Student Number: NJRMAR003

U N I V E R S I T Y O F C A P E T O W N
Department of Electrical Engineering

EEE3094S – Control Systems Engineering
Project 2019 Report

Mark Njoroge

14/09/2019

Declaration

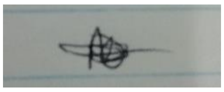
1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretending that it is one's own.

2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.

3. This report is my own work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it of as their own work or part thereof.

Name: Mark Njoroge   Student Number: NJRMAR003

Signature: 

Date: 14/09/2019

Executive Summary

This reports outlines the investigation of an unstable plant system and the design of a best fit controller. It begins with the outlining the nature of the system and deriving the requirements of the final system. Then, the system parameters are identified in order to gain an accurate model of the system.

Once the system is known, the controllers can then be designed, and the best one will be tested both via simulation and implemented in an analogue circuit. The final and best designed controller will be the one that meets all the system requirements.

Contents

1. Introduction and Background on Helicopter Control

A helicopter achieves it's lift by angling it's propeller blades against the horizontal. The steeper the angle, the more lift the helicopter receives and vice versa. This is shown in the figure below:

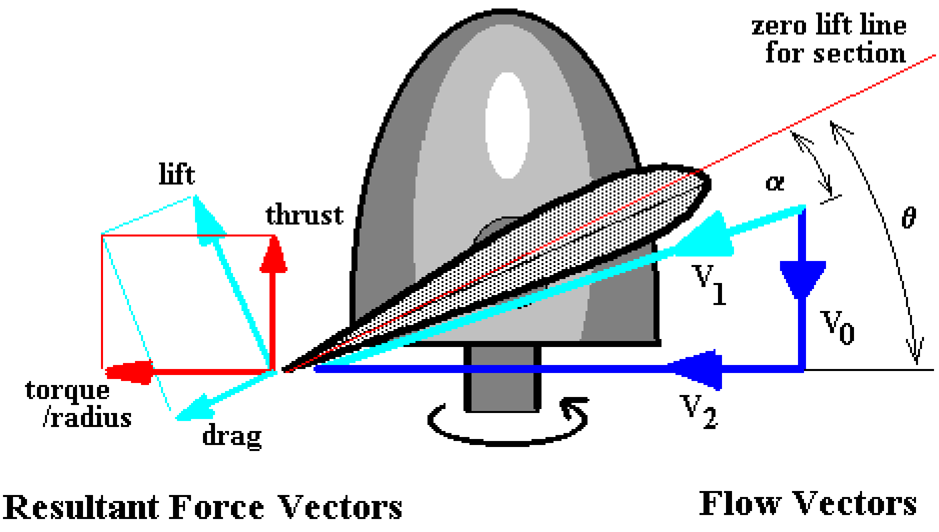


Figure 1: Helicopter Proppellor

The system for this project takes in a voltage and converts it into a required height. The propeller blades are then angled such that the helicopter gains a lift and rises to the required amplitude. However, there is no controlling action to reduce the thrust produced by the blades onces the altitude is reached, and hence maintain the required hovering height. In this case, the helicopter will keep rising indefinitely. In the opposite case, if the input altitude is below the current height of the helicopter, it will decrease its height and will continue to fall until it crashes into the ground.

This system therefore requires a feedback control loop which takes the current height of the helicopter and compares it to the input height. The controller will therefore perform the controlling action such that the helicopter reaches the required height and the angle of the propeller blades will alter accordingly to maintain the required height.

2. Technical Specifications

Analysing the given user requirements and the basic operation of the helicopter model, the technical specifications of the controller can therefore be listed as shown:

1. **Less than 20% Overshoot.**
    - This is to achieve minimal oscillatory action while the helicopter rises or falls to the required height.
2. **The closed loop response time should not be slower than ½ of the plant.**
    - That is, the rise time needs to be half of the plant.
3. **Good noise rejection**
    - Should be able to reject noise from various sources.
4. **Resistant to input and output disturbances**
    - Disturbances should have little to no effect on the final altitude.
5. **Less than 5% steady state error**

- The helicopter is desried to hover at a height directly proportional to the input voltage.
- Upon disturbances, the helicopter should revert back to within 5% original height.

With the specifications listed, the helicopter altitude is expected to closely follow the reference input and thus it will be much more effective to control it's altitude.

3. Modeling, System Identification and Problem Formulation

System Modeling

In order to model the system, a step input is passed through to the simulation software and the output position of the helicopter is observed. The purpose of this is to see the step response of the system and therefore find an appropriate transfer function. The following figure shows the block diagram to illustrate this

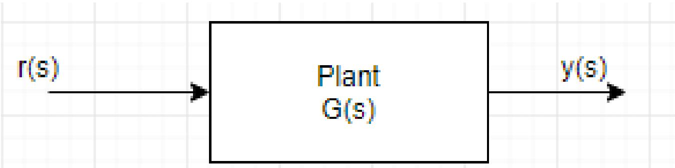


Figure 2: Block diagram of open loop

When a step is input into the system, the helicopter is seen to rise indefinitely. This is an unstable system. The transfer function to model this behavior is given by:

$$g(s) = \frac{A}{s(1+Ts)}$$

This is a second order system due to the presense of an s^2 term. This function can be separated as an integrator multiplying a first order response. Therefore, by differentiating the output of the position, we get a first order stable response which is the velocity of the helicopter.

The figures below show the graphs relating the helicopter's position and velocity when a step is used as the input:

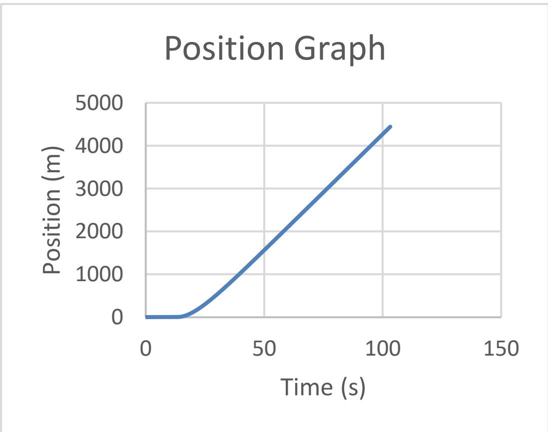


Figure 3: Helicopter Position

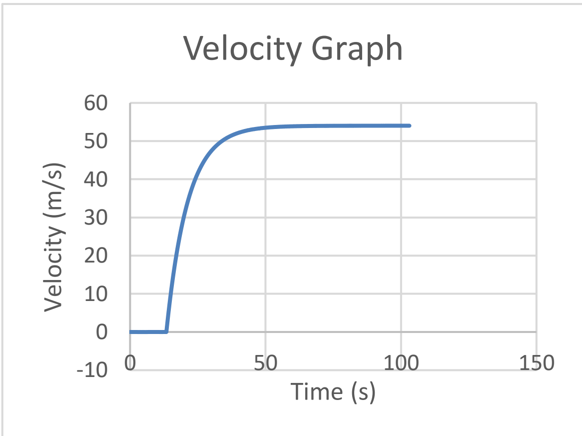


Figure 4: Helicopter speed

As is expected, the resulting velocity due to a step input is stable and models a first order system. The plant for this system is modeled as:

$$g(s) = \frac{A}{Ts+1}$$

Where A is the gain of the plant and T is the time constant.

The A and T values can be calculated directly from the graph and by extension the data acquired from the model used to plot the graph, given a known step input.

In order to accurately find the quantities of A and T, the following procedure was used:

1. The minimal voltage required to get the helicopter flying is 2.5V. However, due to inaccuracies in the circuitry and equipment, it was difficult to get precisely 2.5V and as such the helicopter hovering with a velocity of 0m/s was impossible.
2. As a result of point 1, instead of waiting for the helicopter to be completely level, a voltage just above 2.5V was input and once the velocity of the helicopter was constant, then the step was input into the system.
3. After the velocity once again became constant, the data obtained could then be used to find the gain and time constant of the system.
4. 3 sets of data were used so as to obtain accurate results.

Calculating A:

Given a step size of B, the final velocity is the product of A and B. In order to find the final velocity, the following formulae are used:

$$V_{final} = y_\infty - y_0 = AB$$

$$B = u_\infty - u_0$$

$$\therefore A = \frac{V_{final}}{B} = \frac{y_\infty - y_0}{u_\infty - u_0} = \frac{54.184 - 0.159}{5 - 2.508} = 21.681$$

Calculating T:

For a first order system, the time constant is the amount of time it takes for the output to reach 63% of it's final value. Therefore to find T, the velocity at that time is needed first. The number 63% is an approximate, so for accuracy's sake, the following formula was used to find the velocity

$$V_T = V_{final}(1 - e^{-1}) = 54.184(1 - e^{-1})$$
$$\therefore V_T = 34.150m/s$$

Remembering that the initial velocity of the helicopter was not 0m/s, the initial velocity needs to be added to this calculated value. Therefore the velocity at the required time constant is 34.309m/s.

Now, due to the fact that the data obtained is discrete and sampled every 0.1s, the time at which the required velocity is reached lies in between two time samples. In order to find the exact time at which the velocity is reached, linear interpolation is used. The exact time is therefore found to be 7.892s. This is the rise time.

Adding the integrator back to the original transfer function, the transfer function is complete and is given below:

$$g(s) = \frac{21.681}{s(1 + 7.892s)}$$

One final parameter found during system identification is the gain of the sensor in the feedback loop. This is ideally at unity gain, but again due to real life inacurracies, this is never the case. The techinique used in finding the sensor value is to divide the output of the system in volts with the output height of the helicopter. Up until saturation, the sensor value is relatively constant, and so the average of the sensor value is found.

The sensor calculation is shown:

$$h(s) = \frac{Average\ Output\ Voltage}{Average\ Output\ Height} = 0.77$$

4. Controller Design and Simulation

The transfer function of the plant can be re-written as $g(s) = \frac{2.7472}{s(s+0.1267)}$ and therefore the poles can be found at $s = 0$ and $s = -0.1267$. Because one pole lies at the origin, the system is marginally stable, which is just as bad not being stable, as seen from the output amplitude of the helicopter.

Looking at the specifications, the root locus diagram can be partitioned into a target area. The two main factors influencing this are the % Overshoot and the speed requirements. The %OS can be translated to a damping ratio zeta, which in turn is used to calculate the damping line angle in the root locus. The following formula relates the %OS to the damping ratio:

$$\%OS = 100e^{-\frac{\pi\varsigma}{\sqrt{1-\varsigma^2}}}$$

Substituting in the required overshoot, the formula can be rearranged to

$$\varsigma = \sqrt{\frac{1}{1 + (\frac{\pi}{ln5})^2}} = 0.456$$

Therefore, the required damping line angle is found below:

$$\emptyset = \cos^{-1}(0.456) = 62.87°$$

As for the speed requirements, the rise time is expected to be half of the plant. Which is:

$$\tau = \frac{7.892}{2} = 3.946s$$

The expected settling time can be found on the helicopter simulation, and for this particular system, the expected settling time is 6s. This corresponds to a speed line at $s = -0.5$

Additionally, because the transfer function is of type 1 , and it is supposed to track a POSITION input, there will always be 0 error tracking.

Figure 5 shows the poles of the plant, as well as the target area for the closed loop system:

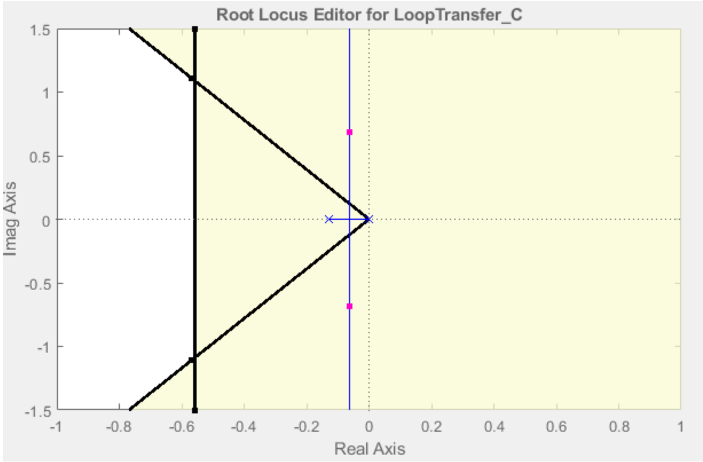


Figure 5: Root Locus plot of the plant

With the root locus plot known, it is now possible to design and make calculations for a controller. The block diagram used for the closed loop control circuit is shown below:

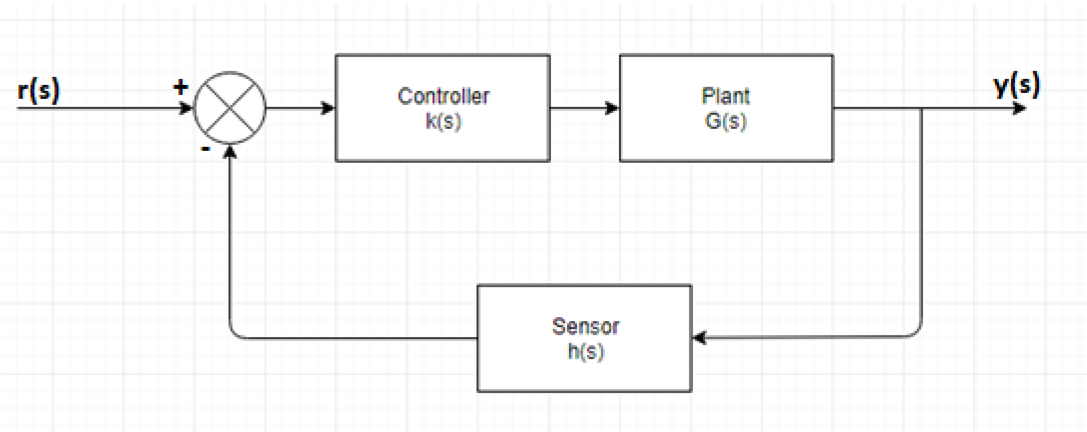


Figure 6: Closed Loop Feedback Model

The closed loop transfer function of the block diagram is

$$g_{cl}(s) = \frac{kg(s)}{1 + hkg(s)}$$

The denominator is characterized by the equation:

$$D(s) = 1 + q(s)$$

Where, $q(s) = hkg(s)$.

$q(s)$ is known as the forward path, and it is the equation we will use to calculate the value of k for the controllers, via the root locus gain. It can be re-written in the form:

$$q(s) = \gamma \frac{N_q^*(s)}{D_q^*(s)}$$

Where $N_q^*(s)$ and $D_q^*(s)$ are monic polynomials and γ is a constant.

Possible controllers:

Proportional Conroller

This controller is just simply a gain. The value of the gain will influence the location of the closed loop poles. With this type of controller and the plant we have:

$$q(s) = \frac{k(0.77)(2.7472)}{s(s + 0.1267)}$$

In order to find the value of k, we use the D(s) equation as it approaches zero, and find the value of the root locus gain at a desired point in the s plane. In this case, we will choose the mid point of the two poles of the plant, which is s=-0.6335.

$$|\gamma| = \left|\frac{D_q^*(s)}{N_q^*(s)}\right|$$

$$\therefore k(0.77)(2.7472) = \left|\frac{-0.6335(-0.6335 + 0.1267)}{1}\right|$$

$$\therefore k = 0.001605$$

Adding a Zero

As seen from the root locus diagram, the speed of the system will never be fast enough to meet the requirements. In order to achieve a greater, we add a zero to the controller to the left of the speed line. The zero will attract the root locus of the closed loop poles and thus speed up the system.
The equation for this controller is given by:

$$k(s) = k(s + 0.58)$$

And therefore,

$$q(s) = \frac{2.115k(s + 0.58)}{s(s + 0.1267)}$$

Choosing a closed loop pole at s = -1, the gain of the controller is found by:

$$2.115k = \left|\frac{-1(-1 + 0.1267)}{(-1 + 0.58)}\right|$$

$$\therefore k = 0.98$$

A problem with this controller is that it is non causal. The system would respond before the input is given. The final iteration of the controller adds a non dominant pole, in order to achieve causality. This is known as a lead controller.

Lead Compensator

The zero for this controller is chosen at $s = -0.58$ and the pole is chosen at $s = -7.42$. This makes:

$$k(s) = \frac{k(s + 0.58)}{(s + 7.42)}$$

And

$$q(s) = \frac{2.115k(s + 0.58)}{s(s + 0.1267)(s + 7.42)}$$

The gain of the controller is found at s = -1.8 and is shown below:

$$2.115k = \left| \frac{-1.8(-1.8 + 0.1267)(-1.8 + 7.42)}{(-1.8 + 0.58)} \right|$$

$$\therefore k = 7.05$$

The following graph shows the closed loop response of each variation of controller. It can be seen that the proportional controller is indeed too slow, and the final lead controller fits within the desired design specifcitations.

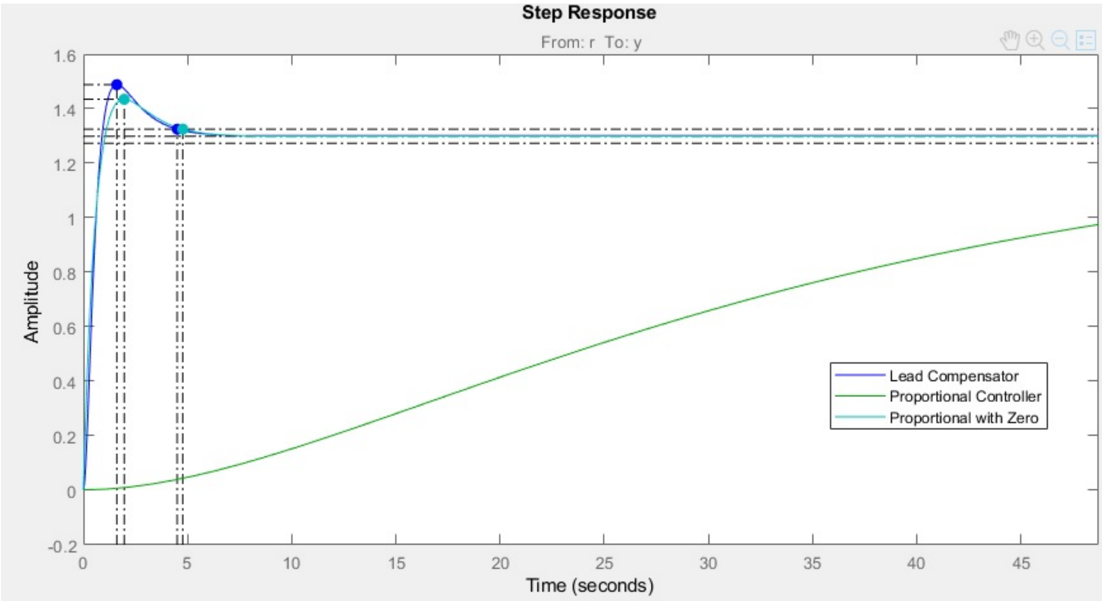


Figure 7: Step Response of each controller

The Lead compensator has an overshoot of less than 20%, and a settling time of <4s. This is in with the technical specifications outlined. Further simulations on sCad can be found in Appendix A.

5. Controller Implementation

Propotional Controller

The following crcuit shows the implementation of a proportional controller:

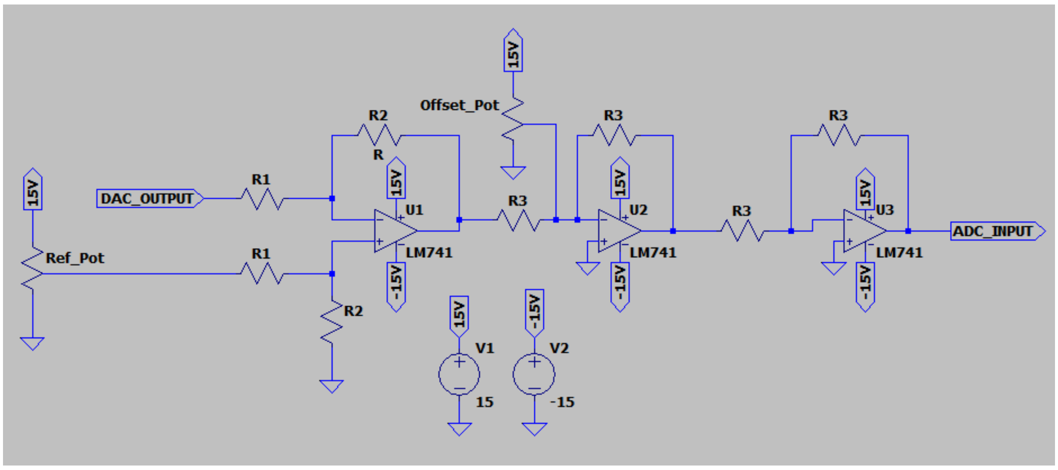


Figure 8: Proportional Controller

The first opamp acts as the summing point in the negative feedback loop. It also incorporates the gain of the proportional controller. R_1 is chosen to be $100k\Omega$ and so R_2 is calculated as follows:

$$R_2 = kR_1$$

$$\therefore\ R_2 = 0.0016\ \times 100000 = 160\Omega$$

Therefore, a series combination of 150Ω and 10Ω is used. Due to tolerances, the resistances of components would not be completely accurate, and as such the gain may vary.

The offset potentiometer is used to set the helicopter input to be 2.5V, which is the minimum voltage required to get the helicopter to fly. This voltage is added with the output from the first amplifier, and passed through an inverting amplifier with unity gain. The output of this amplifier is inversed and so another inverting amplifier with unity gain is used to invert the voltage back to the required voltage. The resistors chosen for the unity gain amplifiers are $10k\Omega$. The final output is then passed onto the DAC.

Lead Compensator

The circuit diagram for a lead compensator is shown in figure 9. This circuit is incorporated into the proportional controller circuit above, with the gain altered. It is placed after the first opamp and before the second opamp. The full circuit is found in Appendix B.

The characteristic equation for this circuit is given as:

$$k(s) = \ k\frac{1 + a\tau s}{1 + \tau s}$$

The k is the gain which is achieved by the differential amplifier at the summing point.

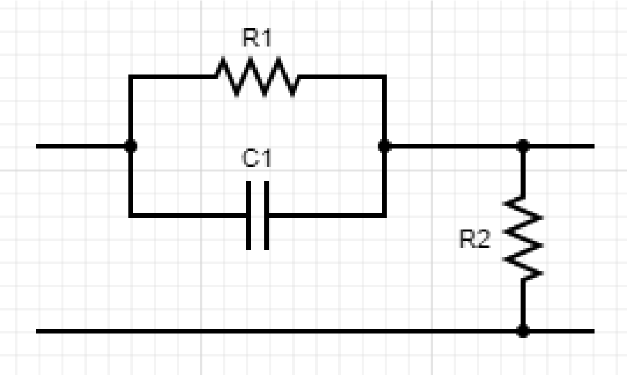


Figure 9: lead compensator

The values of a and τ are used to calculate the component values via the following formulas:

$$\tau = \frac{CR_1R_2}{R_1 + R_2} \qquad \& \qquad a = \frac{R_1 + R_2}{R_2}$$

The designed controller is re-written in a way that it resembles the characteristic equation of a lead compensator.

$$k(s) = 7.05\frac{(s + 0.577)}{s + 0.42} = 0.5455\frac{1 + 1.7346s}{1 + 0.13477s}$$

From this, we see that $\tau = 0.13477$ and $a = \frac{1.7346}{\tau} = 12.87077$

It is easier to calculate resistances from a given capacitor value, therefore the capacitor is chosen arbitrially based on the choice of capacitors available as $470\mu F$. By solving the equations simultaneously, the value of R_1 needed is found to be 3690.6Ω. Therefore a series combination of $3300\Omega, 390\Omega$ and 1Ω is used. This makes a total of 3691Ω.

As for R_2, the value calculated is 310.93Ω. Therefore a series combination of $160\Omega, 150\Omega$ and 1Ω is used.

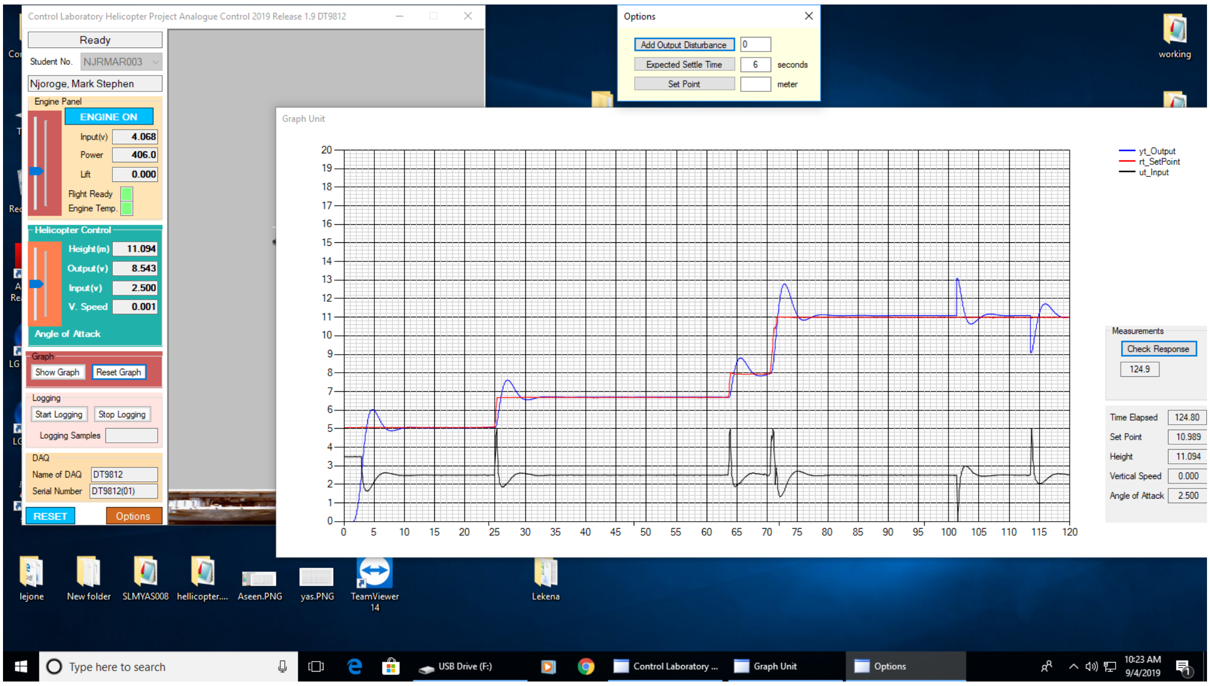
There are more components used in a lead compensator than a proportional controller. As such, there is a higher margin for error in the actual circuit than from the designed controller. It is easy to compensate for tolerances of resistors by measuring the actual resistances and adjusting accordingly.

However, for capacitors it is much more difficult, and one is forced to use the slightly differing value of capacitance from the desired value calculated from the design.

A result of component tolerances is the shifting of desired poles and zeroes in the root locus plot. This shift causes the controller to act differently from what is expected and depending on the robustness of the design, may prove to be the difference between a good controller and a bad controller.

6. System Testing

With the calculated values of resistors and capacitances, the lead compensator circuit built on the breadboard was connected to the system. The set point, controller action and flight path of the helicopter are shown in the figure below:



Figrue 10: Final Demo

As is seen, the helicopter (blue line) has virtually no steady state error tracking with the largest error being 1%. At the multiple steps input, the helicopter's settling height was always the same as the set height. The settling time of the helicopter is approximately 6s. This is in line with the expected settling time as seen in the screenshot. As seen from the first input, the overshoot is slightly above the 20% that was specified in the design requirements, the controller overshoot is 22%. This may be due to inaccuracies from component tolerances, as well as an inability to achieve the exact required resistances.

There is also a bit of oscillatory action before the steady state is achieved. This action causes the speed of the controller to reduce. This also deviates from the desired specification of no oscillations. Thankfully though there is only one dip before steady state. A more robust design would have eradicated this dip and the overshoot as well as settling time would have been well within the specificatons.

Comparing the behavior of the controller with the simulations, it is evident that the actual implementation was less precise. The settling time for the simulations was well before the expected settling time. Furthermore, the overshoot was also larger than the simulated overshoot.

When output disturbances are injected, the system reacts accordingly and reverts back to the steady state, adhering to the technical specifications desired. This means that the controller is capable of dealing with output disturbances and is robust enough. There is also good noise rejection as there is a smooth line on both the controller action and helicopter height.

7. Conclusions

The best controller designed was the lead compensator. It was achieved by adding a zero and a pole to a simple proportional control. The function of the zero was to attract the closed loop poles to the desired area of the root locus plot, and the function of the pole was to give the system causality.

The closed loop performance achieved by this design matched with the technical system requirements produced in section 2. Table 1 shows a summary of the technical specifications and the resulting controller:

Specification	*Desired*	*Result*	*Comment*
Error Tracking	<5%	<1%	Meets Requirements
Overshoot	<20%	22%	Slightly Above requirements
Settling Time	6s	6s	Meets Requirements
Output Disturbance	Good Rejection	Good Rejection	Meets Requirements
Noise Rejection	Rejects component Noise	Steady traces, noise rejected	Meets Requirements

Table 1: Summary of Performance of Controller

The closed loop design was robust enough to meet all the requirements in simulation, however the actual implementation provided an overshoot of slightly above what was required. This may be due to component intolerances.

The bulk of cost of designing this controller lies in the simulation software used in when simulating the designed controllers. The actual implementation would not be that pricey as it is mostly analogue circuit components that have been used, which have a low unit price. The time taken in designing and implementing the controller, however, was quite substantial.

Seeing as the controller managed to match all the requirements, bar the slight overshoot error, the design and implementation of the controller can be considered a success.

8. References

1. "Aerodynamics for Students" , http://www-mdp.eng.cam.ac.uk/web/library/enginfo/aerothermal_dvd_only/aero/propeller/prop1.html [Accessed on 14/09/2019]

2. "Helicopter Aerodynamics of flight", https://www.aircraftsystemstech.com/2017/06/helicopter-aerodynamics-of-flight.html [Accessed on 14/09/2019]

2. M. S. Tsoeu, Lecture Notes: EEE3094S – Control Systems Engineering, Cape Town: University of Cape Town, Electrical Engineering Department, 2019.

Appendix A

Figure i is a macro view of how a helicopter flies. The upwards thrust needs to overcome the weight of the helicopter. For hovering, the two forces need to be equal.

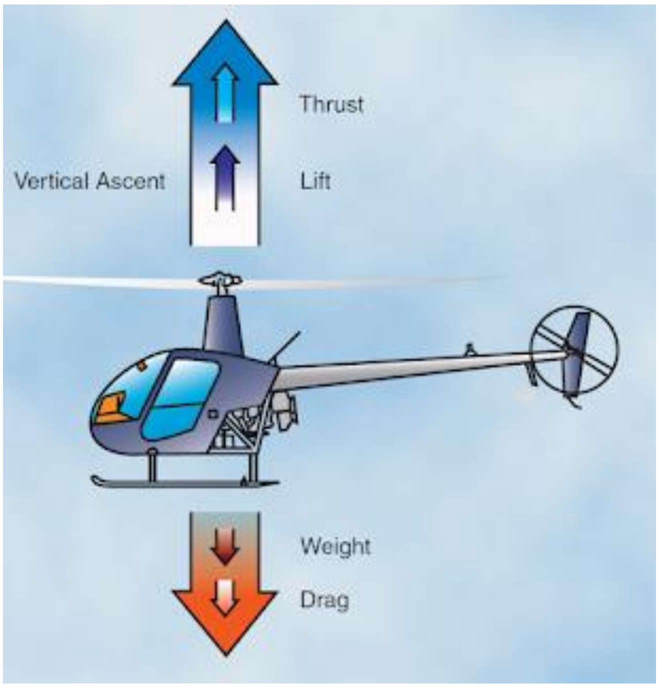


Figure i: Helicopter Flight

A table containing the relevant data in system Identification:

Initial Velocity (m/s)	Final Velocity (m/s)	Step Size (V)	A	T(s)
0.155	54.184	2.491	21.690	7.920
0.158	54.183	2.490	21.697	7.901
0.161	54.184	2.492	21.679	7.851
		Average:	21.680	7.892

Table i: System Identification

The following graphs show the simulation of the proportional and Lead compensator controllers on sCad

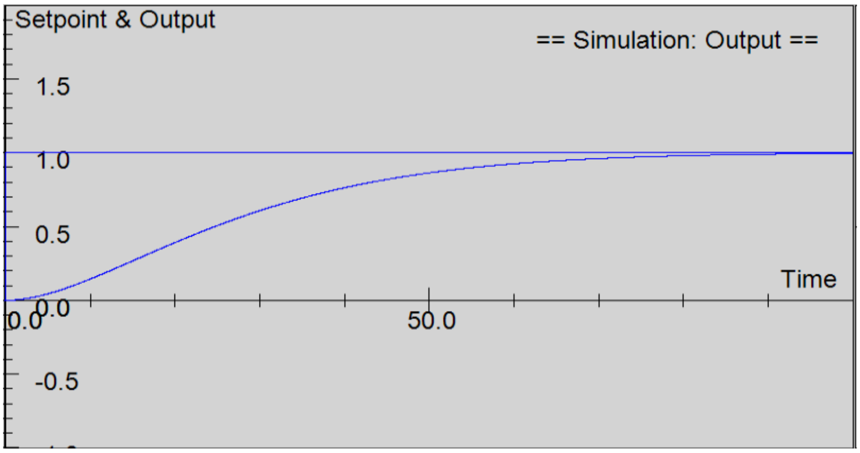


Figure ii: Proportional Simulator Modeleld on sCad

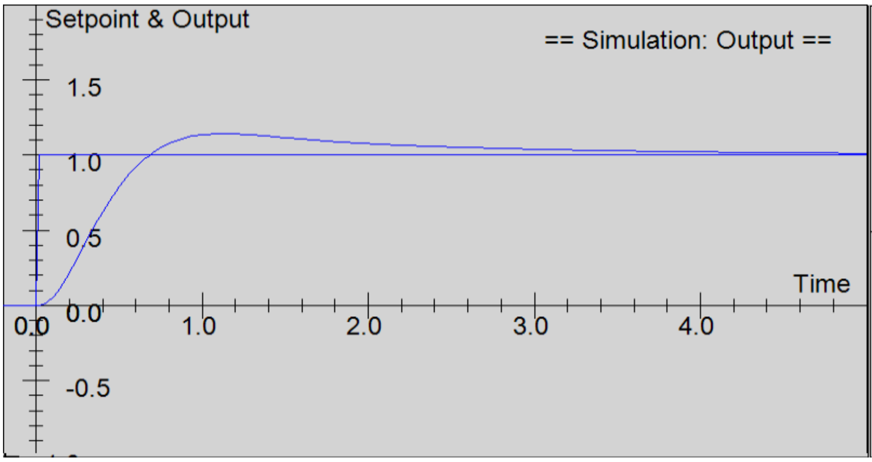


Figure iii: Lead Compensator Modeled on sCad

Appendix B

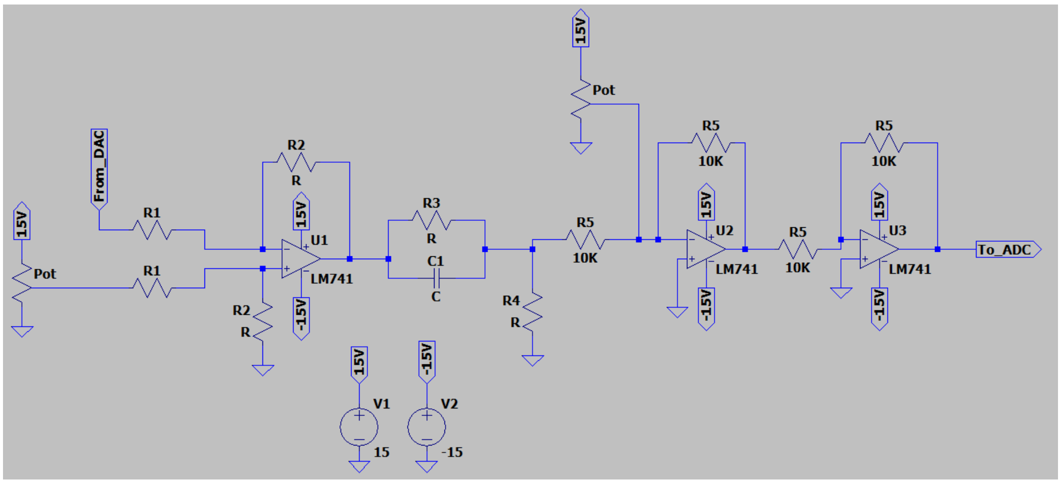


Figure iv: Circuit Diagram of Lead Compensator