COS 495 - Lecture 2
Autonomous Robot Navigation

Instructor: Chris Clark

Semester: Fall 2011

Figures courtesy of Siegwart & Nourbakhsh

Navigation and Control

1. Control Architectures
2. Navigation Example
3. Basic Tools for AUV Navigation

Control Architectures

- Today, most robots control systems have a mixture of planning and behavior-based control strategies.
- To implement these strategies, a *control architecture* is used.
- Control architectures should be:
  - Modular
  - Localized

Control Architectures Desired Characteristics

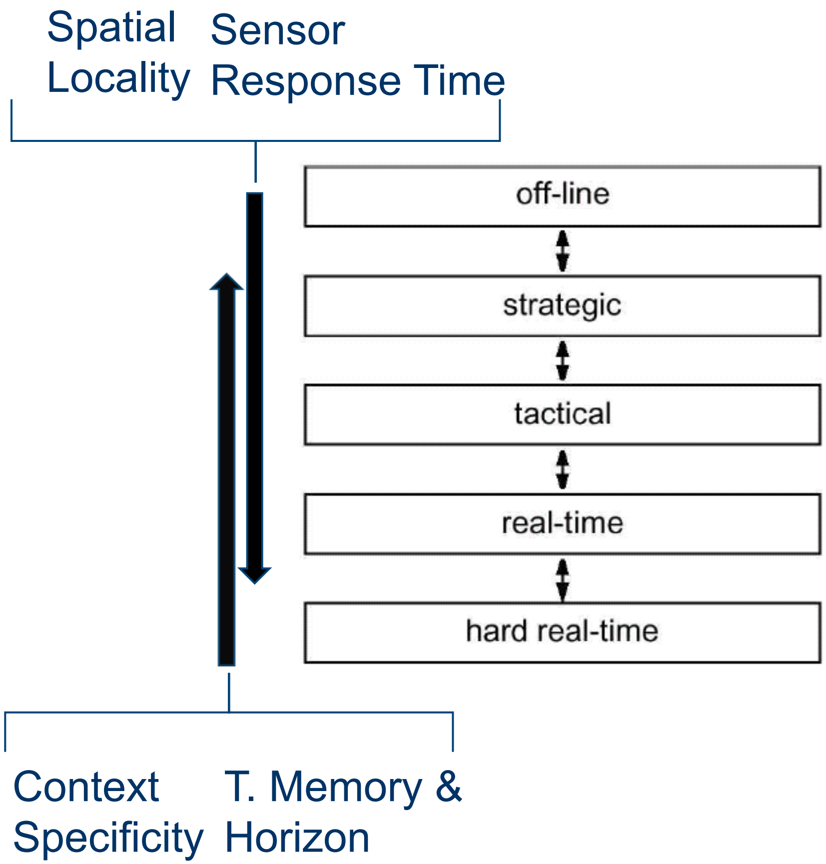
- Code Modularity
  - Allows programmers to interchange environment types sensors, path planners, propulsion, etc.
- Localization
  - Embed specific navigation functions within modules to allow different levels of control (e.g. from task planning to wheel velocity control)

Control Architectures Decomposition

- Decomposition allows us to modularize our control system based on different axes:

1. Temporal Decomposition
   - Facilitates varying degrees of real-time processes
2. Control Decomposition
   - Defines how modules should interact: serial or parallel?

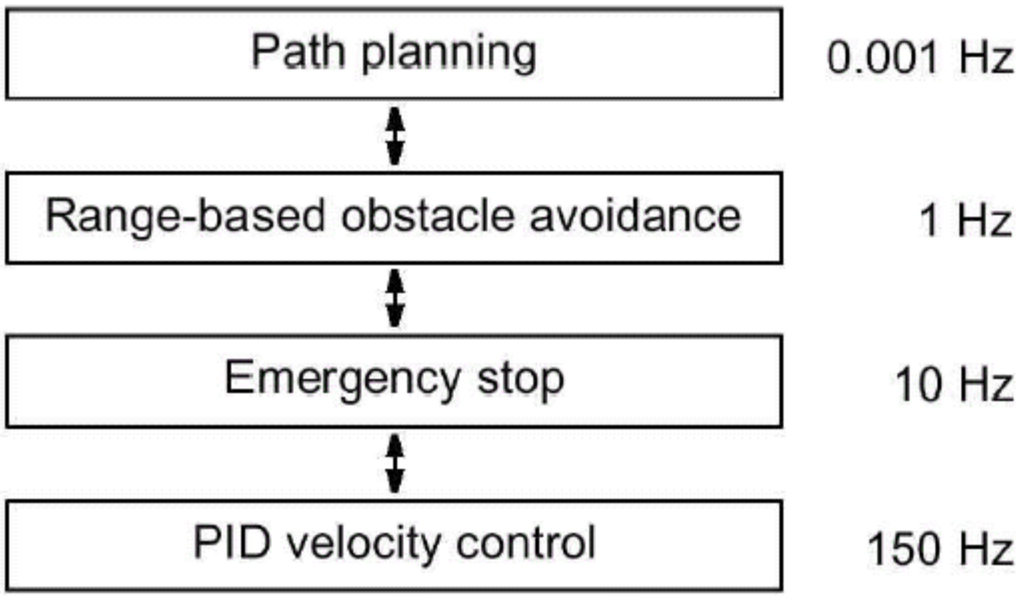
Control Architectures Temporal Decomposition



- Factors affecting temporal decomposition:
  - Sensor response time
  - Temporal memory and horizon
  - Spatial Locality
  - Context Specificity

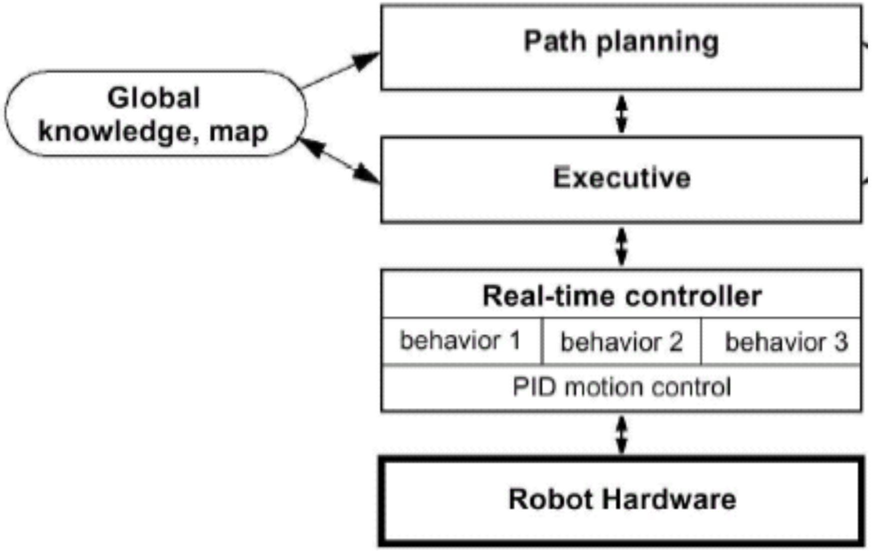
Control Architectures Temporal Decomposition

- Example



Control Architectures
Tiered Architectures

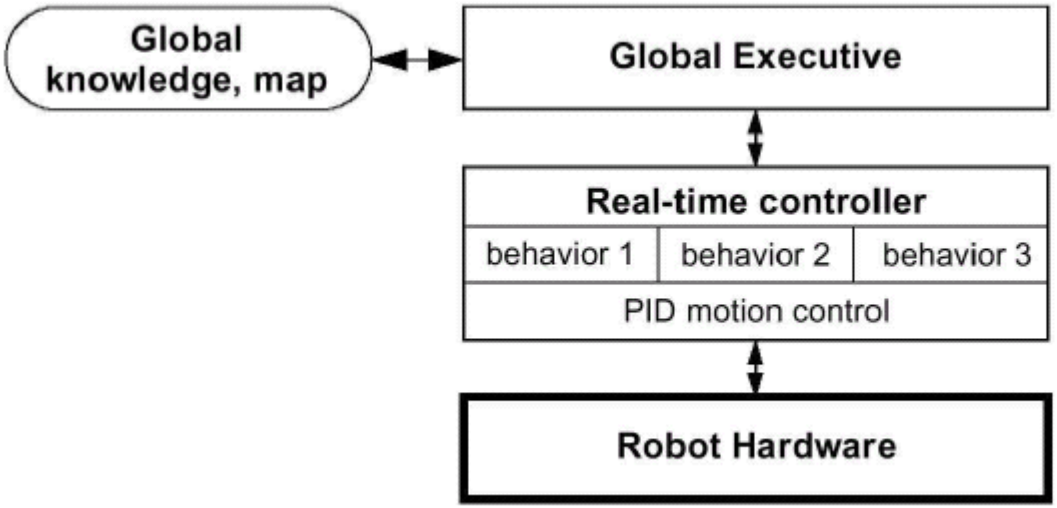
- A general tiered architecture for *episodic* planning
- Role of the Executive is:
  - Switch behaviors
  - Monitor failures
  - Call the planner



- Planning only when required (e.g. blockage)

Control Architectures
Tiered Architectures

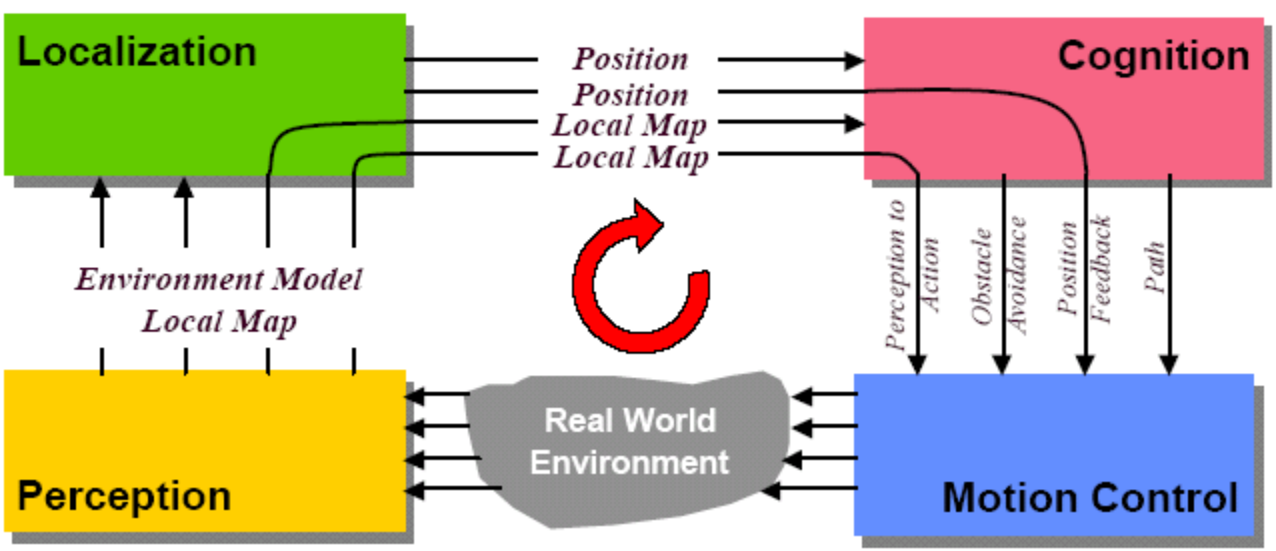
- A tiered architecture for *integrated* planning



- Planning is fast and is embedded as a behavior.

Control Architectures
Control Decomposition

- An example of a control decomposition using a mixture of serial and parallel approaches.

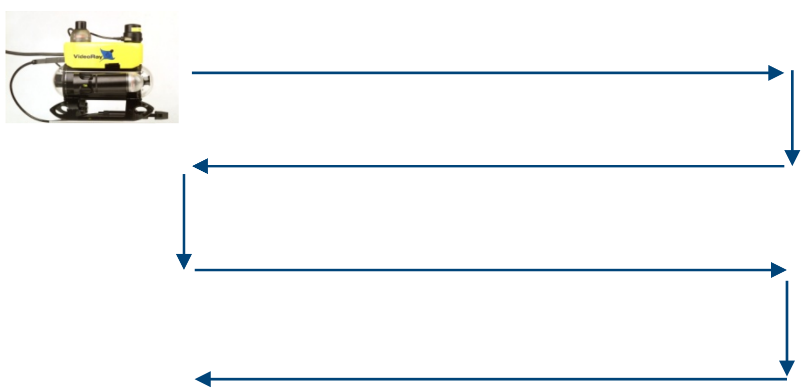


Navigation and Control

1. Control Architectures
2. Navigation Example
    1. Motion Modeling
    2. Estimation and Control
    3. Experiments
3. Basic Tools for AUV Navigation

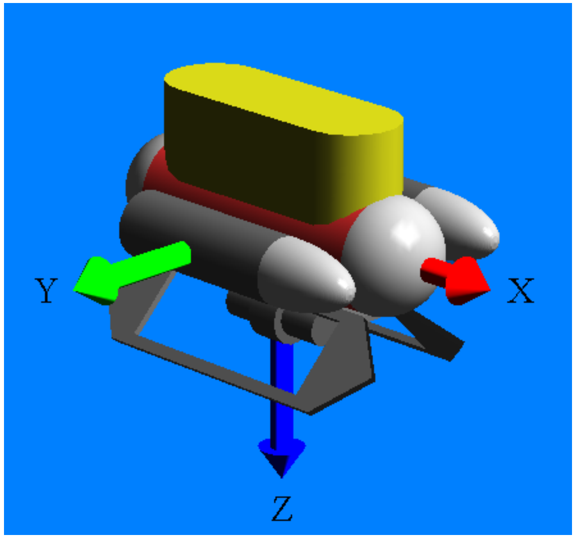
Navigation Example

- Autonomous Control
  - Goal is to enable autonomous trajectory tracking capabilities.
  - Given individual ROVs have autonomous control, multi-vehicle control research will be facilitated.



Equations of Motion

- 6 degrees of freedom (DOF):
- State vectors:
  body-fixed velocity vector:
  earth-fixed pos. vector:



DOF	Surge	Sway	Heave	Roll	Pitch	Yaw
Velocities	u	v	w	p	q	r
Position & Attitude	x	y	z	ϕ	θ	ψ
Forces & Moments	X	Y	Z	K	M	N

Equations of Motion

- The 6-DOF nonlinear dynamic equations of motion can be expressed as:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau$$

$$\dot{\eta} = J(\eta)\nu$$

where:

inertia matrix:	$M = M_{RB} + M_A$
Coriolis & centripetal matrix:	$C(\nu) = C_{RB}(\nu) + C_A(\nu)$
hydrodynamic damping:	$D(\nu)$
restoring forces:	$g(\eta)$
propulsion forces:	τ

Equations of Motion

- Initial Assumptions:
  - The ROV will usually move with low velocity when on mission
  - Almost three planes of symmetry;
  - Vehicle is assumed to be performing non-coupled motions.
  - Horizontal Plane:

$$\begin{aligned} m_{11}\dot{u} &= -m_{22}vr + X_u u + X_{u|u|}u|u| + X \\ m_{22}\dot{v} &= m_{11}ur + Y_v v + Y_{v|v|}v|v|, \\ I\dot{r} &= N_r r + N_{r|r|}r|r| + N, \end{aligned}$$

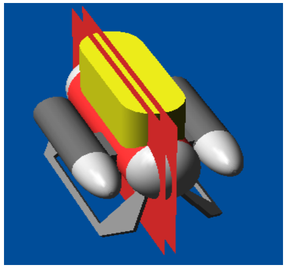
  - Vertical Plan:

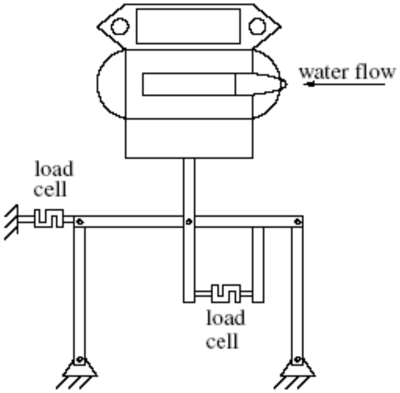
$$m_{33}\dot{w} = Z_w w + Z_{w|w|}w|w| + Z$$

[W. Wang et al., 2006]

Theory vs. Experiment

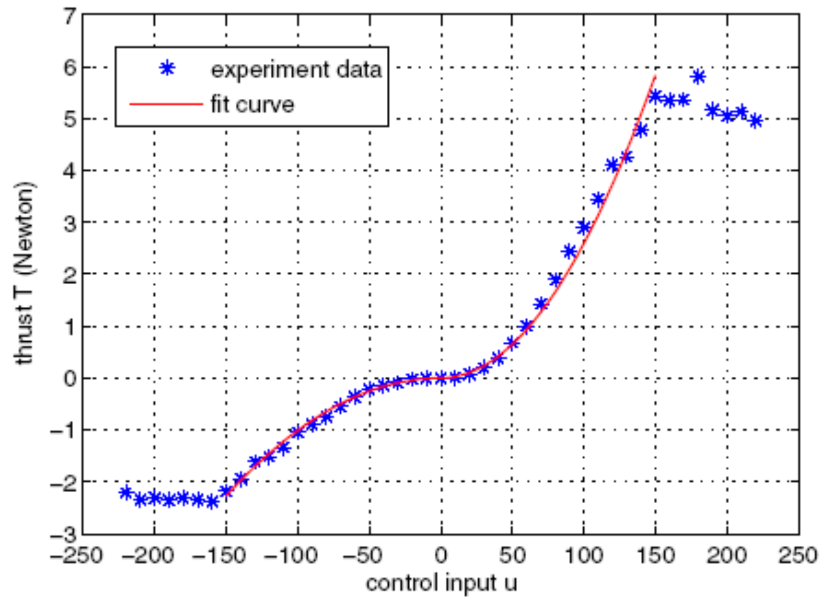
- Coefficients for the dynamic model are pre-calculated using strip theory;
- A series of tests are carried out to validate the hydrodynamic coefficients, including
  - Propeller mapping
  - Added mass coefficients
  - Damping coefficients





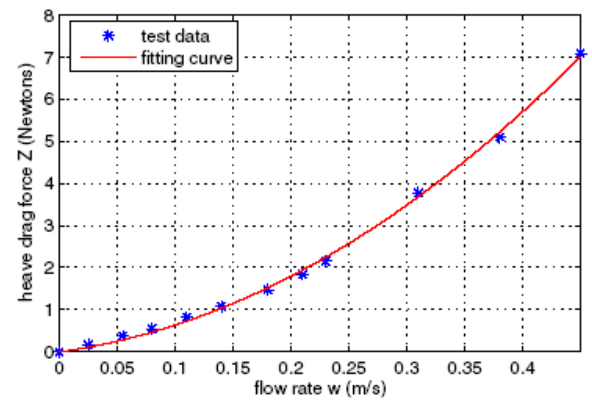
Propeller Thrust Mapping

- The forward thrust can be represented as:

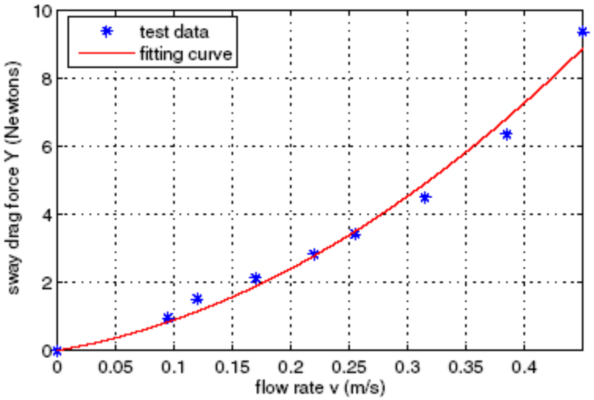


Direct Drag Forces

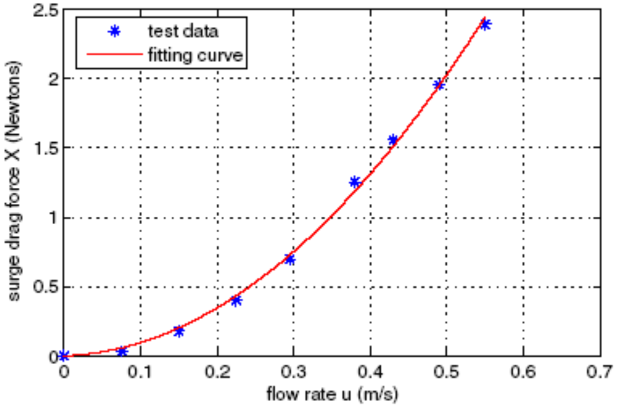
Drag in Heave Direction



Drag in Sway Direction



Drag in Surge Direction

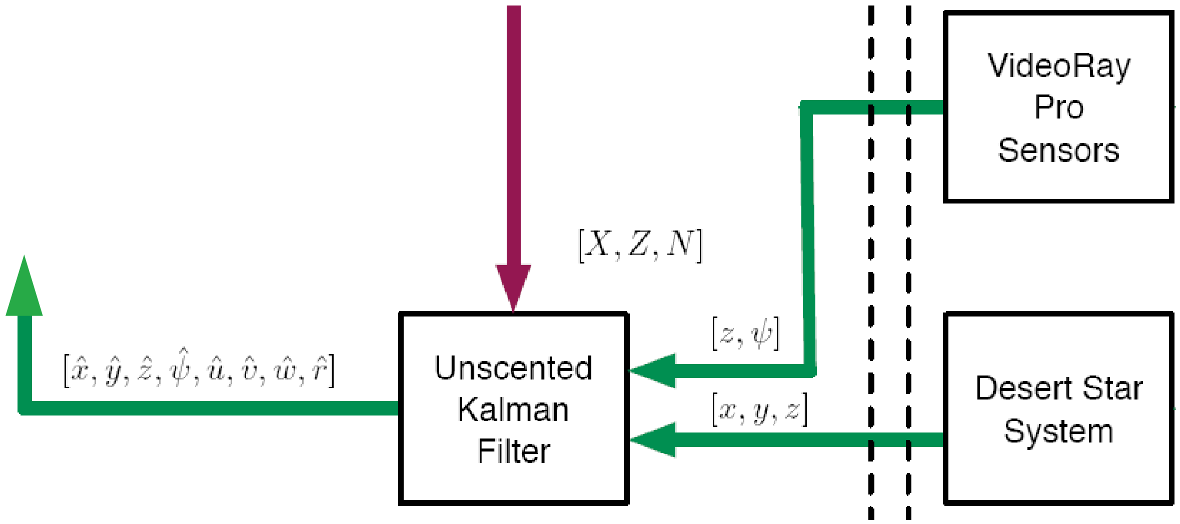


Estimation & Control

- Sensor Overview
  - VideoRay Compass
  - VideoRay Depth Sensor
  - Desert Star Acoustic Positioning System

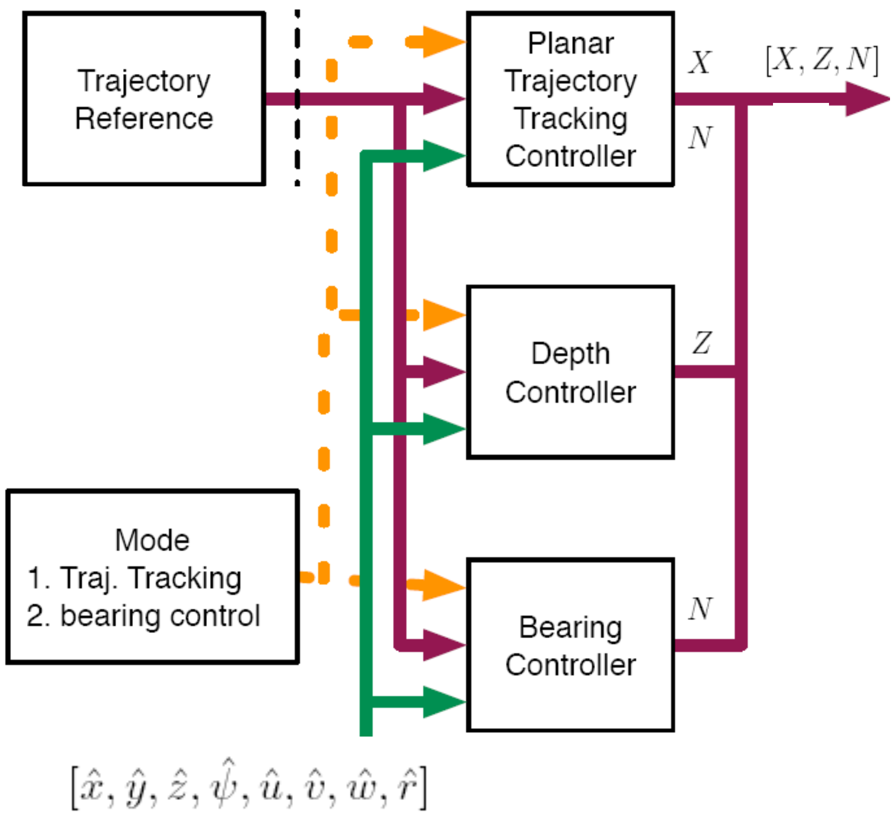
Estimation & Control

- State Estimation
  - We fuse several sensor measurements using an Unscented Kalman Filter:

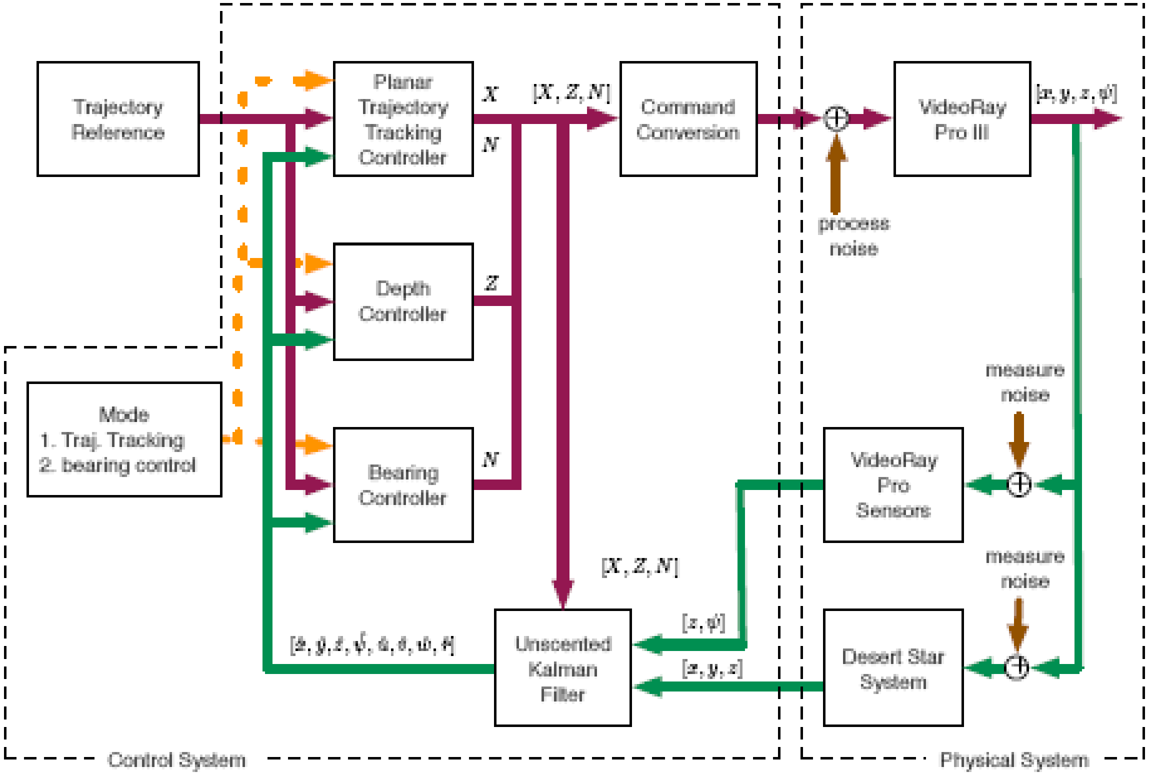


Estimation & Control

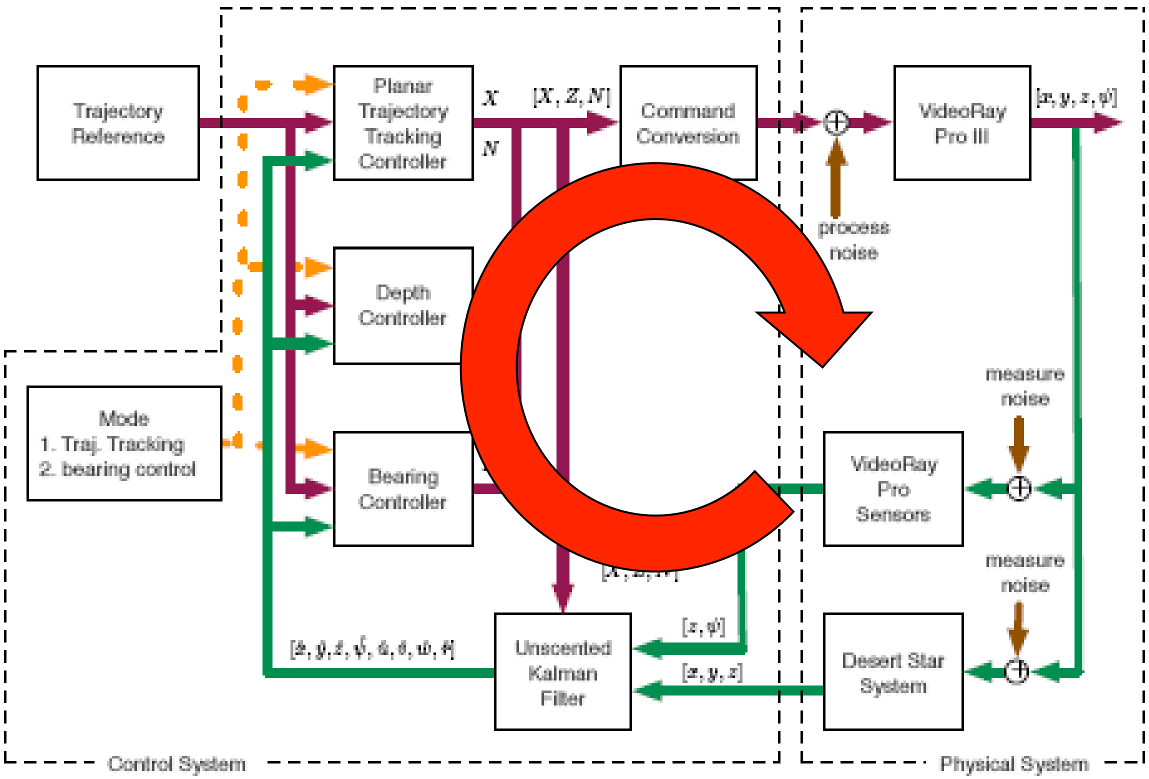
- Trajectory Tracking
  - We use three different controllers:



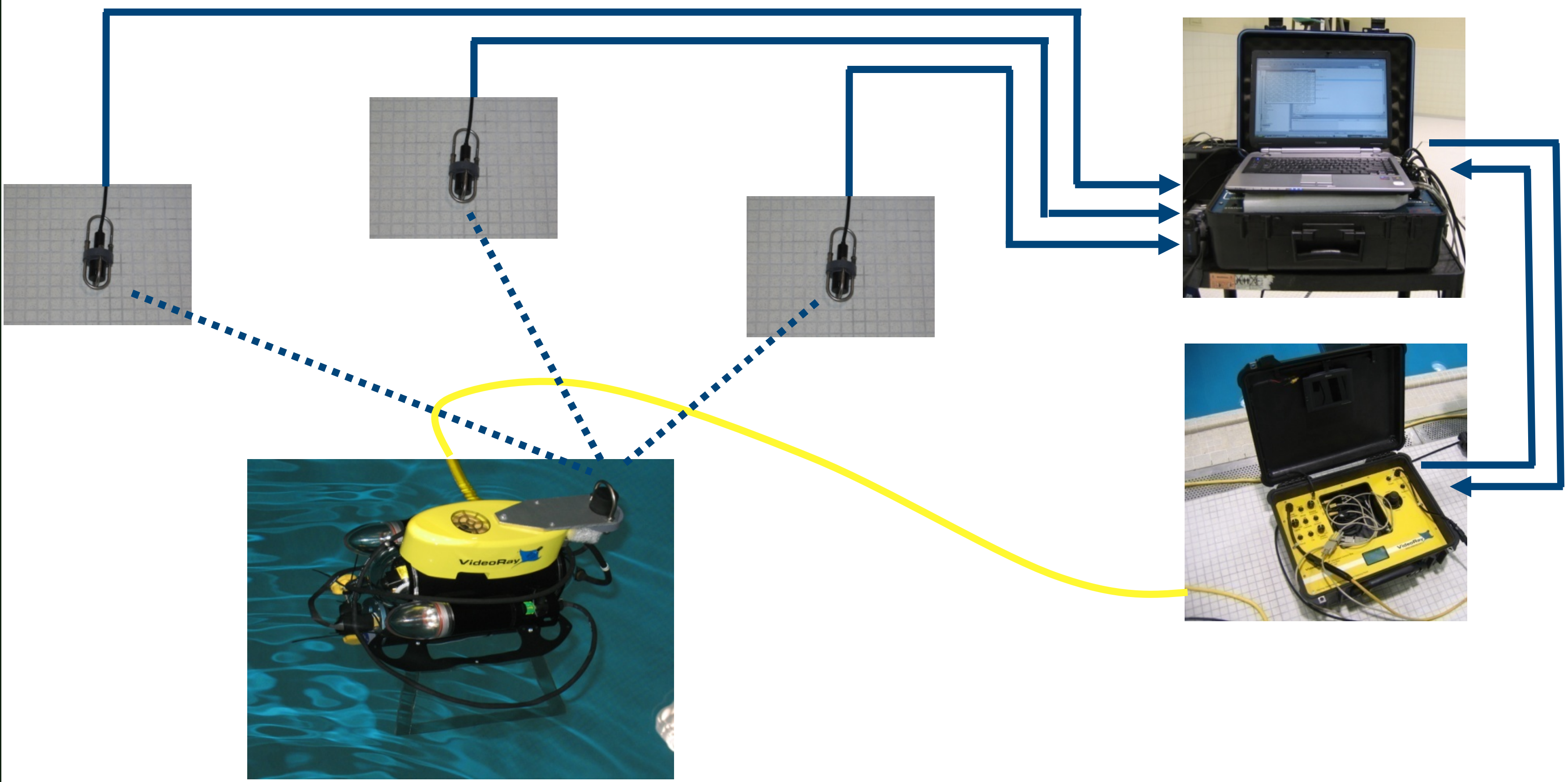
Estimation & Control



Estimation & Control



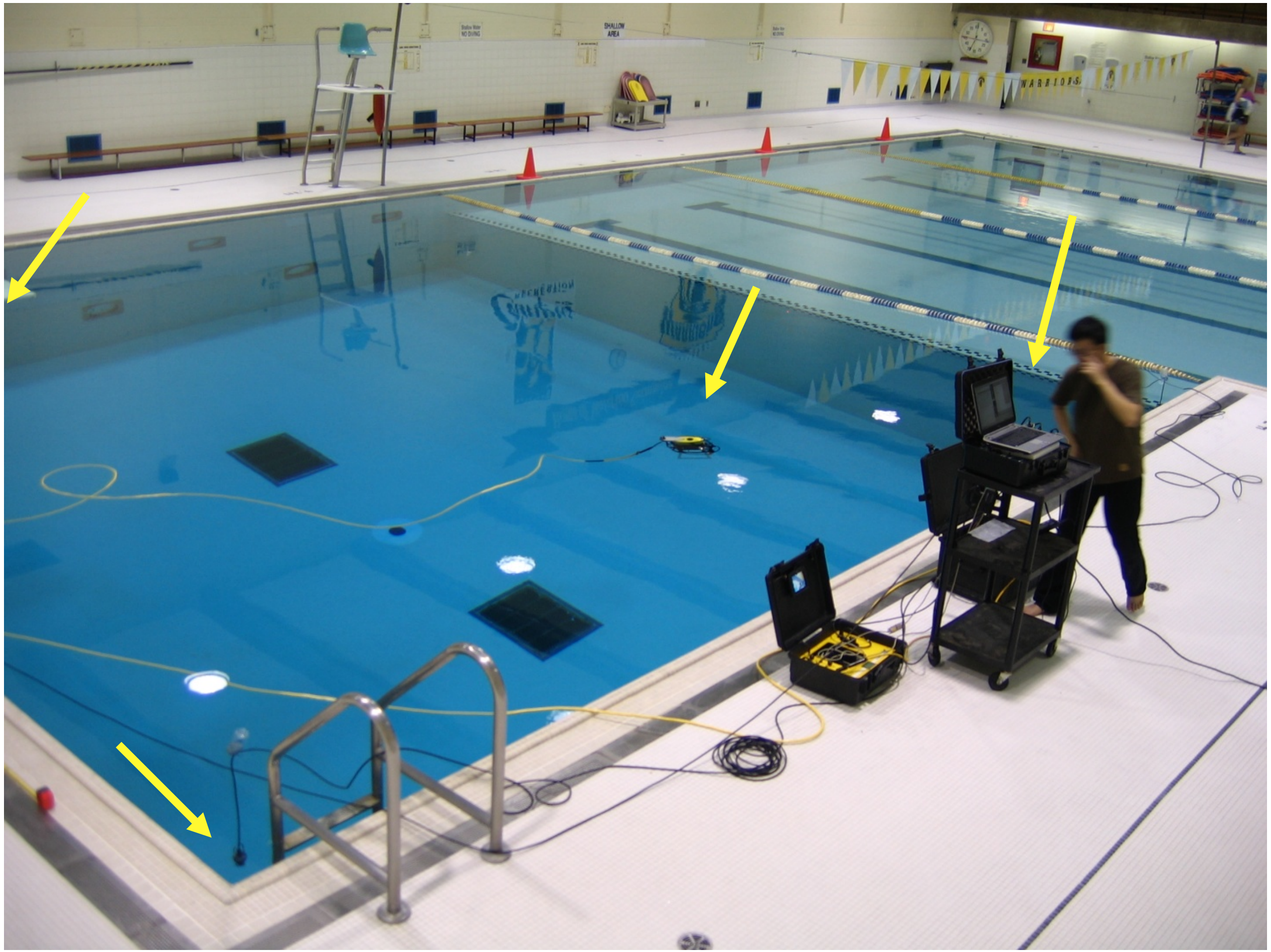
Autonomous Control



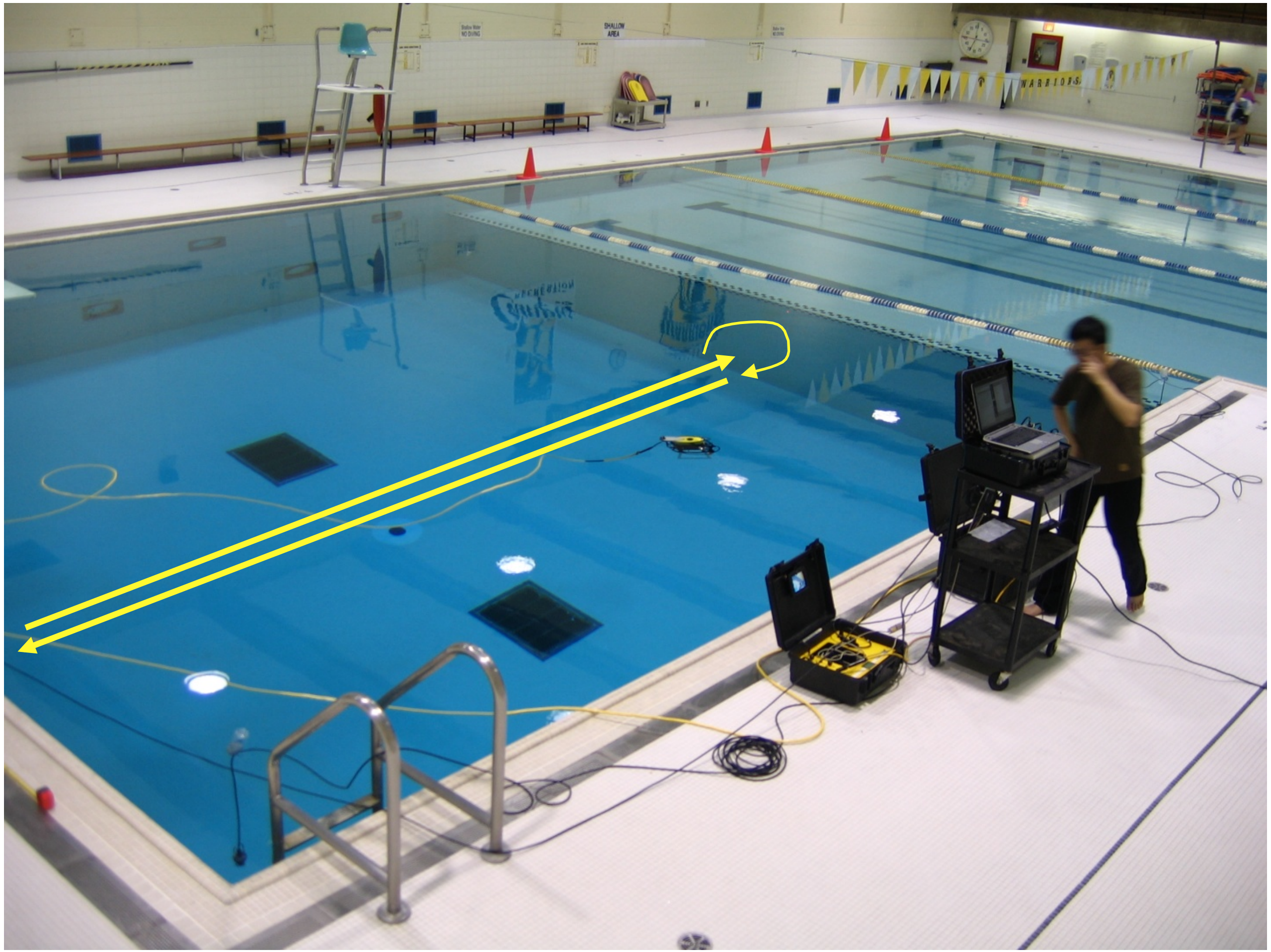
Autonomous Control

- Hardware Modifications
  - Added transceiver
  - Added bouyancy
  - Shifted weight
  - Extended feet

Autonomous Control



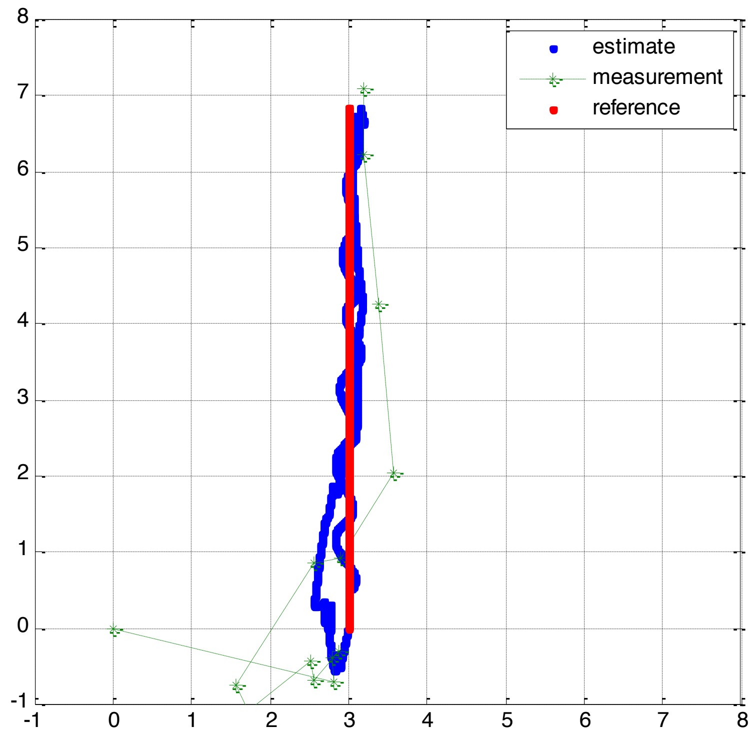
Autonomous Control



Autonomous Control

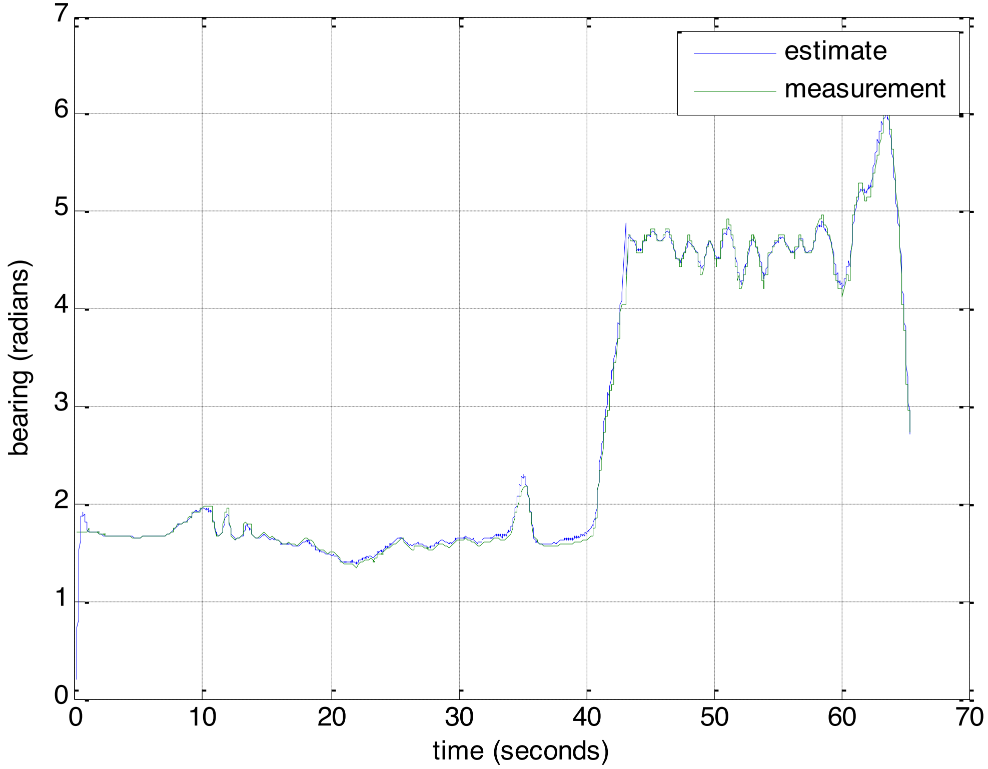
Swimming Pool Experiments

- Sample run: x, y state estimates



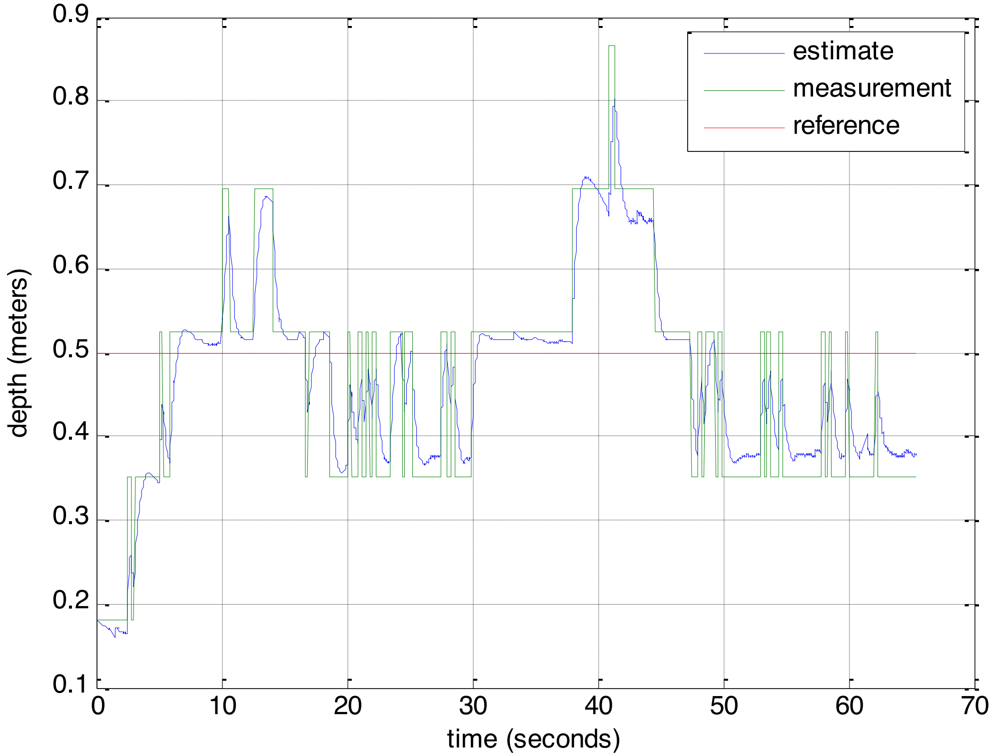
Swimming Pool Experiments

- Sample run: bearing state estimates



Swimming Pool Experiments

- Sample run: depth state estimates



Swimming Pool Experiments

Trial #	Mean of x (m)	Standard Deviation (m)
1	3.186	0.431
2	2.906	0.495
3	3.095	0.129
4	3.040	0.137
5	3.192	0.179
6	2.890	0.265
7	2.966	0.154

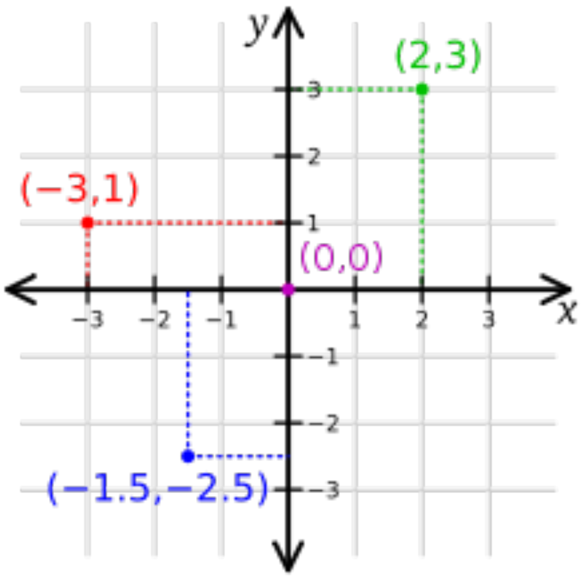
[W. Wang et al., 2006]

Navigation and Control

1. Control Architectures
2. Navigation Example 2
3. Basic Tools for AUV Navigation
    1. Coordinate Frames
    2. Motion Modeling
    3. P Control

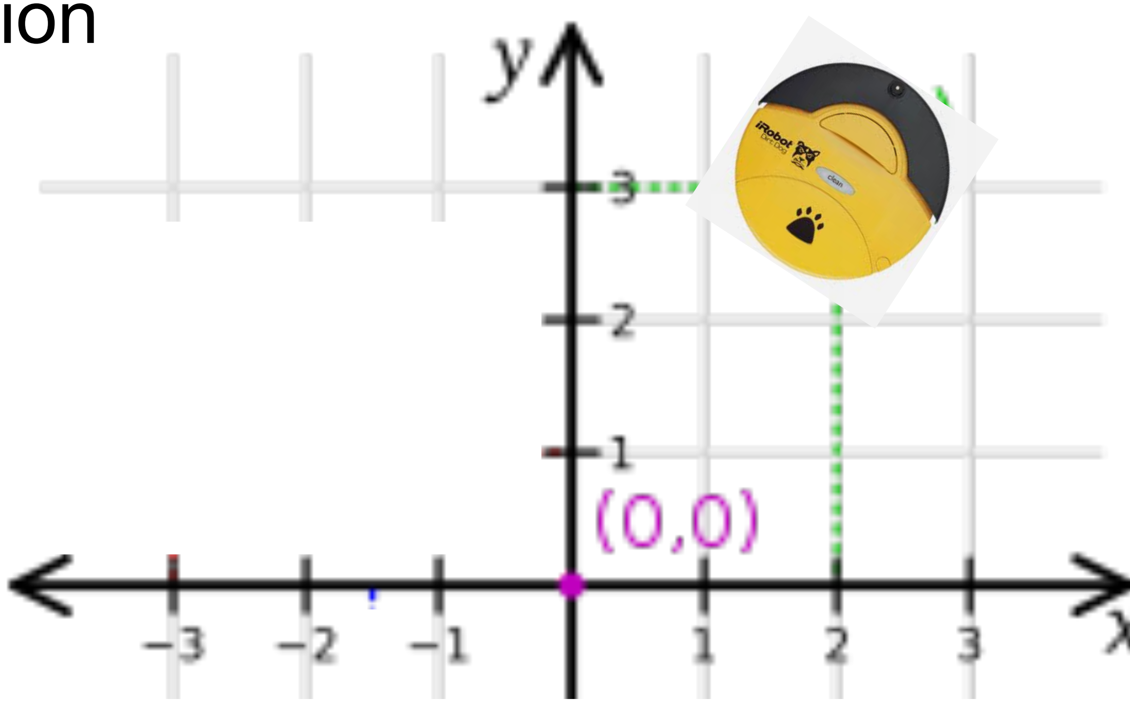
Cartesian Coordinates

- Describes unique position of points in a plane with respect to the axis
- For each dimension there is 1 axis
- Coordinates are measured in “units” in the direction parallel to the axis
- The origin is fixed to the plane



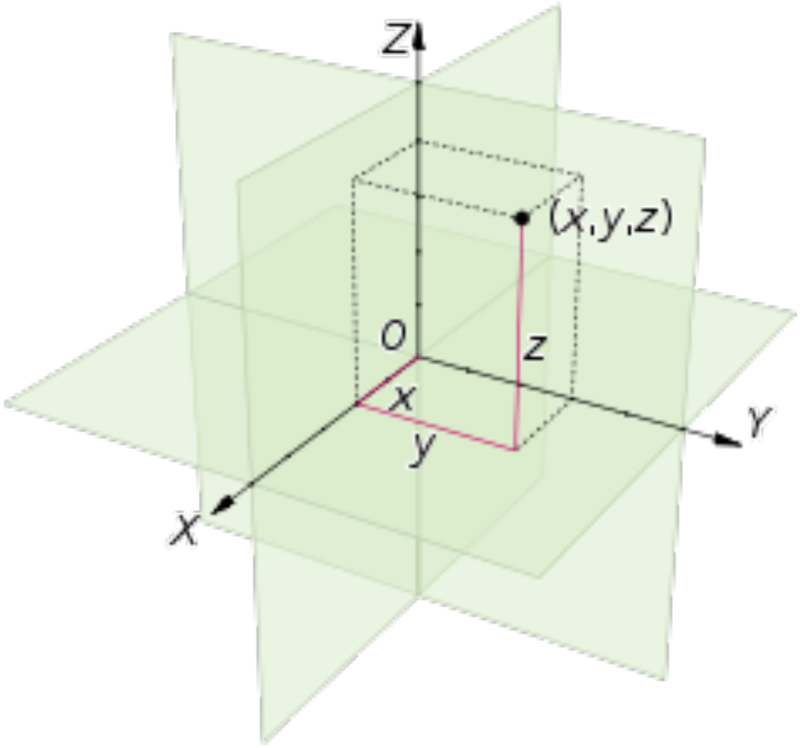
Cartesian Coordinates

- One can use cartesian coordinates to describe a robot's position



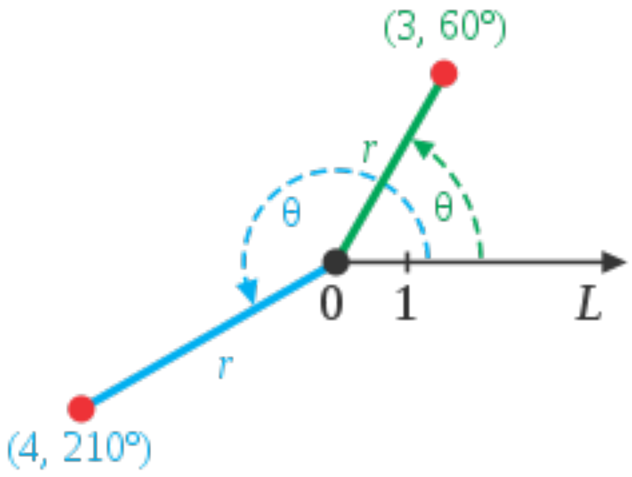
Cartesian Coordinates

- For our underwater robots, we need 3 degrees of freedom to express position



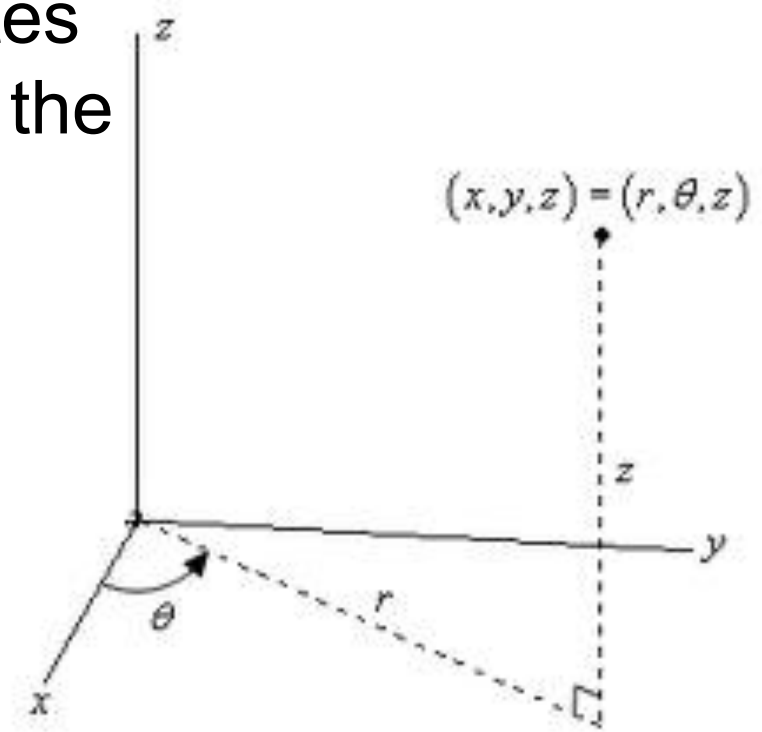
Polar Coordinates

- In polar coordinates, we specify points on a 2D plane using the length of a radius arm and an angle



Cylindrical Coordinates

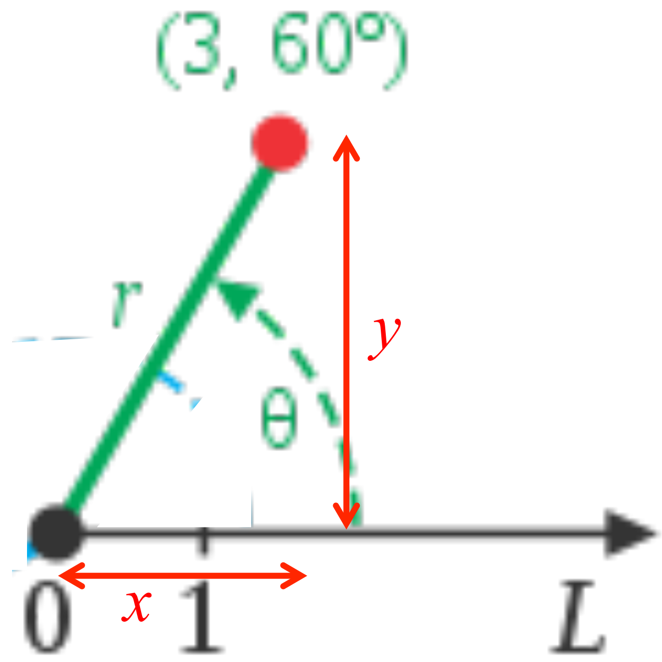
- For specifying point locations in 3D, cylindrical coordinates can be used by specifying the length of a radius arm, an angle, and a height.



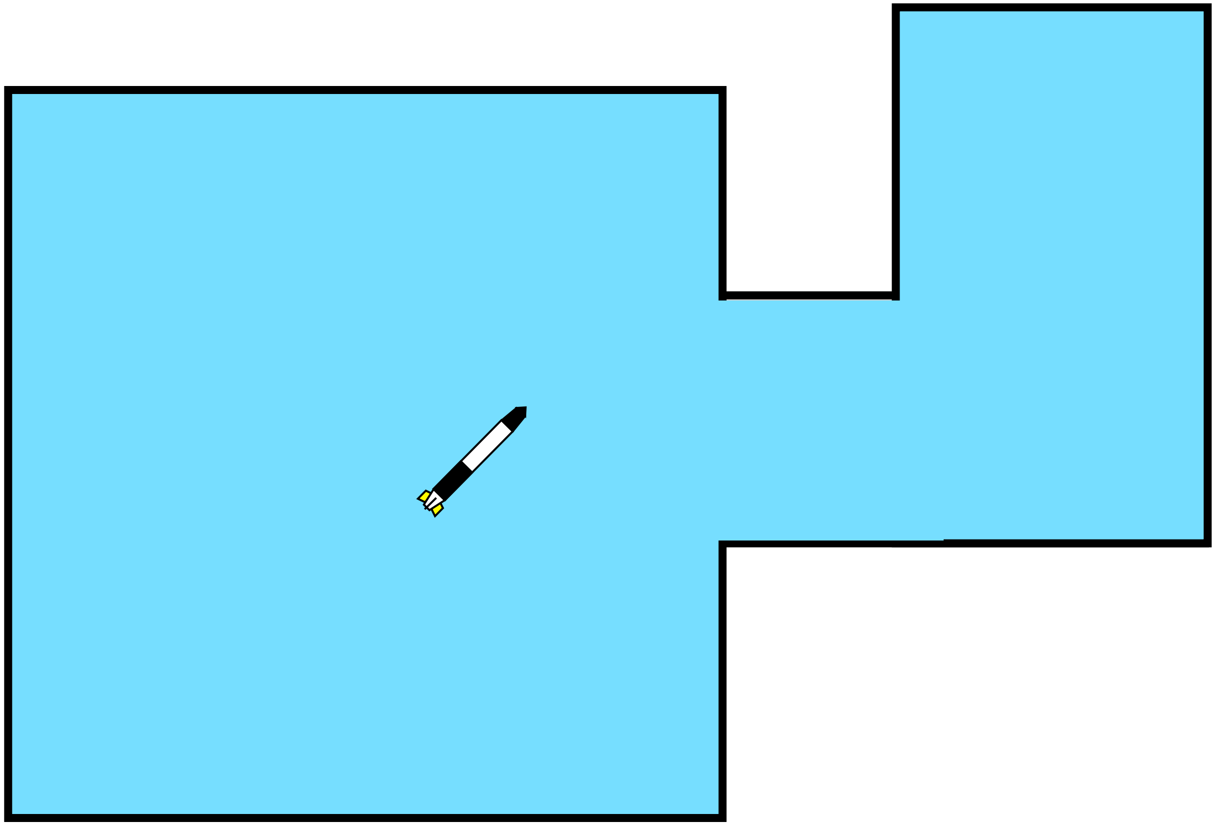
Polar to Cartesian

- How do we convert from polar coordinates to Cartesian coordinates?

$$x = r\cos(\theta)$$
$$y = r\sin(\theta)$$

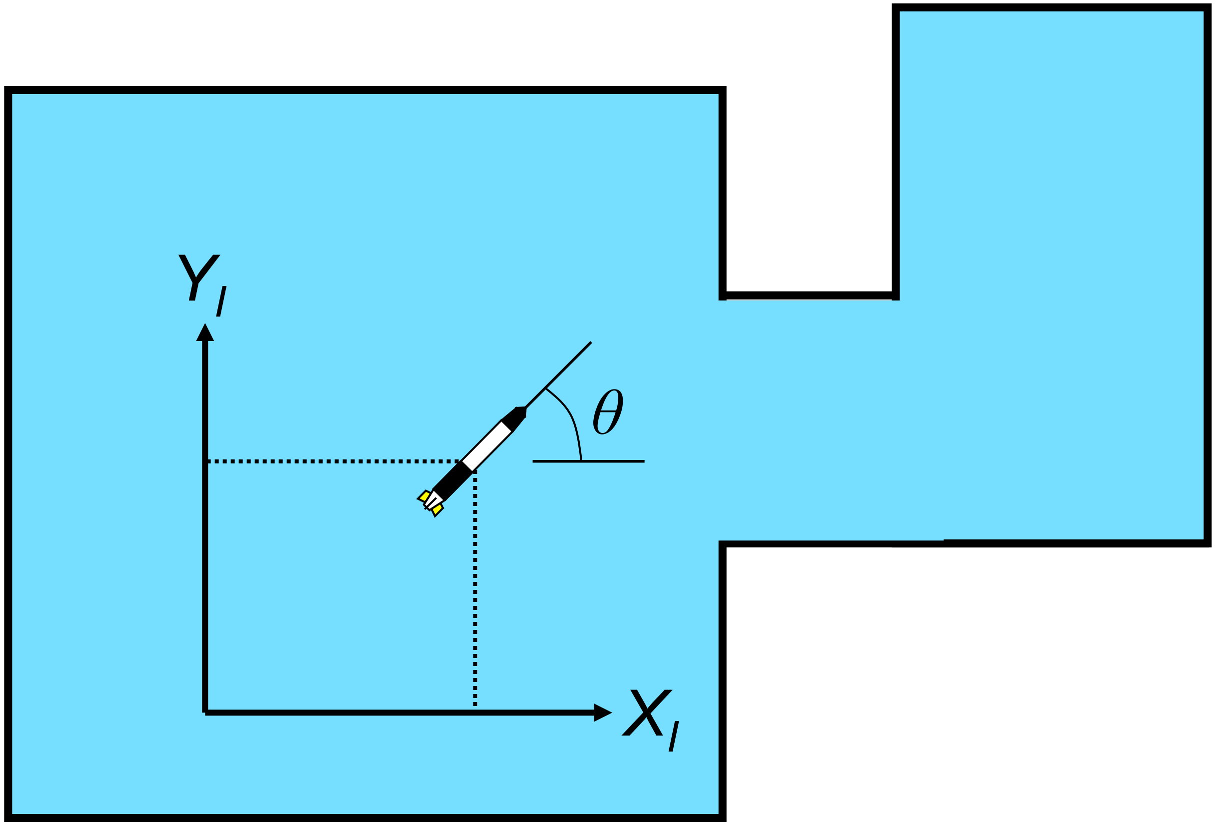


Global (Inertial) Coordinate frame



Global (Inertial) Coordinate frame

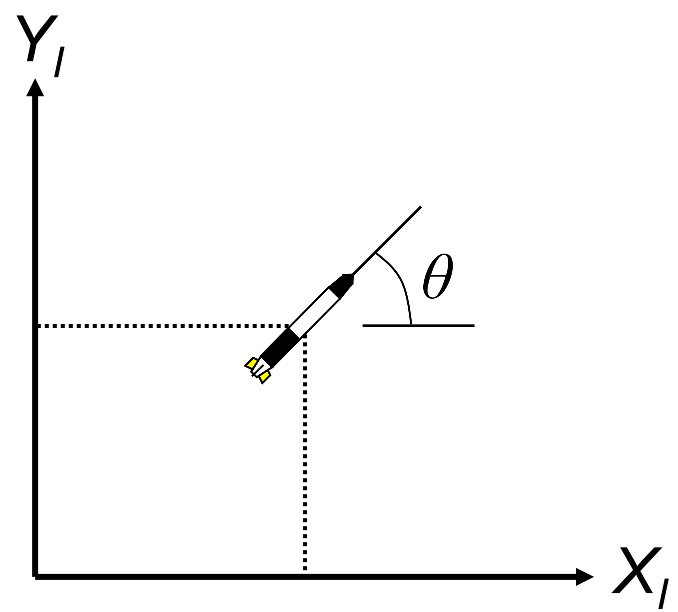
- Anchor a coordinate frame to the environment



Global (Inertial) Coordinate frame

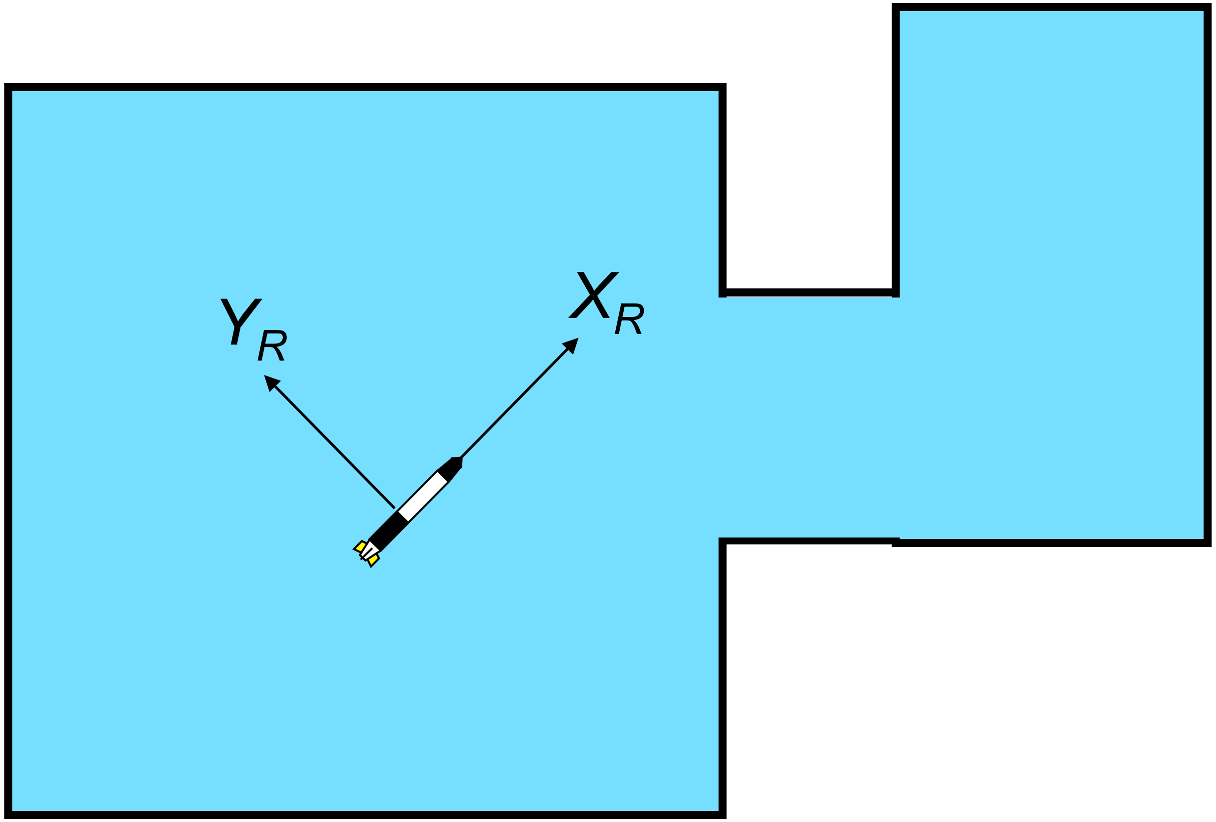
- With this coordinate frame, we describe the robot state as:

$$X_I = [x\ y\ \theta]_I$$



Local Coordinate frame

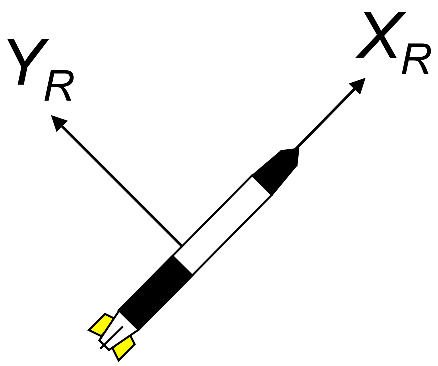
- Anchor a coordinate frame to the robot



Local Coordinate frame

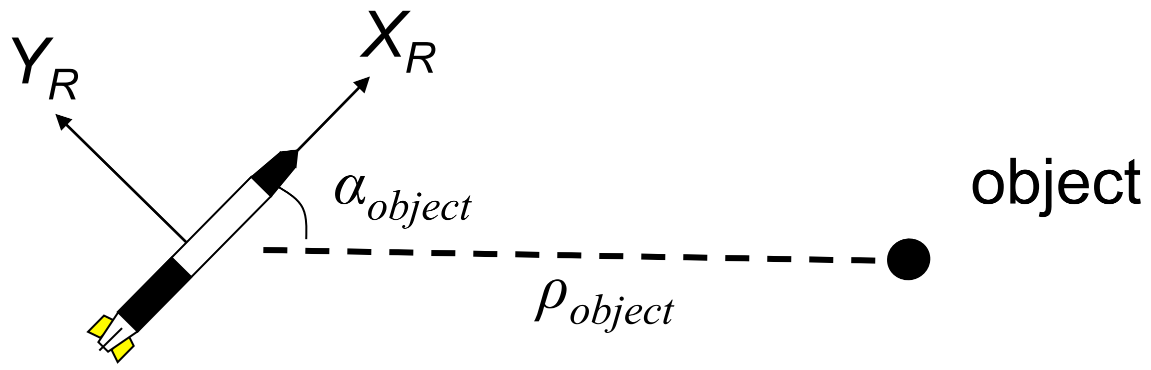
- With this coordinate frame, we describe the robot state as:

$$X_R = [x\ y\ \theta]_R = [0\ 0\ 0]$$



Local Coordinate frame

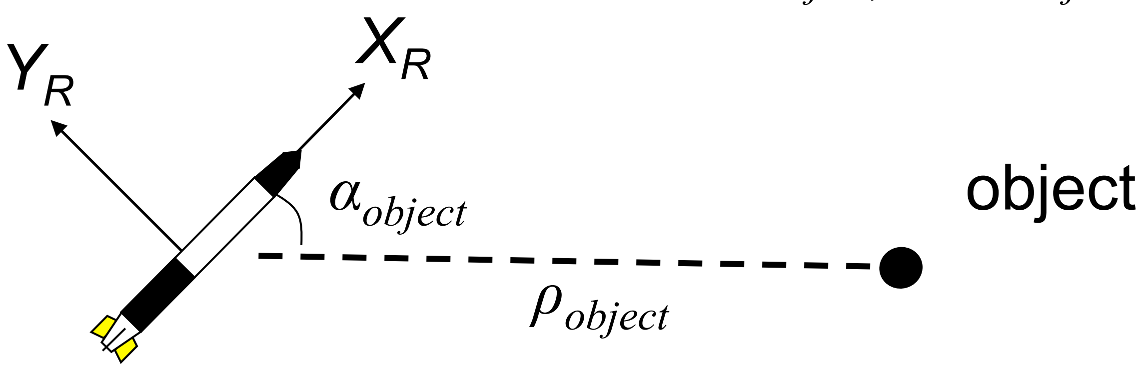
- The local frame is useful when considering taking measurements of environment objects.
  - Consider the detection of an wall using a range finder:



Local Coordinate frame

- The measurement is taken relative to the robot's local coordinate frame $(\rho_{object}, \alpha_{object})$
- We can calculate the position of the measurement in local coordinate frames:

$$x_{object,\,R} = \rho_{object}\cos(\alpha_{object})$$
$$y_{object,\,R} = \rho_{object}\sin(\alpha_{object})$$



Local Coordinate frame

- One can calculate the position of the object in the global coordinate frame ($x_{object,I}$, $y_{object,I}$)

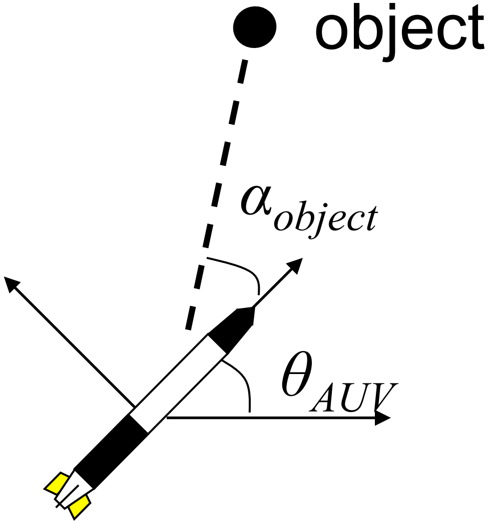
$$x_{object,\,I} = x_{AUV,\,I} + (x_{object,R} \cos(\theta_{AUV,\,I})) + (y_{object,R} \sin(\theta_{AUV,\,I}))$$

$$y_{object,\,I} = y_{AUV,\,I} + (x_{object,R} \sin(\theta_{AUV,\,I})) + (y_{object,R} \cos(\theta_{AUV,\,I}))$$

Local Coordinate frame

- One can calculate α_{object} if the positions are known

$$\theta_{AUV,\,I} + \alpha_{object} = atan2(\,y_{object} - y_{AUV},\, x_{object} - x_{AUV})$$

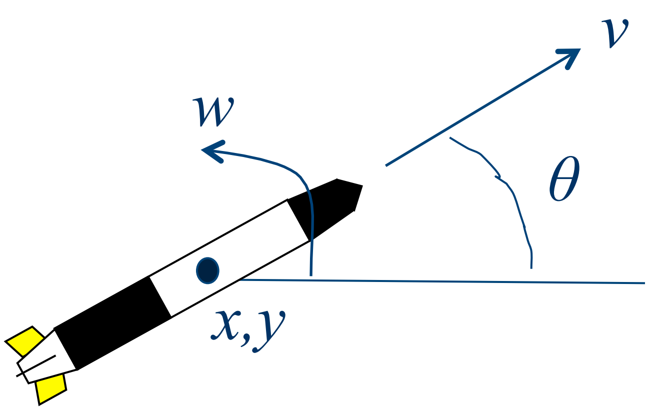


Navigation and Control

1. Control Architectures
2. Navigation Example 2
3. Basic Tools for AUV Navigation
    1. Coordinate Frames
    2. Motion Modeling
    3. P Control

A (simple) motion model

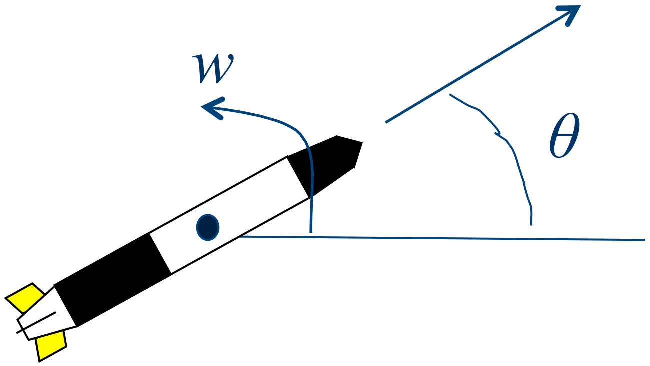
- Consider a robot moving from position x, y in direction θ radians with forward velocity v m/s and rotational velocity w rad/s.



A (simple) motion model

- How much will it rotate in t seconds?

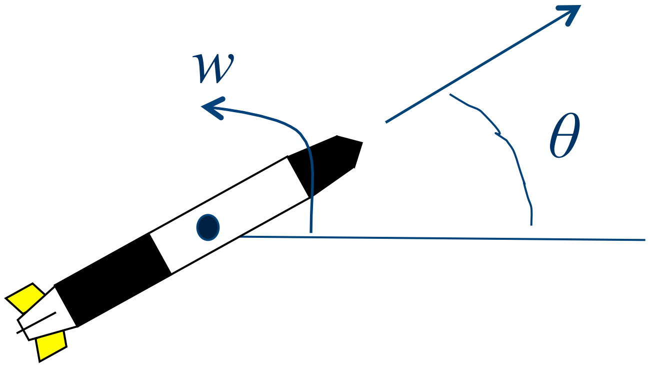
It will rotate a distance of wt radians.



A (simple) motion model

- So, from time step k to time step $k+1$, the angle changes to:

$$\theta_{k+1} = \theta_k + wt$$



A (simple) motion model

- So, from time step k to time step $k+1$, the position changes to:

$$x_{k+1} = x_k + vt\,cos(\,\theta_k)$$
$$y_{k+1} = y_k + vt\,sin(\,\theta_k)$$

Navigation and Control

1. Control Architectures
2. Navigation Example 2
3. Basic Tools for AUV Navigation
    1. Coordinate Frames
    2. Motion Modeling
    3. P Control

P Control

- Proportional Feedback Control – P Control – uses the error between the desired and measured state to determine the control signal.
- If $x_{desired}$ is the desired state, and x is the actual state, we define the error as:

$$e = x_{desired} - x$$

P Control

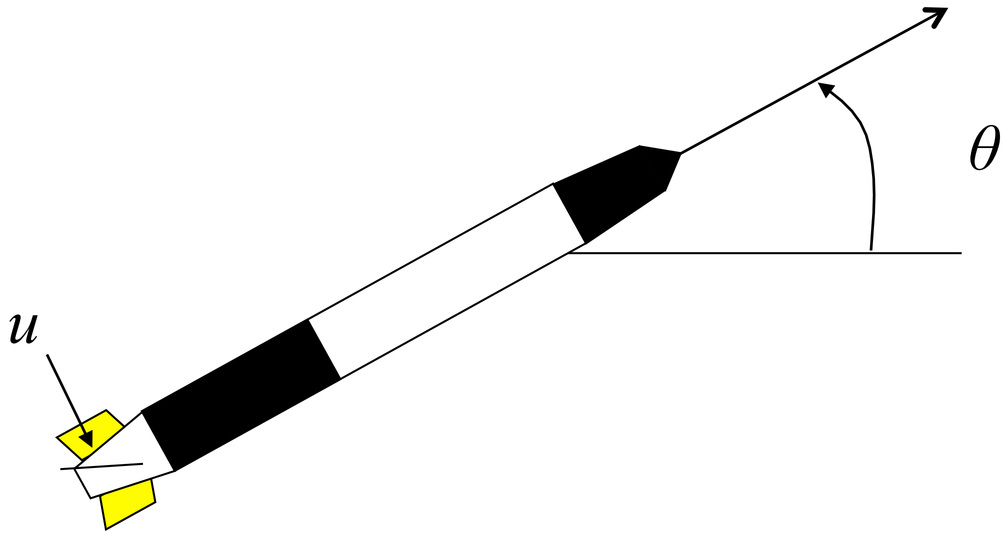
- The control signal u is calculated as

$$u = K_P e$$

where K_P is called the proportional gain.

P Control

- Example:
  - Consider the orientation control of an AUV. Assume the orientation is completely controlled by the rear rudder fins.



P Control

- Example cont':
  - The control signal u is calculated as

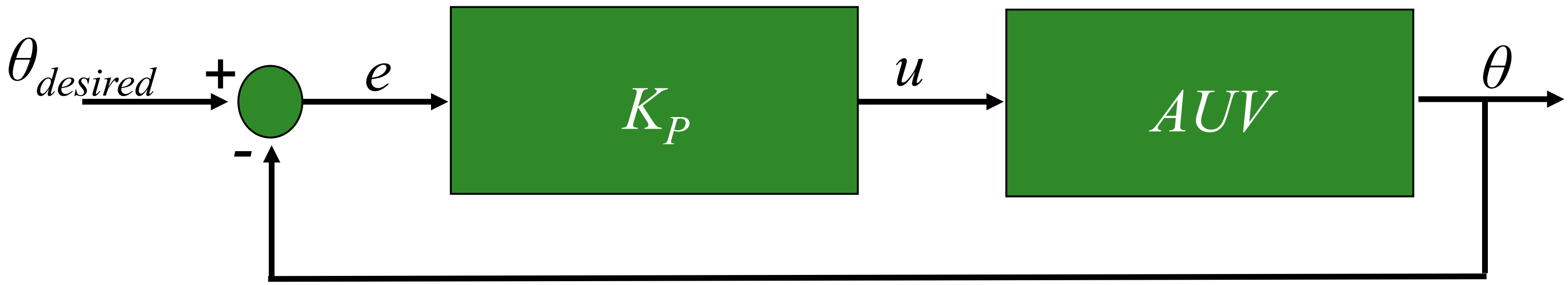
$$u = K_P(\theta_{desired} - \theta)$$

  - Notes:
    - If $\theta_{desired} = \theta$, the control signal is 0.
    - If $\theta_{desired} < \theta$, the control signal is negative, resulting in an decrease in θ.
    - If $\theta_{desired} > \theta$, the control signal is positive, resulting in an increase in θ.
    - The magnitude of the increase/decrease depends on K_p

P Control

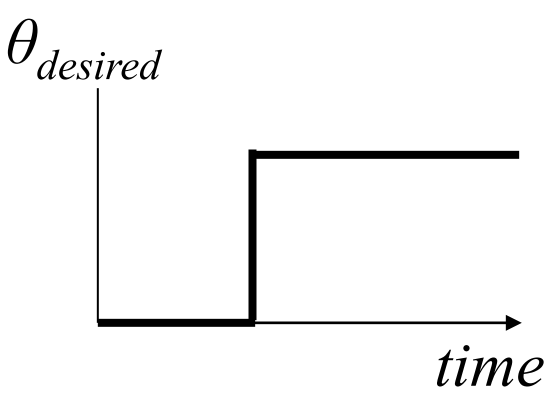
- Block Diagram:

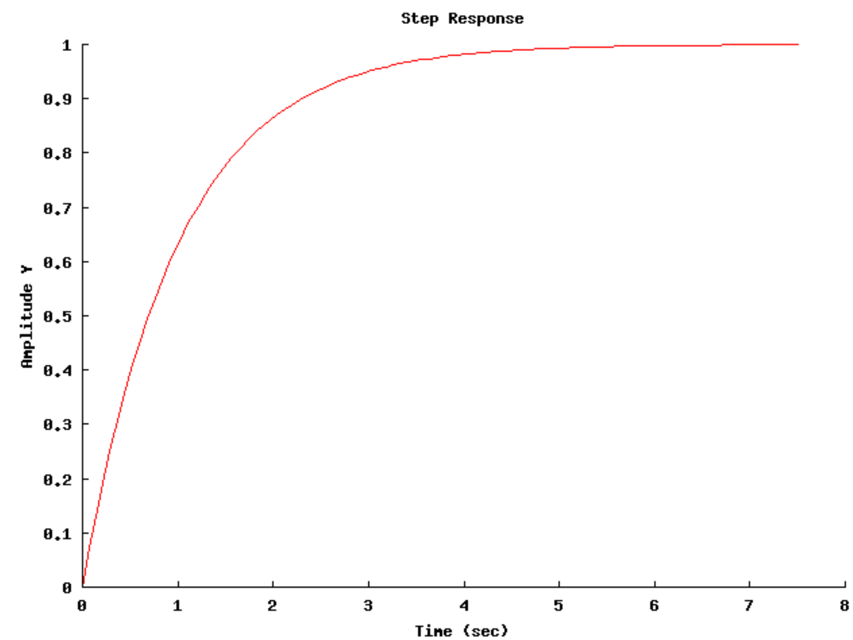
$$u = K_P(\theta_{desired} - \theta)$$



P Control

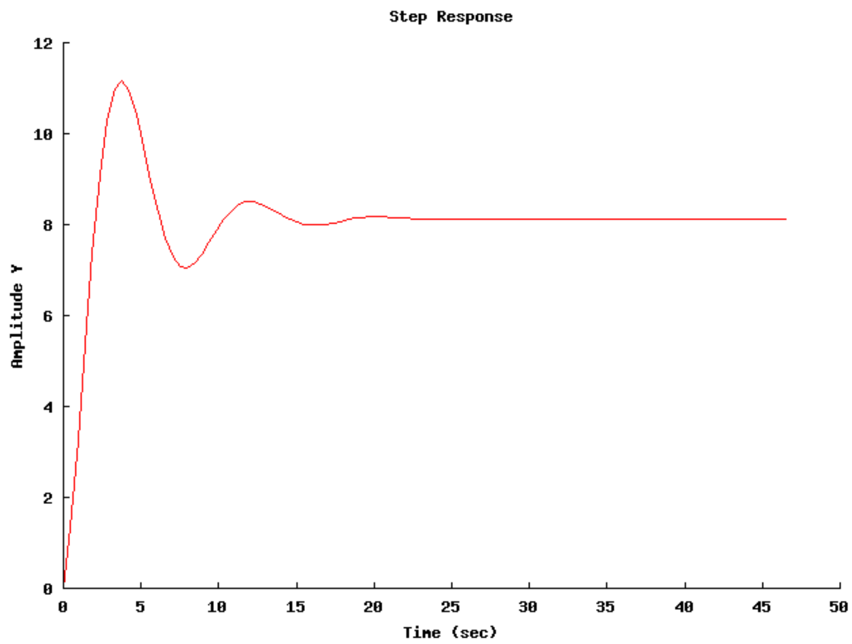
- Time Domain Response of step response
  - Step from $\theta_{desired} = 0$ to $\theta_{desired} = 1.$





P Control

- Time Domain Response:
  - Step from $\theta_{desired} = 0$ to $\theta_{desired} = 8$.
  - Different dynamics in this example… overshoot!



Lab

- Form a group of Three
  - Email instructor one list of names along with a group number
- Start Lab 0
  - Optional for those with MVS and C# experience
- Start Lab 1
  - Code can be downloaded from the internet
- Read Lab 3
  - Can brainstorm in your groups