MAE 3260 Final Group Work: Exploring a System of Interest
Report

Title: "Orinoco Flow" - Enya
People: Maggie Huggins, Kyra McCarrick, Gina Liu, and Kaila Danielson

Topic of Interest: 1D & 2D Autonomous Sailboat

Abstract: This project will focus on developing an autonomous sailboat. (1) Determine a block diagram and ODE that accounts for the inputs and outputs of the watercraft (wind, current, sail angle, speed, etc.) as well as model assumptions (2) We will simulate the one dimensional and two dimensional models of a sailboat. (3) Create open and closed loop controllers (rudder angle, sail angle, etc.).

Students/Roles:

Student	Task/Role
Gina	Create transfer functions and block diagrams describing the 1D instance of the sailboat.
Kaila	Describe model assumptions and simulate the 1D model using a MATLAB script.
Maggie	Create and compare the 1D and 2D models.
Kyra	Simulate the 2D model and help create the models.

List of MAE 3260 concepts or skills used in this group work:

- Models:
  - Model simplifying assumptions
  - ODEs
  - TFs
  - State space
  - Block diagrams
- Closed-loop system:
  - MATLAB simulation
  - Steady state behavior (staying at constant reference velocity)
  - Saturation of input
- Active control:
  - LQR feedback control law
  - Command following (following reference velocity)
  - Disturbance rejection

Model Description and Assumptions

We used a number of assumptions to model our one and two dimensional autonomous sailboats. This allowed us to reduce the complexity of the governing physics and create a simplified model to study. For both models, we worked on a restricted x-y plane, ignoring forces and positions that did not directly contribute to our simplified models of position and velocity. This mainly included the forces of gravity and buoyancy, as well as our vertical orientation and acceleration.

For our one-dimensional model, we modeled a sailboat that travels in a straight-line path. To do this, we assumed that the angle of the sail only controls the force of thrust from the wind (the "power" of the boat) and is independent of the direction the boat moves in. We also make the assumption that the rudder of the watercraft is always parallel to the course such that the boat travels in the x-direction only. This allowed us to ignore the effects of the rudder in this model. The speed of the boat is determined by the portion of the sail that is normal to the wind direction. We also assumed the wind was moving parallel to the boat at all times, with varying wind speeds. This leads us to a single equation of motion for this system:

$$\ddot{x} = \frac{1}{m}\left(U(t)_{wind}^{2} A_{sail} sin(\theta(t)) - b\dot{x}\right)$$

This was found by applying Newton's second law of motion to the sailboat, stating that the force of the wind on the sail added to the force of the drag is equal to the mass times the acceleration of the boat.

For the two-dimensional model, we also made many assumptions about our model in order to simplify the math. Once again, we assumed that the wind was moving in a constant direction (again parallel to the ship's path), as well as stating that the drag coefficient (from both air and water) is unchanging as the boat moves through the water. Our goal was to simulate the boat moving in a straight line against forces like wind, so we found that it was acceptable to apply the small angle approximation as the boat shouldn't stray from motion in the x-direction very much. Using the small angle approximation we were able to linearize the equations of motion found in "Control Algorithms for a Sailboat Robot with a Sea Experiment," giving us this system of equations:

$$\dot{X} = V + \alpha_d a$$

$$\dot{y} = v\theta + \alpha_d a \psi$$

$$\dot{\theta} = \omega$$

$$\dot{V} = \frac{f_s \delta_v - f_r (v_i) - \alpha_v V^2}{M}$$

$$\dot{\omega} = \frac{f_s(P_6 - P_7) - P_8 f_r - \alpha_w \omega}{J}$$

After finding these equations we created the state space equation:

$$\vec{X} = \begin{bmatrix} x \\ y \\ \theta \\ v \\ w \end{bmatrix}, \quad \vec{U} = \begin{bmatrix} f_s \\ f \end{bmatrix}$$

$$\dot{X} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & V_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{2\alpha_v}{M} & 0 \\ 0 & 0 & 0 & 0 & -\frac{\alpha_w}{J} \end{bmatrix} \vec{X} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{\delta_v}{M} & \frac{U_1}{M} \\ \frac{P_6 - P_7}{J} & -\frac{P_8}{J} \end{bmatrix} \vec{U}$$

Overall, we found the 1D model of the sailboat system to be much more approachable than the 2D model. Furthermore, both models assumed that the ship was not straying from a straight course, which is arguably not realistic. In reality, as well, there are many other forces acting on the boat as it moves which we neglect.

1D Sailboat Simulation

For our simulations, we used ChatGPT to generate MATLAB code of our 1D autonomous sailboat that changes the angle of its sail based on varying wind speeds to keep the boat at a constant velocity. We based the MATLAB code off of our equation of motion of our 1D model, and our assumptions which are stated above. Because we assume that the sail angle controls the power/velocity of the boat and is independent of changing the direction of the boat, we made 3 graphs: one of the speed of the wind over time (independent variable), one of the responding sail angle over time, and one of the velocity of the boat over time. The velocity graph should ideally look like a straight line as the sail angle corrects itself to the corresponding wind speed to keep the boat at a constant velocity. We used random values for the mass and dimensions of our sailboat, an arbitrary function for wind, and adjusted our reference velocity for different results.

Overall, this system acted like a closed-loop control system, where the sail angle acted as the control input by adjusting the thrust to push the velocity toward a reference value U_{ref}. The wind in this case is a disturbance to the system. Shown below are two differing results with U_{ref} = 4m/s (Figure 1) and 2m/s (Figure 2), respectively. As we can see in Figure 1, when the wind speed is at its minimum, the sail angle saturates which causes a dip in the velocity of the boat. This is because the demand for the control (sail angle) was too high for the system's physical capabilities, meaning that there was no angle of the sail that could produce enough thrust from the wind to keep the boat at 4 m/s. During this time where the input is saturated, the system acts more like an open-loop system.

When we lowered the reference velocity to 2 m/s in Figure 2, we can see that this is no longer the case and that the wind speed is sufficiently high enough the whole time for the control input (sail angle) to keep the boat at the reference velocity. This is a much more stable value for the system. The steady state for both of these are their respective reference velocities.

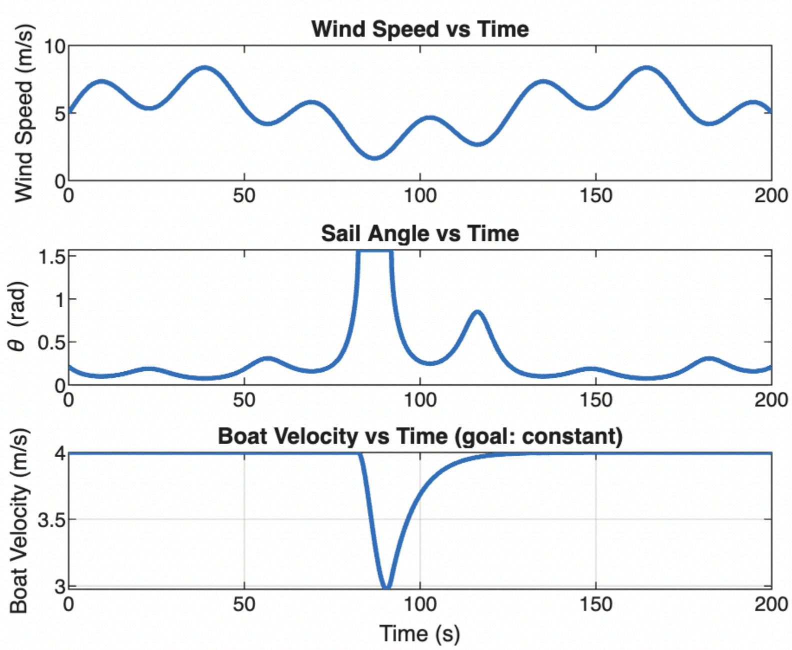


Figure 1: 1D Sailboat Simulation with U_{ref} = 4m/s

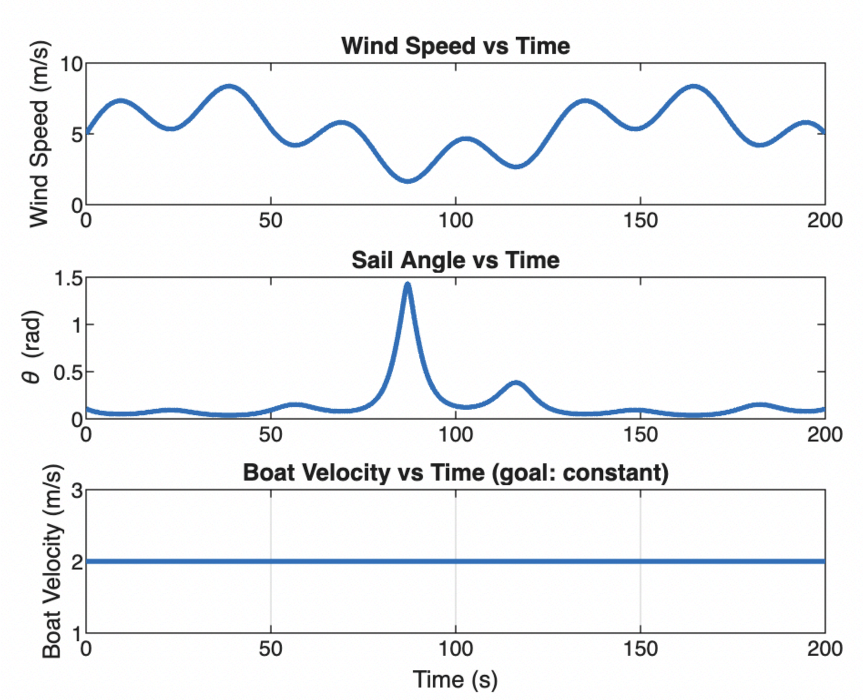


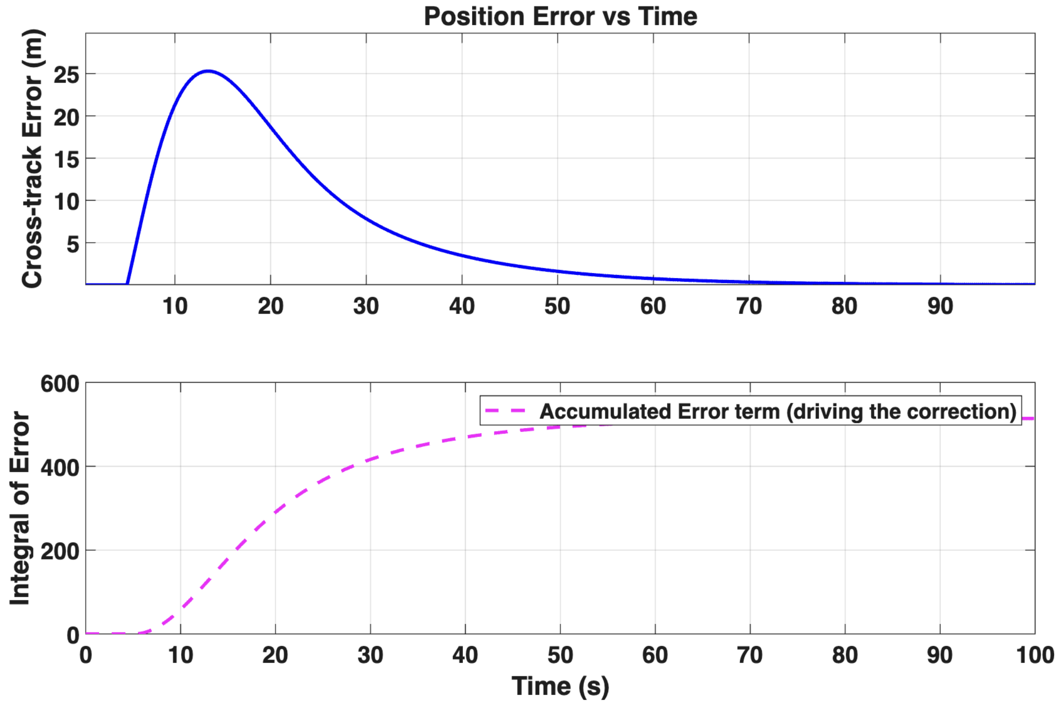
Figure 2: 1D Sailboat Simulation with U_{ref} = 2m/s

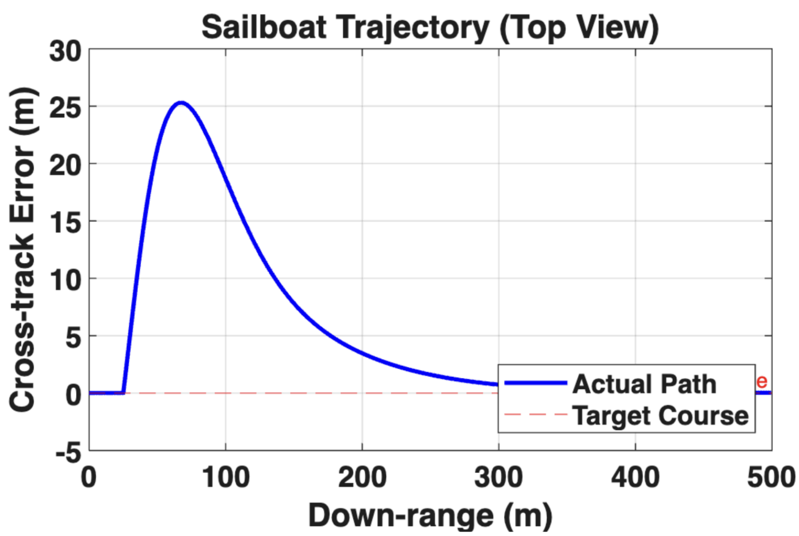
2D Sailboat Simulation

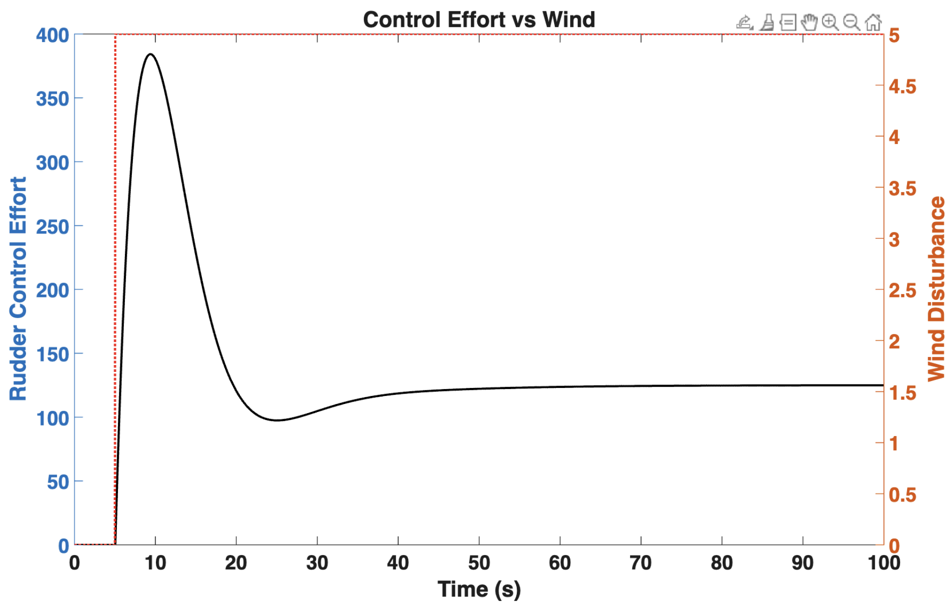
Using our state space model, we asked Google's Gemini to make a MATLAB Live script similar to the racecar live script from class. We wanted to be able to control the LQR values and see how they impacted the course and recovery of the sailboat given certain wind disturbances or step inputs. We refined our requests a few times before we were happy with the output. Here are the results from that code:

Q_y	500
Q_theta	70000
Q_int	3
R_rudder	1

```
Controller Gains Calculated successfully.

--- Simulation Summary ---

Max displacement from course: 25.30 meters

Final steady-state error:     0.0321 meters

SUCCESS: The boat returned to track!
```


Position Error vs Time
Cross-track Error (m)
Integral of Error
Accumulated Error term (driving the correction)
Time (s)


Sailboat Trajectory (Top View)
Cross-track Error (m)
Actual Path
Target Course
Down-range (m)



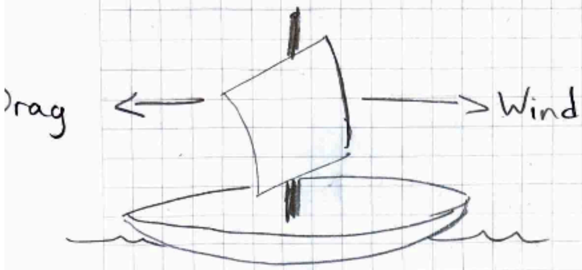
As we can see from the diagrams, the sailboat initially overshoots and deviates from the target course, however, it corrects itself over time. The wind acts as a disturbance on the system. Over time, the course of the boat goes to steady state which aligns with the target course.

1D Model Brainstorm

We used Professor Brian Kirby's textbook, *How Fluid Machines Work,* 2025 edition, to try to make a more accurate model of the drag and lift force acting on the 1D model of the sailboat (assuming the sail acts as a bluff body and not an airfoil). We found drag coefficients for a 60 degree cone (the hull in water), and used the drag coefficients for a cup anemometer in its 180 degree and 0 degree orientations to model the lift force and the drag force on the sail. Our control inputs here are the angle of the sail, the wind force, and our output is the boat position x. We then realized that the drag and lift terms are dependent on the relative wind speed, which is U-xdot, not just U, so we added that in.

Final Groupwork

1D Model



$\Sigma \vec{F} = ma$

, Wind − Drag $= m\dot{v}$

→ variable of interest is speed

→ wind force proportional to area of sail

From p. 493 Fluid Dynamics Textbook (Kirby)

"how cup anemoters work" → modeling sail as a cup

$C = \frac{F}{\frac{1}{2}\rho U^2 A}$ $\quad F = \frac{1}{2}\rho U^2 A C$ → flow → $C = 1.4$ in this orientation

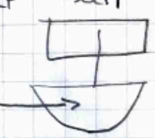
$F_w = \frac{1.4}{2}\rho_w U_w^2 A_{sail} = .7\rho_w V_w^2 A_{sail}$

(to model drag on other side of sail)

flow $\quad C = -.4$ $\qquad$ U in this case is speed of boat, not wind

$F_{\text{drag due to sail}} = \frac{-.4}{2}\rho_{wind} V^2 A = -.2\rho_{wind} V_{boat}^2 A_{sail}$

$F_{\text{drag due to hull}} = -\frac{1}{2}\rho_{water} V_{boat}^2 A_{boat}$ $\qquad A_{boat}$



$.7\rho_{wind} U_{wind}^2 A_{sail} - V_{boat}^2 (.2\rho_{wind} A_{sail} + .5\rho_{water} A_{boat})$

$= m\dot{V}_{boat}$ $\qquad \rho_{wind} \approx 1\,kg/m^3 \quad \rho_{water} \approx 1000\,kg/m^3$

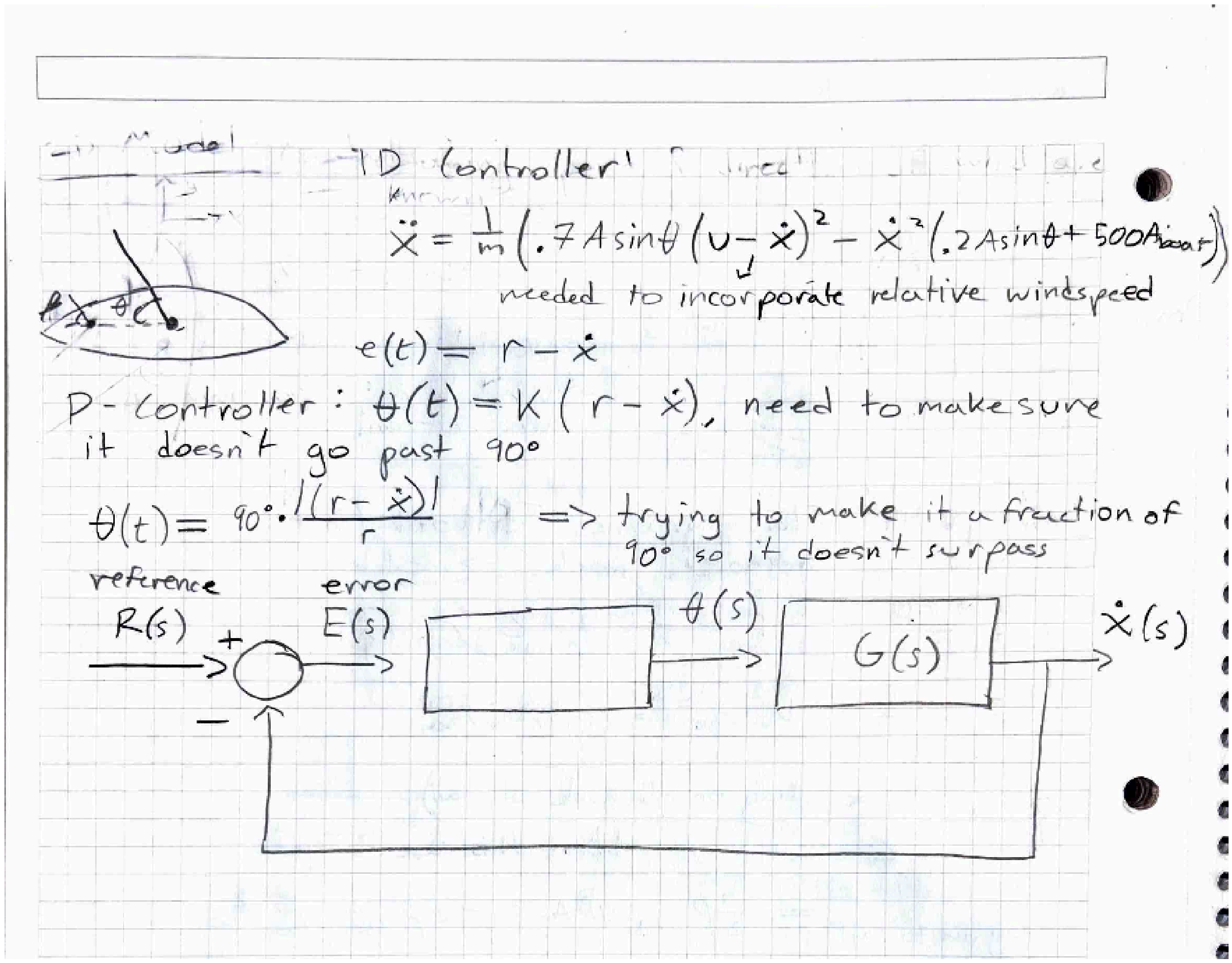
Using drag coefficient of 60° cone for the hull from Kirby's textbook

θ is control input $\qquad A_{sail} = A\sin\theta$

$\ddot{x} = \frac{1}{m}\left(.7A\sin\theta U^2 - \dot{x}^2(.2A\sin\theta + 500A_{boat})\right)$

X is boat position, U is wind speed, θ is angle of sail

output $\qquad$ input (we don't control) $\qquad$ input (we control



1D Model

Preliminary Block Diagram

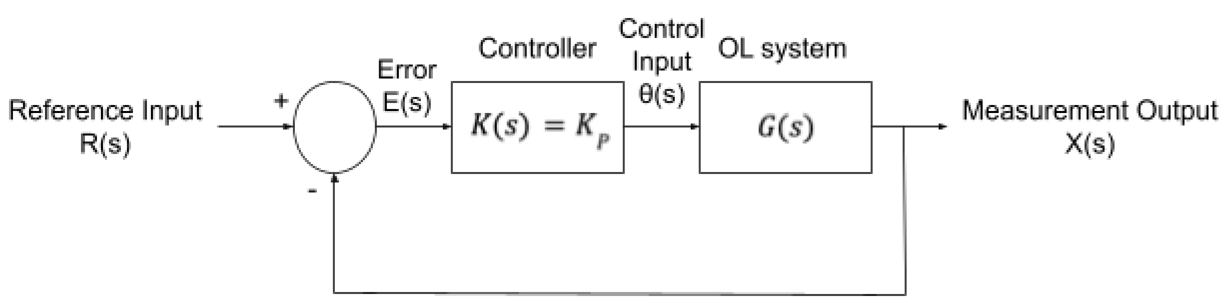


Figure 3: Block diagram using proportional control with unknown G(s).

Simplification

To simplify analysis, we will use the ODE that is linearized with less focus on fluid mechanics as stated in Model Description and Assumptions:

$$\ddot{x} = \frac{1}{m}\left(U(t)^2_{wind} A_{sail} sin(\theta(t)) - b\dot{x}\right)$$

where U_{wind} is the wind speed, A_{sail} is the sail area, θ is the angle of sail, and b is the drag coefficient.

Transfer Functions

X(s)/F(s)

Since $U(t)^2_{wind} A_{sail} sin(\theta(t))$ is dependent on time but can only be known through data collection, let $f(t) = U(t)^2_{wind} A_{sail} sin(\theta(t))$ and $F(s) = \mathcal{L}\{f(t)\}$.

The ODE then becomes:

$$\ddot{x} = \frac{1}{m}\left(f(t) - b\dot{x}\right)$$

Using Laplace Transformations, we get:

$$s^2X(s) - sx(0) - \dot{x}(0) = \frac{1}{m}\left[F(s) - b(sX(s) - x(0))\right]$$

By ignoring the initial condition, the equation becomes:

$$s^2X(s) = \frac{1}{m}\left[F(s) - b(sX(s))\right]$$

And rearranging the equation to obtain:

$$s^2X(s) + \frac{1}{m}b(sX(s)) = \frac{1}{m}F(s)$$

$$X(s)[s^2 + \frac{1}{m}(bs)] = \frac{1}{m}F(s)$$

$$\frac{X(s)}{F(s)} = \frac{1}{ms^2+bs}$$

G(s)

From the block diagram, we can find that:

$$G(s) = \frac{X(s)}{\theta(s)}$$

To simplify the calculations, use the small angle approximation such that *sinθ* = *θ* and let *u(t)* = $U_{wind}(t)^2$*.* This results in:

$$\ddot{x} = \frac{1}{m}\left(u(t)A_{sail}\theta(t) - b\dot{x}\right)$$

As with the previous transfer function, *u(t)* is dependent on time and can only be deduced through data collection. As such,

$$U(s) = \mathcal{L}\{u(t)\}.$$

Using Laplace transformation and ignoring initial conditions:

$$s^2X(s) = \frac{1}{m}\left[U(s)A_{sail}\Theta(s) - b(sX(s))\right]$$

Rearranging this equation, we get:

$$X(s)[s^2 + \frac{1}{m}(bs)] = \frac{1}{m}(U(s)A_{sail}\Theta(s))$$

$$\frac{X(s)}{\Theta(s)} = \frac{U(s)A_{sail}}{ms^2+bs}$$

Therefore, the transfer function, G(s), is

$$G(s) = \frac{U(s)A_{sail}}{ms^2+bs}$$

Block Diagram for 1D

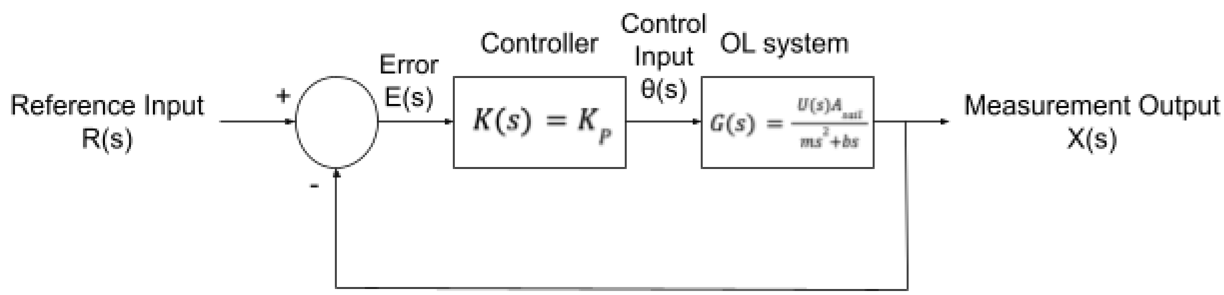


Figure 4: Block diagram using proportional control with known G(s).