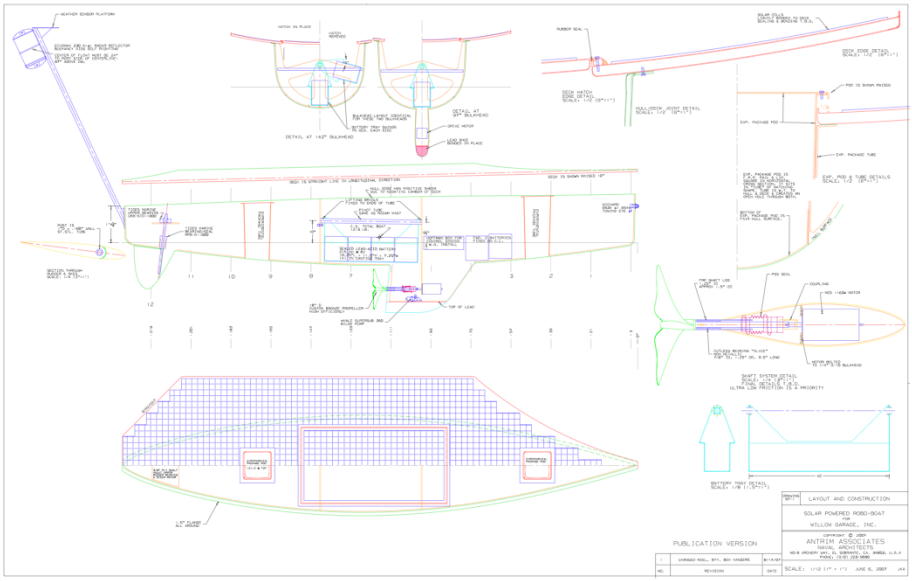
The Overboat



Where did it come from?

Willow Garage

- Menlo Park-based Engineering Research Facility
- Specializes in robotics
- Promotes open-source

Too much work to get it done right and wanted to donate it.

This company is based in Menlo Park. Was actually recently in the news for an autonomous wheeled robot that could navigate their place. They promote open source in all their projects and you can play with what they release.

Why is it here?

- Reprioritization
- Test bed for sensors, software
- And because the Overbot was lonely…

Going to be working with MBARI on interesting software and sensors we can use with it this summer. Possibly working on coordinating missions between the boat and an underwater vehicle.

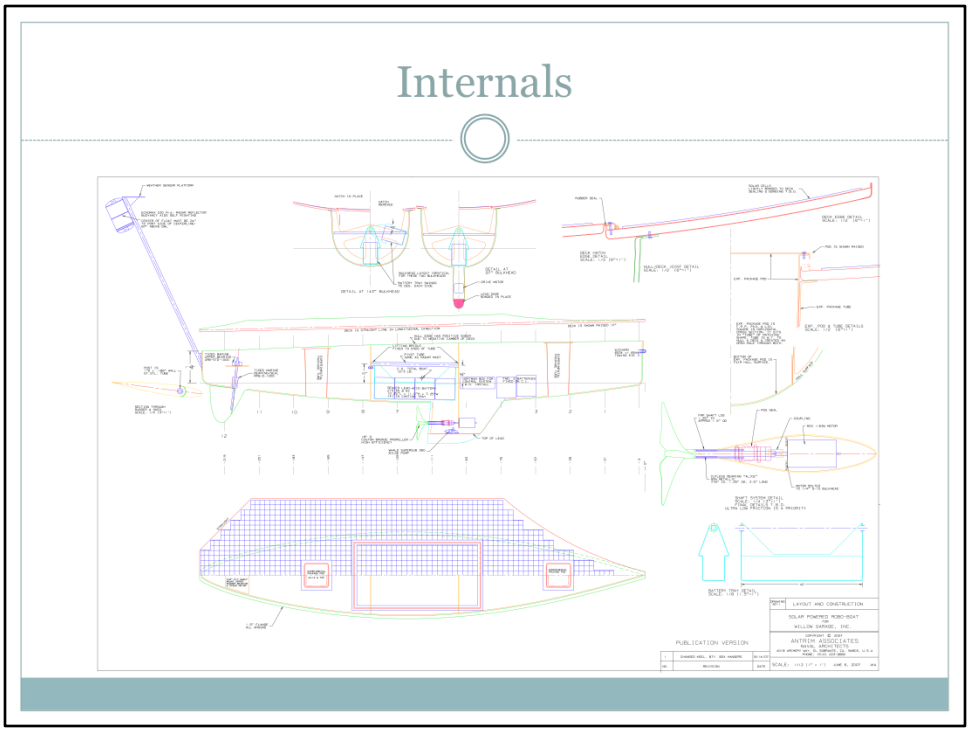
What's it look like?

- 13' surface vehicle
- Bright yellow
- 4 motors
- 6 batteries
- 412 PV cells
- Mobile ballast

Out behind E2, easy to find. In a little sad shape with the PV panels buckling on top. Possibly going to replace them with commercial modules instead of keeping these individual cells.

We have a drive motor, battery tray motor, rudder motor, and a bilge pump to round out the motors.

6 batteries with 4 mobile and 2 stationary.

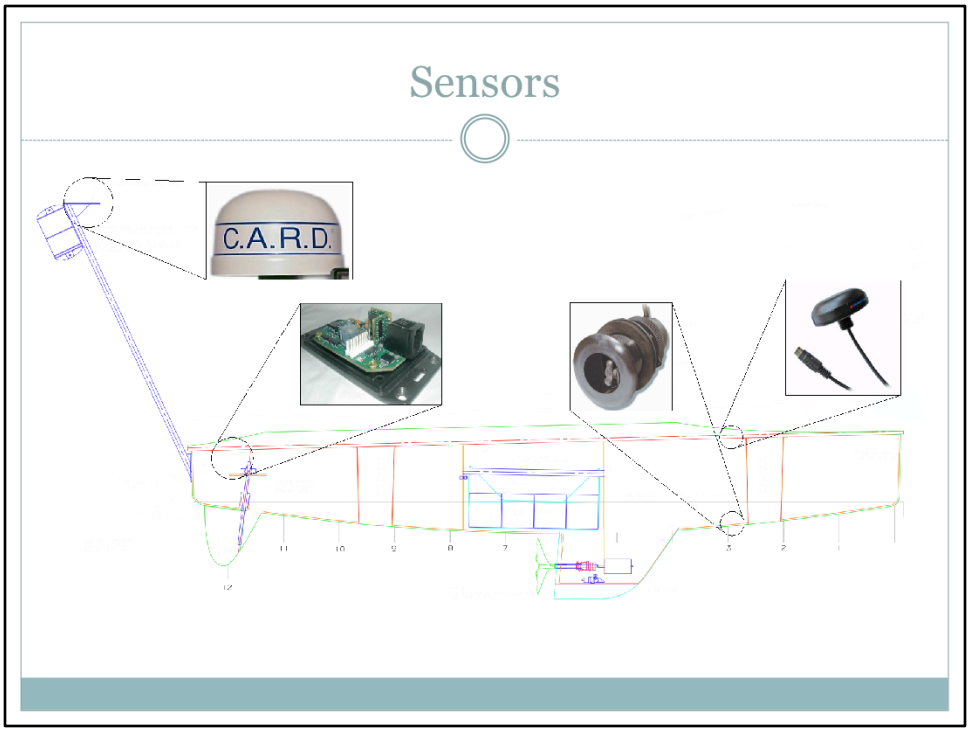


Here's the internal view of the boat.

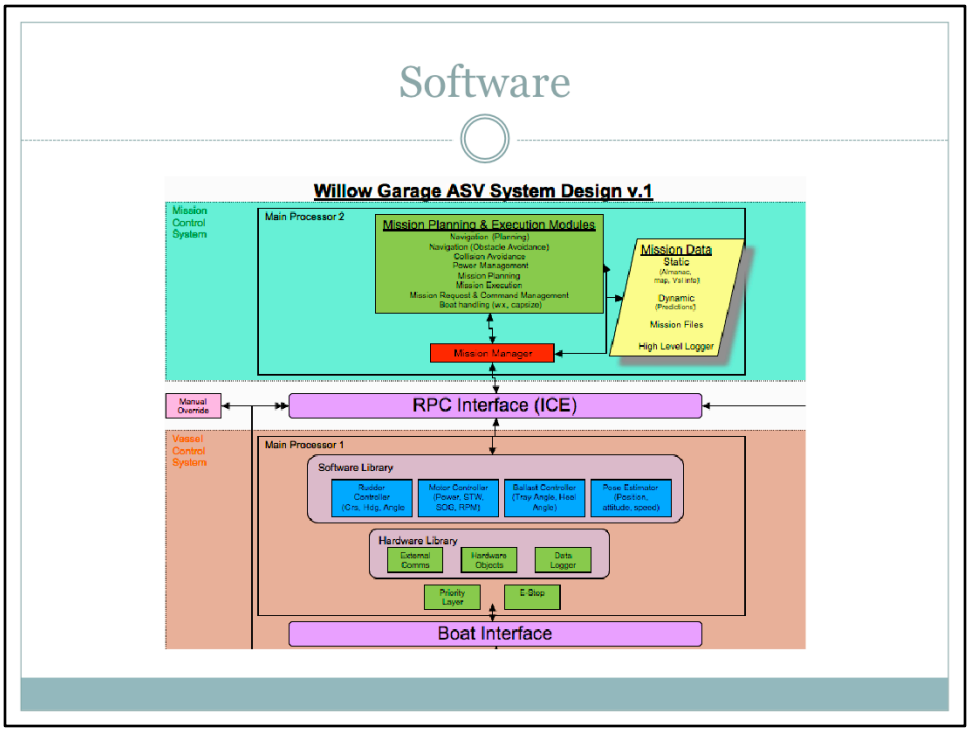
Mast – The mast provides buoyancy to theoretically help right the boat. Also provides a radar target so other boats can see it.

Battery tray – The 4 ballast batteries are on a rotating tray that can tilt the boat. Allows for 60 degree rotation each direction.

Sensor pods – Pods in the bow and stern allow for sensors to be inserted into the boat with access to the water and the air. No connections yet to the main processor so sensors would need to be self-contained, but that's a long-term goal.

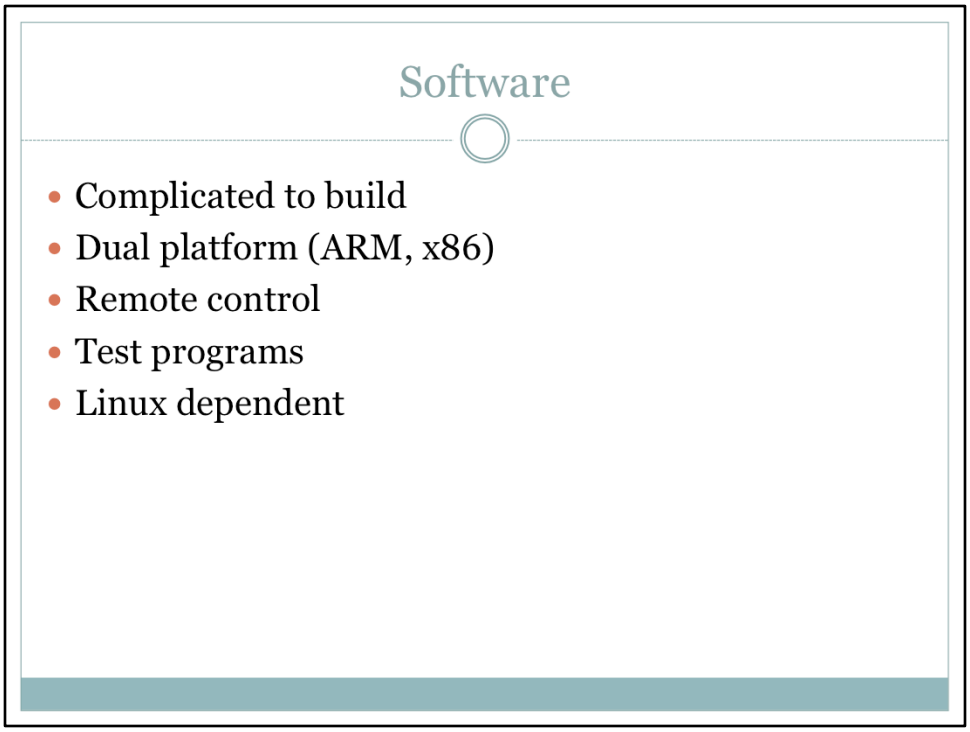


DST800: Water temperature/speed/depth (NMEA2000)
CARD: Passive radar (NMEA2000)
MR-350: GPS (2x) (USB)
Revolution GS: 3-axis magnetometer (NMEA2000)



Create chart of software interaction. More of an appendix to the block diagram of the boat.

Python programs are remote and interact with boat. C++ programs are run directly on the onboard computer.



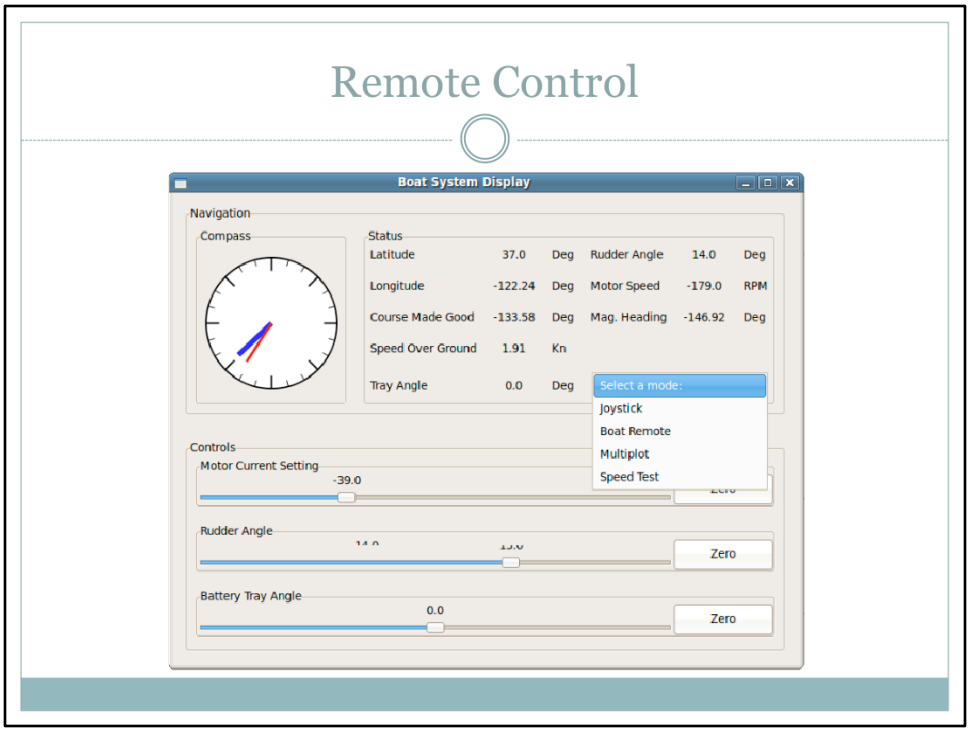
Build system complicated mix of make and jam building for two different architectures. It relies on outdated software packages and doesn't even compile on recent GCC versions.

Software poorly organized and consists mostly of test programs.

Two different programming languages: C++ and Python. C++ for local code and Python for the remote control code.

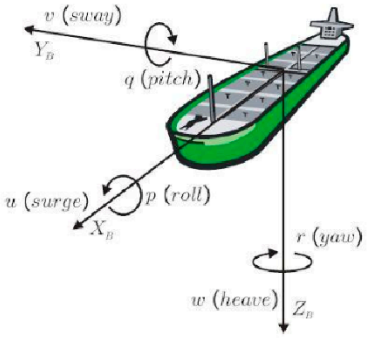
Having looked at most of the software there is some sophisticated waypoint and mission planning on the boat, but no easy way to program those. Looks like an external program was going to be retrofitted for this purpose.

The software also requires a fairly specific configuration of Linux in order to get everything fully running,



This is the boat control program

Terminology



Barauskis and Friis-Hansen. Fast-time Ship Simulator. (2007) pp. 1-8

These are the nautical conventions for all 6 axes.

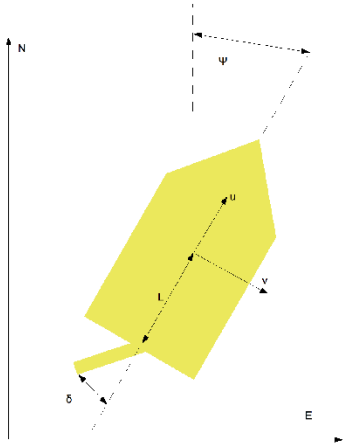
Simulation

- What we're looking for
  - 4 Degrees of freedom (roll, yaw, surge, sway)
  - External forces (current, wind)
  - Power (drain, gain)
  - Global position
  - Real-world response

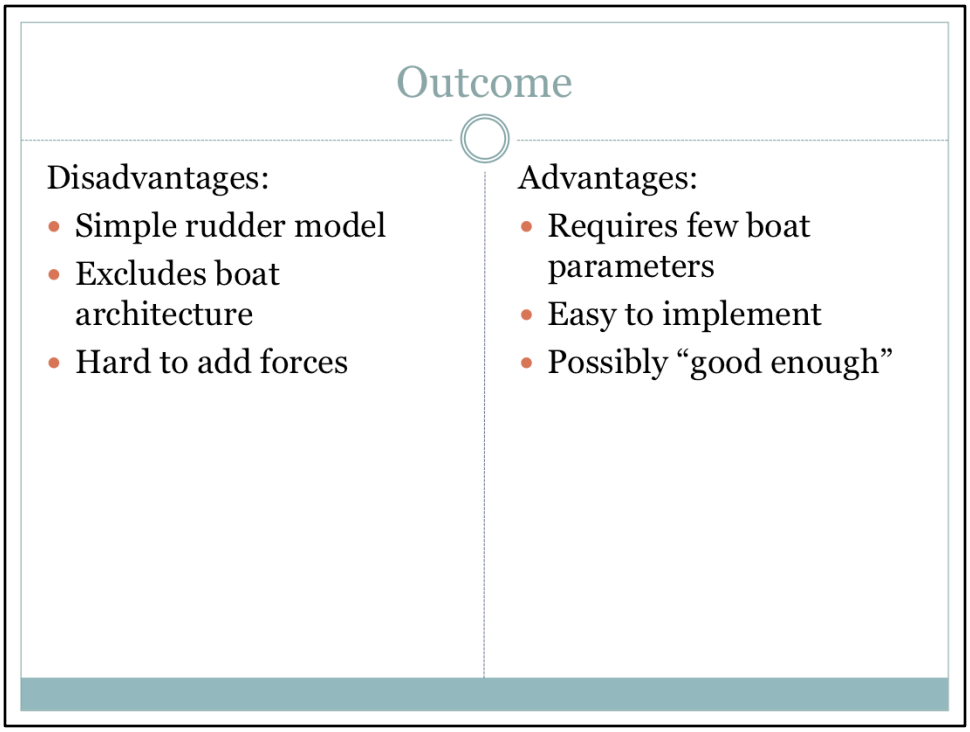
Provided Simulation

- Sun position & power
- Global position
- 4-DOF – inverted bicycle model

$$\dot{n} = u\cos(\Psi) - v\sin(\Psi)$$
$$\dot{e} = u\sin(\Psi) + v\cos(\Psi)$$
$$\dot{\Psi} = \frac{u\tan(\delta)}{L}$$



Phi_dot is simple geometry.



By lacks a proper model I mean there’s no real affect of the rudder as moments or force, it just changes the heading. In reality the rudder flow redirection affect yaw and roll.

Drag is calculated with a simple squared-drag law: drag = C_d*v^2

Autoboat Sim 2.0

- Ties into dsPIC workflow
- Based on 4-DOF model from Fossen 1994
  - Improved rudder model
  - Built around sums of forces and moments
- Requires more boat parameters
  - Need to collect data

Equations of motion

Forces:
- Hydrodynamic (drag, etc.)
- External
- Propulsion
- Control surfaces

$$\begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m & -mz_G & mx_G \\ 0 & -mz_G & I_{xx} & 0 \\ 0 & mx_G & 0 & I_{zz} \end{bmatrix} \cdot \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} X \\ Y \\ K \\ N \end{bmatrix} + \begin{bmatrix} m(vr + x_G r^2 - z_G pr) \\ -mur \\ mz_G ur \\ -mx_G ur \end{bmatrix}$$

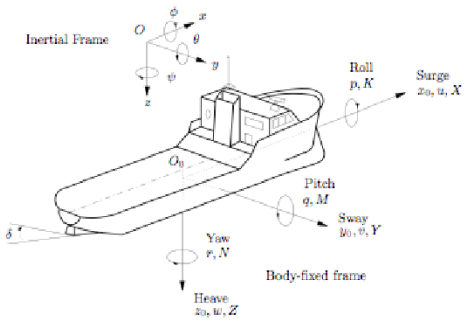
These basic equations of motion take into account the coriolis and centripetal forces. Looks like M_RB*v_dot = tau-C_RB*v. M_RB = matrix mass and inertia due to rigid body dynamics. C_RB*v are from the coriolis and centripetal forces and moments from rigid body dynamics

X_g and Z_g are the coordinates of the center of gravity with respect to the fixed-body coordinate system.

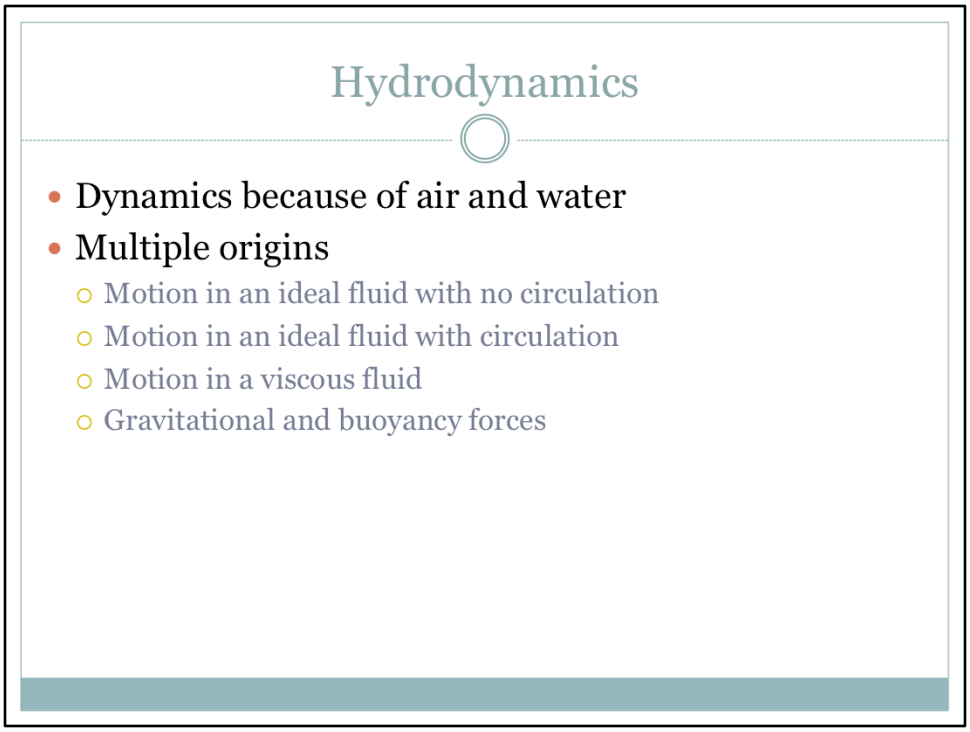
These equations are merely for defining forces in the local frame.

Global Reference Frame

- Equations of motion define dynamics in local inertial frame



Need to convert dynamics into global inertial frame. Easy to do using Euler angles.

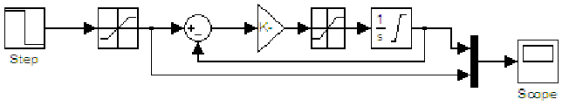


1) Only involves displacement and reveals mass and inertia forces and moments and the Munk moment (moment due to constant translation forcing an increase in the angle of attack, generally a coefficient of "uv")
2) Hull shape is important. Net force acting on it when it moves through with an angle of attack. Lift forces act on the center of pressure and adds to the Munk moment. Terms are uv and ur.
3) Hydrodynamic resistance and is composed of multiple components: frictional resistance (energy lost by moving in a viscous fluid), wave-making resistance (energy lost generating waves), eddy resistance (energy lost in eddies shed from the hull). Terms are abs(u)*u, abs(v)*v, abs(r)*v,abs(v*r, abs(r)*r.
4) Restoring forces and moments dependent on the Euler angles and act on CG and CB. Can be considered equivalent to a mass-damper-spring system.

These hydrodynamic coefficients are on the order of 10^-5, so they shouldn't be too large of a factor in the model, especially with this being a small boat with correspondingly small mass/inertia.

Rudder Lag

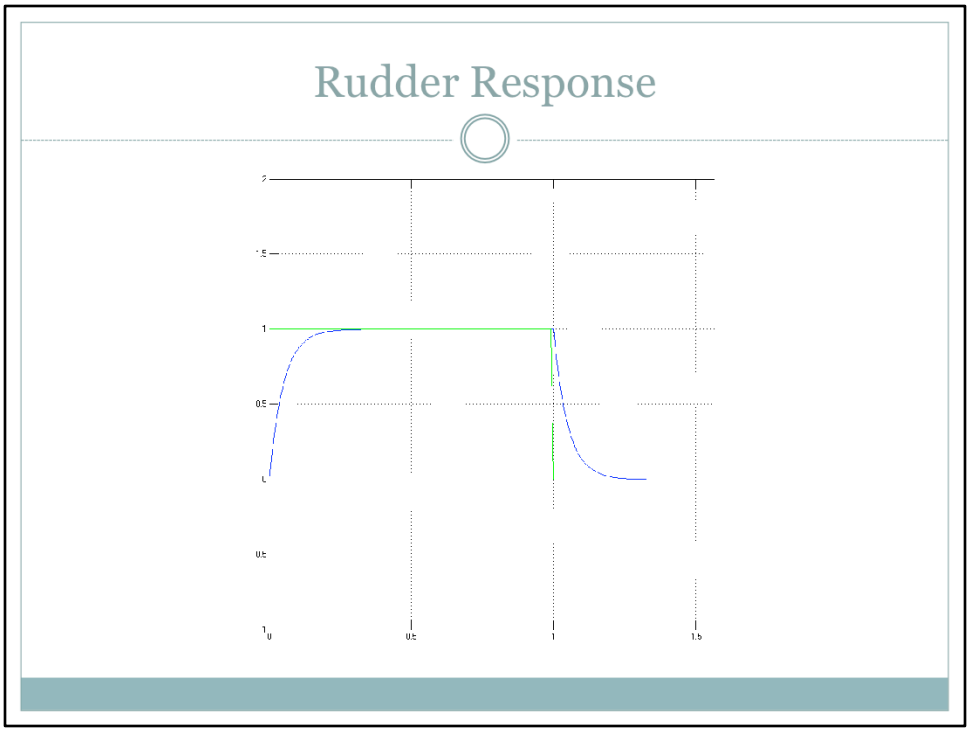
- Rudder angle lags desired rudder position
- Simple 1st-order system



Clamped at input to -1..1, output to -1..1, after gain to -1..1, gain itself is 20.

20 = Tau = rudder_angle_dot_max/proportional_band: rudder_angle_dot_max = 10 <- from original simulation, proportional band = 0.5

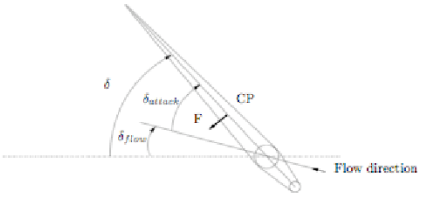
Values were chosen to give an adequate response. Not actually certain if these are realistic or even what proportional band really is.



This is a graph of the rudder response. I mostly made it up to achieve a slight delay. It's not much but it's definitely noticeable in the model and can be tuned easily from gathered data. I mentioned that the time constant is 20.

Rudder Forces

$$F = \begin{cases} \frac{1}{2}\rho C_F A_r V_{av}^2 \sin\left(\frac{\pi}{2}\frac{\delta_{attack}}{\delta_{stall}}\right) \\ \frac{1}{2}\rho C_F A_r V_{av}^2 sign(\delta_{attack}) \end{cases}$$

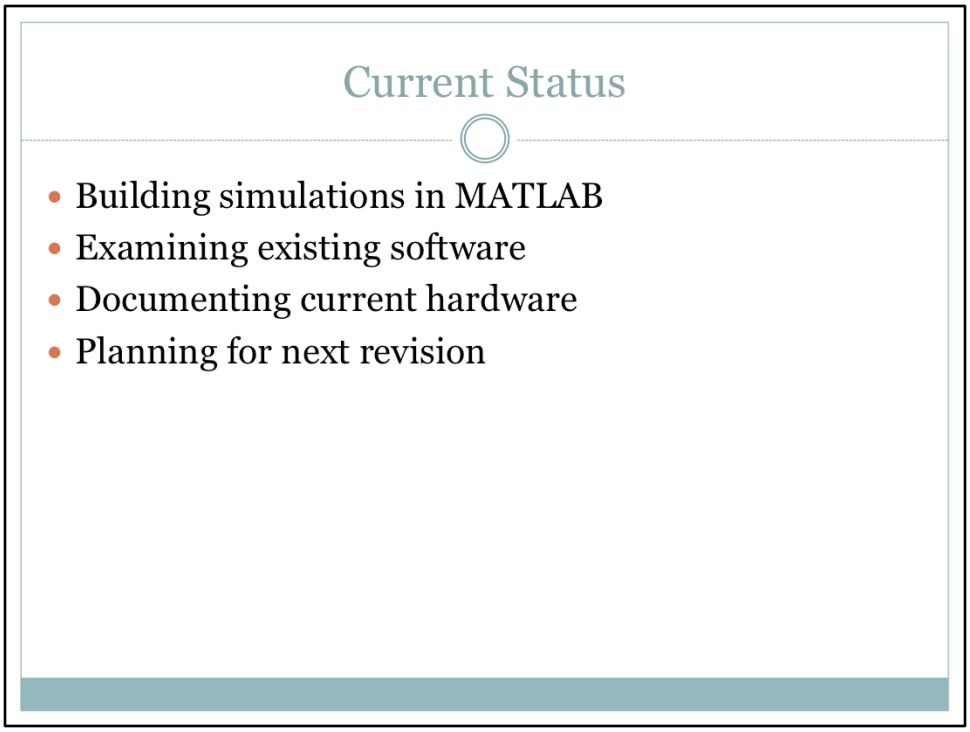


The rudder is treated as a force pushing outward from the center of pressure (CP). Both forces and moments are generated from this model.

P = density of saltwater (rho)
C_F = lift coefficient
A_r = rudder area
V_av = average flow passing the rudder

X_rudder = -Fsin(delta)
Y_rudder = Fcos(delta)
Z_rudder = 0

[K M N] = (CP-CG) x [X Y Z]



I'm currently working on both the old simulation and the next simulation in MATLAB in order to compare them and also have a "safe" more primitive simulation we can use if needed.

The software is pretty complicated, both in the way that it's organized and in the various features in the different programs. I'm continuing to explore the programs and read through the source code. We do have automated documentation, but that's little help when very few comments.

The hardware also takes a decent amount of work to examine. The code, documentation, and even internals all differ from each other and some features aren't implemented or used. (We have 2 extra antennas that aren't connected inside, another cable connected to who knows what. The internals of the boat itself make it difficult to physically follow all the cables as well as look at everything in the boat.

The next revision will be based off the dsPIC. Between the CAN controller and 2 serial ports we should be able to connect all sensors directly to the dsPIC. Depending on processing power, we may need to add a PC to do highlevel control. Wireless communication can be done through simple radios.