Hardware in the loop

Isak Skeie

(If you prefer, you may add a Figure here, but please don't use any of the Figures in the Assignment. Make sure to make and use your own Figures)

Table of Contents

1 Introduction

Hardware In the Loop, is a method of verifying and test hardware before it gets deployed in the real world [1]. By going through this assignment. Software is developed that enables the testing of hardware, putting the hardware in a realistic use case through simulation. Doing so, verifies that the hardware works are expected before being deployed in the real world, where hardware faults can affect and damage other components.

2 Ordinary Software Simulation

2.1 Temperature simulator

The temperature sensor is built as a SUB Vi in a block diagram. To accommodate for the time delay, a subVI from Finn Haugen is used (See Appendix 1). Before the output, an integration Sub Vi is used. This one is also from (See appendix 2).

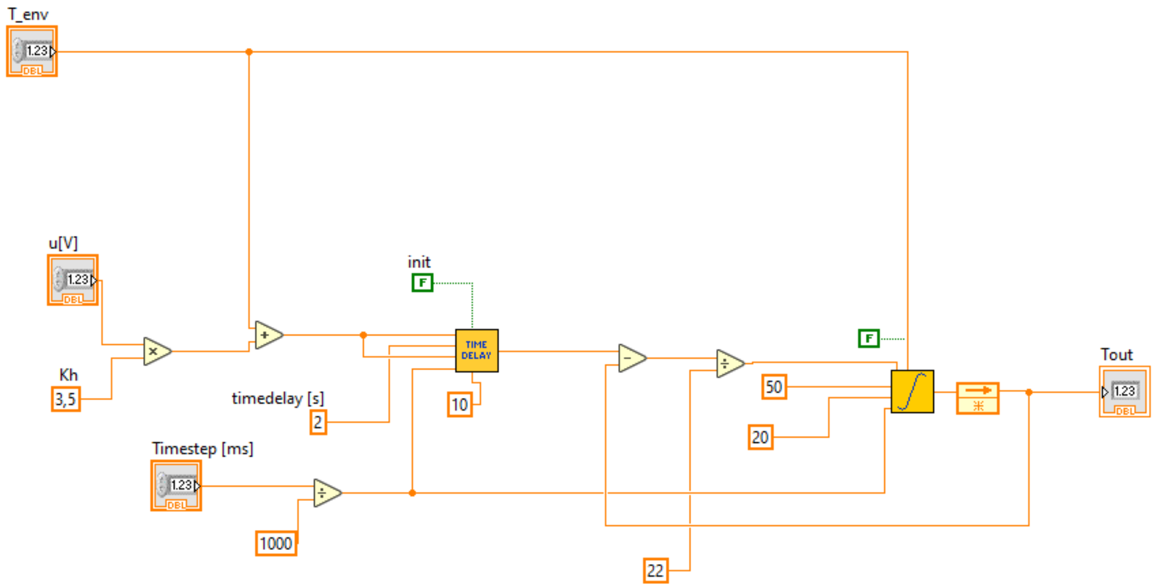

Figure 1: Block diagram for Temperature simulation.

2.2 PID

The simulator is added as a SubVi, with the PID controller created as a formula node. The formula node formats the temperature to a range between 0 and 5 before calculations are made. The output is in the range between 1 and 5. The formula node is set within a While loop, this makes it possible to change the variables in the PID-controller at runtime. The LabView code can be viewed in Figure 2

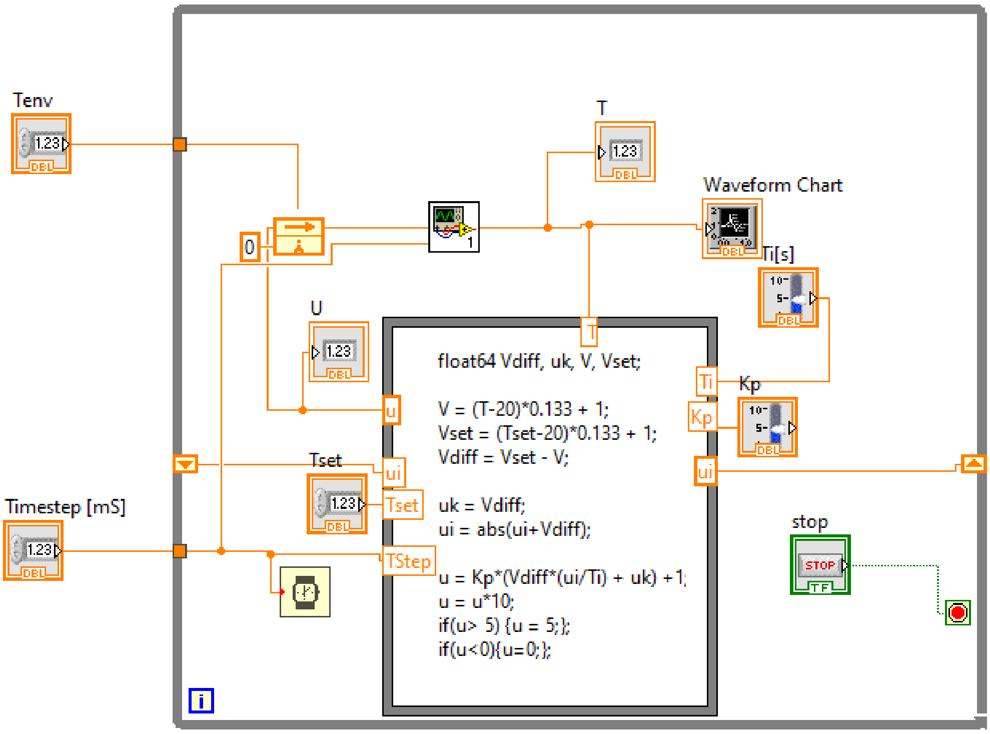

Figure 2: Structure of Block diagram with controller and Sub VI of temperature simulation.

```
float64 Vdiff, uk, V, Vset;

V = (T-20)*0.133 + 1;
Vset = (Tset-20)*0.133 + 1;
Vdiff = Vset - V;

uk = Vdiff;
ui = abs(ui+Vdiff);

u = Kp*(Vdiff*(ui/Ti) + uk) +1;
u = u*10;
if(u> 5) {u = 5;};
if(u<0){u=0;};
```

2.2.1 Tuning

To be able to tune the controller, the Ziegles Nichols method is used. To be able to use this method. The gain is increased until a barely stable oscillation is created. The input, only being from 0 to 5, is not enough to create maintained oscillations. The limit is therefore removed for the tuning. A gain of 4.3 is found to be the point where the systems oscillate. Additionally, the period for these oscillations are 40 seconds.

$Kp = 0.45 * 4.3$=1.9, Tp = 40s* 0.8 = 32s

The result from the PID tuning can be seen in Figure 3. Where a step from 25 to 35 is made to show the response.

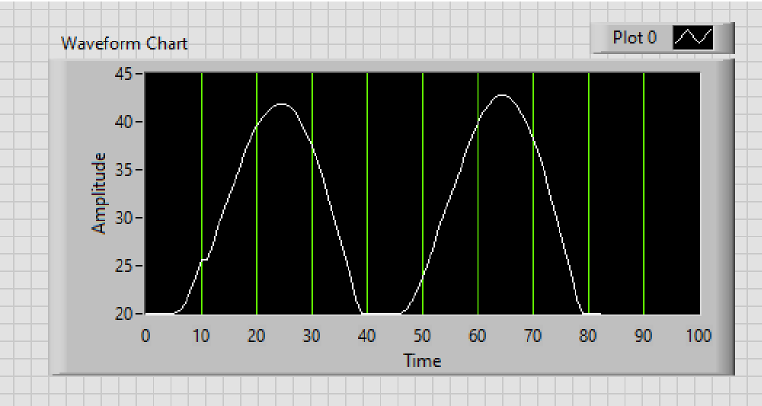

Figure 3: The sustained oscillations for the

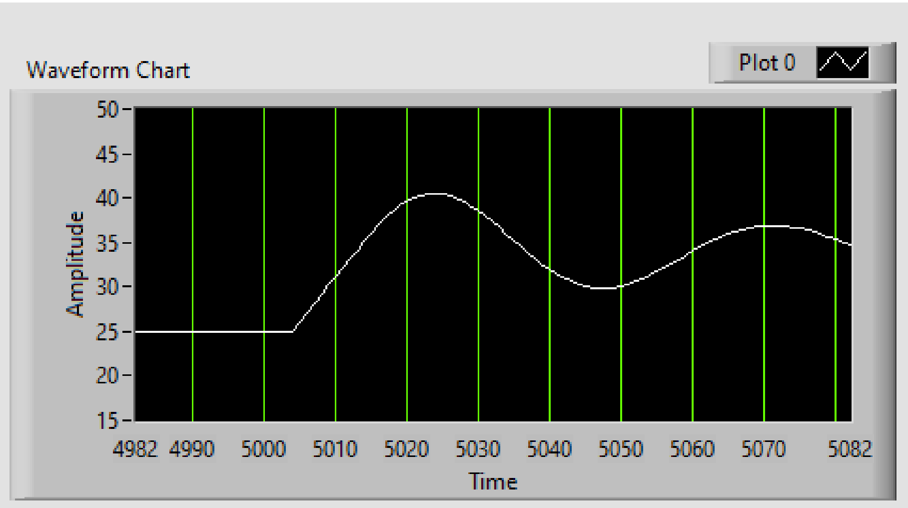

Figure 4: Result of PID controller after tuning.

3 HIL simulation and testing

3.1 Fuji PID

The same structure as for the purely simulated case is used with the FUJI controller. The Formula Node, and all the parameters surrounding it, is replaced by a FUJI PID controller. Additionally, the temperature is scaled from the output of the Temperature system before being input in the PID Controller. The layout can be shown in Figure 5, and the resulting plotted graph in Figure 6

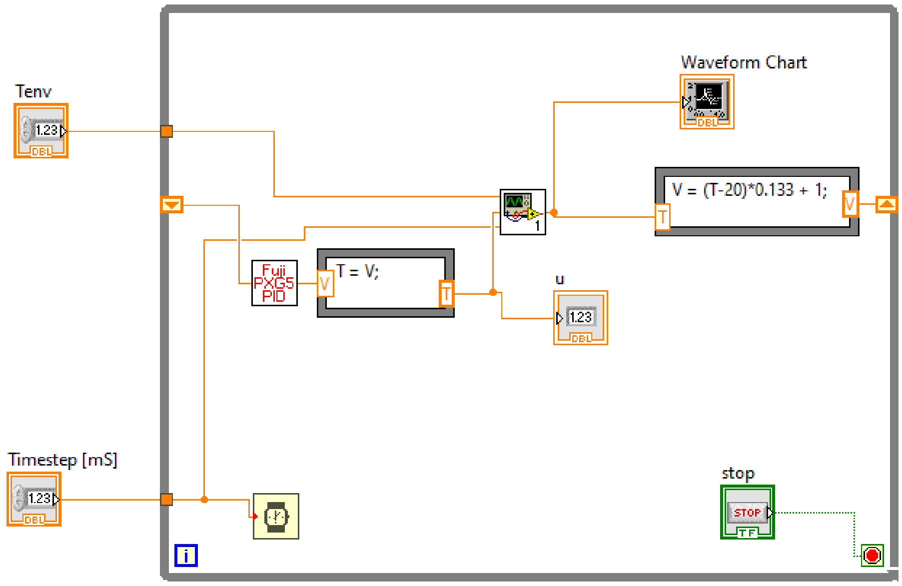

Figure 5: FUJI PID controller simulator used with simulated temperature system

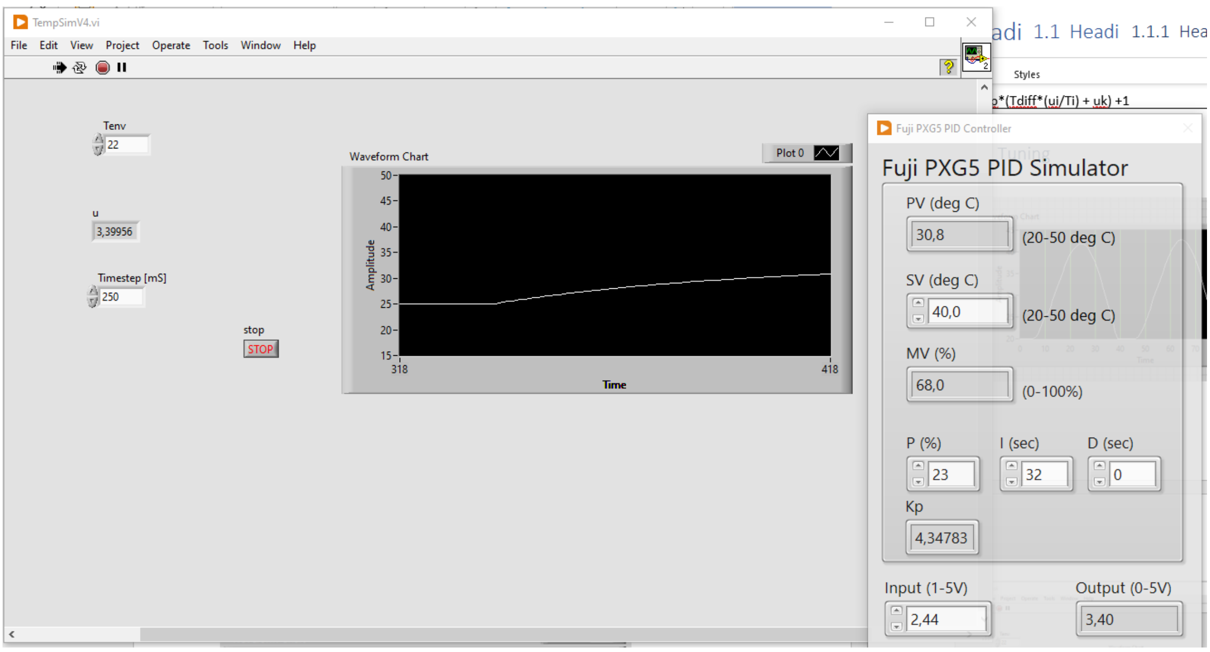

Figure 6: Result from FUJI PID controller.

4 Conclusion

In the final setup a PID controller is going to control the Air heater hardware. By creating a simulated system of the air heater, its possible to test the PID controller. With the simulator created and verified that is works. A Simulated PID controller is made, used to test the Air heater simulation. The PID controller highlights some of the challenges with the Air heater simulation. The time step affect the response of the system, this is a challenge when the time step of the real system is unknown. The scaling of values is hard to verify and could possibly be wrong. Without hardware to verify that parameters are set right, its hard to troubleshoot the setup.

The development of a Hardware in the loop system turned somewhat successful, with minor issues being resolved once proper hardware is present.

5 References

[1] H.-P. Halvorsen, "Hardware in the Loop Simulation and Testing," 30 03 2022. [Online]. Available: https://www.halvorsen.blog/documents/technology/hil/hil.php.

6 Appendix

- Time delay Sub VI from LabVIEW (techteach.no)
- Integrator Sub Vi from LabVIEW (techteach.no)