

Lab 2 Report

KittiCopter Control System

Kian Frassek

EEE3094S Control Systems Engineering

FRSKIA001

Mechatronics: Department of Electrical Engineering

September 21, 2023

Contents

[1. Introduction 3](#_Toc147308827)

[2. System Modelling 3](#_Toc147308828)

[1. Methodology 3](#_Toc147308829)

[1. Modelled System and Results 3](#_Toc147308830)

[2. A free-body diagram illustrating the forces acting on the Kitti copter. 3](#_Toc147308831)

[3. A block diagram of the whole system showing the various signals. 3](#_Toc147308832)

[2. A differential equation describing the motion of the Kitti copter. 3](#_Toc147308833)

[1. A free-body diagram illustrating the forces acting on the kitticopter 3](#_Toc147308834)

[2. A block diagram of the whole system showing the various signals 3](#_Toc147308835)

[3. A differential equation describing the motion of the kitticopter 3](#_Toc147308836)

[4. A transfer function derived from the differential equation. 3](#_Toc147308837)

[3. System Identification 3](#_Toc147308838)

[1. Step Response Test 3](#_Toc147308839)

[1. Figures and Tables 3](#_Toc147308840)

[2. Model Validation 3](#_Toc147308841)

[4. Controller Design 3](#_Toc147308842)

[1. Tests performed on the Controller 3](#_Toc147308843)

[2. Performance Evaluation 3](#_Toc147308844)

[3. Recommendations for Future Iterations 3](#_Toc147308845)

[5. Controller Testing 3](#_Toc147308846)

[6. Discussion and Conclusion 3](#_Toc147308847)

[7. Appendix and Notes 3](#_Toc147308848)

[8. References 3](#_Toc147308849)

# Introduction

Sadly, a cat named Orville passed away in 2012. Controversially, the cat owner decided to convert him into a kitty copter. The report contains information about control system so that Orville can ascend into the sky.

This report consists of multiple sections. Firstly, it evaluates the system using simulation software to identify the characteristic equation of the plant (kitty copter).

Next, a closed-loop proportional controller is designed and built to allow the kitty copter to track the following specifications:

* Tracking of position inputs with >90% accuracy (i.e., the tracking error and effects of disturbances must be <10%).
* Settling time improvement of at least 20% compared to the open-loop system.
* Overshoot of less than 5%.
* Robustness to uncertainty of up to 10% in the aerodynamic constant of the system. (Since you are determining this parameter experimentally, you should make sure your design works even if you don’t get it precisely correct.)
* Robustness to a tolerance of 10% in the components used to assemble the controller.
* BONUS [up to an extra 20%]: Tracking of velocity inputs with >80% accuracy ( (amathoba, 2023)

A diagram of a program

Description automatically generated

(amathoba, 2023)

This Is just an example block diagram of the more detailed one can be seen below.

In this block diagram it can be see that there are signals and processing block. below lies a list of what each of these items do:

Reference: This is a altitude measurement that is used to indicate the height value in meters the Kitty copter must go to.

F(s) pre-filter: This is a scalar value that we will call C for this project that is used to convert meters to volts.

Subtraction circle: used to create closed loop feedback.

Proportional controller G(s): used to multiply by a value or gain that is going to be label K in this project.

Input disturbance summation: used to inflate the disturbance due to gravity this will be represented as an offset.

P(s) plant: Used to describe the Kitty copters fundamental characteristics.

Y(s) output: the current altitude of the helicopter.

Sensor noise: used to input the noise of the sensor this will most likely be ignored in this project due to its minor effect.

H(s) sensor: K value that is used to convert meters to volts.

# System Modelling

In this section we will be trying to identify the exact characteristic equation of the plant as it reacts to a voltage input. This plant is a second order system, but it can be broken down into a first order system times by an integrator. The characteristic equation for a first order system in the LaPlace domain is or .

Where

* A is the gain
* b is the damping coefficient
* S is the complex frequency variable.

The integrator can be represented with the following expression in the laplace domain .

Therefore, the full plant can be represented by the equation

## Methodology

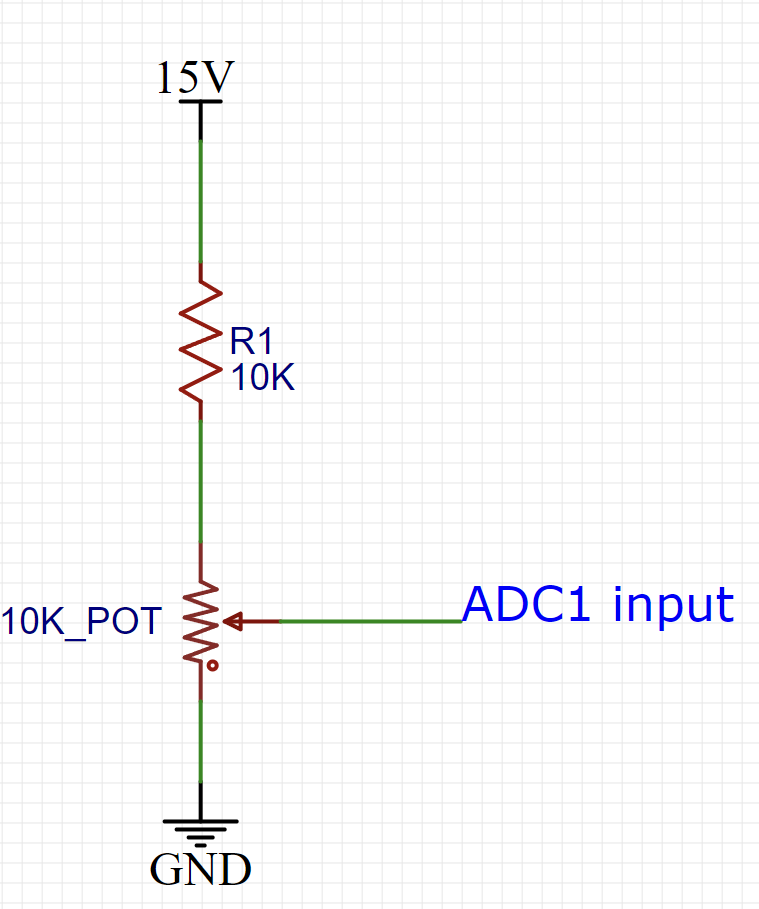
* Using design techniques to better understand the system. Start
  + Draw a free body diagram.
  + Derive the differential equation for the plant.
  + Find the characteristic equation for the plant.
  + Draw a block diagram of the plant.
  + Create a Simulink model to model the plant.
* Build a circuit that will allow you to perform a step test.
  + Image of circuit second diagram.
    - 

Figure : figure showing input for ADC one for first step test.

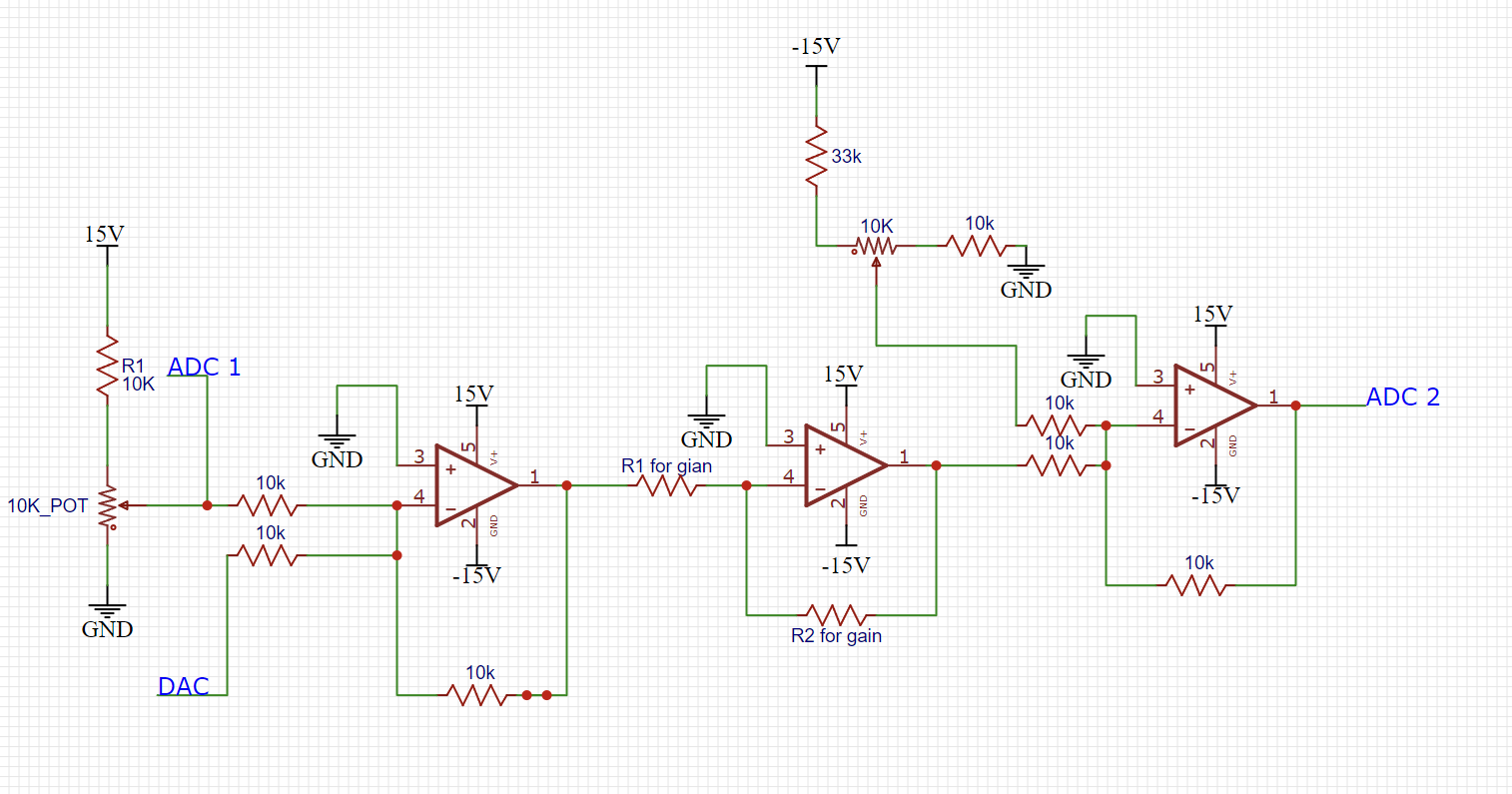
* + Performing the step test using the circuit above.
    - Turn on the computer open the simulation software and select student number.
    - Hover the helicopter by turning the voltage supplied to ADC1 to 2.5V or as close as possible.
    - Start recording data.
    - Then turn the potentiometer such that the voltage going to ADC1 increases to 5V as quickly as possible.
    - Wait for the helicopter velocity to stabilize and then stop recording.
    - Save the data gathered.
    - Repeat this process three times to get an average.
* Take this data and analyze it to find your characteristic equation.
  + First remember you need to differentiate the position graph to get it into velocity which will result in a nice unit step first order response.
  + Calculate C the conversion between meters to volts this can be done using division.
  + Use this first order response to calculate the gain (A) taking into the account the 2.5V amplitude step.
  + Calculate the Tau value using 63.3% of your total rising value. This can then also be used to calculate b.
* After this is done run tests on MATLAB to ensure all data is correct.
* Calculate the gain that will allow your Kitty copter in closed loop to meet the required specifications.
  + This is done via using the closed loop equation for the circuit and the root locus as well as simulation files to determine what the exact gain required is.
  + Calculations will be done to ensure that all specifications are meat.
* Once this has been done build the circuit with that gain characteristic.
  + Below lies an image of the complete circuit.
    - 

Figure : circuit diagram of the controller that needs to be built on Vara board.

* + - Piece of variable board the circuit was then built on a piece of variable word.



Figure : image of circuit built on Vera board

* + - The female headers are used to vary resistive values to allow for variable gains later during testing.
  + Once data has been collected on the section report must be written up to show how well the controller meets the specifications required.

## Modelled System and Results

### A free-body diagram illustrating the forces acting on the Kitti copter.

A diagram of a cat

Description automatically generated

Figure FBD of kitty copter

### A block diagram of the whole system showing the various signals.

A diagram of a diagram

Description automatically generated

Figure : full block diagram

### A differential equation describing the motion of the Kitti copter.

### A transfer function derived from the differential equation.

Assuming initially at rest.

# System Identification

## Step Response Test

### Figures and Tables

#### Calculating the offset due to the step disparity

Graph showing the input steps with voltage on the Y axis and time in seconds on the X axis.

A screenshot of a computer

Description automatically generated

Figure : graph showing the input steps.

|  |  |  |
| --- | --- | --- |
|  | Start voltage value | End voltage value |
| Step 1 | 2.47600000000000 | 5 |
| Step 2 | 2.51500000000000 | 5 |
| Step 3 | 2.51500000000000 | 5 |
| Average | 2.5 | 5 |

Figure : table used to calculate the offset error due to the change in the unit step input

Offset = 5 – 2.5 = 2.5 V.

#### Calculating C

The data in the table below was only taken when the output voltage was changing to avoid errors created by saturation.

|  |  |
| --- | --- |
| Step data was gather from |  |
| Step 1 | 0.699959122496253 |
| Step 1 | 0.700040317161672 |
| Step 1 | 0.699973635644609 |
| Step 1 | 0.699974312869253 |
| Step 1 | 0.699962681925613 |
| Step 1 | 0.700059916117436 |
| Step 1 | 0.699988509709296 |
| Step 1 | 0.699989026665204 |
| Step 1 | 0.699989553953828 |
| Step 1 | 0.700059512001587 |
| Step 1 | 0.699962413080248 |
| Step 1 | 0.700062227753578 |
| Step 1 | 0.700041999160017 |
| Step 1 | 0.700031725888325 |
| Step 1 | 0.699955083096272 |
| Step 1 | 0.699992934360206 |
| step 2 | 0.699932111337407 |
| step 2 | 0.700116298388437 |
| step 2 | 0.699935504675911 |
| step 2 | 0.699906803355079 |
| step 2 | 0.700014865467519 |
| step 2 | 0.699915110356537 |
| step 2 | 0.699906178796408 |
| step 2 | 0.700025303643725 |
| step 2 | 0.700047630388188 |
| step 2 | 0.700033553293815 |
| step 2 | 0.699916089783931 |
| step 2 | 0.700019657951641 |
| step 2 | 0.699963174369361 |
| step 2 | 0.699982752673336 |
| step 2 | 0.700008079502303 |
| step 2 | 0.699992425963796 |
| step 2 | 0.699964476021314 |
| Average | 0.699991605692488 or 0.7 |

Figure : table used to calculate C the conversion value between meters to volts.

There for the answer is C = 0.7

#### Table and figures used to calculate the first order characteristic equation.

A screenshot of a graph

Description automatically generated

Figure : image showing the three different step report response data is collected response. With the response or height of the helicopter being represented in blue, the location of being represented via the green line and the Max value reached being represented by the red line.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Step 1 | 41.3000000000034 | 7.20000000000010 | 0.138888888888887 |
| Step 2 | 41.3100000000172 | 7.10000000000201 | 0.140845070422495 |
| Step 3 | 41.3000000000034 | 7.09999999999999 | 0.140845070422535 |
| Ave | 41.3033333333414 | 7.13333333333404 | 0.140193009911306 |

Figure : table used to calculate the values from the step response such as or the total gain of the system, Tau and b.

#### Calculating A

#### Characteristic equation of plant for the output velocity of the plant.

*Or*

#### Characteristic equation of plant in terms of altitude.

*or*

## Model Validation

### Checking velocity

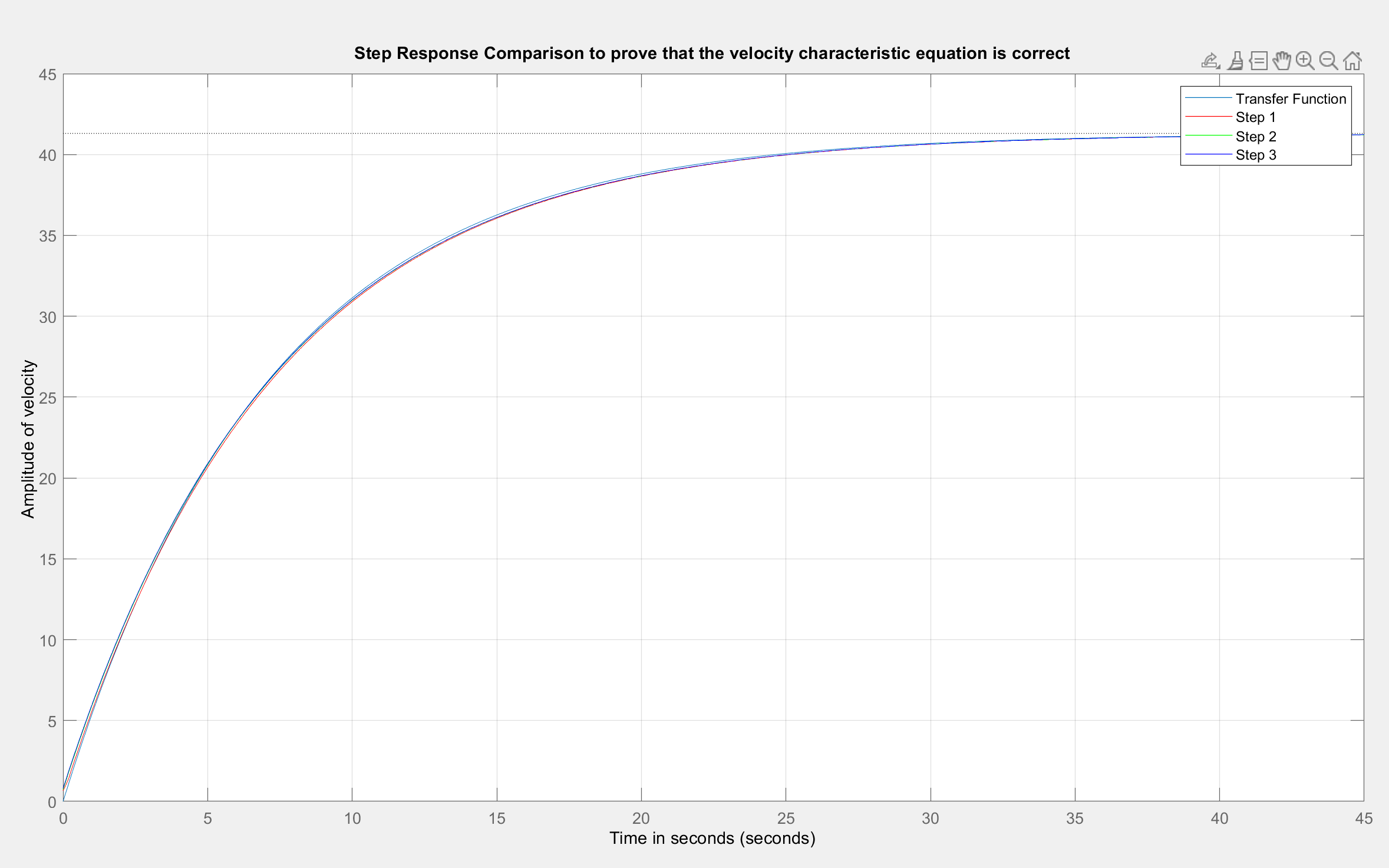


Figure : graph made with MATLAB code showing that the transfer function is correct.

### Check position.

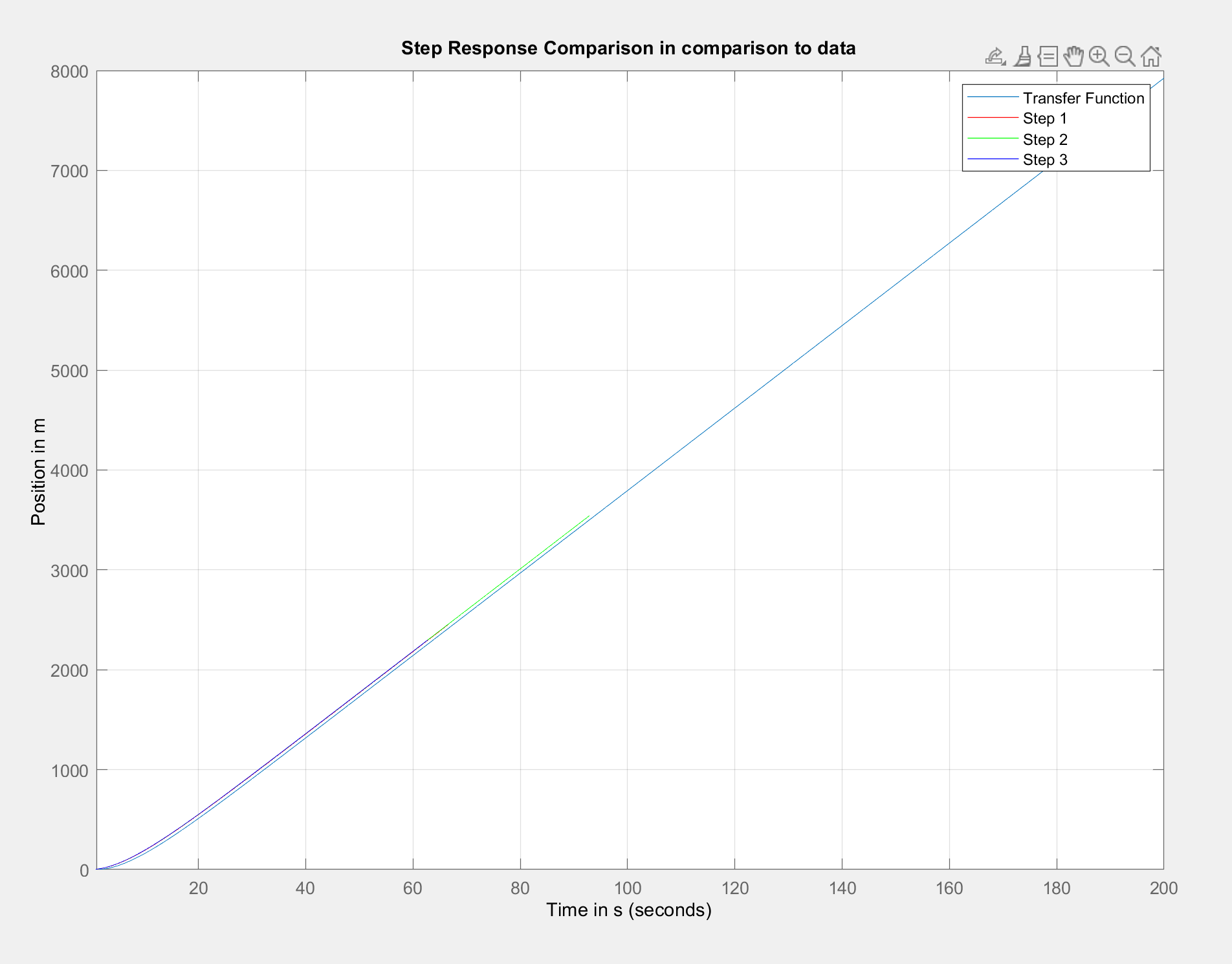


Figure : graph made with MATLAB code showing that the transfer function is correct.

From the two graphs above we can see our system is extremely close to the actual data gathered indicating our characteristic equation is correct.

# Controller Design

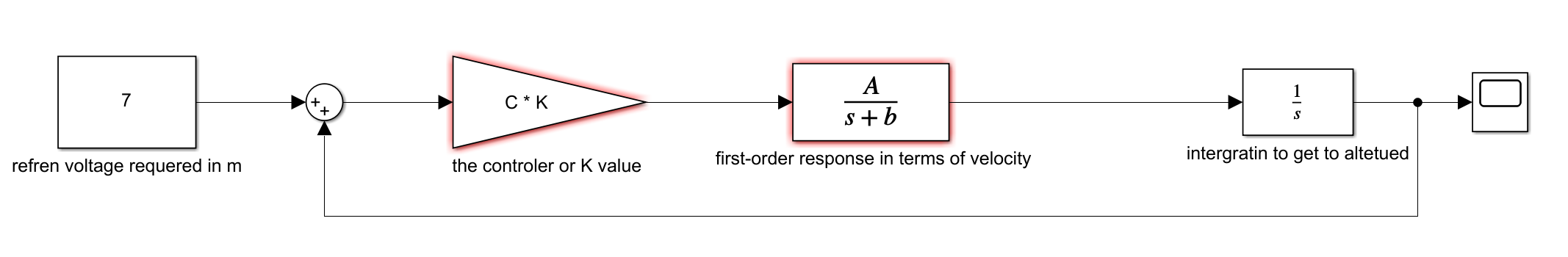
Deriving the formula for the close loop system. From the black diagram.

Step 1: Rearranging to get C K in the controller.

A diagram of a mathematical problem

Description automatically generated

Step 2: ignoring gravity.



Step 3: deriving the formula

Where:

C : conversion constant from voltage to meters.

K : is the proportional controller gain value is.

Stander second order transfer function has the for of:

There for:

* Calculation
* Calculating

* Calculate A

For this lab we were told to us a proportional controller whit the following characteristics.

* Tracking of position inputs with >90% accuracy (i.e., the tracking error and effects of disturbances must be <10%).
  + If the desired altitude required is 7m then
  + This will need to be calculated depending on the set point input.
* Settling time improvement of at least 20% compared to the open-loop system.
  + Settling time of the plant
  + Setting time with control system settling time of the plant

This has no solution for k. Therefor it is impossible to change k to get the correct time.

If instead we consider it as the settling time of the position of the plant first order system it has no settling time.

A graph on a white background

Description automatically generated

It can be seen from this graph that this thing doesn’t settle.

For example, it has still not settled at s. Therefore if we're using the settling time from this graph any settling time value would be appropriate.

* overshoot of less than 5%.

* + 0.05 >

0.05 >

0.05 >

Let

Or

There for

* Robustness to uncertainty of up to 10% in the aerodynamic constant of the system. (Since you are determining this parameter experimentally, you should make sure your design works even if you don’t get it precisely correct.)

Theoretical test of robustness of the system.

If the controller gain value of 3.3 is chosen. The following graphs will result when taking a 10% uncertainty on the gain (A) of the helicopter and on the tau value of the helicopter.

Results from A = A(1+0.1 ) and is equal to

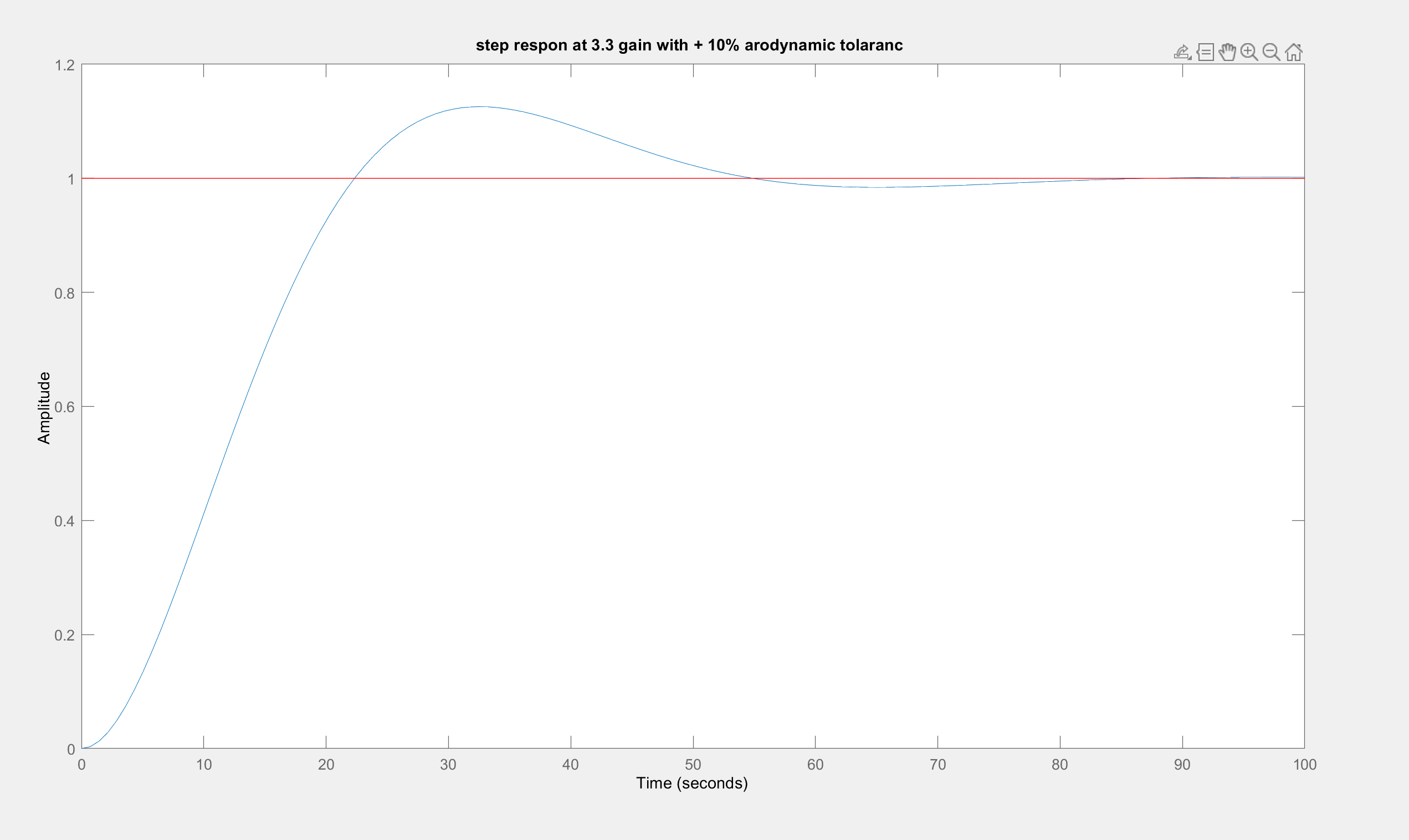


Figure : system showing how the tracking will work with +10% aerodynamic tolerance

From figure 14 it can be seen that it will still track accurately. However, it will no longer be able to track with an overshoot under 5%. The time for the system to become stable has also increased.

Tracking at -10% aerodynamic constants

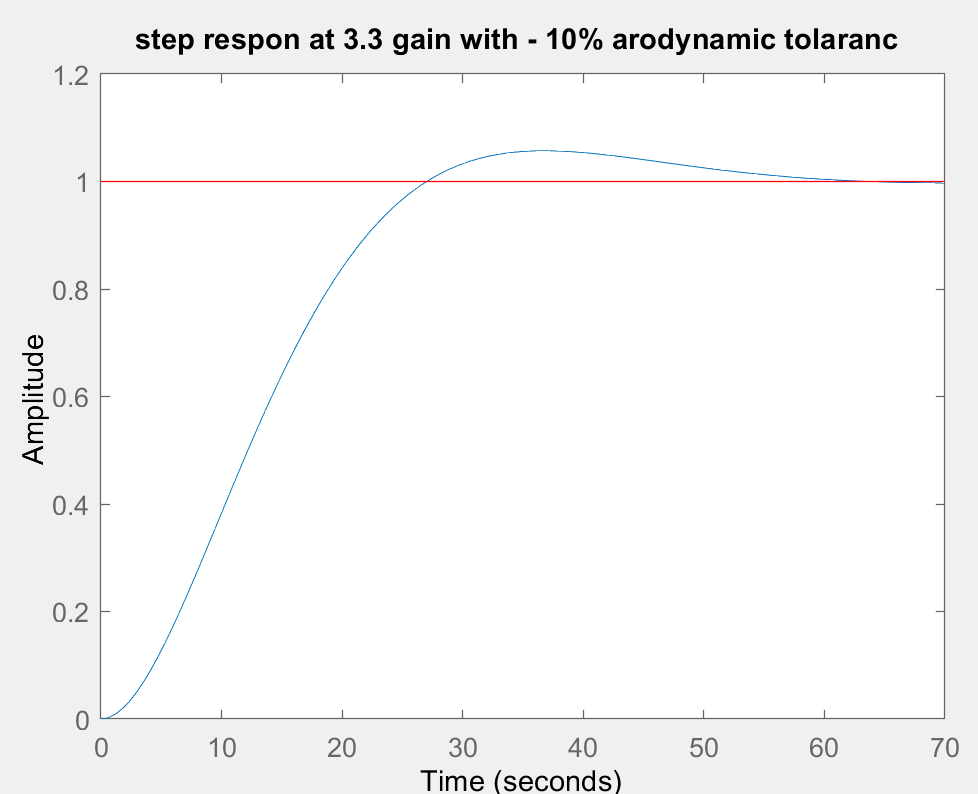


Figure : figures showing tracking abilities at -10% aerodynamic constant.

This shows theoretically the system will tracks well within the specifications required at -10% aerodynamic constant with reduced speed.

* Robustness to a tolerance of 10% in the components used to assemble the controller.

The main two resistors that will affect tolerance are the two resistors used to derive the gain. The other resistors in the circuit can be ignored because they are accounted for by manual changing or potentiometers.

If a 3.3 K and 1 mega ohm resistor are used to get a gain of 3.3 and both these resistors have a +-5% tolerance.

Inverting Opamp gain formula

Response when the gain is set to

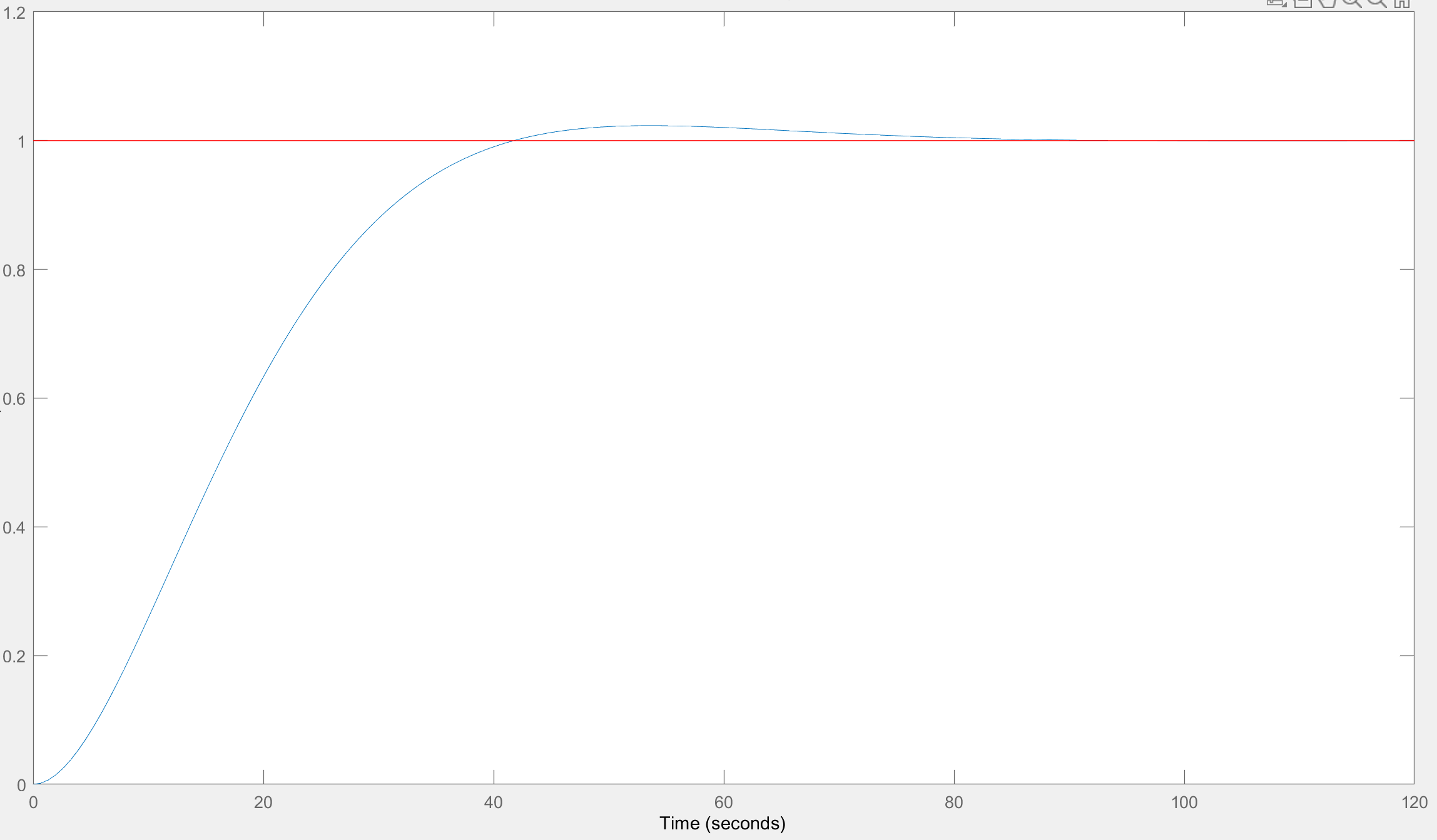


Figure : response showing the result when the gain is set to for Max gain due to resistor tolerances.

Response when the gain is set to 0.0029.

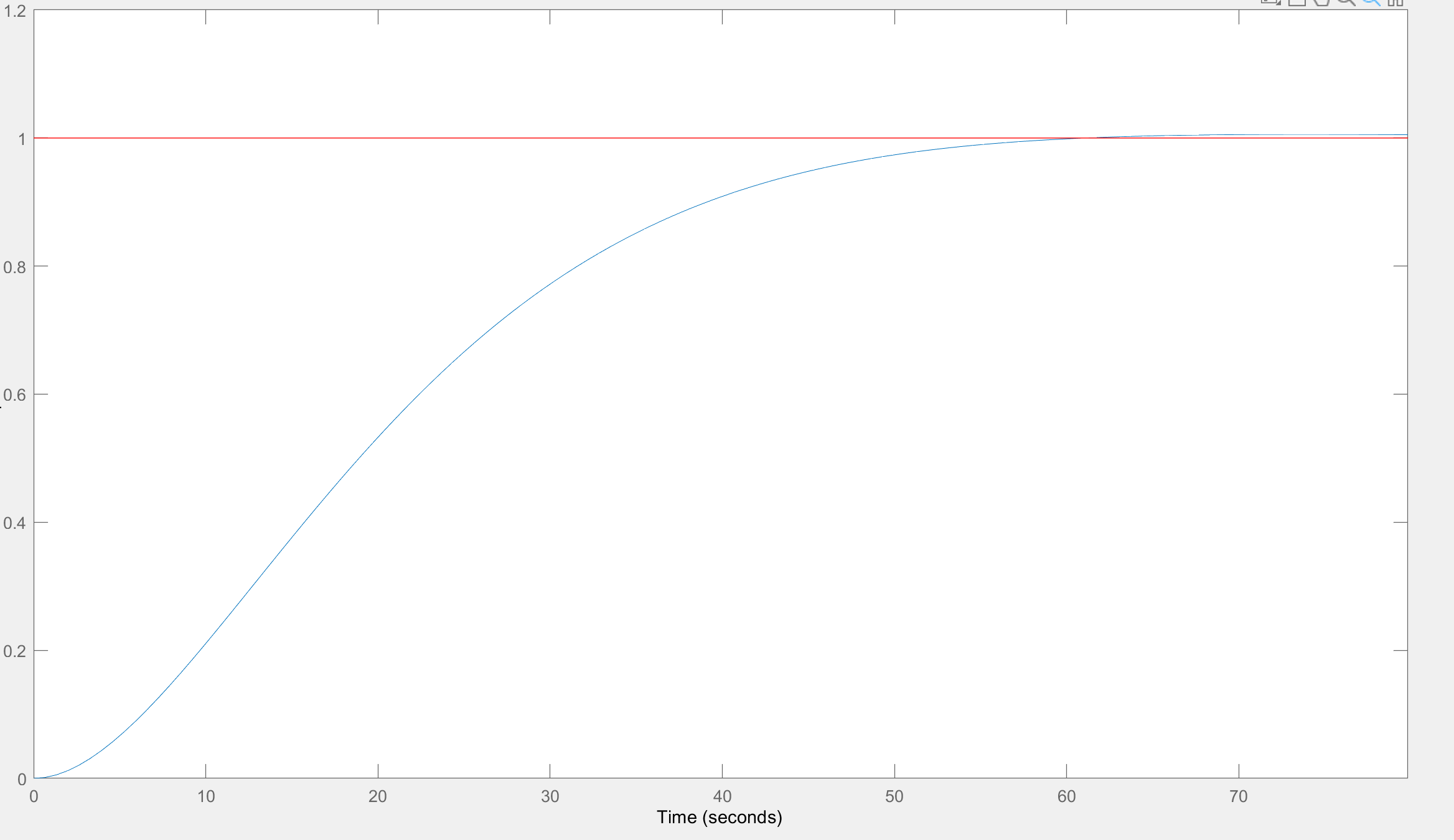


Figure : response showing the result when the gain is set to 0.0029 for minimum gain due to resistor tolerances.

* BONUS [up to an extra 20%]: Tracking of velocity inputs with >80% accuracy ( (amathoba, 2023)
  + Calculating the Gain required.
  + Finale value theorem
    - Velocity error constant =
    - Percentage error =
    - 20% =
    - Therefore
    - is true there it a c follow any ramp.

A screenshot of a computer program

Description automatically generated

Figure : code used to prove it doesn't matter what gain value is used the graph will always track the ramp function

Output



Figure output from the test code.

A graph with a red blue and white line

Description automatically generated

Figure graphs from test code.

Because the 20% error is more than the difference between the control altitude and the desired altitude proves that the value is tracking the ramp within the parameters required.

The following parameters we're chosen choose

A gain value of 0.0033 was chosen as it was easily configurable with resistors and operational amplifiers meaning the number of resistors used could be reduced and still be close to the less than 5% overshoot requirements.

## Tests performed on the Controller.

### Theoretical testing

A screenshot of a computer program

Description automatically generated

Figure : code used to plot the step response of the control system

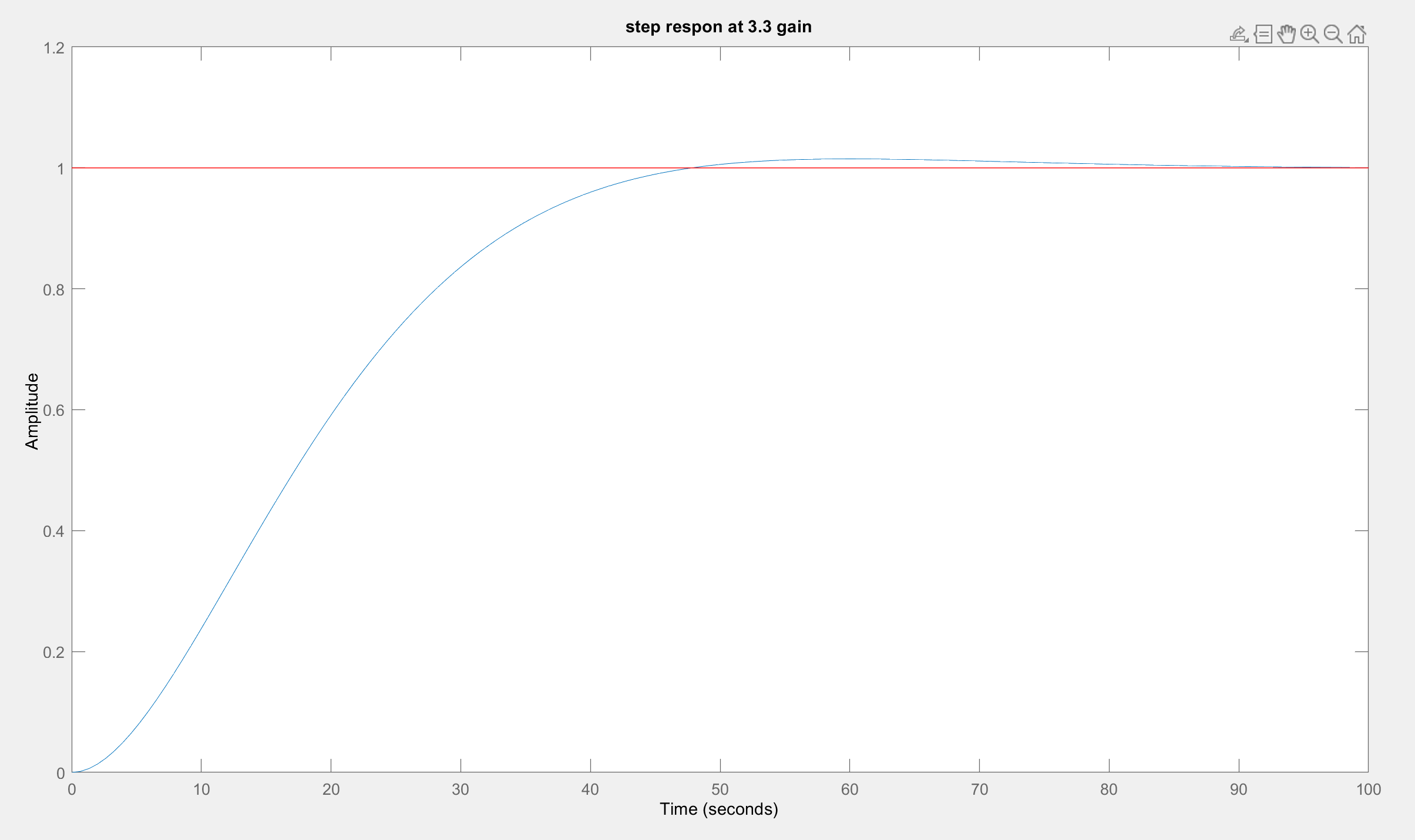


Figure : plot showing the theoretical response at 3.3 gain

This result is below 5% overshoot and has a reasonable rise time. The fact that it below 5% even though it should be slightly above 5 % overshot at 3.3 gain is most likely due to rounding errors in my calculations.

This proves theoretically that the controller can work at 3.3 gain.

## Performance Evaluation

### Practical performance

A screenshot of a computer

Description automatically generated

Figure : image of performance evaluation using control system designer.

As can be seen from the control system designer the system works at a higher gain than we expected regarding overshoot but will not meet the required time specification with any pole configuration.

## Recommendations for Future Iterations

From theoretical analysis I suggest using the next type of controller E.g. a type 2. For better tracking of a ramp and a more accurate and faster step response.

# Controller Testing

To test the controller the following circuit was built and connect to the simulation software on the computer in the control labs.

A diagram of a circuit

Description automatically generated

Then the second potentiometer was tern to account for the offset due to gravity. If ADC 1 is reading a value of zero this would entail turning the second potentiometer so that you get a value of 2.5 volts out of ADC 2

Then data is recorded and potentiometer stepped the voltage at ADT 1 to about 7 volts.

### Physical testing in control lab

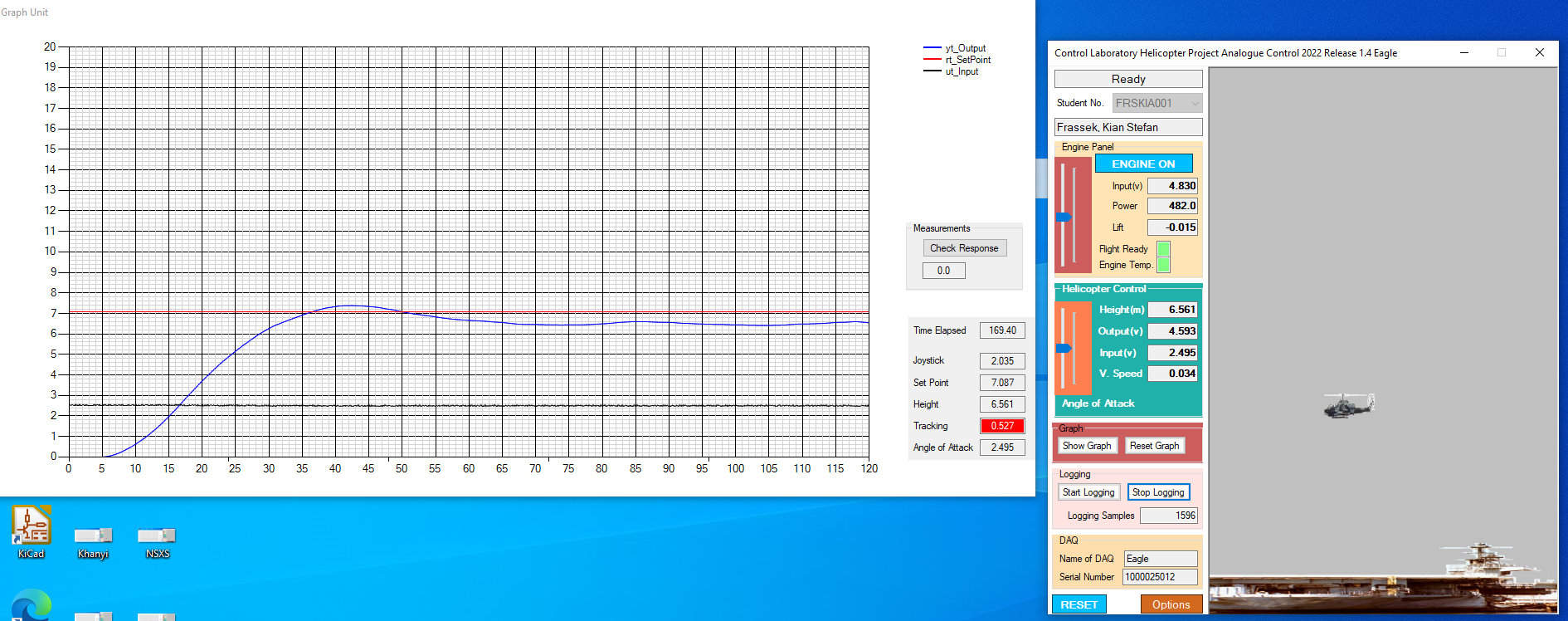
The first Test performed at 3.3 gain via the use of a 3.3K and 1 mega ohm resistor. 

Figure 22: full zoomed out image of first 3.3 game test.

Graphs showing the result at 3.3 gain. The Y axis representing altitude in meters and the X axis representing time in seconds.

A graph with lines and a curve

Description automatically generated

Figure 23: zoomed in image of step response from first 3.3 gain test.

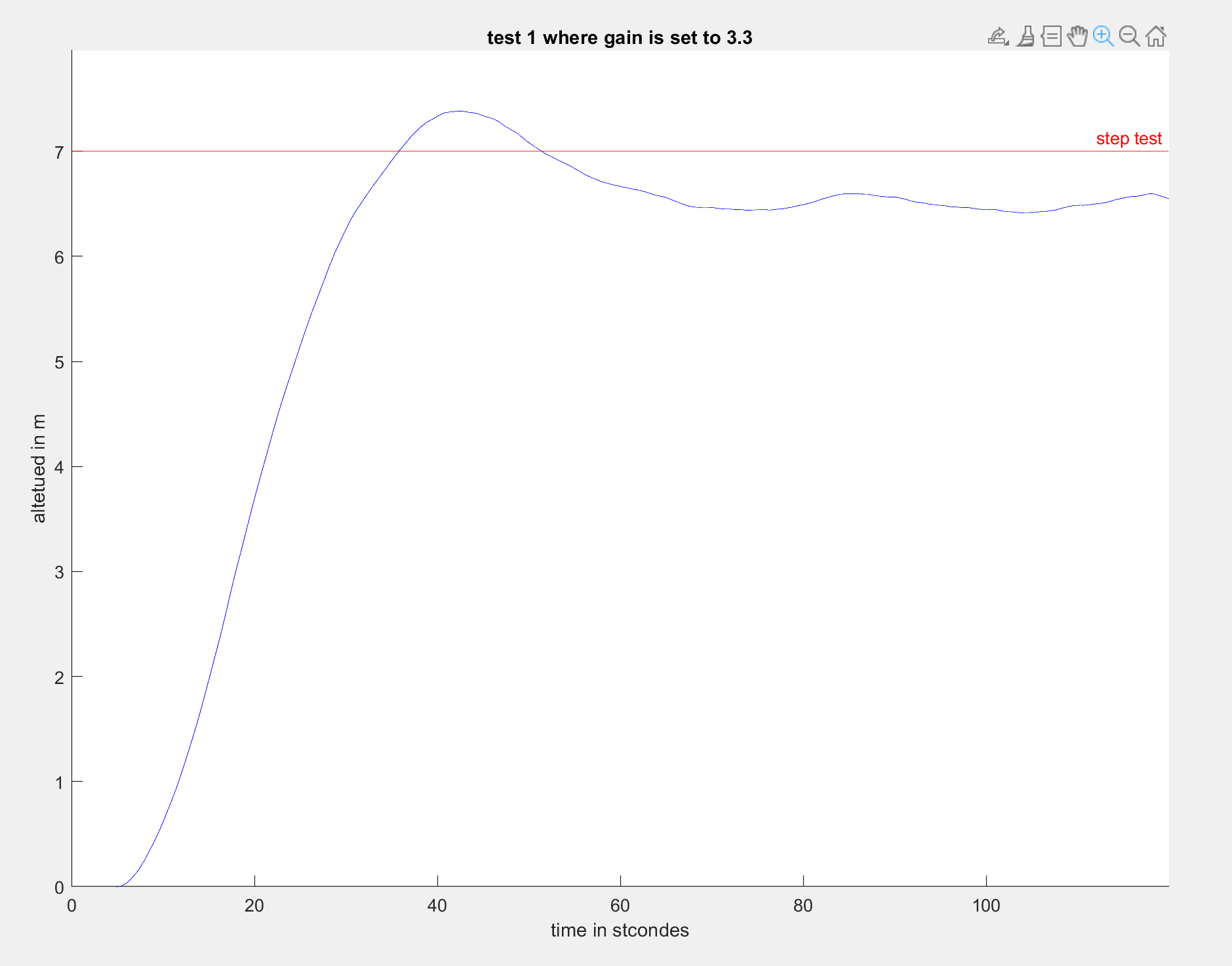


Figure 24: graph of the data logged.

The second test performed at 3.3 gain via the use of a 3.3K and 1 mega ohm resistor.

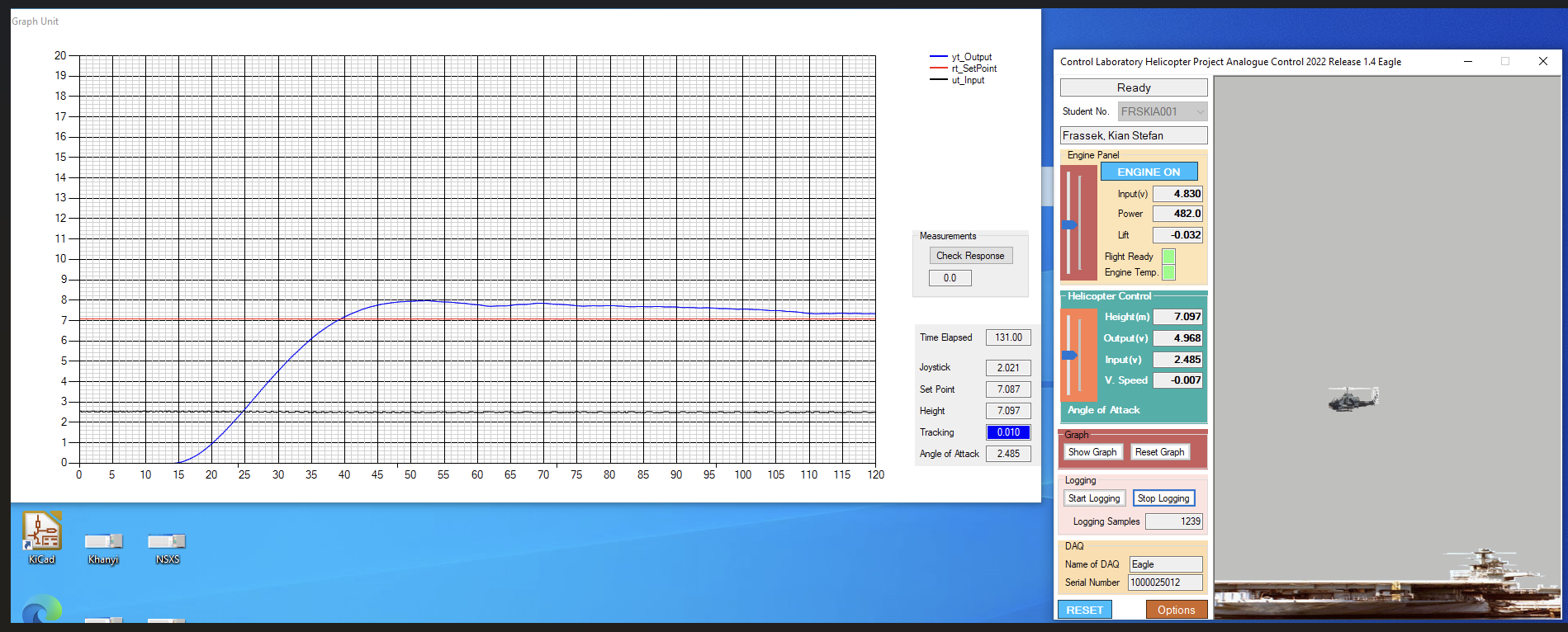


Figure 25: full zoomed out image of first 3.3 game test.

Graphs showing the result at 3.3 gain. The Y axis representing altitude in meters and the X axis representing time in seconds.

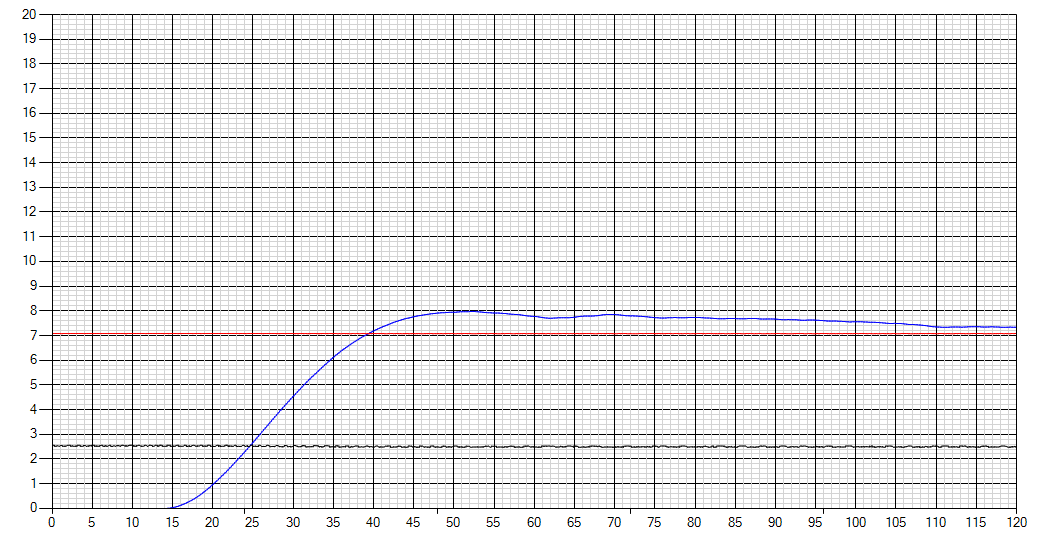


Figure 26: zoomed in image of step response from first 3.3 gain test.

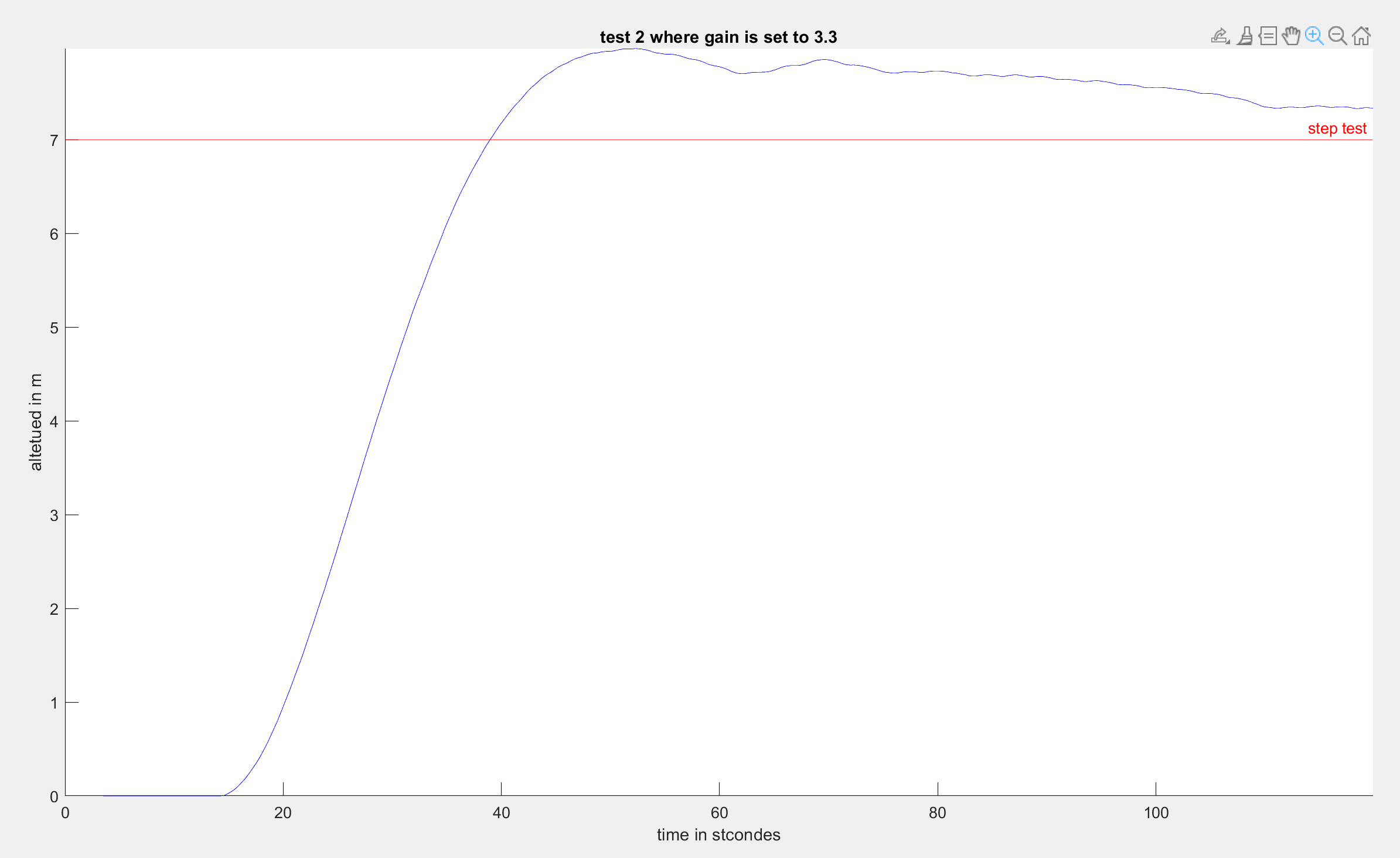


Figure 27: full zoomed out image of first 3.3 game test.

The Test performed at 33 gains via the use of a 33K and 1 mega ohm resistor.

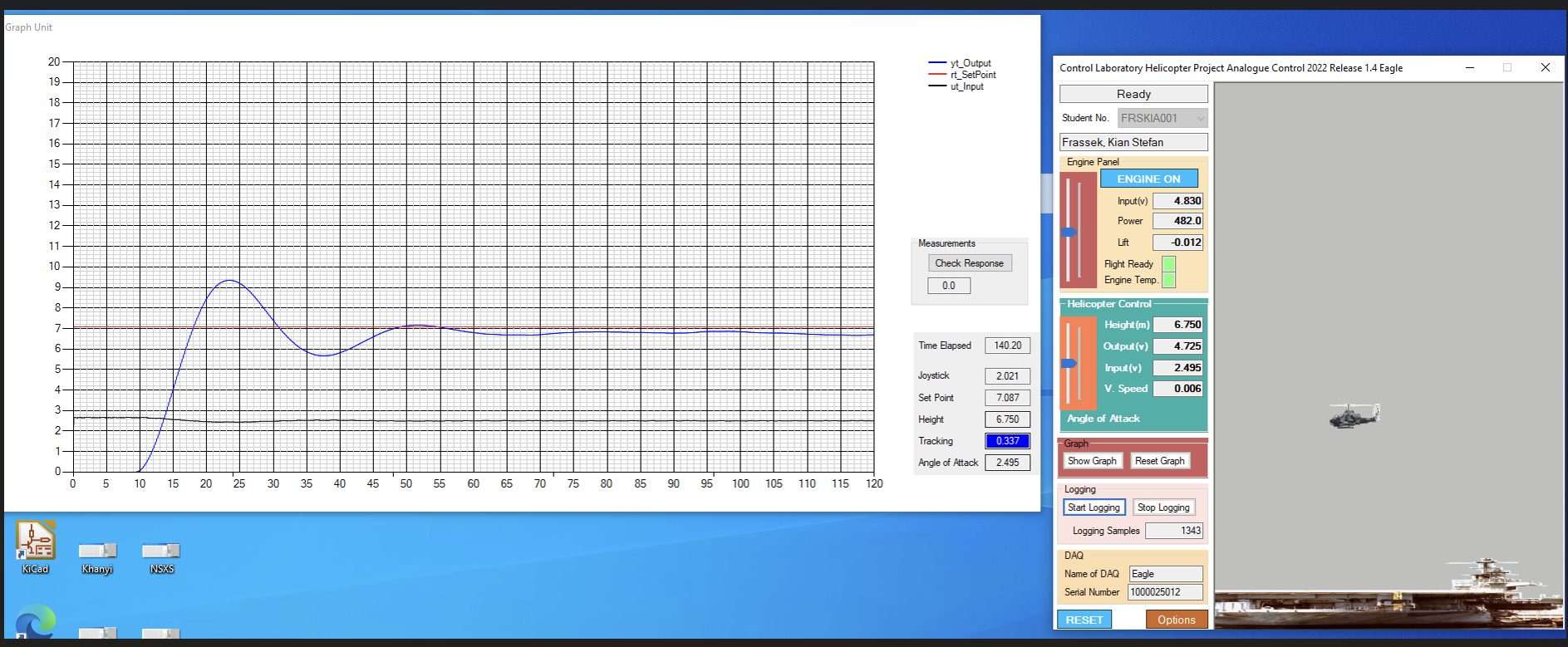


Figure 28: full zoomed out image of first 3.3 game test.

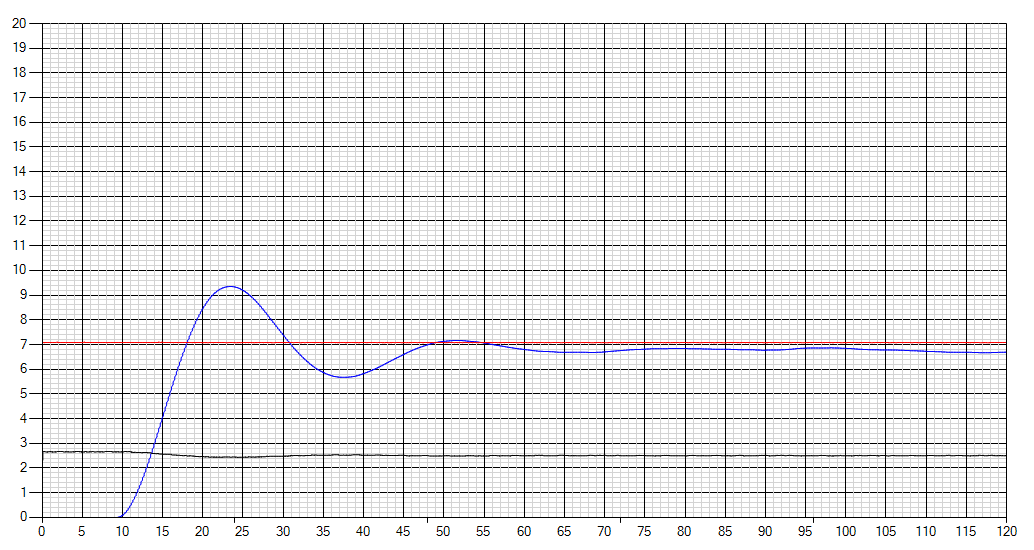


Figure 29: zoomed in image of step response from first 3.3 gain test.

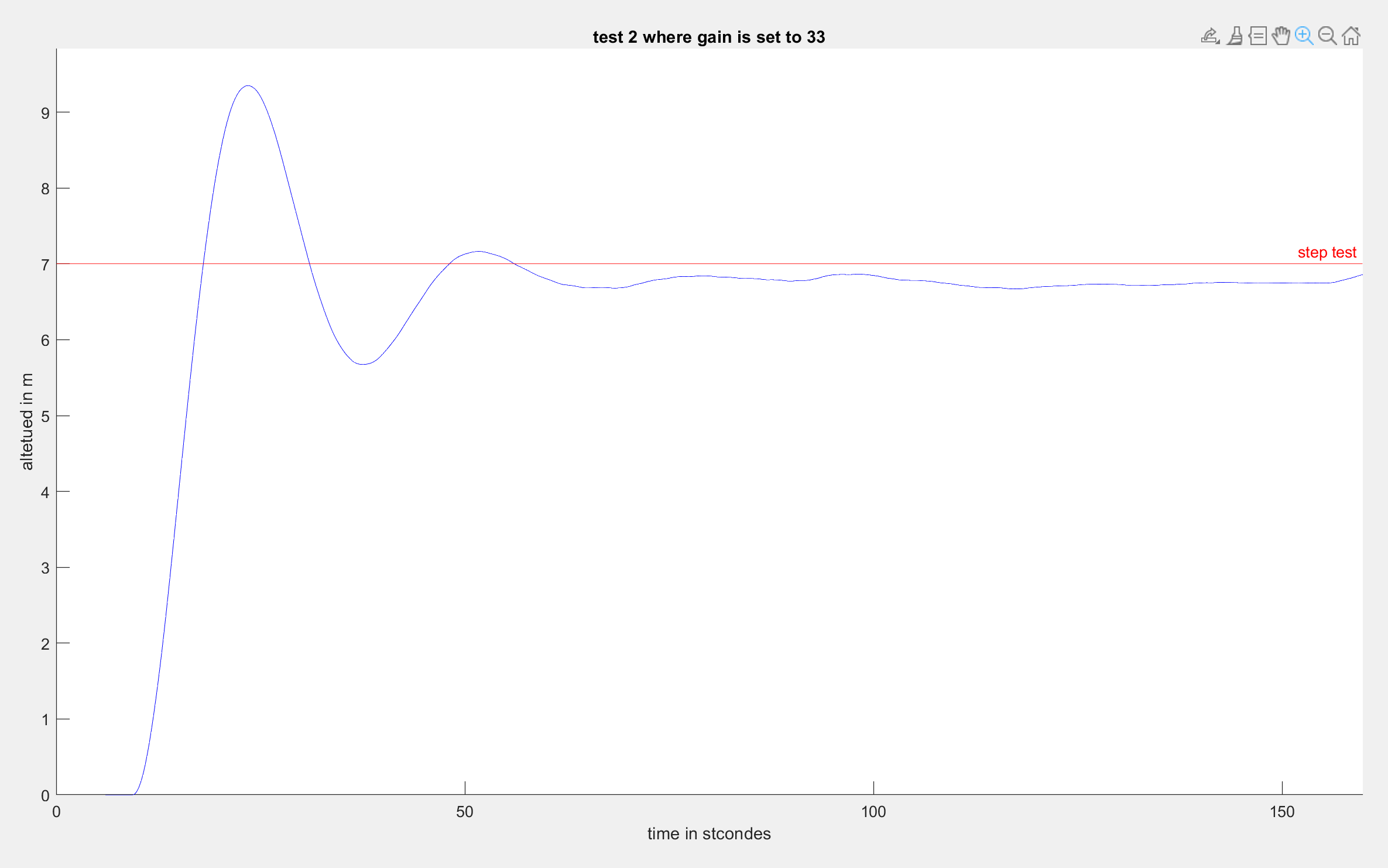


Figure 30: full zoomed out image of first 3.3 game test.

### Physical performance from controls lab computers

It can be seen the circuit is extremely noisy and is affected by the offset error of the Opamps.

This can be seen because the theoretical overshoot of 5% was greatly exceeded again of 3.3. And it's settled outside the range of 10% accuracy. In other words, this circuit would only be suggestible if you were under a tight budget scenario and could not afford to improve your controller.

### Recommendations from physical testing

I would also suggest changing the 33 and 10K ohm resistors connected to the pot on the Right-hand side of the circuit design to a larger value.

This value could be calculated like so

Let

This resistor configuration would allow for a much more accurate potentiometer adjustment within the desired range. This would allow for easier reduction of the offset error of the operational amplifiers.

Capacitors could also be added to stabilize the system, but this would have to be done carefully to ensure that you are not converting it into a type 2 system while doing this.

Probably the most optimal solution would be to create a circuit that only uses one op amp and try and increase the order of magnitude of the type of the system or in other words increase it to a type 2 system.

# Discussion and Conclusion

In conclusion it is possible to build this type of controller to control the altitude of a Kitty copter. However it is highly sensitive to noise and would not be effective solution for this specific Kitty copter. The reason for this is because the system is highly affected by noise. The circuit built requires a small gain of 0.003 which is hard to accurately produce with 5% tolerant resistors. Also loading effects play a large role as well as offset errors created by Opamps.

If this experiment is done again, it is recommended that's a different circuit is built with higher quality components. For example better quality op amps, resistors with better tolerances and stabilization capacitors I used to build the circuits.

# Appendix and Notes

Below lies it gets hub link to a repository with MATLAB files and all data collected.

Below lies all the MATLAB code used for this project

% Initialize cell array to store data for each step

step\_array\_of\_array = cell(3, 1);

A\_prime = []; % veriable use to stor the totale gain of the plant from physical testing

start\_step\_value = []; % used to recored starting value form sep test

ending\_step\_value = [];% used to recored the ending value form sep test

constat\_V\_to\_m = []; %That he used to convert from volts to metres

for step\_number = 1:3

% Read data from the CSV file

step\_array\_of\_array = readmatrix(['step ', num2str(step\_number), '.csv']);

% Calculate delta t and delta y

delta\_time = diff(step\_array\_of\_array(:, 1));

delta\_position = diff(step\_array\_of\_array(:, 4));

velocities = [0; delta\_position ./ delta\_time];

% Add velocities to the main array

step\_array\_of\_array(:, 5) = velocities;

% Initialize variables for data extraction

step\_correct\_start = [];

time\_start = 0;

start\_higt = 0;

for i = 1:length(velocities)

if (velocities(i) > 0.5) && (time\_start == 0)

time\_start = step\_array\_of\_array(i, 1);

start\_higt = step\_array\_of\_array(i, 4);

end

% anilisine and storing each steps perameters

if velocities(i) > 0.5

step\_correct\_start = [step\_correct\_start; step\_array\_of\_array(i, 1) - time\_start, velocities(i), step\_array\_of\_array(i, 4) - start\_higt];

if step\_array\_of\_array(i, 3) < 10

constat\_V\_to\_m = [constat\_V\_to\_m; step\_number, step\_array\_of\_array(i, 3) / step\_array\_of\_array(i, 4)];

end

end

end

% Store corrected data in the cell array

steps{step\_number} = step\_correct\_start;

% Calculate A\_prime

A\_prime = [A\_prime; max(steps{step\_number}(:, 2))];

disp("A\_prime for " + step\_number + ' is ' + max(steps{step\_number}(:, 2)));

% Store start and end values

start\_step\_value = [start\_step\_value; step\_array\_of\_array(5, 2)];

ending\_step\_value = [ending\_step\_value; step\_array\_of\_array(500, 2)];

end

% Calculate damping coefficient

index = [];

tau = [];

b\_The\_damping\_coefficient = [];

for step\_number = 1:3

targetValue = A\_prime(step\_number) \* 0.63;

[~, index] = min(abs(steps{step\_number}(:, 2) - targetValue));

tau = [tau; steps{step\_number}(index, 1)];

end

b\_The\_damping\_coefficient = 1 ./ tau;

b\_The\_damping\_coefficient\_ave = mean(b\_The\_damping\_coefficient);

% Plot step response form data gathered

figure;

for step\_number = 1:3

subplot(3, 1, step\_number);

plot(steps{step\_number}(:, 1), steps{step\_number}(:, 2));

title(['Step ', num2str(step\_number)]);

xlabel("time in s");

ylabel("velocity in m/s");

line(xlim, [A\_prime A\_prime], 'Color', 'r'); % Adding the maxe value line

line([tau(step\_number) tau(step\_number)], ylim, 'Color', 'g'); % Adding the tau value line

end

% Calculate average values

A\_prime\_ave = mean(A\_prime);

ending\_step\_value\_ave = mean(ending\_step\_value);

start\_step\_value\_ave = mean(start\_step\_value);

b = ending\_step\_value\_ave - start\_step\_value\_ave;

constat\_V\_to\_m\_ave = mean(constat\_V\_to\_m(:, 2));

% Calculate tau

tau\_ave = mean(tau);

A = A\_prime\_ave / (b \* constat\_V\_to\_m\_ave);

disp("A prime ave = " + A\_prime\_ave + newline + "the value of b = " + b + newline + ...

+ newline + "k average = " + constat\_V\_to\_m\_ave ...

+ newline + "A ave = " + A ...

+ newline + "tau and b = " + tau\_ave);

% Define the transfer function G

s = tf('s');

G = tf(A\_prime\_ave, [tau\_ave, 1]);

% Plot step response for G for velocity

figure;

hold on;

step(G);

% Plot step responses for each step

plot(steps{1}(:, 1), steps{1}(:, 2), 'r');

plot(steps{2}(:, 1), steps{2}(:, 2), 'g');

plot(steps{3}(:, 1), steps{3}(:, 2), 'b');

legend('Transfer Function', 'Step 1', 'Step 2', 'Step 3');

xlabel('Time in seconds');

ylabel('Amplitude of velocity');

title('Step Response for velosity');

grid on;

hold off

% Define controller and plant

k = 0.0033;

gain = tf(k);

Constat\_V\_to\_m\_ave = tf(constat\_V\_to\_m\_ave);

plant = tf(A \* constat\_V\_to\_m\_ave / (s + ((s^2) \* tau\_ave)));

% ploting the step responses for each step potition

figure;

hold on;

step(plant)

plot(steps{1}(:, 1), steps{1}(:, 2), 'r');

plot(steps{2}(:, 1), steps{2}(:, 2), 'g');

plot(steps{3}(:, 1), steps{3}(:, 2), 'b');

tow\_ave

legend('Transfer Function', 'Step 1', 'Step 2', 'Step 3');

xlabel('Time in seconds');

ylabel('Amplitude of velocity');

title('Step Response Comparison');

grid on;

hold off;

gain

% Calculate Kv

Kv = dcgain(s \* G);

% Calculate steady-state error for ramp input

ess = 1 / Kv;

disp("Steady-state error: " + ess);

%controlSystemDesigner(plant);

% Closed-loop system

G\_closed\_loop = feedback(plant \* gain, 1);

% ploting the closed loop respons

s = tf('s');

figure;

hold on;

step(G\_closed\_loop);

step(tf(1), "r");

title("step respon at 3.3 gain");

hold off;

% Plot ramp response

figure;

hold on;

step(G\_closed\_loop/s);

step(1/s, "r");

xlim([0, 200]);

title('Ramp Response');

hold off;

% Interpolate values for error calculation

t = 0:0.01:200;

y1 = step(G\_closed\_loop/s, t);

y2 = step(1/s, t, "r");

t\_desired = 120;

y1\_desired = interp1(t, y1, t\_desired);

y2\_desired = interp1(t, y2, t\_desired);

difference = abs(y1\_desired - y2\_desired);

calculating20error = t\_desired - t\_desired \* (1 - 0.2);

disp("Difference between the two values = " + difference + " 20% error = " + calculating20error);

% Anilisiny the controles data with gain at 3.3

test\_time = [];

for i = 1:2

% Read data from the CSV file

step\_array\_of\_array = readmatrix(['test 3.3 gain ', num2str(i), '.csv']);

figure

hold on

plot(step\_array\_of\_array(:,1),step\_array\_of\_array(:,4),"B");

xline([0 ,120]);

yline(7,"r","step test");

title("test " +i+" where gain is set to 3.3");

xlabel("time in stcondes");

ylabel("altetued in m");

end

% Anilisiny the controles data with gain at 33

step\_array\_of\_array = readmatrix(['gain at 33.csv']);

figure

hold on;

plot(step\_array\_of\_array(:,1),step\_array\_of\_array(:,4),"B");

xline([0 ,160]);

yline(7,"r","step test");

title("test " +i+" where gain is set to 33");

xlabel("time in stcondes");

ylabel("altetued in m");

hold off;

% cheching areodnamic tolerensis +10%

tua\_toleran\_up = tau\_ave\*(1.1);

A\_tolerans\_up = A\_prime\_ave\*(1.1);

G\_tolarece\_up = tf(A\_tolerans\_up\*constat\_V\_to\_m\_ave, [tua\_toleran\_up, 1]);

figure;

hold on;

step(G\_tolarece\_up);

% Plot step responses for each step

plot(steps{1}(:, 1), steps{1}(:, 2), 'r');

plot(steps{2}(:, 1), steps{2}(:, 2), 'g');

plot(steps{3}(:, 1), steps{3}(:, 2), 'b');

legend('Transfer Function', 'Step 1', 'Step 2', 'Step 3');

xlabel('Time in seconds');

ylabel('Amplitude of velocity');

title('Step Response for velosity with + 10% arodynamic tolaranc');

grid on;

hold off;

% effect on the control system

plant\_10\_plus = tf(A\_tolerans\_up \* constat\_V\_to\_m\_ave / (s + ((s^2) \* tua\_toleran\_up)));

% Closed-loop system +10%

G\_closed\_loop\_10\_plus = feedback(plant\_10\_plus \* gain, 1);

% ploting the closed loop respons

figure;

hold on;

step(G\_closed\_loop\_10\_plus);

step(tf(1), "r");

title("step respon at 3.3 gain with + 10% arodynamic tolaranc ");

hold off;

% cheching areodnamic tolerensis +10%

tua\_toleran\_down = tau\_ave\*(0.9);

A\_tolerans\_down = A\_prime\_ave\*(0.9);

G\_tolarece\_down = tf(A\_tolerans\_down\*constat\_V\_to\_m\_ave, [tua\_toleran\_down, 1]);

figure;

hold on;

step(G\_tolarece\_down);

% Plot step responses for each step

plot(steps{1}(:, 1), steps{1}(:, 2), 'r');

plot(steps{2}(:, 1), steps{2}(:, 2), 'g');

plot(steps{3}(:, 1), steps{3}(:, 2), 'b');

legend('Transfer Function', 'Step 1', 'Step 2', 'Step 3');

xlabel('Time in seconds');

ylabel('Amplitude of velocity');

title('Step Response for velosity with - 10% arodynamic tolaranc');

grid on;

hold off;

% effect on the control system

plant\_10\_min = tf(A\_tolerans\_down \* constat\_V\_to\_m\_ave / (s + ((s^2) \* tua\_toleran\_down)));

% Closed-loop system +10%

G\_closed\_loop\_10\_min = feedback(plant\_10\_min \* gain, 1);

% ploting the closed loop respons

figure;

hold on;

step(G\_closed\_loop\_10\_min);

step(tf(1), "r");

title("step respon at 3.3 gain with - 10% arodynamic tolaranc ");

hold off;

# References

*amathoba.* (2032, 10 04). Retrieved from Lab 2: Proportional Control: https://amathuba.uct.ac.za/d2l/le/lessons/14465/topics/1430215