

**Virginia Tech**  
Blacksburg, Virginia 24060  
540.231.7385  
E-mail: [scsouth@vt.edu](mailto:scsouth@vt.edu)

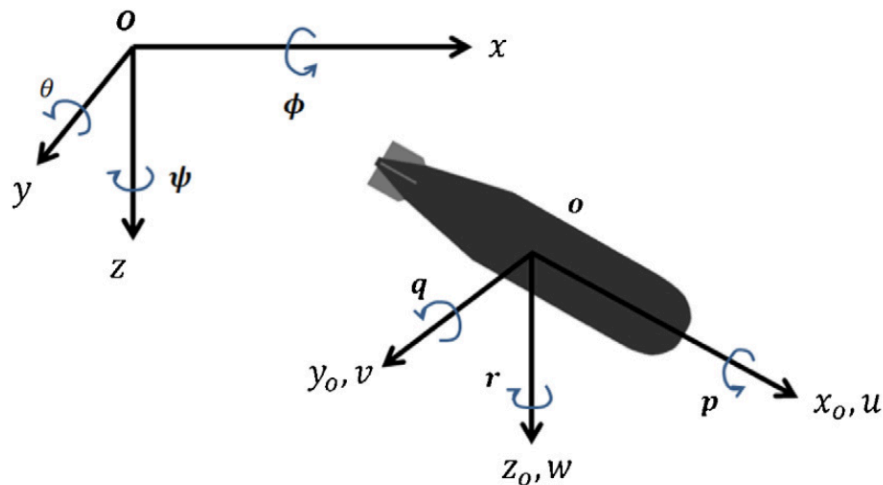
Date: October 31, 2019  
To: ME 5554 / AOE 5754 / ECE 5754 Students  
From: Dr. Steve Southward  
Subject: Phase I Development Program (i.e. your Midterm Project)

## Background

Kraken Robotics, located in Newfoundland Canada, has developed a family of Hovering Autonomous Underwater Vehicles (H-AUV) known as ThunderFish (<https://krakenrobotics.com/products/thunderfish/>). These H-AUV's are equipped with revolutionary Synthetic Aperture Sonar (SAS) systems that enable high-resolution seabed imaging. Your objective for this project is to design a state-feedback control system that can control the H-AUV depth and longitudinal position. To do this, we must first develop LTI equations of motion.

## Equations of Motion

Figure 1 indicates the conventional coordinate system nomenclature defined by: *DTNSRDC Revised Standard Submarine Equations of Motion*, J. Feldman, 1979.



**Figure 1. Standard coordinate system nomenclature for AUV's.**

where:

- $O$  is the origin of the Earth-fixed (absolute) coordinate system
- $\phi$  is the roll angle about the  $x$ -axis (not using for this project)
- $\theta$  is the pitch angle about the  $y$ -axis (rad)
- $\psi$  is the yaw angle about the  $z$ -axis (not using for this project)
- $x$  is the position in the  $x$ -direction (m)
- $y$  is the position in the  $y$ -direction (not using for this project)
- $z$  is the position in the  $z$ -direction (m)

and:

- o is the origin of the AUV body-fixed (local) coordinate system, which is located at the center of mass
- $p$  is the roll angle rate about the  $x_o$ -axis (not using for this project)
- $q$  is the pitch angle rate about the  $y_o$ -axis (rad/s)
- $r$  is the yaw angle rate about the  $z_o$ -axis (not using for this project)
- $x_o$  is called the “Surge” direction (m)
- $y_o$  is called the “Sway” direction (not using for this project)
- $z_o$  is called the “Heave” direction (m)
- $u$  is the velocity in the  $x_o$ -direction (m/s)
- $v$  is the velocity in the  $y_o$ -direction (not using for this project)
- $w$  is the velocity in the  $z_o$ -direction (m/s)

For this project, we are only interested in pitch ( $\theta$ ), heave ( $z$ ), and surge ( $x$ ) motions. This model is commonly referred to as a “pitch-axis” model. The linearized mapping between the AUV-fixed coordinates and the world coordinates are given by:

$$\begin{aligned}\dot{x}(t) &= u(t) \\ \dot{z}(t) &= w(t) \\ \dot{\theta}(t) &= q(t)\end{aligned}$$

The following assumptions have been used to simplify the model:

- The AUV is neutrally buoyant, i.e. its weight is equal to its buoyancy
- The AUV-fixed coordinate system is located at the AUV center of mass
- The center of mass is at the same point as the center of buoyancy
- The pitch angle is relatively small

The linearized dynamic equations of motion are obtained from Newton’s second law:

$$\begin{aligned}\sum (\text{Forces in the } x_o \text{ direction}) &= m\dot{u}(t) = X_{ext}(t) \\ \sum (\text{Forces in the } z_o \text{ direction}) &= m\dot{w}(t) = Z_{ext}(t) \\ \sum (\text{Moments around the } y_o \text{ axis}) &= I_y\dot{q}(t) = M_{ext}(t)\end{aligned}$$

where:  $m$  is the mass of the AUV (kg)  
 $I_y$  is the mass moment of inertia about the  $y_o$  axis ( $\text{kg}\cdot\text{m}^2$ )

The linearized external forces and moments are given by:

$$\begin{aligned}X_{ext}(t) &= -X_{\dot{u}}\dot{u}(t) - X_u u(t) + X_T(t) \\ Z_{ext}(t) &= -Z_{\dot{w}}\dot{w}(t) - Z_w w(t) - Z_{\dot{q}}\dot{q}(t) - Z_q q(t) + Z_s(t) + Z_b(t) \\ M_{ext}(t) &= -M_{\dot{q}}\dot{q}(t) - M_q q(t) - M_{\dot{w}}\dot{w}(t) - M_w w(t) - x_s Z_s(t) - x_b Z_b(t)\end{aligned}$$

where:

- $X_{\dot{u}}$  is the  $x_o$ -direction “added mass” from accelerating water (kg)
- $X_u$  is the  $x_o$ -direction hydrodynamic “drag” coefficient (N•s/m)
- $Z_{\dot{w}}$  is the  $z_o$ -direction “added mass” from accelerating water (kg)
- $Z_w$  is the  $z_o$ -direction hydrodynamic “drag” coefficient (N•s/m)
- $Z_{\dot{q}}$  is the  $z_o$ -direction “added mass” caused by rotation (kg•m)
- $Z_q$  is the  $z_o$ -direction hydrodynamic drag caused by rotation (kg•m/s)
- $M_{\dot{q}}$  is the “added rotational inertia” about the  $y_o$ -axis (kg•m<sup>2</sup>)
- $M_q$  is the moment “drag” coefficient about the  $y_o$ -axis (N•m•s)
- $M_{\dot{w}}$  is the “added rotational inertia” about the  $y_o$ -axis (kg•m)
- $M_w$  is the moment “drag” coefficient about the  $y_o$ -axis (N•s)
- $x_s$  is the position of the stern (rear) control surface in the  $x_o$ -direction (m)
- $x_b$  is the position of the bow (front) control surface in the  $x_o$ -direction (m)

and the **external control inputs** are:

- $X_T$  is the thrust in the  $x_o$ -direction (N)
- $Z_s$  is the stern thrust in the  $z_o$ -direction (N)
- $Z_b$  is the bow thrust in the  $z_o$ -direction (N)

The **pitch angle  $\theta$** , the **depth  $z$** , and the **position  $x$**  are directly measured with sensors to represent the outputs of the model.

**Problem 1. [10 pts.]** Derive a complete state-space model for the AUV, including a state equation and an output equation. Hint: The MathWorks LiveScript feature has a built-in equation-editor which is very easy to use (simpler than LaTeX), and allows you to embed formatted equations directly in your Matlab script.

**Problem 2. [2 pts.]** Using the numerical values and formulas listed below, compute the following parameters at the top of your Matlab script.

$$\begin{aligned}
 m &= 500 \text{ (kg)} & Z_q &= 9.5e2 \text{ (kg}\cdot\text{m/s)} \\
 L &= 25 \text{ (m)} & M_{\dot{q}} &= \left(\frac{1}{20}\right)I_y \text{ (kg}\cdot\text{m}^2) \\
 I_y &= \left(\frac{1}{300}\right)mL^2 \text{ (kg}\cdot\text{m}^2) & M_q &= 1.1e3 \text{ (N}\cdot\text{m}\cdot\text{s)} \\
 X_{\dot{u}} &= \left(\frac{1}{30}\right)m \text{ (kg)} & M_{\dot{w}} &= \left(\frac{1}{40}\right)I_y \text{ (kg}\cdot\text{m)} \\
 X_u &= 94 \text{ (N}\cdot\text{s/m)} & M_w &= 320 \text{ (N}\cdot\text{s)} \\
 Z_{\dot{w}} &= \left(\frac{1}{10}\right)m \text{ (kg)} & x_s &= -\left(\frac{1}{3}\right)L \text{ (m)} \\
 Z_w &= 4.7e2 \text{ (N}\cdot\text{s/m)} & x_b &= \left(\frac{1}{3}\right)L \text{ (m)} \\
 Z_{\dot{q}} &= \left(\frac{1}{20}\right)m \text{ (kg}\cdot\text{m)} & U_{max} &= 3e6 \text{ (N)}
 \end{aligned}$$

$$\text{where: } |X_T| \leq U_{max} \quad |Z_s| \leq U_{max} \quad |Z_b| \leq U_{max}$$

**Problem 3. [10 pts.]** Use the variables defined in Problem 2 to construct the **A**, **B**, **C**, and **D** matrices for your linear state-space model, then use these matrices to generate an LTI object called `ol_sys`. You are NOT allowed to directly substitute numerical values for the parameters above into your matrices!

You must also define short descriptive names for each of the states, inputs, and outputs and embed these names in the LTI state-space model object using the **SET command**.

**Problem 4. [2 pts.]** Apply the MINREAL function in Matlab to your LTI object from Problem 3 to determine whether your state-space model is a minimum realization. Explain your results.

**Problem 5. [10 pts.]** Show that the open-loop system is Completely Controllable using all of the control inputs. Determine (and clearly document) whether any subsets of control inputs results in a Completely Controllable system. Justify your results.

**Problem 6. [6 pts.]** Compute the open loop poles, the natural frequencies with units of Hz, and the damping ratios for each eigenvalue. Tabulate and display your results using the **Matlab TABLE function**.

**Problem 7. [10 pts.]** Define an initial condition vector using the following initial values:

$$\begin{array}{ll} x(0) = 50 & u(0) = 100 \\ z(0) = 350 & w(0) = -300 \\ \theta(0) = 45^\circ & q(0) = 0 \end{array}$$

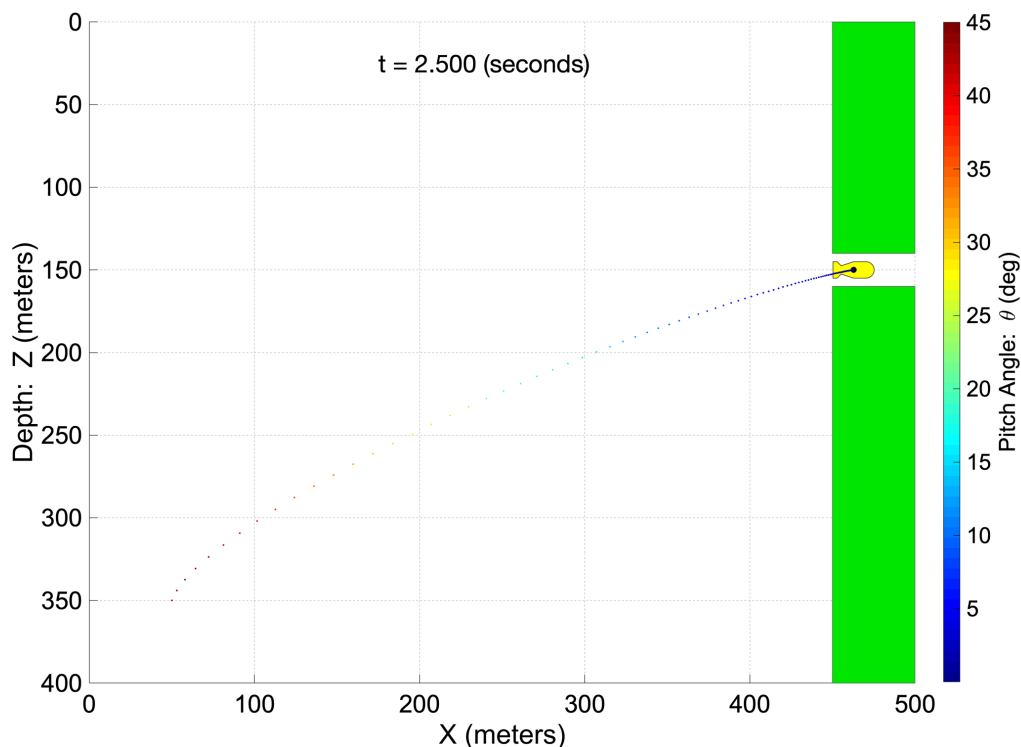
Use the **INITIAL** function in Matlab to simulate the initial value response of the open-loop system. Using the output data from this simulation with the **LSIMINFO** function, estimate and display the 5% settling times for each of the open-loop response outputs.

Plot the open-loop response **outputs** on a single figure. You may combine signals with the same units on the same axes, but if signals have different units, they must be in a separate axes. Properly annotate your figure with axis labels, grid lines, and a legend.

Place a marker on each output curve at the time corresponding to the 5% settling time.

**Problem 8. [15 pts.]** Design a full-state feedback controller that meets the following performance requirements when the closed-loop system is simulated using the same initial condition vector that was used in Problem 7:

- Each control signal must be bounded by the limit specified in Problem 2
- The AUV must completely enter the tunnel without contacting the barrier
- Minimize the time it takes to enter the tunnel



The Matlab function **animate\_auv.p** is provided for you to generate the plot above using output data from your simulation. To use this function, you must provide the  $x(t)$ ,  $z(t)$ , and  $\theta(t)$  data, where  $x$  and  $z$  are in meters, and  $\theta$  is in radians. You can also input an optional Boolean value (savemovie) to save the animation to a .mpg file.

**animate\_auv(t,x,z,theta,savemovie)**

Document a minimum of five design iterations that demonstrate that you attempted to achieve the requirements. Your documented design iterations must show sufficient improvement and must clearly indicate that you made an earnest attempt to meet the requirements.

Your documentation for each iteration should at least include:

- The desired closed-loop poles you chose
- An indication of whether you meet the requirement for entering the tunnel without contact
- An indication of whether you meet the requirement for control signal saturation
- 5% settling times for each of the outputs

**Final Design Plot. [5 pts.]** For your FINAL closed-loop design ONLY, plot a single figure window with three axes (3 rows and 1 column). The topmost axes should include the two position outputs, the middle axes should be the pitch angle, and the bottom axes should include both control signals. Properly annotate each axes with clearly defined axis labels, grid lines, and legends.

Place a marker on each output curve at the time corresponding to the 5% settling time.

#### **Submission Rules:**

- You MUST use the Matlab LiveScript feature to generate your Midterm Project results. This will serve as a formatted report that includes your Matlab code, any outputs to the command window, and any plots you generate.
- You MUST submit both the .mlx (LiveScript) file as well as a PDF version of your LiveScript output.
- You MUST use the “Output Inline” option in LiveScript. You CANNOT use the “Output on Right” option which places plots on the right-hand side of the page and the code on the left-hand side.
- You MUST insure that the plots you generate are sufficiently large that the PDF displays them at a publication-quality size (i.e. preferably with a width that is close to the margin-to-margin edges).
- You MUST NOT display long columns of output data in your LiveScript!!