Student Number: TLBCAL002

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 Engineering
Project 2019 Report

Callum Tilbury

September 2019

Executive Summary

This project entails designing, simulating and building an analogue controller for a simulated helicopter model. The model is greatly simplified and has a single degree of freedom – the angle of attack of its rotors. By controlling this parameter, the helicopter is able to move vertically. This report provides an overview to the problem, along with some performance requirements for the design. Modeling of the system is discussed, followed by the design of an appropriate controller transfer function, the formation of an reliable analogue circuit, and finally the implementation of the system and analysis of its results.

Contents

1. Introduction and Background on Helicopter Control

Harnessing flight is a complex yet brilliant feat of engineering, and the helicopter is no exception. Fundamentally, most human flight is based on the concept of an *airfoil* — shown in figure 1a alongside. As air passes over the airfoil, a lift force is generated. The magnitude of this lift is a function of the *angle of attack* — the angle which the chord-line makes with the direction of the relative wind. Whereas an airplane exploits this phenomenon by moving *forwards* (pushing its wings through the air) via thrust from its engines, a helicopter *rotates* its 'wings' (appropriately called *rotors*) which each generate lift as they move rotationally through the air. No forward movement of the fuselage (the body of the aircraft) is required, and thus a helicopter can take off vertically.

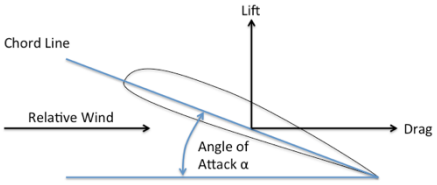


Figure 1a: Diagram of a generic airfoil [1]

Since the magnitude of the lift force generated by an airfoil is functionally dependent on the angle of attack (though not a linear relationship), one is able to control the magnitude of the upwards acceleration of the craft by actuating this angle, α. The specifics of this actuation are incredibly elegant, but are unfortunately outside of the scope of this report. Instead, the helicopter plant and actuator are considered as a single "black-box", g(s), that needs to be controlled. Moreover, since a real helicopter is multivariate and non-linear, a simple *simulation* controlling only the angle of attack was used for the entirety of this report. By adjusting the voltage on a DAC input to a computer, the angle of attack could be controlled. The control panel for this simulation is shown in figure 1b.

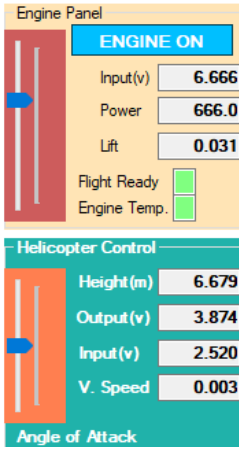


Figure 2b: Control Panel for simulation

The purpose of controlling g(s) is to ensure **safety**, **accuracy**, and **comfort** for the pilot and passengers of the helicopter. A pilot should be able to use this control system to select a desired height, and the craft should move to that height with sufficient speed and minimal oscillations. The control action should be smooth and remain efficient – thus keeping costs down, while also exhausting the least amount of fuel necessary. This is an important environmental cost considering the impact the aviation industry can have on climate change. The "cost" of this control action can be quanitified, with one example being the Integral Square Control: $ISC = \int_0^{t_f} u(t)^2 dt$. This suggests how much the controller is "working" in order to control the plant.

2. Technical Specifications

There are a handful of technical specifications to which the helicopter's closed response should adhere. These are given below, along with some motivation for each one. Note in figure 1c how the *speed* and *overshoot* requirements are depicted.

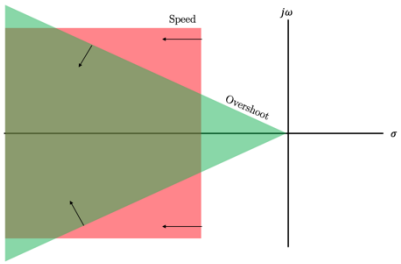


Figure 3c: S-plane Restrictions (Author's own diagram)

- In the worst case, there should be a *finite* steady-state error, and this should not be more than 5% of the setpoint value.
    - This is vitally important for the pilot to maneuver the vehicle accurately and safely. With a large steady-state error present, the pilot cannot be sure of her final position when moving from one place to another, and this severely limits her ability to do complex tasks.
- Moreover, neither the noise nor any disturbances should cause the steady-state to violate the above condition
    - Disturbances and noise are an inevitable component of any practical system, such as this helicopter, and thus rejecting them properly is fundamentally important.
- The speed of the closed loop should be at least *twice* as fast as that of the open loop (i.e. it should reach its steady state value in half the amount of time)
    - The speed of the craft is an important factor in convenience, as well as safety. Speed also enables the controller to navigate the helicopter out of a dangerous situation quickly. By choosing the closed loop parameters carefully, much better performance can be achieved, and this is, of course, preferable.
- The transient response should not have an overshoot of more than 20%
    - Overshoot is a dangerous and possibly life-threatening issue in this context. Consider an example of a pilot wanting the helicopter to hover 10 metres above the sea in order to rescue a drowning person. If the craft 'overshoots' as it is moving to its desired height, it may land in the water and many people will likely die. This emphasizes how critical this requirement is.
    - Overshoot relates directly to the damping coefficient restriction in the s-plane, by the formula: $\%\ \text{Overshoot} = 100 \exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)$. Rearranging, it can be found that, for this project, $\zeta \geq 0.46$

3. Modeling, System Identification and Problem Formulation

By examining the open-loop step-response of the helicopter, an appropriate *model* of the plant was derived. Of course, the model was not completely accurate, but it provided a valuable insight into the workings of the plant, and thus enabled the design of an appropriate controller.

After some experimentation, it was found that the helicopter had a *hover voltage* of 2.5V – the input that provided zero net lift. Using this fact, the helicopter was brought to an approximate hover, and then various step inputs were injected – voltage values above 2.5V.

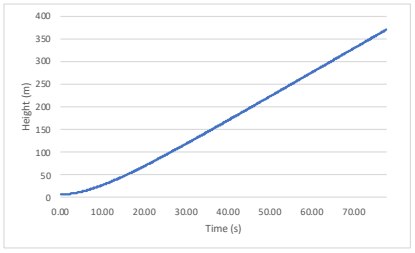


Figure 4: Output Height Step Response

Consider one of the output height step response graphs, shown in Figure 2. Two parts of the response were identified: a transient phase and steady-state phase. The latter's response – a constant gradient – suggests that the plant is marginally stable, which implies that it has a pole at $s = 0$. The former, however, indicates that this is not the *only* pole of the system. If it was, the step response would be that of an ideal integrator, $\frac{1}{s}$. Yet, the non-linear transient response reveals that the plant is *at least* a second-order system, with another pole $s < 0$ (stable). The plant was hence modelled as:

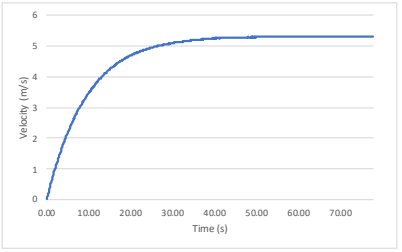


Figure 5: Output Average Velocity Step Response

$$g(s) = \frac{\mathrm{z(s)}}{\mathrm{p(s)}} = \frac{\mathrm{A} \cdot z^*(\mathrm{s})}{\mathrm{s}(\tau\,\mathrm{s} + 1)}$$

where $s = \frac{1}{\tau}$ is the second pole mentioned, z(s) are the zeros of the plant, and z*(s) is the monic polynomial equivalent of z(s). Unfortunately, determining the unknown parameters of a marginally stable system is a difficult task. Instead, the derivative of the plant was considered – $\tilde{g}(s)$. Theoretically, this entailed multiplying by s in the Laplace domain, which eliminated the marginally stable pole.

The generic response for this now *first*-order system is given below:

$$\tilde{g}(s) = \frac{\mathrm{A} \cdot z^*(\mathrm{s})}{\tau\,\mathrm{s} + 1} = \frac{\mathrm{A}(\alpha\,\mathrm{s} + 1)}{\tau\,\mathrm{s} + 1}$$

Since the plant must be *causal*, the order of $z^*(s)$ must be less than or equal to that of the denominator, and is here written as $z^*(s) = (\alpha s + 1)$.

Practically, taking the derivative of the plant response entailed calculating the average *velocity* between each sample – that is, for the n[th] sample: $v[n] = \frac{x[n]-x[n-1]}{t[n]-t[n-1]}$. This response is shown in figure 3.

Observe that this step response is a discrete approximation of $v(s) = \frac{B}{s} \cdot \tilde{g}(s)$, where B is the magnitude of the input step.

By the initial value theorem with the observation $v(0^+) = 0$, and knowing $A, B, \tau \neq 0$:

$$\lim_{t\to 0} v(t) = 0 \quad \Rightarrow \quad \lim_{s\to\infty} sv(s) = \lim_{s\to\infty} B\tilde{g}(s) = \frac{AB\alpha}{\tau} = 0 \quad \Rightarrow \quad \alpha = 0.$$

Hence,

$$v(s) = \frac{B}{s} \cdot \tilde{g}(s) = \frac{AB}{\mathrm{s}(\tau s + 1)} = \frac{AB}{s} - \frac{AB\tau}{\tau s + 1} = \frac{AB}{s} - \frac{AB}{s + \frac{1}{\tau}}$$

Considering the response in the time domain:

$$\mathcal{L}^{-1}[v(s)] = v(t) = AB\left(1 - e^{-\frac{t}{\tau}}\right)$$

This equation clearly agrees with the response shown in figure 3. Notice that $\lim_{t\to\infty} v(t) = BA$, and since B is known:

$$A = \frac{\lim_{t\to\infty} v(t)}{B}$$

Hence, the parameter A can be estimated from the output velocity step response. Multiple steps of varying magnitudes were injected into the plant, and the final modelled parameter was chosen as the average result.

Furthermore, τ can be found by rearranging $v(t)$:

$$\tau = \frac{-t}{\ln\left(1 - \frac{v(t)}{AB}\right)}$$

To minimize the effect of noise, the value of τ for a particular step was calculated for every velocity sample in the central region of the step response curve (as this is where the achieved response fit the theoretical curve best), and then averaged. Multiple tests were done, with different step sizes, and each test had an average value τ_k , and a standard deviation value, σ_k. It was noted that larger step sizes achieved smaller standard deviations (due to the higher signal-to-noise ratio of the step signals), and were thus more accurate. The final τ parameter accounted for this by taking a *weighted* averaged of the tests – weighted by the inverse of σ_k:

$$\tau = \frac{\sum_k^n \left(\tau_k \cdot \frac{1}{\sigma_k}\right)}{\sum_k^n \left(\frac{1}{\sigma_k}\right)}$$

There are now methods for finding A and τ, and these are the only unknowns in $\tilde{g}(s)$, as well as in $g(s)$. Hence, the system can be modelled.

The described process above was followed using four different step sizes, injected by the DAC output from an STM32F0 microcontroller. The resulting plant model is given below:

$$g(s) = \frac{\mathrm{A}}{\mathrm{s}(\tau\,\mathrm{s} + 1)} = \frac{14.9197}{\mathrm{s}(9.3767\mathrm{s} + 1)}$$

Finally, the helicopter simulation is required to track a given *height*, but the 'measurement' of this signal is done by a DAC – an analogue voltage. Hence, there is a transfer function for the sensor that must be accounted for. The nature of this function was found using $h[n] = \frac{Voltage\ Output\ [n]}{Height\ Output\ [n]}$. Calculating this value for every sample, it was found to be constant:

$$h(s) = 0.58$$

The block diagram of closed loop system is thus:

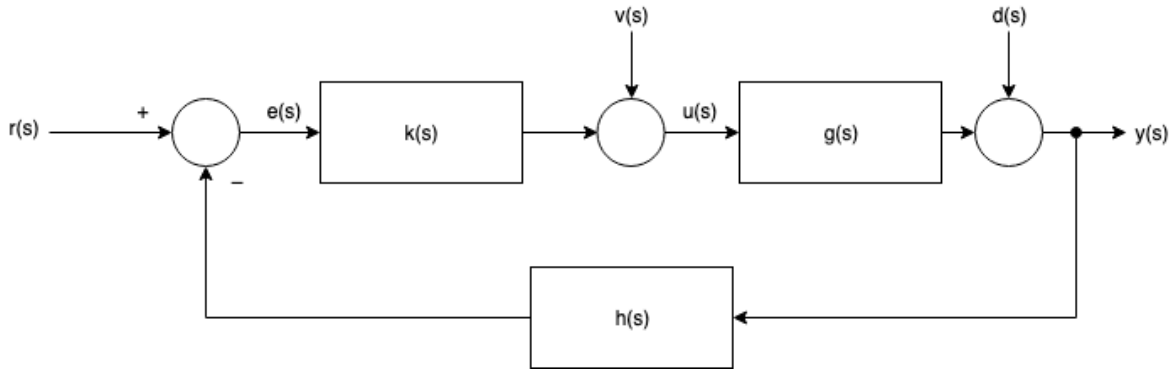


Figure 6: Block Diagram of Closed Loop (Author's own work)

where $k(s)$ is the designed controller, $v(s)$ are the input disturbances, $d(s)$ are the output disturbances, $r(s)$ is the desired output, $e(s)$ is the error between the desired output and the actual output, and $u(s)$ is the control signal.

Real-life examples of disturbances include:

- Input: Change in ground height, change in gravitational force, change in air pressure, change in weather
- Output: Change in efficiency of engine, change in friction of rotor bearings, change in swashplate calibration

4. Controller Design and Simulation

There are at least three possible controllers that could be used to control the plant, and each is summarized in the table below:

	Proportional Controller	**Lead-Compensator**	**PI-Lead-Compensator**
Form	$k(s) = K$	$k(s) = K\left(\frac{s+a}{s+b}\right)$	$k(s) = K\left(\frac{s+a}{s} \cdot \frac{s+b}{s+c}\right)$
Advantage	Extremely simple circuit – a single non-inverting amplifier would work	Good balance of simplicity and flexibility for project requirements	Highly flexible; completely rejects step input disturbances
Disadvantage	Cannot speed up the system much; very easily creates large oscillations (overshoot)	Cannot reject input disturbances with zero steady-state error	Complex circuity required for implementation

Considering the context of this project, the **Lead-Compensator** controller was chosen, where $s = a, s = b$ are the designed pole and zero combination. The controller should be configured in **output feedback** — which is the most simple and direct approach, as the achieved output is compared *directly* to the desired output.

According to the principle of Internal Model Control, in order to track a desired output $r(s)$ with zero steady-state error, $r(s)$ must appear in the forward path of the system, $q(s) = kg(s)$. Thus, since it is required for the system to track a step input, there must exist an integrator term, $\left(\frac{1}{s}\right)$, in the forward path, $q(s)$. Notice that $g(s)$ already contains an integrator term, and hence type number correction – the process of adding integrators to eliminate steady-state error – is unnecessary.

Having said that, to reject input disturbances completely (with zero steady-state error), there is a more stringent requirement: the controller would need an *additional* integrator term, $\left(\frac{1}{s}\right)$, in the forward path, $q(s)$. This is why the Proportional-Integrator-Lead-Compensator controller is advantageous over the simpler lead-lag circuit – input disturbances can be appropriately handled. Nevertheless, considering that a minor steady state error is acceptable, the lead compensator controller is good enough. The steady-state error as a function of a *step* input disturbance, $\frac{V}{s}$, for this controller is given as:

$$e_\infty = \frac{Vb}{0.58Ka}$$

It is difficult, however, to know the size of the step input disturbance, V, and hence it is difficult to design accordingly. Instead, the controller will be designed and the effect of the error will be considered empirically.

The purpose of adding the zero, $(s + a)$, is to move the slow, dominant poles further left – thus speeding up the system, and ensuring stability. In the proposed configuration, provided the gain is large enough, the closed-loop poles will not be slower than the this zero value. Recall that the controller is required to speed up the system by a factor of two or more, which means the closed loop poles must lie on the left of $s = -2 \times \frac{1}{\tau} = -0.21$. A value of $a = 1$ was hence chosen, such that all poles will be considerably faster than the requirements.

The pole, $(s + b)$, is only added to make the controller physically realizable – it should thus be non-dominant. Its specific value, however, is less strict, and was chosen as $b = 10$.

The closed loop transfer function is thus:

$$g_{CL}(s) = \frac{kg(s)}{1 + hkg(s)} = \frac{K\left(\frac{s+1}{s+10}\right) \cdot \left(\frac{14.92}{s(9.38s+1)}\right)}{1 + 0.58\,K\left(\frac{s+1}{s+10}\right) \cdot \left(\frac{14.92}{s(9.38s+1)}\right)}$$

$$\Longrightarrow g_{CL}\,(s) = \frac{14.92\,K(s+1)}{s(s+10)(9.38\,s+1) + 8.65\,K(s+1)} = \frac{14.92\,K(s+1)}{9.38s^3 + 94.8s^2 + (10 + 8.65K)s + 8.65K}$$

Of course, trying to understand the effect of varying the controller gain, K, is very challenging when viewed in the form above. Instead, a root locus plot is helpful and is given in figure 5.

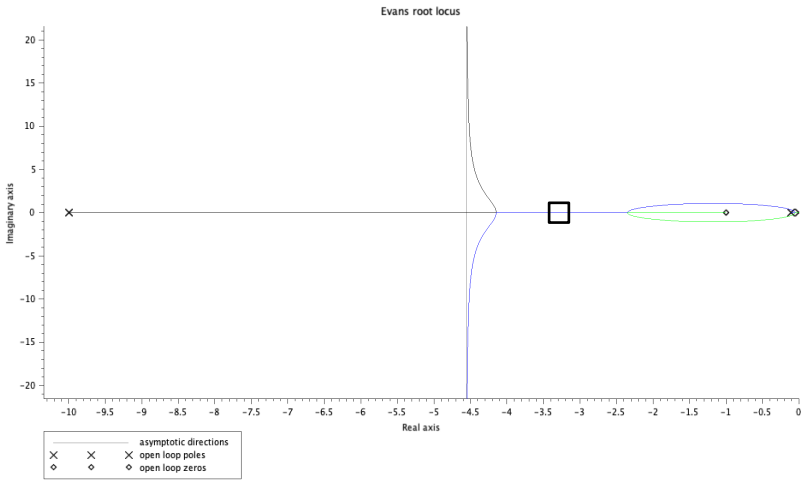


Figure 7: Root Locus Plot for Lead-Compensator Controller

There are three important K values to consider in this plot – the breakaway points. These are the values of K where $\frac{d}{ds}\gamma(s) = 0$, where $\gamma(s)$ is the root locus gain of the forward path, $kg(s)$. The breakaway points were found using SciLab, and (from right-to-left in the s-plane) correspond to gain values of:

$$K = 0.018 \quad ; \quad K = 18.78 \quad ; \quad K = 19.59$$

Since it is desirable to have low oscillatory action in this system, poles should *ideally* be chosen where the root locus path has zero imaginary component. This is clearly when the gain is $18.78 < K < 19.59$. This is a noticeably small range, but by choosing $K = 19$, even with the tolerances, the resulting pole positions should be acceptable.

Hence, the proposed controller is:

$$k(s) = 19\left(\frac{s+1}{s+10}\right)$$

A simple step response was simulated in SciLab, and is shown below in figure 6. Note that to accuarately emulate an op-amp based controller, saturation limits were added to the controller at $\pm$15V.

The desired output was stepped up to +10 at $t = 1$, and it is clear that the controller smoothly rises to its final value within 4 seconds (which is 5 time constants). The open loop time constant was $\tau = 9.38$ seconds, and hence the closed loop is much faster than the requirements. Furthermore, there is no overshoot/oscillatory action, which is a good thing. At $t = 10$, an output step was added, and is quickly rejected within the appropriate amount of time. However, at $t = 20$, an input disturbance is added, and the controller cannot handle this – as discussed previously. A steady-state error exists, and this may or may not contravene the project requirements – depending on the magnitude of the disturbance. Since this magnitude is not known, empirical testing is required to ensure the design criteria are still met.

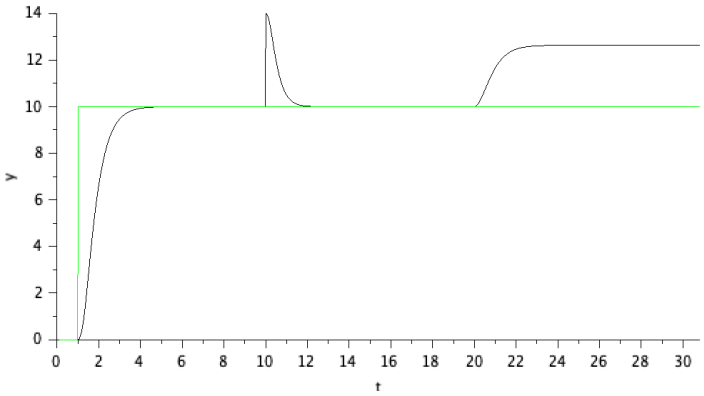
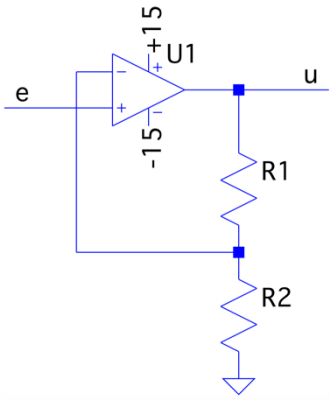
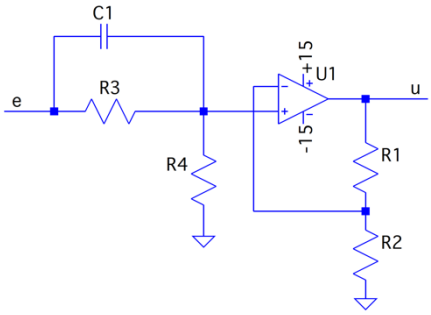


Figure 8: Response to various step inputs

5. Controller Implementation

Two op-amp controller implementations are given in the table below:

	Proportional Controller	**Lead-Compensator Circuit**
Circuit Diagram		
Derivation	This circuit is a simple non-inverting amplifier. The transfer function is known to be: $$k(s) = 1 + \frac{R_1}{R_2}$$ For example, to achieve a gain of 19, one can select $R_1 = 18k\Omega$, $R_2 = 1k\Omega$.	The first half of this circuit is a simple impedance divider, with $$k_1(s) = \frac{z_2(s)}{z_1(s) + z_2(s)}$$ where $z_1 = R_3 \| \| C_1$ and $z_2 = R_4$. $$R_3 \| \| C_1 = \frac{R_3 \cdot \frac{1}{sC_1}}{R_3 + \frac{1}{sC_1}} = \frac{R_3}{sR_3C_1 + 1}$$ Therefore, $k_1(s) = \frac{R_4}{\frac{R_3}{sR_3C_1+1} + R_4} = \cdots$ $$\ldots = \frac{s + \frac{1}{R_3C_1}}{s + \frac{1}{R_4C_1}}$$ The second half of the circuit is a non-inverting amplifier, which just applies a gain. The overall transfer function is: $$k(s) = \frac{s + \frac{1}{R_3C_1}}{s + \frac{1}{R_4C_1}} \left(1 + \frac{R_1}{R_2}\right)$$
Effect of Tolerances	Consider if instead of their nominal values, the resistors had actual resistances at the upper/lower ends of their tolerances. For 5% resistors, for example: $$\tilde{k}(s) = 1 + \frac{105\% \cdot R_1}{95\% \cdot R_2} = 1 + 1.11\frac{R_1}{R_2}$$ Using the numerical example above, with a nominal gain of 19, the actual gain would be 20.98. This new controller gain value would cause higher oscillations in the proportional controller closed loop.	Consider again if the components had actual values at the extremes of their tolerances. Capacitors often have a tolerance of 20%, and this really affects the result. For example: $$k(s) = \frac{s + \frac{1}{95\%R_3 \cdot 80\%C_1}}{s + \frac{1}{95\%R_4 \cdot 80\%\, C_1}} (\ldots)$$ This means the desired pole and zero of the controller could be almost 25% larger or smaller than their designed values. Fortunately, in this root locus design, there is a large amount of room for error and tolerances. However, if tolerances end up negatively affecting the physical implemetation, component values would have to be rethought.

6. System Testing

The final implemented system circuit is shown in the figure below:

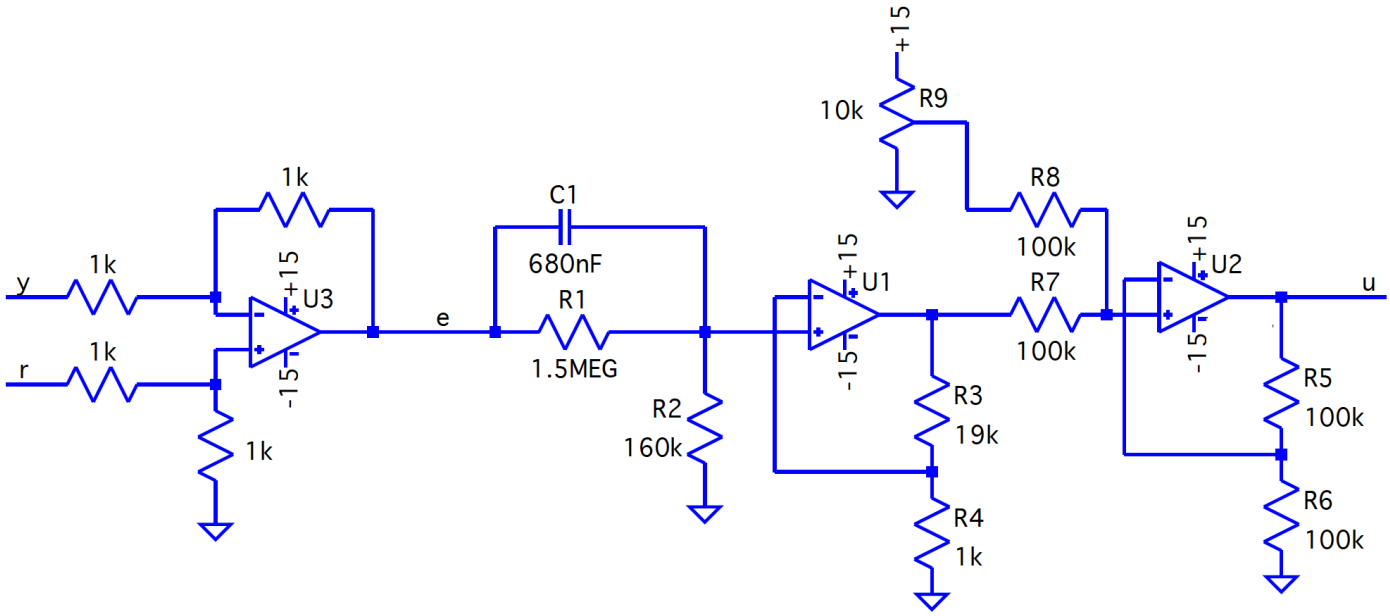


Figure 9: Full Circuit Implementation

Note that in order to minimize the steady-state error due to the input disturbance, an offset voltage was added to $u(s)$ when it was fed into the plant – this explains the inclusion of a non-invering summing amplifier, U2. Ideally, instead, the type number of the controller circuit should be increased (as in the PI lead-compensator), but there is a limit on the number of op-amps allowed, and hence this "calibration" approach is necessary.

As for the interaction with the plant – which is actually a computer simutlation, y(t) is the output of the computer's DAC – the actual height of the craft in volts; r(t) is the desired output height, also in volts. The control signal, u(t), is the input into the computer's ADC which controls the angle of attack of the helicopter rotors.

It is worth noting that the resistor values used in the passive lead compensator (R_1& R_2) are fairly large, particularly $R_1 = 1.5M\Omega$. This is usually discouraged, due to the deteroiating noise performance at the extremely large and small resistance values. However, since the capacitor, C_1, is exposed to alternating polarities – depending on the action required by the controller – it cannot be electrolytic. This severely limits the capacitance available for use, and thus pushes the resistor values to be quite large, and thus possibly very inaccurate.

Interestingly enough, however, it was the capacitor that was most inaccurate. When set-up in a simple RC charging circuit, the expected time constant (and thus capacitance value) was *half* of its nominal value. This drastically changed the position of the poles and zeros of the controller, and thus completely changed the dynamics of the system.

A real-life achieved response is shown in figure 8, below. The red line is r(t) – the desired output height; the blue line is y(t) – the actual output height; the black line is u(t) – the control signal.

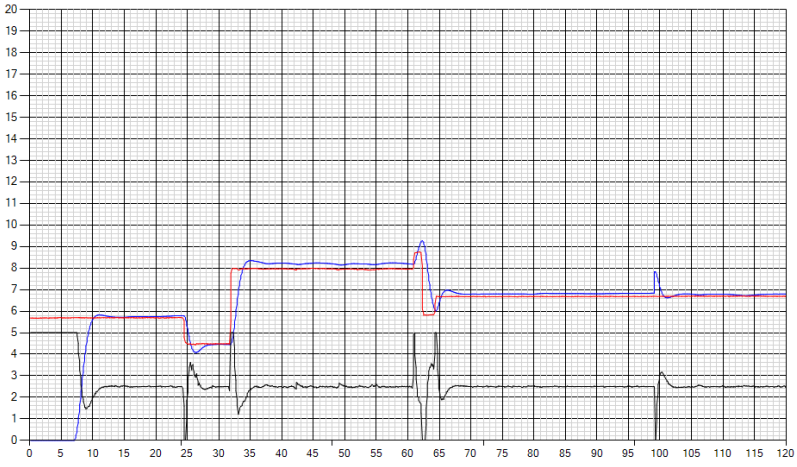


Figure 10: Achieved response with analogue controller

The speed of the system, as was the case in simulation, is clearly very good. The output of the helicopter reaches the desired output around 5 seconds after the step input is injected into r(t). This holds true for both big and small steps, upwards and downwards. This satisfies the design requirement of the closed loop needing to be at least twice as fast as the plant connected in open loop.

The controller does not exhibit bang-bang characteristics – merely switching between

saturating and being completely off, and this is a good thing. The control signal, u(t), remains fairly smooth throughout, and is thus acting efficiently – which is an important 'cost' consideration.

Overshoot is minimal, and easily falls within the 20% criteria.

Disturbance rejection is a tricky one. The output disturbance is easily rejected, as seen in the final 'blip' of the response in figure 8. However, the input disturbance, as mentioned frequently prior to this, results in a steady-state error. What is fascinating is that this error is *not* constant over the range of the helicopter height values. A calibration can be set using the variable offset (non-inverting summing amplifier, U2). However, as soon as the helicopter moves further up or down, the error again exists, and worsens the further one moves from the calibrated point. This evidence suggests that there is some sort of height influence on the magnitude of the input disturbance. And yet, this was not evident in the early-stages of testing the plant in open-loop. More research would have to be done into modelling the input disturbance such that it can be appropriately rejected through design.

The actual output results when compared to both the theoretical and simulated output results are well matched. The speed of the system is accurately achieved, and the observed disturbance rejection (lack thereof) – though not ideal – can be explained mathematically, and is also seen in simulation. This coherence between the three fields is a positive sign.

7. Conclusions

Using a **lead-compensator** circuit in **an output feedback** configuration is a good way of meeting the controller requirements. It proved to be a good blend of simplicity (as more sophisticated controllers became terribly complex in circuitry) and performance (it still performed very well in speed-up and in overshoot requirments).

Choosing the poles and zeros for this controller using the **root locus** method is a wonderfully visual approach, and gives great insight into the nature of the system as a whole. By looking at the loci for this controller, one can see that there is an *infinite gain margin* – for the closed loop can never became unstable (with a positive controller gain). This is an encouraging characteristic and it gives the users some peace of mind.

Notice that even with fairly large variations in the system model parameters, $g(s)$, the closed loop should still perform fairly well. The loci will likely follow similar paths as before, and the system requirements will likely still be met. This means the closed loop is *robust*, as uncertainties in the system model will not affect the achieved results.

Finally, as seen in the empirical results, the control signal, $u(t)$, remained smooth and efficient for the duration of the flight, and this implies that the cost of controlling the helicopter is acceptbale. At the very least, the cost is the same of controlling the helicopter manually.

8. References

[1]"Angle of Attack", AviationChief.Com, 2019. [Online]. Available: http://www.aviationchief.com/angle-of-attack.html. [Accessed: 15- Sep- 2019].