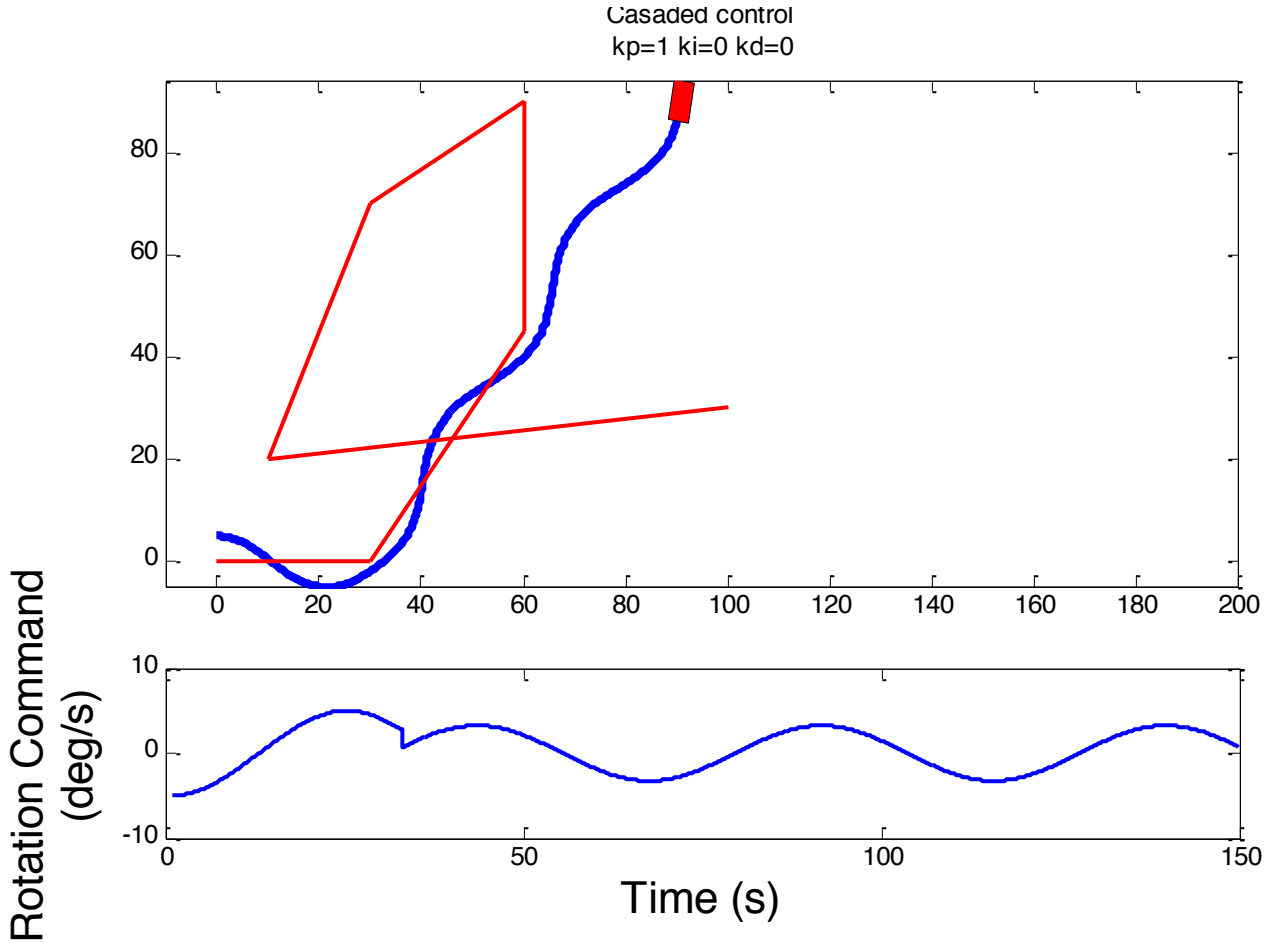


Calculating the error

- Consider two error measures,
  - d
  - ϕ
- The orientation error is used to estimate when we are heading too much in the wrong direction
  - $\text{if } |\phi| < 90^\circ \; \omega \text{ is bounded}$
- The distance error is what we feed to our controller.

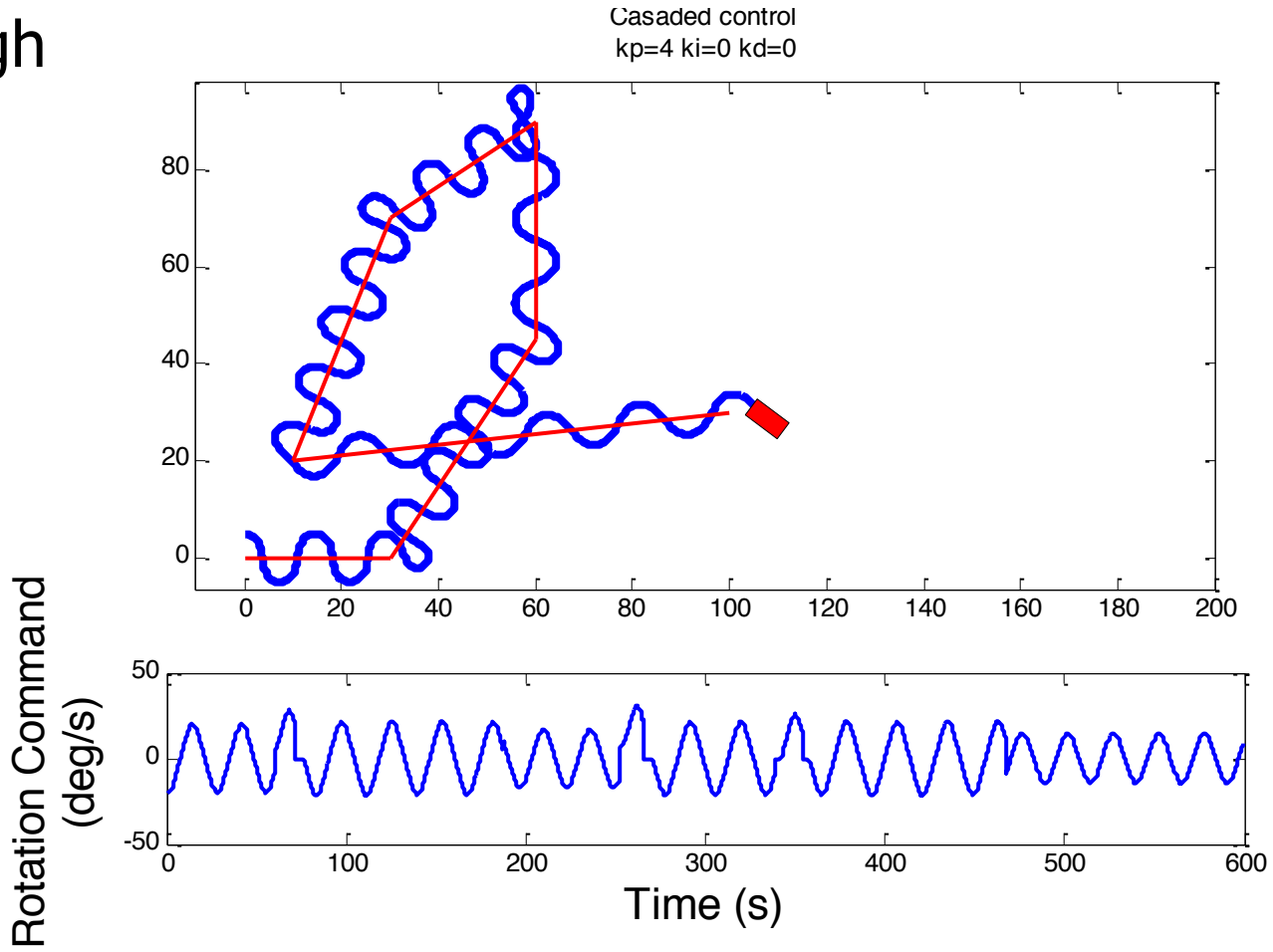
P-Controller

Gain too low



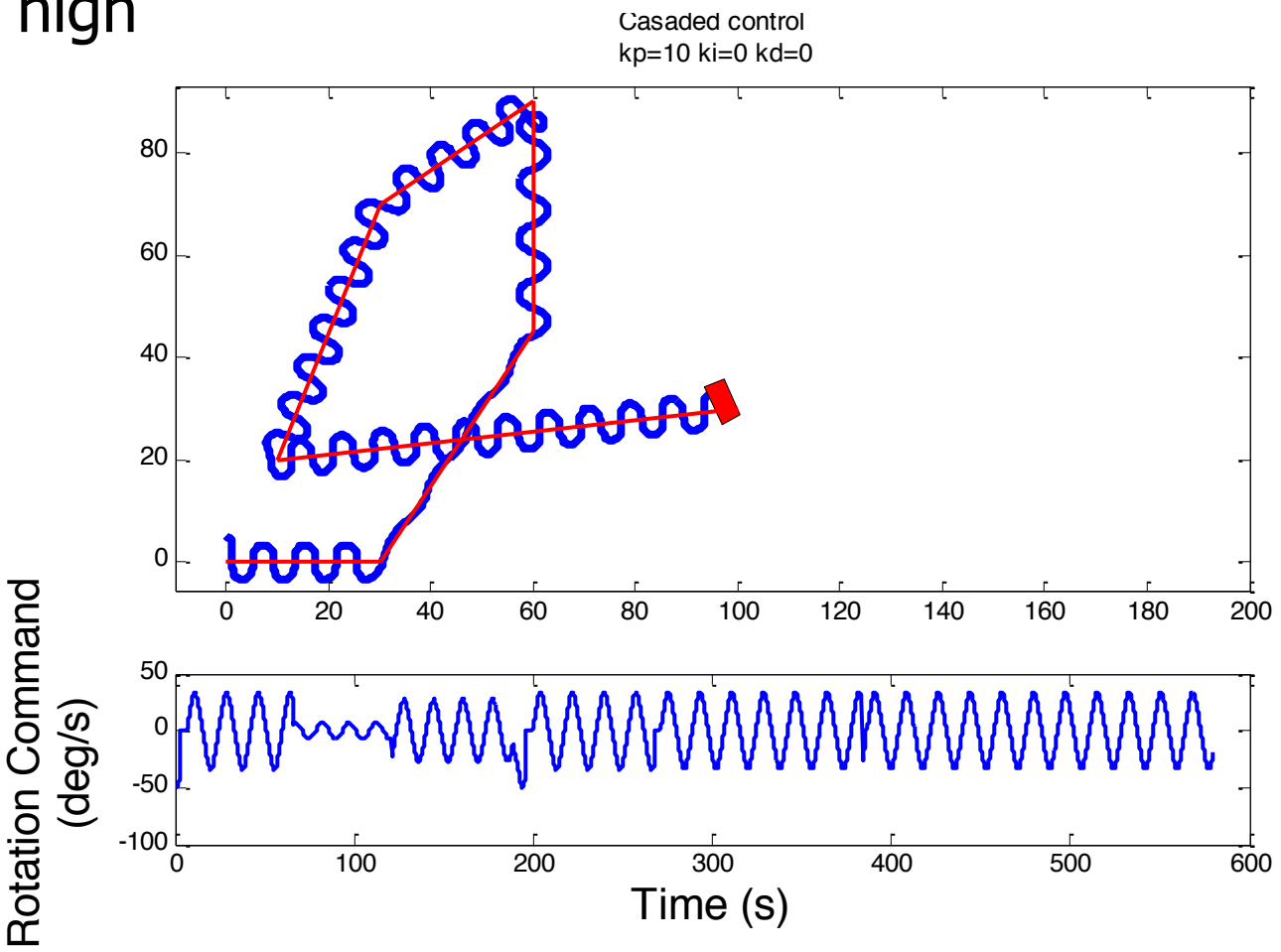
P-Controller

Gain high



P-Controller

Gain too high



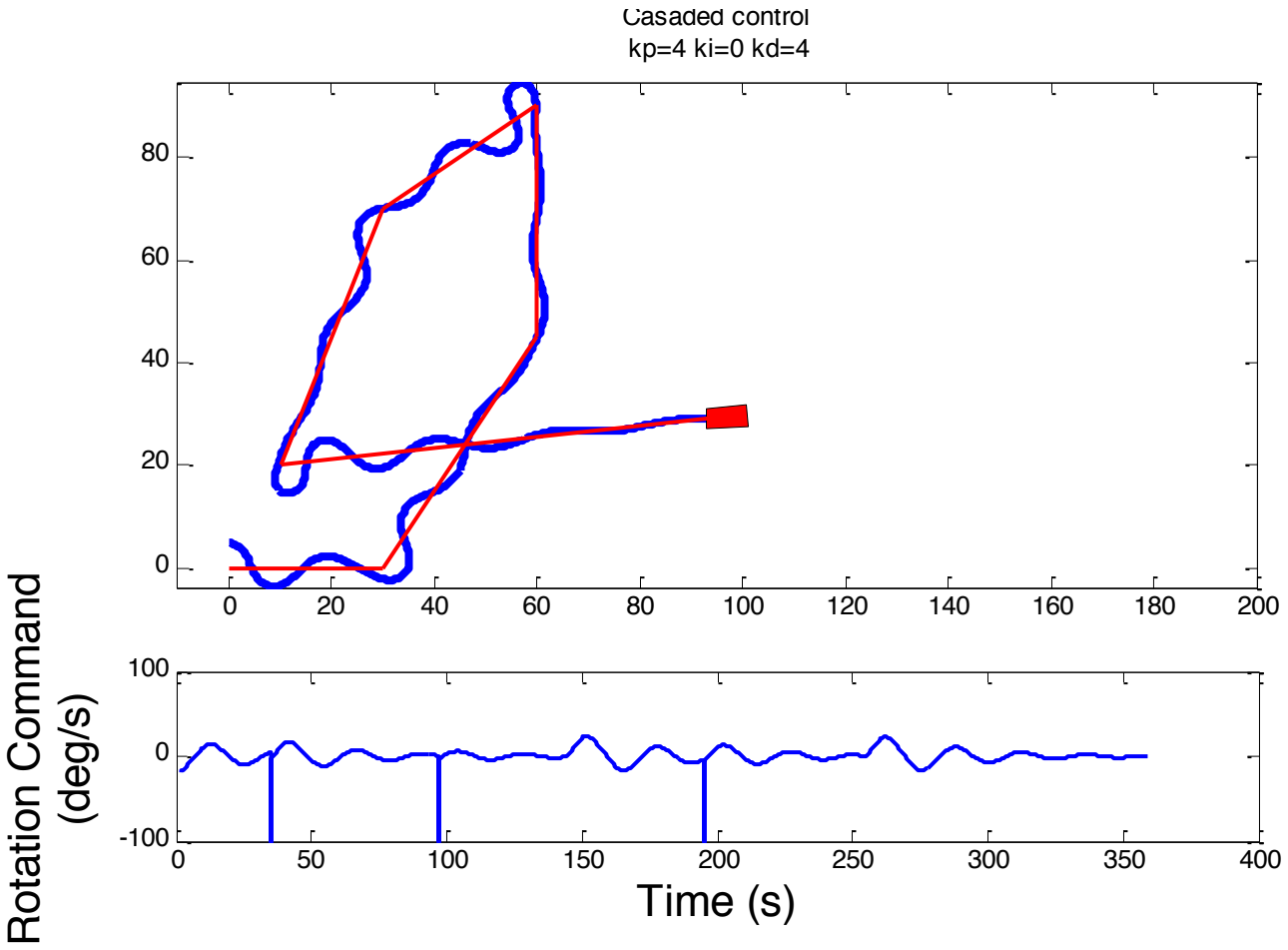
Pd-Controller

- We are using also the derivative of the error:
  - $\dot{e} = \frac{d_t - d_{t-1}}{dt}$

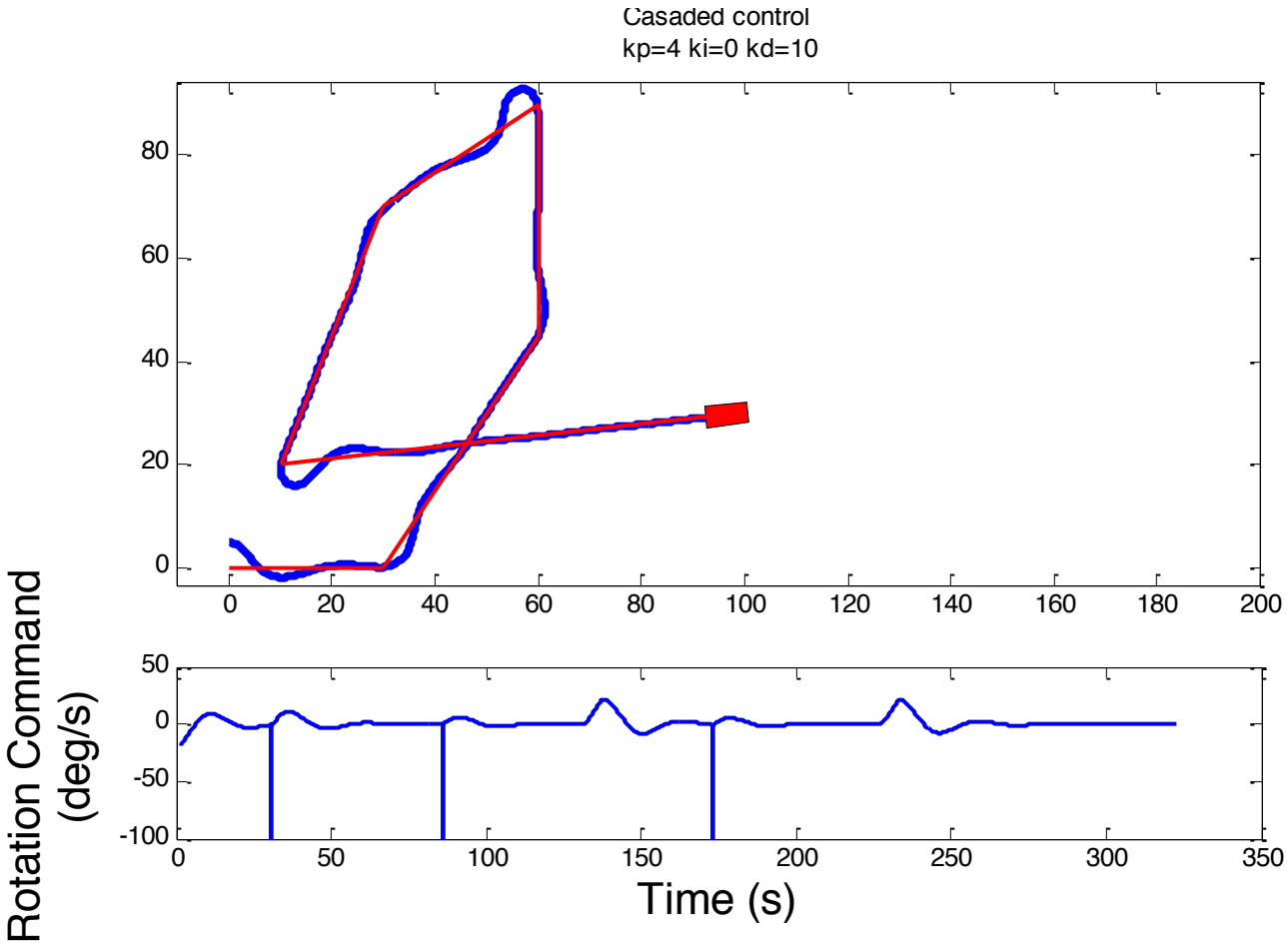
  to calculate the turnrate ω:

$$\omega = k_p \cdot d_t + k_d \cdot \dot{e}$$

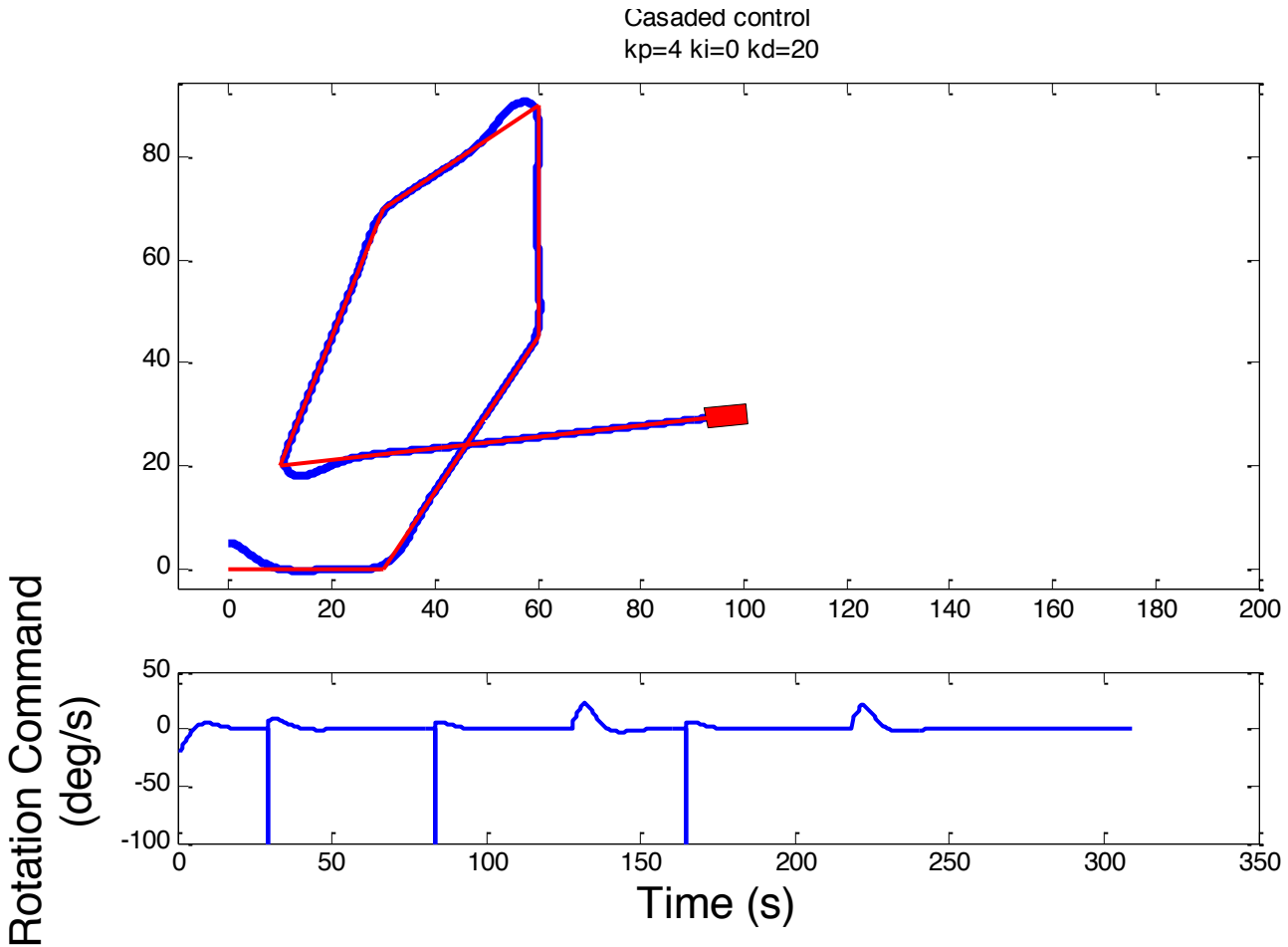
Pd-Controller



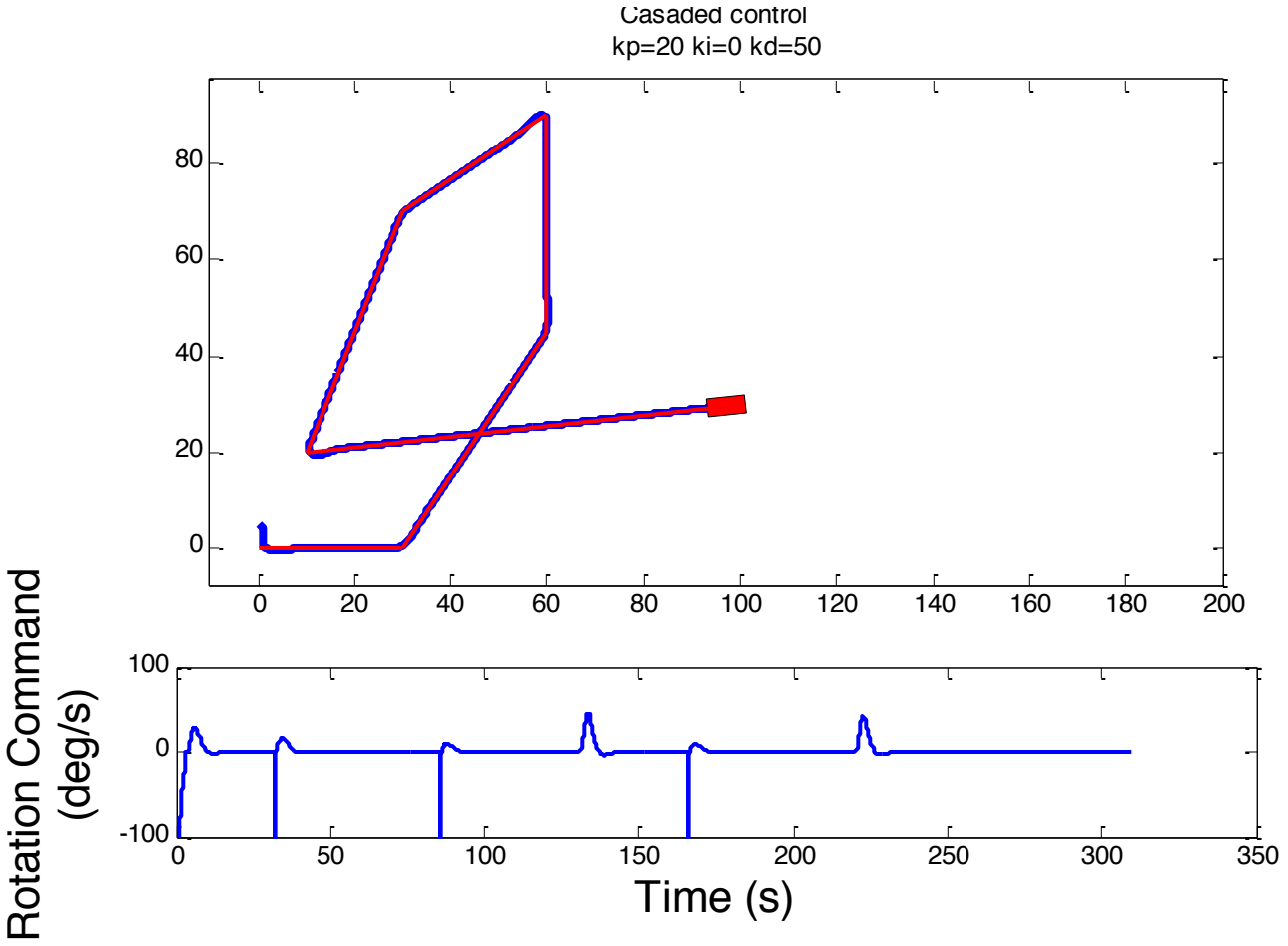
Pd-Controller



Pd-Controller



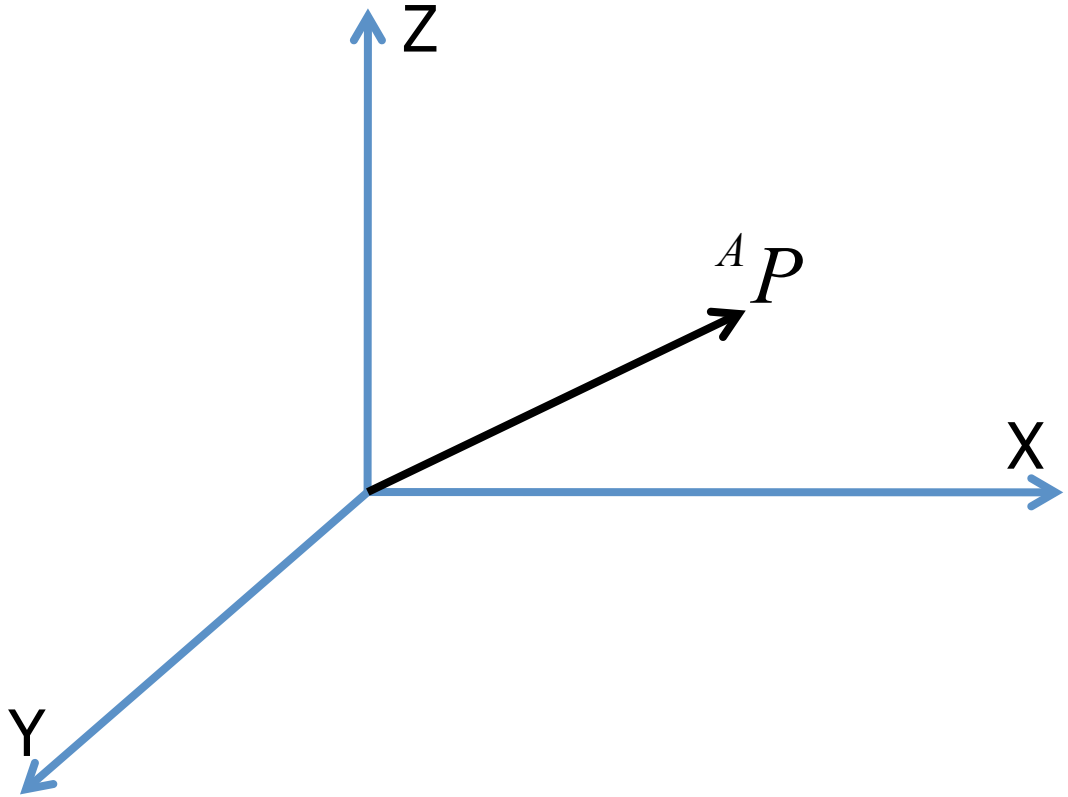
Pd-Controller



Position Representation

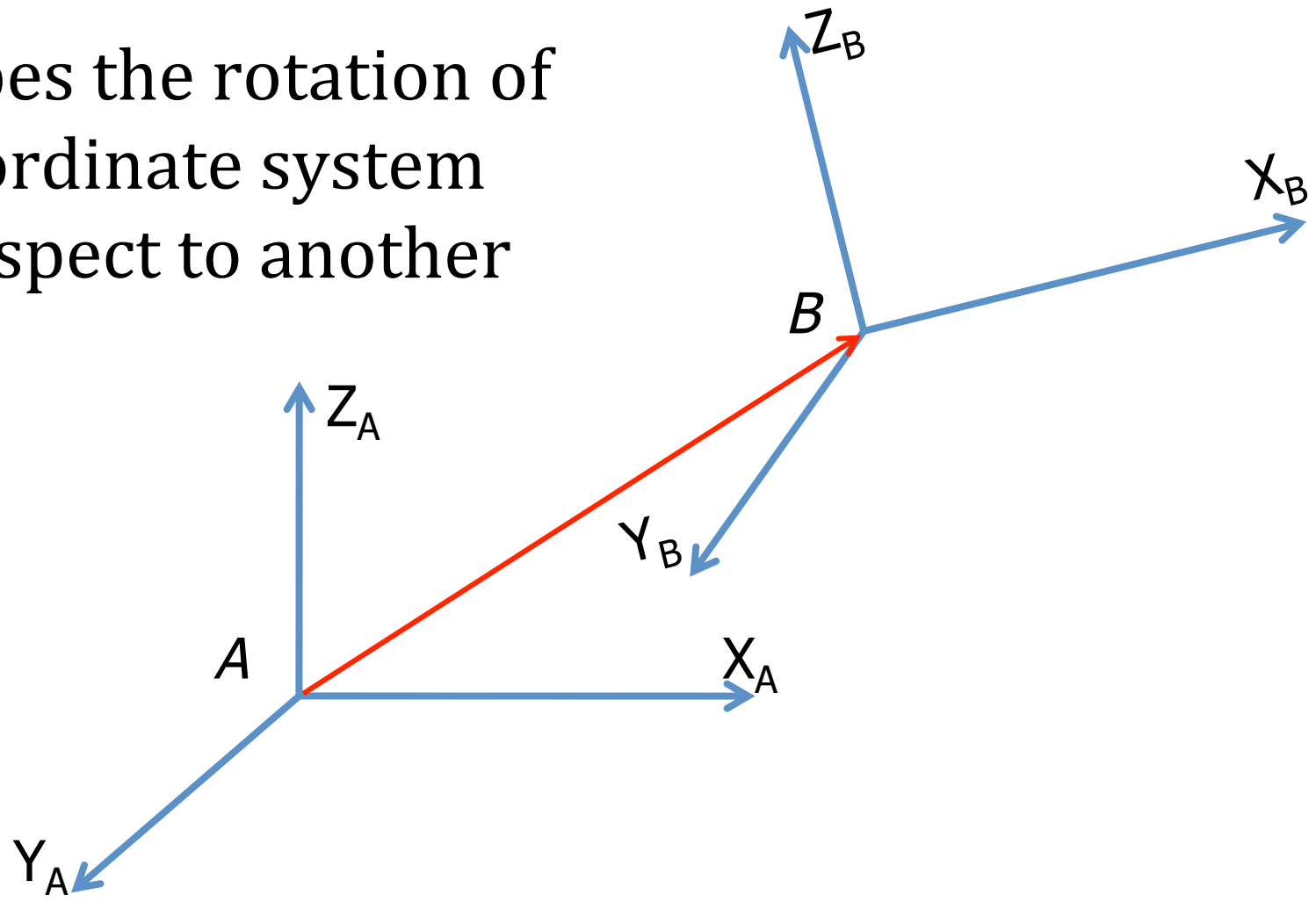
- Position representation is:

$$ {}^{A}P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} $$



Orientation Representations

- Describes the rotation of one coordinate system with respect to another

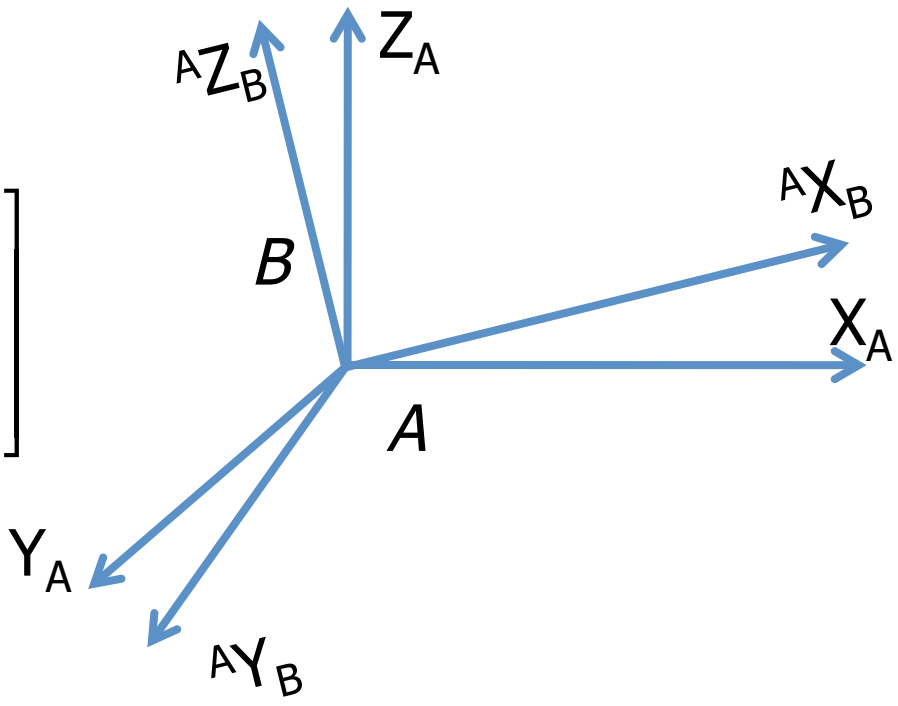


Rotation Matrix

- Write the unit vectors of B in the coordinate system of A.
- Rotation Matrix:

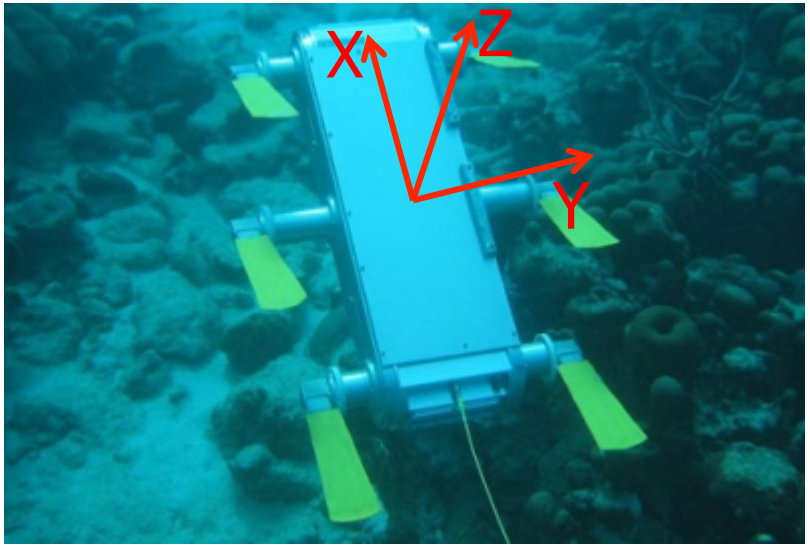
$$
{}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_B & {}^{A}\hat{Y}_B & {}^{A}\hat{Z}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}
$$

$$
= \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix}
$$



Euler Angles

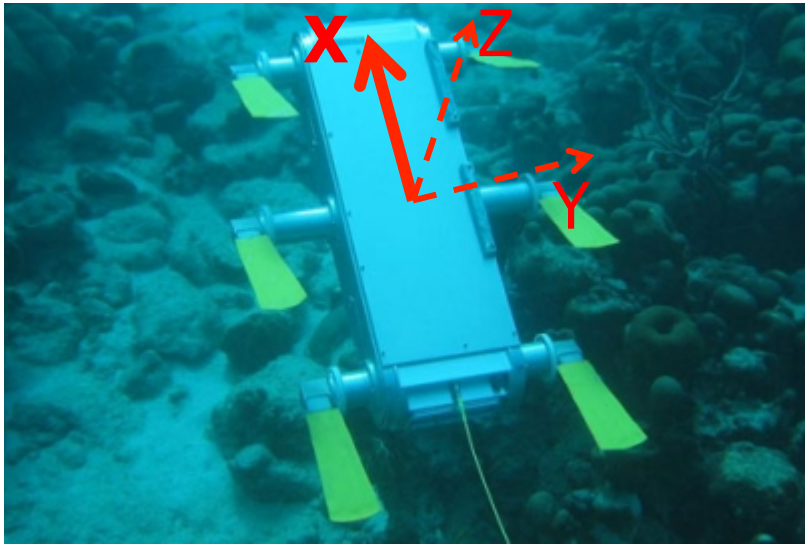
- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw



Euler Angles

- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

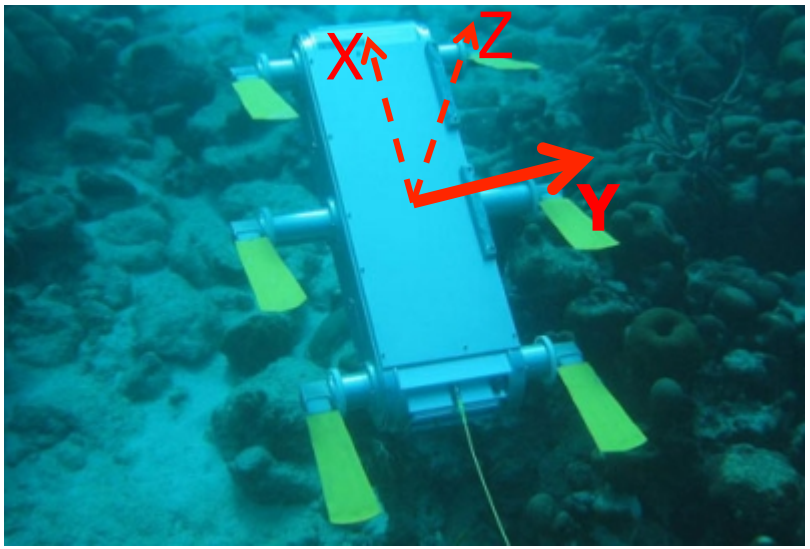
Roll



Euler Angles

- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

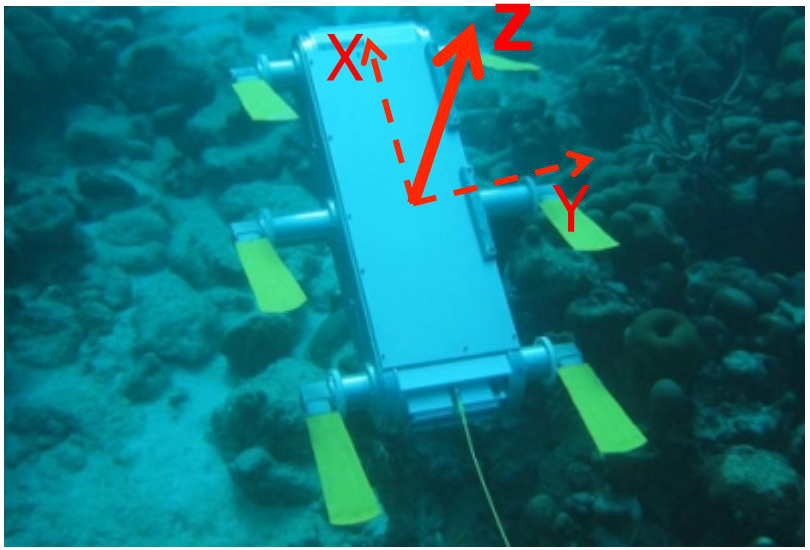
Pitch



Euler Angles

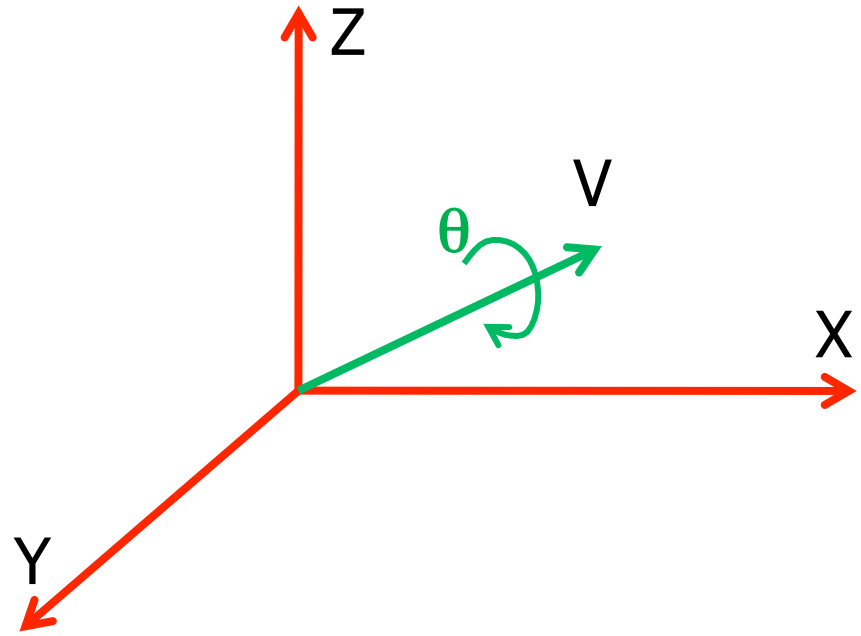
- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

Yaw



Axis-Angle Representation

- Represent an arbitrary rotation as a combination of a vector and an angle



Quaternions

- Are similar to axis-angle representation
- Two formulations
  - Classical
  - Based on JPL's standards

    W. G. Breckenridge, "Quaternions - Proposed Standard Conventions," JPL, Tech. Rep. INTEROFFICE MEMORANDUM IOM 343-79-1199, 1999.
- Avoids Gimbal lock

- See also: M. D. Shuster, "A survey of attitude representations," Journal of the Astronautical Sciences, vol. 41, no. 4, pp. 439–517, Oct.–Dec. 1993.

Quaternions

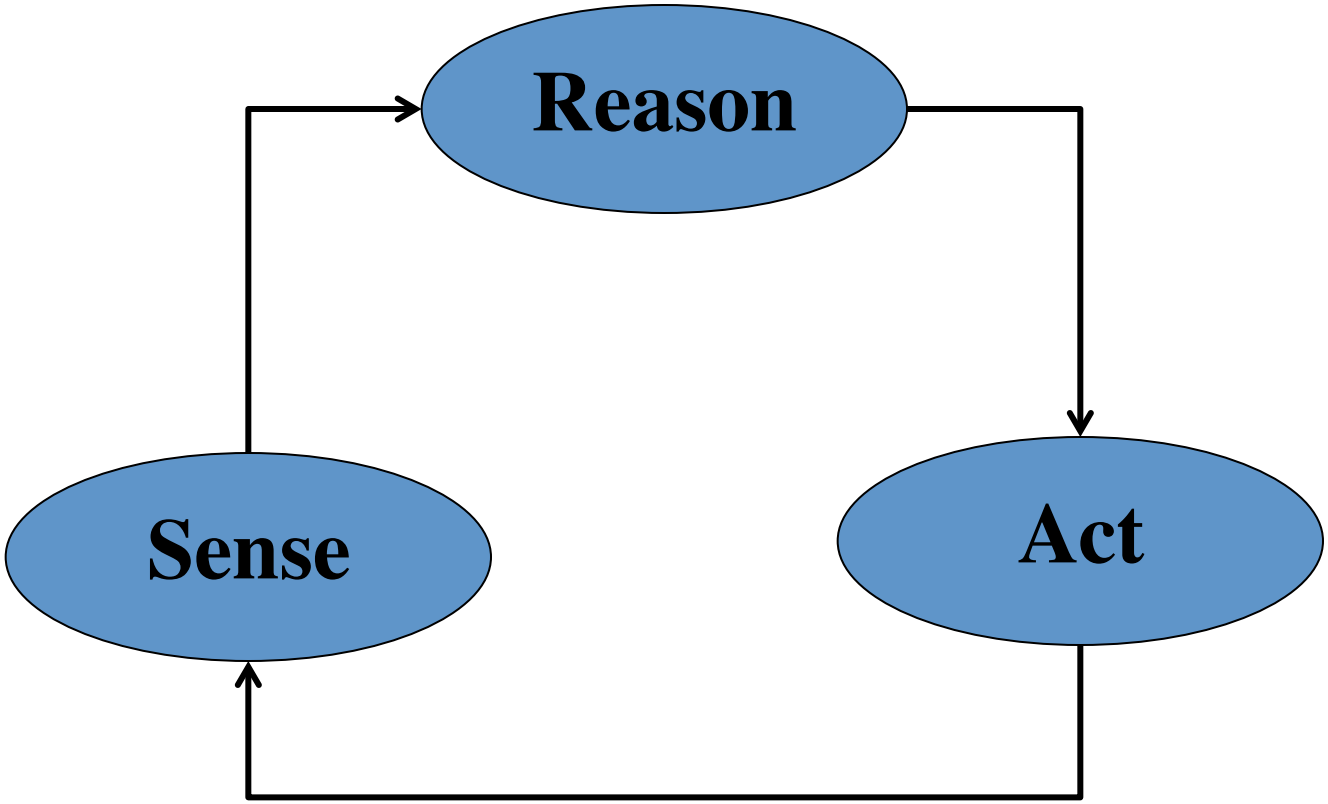
	Classic notation	JPL-based
	$\bar{q} = q_4 + q_1 i + q_2 j + q_3 k$	$\bar{q} = q_4 + q_1 i + q_2 j + q_3 k$
	$i^2 = j^2 = k^2 = ijk = -1$ $ij = -ji = k,\ jk = -kj = i,\ ki = -ik = j$	$i^2 = j^2 = k^2 = -1$ $-ij = ji = k, -jk = kj = i, -ki = ik = j$
Vector Notation	$\bar{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix},\ q_0 = \cos\left(\theta/2\right),\ \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \sin\left(\theta/2\right)\cos(\beta_x) \\ \sin\left(\theta/2\right)\cos(\beta_y) \\ \sin\left(\theta/2\right)\cos(\beta_z) \end{bmatrix}$	$\bar{q} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix},\ \mathbf{q} = \begin{bmatrix} k_x \sin\left(\theta/2\right) \\ k_y \sin\left(\theta/2\right) \\ k_z \sin\left(\theta/2\right) \end{bmatrix},\ q_4 = \cos\left(\theta/2\right)$
		$\left\|\bar{q}\right\| = 1, \bar{q} \otimes \bar{p}, \mathbf{q} \times \mathbf{p}, \bar{q}_I, \lfloor \mathbf{q} \times \rfloor$

See also: N. Trawny and S. I. Roumeliotis, "Indirect Kalman Filter for 3D Attitude Estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep. 2005-002, March 2005.

Three Main Challenges in Robotics

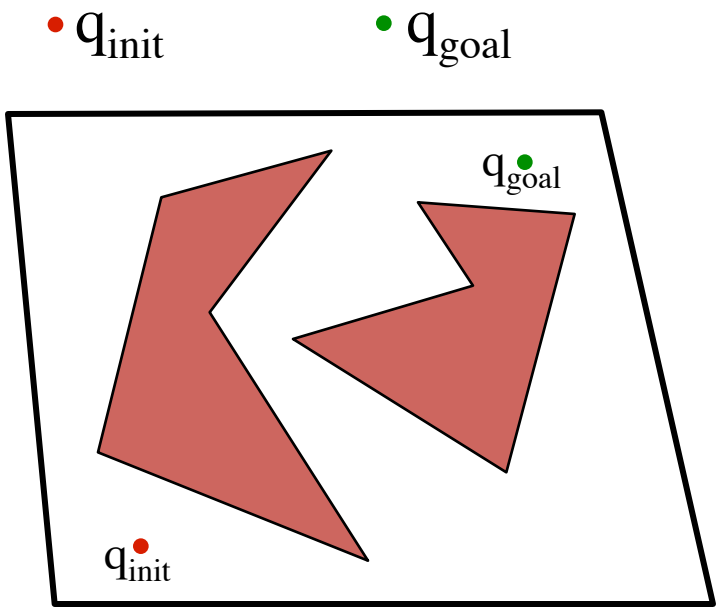
1. Where am I? (Localization)
2. What the world looks like? (Mapping)
   - Together 1 and 2 form the problem of *Simultaneous Localization and Mapping* (SLAM)
3. How do I go from **A** to **B**? (Path Planning)
   - More general: Which action should I pick next? (Planning)

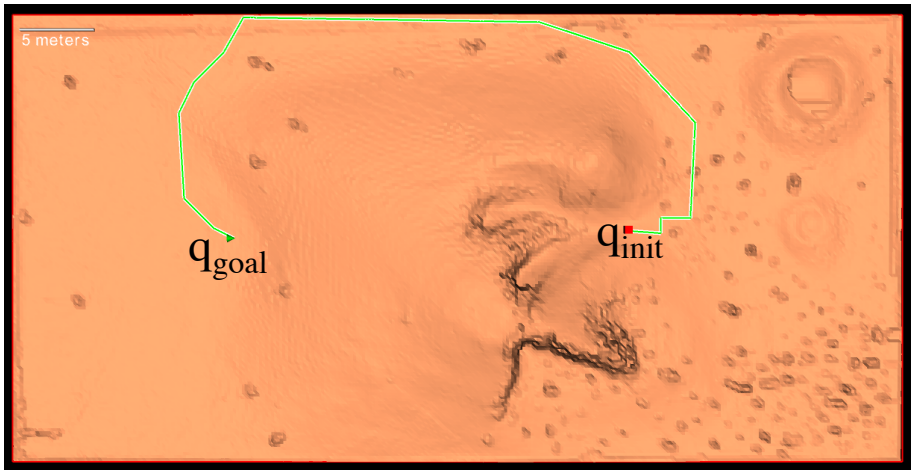
Robot



Motion Planning

- The ability to go from **A** to **B**
  - Known map – Off-line planning
  - Unknown Environment –Online planning
  - Static/Dynamic Environment





Path Planning

World

- Indoor/Outdoor
- 2D/2.5D/3D
- Static/Dynamic
- Known/Unknown
- Abstract (web)

Robot

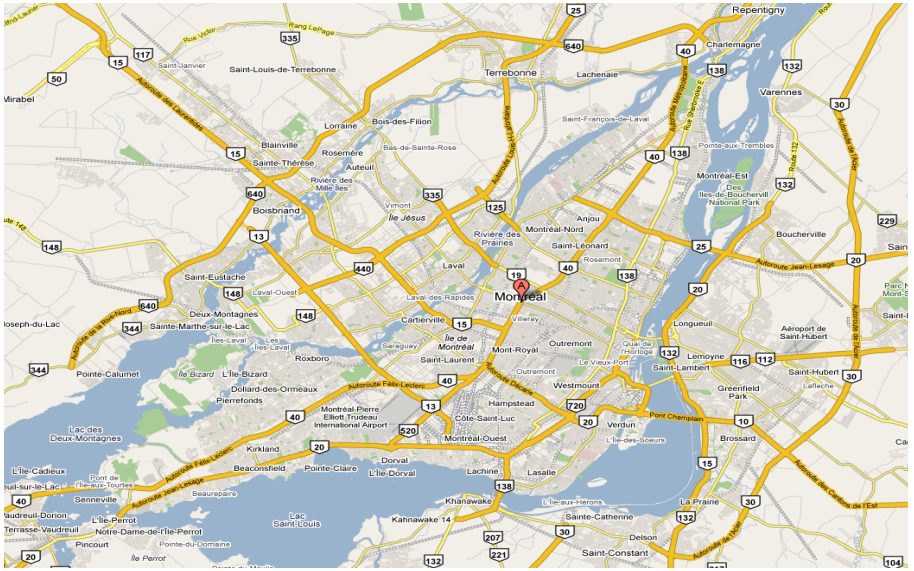
- Mobile
  - Indoor/Outdoor
  - Walking/Flying/Swimming
- Manipulator
- Humanoid
- Abstract

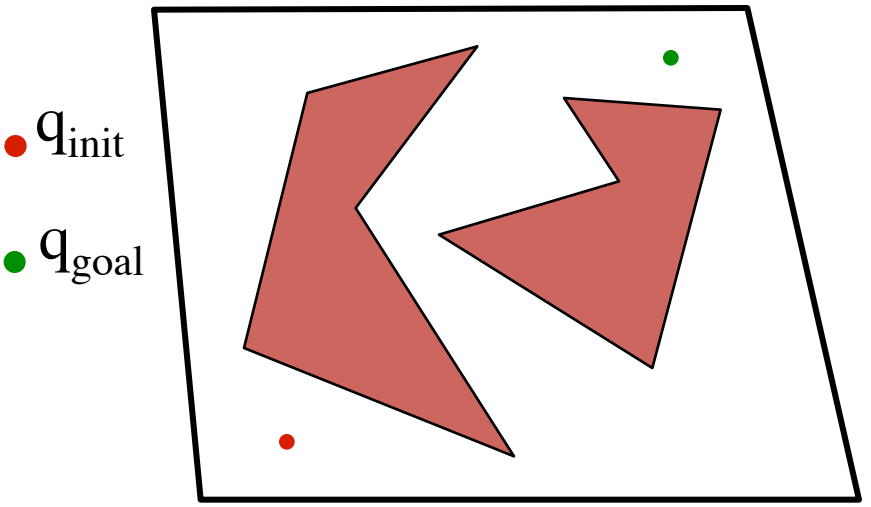
Map

- Topological
- Metric
- Feature Based
- 1D,2D,2.5D,3D

Path Planning: Assumptions

- Known Map
- Roadmaps (Graph representations)
- Polygonal Representation



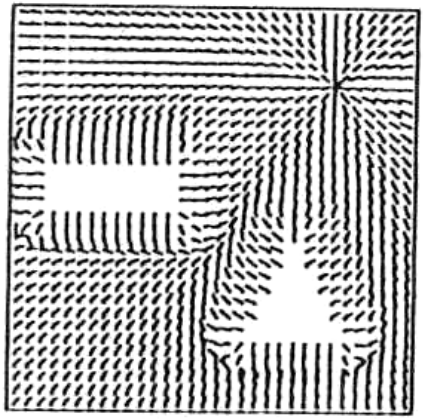


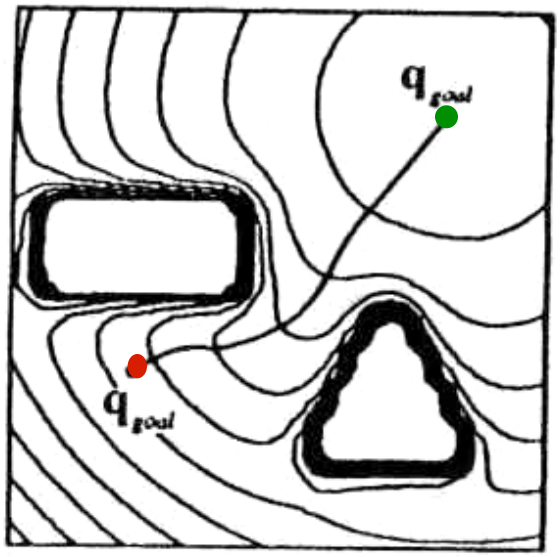
Local techniques

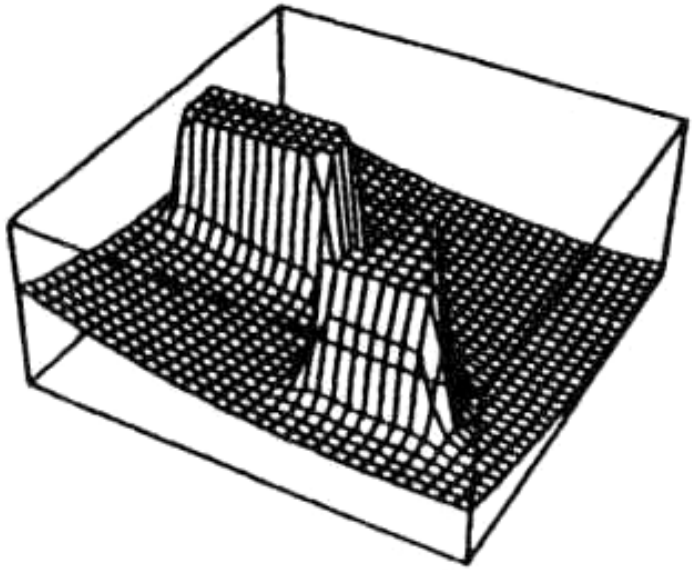
Potential Field methods

- compute a repulsive force away from obstacles
- compute an attractive force toward the goal

→ let the sum of the forces control the robot



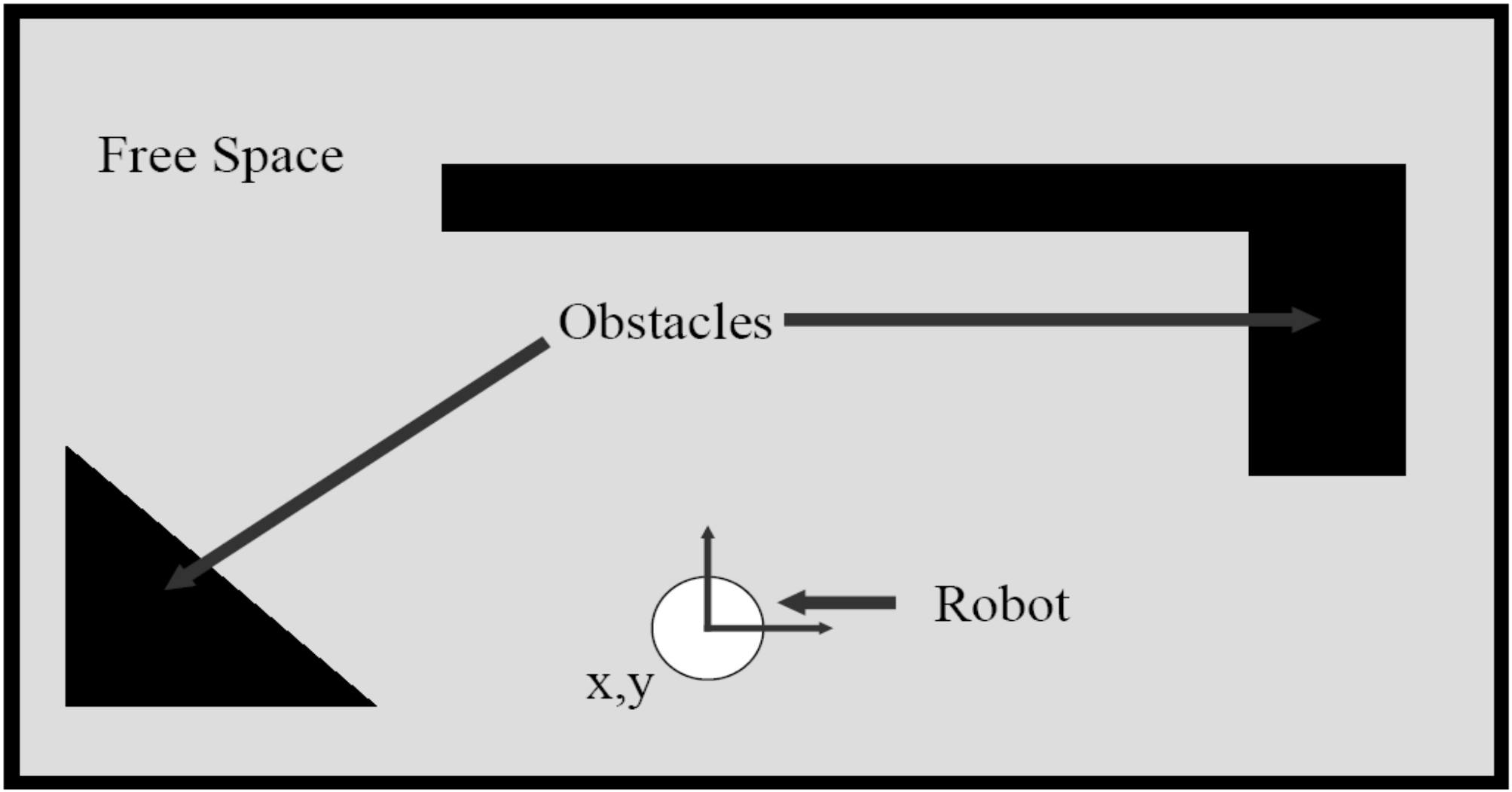




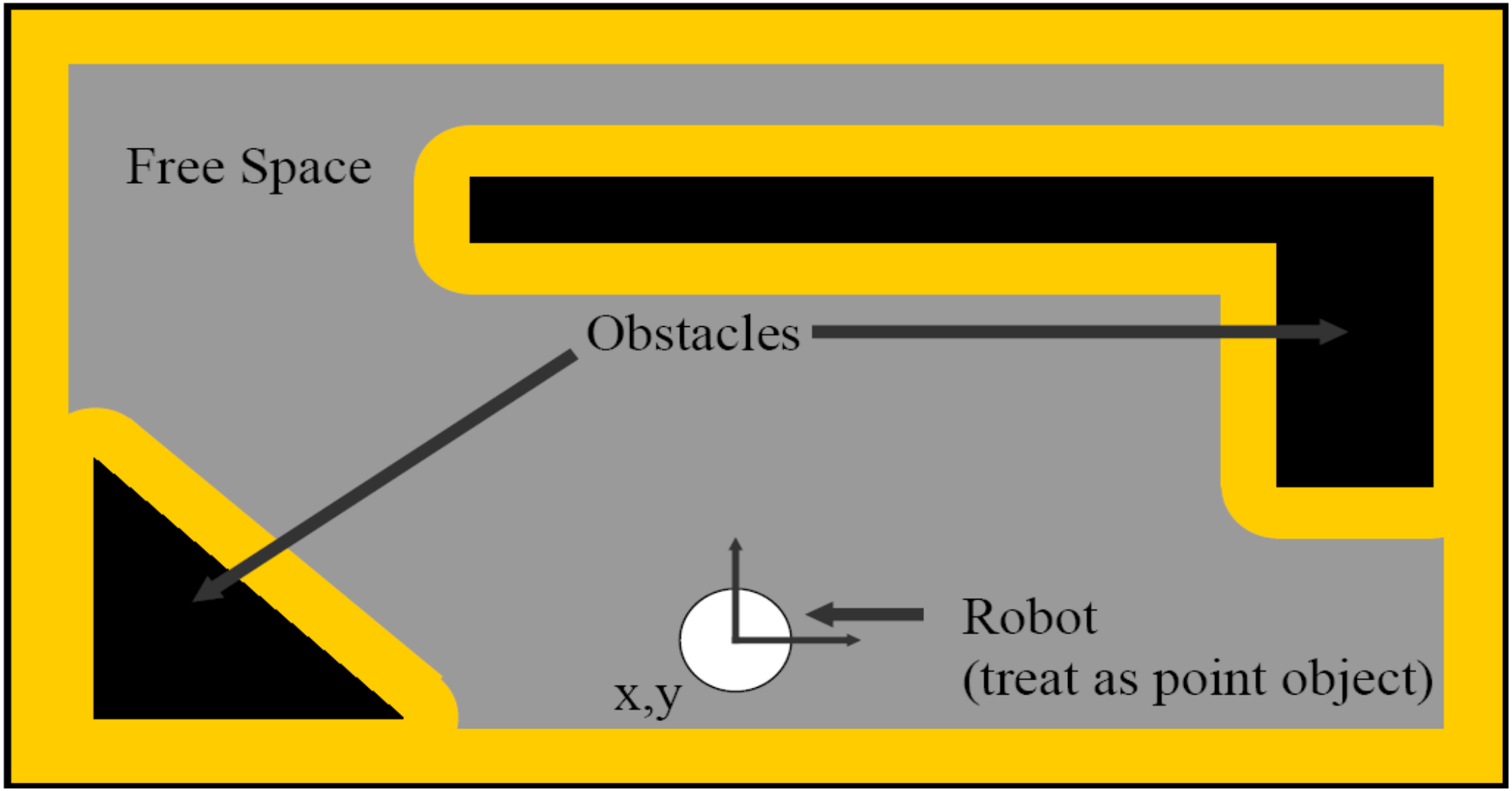
key advantages?

UNIVERSITY OF SOUTH CAROLINA

Configuration Space



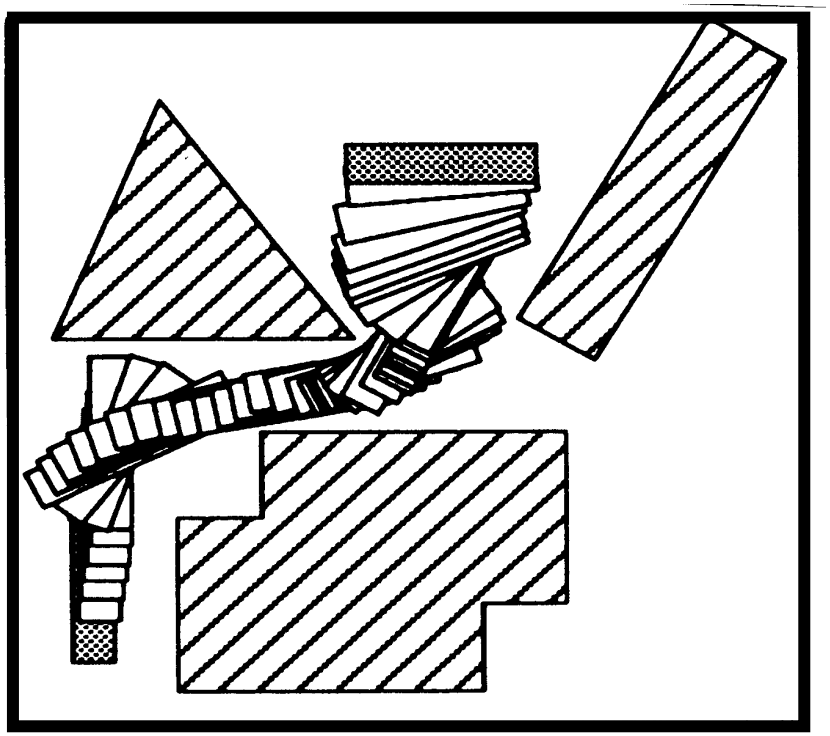
Configuration Space

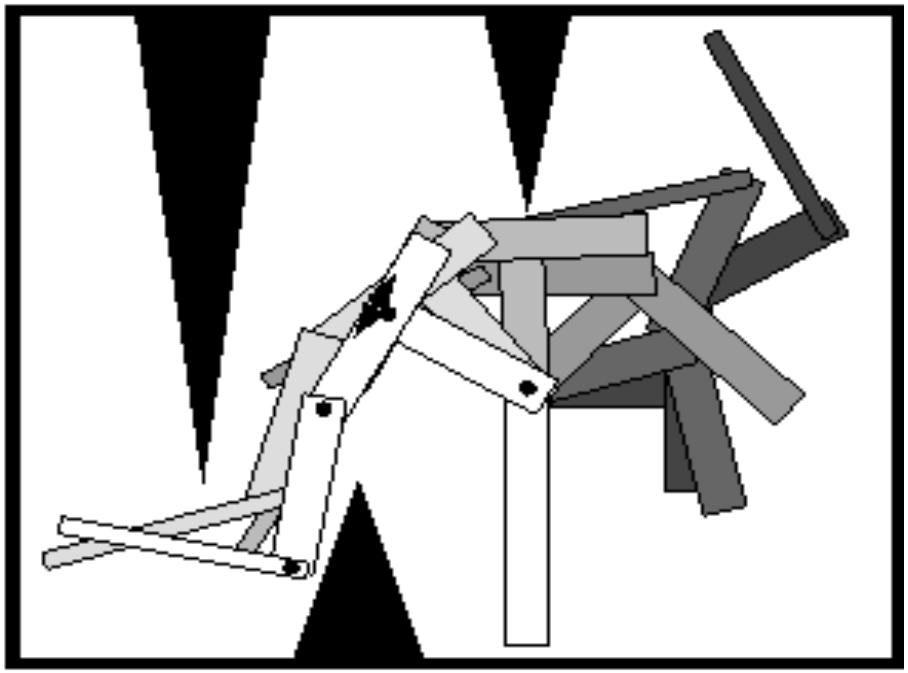


Definition

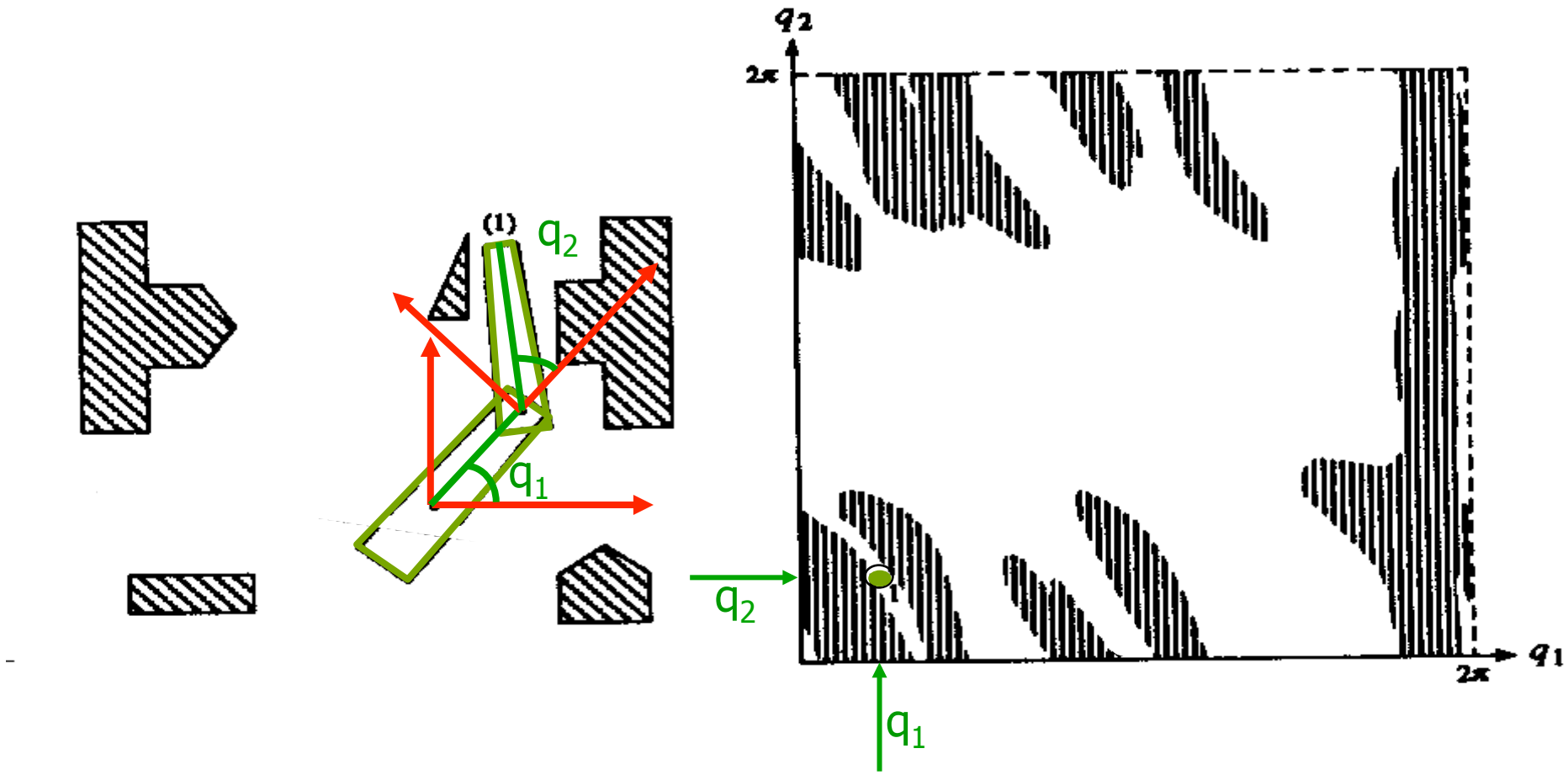
- A robot configuration is a specification of the positions of all robot points relative to a fixed coordinate system
- Usually a configuration is expressed as a "vector" of position/orientation parameters

What is a Path?





Tool: Configuration Space (C-Space C)



Probabilistic Roadmaps PRMs

The basic idea behind PRM is to take random samples from the configuration space of the robot, testing them for whether they are in the free space, and use a local planner to attempt to connect these configurations to other nearby configurations. The starting and goal configurations are added in, and a graph search algorithm is applied to the resulting graph to determine a path between the starting and goal configurations.

Kavraki, L. E.; Svestka, P.; Latombe, J.-C.; Overmars, M. H. (1996), "Probabilistic roadmaps for path planning in high-dimensional configuration spaces", IEEE Transactions on Robotics and Automation 12 (4): 566–580.

Rapidly-exploring Random Trees

- A point P in C is randomly chosen.
- The nearest vertex in the RRT is selected.
- A new edge is added from this vertex in the direction of P, at distance ε.
- The further the algorithm goes, the more space is covered.

Fundamental Problems In Robotics

- How to Go From A to B ? (Path Planning)
- What does the world looks like? (mapping)
  - sense from various positions
  - integrate measurements to produce map
  - assumes perfect knowledge of position
- Where am I in the world? (localization)
  - Sense
  - relate sensor readings to a world model
  - compute location relative to model
  - assumes a perfect world model
- Together, the above two are called SLAM
  (Simultaneous Localization and Mapping)

Introduction to Mapping

- What the world looks like?
- Knowledge representation
  - Robotics, AI, Vision
- Who is the end-user?
  - Human or Machine
- Ease of Path Planning
- **Uncertainty!**

Simultaneous Localization And Mapping

SLAM is the process of building a map of an environment while, at the same time, using that map to maintain the location of the robot.

- Problems for SLAM in large scale environments:
  - Controlling growth of uncertainty and complexity
  - Achieving autonomous exploration

Localization

- Tracking: Known initial position
- Global Localization: Unknown initial position
- Re-Localization: Incorrect known position
  - (kidnapped robot problem)

Uncertainty

Central to any real system!

Localization

Initial state
detects nothing:

Moves and
detects landmark:

Moves and
detects nothing:

Moves and
detects landmark:

Bayesian Filter

- Estimate state ***x*** from data Z
  - *What is the probability of the robot being at x?*
- ***x*** could be robot location, map information, locations of targets, etc…
- Z could be sensor readings such as range, actions, odometry from encoders, etc…)
- This is a general formalism that does not depend on the particular probability representation
- Bayes filter **recursively** computes the posterior distribution:

$$Bel(x_T) = P(x_T \mid Z_T)$$

Derivation of the Bayesian Filter

Estimation of the robot's state given the data:

$$Bel(x_t) = p(x_t \mid Z_T)$$

The robot's data, Z, is expanded into two types: observations o_i and actions a_i

$$Bel(x_t) = p(x_t \mid o_t, a_{t-1}, o_{t-1}, a_{t-2}, \ldots, o_0)$$

Invoking the Bayesian theorem

$$Bel(x_t) = \frac{p(o_t \mid x_t, a_{t-1}, \ldots, o_0)\, p(x_t \mid a_{t-1}, \ldots, o_0)}{p(o_t \mid a_{t-1}, \ldots, o_0)}$$

Derivation of the Bayesian Filter

Denominator is constant relative to x_t

$$\eta = 1 / p(o_t \mid a_{t-1}, \ldots, o_0)$$

$$Bel(x_t) = \eta p(o_t \mid x_t, a_{t-1}, \ldots, o_0) p(x_t \mid a_{t-1}, \ldots, o_0)$$

First-order Markov assumption shortens first term:

$$Bel(x_t) = \eta p(o_t \mid x_t) p(x_t \mid a_{t-1}, \ldots, o_0)$$

Expanding the last term (theorem of total probability):

$$Bel(x_t) = \eta p(o_t \mid x_t) \int p(x_t \mid x_{t-1}, a_{t-1}, \ldots, o_0) p(x_{t-1} \mid a_{t-1}, \ldots, o_0) dx_{t-1}$$

Derivation of the Bayesian Filter

First-order Markov assumption shortens middle term:

$$Bel(x_t) = \eta\, p(o_t \mid x_t) \int p(x_t \mid x_{t-1}, a_{t-1}) p(x_{t-1} \mid a_{t-1}, \ldots, o_0) dx_{t-1}$$

Finally, substituting the definition of $Bel(x_{t-1})$:

$$Bel(x_t) = \eta p(o_t \mid x_t) \int p(x_t \mid x_{t-1}, a_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

The above is the probability distribution that must be estimated from the robot's data

Iterating the Bayesian Filter

- Propagate the motion model:

$$Bel\_(x_t) = \int P(x_t \mid a_{t-1}, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

Compute the current state estimate before taking a sensor reading by integrating over all possible previous state estimates and applying the motion model

- Update the sensor model:

$$Bel(x_t) = \eta P(o_t \mid x_t) Bel\_(x_t)$$

Compute the current state estimate by taking a sensor reading and multiplying by the current estimate based on the most recent motion history

Different Approaches

Kalman filters (Early-60s?)
- Gaussians
- approximately linear models
- position tracking

Extended Kalman Filter
Information Filter
Unscented Kalman Filter

Multi-hypothesis ('00)
- Mixture of Gaussians
- Multiple Kalman filters
- Global localization, recovery

Discrete approaches ('95)
- Topological representation ('95)
- Uncertainty handling (POMDPs)
- occas. global localization, recovery
- Grid-based, metric representation ('96)
- global localization, recovery

Particle filters ('98)
- Condensation (Isard and Blake '98)
- Sample-based representation
- Global localization, recovery
- Rao-Blackwellized Particle Filter

Bayesian Filter : Requirements for Implementation

- Representation for the belief function
- Update equations
- Motion model
- Sensor model
- Initial belief state