

**Temperature Control**

18. May 2018

Authors:

Jesper Hjort

Kim C. Nielsen

Lasse A. Frederiksen

Mads H. Villadsen

Supervisor:

Waqas A. Khan

Contents

[**1 Introduction**](#_fdwlf5dvaay) **3**

[1.2 Planning](#_8gfsiyvqxngz) 3

[**2 Construction**](#_qizarua3yp7u) **4**

[2.1 The physical system](#_8dd1y5oj3sob) 4

[2.1.1 Components](#_oymba5z1o528) 4

[2.1.2 System requirements](#_xmu44bjzog5) 4

[2.1.3 Physical limits to the system](#_jwtk7pk85j1h) 4

[2.2 Measurements](#_bzdo1kp37jee) 4

[2.2.1 Collecting data](#_d2uih3f9sc36) 4

[2.2.2 Transfer functions](#_vddbcfkif3e1) 5

[**4 Modelling**](#_f57p5w7nyttm) **7**

[4.1 Controller Design](#_ehe0dxcc6uq6) 8

[4.3 Tuning process](#_kuwegxrn9xnw) 9

[**5 Circuit Realization**](#_vd7le9yhreyc) **11**

[5.1 Controller Realization](#_d5z96l4kcdei) 12

[**6 Measurements**](#_aqbib4uvm6t) **13**

[**7 Discussion**](#_alq8jvjmb3hy) **13**

[**8 Conclusion**](#_o9mgxeo7ze3m) **13**

# 

# **1 Introduction**

This report documents our semester project regarding the course ‘Dynamiske Systemer’. The reader of this report can expect an overview of the individual steps which is involved, in order to build the temperature control system.

The temperature control system involves a box made out of flamingo which contains a couple of power resistors. A power supply is applied to heat up the power resistors and a temperature sensor is used to measure the temperature in the box. The measurement from the temperature sensor is applied as an input to a controller that adjusts the temperature in the box to a desired value.

## **1.2 Planning**

In this section we would like to talk about the planning of our project and why we did it in the order that we did and won’t talk as much about why the particular persons were assigned the specifics jobs as this isn’t as important as the order of doing things.

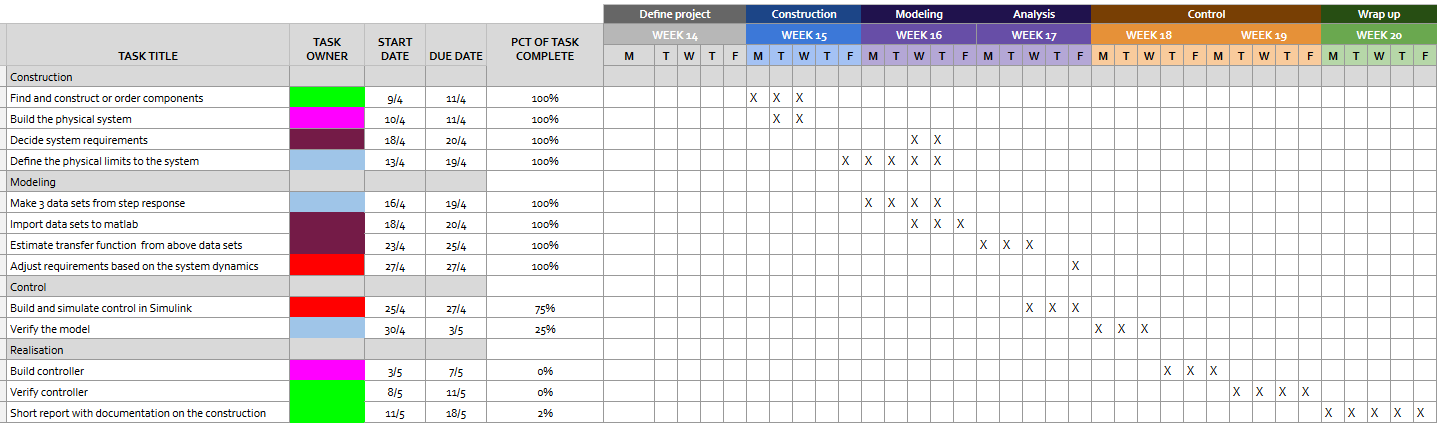
we planned our project using a gantt chart which is where you list the jobs that needs to be done and in the order they need to be done and then you put in how much time each task is approximately going to take so you can get a overview as to how much time you have.

figure x.x our gantt chart can be seen more clearly here <https://docs.google.com/spreadsheets/d/1frERea3dO8Y5UjCMWNYFHJWTBxkFqzdtrv86qszZThA/edit?usp=sharing>

We will start by explaining the individual tasks.

we started by doing some construction, the construction tasks included building the physical system so that we would have something to measure upon and was needed before we could continue to the next stages. along with building the physical system we also ordered what we needed to build it.

after we had the physical system built we could start to set up some requirement for the system so that we would know how fast we wanted it. along with setting some requirements for the system it was also important for us to set some physical limitations for the system.

after the construction was done we moved on to the next stage and started moddeling the system. In this stage we started by making some datasets and in our case we made 3 so we could create a better average of the system from different step responses. these data were then imported into matlab for further analysis so that we could start estimating some transfer functions and then adjust the requirements accordingly.

we then started the control stage, this stage was all about simulating and building the system in simulink so that we could get an idea as to whether or not the system could work properly or not. and of course verify the model. after doing the final verification in simulink we could move on to the last stage which was the realisation stage, in this stage we build the controller and of verify it to make sure it works like we want it to work. Then as the last step we needed to compose a report explaining the work we have done on the project.

# **2 Construction**

## **2.1 The physical system**

### **2.1.1 Components**

* Thermo box made of EPS expanded polystyrene aka styrofoam
* The power resistors
* The temperature sensor

### **2.1.2 System requirements**

* Temperature should be within 30℃ and 50℃
* Operating area between 0 V and 10 V
* Steady state to be reached within 60 - 120 minutes

### **2.1.3 Physical limits to the system**

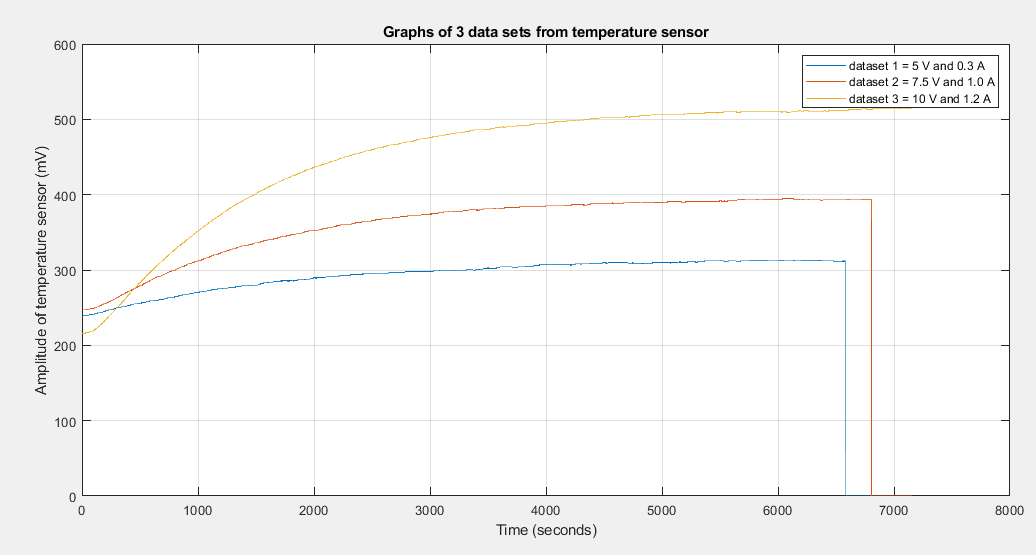
During some initial testing we realized that the box material started melting around 100℃

## **2.2 Measurements**

### **2.2.1 Collecting data**

Having build the physical system we collected 3 data sets by applying 3 different input voltages and measuring the system output for each.

The result can be seen in figure 2.1



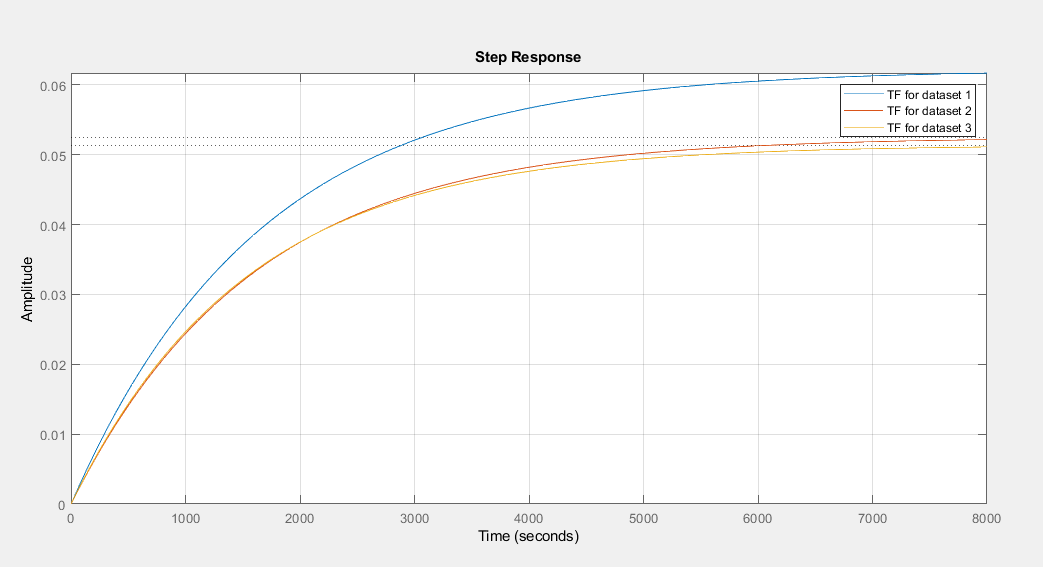
*Figure 2.1 - Graphs of 3 data sets from temperature sensor*

### **2.2.2 Transfer functions**

Next we found the transfer functions for the 3 data sets by reading of the time constants from the graphs and assuming 1. order system on the form:

G = k/(tau\*s+1)

Then the transfer functions for each data set was used to find the step responses as seen in figure 2.2



*Figure 2.2 - Step responses for the 3 transfer functions*

Now to come up with a transfer function for our overall system we took the average time constants and gains from the previously found 3 transfer functions.

This resulted in TF G = 0.05538/(1528\*s+1)

The step response for this transfer function is shown in figure 2.3

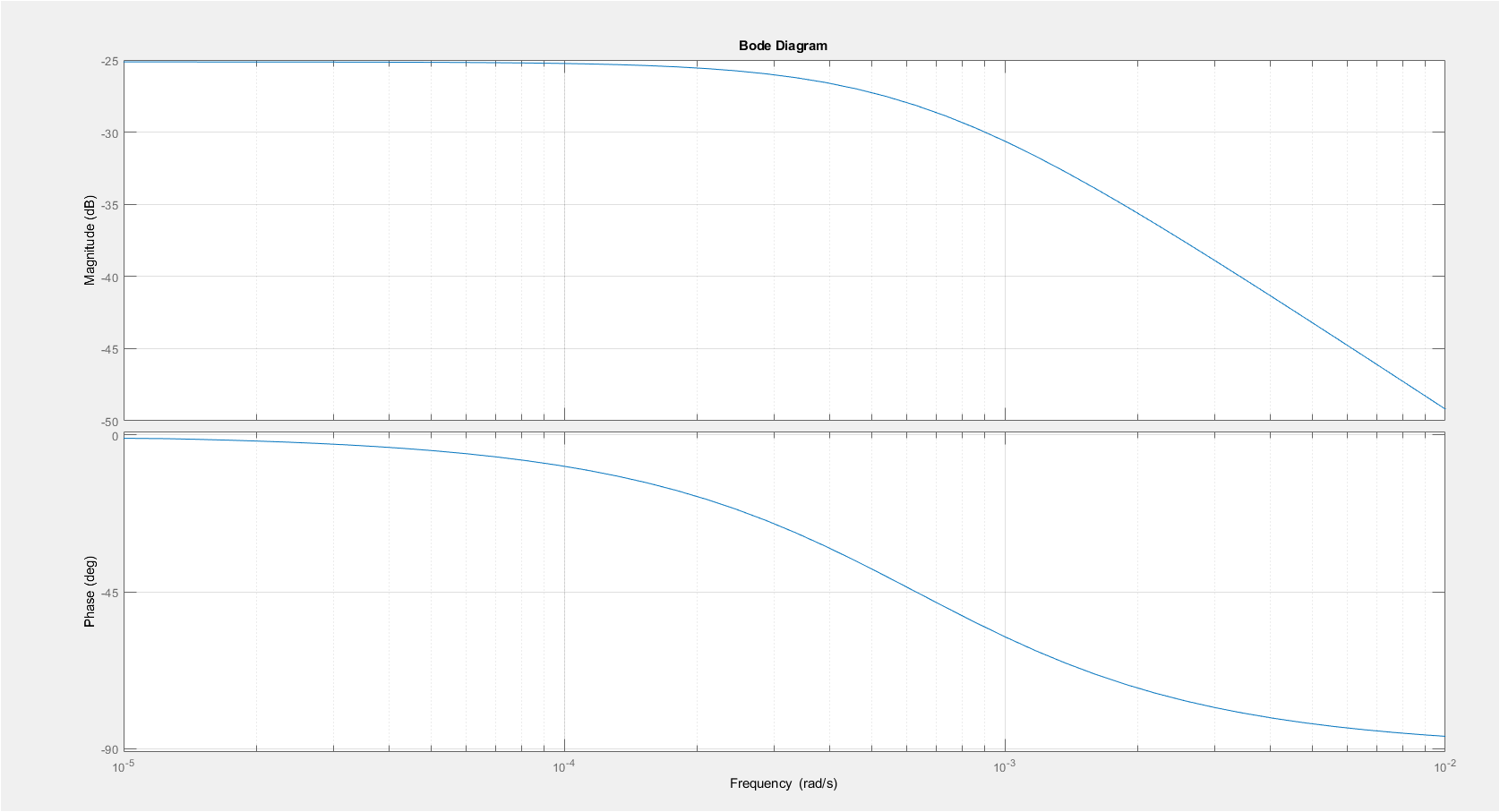


*Figure 2.3 - Step response for the overall average transfer function*

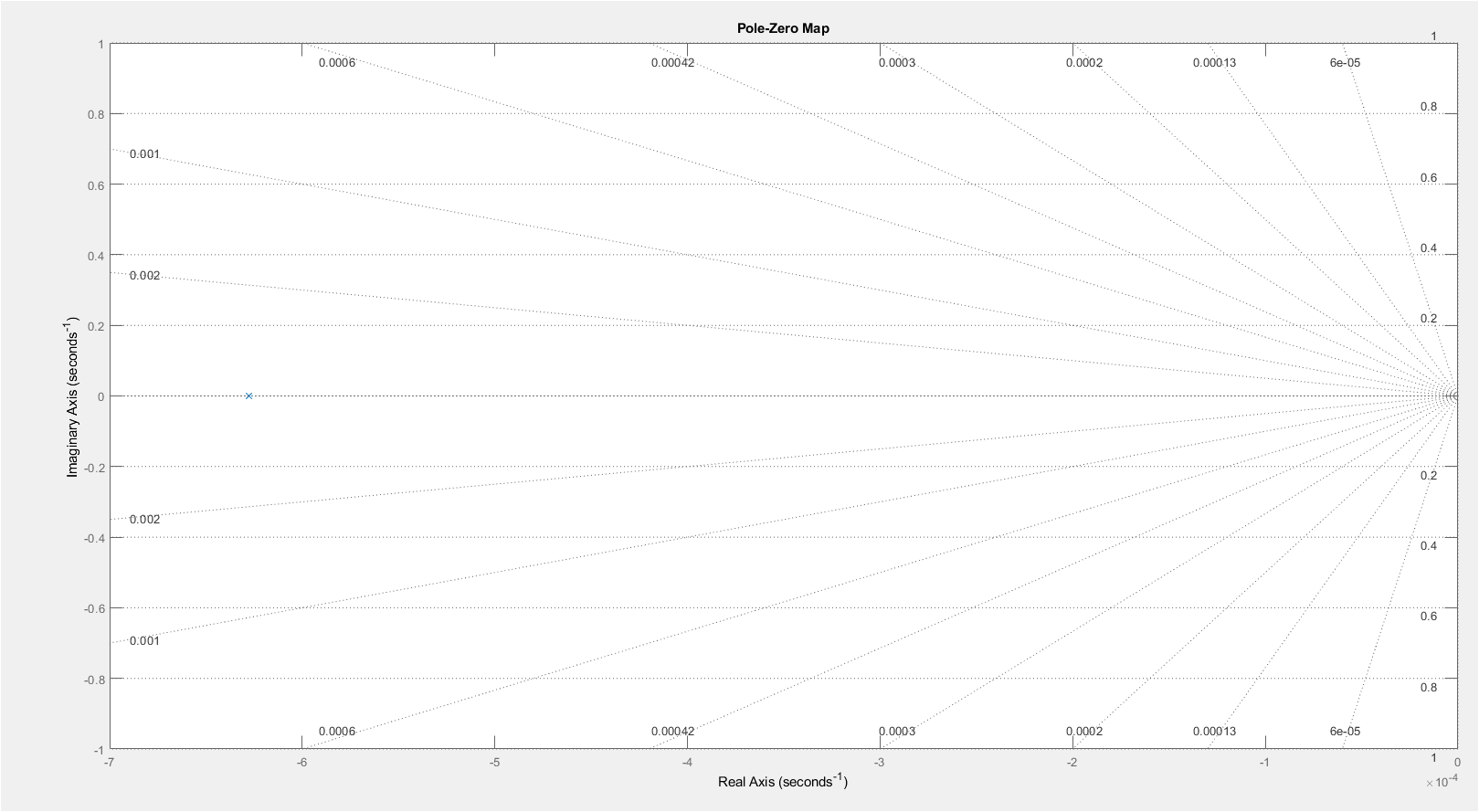
Analog discovery measurements?

# **4 Modelling**

On figure xx



*Figure xx - Bode plot, average transfer function*



*Figure xx - Pole-zero plot*

## **4.1 Controller Design**

Once the system itself has been identified and measured, one will have the characteristics of the system. These characteristics can be expressed via a transfer function, which can be used as a baseline for the design of a suitable controller. There are different types of controllers, but during the course we have only been introduced to PID controlling. This is fortunate since the PID is the most commonly used feedback controller.

The controller can consist of three parts:

* Proportional - acts on the present error
* Integral - acts on the accumulation of past errors
* Derivative - acts on future error values

Each of these controller parts can be adjusted to provide a desirable output based on specific parameters such as overshoot, steady state error, settling time etc. The design of the applied controller is based of a various tests in Simulink.

Based on various tests and experiences, a single proportional control results in a steady state error. A controller that involves a proportional control in combination with an integral control results in eliminating the steady state error, but the integral part leads to a small overshoot. The PI control already indicated an acceptable response, however the response could be further improved with a PID control, since the derivative control is able to reduce the overshoot and even improve the settling time, however, during a test of a PID control in simulink did not turn out to be a success. It was not possible to apply the derivative control in practice, since every performed simulation indicated that the derivative control had no effect, which is why the PI control was chosen as the controller in the end.

On figure xx a simulink diagram is indicated. This diagram has been used to simulate the behavior of the system with an implemented controller. The diagram consists of a step input, subtractor, controller, adder, transfer function and a scope. The step input is configured to have a final value of 0.5, since the system in practice, should never exceed 50 degrees celsius. This is based on the fact that the LM35DZ temperature sensor reads 1 degrees celsius for every 10 mV. 0.5 V is equal to 500 mV, meaning that you get a final temperature of 500/10 which is equal to 50 degrees celsius.

The controller is made as a subsystem and marked with a dark green color. It takes one input and has to outputs, one for the proportional control and one for the integral control. The internal diagram of the subsystem can be seen on figure xx.

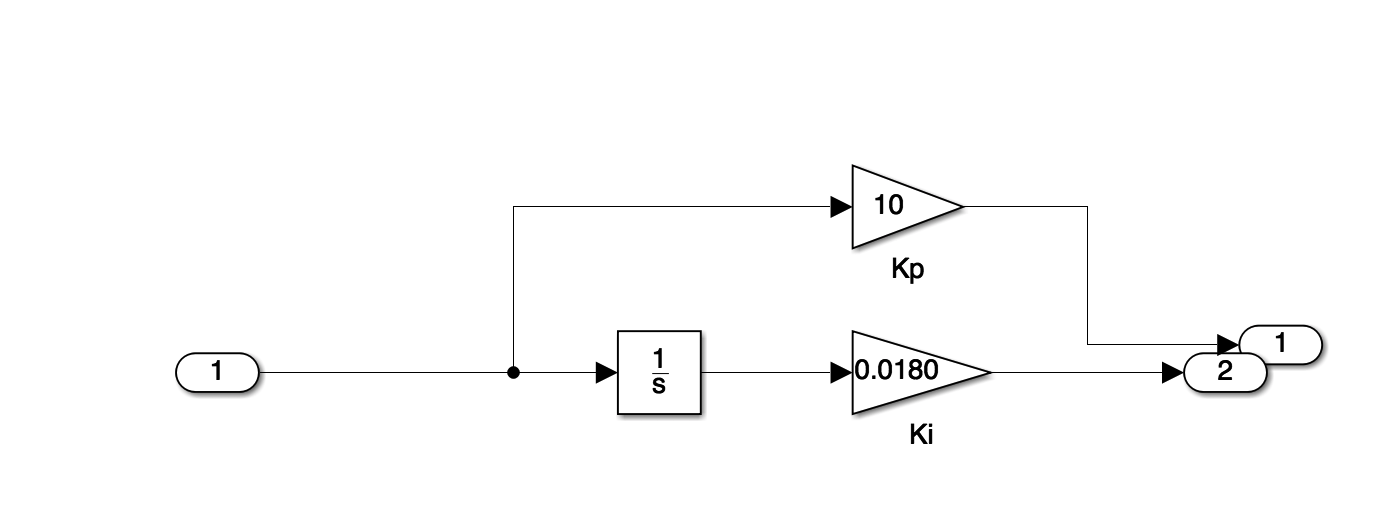
After the controller block, a transfer function can be seen. This transfer functions is an average transfer function based on the earlier calculations concerning the three transfer functions from figure xx.

The last part of the simulink diagram is a scope which is used to plot the output of the system and the step input. An example of a plot with a tuned controller can be seen on figure xx.



*Figure xx - Simulink diagram*

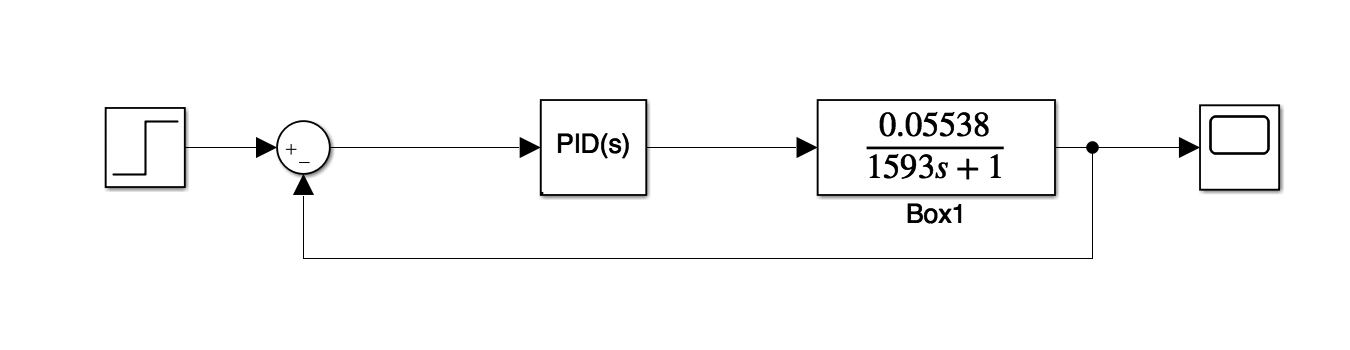
On figure xx the actual controller is indicated. The controller consists of a proportional part and an integral part. The proportional is implemented as a gain block with the value 10 and the integral part is implemented with an integrator and a gain block with the value 0.0180. These values are based on the tuning process, which will be covered in the ‘Tuning process’ section.



*Figure xx - The Controller*

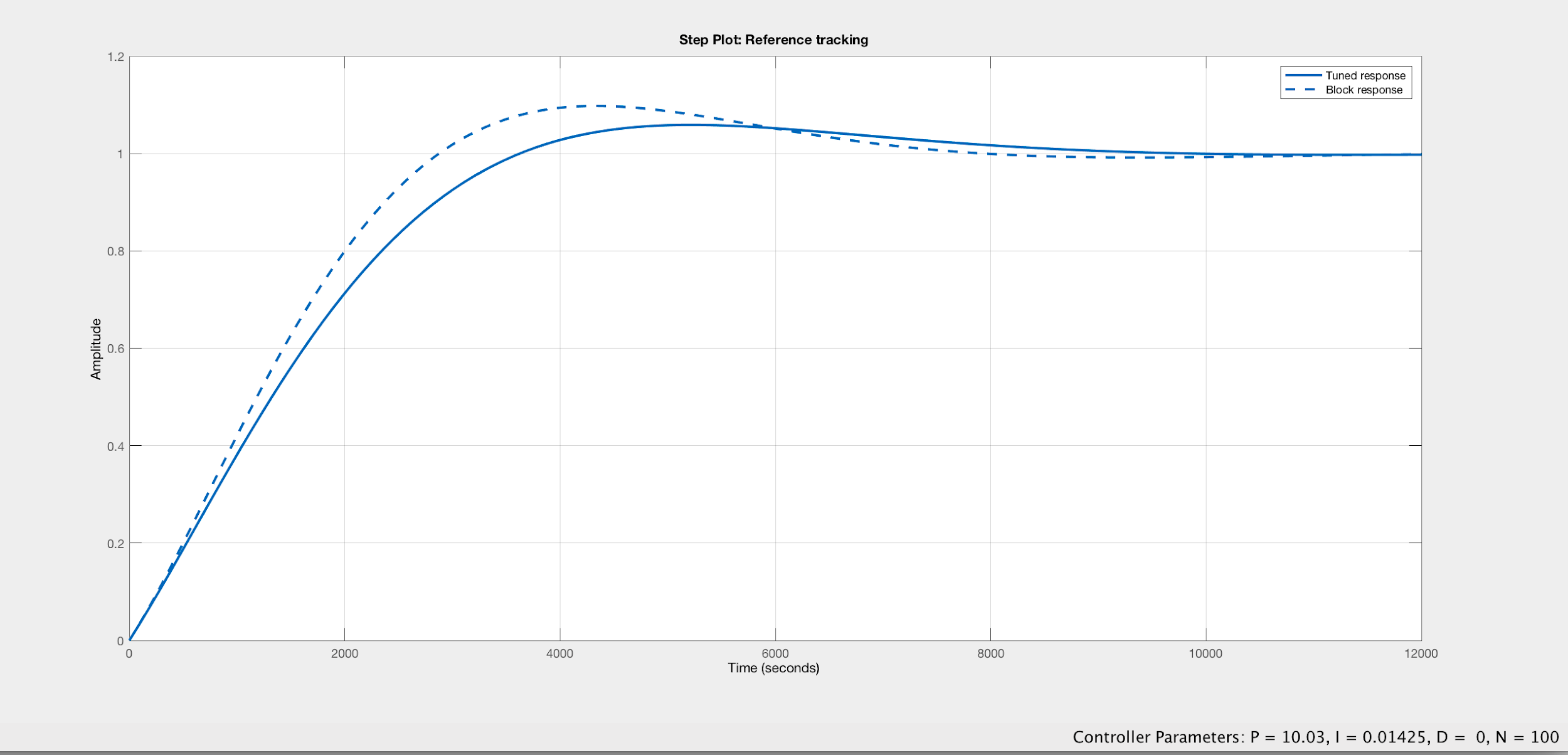
## **4.3 Tuning process**

In order to easily tune a PID controller, simulink offers a PID tuner application which can be accessed via a block called ‘PID(s)’. The block can also be used to implement a PID controller directly by setting the various gain values manually, but the gain values can also be set with the tuner app. The usage of the tuner app can be seen on figure xx.



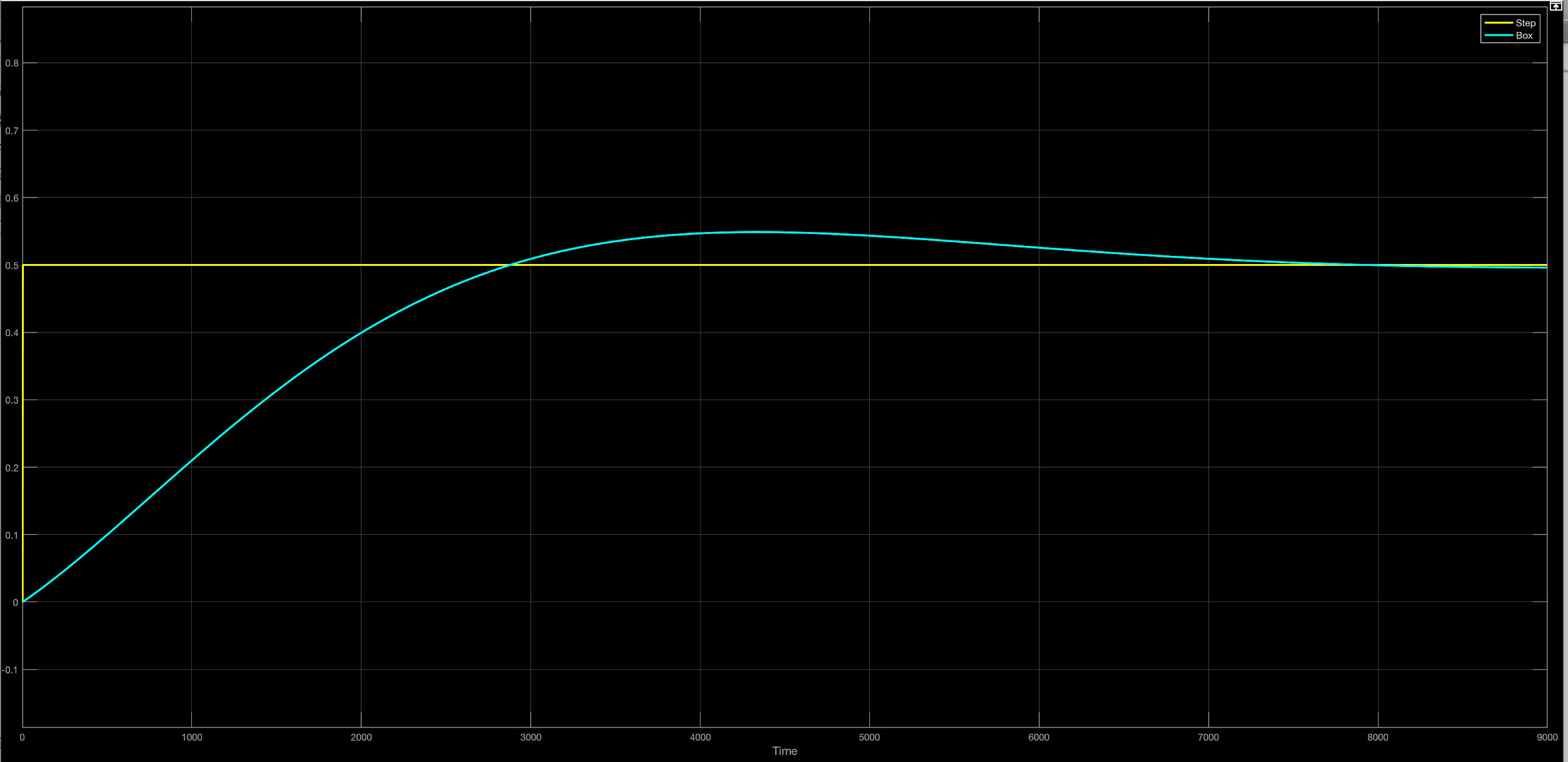
*Figure xx - Simulink model with PID block*

On figure xx the tuner app is indicated. The app shows two responses, where the dotted line is earlier block response. The other response is fully colored and can be adjusted based on e.g. response time and transient behavior. By adjusting the aforementioned parameters the controller parameters in terms of proportional gain, integral gain and derivative gain changes. The desired response was reached, when the proportional gain had the value 10 and the integral gain had the value 0.0180. The derivative gain had the value 0, since the derivative control seems to have no effect.



*Figure xx - PID Tuning app*

Once the proper gain values had been found during the tuning process with the tuner app, the implementation of the controller is done. The resulting response, based on the tuned controller parameters, can be seen on figure xx. The response is marked with cyan color. The response has very little overshoot and almost no steady state error. The only downside that can be seen is the settling time, since it takes a bit less than 8000 seconds before it reaches steady state. The settling time could have been decreased with a derivative control, but since it was not possible to perform a simulation in simulink with derivative control, the derivative control was not used. Another important thing to add concerning the settling time is also that temperature control is usually very slow, which also explains the settling time indicated on figure xx.

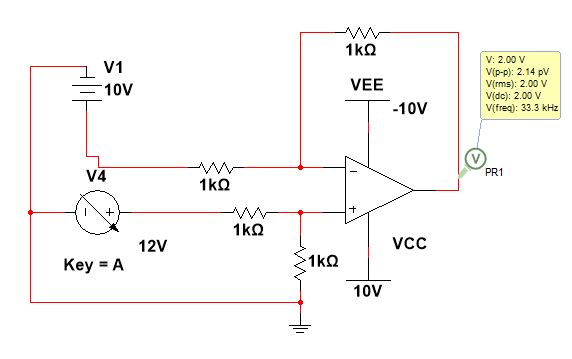


*Figure xx - Final response with tuned values*

# **5 Circuit Realization**

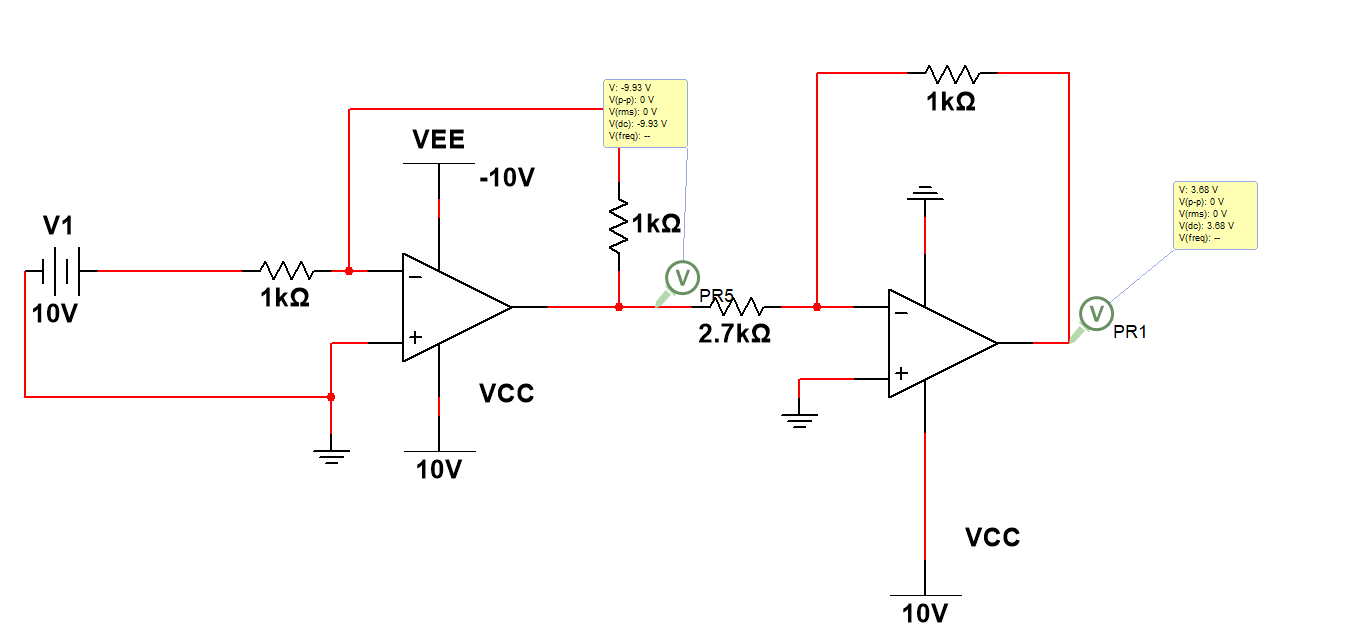
Beregninger...

On figure xx a subtractor circuit is indicated. The components are based on the calculations mentioned on figure xx. As indicated on the figure, the resistors all have the same value, meaning that the subtractor circuit acts as a unity gain buffer. The circuit has been implemented in the simulation tool ‘Multisim’ with input voltages of 10V and 12V. The circuit subtracts the voltages at the output resulting in 2V as expected.



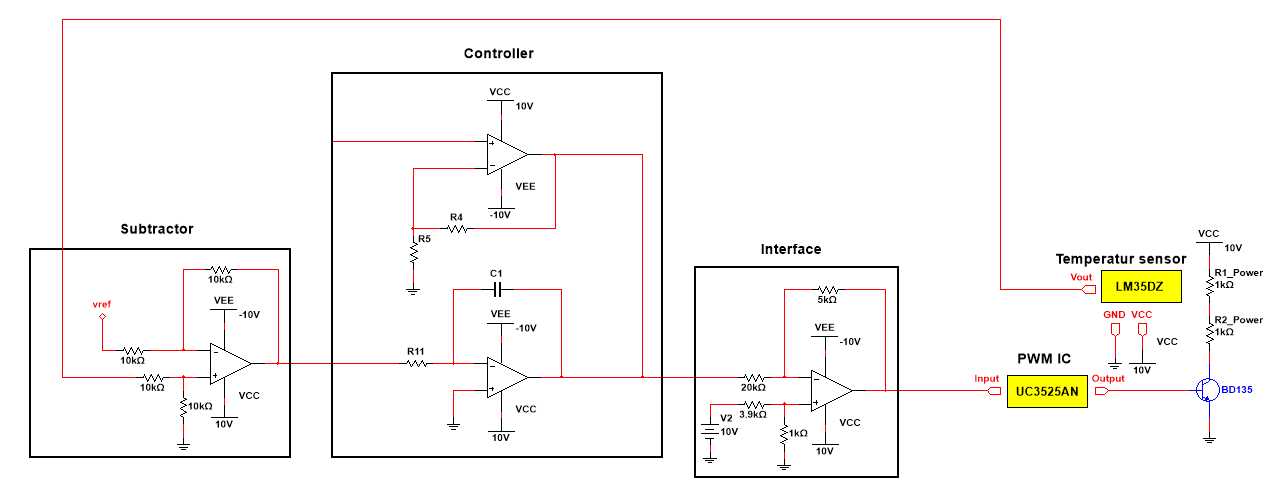
*Figure xx - Subtractor Circuit*

On figure xx a DC voltage downscale circuit is indicated. The component values are based on the earlier calculations mentioned on figure xx.



*Figure xx - DC Voltage downscale circuit*

The entire circuit can be seen on figure xx. The circuit has been divided into sections indicating the individual circuits, in which some of them were mentioned earlier in terms of subtractor and interface.



*Figure xx - The Total Circuit*

* Subtractor circuit and calculations to dimension the circuit
* Interface circuit
* PWM Chip

## **5.1 Controller Realization**

* Calculate component values based on gain values received in the tuning process
* Controller circuit part
* Simulation results from circuit
* Build circuit and verify it compared to simulations
* Include multisim tests, individual/block tests

# **6 Measurements**

* description of data gathered from test of entire system with controller etc.

# **7 Discussion**

Did the circuit work? Why not or describe which criterias it managed to fulfill

# **8 Conclusion**

During this project has given us the opportunity to learn more about how to design a suitable controller that works with a specific system. The system in this project is about controlling the temperature inside a box with a heating element.

What did we learn, did everything go as planned? Write briefly about the working process and talk about what you did in the different sections.