Multi-Degree of Freedom Boat Model

Authors
Jared Jacobowitz and Yair (Andrew) Gross

14 May 2021
ME301 - Spring 2021

Contents

List of Figures

Introduction

The motivation for this project originated from the desire to model a multi-degree of freedom system that allowed for breaking up the degrees of freedom for project work allocation within the group. A boat model became the obvious choice due to the six degrees of freedom, specifically three translational degrees, referred to as heave, sway, and surge, and three rotational degrees, referred to as roll, pitch, and yaw (see Figure 1a). To reduce the project within the scope and time frame of this course, the boat model was simplified to three degrees, specifically, heave, roll, and pitch. To aid in the vibrational analysis of the system, the project was modeled and analyzed using transfer functions (i.e. in the Laplace/frequency domain). Despite the reduction of the degrees of freedom, the choice of the degrees that were kept still allowed for realistic boat motion simulation.

Assumptions

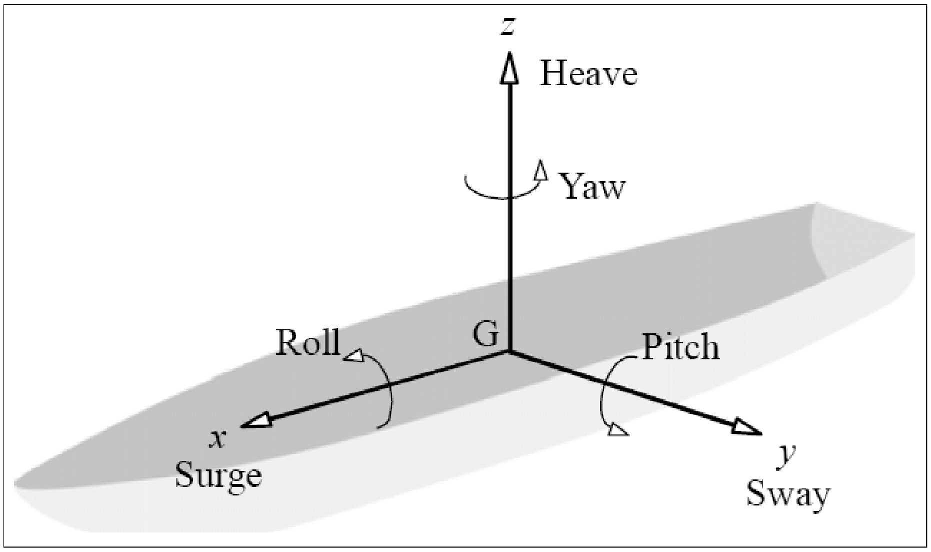
The following assumptions were made:

1. The waves are small in amplitude and low in frequency. This enabled the follow "sub-assumptions":
   - The vertical motion of the waves was small enough that the boat was always in contact with the water.
   - The angles of the waves were small enough that small angle approximations could be used.
   - The wavelength was long enough that the boat would perceive the waves as linear/planar.
2. The boat is a solid rectangular block of uniform density.
3. The motion in the water involved minor viscous damping.

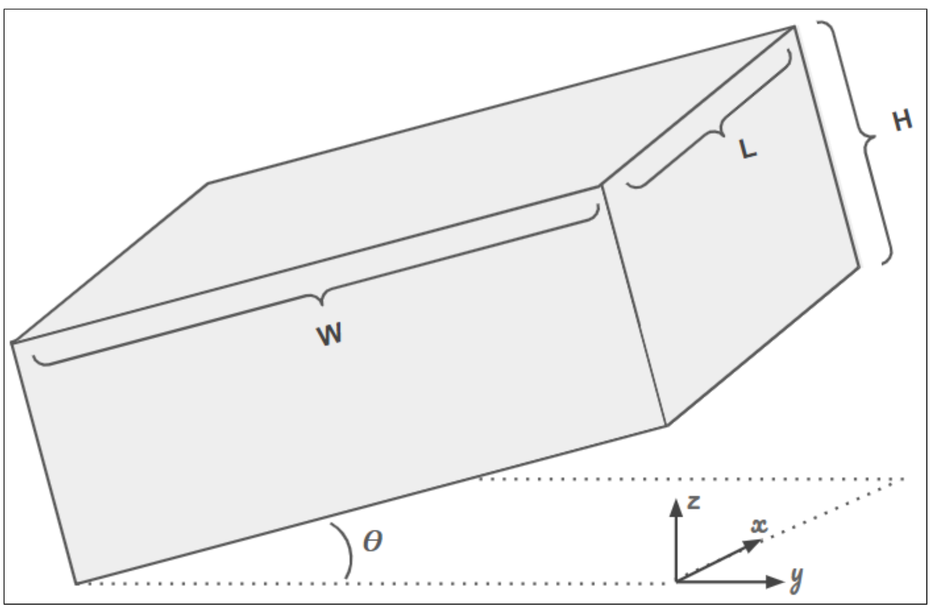
Modeling

The meanings of roll, pitch, heave, and other degrees of freedom of a boat are shown in Figure 1a below. This diagram is accompanied by a depiction of the simplified construction in which the boat is considered to be a uniform rectangular block with dimensions W, L, and H (Figure 1b). As shown, the heave of a boat is its up-and-down motion, or its translation along the z axis. Roll represents the rotation of the boat around the x axis, with the roll angle indicated by θ. Pitch represents the boat's rotation around the y axis, with pitch angle indicated by ϕ (not shown in Figure 1b).

Dimensions and Coordinate System



(a) Degrees of Freedom



(b) Boat Dimensions

Figure 1: (a) Illustration of the degrees of freedom of a boat (heave, surge, sway, roll, pitch, and yaw) relative to the boat position and to a fixed coordinate system. (b) Depiction of the boat dimensions and positional parameters relative to the axis directions.

The simulated system was constructed by first individually modelling the boat motion in each individual degree of freedom being considered. Treating the boat as a rectangular block of uniform density allowed for pitch and roll to be treated almost identically. Therefore, the degrees of freedom could be conveniently divided into two categories, translational (heave) and rotational (pitch and roll). These were analyzed, modelled, and simulated independently before being combined into a unified, 3-DOF system.

Heave

Model Development

Vibration is caused by inertia and a restoring force (or multiple restoring forces), where a displaced object is pushed towards equilibrium by the restoring force(s), overshoots due to inertia extending the motion, and pulled back by the restoring force(s), overshooting again and restarting the cycle. For the heave motion of the boat, the inertia comes from the boat possessing mass, and the restoring forces are gravity, F_g, pointing down and the buoyancy force, F_b, pushing up (Figure 2). Buoyancy force is an upward force from a fluid to counteract the fluid displaced by an object, and can be found from the following equation:

$$F_b = \rho g \cancel{V}_{sub} = \rho g W L x \tag{1}$$

where ρ is the fluid density (in this case: ocean water), $\cancel{V}_{sub}$ is the submerged volume of the boat, and x is the distance of the COM from the water, where negative means the center of mass (COM) is below the water level and positive means above. Comparing the form of this force to the spring force of the form $F = kx$, the rho, g, W, and L terms can be combined into a single heave spring constant:

$$k_h = \rho g W L \tag{2}$$

giving

$$F_b = k_h x \tag{3}$$

Heave FBD

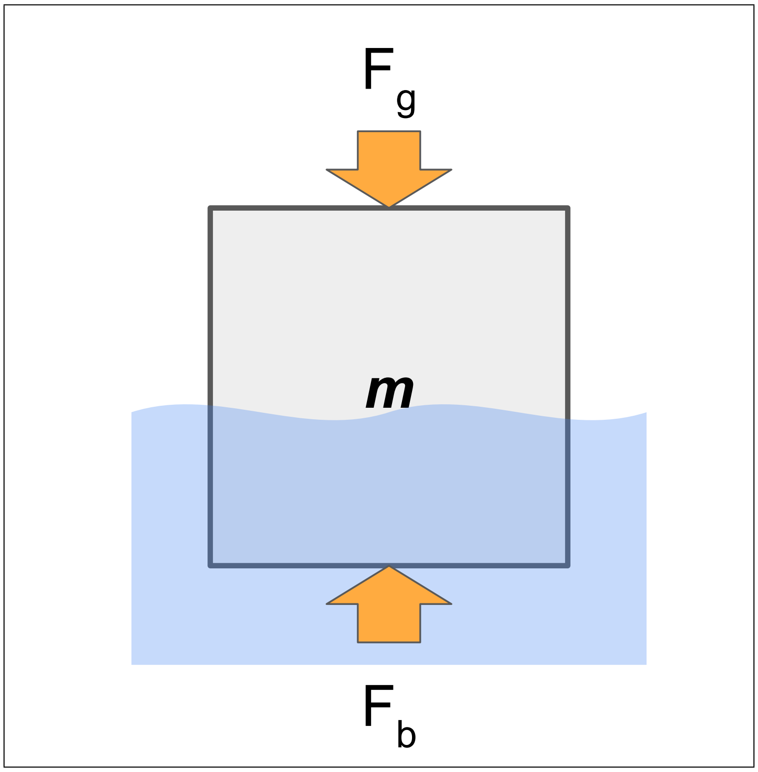


Figure 2: Gravitational and buoyancy forces acting on the boat in heave.

Because boats also experience roll and pitch, the angle of the boat in the water needed to be considered, as the submerged volume would change with the angle of the boat. It will be shown that the small angle approximation allows for the effect of the roll and pitch angle to be neglected. *Note*: because the roll and pitch effects on the boat's submerged volume are equivalent, only roll is considered for the below derivation.

Considering a boat at roll angle θ with the COM a distance x submerged below the water level (Figure 3), the submerged area is the trapezoid area with base lengths a and $a+b$ and height W. From similar triangles, it can be seen that $b = 2(c+d)$ where

$$c = \frac{x}{\cos(\theta)} \tag{4}$$

$$c + d = \frac{W}{2}\tan(\theta) \tag{5}$$

The a side length is half the boat height, H, minus length d, i.e.:

$$a = \frac{H}{2} - d \tag{6}$$

Combining all the results into the trapezoid area equation, the following is found:

$$A = \frac{a + (a+b)}{2} W = \left(a + \frac{b}{2}\right) W = (a + c + d)W = \left(\frac{H}{2} - d + \frac{x}{\cos(\theta)} + d\right) W \quad (7)$$

$$= \left(\frac{H}{2} - \frac{x}{\cos(\theta)}\right) W \quad (8)$$

Because the COM is below the water line in this situation, the x value is actually negative, so the sign for A is flipped in the last line to account for that.

Boat in Roll with the COM Below the Water

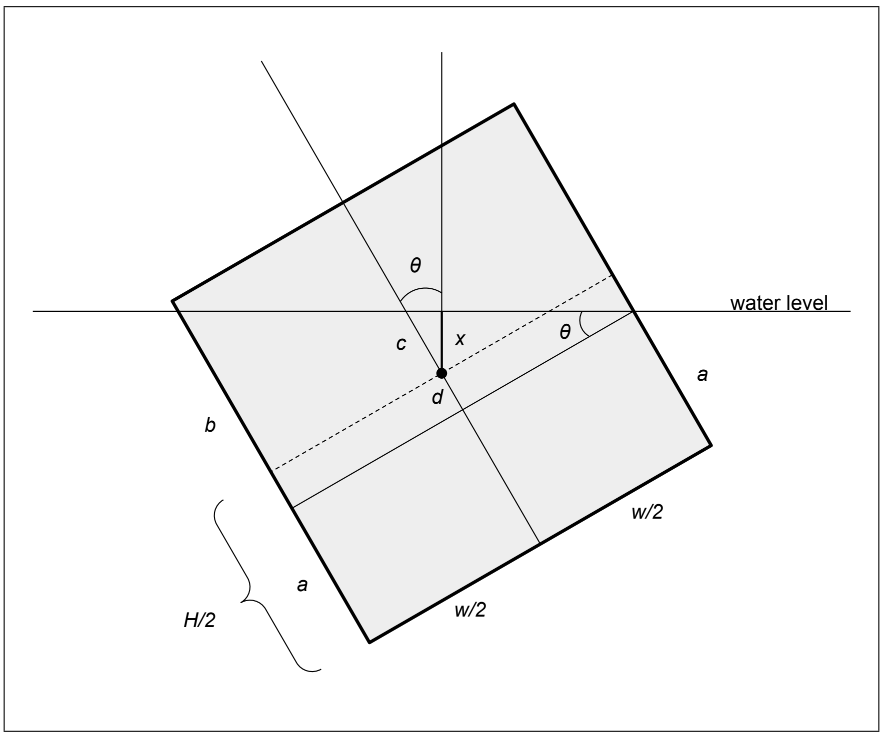


Figure 3: Labeled boat experiencing roll where the COM is below the water level. This figure is used for determining the submerged area in terms of roll.

The same analysis can be performed for the situation where the center of mass is above the water (Figure 4). As before, similar triangles gives $b = 2(c + d)$ where

$$c = \frac{x}{\cos(\theta)} \quad (9)$$

$$d = \frac{W}{2} \tan(\theta) \quad (10)$$

$$a = \frac{H}{2} - (c + d) \quad (11)$$

Combining all the results into the trapezoid area eqution, the following is found:

$$A = \frac{a+(a+b)}{2}W = \left(a + \frac{b}{2}\right)W = (a+c+d)W = \left(\frac{H}{2} - d - \frac{x}{\cos(\theta)} + d\right)W \quad (12)$$

$$= \left(\frac{H}{2} - \frac{x}{\cos(\theta)}\right)W \quad (13)$$

The same equation for the submerged area was found as the case where the COM is below the water, both of which include a $1/\cos(\theta)$ term.

Boat in Roll with the COM Above the Water

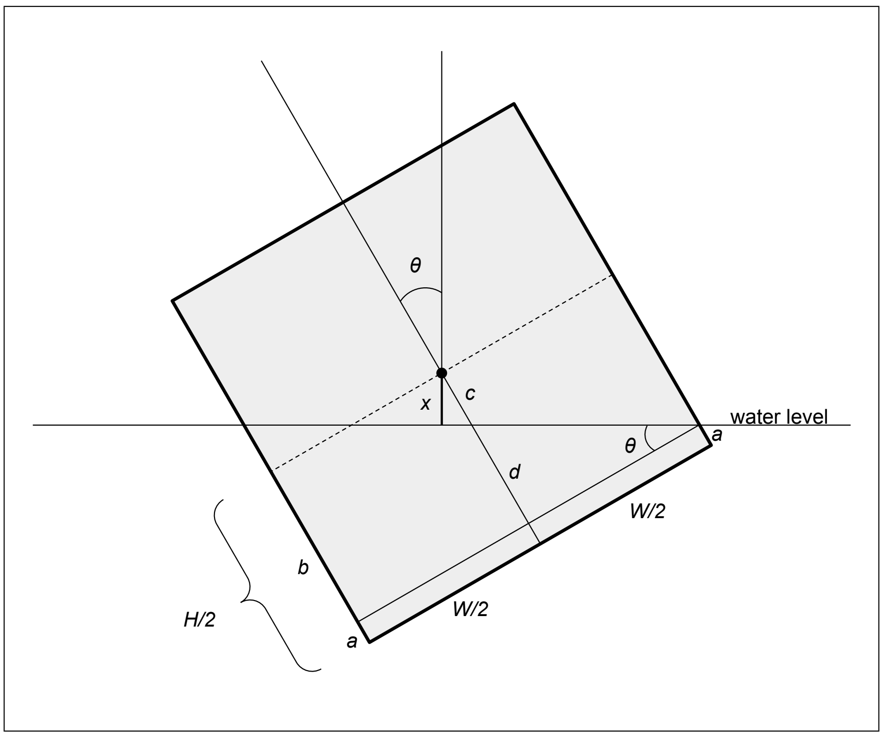


Figure 4: Labeled boat experiencing roll where the COM is above the water level. This figure is used for determining the submerged area in terms of roll.

Taking the small angle approximation of $\cos(\theta) \approx 1$, one can see that the submerged area is not effected by the angle, showing that that this approximation of heave is independent of roll and pitch.

When static, a boat will not rest with the COM at the water line, instead resting at an equilibrium height that can be determined from the shape and mass of the boat. To determine where the COM will be, Newton's second law with zero acceleration is used to find depth of the COM below the waterline when in equilibrium.

$$\sum F = \cancelto{0}{ma} = F_b - F_g \rightarrow k_h x_{ss} = mg \rightarrow x_{ss} = \frac{mg}{k_h} \quad (14)$$

For the dynamic case, Newton's second law can again be used. The variables are written in terms of velocity, v, where possible.

$$\sum F = m\dot{v} = F_b - F_g = k_h\left(x_{ss} - \int v\,\mathrm{d}t\right) - mg \tag{15}$$

$$\rightarrow m\dot{v} + k_h \int v\,\mathrm{d}t = k_h x_{ss} - mg \tag{16}$$

Converting the above differential equation into the Laplace domain enables easier variable isolation.

$$\mathcal{L}\{\} \rightarrow V(s)\left(ms + \frac{k_h}{s}\right) + \underbrace{\frac{k_h}{s}x_0 - mv_0}_{\text{initial conditions}} = k_h x_{ss} - mg \tag{17}$$

$$V(s) = \frac{k_h\left(x_{ss} - \frac{x_0}{s}\right) - mg + mv_0}{ms + \frac{k_h}{s}} \tag{18}$$

The velocity equation can be written as $V = F/Z$, showing the force and impedance values, where

$$Z_h = ms + \frac{k_h}{s} \tag{19}$$

This term will be useful for later combining the heave motion with the roll and pitch motions in MATLAB.

The velocity can then be integrated to get position, being sure to account for the initial conditions.

$$X(s) = \frac{V(s)}{s} + \frac{x_0}{s} \tag{20}$$

Initial Displacement Response

Using MATLAB, the system can be converted back into the time domain using the `impulse` function. After providing the system with boat dimensions and properties, an initial displacement response can be observed, with the system oscillating sinusoidally (Figure 5) as expected in this simple vibration response. The system was provided an initial displacement of $0.1m$, though the displacement plot shows the COM started at a smaller value. This is because the displacement was provided to the mass already at rest, and the mass's steady-state position was not with the COM at the water level but about $0.3m$ below it.

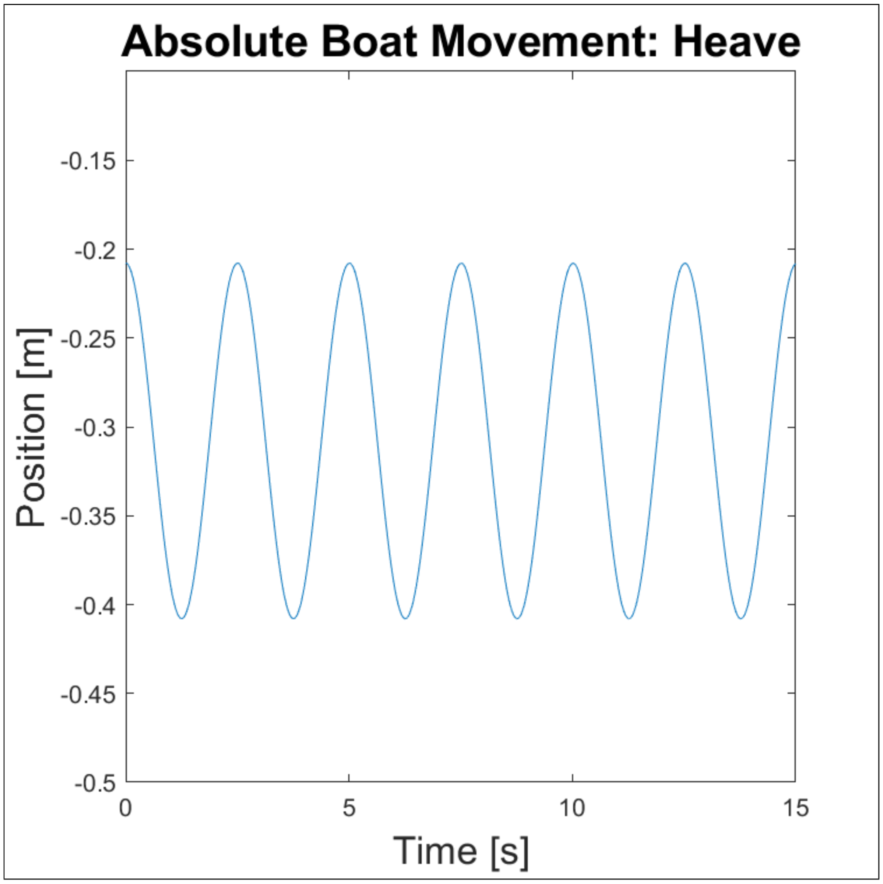


Figure 5: Heave motion of the boat after an initial displacement of $0.1m$ from equilibrium.

Forced Response

Instead of simply experiencing an initial displacement, a boat will also experience the effects of the waves. This can be modeled as a forcing on the system, where, for heave, the wave moves up and down at some frequency and amplitude. Simulating this in MATLAB, the system is seen to have two vibrations frequencies superimposed (Figure 6). The faster frequency is that of the mass's natural frequency of bobbing in the water, and the slower frequency is the forcing of the wave, in this case modeled as a sine wave.

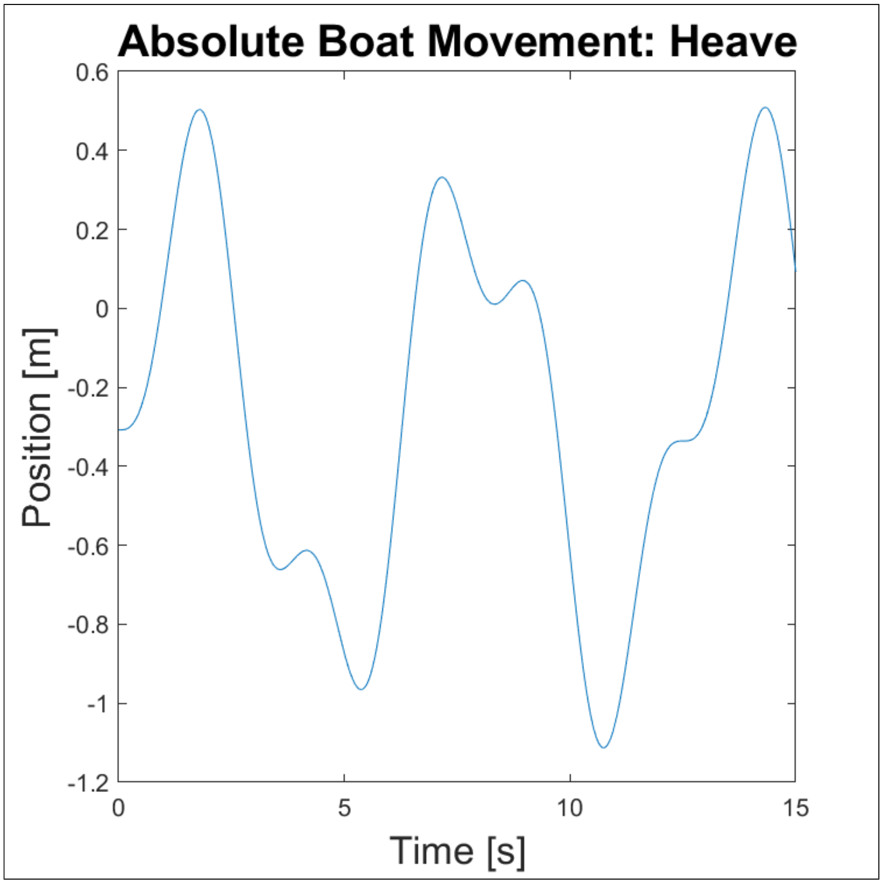


Figure 6: Heave motion of the boat when both initially displaced and forced by the wave. The response is a superposition of the two displacement plots (that of the boat's displacement response seen in Figure 5 and that of the wave).

Roll/Pitch

Model Development

As discussed above, the reduction of the modelled boat to a uniform-density rectangular block allows pitch and roll to be treated identically, requiring only a change in dimensions, swapping the length, L, and width, W, of the boat. The following derivation will exclusively discuss roll, and will conclude by adapting the resulting equations of motion to describe pitch as well.

The underlying cause of roll, or the rotational oscillation of the boat around its x-axis, is a restoring torque on the boat caused by the buoyancy force, F_b. The force of gravity always acts through the boat's COM, and therefore never exerts a torque on the boat. The buoyancy force, however, acts through a point known as the **center of buoyancy**, indicated below as point B, which is defined to be the centroid of the displaced volume of water. The direction in which F_b acts is always straight up, parallel to the z axis. As can be seen in the following diagram, the center of buoyancy shifts away from the boat's centerline as the boat tilts away from equilibrium, resulting in a restoring torque.

Restoring Torque Resulting from Buoyancy Force

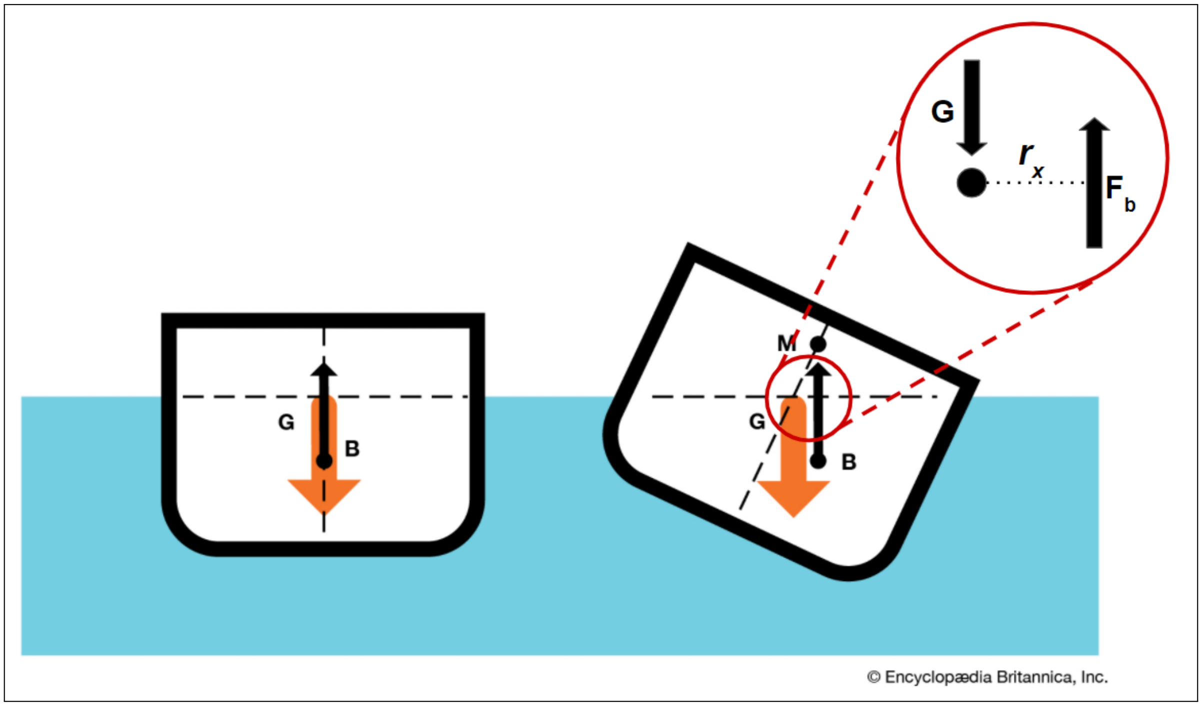


Figure 7: Side-by-side illustrations of a boat in equilibrium (left) and rotated out of equilibrium (right) with a zoomed-in diagram of the buoyancy force F_b, center of mass, and moment arm r_x.

Since the buoyancy force has been established as the only source of torque acting on the system, the application of Newton's second law for rotational motion yields:

$$I\ddot{\theta} = \sum \tau = F_b r_x \tag{21}$$

where r_x is the moment arm of the buoyancy force to the COM.

To find the net torque acting on the boat, the submerged volume was divided into two sections, considered separately. The divider is the horizontal line passing through the elevated submerged corner of the boat as shown in Figure 8.

Restoring Torque Resulting from Buoyancy Force

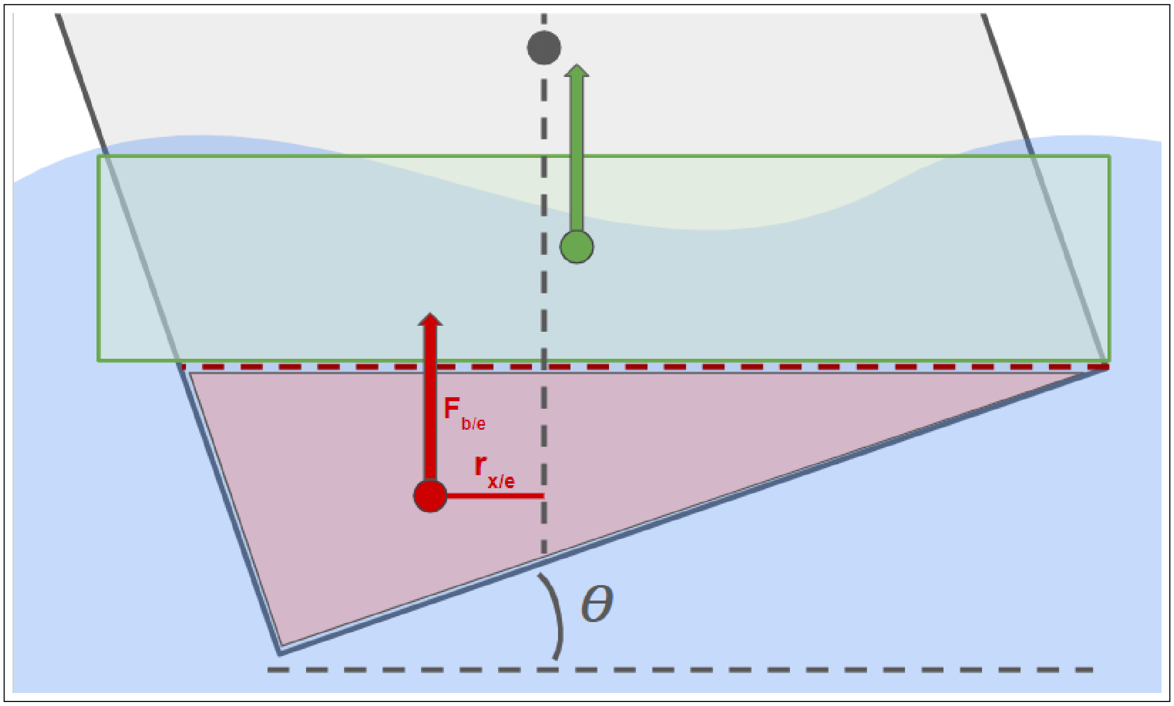


Figure 8: Illustration of submerged volume division. The volume highlighted in green corresponds to a buoyancy force, but no significant torque. The volume highlight in red corresponds to both a buoyancy force and resulting torque.

The buoyancy force on the upper rectangular volume (highlighted in green in Figure 8) acts through a local center of buoyancy very close to the boat's centerline. This is a result of the rectangular shape of the boat, as well as a small angle assumption (see *Assumptions*). The moment arm of that force is approximately zero and, therefore, exerts no significant torque on the boat and is neglected. The entire restoring torque on the boat can therefore be approximated by

$$\sum \tau \approx (F_{b,e})(r_{x,e}) \tag{22}$$

where $F_{b/e}$ and $r_{x/e}$ are the effective buoyancy force and moment arm, i.e. the force and moment arm caused by the lower, triangular section of the submerged volume (highlighted in red in Figure 8).

The magnitude of the effective buoyancy force is proportional to the effective submerged volume, $\forall_e$. More specifically, the effective buoyancy force was found to be:

$$F_{b,e} = \rho g \forall_e \tag{23}$$

Effective volume was found geometrically to be:

$$\forall_e = \frac{1}{2}Wh = \frac{1}{2}W^2 \tan(\theta) \tag{24}$$

Applying the small angle assumption as follows

$$\tan(\theta) = \frac{\sin(\theta)}{\cos(\theta)} \approx \frac{\theta}{1} = \theta \tag{25}$$

the volume can be simplified to the following:

$$\forall_e = \frac{1}{2}W^2\tan(\theta) \approx \frac{1}{2}W^2\theta \tag{26}$$

finally giving the approximate force to be:

$$F_{b,e} \approx \frac{1}{2}\rho g W^2 \theta \tag{27}$$

These geometric relationships may be more clearly observed in the following diagram of the effective submerged volume (Figure 9). The diagram shows the same red triangle as in Figure 8, but with more detailed dimensions, including the location of the triangle's centroid (center of buoyancy), which, for a right triangle, is definitionally one third of the base and one third of the height from the triangle's right angle (bottom left, in this case).

Restoring Torque Resulting from Buoyancy Force

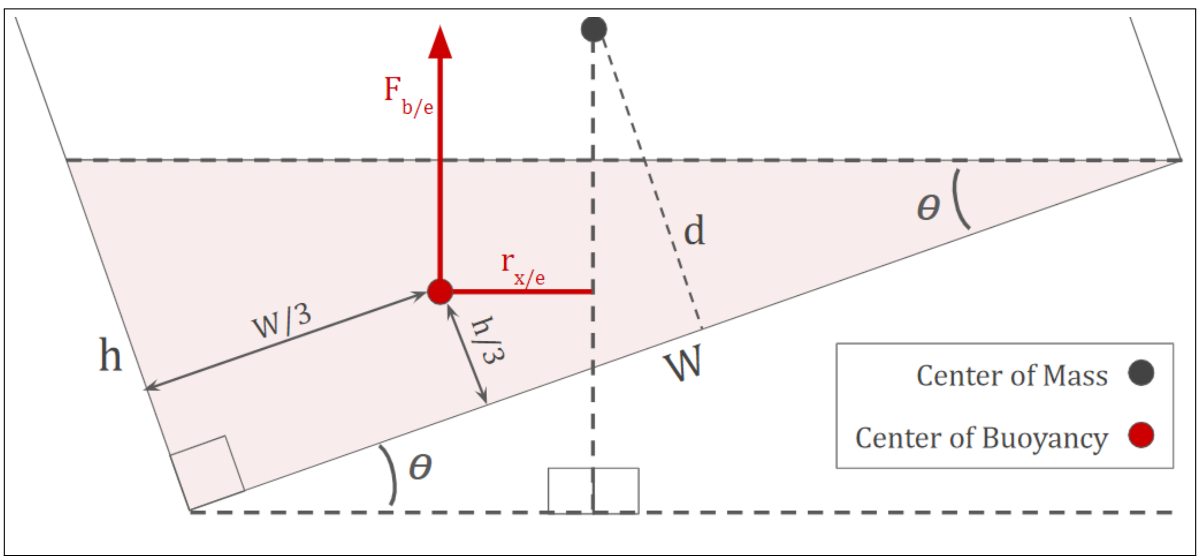


Figure 9: Detailed diagram of the effective submerged volume, with labeled dimensions and the triangle's centroid location (center of buoyancy).

The moment arm, $r_{x,e}$, can also be solved for geometrically based on Figure 9, producing the following expression:

$$r_{x,e} = \frac{W}{6}\cos(\theta) + \left(d + \frac{h}{3}\right)\sin(\theta) \tag{28}$$

which can be simplified using the same small angle assumption introduced earlier, by which $\sin(\theta) \approx \theta$ and $\cos(\theta) \approx 1$ for small θ.

$$r_{x,e} \approx \frac{W}{6} + \left(d + \frac{h}{3}\right)\theta \tag{29}$$

Combining the force and moment arm, the net restoring torque on the boat can be calculated:

$$\tau = (F_{b,e})(r_{x,e}) = \left(\frac{1}{2}\rho g W^2 \theta\right)\left(\frac{W}{6} + d\theta + \frac{h}{3}\theta\right) = \frac{1}{12}\rho g W^3 \theta + \frac{1}{2}\rho g W^2 \left(d + \frac{h}{3}\right)\theta^2 \tag{30}$$

The small angle assumption requires the assumption that $\theta^n \approx 0$ for $n > 1$, i.e. higher powers of θ will shrink to zero, requiring all but the first order terms to be discarded. Thus the above expression can be reduced to:

$$\tau = (F_{b,e})(r_{x,e}) \approx \frac{1}{12}\rho g W^3 \theta \tag{31}$$

As discussed earlier, the pitch and roll degrees of freedom can be treated identically, with a simple perspective-shift switching the boat length L and width W.

Making the comparison to a spring force of $F = kx$, the non-θ terms can be combined into a single spring constant. Thus, the boat's rotational spring constants for roll, k_r, and pitch, k_p, are:

$$k_r = \frac{1}{12}\rho g W^3 \qquad\qquad k_p = \frac{1}{12}\rho g L^3 \tag{32}$$

Initial Displacement Response

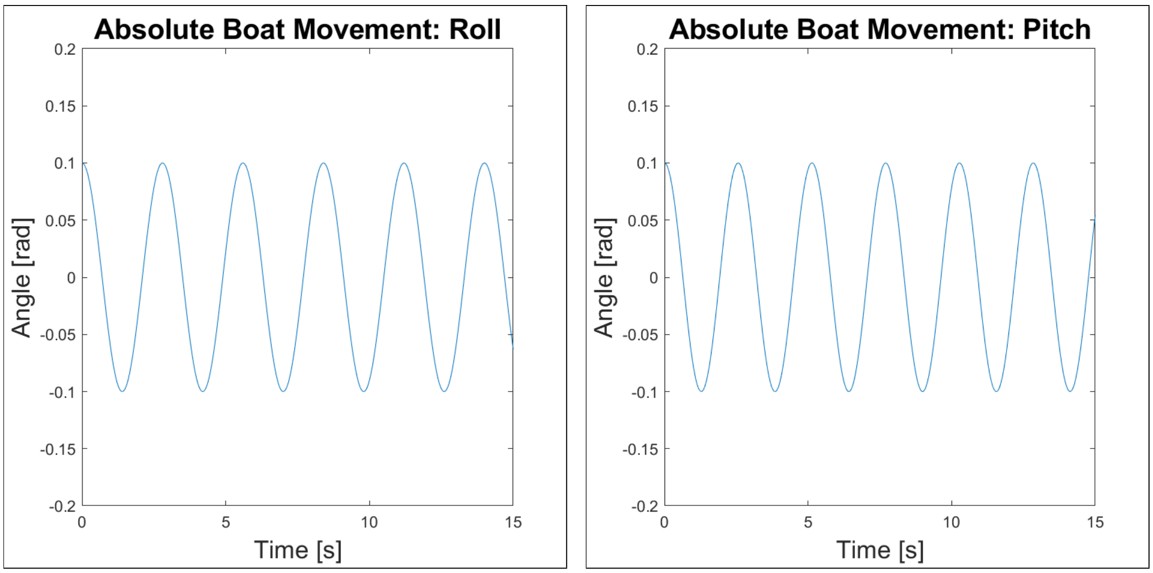
MATLAB was used again to model the boat rotational motion as a simple mass-spring system using the spring constants found above (32). This was done by finding the Laplacian impedance of the system. I_r is the roll moment of inertia, or the moment of inertia around the x axis.

$$I_r = \frac{1}{12}m(W^2 + H^2) \tag{33}$$

where $m = \rho WHL$. Once again, this equation can be adapted to the boat's pitch motion simply by exchanging width and length, yielding the following rotational impedance equations:

$$Z_r = I_r s + \frac{k_r}{s} \qquad\qquad Z_p = I_p s + \frac{k_p}{s} \tag{34}$$

Initial displacements can then be considered as a constant torque in each rotational dimension of magnitudes $k_r\theta_0$ and $k_p\phi_0$, and modelled by the same process used to model the heave response above (see *Initial Displacement Response*). As expected, the response to an initial rotational displacement was a simple sinusoidal oscillation (Figure 10). This was true for both roll and pitch.



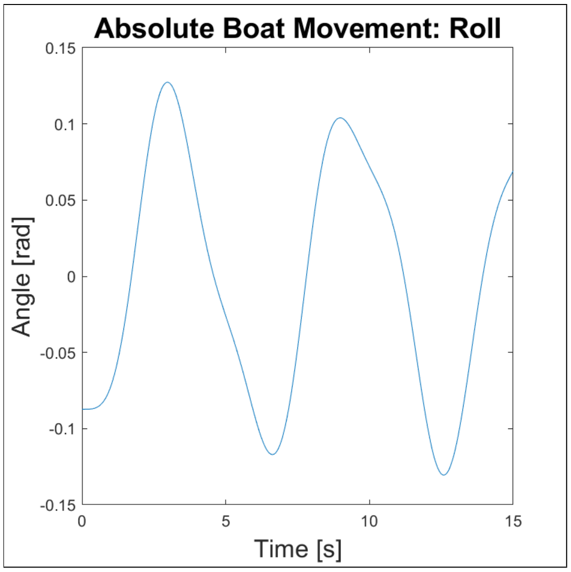
(a) Roll

(b) Pitch

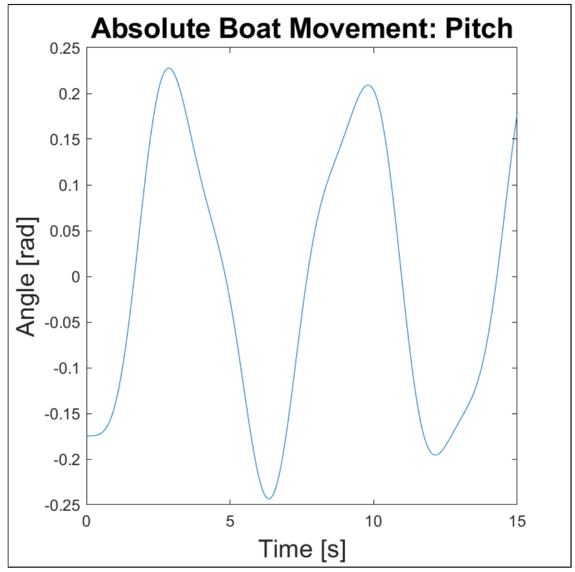
Figure 10: Roll (a) and pitch (b) responses to an initial displacement.

Forced Response

Again, as was done for heave, pitch and roll responses were simulated with forcing. The forcing function used was a cosine wave with a frequency significantly lower than the natural frequency of the boat's oscillation in either rotational degree of freedom. The responses are shown below.



(a) Roll



(b) Pitch

Figure 11: Roll (a) and pitch (b) responses to low-frequency sinusoidal forcing.

Combined System

The impedance matrix for the combined system can be created by combining the impedances found in equations (19) and (34) for each system into a single impedance matrix:

$$Z = \begin{bmatrix} Z_h & 0 & 0 \\ 0 & Z_r & 0 \\ 0 & 0 & Z_p \end{bmatrix} \tag{35}$$

The wave forcing will be a sine function for heave and cosine for roll and pitch, which accounts for slope (derivative) of the wave causing the roll/pitch.

$$F = \begin{bmatrix} A_h \frac{\omega}{s^2+\omega^2} \\ A_r \frac{s}{s^2+\omega^2} \\ A_p \frac{s}{s^2+\omega^2} \end{bmatrix} \tag{36}$$

where A_h, A_r, and A_p is the wave amplitude for heave, roll, and pitch, respectively.

The velocity and then position in the Laplace domain can then be found as:

$$V(s) = Y(s)F(s) \rightarrow X(s) = \frac{V(s)}{s} + \frac{x_0}{s} \tag{37}$$

where $Y(s) = [Z(s)]^{-1}$ and is the admittance matrix.

Using MATLAB's `impulse` function, the time domain response of the system can be determined. Plotting the results (Figure 12) it can be seen that all the systems are experiencing forced oscillation, as expected. Notice how the responses are the same as the forced responses of the individual motions seen earlier. This is because the three degrees of freedom are independent of one another. This is reflected in the impedance matrix which is a diagonal matrix (a system with dependent motions would have terms on the off diagonals corresponding to the dependencies, i.e. interactions between the degrees of freedom).

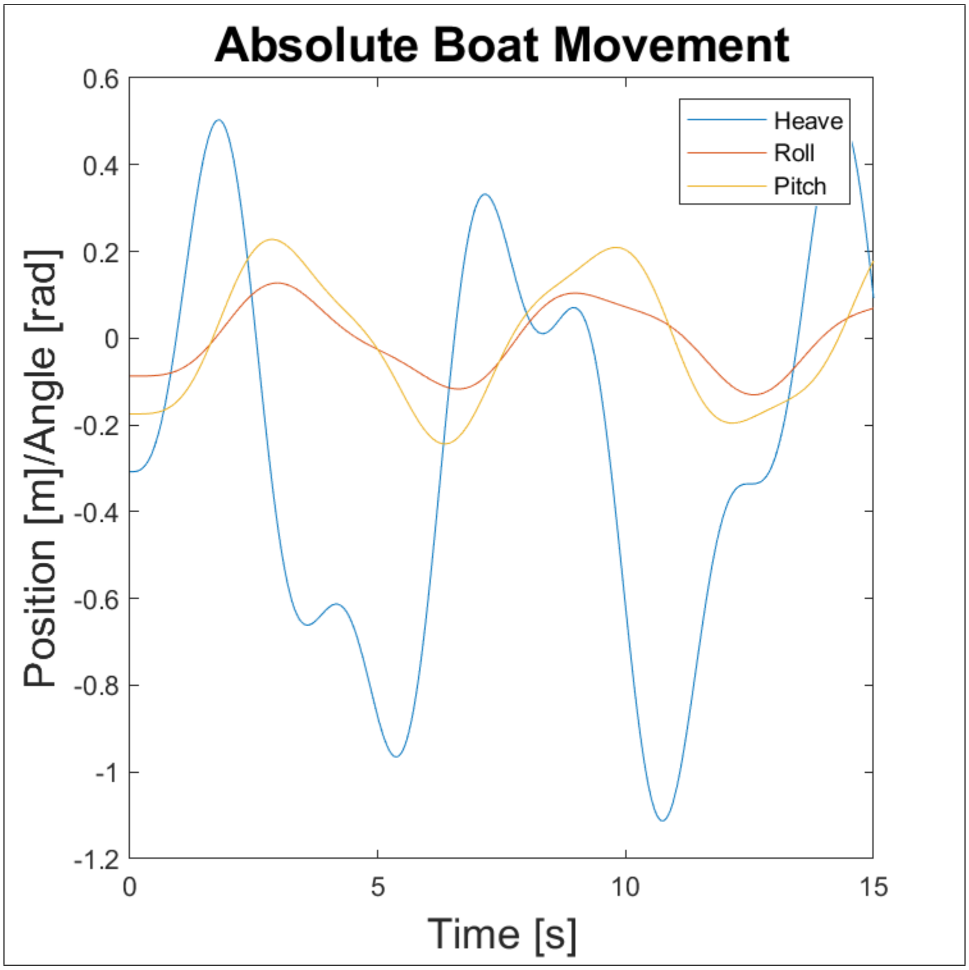


Figure 12: Heave (blue), roll (red), and pitch (yellow) position or angle versus time plot with $0.1m$ downward initial displacement in heave and $0.1rad$ initial roll/pitch angular displacement. This is the motion of the boat as viewed by an inertial reference frame.

Vibrations Analysis

Natural Frequency

There are two methods that can be used for finding the natural frequency of a vibrating system. The first is by looking at the differential equation of the system where, when in cononical form, the squared natural frequency is the coefficient on the non-differential term.

This can be seen in the generic second order ODE:

$$\ddot{q} + \underbrace{\frac{k}{m}}_{\omega_n^2} q = 0 \tag{38}$$

where q is the variable, k is the spring constant, and m is the mass. In the case of rotational motion, m is replaced by the moment of inertia, J. The above gives the equation for the natural frequency to be:

$$\omega_n = \underbrace{\sqrt{\frac{k}{m}}}_{\text{linear}} \text{ or } = \underbrace{\sqrt{\frac{k}{J}}}_{\text{rotational}} \tag{39}$$

Given some boat properties (the exact parameters can be seen later in Listing 1), the natural frequency for heave, roll, and pitch can be found to be $2.509 rad/s$, $2.244 rad/s$, and $2.447 rad/s$, respectively (Figure 13).

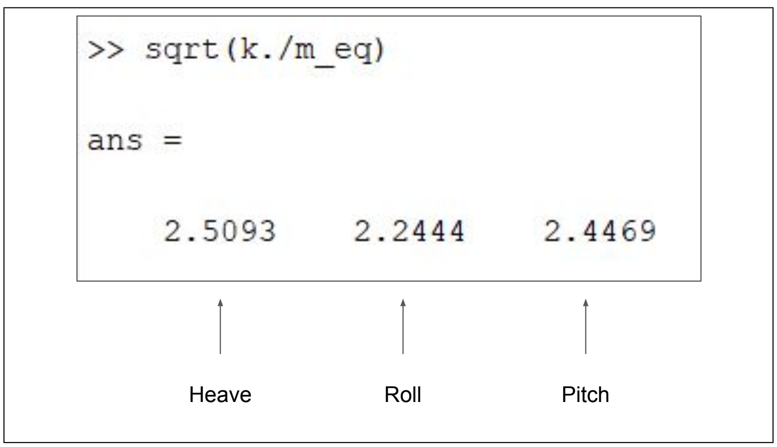


Figure 13: The natural frequency calculation of $\omega_n = \sqrt{\frac{k}{m}}$ performed in MATLAB. The k variable is the vector of spring constants for heave, roll, and pitch, and the m_{eq} vector contains the mass, moment of inertia for roll, and moment of inertia for pitch. The division is performed element-wise.

The second method for determining the natural frequency of the system is to analyze the system in the frequency domain using the frequency response function. The frequency response function, $H(s)$, is a function that describes how a system responds to inputs of various frequencies. The frequency response function can be calculated using the following equation:

$$H(s) = \frac{Y(s)}{s} \tag{40}$$

Performing this calculation in MATLAB, a bode plot can be made of the system (Figure 14). A bode plot shows the frequency response split into magnitude, which describes *how much* the system responds, and phase, which describes the *phase shift* of the response. Because the system is a 3×3 system, the bode plot is split into nine sections, each containing a magnitude and phase plot. The rows correspond to the system being analyzed and the columns correspond to what force the system is being analyzed with respect to. So, the diagonal elements correspond to each system in reference to itself, i.e. the first diagonal element purely corresponds to the heave response, second to roll, and third to pitch. The large peaks of the magnitude plots correspond to frequencies at which the system resonates. Using the MATLAB interface, the peaks were selected, which showed the corresponding frequencies. Compared to the natural frequencies calculated earlier, the values matched exactly.

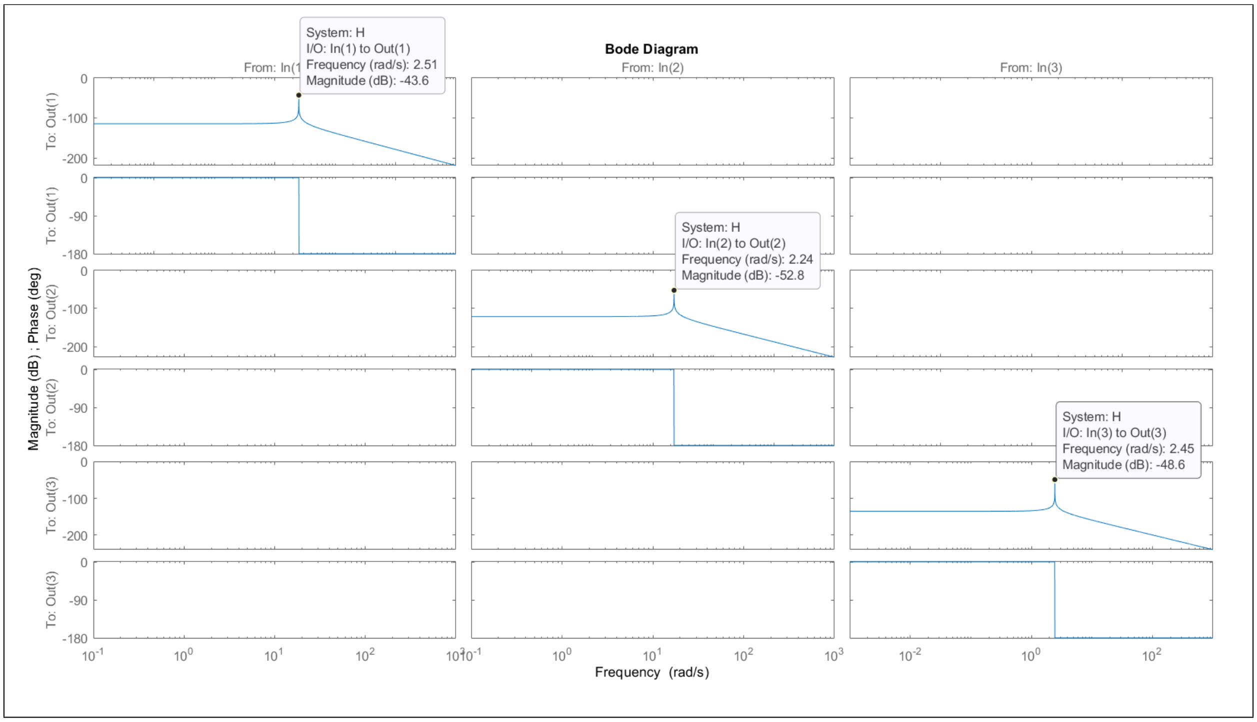


Figure 14: Bode plot of the frequency response function for the combined heave, roll, and pitch system. The selected peaks correspond to the frequencies at which the system resonates, i.e. the natural frequencies. Notice the empty off diagonal terms symbolizing the independence of each degree of response.

Notice how only the diagonal elements are filled while all the off diagonal elements are empty, just like the impedance matrix in equation (35). As mentioned earlier, the off diagonal terms correspond to responses with respect to another, so the first column second row corresponds to the roll response with respect to the heave forcing. Having all the off diagonals be empty corresponds to the three degrees of freedom being independent of one another. This is also reflected in the earlier impedance matrix where there were no interaction terms in the off diagonal cells.

Rather than driving at $1rad/s$ for all waves as was done earlier, the system is driven at the heave natural frequency, $2.5093rad/s$. The plot (Figure 15) shows heave resonate as well as pitch. Looking back (Figure 13), the pitch natural frequency is very close to that of the heave system, which is why both increase, although pitch increases slower.

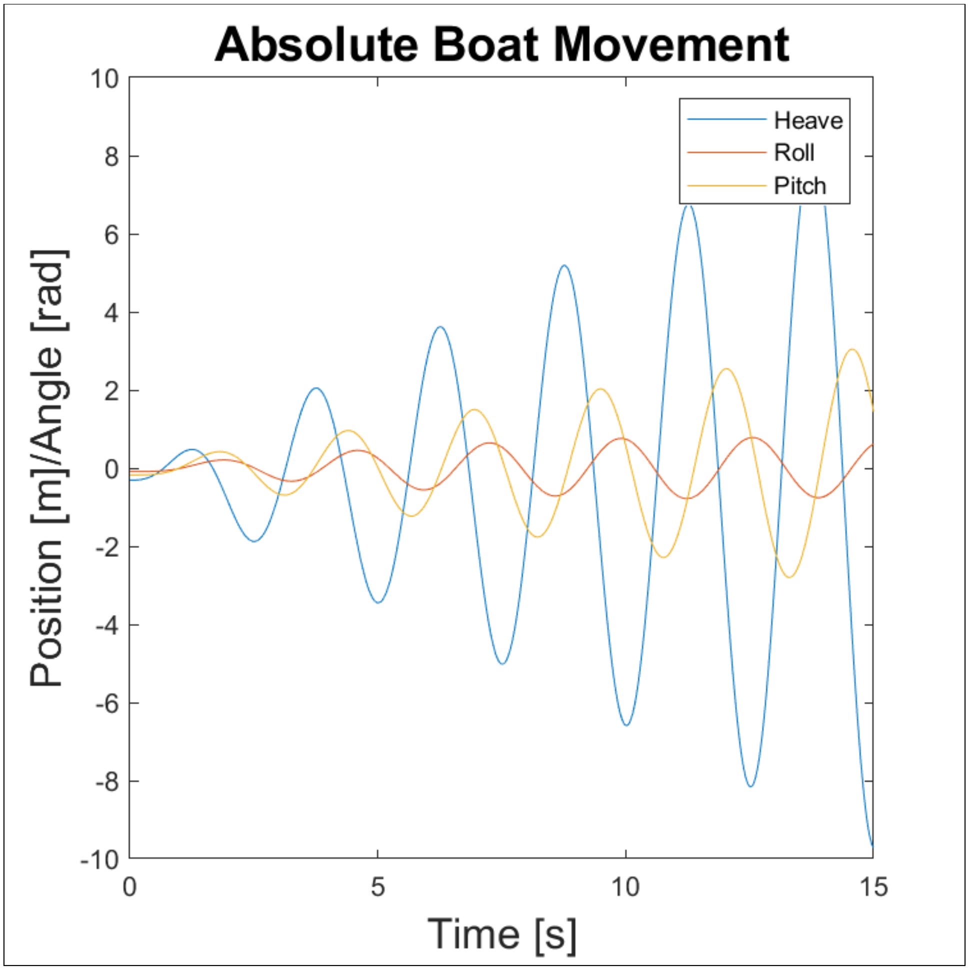


Figure 15: Driving the combine system at the heave natural frequency. The heave resonance is a given and the pitch resonance is caused by the system's natural frequency being very close to the that of heave.

Simulations

To provide further insight into the movements of the boat relative to a forcing wave and as a function of the boat dimensions and parameters, a series of simulations were created. Perhaps most crucially, these also provided the means for a series of simple and intuitive "gut-check" tests, which led to the resolution of such errors as the consideration of both the height and angle of the wave as sine functions, when in fact the wave surface angle is the derivative of the wave height, and should therefore be driven by a cosine function.

All animations were created with MATLAB. The animation process involves plotting the position of the boat for each time step, saving the resulting image to a vector of such images, or "frames", and using the `VideoWriter` tool to save the frames as an MPEG-4 file with a selected frame rate. The animations shown in the next section were generated using following simulation parameters, including boat dimensions, initial conditions, and wave features:

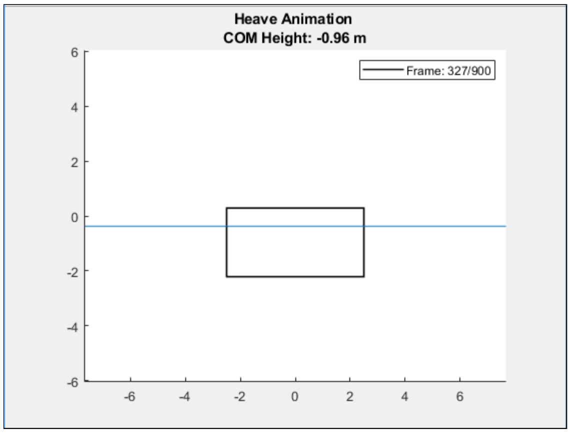
```matlab
%% Selected Parameters

% Boat dimensions
L = 11;                         % m, length of boat
W = 5;                          % m, width of boat
H = 2.5;                        % m, height of boat
rho = 640;                      % kg/m^3, density of boat (uniform)

% Initial conditions
x0 = [0, 0, 0];                 % initial position of [x, theta, phi]
v0 = [0, 0, 0];                 % initial d/dt of [x, theta, phi]

% System damping
R1 = 0.1;                       % Nms, Heave damping
R2 = 0.1;                       % Nms, Roll damping
R3 = 0.1;                       % Nms, Pitch damping

% Wave parameters
w = 1;                          % rad/s, wave frequency
max_height = .5;                % m, maximum height of wave
max_theta = 5*pi/180;           % rad, maximum theta angle of wave (roll)
max_phi = 10*pi/180;            % rad, maximum phi angle of wave (pitch)
```

Listing 1: Parameters used in the MATLAB simulations

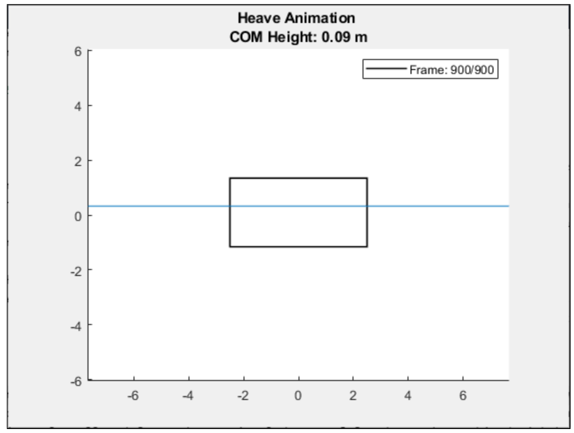
2D Animations

The first animations created were two-dimensional, and simulated each degree of freedom individually. These were useful for establishing an intuitive confirmation of each degree of freedom's functionality before attempting to combine them into a single system. Full recordings of the 2D animations can be viewed on YouTube following these links for heave, roll, and pitch.

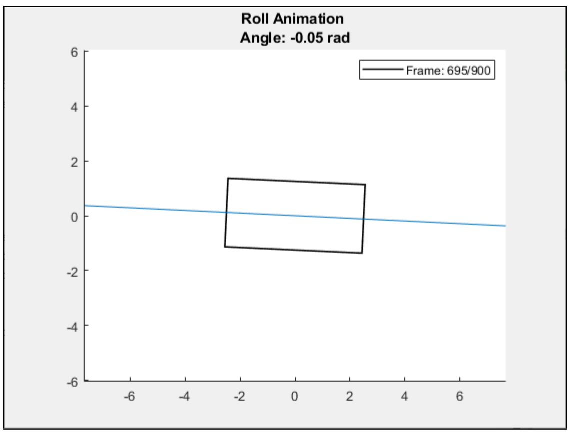
2D Animation Stills



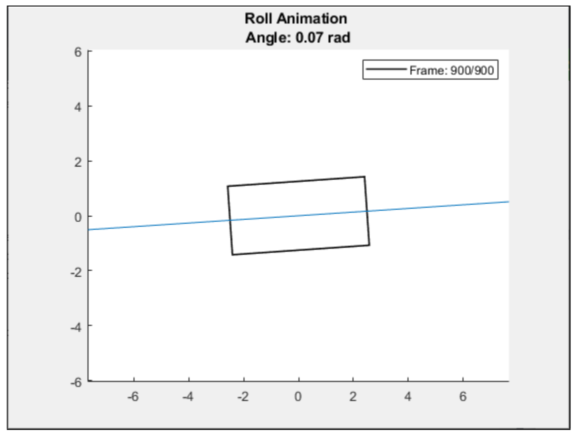
(a) Heave 1



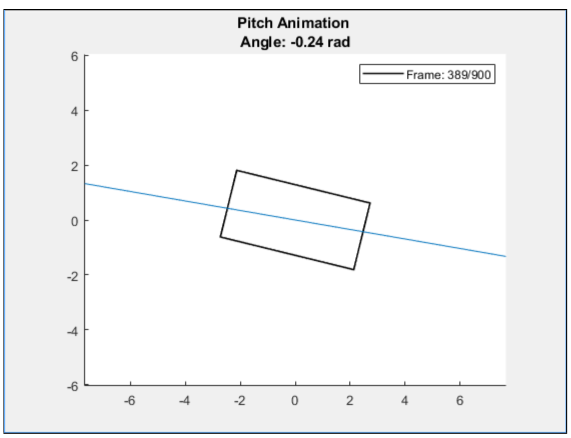
(b) Heave 2



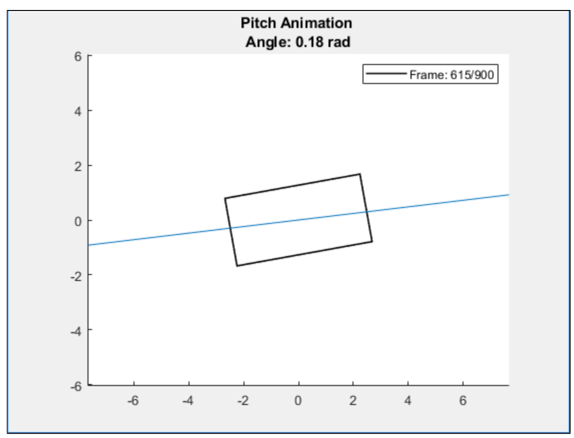
(c) Roll 1



(d) Roll 2



(e) Pitch 1



(f) Pitch 2

Figure 16: Pairs of screenshots from each of the three two-dimensional, 1-DOF animations performed with the given parameters (Listing 1).

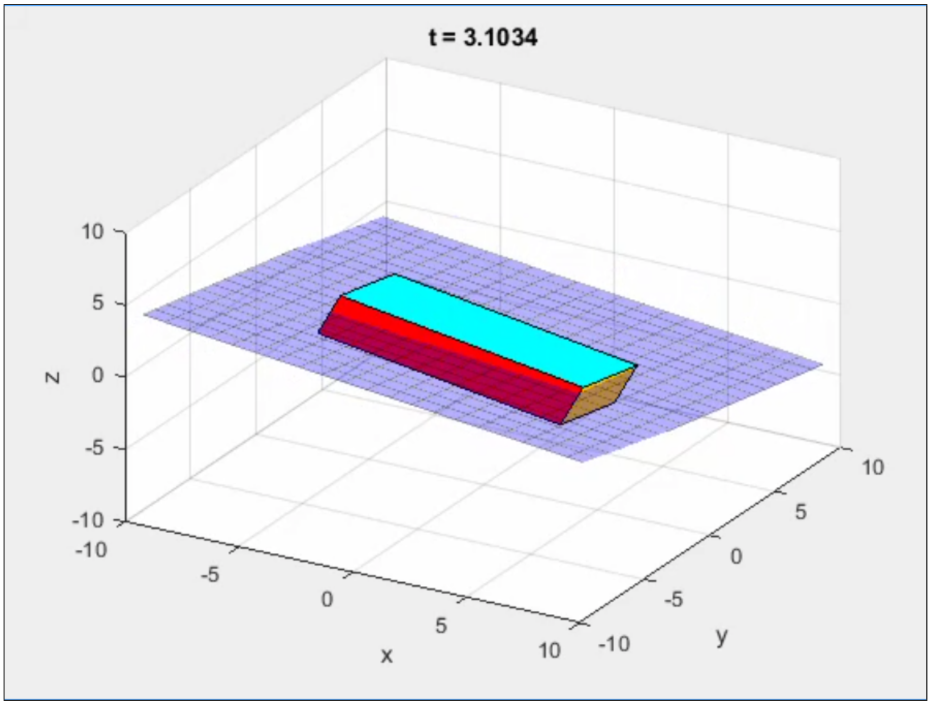
3D Animations

Once all three degrees of freedom could be modeled and animated independently, they were combined into a single 3-DOF system and animated in three dimensions. This animation proved to be both a satisfying culmination of the described model, as well as a highly potent debugging tool, leading to the identification and resolution of numerous errors and typos in the model construction.

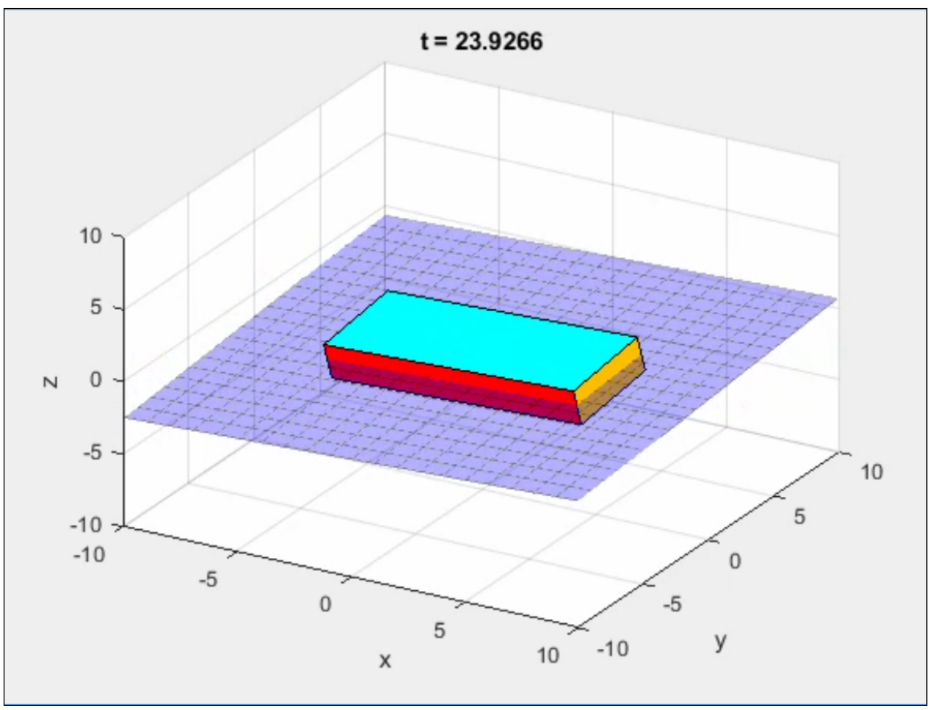
As discussed earlier, the boat was modeled as a uniform rectangular block and the wave as a moving plane. For clarity and consistency, these mathematical simplifications are also reflected in the animations. The simulation uses dimensions typical of a flat-bottomed boat. However, the density was chosen to be approximately that of a typical firm wood. Therefore, the boat in the animation can be expected to behave like a large solid wooden block.

The full recording of this 3D animation (Figure 17) can be viewed on YouTube. To see it, follow this link.

3D Animation Stills



(a)



(b)

Figure 17: Two still frames of a 3D, 3-DOF simulation using the same parameters as above (Listing 1) showing two different positions of the boat as a rectangular block and the wave as a plane.

For comparison, the simulation was repeated after varying the height of the boat. Note that the density of the boat was fixed, not the mass. Therefore, the mass of the boat changed significantly.

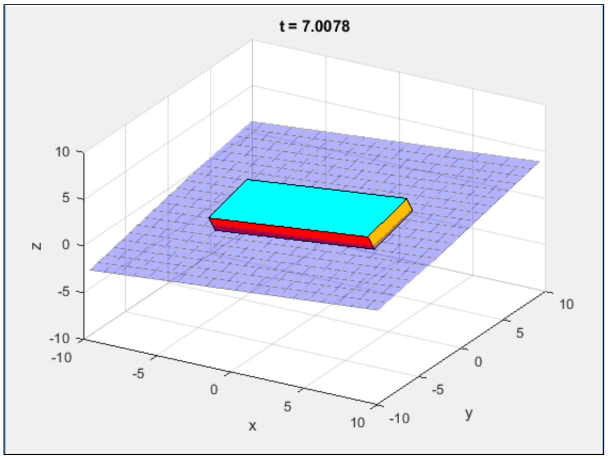
In particular, the height was first decreased from an initial value of $2.5m$ to just $1.5m$. This resulted in a flatter and lighter boat with natural frequencies approximately 30% higher in each degree of freedom. Because the system forcing (from the wave) was at a lower frequency, this change in dimensions resulted in a milder boat behavior. As seen below (Figures 18a and 18b), the lighter boat remains nearly co-planar with the forcing wave, while the boat's oscillations relative to the wave remain small.

The height of the boat was then increased to $5m$ for a final simulation with a thicker and heavier boat. This change resulted in a significantly lower natural frequency than the original simulation, and much closer to the forcing frequency. As expected, this resulted in more significant oscillations of the boat relative to the wave surface (Figures 18c and 18d). The angular displacements of the boat in this simulation became so extreme that they can be considered a violation of the small-angle assumption that was made in the construction of the model. That violation, particularly when combined with what may be an unrealistically low damping value, is the reason that the animation no longer appears to be realistic.

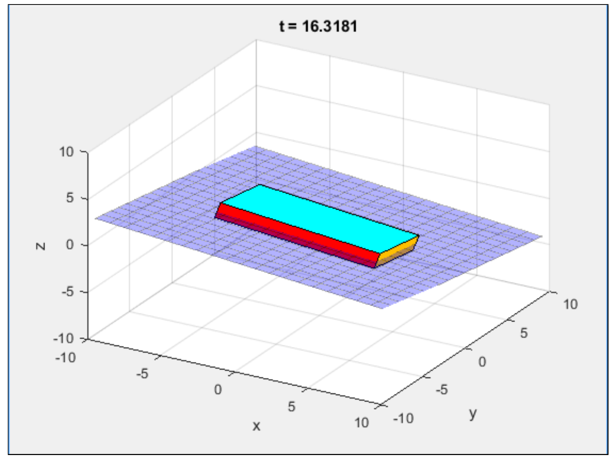
The full recorded animation for each of these simulations can be found on YouTube. Follow these links for the thin-boat simulation and the thick-boat simulation.

In further demonstrating the limitations that the small-angle assumptions place on this model, the simulation was run under conditions that severely violated those assumptions. The first of those simulations, linked here, models a wave with a maximum displacement angle of $\theta = 180°$ (affecting roll), but no displacement with respect to the other degrees of freedom. The second such simulation, linked here, exchanged roll for pitch and adds heave displacement to the model, further exacerbating the already drastically unrealistic nature of the resulting animation.

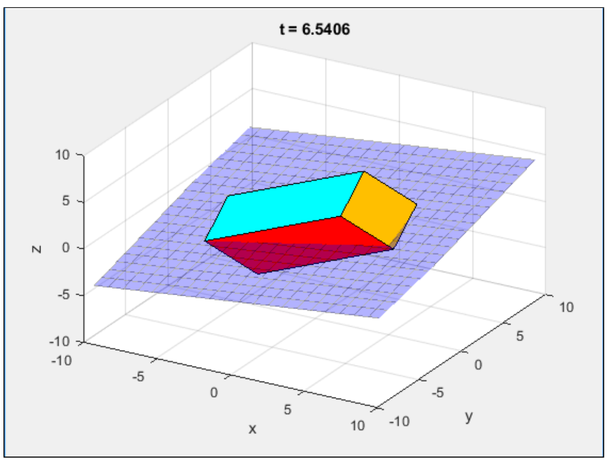
Modified-Thickness 3D Animation Stills



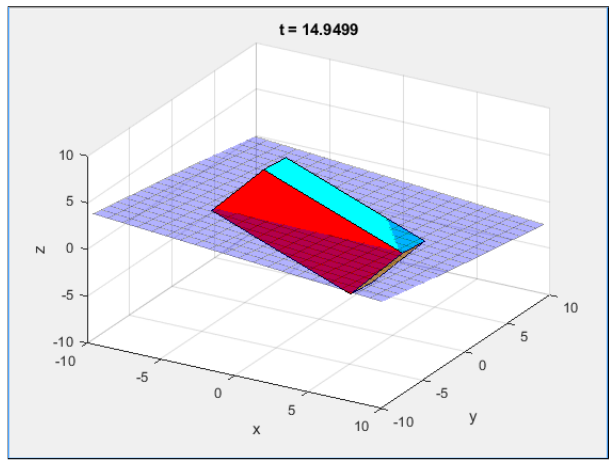
(a) Thin Boat 1



(b) Thin Boat 2



(c) Thick Boat 1



(d) Thick Boat 2

Figure 18: Still frames from the modified-thickness simulations discussed above, showing (a)-(b): the relatively co-planar motion of the thin boat relative to the forcing wave, and (c)-(d): the comparatively extreme oscillations of the thick boat relative to the forcing wave.

Conclusion

Overall, a 3-DOF boat model was developed, showing the heave, roll, and pitch of a boat in response to displacements and wave forcing. Small angle approximations simplified the models to a manageable level for simulating and modeling within the scope of the class and project. The natural frequencies were determined based on the ODEs and frequency response functions. The combined system showed the same response of each system individually since the degrees of freedom were independent (a consequence of the small angle approximations). When forced at the natural frequency of heave, resonance was observed in heave, as expected, and pitch, due to the similar natural frequencies of the two systems. Final two dimensional and three dimensional simulations were produced, clearly displaying the motion of the models and the effects of parameter changes on the model dynamics.

Future Work

Without the small angle assumptions, the minor effects of each system on one another would produce interesting and more realistic results due to their interactions. Additionally, adding the remaining three degrees of freedom to the boat dynamics would further complicate the system, adding to the accuracy and intricacies of how an actual boat responds to various wave inputs. Those two changes may be difficult to investigate using Laplace equations, so time-domain ODEs could be solved with an ODE solver and the vibration dynamics determined in the time-domain using eigenvalue analysis of the 6-DOF system.

Furthermore, forgoing the some of the simplifying assumptions used would allow the equations to be generalized and to describe other boat shapes. This would be a crucial step in making this model more true-to-life and of potential use in modeling real scenarios. This achievement would pose challenges both with regards to the difficulty of describing and implementing the equations correctly, as well as the computational demands that such a model would surely place on the host computer.

Appendix: MATLAB Code

```
clear all; close all; clc;

s = tf('s');

%% Selected Parameters

% Boat dimensions
L = 11;                           % m, length of boat
W = 5;                            % m, width of boat
H = 2.5;                          % m, height of boat
rho = 640;                        % kg/m^3, density of boat (uniform)

% Initial conditions
x0 = [0, 0, 0];                   % initial position of [x, theta, phi]
v0 = [0, 0, 0];                   % initial d/dt of [x, theta, phi]

% System damping
R1 = 0.1;                         % Nms, Heave damping
R2 = 0.1;                         % Nms, Roll damping
R3 = 0.1;                         % Nms, Pitch damping

% Wave parameters
w = 1;                            % rad/s, wave frequency
max_height = .5;                  % m, maximum height of wave
max_theta = 5*pi/180;             % rad, maximum theta angle of wave (roll)
max_phi = 10*pi/180;              % rad, maximum phi angle of wave (pitch)

% Simulation parameters
tf = 15;                          % s, time span of simulation
n = 900;                          % number of timesteps
lims = [10, 10, 10];              % x, y, and z axis limits (+/-)
fps = 30;                         % frame rate of animation video
filename = 'figures/Boat';        % name of generated mp4 file


%% Derived and Fixed Parameters

% Environmental constants
g = 9.81;       % m/s^2, gravity
rho_w = 1027;   % kg/m^3, saltwater density

% Boat parameters (https://dynref.engr.illinois.edu/rem.html)
m = W*L*H*rho;                    % kg, mass of boat
I_theta = (1/12)*m*(W^2+H^2);     % kg*m^2, roll moment of inertia
I_phi = (1/12)*m*(L^2+H^2);       % kg*m^2, pitch moment of inertia

% System spring constants
k1 = rho_w*g*W*L;               % kg/s, Heave spring constant
k2 = (1/12)*rho_w*g*L*W^3;      % Nm/rad, Roll spring constant
k3 = (1/12)*rho_w*g*W*L^3;      % Nm/rad, Pitch spring constant

```

```matlab
% Vector forms
m_eq = [m, I_theta, I_phi];      % "mass-equivalent", i.e. - m or I
R = [R1, R2, R3];
k = [k1, k2, k3];
A = [max_height, max_theta, max_phi];

% Adjust initial conditions to make them absolute
x0 = x0 + [H/2-m*g/k1, -max_theta, -max_phi];

% Adjust limits vector for more convenient use
lims = [[-1, 1]; [-1, 1]; [-1, 1]] .* lims';

%% Laplace Domain

func = [w, s, s];

% Impedance matrix
Z1 = m*s + k1/s + R1;            % Heave impedance
Z2 = I_theta*s + k2/s + R2;      % Roll impedance
Z3 = I_phi*s + k3/s + R3;        % Pitch impedance
Z = [Z1, 0, 0;
     0, Z2, 0;
     0, 0, Z3 ];

% Effective forces
F0 = (k.*x0/s - m_eq.*v0)';          % Effects of initial conditions
F1 = max_height*k1*w/(s^2+w^2);      % Heave forcing
F1 = F1 - (rho_w*g*L*W*H/2-m*g)/s;  % Height equilibrium adjustment
F2 = max_theta*k2*s/(s^2+w^2);       % Roll forcing
F3 = max_phi*k3*s/(s^2+w^2);         % Pitch Forcing
F = F0 + [F1, F2, F3]';              % Comlete laplacian force vector

V = Z\F;
X = V/s + x0'/s;


%% Run Simulation

% Time frame
t = linspace(0, tf, n)';

% Generate response
h = sin(w*t)*A(1);           % Wave height h wrt time
theta = cos(w*t)*A(2);       % Wave angle theta wrt time
phi = cos(w*t)*A(3);         % Wave angle phi wrt time
wave = [h, theta, phi];
x = impulse(X, t);           % Position of boat wrt time

% Plot response
figure(1)
plot(t, x)
xlabel("Time [s]","FontSize",16)
ylabel("Position [m]/Angle [rad]","FontSize",16)
title("Absolute Boat Movement","FontSize",18)
```

```
legend(["Heave", "Roll", "Pitch"])
set(gcf, "PaperPosition", [0 0 5 5]);
set(gcf, "PaperSize", [5 5]);
saveas(gcf, "figures/Absolute.png");

figure(2)
plot(t, x-wave)
xlabel("Time [s]","FontSize",16)
ylabel("Position [m]/Angle [rad]","FontSize",16)
title("Boat Movement Relative to Wave","FontSize",18)
legend(["Heave", "Roll", "Pitch"])
set(gcf, "PaperPosition", [0 0 5 5]);
set(gcf, "PaperSize", [5 5]);
saveas(gcf, "figures/RelativeCombined.png");

names = ["Heave", "Roll", "Pitch"];
for indx=1:3
    figure(100+indx)
    plot(t, x(:,indx))
    title("Absolute Boat Movement: " + names(indx),"FontSize",18)
    xlabel("Time [s]","FontSize",16)
    if names(indx) == "Heave"
        ylabel("Position [m]","FontSize",16)
    else
        ylabel("Angle [rad]","FontSize",16)
    end
    set(gcf, "PaperPosition", [0 0 5 5]);
    set(gcf, "PaperSize", [5 5]);
    saveas(gcf, "figures/Absolute" + names(indx) + ".png");

    figure(200+indx)
    plot(t, x(:,indx) - wave(:,indx))
    title("Boat Movement Relative to Wave: " + names(indx),"FontSize",18)
    xlabel("Time [s]","FontSize",16)
    if names(indx) == "Heave"
        ylabel("Position [m]","FontSize",16)
    else
        ylabel("Angle [rad]","FontSize",16)
    end
    set(gcf, "PaperPosition", [0 0 5 5]);
    set(gcf, "PaperSize", [5 5]);
    saveas(gcf, "figures/Relative" + names(indx) + ".png");
end

%% Vibration Analysis
Y = inv(Z);                  % Admittance matrix
w = [0:0.001:1000];          % frequency vector

% Bode Plot
figure()
bode(Y/s, w);    % H = Y/s; Frequency response function

%% 2D Animation
frames = length(t);
```

```
x_max = max(x);

% Loop to create a 2D animation of each Heave, Roll, and Pitch
for indx=1:length(names)
    % Determine plot limits to fit everything nicely
    lim = max([x_max(indx), W/2, H/2, L/2]);
    lim = 1.1*lim;

    fig = figure("Name",names(indx) + "2D","NumberTitle","off");
    % https://www.mathworks.com/matlabcentral/answers/94495-how-can-i-
   create-animated-gif-images-in-matlab#answer_103847
    filename = "figures/" + names(indx) + '.gif';
    for frame=1:frames
        clf;
        if names(indx) == "Heave"
            draw_rectangle([0, x(frame,1)], W, H, 0);
            hold on
            draw_angled_line([0, wave(frame,1)], 3*lim, 0);
            hold off
        else
            draw_rectangle([0, 0], W, H, x(frame,indx));
            hold on
            draw_angled_line([0, 0], 3*lim, -wave(frame,indx));
            hold off
        end

        xlim([-lim,lim])
        ylim([-lim,lim])
        axis equal
        legend(["Frame: " + frame + "/" + frames])
        if names(indx) == "Heave"
            title(["Heave Animation " ...
                "COM Height: " + round(x(frame,1),2) + " m"])
        else
            title([names(indx) + " Animation " ...
                "Angle: " + round(x(frame,indx),2) + " rad"])
        end

        movieVector(frame) = getframe(fig, [0,0,560,420]);
    end

    myWriter = VideoWriter(filename, 'MPEG-4');
    myWriter.FrameRate = fps;

    open(myWriter);
    writeVideo(myWriter, movieVector);
    close(myWriter);
end

%% 3D Animation

fig = figure("Name","3D Animation","NumberTitle","off");

% Create animation slide for each timestep
```

```
for k=1:length(t)

    % Reset frame
    clf
    hold on

    % Plot surfaces of box (boat)
    [boat_vertices, boat_faces] = boatCoords(W, L, H, num2cell(x(k,:)));
    patch('Vertices', boat_vertices, 'Faces', boat_faces, 'FaceVertexCData
    ',hsv(8),'FaceColor','flat')

    % Plot wave plane
    [X, Y, Z] = waveCoords(num2cell(wave(k,:)), lims);
    surf(X, Y, Z, 'FaceColor','b','FaceAlpha',.3,'EdgeAlpha',.3)

    % Format the plot
    grid on
    xlim(lims(1,:))
    ylim(lims(2,:))
    zlim(lims(3,:))
    xlabel('x')
    ylabel('y')
    zlabel('z')
    title(['t = ', num2str(t(k))])
    view([30 35])

    movieVector(k) = getframe(fig, [0,0,560,420]);

end

myWriter = VideoWriter(filename, 'MPEG-4');
myWriter.FrameRate = fps;

open(myWriter);
writeVideo(myWriter, movieVector);
close(myWriter);


%% Animation Functions

% Return boat vertices and faces for use in patch function
function [vertices, faces] = boatCoords(W, L, H, p)

    % Boat position and orientation
    [h, theta, phi] = p{:};

    % Every n-length permutation of 1 and -1, where n=3
    % -- Never changes - can be moved outside of function/loop
    [a, b, c] = ndgrid([-1, 1]);
    direction = [a(:), b(:), c(:)];

    % Axis rotation matrices
    rotx = [1, 0, 0; 0, cos(phi), sin(phi); 0, -sin(phi), cos(phi)];
    roty = [cos(theta), 0, sin(theta); 0, 1, 0; -sin(theta), 0, cos(theta)
```

```
];

    % Vertex locations relative to boat orientation
    vertices = direction.*[L/2, W/2, H/2] + [0, 0, h];
    vertices = (roty*(rotx*vertices'))';

    % Identify faces by their vertices (based on directions list)
    % -- Never changes - can be moved outside of function/loop
    faces = [   1, 2, 4, 3;
                5, 6, 8, 7;
                1, 2, 6, 5;
                3, 4, 8, 7;
                1, 3, 7, 5;
                2, 4, 8, 6];
end

function [X, Y, Z] = waveCoords(q, lims)
    % Position and orientation of wave
    [h, theta, phi] = q{:};

    % Generate meshgrid plane
    xgrid = lims(1,1):1:lims(1,2);
    ygrid = lims(2,1):1:lims(2,2);
    [X,Y] = meshgrid(xgrid, ygrid);
    Z = zeros(size(X));
    plane(:,1,:) = X;
    plane(:,2,:) = Y;
    plane(:,3,:) = Z;

    % Axis rotation matrices
    rotx = [1, 0, 0; 0, cos(phi), sin(phi); 0, -sin(phi), cos(phi)];
    roty = [cos(theta), 0, sin(theta); 0, 1, 0; -sin(theta), 0, cos(theta)];

    % Rotate plane
    plane = pagemtimes(plane,rotx);
    plane = pagemtimes(plane,roty);

    % Return final X, Y and Z matrices
    X = squeeze(plane(:,1,:));
    Y = squeeze(plane(:,2,:));
    Z = squeeze(plane(:,3,:)) + h*ones(size(X));

end

% angled rectangle source
% https://www.mathworks.com/matlabcentral/answers/380257-how-do-i-draw-and-rotate-a-rectangle#comment_750053
function []= draw_rectangle(center_location,width,height,theta)
    center1=center_location(1);
    center2=center_location(2);
    R = ([cos(theta), -sin(theta); sin(theta), cos(theta)]);
    X=([-width/2, width/2, width/2, -width/2]);
    Y=([-height/2, -height/2, height/2, height/2]);
```

```
    for i=1:4
        T(:,i)=R*[X(i); Y(i)];
    end
    x_lower_left=center1+T(1,1);
    x_lower_right=center1+T(1,2);
    x_upper_right=center1+T(1,3);
    x_upper_left=center1+T(1,4);
    y_lower_left=center2+T(2,1);
    y_lower_right=center2+T(2,2);
    y_upper_right=center2+T(2,3);
    y_upper_left=center2+T(2,4);
    x_coor=[x_lower_left x_lower_right x_upper_right x_upper_left];
    y_coor=[y_lower_left y_lower_right y_upper_right y_upper_left];
    patch('Vertices',[x_coor; y_coor]','Faces',[1 2 3 4],'Facecolor','none
    ','Linewidth',1.2);
    axis equal;
end

function []=draw_angled_line(center_location, x_length, theta)
    x_center = center_location(1);
    y_center = center_location(2);
    x1 = x_center-x_length/2;
    x2 = x_center+x_length/2;
    y_length = x_length*tan(theta);
    y1 = y_center-y_length/2;
    y2 = y_center+y_length/2;
    plot([x1, x2], [y1, y2]);
end
```