A Control Method for a Bermuda Rigged Sailing Boat

Yinuo Zheng Zoe Loughton Ollie Jaques Alfred Brown

November 18, 2020

Abstract

Whilst sailing a boat, the rudder must be adjusted to turn the boat to the correct direction, changing both the angle of the rudder and the boat. The problem is how the change in angle of rudder leads to the turning of a moving boat over time, and how one can operate this process automatically. Here we use MATLAB to build a rudder control mechanism that processes the error between the current direction and required direction that the sailing boat is travelling in. Our model then produces the correct angle for the rudder to move in order to turn the boat to the required direction. We use proportional and derivative control methods to build the model and control our rudder angle. We then improve it to a more realistic model by considering varied conditions that may influence the boats turning performance. This an immature product of an automatic direction adjustment system for sailing boats.

1 Introduction

Direction control systems are an important device in modern navigation of sailing boats. Creating a rudder control model that can handle and feedback by adjusting the rudder to a proper angle, after receiving the target direction, will help people to control the direction of the sailing boat.

This report will exhibit the creation and developments of this control model. We aim to design a direction control model to automatically adjust the direction of a sailing boat by controlling the rudder angle. To do this we built our initial model. In the initial model, a proportional control method is applied to simulate the reaction of a sailing boat's direction when changing the rudder angle. Then the performance of this model is developed by introducing a derivative control method. After that, researches in the limitation of the rudder angle and the effect of drag are carried out to make the model more realistic. Lastly, other factors will be considered, such as latency and noise, which counters for a lack of accuracy in the compass signals. The performance of the direction control model is tested by giving unstable and delayed setting direction. Then, the model is further developed to minimize the influence of those factors.

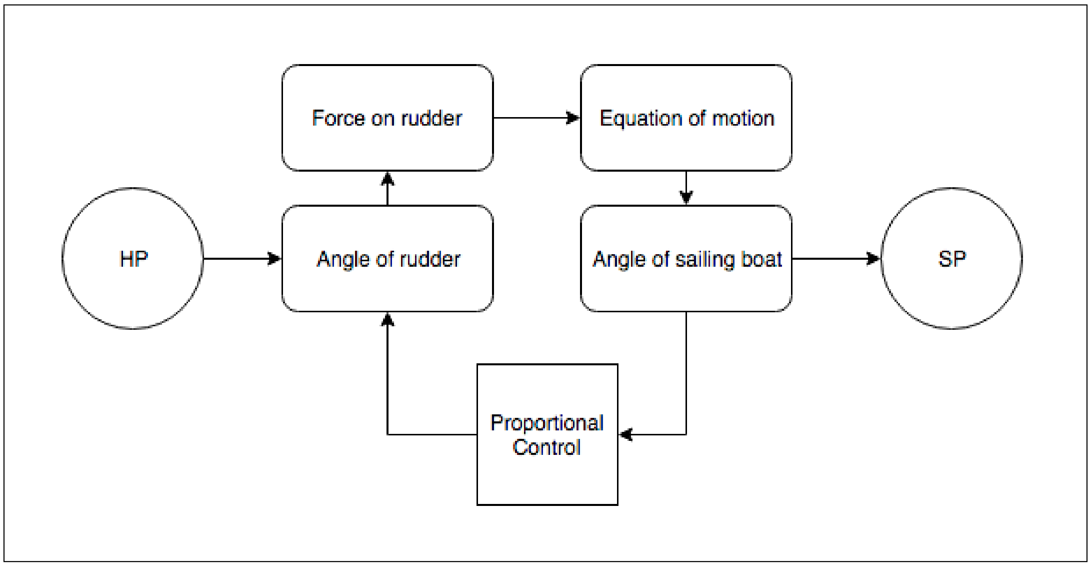
This control model can be used in real life to control the direction of a sailing boat. However, the application of this model is restricted to the assumptions made in the derivation of the control mechanism. In the real world the direction a sailing boat travels is the consequence of a very complex set of variables, which are very difficult to model. This means that the performance of this control model in real life is unpredictable and unreliable. The model needs more tests and improvements before applied to real world.

2 Model

2.1 A general proportional control method for a sailing boat

The derivation for our model will be broken down into five sections for more clarity. The model will be generalised to work for any sailing boat by using generalised variables and proportionality constants. Furthermore, it will work as a correctional method if the boat has a set direction (SP) and is diverging towards another direction (HP) *and* if the sailing boat wishes to turn from one direction (HP) to different direction (SP).To test the model, actual data for a Bermuda rigged sailing boat will be substituted into the variables and the results will be plotted. The main steps analysed in the model are summarised in the following flow chart below:

Figure 1: An overview of the processes our control method will be undertaking in order to keep the sailing boat on course.



2.2 Proportional control and angle of rudder

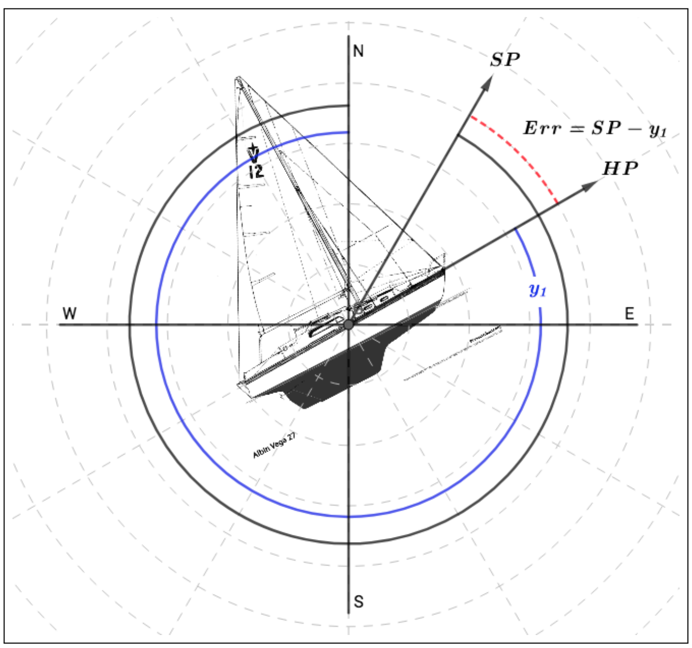


Figure 2: A diagram of the directional conventions we will be using for our model and an example of how the error is calculated. All angles will be taken west of north, as is the convention in sailing. ***SP*** is the so-called set point of the boat and represents the direction (as an angle west of north) in which the boat is supposed to travel. ***HP*** represents the heading point and is the *actual* direction (whose angle west of north is represented by y_1) in which the boat is travelling. The error ***Err*** is the difference between the set point and the heading point.

[1] As visually represented above, the error of the boat can be calculated as the difference between the set point and the heading point. The set point is a fixed angle that remains constant at all times relative to a compass. This may be determined by a compass on-board the boat or by a GPS signal. The heading point, i.e. the angle that the boat is currently heading towards, instead, is dependent on the dependent variable y_1 - the angle of the boat at an instance of time $t = t_n$, $n \geq 0$.

[1]The image for this sailing boat was found under $http://cdn.bluewaterboats.org/gallery/albin-vega-27/albinvega27-lines.gif$ [accessed 3. Dec. 2017] and the diagram was created using GeoGebra's online graphing calculator

Since the angle of the boat, y_1, is dependent on the angle of the rudder relative to the boat, we can formulate a relationship between the angle of the rudder and the error:

$$R_a = G_p Err = G_p(SP - y_1) \tag{1}$$

where R_a is the rudder angle relative to the boat and G_p is the proportional gain, which scales the error angle and depends on the properties of the boat being analysed. The convention we are using for the rudder angle is that R_a is positive if the rudder rotates left from its neutral position and negative if it rotates to the right. This means that if the boat needs to rotate to the right, indicating that the set point angle is less than y_2 and thus $Err < 0$, then the rudder will need to rotate to the right too in order to achieve this and will thus, by our convention, also be less than zero, showing indeed that the sign convention of equation (1) is correct. The rudder is in neutral when $R_a = 0$ and in an ideal world, since we are neglecting rolling motion, should result in the sailing boat moving straight ahead.

2.3 Determining the force on the rudder

In order to model the turning moment of the sailing boat we need to model the resultant force that acts on its rudder. The following assumptions will be made in order to model this force:

- The boat and rudder are stream-lined bodies and no eddies are produced at the stern of the boat, such that there is a uniform distribution of forces on the rudder.
- The density of water is constant. There are no debris or other particulates swimming around in the water.
- The maximum force on the rudder is produced when the $R_a = \pm 90°$ and there is zero force on the rudder when $R_a = 0$.

Thus, it is proposed that:[2]

$$F_r = K\rho_w A_r V_{rel}^2 sin(R_a) \tag{2}$$

where,
- F_r is the magnitude of the resultant force on the rudder
- K is a constant of proportionality.
- ρ_w is the density of the fluid through which the boat is moving.
- A_r is the surface area of the rudder facing into the fluid stream.
- V_{rel}^2 is the square of the relative velocity of the rudder. The relative velocity is the difference of the velocity vectors that describe the velocity of the water and the boat.
- $sin(R_a)$ is the sin of the rudder angle and can also be expressed as $sin(G_p(SP - y_1))$.

Hence, we can also re-write F_r as,

$$F_r = K\rho_w A_r V_{rel}^2 sin(G_p(SP - y_1)) \tag{3}$$

which allows for the control method to be applied. A visualisation of the force is shown in Figure 3, where $|\underline{V}_{water}|$ is the magnitude of the velocity of the water and $|\underline{V}_{boat}|$ is the magnitude of the velocity of the boat

NB, the units for F_r are $\left[\frac{kg}{m^3}\right][m^2]\left[\frac{m^2}{s^2}\right] = [kgms^{-2}] = N$, which is indeed the correct unit for a force.

2.4 Rotational drag force on the boat

As the sailing boat begins to turn, due to a rotation of the rudder, there will also be a resistive drag force that opposes this turning motion. This rotational drag is mainly caused by the underside of the sailing boat, which acts as a buff body [Jon17]. In simple terms, this means that the surrounding fluid finds it difficult to move around the body, resulting in a large drag effect. (This is also why large sailing boats have large flat keels, in order to stop the boat from rolling over). Since there are two fluids acting on a sailing boat at all times, namely the water and the air, rotational drag can also be caused by the sails. To turn a sailing boat into the wind is near to impossible (and dangerous) if the sails are orthogonal to the wind because the drag force can become so large that it will roll the boat over. An experienced sailor will know that one needs to release the sails to swing freely, so as to then let the

[2] This model is based on another model that can be found under [Żel14]

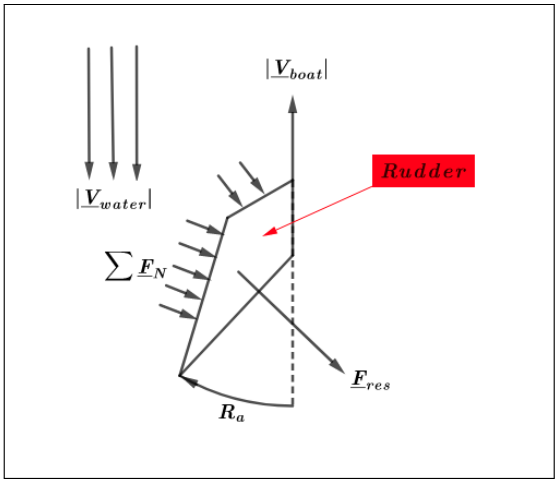


Figure 3: A visual representation of the forces on the rudder. Note that all the forces produced by the momentum of the water particles on the rudder act normal to the rudder surface and that the sum of these forces, $\sum \underline{F}_N$, is equivalent to a resultant force acting through the centre of mass of the rudder.

sails become parallel to the wind and reduce the drag. For our model of the rotational drag, we will assume that an experienced sailor is in command and we can thus neglect the drag produced by the air and focus only on the drag produced by the water.

The standard equation for drag is,

$$D = \frac{1}{2} C_D \rho_w V_{rel}^2 A_D \tag{4}$$

where,
-C_D is the drag coefficient
-ρ_w is the density of the fluid - in this case water.
-V_{rel} is the relative velocity of the boat to the velocity of the water.
-A_D is the reference area and usually just the projected frontal area of the body submerged in the fluid.[3]

Since we are interested in how the drag is affected by the rotation of the boat, we propose to substitute the angular velocity of the boat for one of the linear velocity terms in the equation above. Furthermore, we will only be considering the density as an infinitesimal sheet of fluid on the reference area of the boat, so as to remain dimensionally correct. Since the angular velocity can be described as $\dot{y}_1$, the equation for drag now becomes:

$$D = \frac{1}{2} C_D \rho_w V_{rel} A_D \dot{y}_1 \qquad NB, \left[\left[\frac{kg}{m^2} \right] \left[\frac{m}{s} \right] [m^2] \left[\frac{1}{s} \right] = N \right] \tag{5}$$

2.5 Determining the equation of motion

The equations that govern the motion of a sailing boat are extremely complex because they are dependent on a myriad of different variables. Since a sailing boat has 6 degrees of freedom in its motion; roll, pitch and yaw motion, which describe its rotational freedom; and its ability to move left-right (drifting), up-down (bobbing) and back and forth (bad sailing), which describe its translational freedom. For this model, however, we are only interested in the sailing boats rotational yawing motion, so we will neglect all other forms of motion (for now). Thus, our assumptions are:

- The sailing boat does not bob, drift, pitch or roll at any point in time.
- Consequently, the sailing conditions are ideal. There are no waves or gusts of wind.
- The sailing boat acts as one rigid body at all times, including the rudder.
- The rudder can move from $-90°$ to $+90°$ at any point in time without causing any additional forces.

[3]For more information on these variables, see $https : //en.wikipedia.org/wiki/Drag_coefficient$ or $https : //www.ole.bris.ac.uk/bbcswebdav/pid{-}2627553{-}dt{-}content{-}rid{-}6894634_2/courses/AENG11101_2016/P3{-}Fluid\%20Behaviour.pdf$

- The drag force can be generalised to act at the stern of the boat as it rotates.

The equations that govern the rotational motion of a rigid body about its centre of mass are given by,

$$\underline{\dot{H}_c} = \sum \underline{M_c} = \underline{r_{ai}} \times \sum_{i=1}^{n} \underline{F_i} \tag{6}$$

$$\underline{\dot{H}_c} = \underline{\underline{I_c}}\,\underline{\epsilon}^{4} \tag{7}$$

where,
$\underline{\dot{H}_c}$ is the angular momentum of the rigid body about the centre of mass.
$\sum \underline{M_c}$ is the sum of the moments about the centre of mass, also known as the torque.
$\underline{r_{ai}}$ is the vector from the centre of mass to an arbitrary point i, where a force is acting.
$\sum_{i=1}^{n} \underline{F_i}$ is the sum of forces acting at points i on the rigid body.
$\underline{\underline{I_c}}$ is the moment of inertia tensor for a rigid body about its centre of mass.
$\underline{\epsilon}$ is the angular acceleration of the rigid body.

For our sailing boat model we can use the equations shown above by substituting in the forces F_r and D and $\ddot{y_1}$ for ϵ. We also need to introduce two new quantities, namely r_{cr} and r_{cs} which describe the lengths from the centre of mass of the sailing boat to the centre of the rudder and the stern, respectively.

Thus, our governing equation becomes:

$$\underline{\underline{I_c}}\,\underline{\epsilon} = \underline{r_{cr}} \times \underline{F_r} - \underline{r_{cs}} \times \underline{D} \tag{8}$$

$$\Rightarrow \underline{\epsilon} = \frac{1}{\underline{\underline{I_c}}} \left[\underline{r_{cr}} \times \underline{F_r} - \underline{r_{cs}} \times \underline{D} \right] \tag{9}$$

In order to simplify calculations, we will only consider the scalar version of the equation above. Substituting in the previous equations we finally get our governing equation:

$$\ddot{y_1} = \frac{1}{I_c} \left[r_{cr} K \rho_w A_r V_{rel}^2 sin(G_p(SP - y_1)) - r_{cs} \frac{1}{2} C_D \rho_w V_{rel} A_D \dot{y_1} \right] \tag{10}$$

Using ode45 in Matlab, we can solve this differential equation and plot the results for our control method.

2.6 Variables

To obtain some realistic results from our differential equation, we must find some realistic values to input. The average boat speed for a typical 30ft yacht is a cruising speed of 5-6 knots. This translates to being 2.57-3.09 m/s. [Cay17] A 30ft long boat usually has a width of about 8.5ft. [Boa] This gives the dimensions of 2.59m by 9.14m. The length to the centre of the rudder will be about half of the boat length so 9.14/2 = 4.5m. The recommended area of a rudder for a 30ft boat is about 3ft^2 which is $0.28m^2$. The value for rotational drag coefficient comes from the drag coefficient for a streamline body in water which is equal to the unit-less value of 0.04. To calculate the moment of inertia for a three dimensional sailing boat is extremely complex, but we know it will be in the form $\alpha m(l^2 + w^2)$, where α is some constant and m, l and w are the mass, length and width of the boat, respectively. We estimate the boat to have a mass of around 500 kg. Using this and the information mentioned afore we estimate the moment of inertia to be roughly around 100 kgm^2 around the normal axis, orthogonal to the plane of the water that the boat is travelling on. All the data is summarised in the following table. [5]

[4] Please not that the true equation for the derivative of angular momentum for a rigid body is $\underline{\dot{H}_c} = \underline{\underline{I_c}}\,\underline{\epsilon} + (\underline{\omega} \times (\underline{\underline{I_c}} \cdot \underline{\omega}))$, but we have decided to ignore the second term for simplicity.

[5] The information presented is extremely difficult to verify as there are so many different types of Bermuda rigged sailing boats. The main sources that were used are found under the following websites: http://u-fabboats.com/boatkits/boatkitsizes.htm, https://www.boatdesign.net/threads/speed-of-an-average-sailboat.18365/ and $https://www.theyachtmarket.com/boats_for_sale/1439229/$

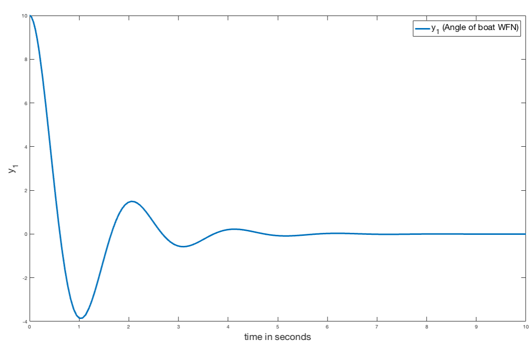
Table 1: Example data for a Bermuda rigged sailing boat

Particulars	Symbols	Values
Boat speed	V_{rel}	3 m/s
Length to centre of rudder	r_{cr}	4.6m
Length to stern	r_{cs}	4.5m
Moment of inertia	I	100 kgm^2
Set point	SP	$[0, 360]°$
Heading point	HP	$[0, 360]°$
Proportional gain	G_p	0.1
Area of rudder	A_r	0.28m^2
Rotational drag coefficient	C_D	0.04
Density of water	p_w	1000
Force on rudder proportionality constant	K	50
Drag coefficient of the sails	C_{ds}	0.004
Density of air	ρ_a	$1.225 kg/m^3$
Velocity of the wind	V_w	4m/s
Area of the sails	A_s	$34.6m^2$

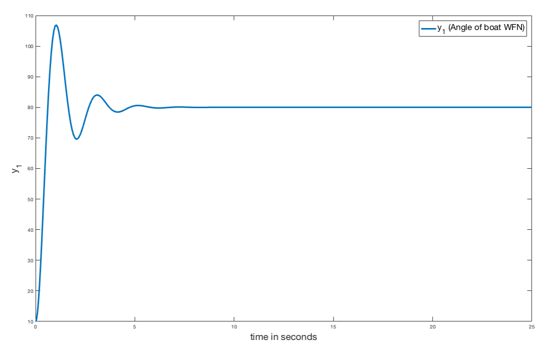
2.7 Results

Figure 4: Results for proportional control method using the data from Table 1 and changing only the heading point HP and set point SP. To see the Matlab code that generated these graphs, please see the Appendix 5.1.

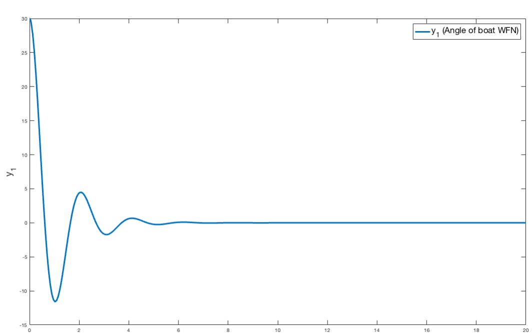
(a) Result for changing from 10 to 0 degrees west from north for t = [0,10]



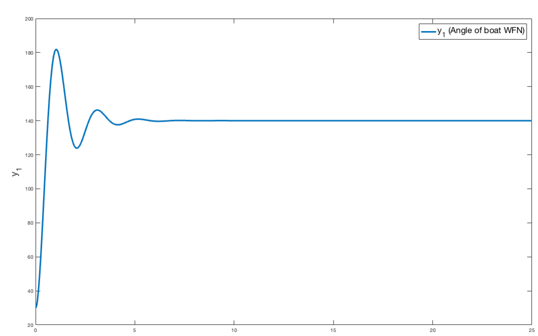
(b) Result for changing from 10 degrees to 80 degrees west from north for t = [0,25]



(c) Result for changing from 30 degrees to 0 degrees west from north for t = [0,20]



(d) Result for changing from 30 degrees to 140 degrees west from north for t = [0,25]



The results show that the sailing boat is able to change from one angle to another angle in a relatively short interval of time. For all of the changes in angle shown above, the sailing boat overshoots, but always converges to the desired angle. The amplitude of the overshoots decrease very rapidly, taking on average only four overshoots before it homes in on the desired angle. This is a very decent result and any experienced boatman will know that a boat will overshoot naturally, because there are too many variables that influence the boats motion to be able to determine precisely how it will react to be able to control its motion perfectly. Nevertheless, a better control method can be devised...

3 Improving the model

The aim of this section is to improve on the model of the previous section. This can be done by refining and possibly eradicating some of the assumptions that the previous model made to produce a more useful and realistic control method.

3.1 Adding derivative control

As we can see from the graphical results produced, our current model overshoots SP and oscillates around it before homing in on the correct angle. For our first improvement of the model we aim to largely reduce this by adding derivative control. This uses the idea of changing the angle of the rudder by not just its error from the set point (SP) but also the derivative of the error, which should help dampen the change in rudder angle and reduce the error to 0 much faster.

From our original model we defined the error of our sailing boat to be $SP - y_1$ and using this we can calculate its derivative fairly easily:

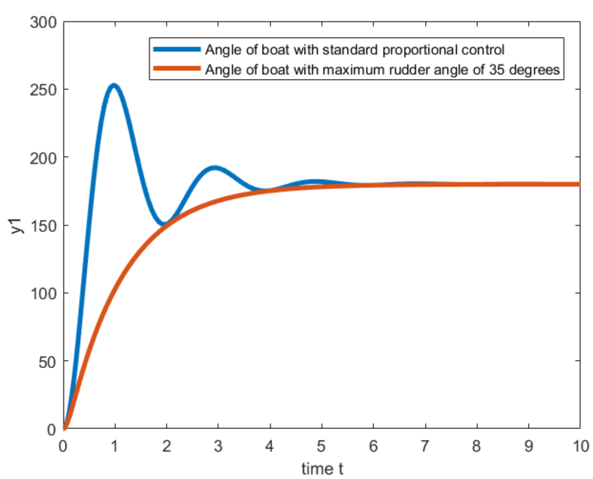
$$\frac{d(Err(t))}{dt} = \frac{d(SP - y_1)}{dt} = -\frac{dy_1}{dt} = -\dot{y_1} \tag{11}$$

This result seems logical as it simply shows that the rate of change of angle error is the negation of the angular acceleration. Adding this new derivative control term to our control equation creates the following:

$$\ddot{y_1} = \frac{1}{I_c}\left[r_{cr}K\rho_w A_r V_{rel}^2 sin(G_p(SP - y_1) - G_d\dot{y_1}) - r_{cs}\frac{1}{2}C_D\rho_w V_{rel}A_D\dot{y_1}\right] \tag{12}$$

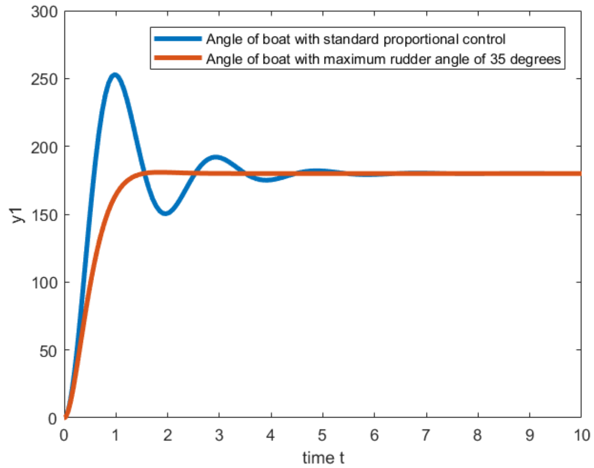
where G_d is the derivative gain, used to scale the derivative control. We then changed our MATLAB code to include this change and plotted both the sail boat with proportional control and the sail boat with additional derivative control, to see how it affected the results, starting with a G_d of 1:

Figure 5: Derivative control vs Proportional Control from 0° to 180° for $t = [0, 10]$ and with a G_d of 1



This graph shows that the oscillations have been successfully eliminated however the gradient of y_1 with derivative control is significantly less than without it and as a consequence its curve matches almost exactly with the troughs of the oscillations and therefore both curves reach the SP at the same time. This suggests that the derivative control is overcompensating and so G_d should be reduced to eliminate this. After several tests we found the most suitable G_d to be 0.3:

Figure 6: Derivative control vs Proportional Control from 0° to 180° for $t = [0, 10]$ and with a G_d of 0.3



Here it is clear that the derivative control has massively improved the speed of the controller homing in on SP in less than a 3rd of the time as the standard proportional control.

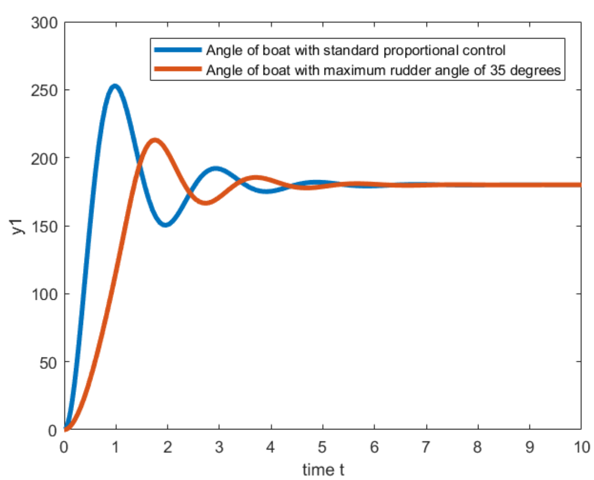
3.2 Controlling the rudder angle

To further improve the model we next looked at the movement of the rudder itself. From our research we found that it is very rare for a rudder to be able to turn more than 35° [Bar16] as this is the critical angle of the water, where its flow around the rudder will start to separate, and any angle greater than this will slow the sail boat down instead of turning it. This restraint was added to the equation by checking the sign of the intended angle change, and then taking the minimum value between the magnitude of the intended angle change and 35°:

$$\ddot{y_1} = \frac{1}{I_c}\left[r_{cr}K\rho_w A_r V_{rel}^2 sin(G_p \times sign(SP - y_1) \times min(\mid SP - y_1 \mid, 35)) - r_{cs}\frac{1}{2}C_D\rho_w V_{rel}A_D\dot{y_1}\right] \tag{13}$$

Once again this improvement was added to the matlab code and plotted against our original model to see the differences:

Figure 7: Standard Control vs Control with maximum rudder angle, from 0° to 180° for $t = [0, 10]$



As expected the gradient of y_1 with a maximum rudder angle is less than without it as the initial turning angle will be much less, however a perhaps surprising consequence is that it doesn't appear to take any longer to turn the boat to SP, this is due to the fact the size of the oscillations is largely reduced as there are much less sudden changes in the rudder angle.

3.3 Including wind drag and lift

$$D = \frac{1}{2} C_{ds} \rho_a (V_w + V_{rel})^2 A_s \tag{14}$$

where,
-C_{ds} is the drag coefficient of the sails
-ρ_a is the density of the fluid - in this case air.
-V_w is the velocity of the wind.
-A_s is the area of the sails
As before when looking at drag, we propose to substitute the angular velocity for one of the linear velocity terms.

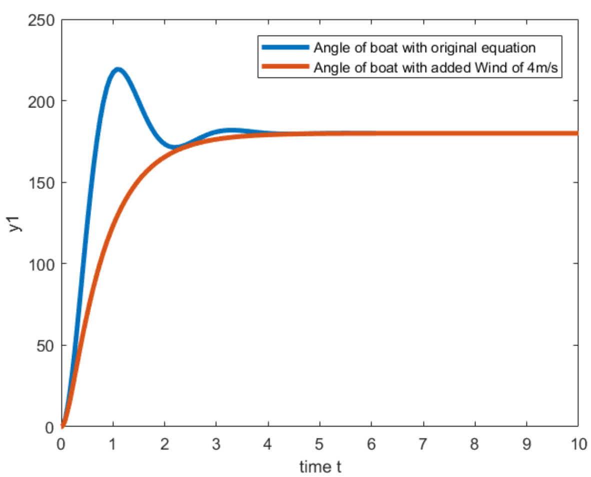
$$D = \frac{1}{2} C_{ds} \rho_a (V_w + V_{rel}) A_s \dot{y}_1 \tag{15}$$

This substitutes into the equation like so:

$$\ddot{y}_1 = \frac{1}{I_c} \left[r_{cr} K \rho_w A_r V_{rel}^2 sin(G_p(SP - y_1)) - r_{cs} \frac{1}{2} (C_{Dr} \rho_w V_{rel} A_D \dot{y}_1 - C_{Ds} \rho_a (V_w + V_{rel}) A_s \dot{y}_1) \right] \tag{16}$$

After inputting this into our matlab code, with a C_{ds} of 0.004, air density of 1.225, area of sails of $35m^2$ and standard wind speed of 4m/s, created the following graph:

Figure 8: Standard Control vs Control with Wind, from 0° to 180° for $t = [0, 10]$



Its clear from this that the added wind doesn't affect how fast the boat reaches SP as, similarly to adding derivative control, it dampens the oscillations and makes the boat turn more gradually, this is due to the added wind opposing the boat as it tries to turn.

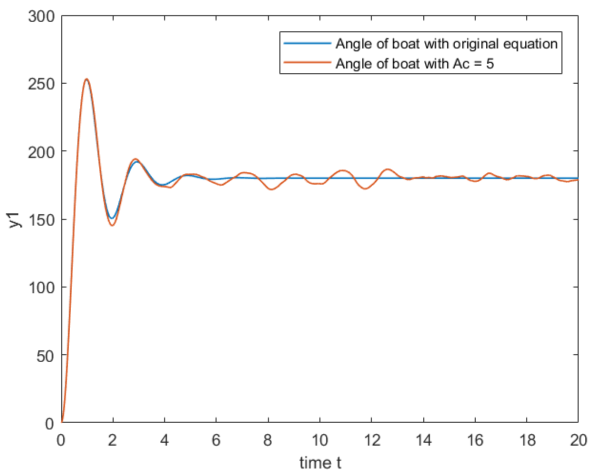
3.4 Noise

Another assumption from our first model was that the controller was 100% accurate and can set the rudder angle to exactly as desired. This is obviously impossible to do in reality, with slight variation around the desired angle always present. To simulate this in our model we used a random variable X, and decided to distribute this normally as it seems logical that the closer to the desired angle that the rudder angle really is, the more likely it is to occur. Using this new random variable X, the equation now becomes:

$$\ddot{y}_1 = \frac{1}{I_c} \left[r_{cr} K \rho_w A_r V_{rel}^2 sin(G_p(SP - y_1 + X)) - r_{cs} \frac{1}{2} C_D \rho_w V_{rel} A_D \dot{y}_1 \right] \tag{17}$$

Where $X \sim N(0, Ac)$, and Ac is the accuracy of the controller or standard deviation. The following graph plots the affects of noise added to our equation:

Figure 9: Standard Control vs Control with Accuracy of 5, from $0°$ to $180°$ for $t = [0, 20]$



The added noise means that the graph will continuously oscillate around the SP very slightly, but tending towards an average of SP as t increases and therefore the boat will be going in the direction of SP.

3.5 Latency

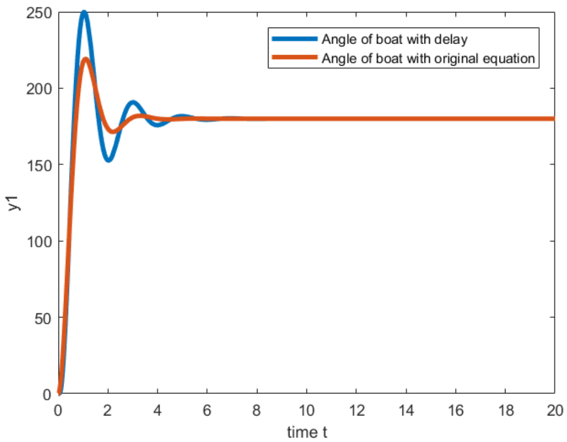
Our final and most complex addition to the model is latency, this is the idea that a controllers data inputs are not true for the instant in time at which they are used to control the system. I.E in our case the control system of a sail boat cannot know the direction of the sail boat or its angular velocity at the exact time when it calculates what angular acceleration the rudder should have to turn the boat to the SP, there will always be a slight delay. To implement this feature we changed our ODE into a delay differential equation or DDE:

$$\ddot{y_1}(t) = \frac{1}{I_c}\left[r_{cr}K\rho_w A_r V_{rel}^2 sin(G_p(SP - y_1(t-\tau)) - r_{cs}\frac{1}{2}C_D\rho_w V_{rel}A_D\dot{y_1}(t-\tau)\right] \tag{18}$$

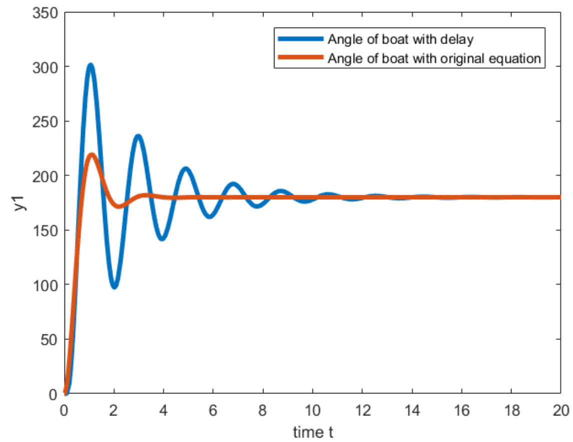
Which has a history of $y_1(t) = HP$ and $\dot{y_1}(t) = 0$ for $t \leq 0$ and where τ is the time delay between the data being true and being processed by the controller. This was more difficult to add to our Matlab code as ode45 cannot handle this kind of equation as so we instead used dde23, a function used to solve delay differential equations. With the delay added to the equation we plotted various delay values against our original equation:

Figure 10: Results for proportional control method vs delay equation from 0 to 180 degrees for t = [0, 20]

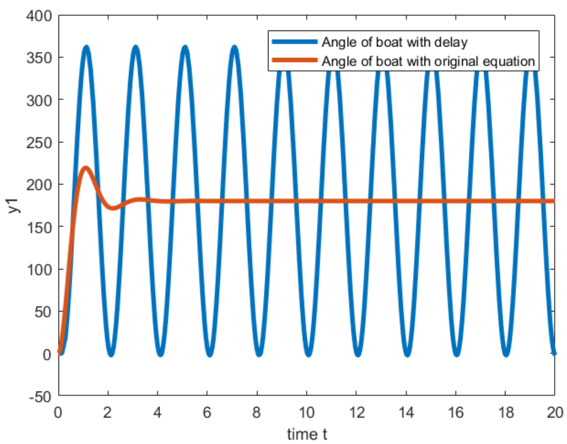
(a) Delay = 0.05



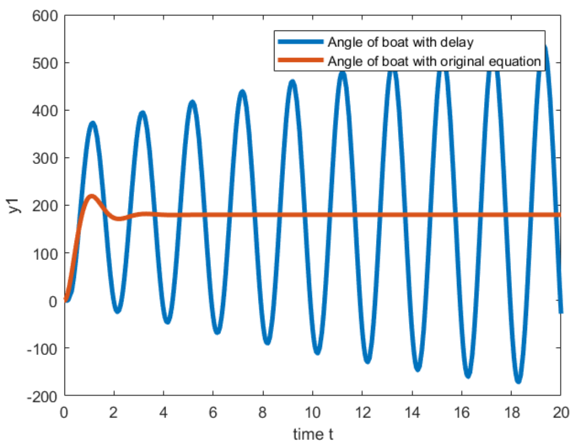
(b) Delay = 0.1



(c) Delay = 0.143



(d) Delay = 0.15



Here we can see that as the delay increases so do the sizes of oscillations, meaning that ship takes longer to reach SP until eventually for a delay of 0.143 the osculations stabilize and ship does a full 360 degrees rotation clockwise and then anti-clockwise on repeat. Any delay larger than this and the boat starts to turn more and more as the oscillations diverge, this makes sense as a controller with too much of a delay from its inputs will not be able steer the boat as its estimate for the boats angle and angular velocity will be too far from reality.

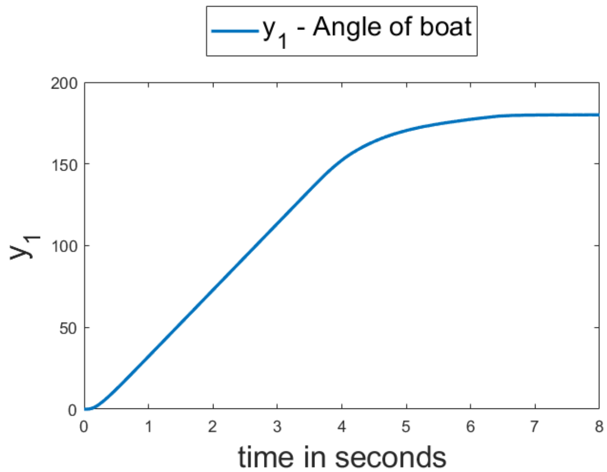
3.6 Final Model

To complete our model we combined all of the extra features laid out above and created our final control equation:

$$\begin{aligned}\ddot{y_1}(t) =& \frac{1}{I_c}\Big[r_{cr} K \rho_w A_r V_{rel}^2 sin(G_p \times sign(SP - y_1(t-\tau) - G_d \dot{y_1}(t-\tau)) \\ & \times min(|\ SP - y_1(t-\tau) - G_d \dot{y_1}(t-\tau) + X\ |, 30)) \\ & - r_{cs}\frac{1}{2}(C_{Dr}\rho_w V_{rel} A_D \dot{y_1}(t-\tau) - C_{Ds}\rho_a(V_W + Vrel)A_s \dot{y_1}(t-\tau)\Big]\end{aligned} \tag{19}$$

We created a new matlab function, see Appendix 5.2, to incorporate this equation and then plotted our final graph:

Figure 11: Final Controller against time, from 0° to 180° for $t = [0, 8]$



This is our most realistic result as adding all these extra factors has slowed down the turning of the ship and completely removed any oscillations, it has also, for the most part, linearised the turning motion of the ship which also makes sense as you would want to turn at a steady speed for maximum control and comfort.

4 Evaluation and Conclusion

This model has the potential to be used as the preliminary design of an autopilot system on ship. The performance of this direction control model in real life cannot be guaranteed however as it is limited to small ranges of velocities, masses, and delays before the equation starts to brake down. In addition, if used on a real sailing boat the model has to face more complicated situations and challenges such as changing wind speed and direction. The model still needs to have multiple improvements and more tests before it can become a product that can be used in real life.

In further studies, the integral control method could be introduced to the model as it potentially will help accelerate the process to the correct setting point. Secondly, hydrodynamics about the boat will also need to be included in future development. In the current model, the natural pitch, roll and yaw motion of the boat when it is moving in the waves, have not been considered. Pitch, roll and yaw motion for a boat are the angular motion about its transverse axis, longitudinal axis and horizontal plane separately. Those movements may affect the difficulty for a boat to turn and the boat speed. One could also expand the model to not just think about heading in a direction but instead to a specific location as most real control systems would, this would mean a changing angle set point when the boat is moved laterally.

5 Appendix

5.1 Code used for the Proportional Control Method

```
function plotsailingboatcontrol()
%
%PLOTSAILINGBOATCONTROL is a runnable function that plots the angle of
%sailing boat as the boat turns from one angle to another.
%
%USAGE Simply run plotsailingboatcontrol() in the command-line or run it
%      from the window. To change parameters of the control method, they
%      need to be changed in the code.

v = 3; %Boat speed
Lr = 4.6; %Length to centre of rudder
Ls = 4.5; %Length to centre of stern
I = 100; %Moment of inertia
```

```
Gp = 0.1; %Proportional gain
SP = 0; %Set degree (some degree west of north)
RA = 0.28; %Area of rudder (m^2)
K = 50; %Force on rudder proportionality constant
DC = 0.04; %Rotational drag coefficient
den = 1000; %Density of water

%Using ode45 to solve the second order differential equation
s = @(t,y) [y(2); den*(1/I)*(Lr*K*sind(Gp*(SP - y(1)))*RA*v^2 - 0.5*Ls*v*DC*y(2))];
[t,y] = ode45(s,[0 10], [10 0]);

%Creating the plot and legend
plot(t,y(:,1), 'Linewidth',3);
legend({'y_1 (Angle of boat WFN)'},'Fontsize', 17)
xlabel('time in seconds', 'Fontsize', 19);
ylabel('y_1','Fontsize',19);
end
```

5.2 Code for PD Control with added Noise and Latency

```
function plotsailingboatcontrol2()
%Using dde23 to solve the second order delay differential equation
sol = dde23(@dde,[0.05 0.05],[0 0]', [0 8]);

%Creating the plot and legend
plot(sol.x(1,:),sol.y(1,:), 'Linewidth',2);
legend({'y_1 - Angle of boat'},'Fontsize', 17, 'Location', 'northoutside')
xlabel('time in seconds', 'Fontsize', 19);
ylabel('y_1','Fontsize',19);
end

function f = dde(t, y, Z)
v = 5; %Boat speed
Lr = 4.6; %Length to centre of rudder
Ls = 4.5; %Length to centre of stern
I = 180; %Moment of inertia
Gp = 0.1; %Proportional gain
SP = 180; %Set degree (some degree west of north)
RA = 0.28; %Area of rudder (m^2)
K = 50; %Force on rudder proportionality constant
DC = 0.04; %Rotational drag coefficient
den = 1000; %Density of water
Ac = 5; %Accuracy of Controller
Gd = 0.3; % derivative gain
DCs = 0.004; %Drag Coefficient for sails
DenA = 1.225; %Density of Air
Vw = 4; %wind Velocity
SA = 35; % sail area

Var = 1:10000;
for i = 1:10000
    Var(i) = randn*Ac;
end

ylag = Z(:,1);
f = ones(2, 1);
```

```
f(1) = ylag(2);
f(2) = den*(1/I)*(Lr*K*sind(sign(SP - ylag(1) - Gd*ylag(2))*Gp*min(abs(SP - ylag(1)
    - Gd*ylag(2) + Var(round(5*t, 0) + 1)), 35))*RA*v^2 - 0.5*Ls*v*DC*ylag(2) -
    0.5*DCs*DenA*(v+Vw)*SA*ylag(2));
end
```

References

[Bar16] Prateek Baranwal. Rudder angle onboard ship limited to 35°. 2016.

[Boa] U-Fab Boats. Sboat kit size chart selected typical examples.

[Cay17] Crag Cay. Speed of an average sailboat. 2017.

[Jon17] Dorian Jones. Fluid flow behaviour lecture slides. pages 8–10, 2017.

[Żel14] K Żelazny. Approximate method of calculating forces on rudder during ship sailing on a shipping route. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, 8(3):459–464, 2014.