ECEN315 - Open Loop Response of a Motorised, Propeller Driven Pendulum

Joshua Benfell - 300433229

Abstract—This report covers the derivation of the open loop response for a properller driven pendulum for the purpose of understanding how such a system will respond to an applied voltage at the motor. The methods for doing such a derivation are provided as well as the simulated response of the system to various step inputs. Beyond that areas where error was introduced into the model are discussed along with the effect of changing those values. Evaulation is then done on whether the model is a suitable representation of the physical equivalent. Finally expectations of a physical system are presented along with the possibility of designing a PID controller in a future study and where a similar system might be applied.

1 Introduction

The aim of this report is to derive the theoretical open loop response for a motorised, propeller driven pendulum arm. By deriving this, we aim to understand how such a pendulum responds to an applied voltage. From this an understanding of the real system will be gained along with the shortcomings of the model representing that system.

section 2 will cover some control theory and terms to aid the understanding. section 3 is where the process to derive the transfer function of this system is coevered. section 4 will present the results of the model and discuss them and the system. section 5 will present a summary of the results as well as the follow up to deriving this model.

2 Background

Fig. 1. Block diagram for full system

An open loop system is a system where the output is not fed back into the system. The alternative is a closed loop system where the input is fed back into the system, typically subtracted from the input to get the error. The full system (fig. 1) that this paper will be modelling can be done so as an open loop system. While the overarching system is an open loop system, some of the systems that make up the blocks are themselves, closed loop systems, while others are open loop systems.

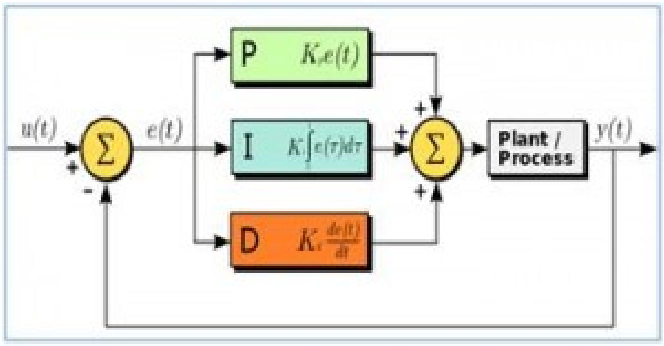


Fig. 2. Diagram of a PID Controller [1]

An example controller that could be implemented with this system to make it a closed loop system is a proportional, integral, derivative (PID) controller. This controller has three components to it all revolving around the error. In this closed loop system, the final output is subtracted from the input resulting in the error, as mentioned above. Putting this through a gain (P) controller, which is just represented as a constant value, will help make the system more responsive, however, there is a consequence of making the system more oscillatory and leaving the system with a steady state error (the difference between target and output). The I controller integrates over all errors (also multiplies by a gain constant) and aims to remove the steady state error, it also can speed up the response. This is done by lowering the gain of the controller. The

D controller, differentiates over the error and allows the controller to predict the future of the system, and react earlier making it more responsive. Summing these all together provides a PID signal which is then put into the plant instead of the original input [1]. Something like this is the eventual goal with the modelled system.

One of the evaluated characteristics of this system is it's stability. Stability of a system is defined as whether or not the output grows without bounds, if it does then the system is unstable, whereas it is stable if it settles on a value. This is regardless of any oscillation that would occur in the system. Stability can also be pre-determined by looking at the poles of the system, i.e. the values of s that make the denominator of the transfer function 0. If the real component of the poles is negative then the system is stable and vise versa if it is positive.

3 METHOD

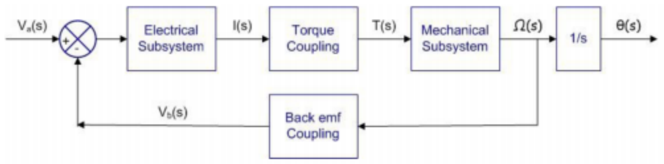


Fig. 3. Block diagram that describes the motor response to an applied voltage. [2]

The block diagram in fig. 1 illustrates an approximation of the full powered pendulum system. This system takes an input voltage into the motor causing the shaft to spin with angular velocity $\omega(t)$. Due to the propeller on the motor, this speed at which the shaft spins will generate a force which is represented by the constant K_p. The torque is then obtained by mutliplying by the distance from the pivot to the center of the motor. Finally this torque is applied to the pendulum arm, resulting in an angular displacement of the arm.

The first block to look at is the one that describes how the motor responds to an applied voltage. This

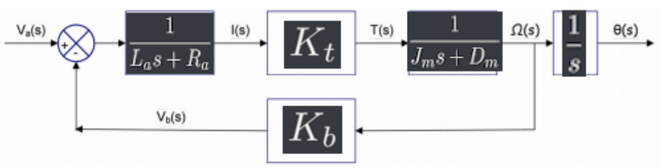


Fig. 4. Block diagram for motor and prop with equations in corresponding blocks [2]

behaviour can be broken down into smaller subsystems as shown in fig. 3.Each of those systems can be filled with the appropriate transfer function consisting of measurable properties of the motor with propeller as show in fig. 4 where:

- L_a is the inductance of the armature
- R_a is the resistance of the armature
- K_t is the torque constant that relates the current to the torque produced
- J_m is the inertia of the load (propeller)
- D_m is the damping coefficient of the load (propeller)
- K_b is the back emf constant, relating the angular velocity to an induced voltage.

The block diagram can be simplified into a generic transfer function (eq. (1)) the represents the response of the motor to an applied voltage (See Appendix C for full derivation). This transfer function was then plugged into MATLAB and evaluated with step inputs of amplitude 2, 3, 4, 5 and 6V.

$$\frac{\Omega(s)}{V(s)} = \frac{\frac{K_t}{J_m L_a}}{s^2 + \frac{J_m R_a + D_m L_a}{J_m L_a} s + \frac{R_a D_m + K_t K_b}{J_m L_a}} \quad (1)$$

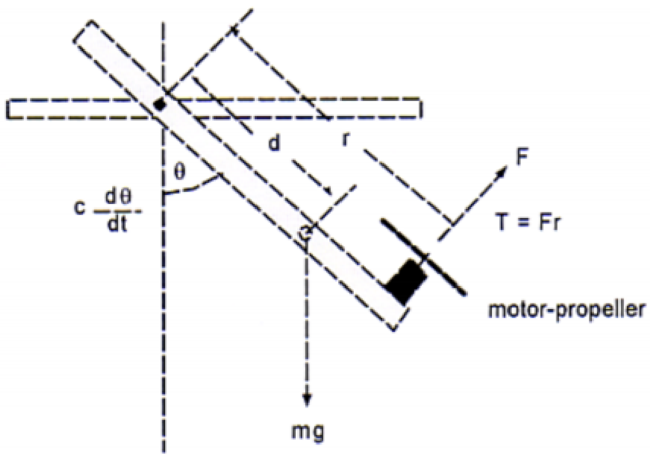


Fig. 5. Diagram of the powered pendulum

The next block to expand upon is the one that represents the pendulums response, as the two in the middle are just constants. For this block, a force is being applied (from the motor) which results in the angular displacement of the pendulum arm. An expression for net torque ($\tau_{net} = \alpha J_p$) must then be derived. Using the diagram in fig. 5, it can be seen that there are three main torques acting on the arm; The torque from the propellered motor ($\tau_m = Fr$), the torque induced by friction around the pivot

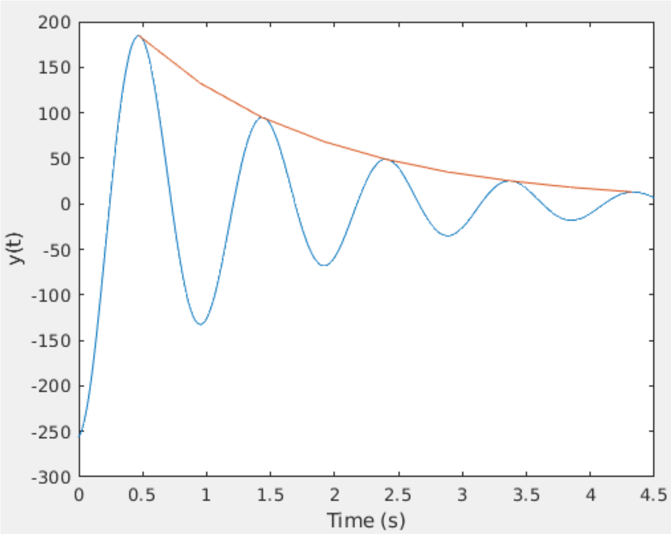


Fig. 6. Undriven penudulum data with peaks connected to resemble an exponential

when the arm is moving ($\tau_f = c\frac{\mathrm{d}\theta}{\mathrm{d}t}$) and the component of the gravity vector that is perpendicular to the arm ($\tau_g = dmg\sin\theta$).

Combining the different torques of the system into one equation allows us to find an expression for the torque produced by the motor (eq. (2)). After making the assumption that $\sin\theta = \theta$ and that α is the second derivative of angular displacement, eq. (2) takes the form of a second order homogeneous ordinary differential equation. Applying the Laplace transform to this equation allows us to rearrange it so that it resembles a transfer function with angular displacement as the output to an applied torque (eq. (3)).

$$\begin{aligned}\tau_{net} &= \tau_m - \tau_f - \tau_g\\ \tau_m &= \tau_{net} + \tau_f + \tau_g\\ \tau_m &= \alpha J_p + c\frac{\mathrm{d}\theta}{\mathrm{d}t} + dmg\sin\theta\\ \tau_m &= J_p\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + c\frac{\mathrm{d}\theta}{\mathrm{d}t} + dmg\sin\theta\\ \tau_m &= J_p\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + c\frac{\mathrm{d}\theta}{\mathrm{d}t} + dmg\theta\end{aligned} \tag{2}$$

$$\frac{\Theta(s)}{\mathrm{T(s)}} = \frac{\frac{1}{J_p}}{s^2 + \frac{c}{J_p}s + \frac{dmg}{J_p}} \tag{3}$$

Before plotting this in MATLAB for evaluation, the value for each variable need to be found. For d, m and g this is relatively easy as it is just measuring the distance to the center of mass, weighing it and using the constant value for gravity on earth of $9.81\frac{m}{s^2}$. To find J_p and c the pendulum was let to fall and the oscillation recorded. An exponential was then fit through the peaks of the plot (fig. 6) to find the values of A and B in eq. (4) using the non-linear model (nlm) function in MATLAB (Can also see Appendix D for a manual method). eq. (4) is the equation that models the motion of the pendulum where ω is given by eq. (5). When finding A and B, only the non-oscillitary component of eq. (4) is important and used. Once found, eq. (5) can be rearranged for J_P and then $c = 2BJ_p$.

$$\begin{aligned}y &= Ae^{-\frac{c}{2J_p}t}\cos(\omega t + \phi)\\ &= Ae^{-Bt}\cos(\omega t + \phi)\end{aligned} \tag{4}$$

$$\begin{aligned}\omega &= \sqrt{\frac{mgd}{J_p} - \left(\frac{c}{2J_p}\right)^2}\\ &= \sqrt{\frac{mgd}{J_p} - B^2}\end{aligned} \tag{5}$$

4 Results and Discussion

4.1 Motor Response

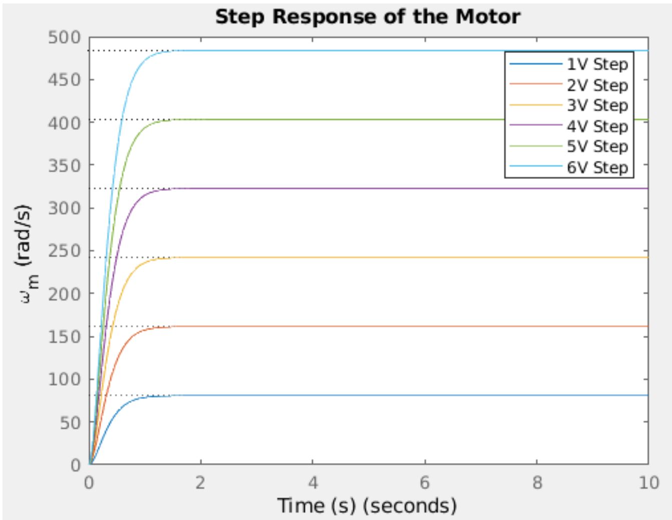


Fig. 7. Step response of the motors transfer function

TABLE 1
Table of Motor Parameters

Motor Parameters	Value
$R_a(\Omega)$	6.3
$La(H)$	0.797
$Kb(V.s/rad) = Kt(N.m/A)$	0.0043
$Dm(N.m.s/rad)$	0.00000553
$Jm(kgm2)$	0.00000241

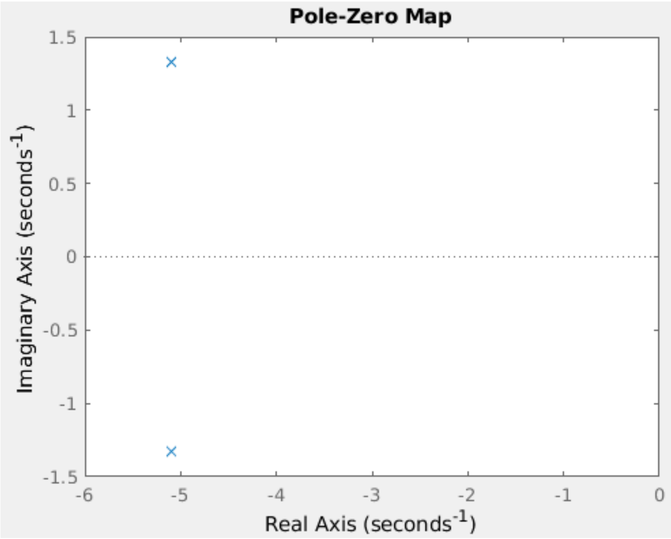


Fig. 8. PZ Plot of the motors transfer function

Step inputs ranging from 1V to 6V were applied to the transfer function of the motor (eq. (1)) using the provided values in table 1, the resulting plot is fig. 7. From this plot it can be seen that the motor winds up to a final angular velocity. The steady state gain of the motor is $80.6316[radians/Voltsecond]$ which means that each steady state value is the step inputs amplitude multiplied by the gain. The motor settles on this after $1.0374s$ with a time constant of $0.196s$ (found by $\frac{1}{\omega_n\zeta}$).

Looking at fig. 8 shows that the response of a motor with the parameters given in table 1 is complex and oscillitary. However, the step response indicates that this is a very small and insignificant oscillation that makes the system indistiguishable from one with purely real poles.

As mentioned prior, the parameter values in table 1 were provided for this experiment. They are the result of averaging prior similar experiments measurements. With this comes room for error as each motor has it's own independent properties.

Error around R_a and L_a exists because these are values that vary as the motor spins due to the connection between the coils and brushes changing as they become more and less connected. So if not enough measurements were taken with the motor shaft in different positions, the average of those points could be further off from the true value. If these values are to change then the damping and steady state gain of the system are subject to change. The steady state gain depends on R_a of the two variables currently being discussed, an increase in this causes a decrease in the gain. It also causes and increase in damping. If L_a were to increase then the damping from the resistive component of the motor will decrease.

K_b and K_t can be measured by measuring the voltage across the motor and dividing by the angular velocity of the motor [2]. This introduces errors through the measuring apparatus. If a tachometer was used then the refresh rate of the device and how reflective the propeller and background are could effect the readings. If a spectrum analyser was used then microphone quality and distance are the main factors, but it is possible to get a more accurate reading. However, it's uncertain how these values were obtained. If the measured values K_t or K_b change, then the other will two as they are equivalent [2], this changes affects the steady state gain and an increase will decrease the gain more than it increases it.

Measuring D_m encounters the same problems as K_b and K_t as shown in [2]. If this variable increases then the damping of the system also increases, as this is the damping coefficient, and the steady state gain decreases as this increase indicates a possible decrease in angular velocity.

J_m errors can arise from the measurement in D_m and however the mechanical time constant is measured. If this value would increaes then the damping would decrease due to less being less susceptible to change.

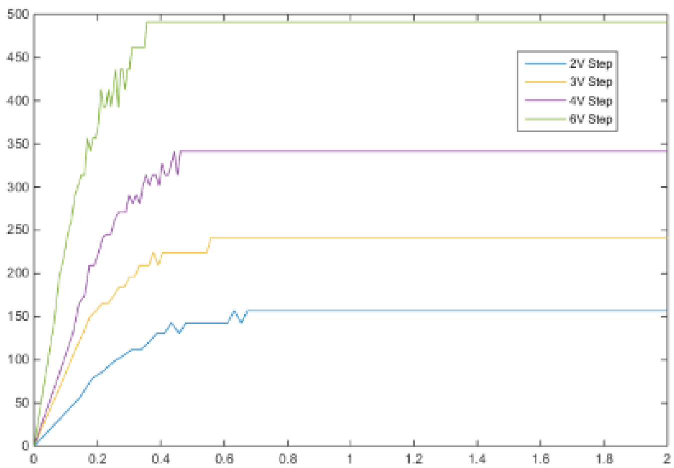


Fig. 9. Example Step response of the physical motor

The motor response from this experiment fig. 7 can be directly compared to the response from a prior experiment fig. 9. From the previous response it is shown that this experiments theoretical response matches the response of a physical motor in overall shape. The pysical response has some mechanical distortions in it that are difficult to model. The settling time appears to shorten with higher

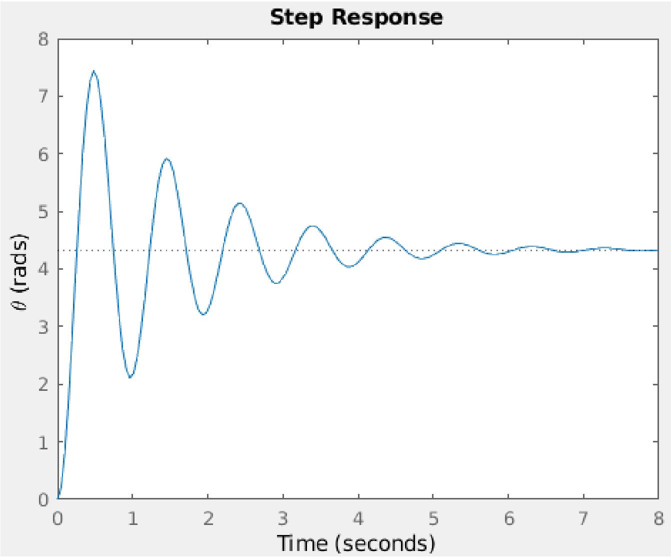


Fig. 10. Step response of the pendulum transfer function

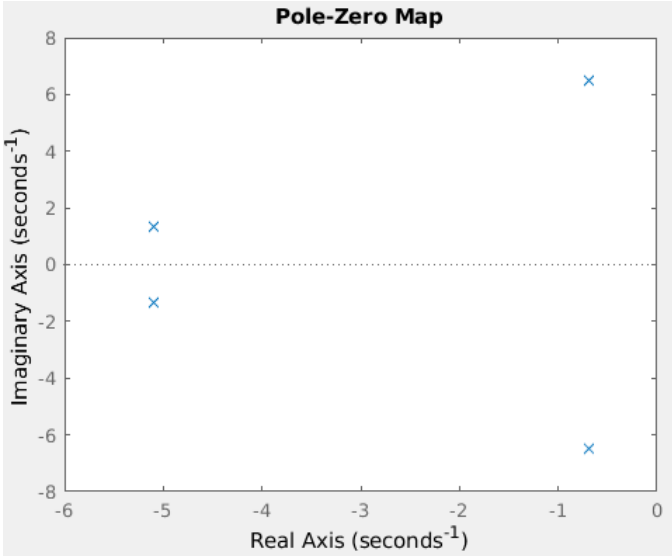


Fig. 11. Poles and Zeroes of the Final System

steps in fig. 9 which is a limitation of the model provided by this report as fig. 7 shows a constant settling time.

4.2 K_p

This coefficient affects the gain of the overall pendulum system and is linearly proportional to it. It's found by measuring the steady state force exerted by the propeller arm for a given input voltage and has units $\frac{Ns}{rad}$. Any error that could be introduced here comes from the accuracy of measurements.

4.3 r

This coefficient affects the gain of the overall pendulum system and is linearly proportional to it. It is found by using a ruler and any error is in the measurement and accuracy of the measurement.

4.4 Pendulum

Using the methods described in the pendulum section of section 3 (and Appendix D), c was found to be $0.0075\frac{kgm^2s}{rad}$ and J_p was found to be $0.0054\frac{rad}{kgm^2s}$. The step response (for an applied torque of $1Nm$) to the system is fig. 10 and indicates that adding this to the rest of the system will add a greater oscillatory component. This response is far too large for the model to realistically hold up, for the inputted torque as the pendulum does a full 360 before springing back past the origin.

The data used to model this was also supplied similar to the other aforementioned parameters. I am unable to comment on areas for error with this data as I am unsure on it's origin. It could be synthesised or measured and depending on which one indicates the level of error present. On top of that the data has been provided unitless so there is a lack of certainty on what the y axis is measuring, as the x axis can be assumed to be time. However, the effects of these variables changing can be discussed. Should c increase the amplitude of oscillations in the pendulum will decrease as the system is more damped. Should J_p increase, then the amplitude of the oscillations will increase as the system becomes less damped and more resistant to changes.

The other variables important to this transfer function are d the distance from the pivot to the center of mass, g the acceleration due to gravity which can be assumed to be the constant of $9.81\frac{m}{s^2}$ and m the mass of the arm. Error can come from the accuracy of these measurements, and g, realistically changes as the arms moves, but it is typically not enough to care about in this situation as the arm is short. However, these values are unlikely to vary greatly between similarly constructed arms, and will only do so otherwise.

4.5 Combined System

Combining all the major blocks in this system results in a transfer function that looks like eq. (7). From this transfer function and the pole-zero plot (fig. 11), the system is expected to be stable and underdamped with, from the transfer function, a steady state gain of $\frac{K_tK_pr}{dmgR_aD_m+dmgK_tK_b}$ =

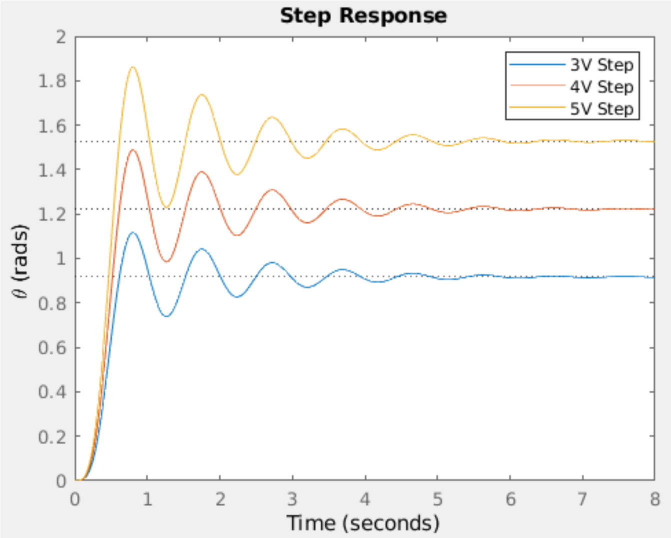


Fig. 12. Step Response of the Final System

$0.3056rad/V$. This assumption aligns with the results of the two major non-coefficient systems as well, as these are both oscillatory and stable. Using the linear system analyser in MATLAB shows that the steady state gain is indeed this value and that each system has a settling time of $4.28s$. Additionally, from fig. 12, where the system was plotted for step inputs of size $3V$, $4V$ and $5V$, the system is is visibly underdamped and stable. Each of these steps has a final value of $52.53°$, $70.04°$ and $87.55°$ respectively.

A settling time of $4.28s$ indicates that this system is going to be oscillating quite violently as that's the time it takes 4 oscillations to happen, and as input voltage is increased, the amplitude of these oscillations also does.

$$V(s) = s^4 + \frac{J_pJ_mR_a + J_pD_mL_a + J_mL_ac}{J_pJ_mL_a}s^3 + \frac{R_aD_mJ_p + K_tK_bJ_p + J_mR_ac + D_mL_ac + dmgJ_mL_a}{J_pJ_mL_a}s^2 + \frac{R_aD_mc + K_tK_bc + dmgJ_mR_a + dmgD_mL_a}{J_pJ_mL_a}s + \frac{dmgR_aD_m + dmgK_tK_b}{J_pJ_mL_a} \tag{6}$$

$$\frac{\Theta(s)}{V(s)} = \frac{\frac{K_tK_pr}{J_pJ_mL_a}}{eq.\ (6)} \tag{7}$$

4.6 Shortcomings of the model

As mentioned in section 3, an assumption that needed to be made to simplify the differential equation was that $\sin x = x$. This assumption is typically done for very small angles as it encompasses the linear section of a sinusoid and is valid for up to $20°$ as beyond that the error gets too large [3]. Because of this assumption, it is expected that a real system, the torque from gravity will have less of an impact causing the pendulum to have greater oscillations or oscillate for longer, and a higher steady state gain. This is particularly important as the settling angles of our model for the given step inputs are beyond the point where this assumption starts to break.

Another simplification made by this model is the drag on the propeller. In this model the drag is modelled as the linear relation $c\frac{\mathrm{d}\theta}{\mathrm{d}t}$, where c is the proportionality constant that is assumed to represent friction and drag, if it just represents drag then that is another shortcoming of this model. For a real system, a more accurate model for the drag is that it is proportional to V^2 or $\frac{\mathrm{d}\theta}{\mathrm{d}t}^2$ [4]. This means that a real system is expected to take longer to settle the higher the applied voltage (from resting) as the higher voltage would incite a faster initial speed. From this increased drag, the size of the oscillation would be reduced as the system is now more damped and not going as fast to cause as big of an overshoot.

From these two limitations of the model, it becomes inaccurate for the step values used in the plot fig. 12. This is because the system, responds with a higher angular displacement than the model allows for. So for small input voltages this model will hold up as the output angle will be closer to falling in an acceptible range for one of the assumptions. This also covers the other assumption around drag, as the pendulum arm won't be going fast enough to need to consider the V^2 component and leaving it as a linear relationship will be good enough.

Due to the various amounts of error sources stated in the above sections, the model is unable to accurately represent a real system. This isn't so much of an issue as it still allows a general representation of how a system, that resembles the one described in this report, will respond as the values are taken from real systems and are within the correct range. This means that real systems will have different amounts of damping, response times, overshoots, periods etc.

5 Conclusion

In this report the process to derive and model a propellor driven pendulum was presented. This model was then used and evaluated, primarily the potential sources of error and how variations in measurements would affect the system, and the accuracy of the model. The resulting model was found to be oscillatory and stable and provided a good representation of the system, for small input voltages as this would produce a small output angle. Due to the large amount of room for error with the amount of variables that can change from system to system, it is expected that a physical system will have variation in it's response, but will maintain the same overall shape of response.

Following on from this report, the next step is to model a potential control system for the system modelled in this report. The control systems aim will be to smoothly transistion the arm from one angle to another selected by the user. By making this model, the ground work will be laid out for other implementations of a PID controller for controlling angular position in pendulum type construction, such as an inverted pendulum or legs.

References

[1] T. Agarwal, "The working principle of a pid controller for beginners," https://www.elprocus.com/the-working-of-a-pid-controller/, accessed: 2018-05-19.

[2] G. Gouws, *Understanding small, permanent magnet brushed DC motors*. Victoria University of Wellington, 2008. [Online]. Available: https://ecs.wgtn.ac.nz/foswiki/pub/Courses/ECEN315_2020T1/Assignments/DCmotor_transfer_function.pdf

[3] D. Russel, "The simple pendulum," 2020. [Online]. Available: https://www.acs.psu.edu/drussell/Demos/Pendulum/Pendula.html

[4] G. Gupta and S. Abdallah, "Propeller force-constant modeling for multirotor uavs from experimental estimation of inflow velocity," *International Journal of Aerospace Engineering*, vol. 2018, p. 9632942, Apr 2018. [Online]. Available: https://doi.org/10.1155/2018/9632942

Appendix A
Matlab Code

```
%Script for ECEN315 Lab 2
clear
clf
R_a = 6.3; %Ohms
L_a = 0.797; %H
K_b = 0.0043; %Vs/rad
K_t = K_b; %Nm/A
D_m = 0.00000553; %Nms/rad
J_m = 0.00000241; %Kgm^2

numerator = K_t/(L_a*J_m);
sSqCoef = 1;
sCoef = (L_a * D_m + R_a * J_m)/(L_a * J_m);
coef3 = (R_a * D_m + K_t * K_b)/(L_a * J_m);

sys = tf(numerator , [sSqCoef sCoef coef3])
stepCoefs = [1 2 3 4 5 6];
steps = 6;

Legend = cell(steps, 1);

for i = 1:steps
    step(i * sys, 10)
    hold on
    Legend{i} = strcat(num2str(i), 'V Step');
end
hold off
ylabel("\omega_m (rad/s)");
xlabel("Time (s)");
title("Step Response of the Motor");
legend(Legend)

stepResponseInfo = stepinfo(stepCoefs.*sys)%, 'SettlingTimeThreshold',
    0.000005); %Include this setting to know approx when oscillation ends

%units of steady state gains rad/Vs
Gain = numerator/coef3
steadyStateValue = Gain * stepCoefs

SettlingTime = [stepResponseInfo(1:6).SettlingTime] %The inductance of the
    motor is likely too high which is causing the critical damping/overdamping
    and thus quick settling time.
damp(sys)
figure(2)
pzmap(sys)
```

```
% MATLAB Code for Lab 3
% Finding Coefficients from un-driven Damped pendulum
clear
clf
```

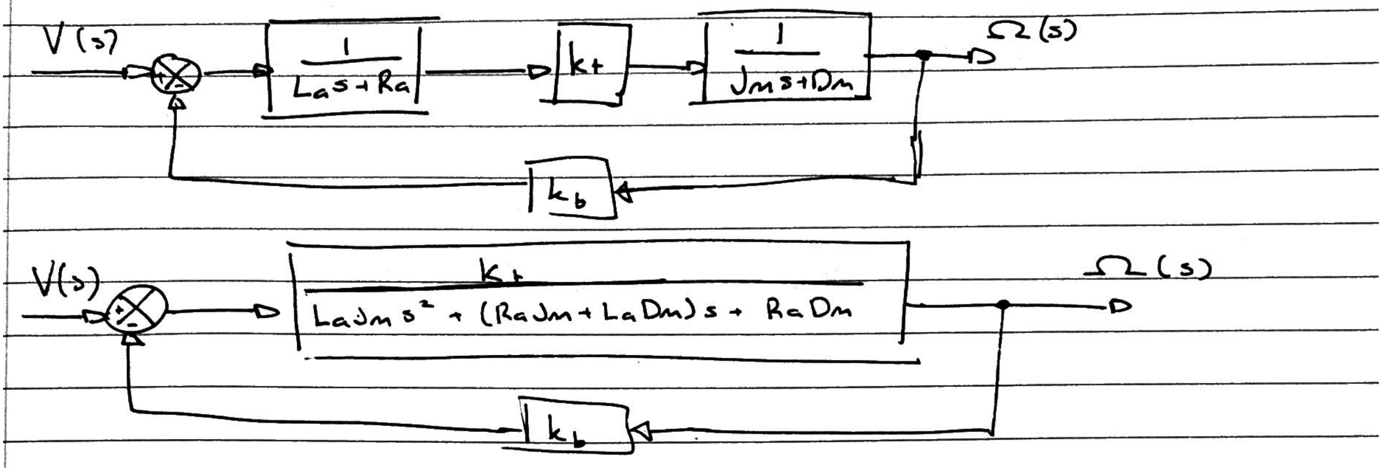
```

m = 0.168; %kg
g = 9.81;  %m/s^2
d = 0.14; %m distance from pivot to cog
r = 0.165; %m length of pendulum arm
T = 0.97; %Period in seconds
f = 1/T; %freq
w = 2*pi*f; %angular frequency

data = readtable("Data.csv"); % load csv data

figure(1)
plot(data.Var1, data.Var2); % plot csv data
xlabel("Time (s)")
ylabel("y(t)")

positiveValues = abs(data.Var2); % make all peaks positive

[peaks, locs] = findpeaks(positiveValues); % find all peaks
peaksTable = table(data.Var1(locs), peaks); % convert to table

hold on
plot(peaksTable.Var1, peaksTable.peaks); % plot line connecting all peaks
hold off

%Fitting coefficients to the function
modelfun = @(b,x) b(1)*exp(-b(2).*x(:, 1)); % function to model
beta0 = [200, 1]; % initial Guesses
model = fitnlm(peaksTable, modelfun, beta0); % the non-linear model we get out
coefs = model.Coefficients{:, 'Estimate'}; % Coefficients we are looking for


A = coefs(1); % Our A coefficient
B = coefs(2); % Our B coefficient

J_p = (m*d*g) / (power(w,2)+power(B,2)) % moment of inertia we are looking for
c = 2 * B * J_p % damping coeff we are looking for

% Transfer Function

numerator = 1/J_p;
coefB = c/J_p;
coefC = (d*m*g)/J_p;

sys = tf(numerator, [1, coefB, coefC])
figure(2)
step(sys)


% Combined Transfer function
% Lab 2 Transfer Function
R_a = 6.3; %Ohms
L_a = 0.797; %H
```

```
K_b = 0.0043; %Vs/rad
K_t = K_b; %Nm/A
D_m = 0.00000553; %Nms/rad
J_m = 0.00000241; %Kgm^2

numeratorLab2 = K_t/(L_a*J_m);
coefALab2 = 1;
coefBLab2 = (L_a * D_m + R_a * J_m)/(L_a * J_m);
coefCLab2 = (R_a * D_m + K_t * K_b)/(L_a * J_m);

sysLab2 = tf(numeratorLab2 , [coefALab2 coefBLab2 coefCLab2]);
ylabel("\theta (rads)")

K_p = 0.0053;

sys3 = sysLab2 * sys * K_p * r
Leg = cell(3, 1);
figure(3)
hold on
for i = 3:5
    step(i * sys3);
    Leg{i-2} = strcat(num2str(i), 'V Step');
end
hold off
ylabel("\theta (rads)")
legend(Leg)

linearSystemAnalyzer(sys3 .* [3 4 5])

figure(4)
pzmap(sys3)


%Algebra
algFourth = J_p * J_m * L_a;
algNum = K_t * K_p * r / algFourth;
algThird = (J_m * J_p * R_a + J_p * D_m * L_a + J_m * L_a * c) / algFourth;
algSecond = (R_a * D_m * J_p + K_t * K_b * J_p + J_m * R_a * c + D_m * L_a * c
    + d * m * g * J_m * L_a) / algFourth;
algFirst = (R_a * D_m * c + K_t * K_b * c + d * m * g * J_m * R_a + d * m * g
    * D_m * L_a) / algFourth;
algZeroth = (d * m * g * R_a * D_m + d * m * g * K_t * K_b) / algFourth;

algSys = tf(algNum, [1 algThird algSecond algFirst algZeroth])
```

Appendix B
Risk Assessment

Hazard	Cause	Probability	Severity	Mitigation
Getting fingers/jeans etc. cut by Spinning propellor	Turning the fan on	Unlikely	Low	Turn fan off when not in use. Don't stick your finger in the path of the blade.
Hot motors	Running motors too long	Unlikely	Low	Don't run the motor for too long and turn off when not in use
Tripping on cables	Walking over areas that have cables on the floor. E.g. getting fan box	Likely	Medium	Watch where you are walking if there are cables on the ground.
Electric shocks/shorts	Power supplies, exposed wires.	Likely	Medium	Be aware of exposed wires that have a potential over them. Don't turn the power supplies too high. Don't touch exposed cables if you can conduct. Don't stick a fork in mains.
Integral wind up	Leaving fan off for a while with program running, then turning on.	Likely	High	Reset the program if the fan has been off for a while. Include some limitations/safety in code.
Lego pains. Things stuck in feet	Not wearing shoes	Unlikely	Low	Wear shoes
general toxic fumes	Soldering	Likely	High	Well ventilated and don't breath in the fumes
Things flying in eyes	Fan blades snapping. Wire clippings	Likely	High	Wear googles/glasses and exercise safety squints
Capacitors	Reverse polarity electrolytics	Likely	Medium	Point away if they are about to blow. If you smell smoke, un plug
Fire	Shorts.	Unlikely	Medium	Don't short anything. If a fire is discovered, then pull the alarm/put it out with the extinguisher.

APPENDIX C
DERIVATION OF THE MOTOR TRANSFER FUNCTION



$$\frac{\Omega(s)}{V(s)} = \frac{\dfrac{kt}{LaJm s^2 + (RaJm + LaDm)s + RaDm}}{1 + \dfrac{kt\,kb}{LaJm s^2 + (RaJm + LaDm)s + RaDm}}$$

$$= \frac{kt}{LaJm s^2 + (RaJm + LaDm)s + RaDm + kt\,Kb}$$

$$= \frac{\dfrac{Kt}{LaJm}}{s^2 + \dfrac{(RaJm + LaDm)}{LaJm} s + \dfrac{RaDm + kt\,kb}{LaJm}}$$

Fig. 13. Full derivation of fig. 4 from block diagram to transfer function

APPENDIX D
MANUAL DERIVATION OF A AND B FOR AN UNPOWERED DAMPED PENDULUM

Model an exponential by drawing a line through the peaks we can ignore the sinusoidal component of the function

$y(t) = A e^{-Bt}$ ~~Ar~~

take two points

$(0.47, 184.6632)$
and
$(3.85, 18.0729)$

$A_1 \quad 184.6632 = A e^{-B(0.47)}$

$A_2 \quad 18.0729 = A e^{-B(3.85)}$

$$\frac{A_1}{A_2} = \frac{184.6632}{18.0729} = \frac{e^{-B(0.47)}}{e^{-B(3.85)}}$$

$$= e^{-B(0.47)} e^{B(3.85)}$$

$$= e^{B(\cancel{0.47}\, 3.85 - 0.47)}$$

$$10.217685 = e^{3.38B}$$

$$\ln(10.217685) = 3.38B$$

$$B = \frac{\ln(10.217685)}{3.38}$$

$$= 0.6876$$

$$= 0.688$$ agrees with NLM.

~~184.66~~

$A = 255.113$

Fig. 14. Working for finding A and B using two peaks of a damped sinusoid

Appendix E
Manual Derivation of the final systems Transfer Function

Block 1

$$\frac{\Omega_m(s)}{V_a(s)} = \frac{\frac{k_t}{J_mL_a}}{s^2 + \frac{J_mR_a + D_mL_a}{J_mL_a}s + \frac{R_aD_m + k_tk_b}{J_mL_a}}$$

Block 2

$kp = 0.0053.$

$$\frac{F(s)}{\Omega(s)} = 0.0053 = k_p$$

Block 3

$r =$

$$\frac{T(s)}{F(s)} = r =$$

Block 4

$$\frac{\Theta(s)}{T(s)} = \frac{\frac{1}{j_p}}{s^2 + \frac{c}{j_p}s + \frac{d_{mg}}{j_p}}$$

$$\frac{\Theta}{V} = \frac{\Omega}{V} \times \frac{F}{\Omega} \times \frac{T}{F} \times \frac{\Theta}{f}$$

$$= \frac{\frac{k_t}{\cancel{J_mL_a^2}}}{J_mL_a s^2 + \frac{J_mR_a + D_mL_a}{\cancel{J_mL_a}}s + \frac{R_aD_m + k_tk_b}{\cancel{J_mL_a}}} \times k_p \times r \times \frac{1}{j_ps^2 + cs + d_{mg}}$$

$$= \frac{k_t k_p r}{\begin{gathered}J_pJ_mL_a s^4 + (J_mR_a + D_mL_a)j_p s^3 + (R_aD_m + k_tk_b)j_p s^2 \\ + J_mL_a c s^3 + c(J_mR_a + RD_mL_a)s^2 + c(R_aD_m + k_tk_b)s \\ + d_{mg}J_mL_a s^2 + d_{mg}(J_mR_a + D_mL_a)s + d_{mg}(R_aD_m + k_tk_b)\end{gathered}}$$

$$= \frac{k_t k_p r}{\begin{gathered}s^4(j_pJ_mL_a) + s^3(j_pJ_mR_a + j_pD_mL_a + J_mL_a c) \\ + s^2(R_aD_mj_p + k_tk_bj_p + J_mR_ac + D_mL_ac + d_{mg}J_mL_a) \\ + s(R_aD_mc + k_tk_bc + d_{mg}J_mR_a + d_{mg}D_mL_a) \\ + (d_{mg}R_aD_m + d_{mg}k_tk_b)\end{gathered}}$$

Fig. 15. Working to find the full systems transfer function