Assignment 2

Simulator of air heat controller

Implementation of the mathematical model

The implementation of the air heater in the simulation is to be based upon the following mathematical model (taken from *"Modeling, Simulation and Control"),*

$$\theta T'(t) = K_h[u(t-\tau)] + [T_{env}(t) - T(t)]$$

Solving with respect to the derivative of temperature,

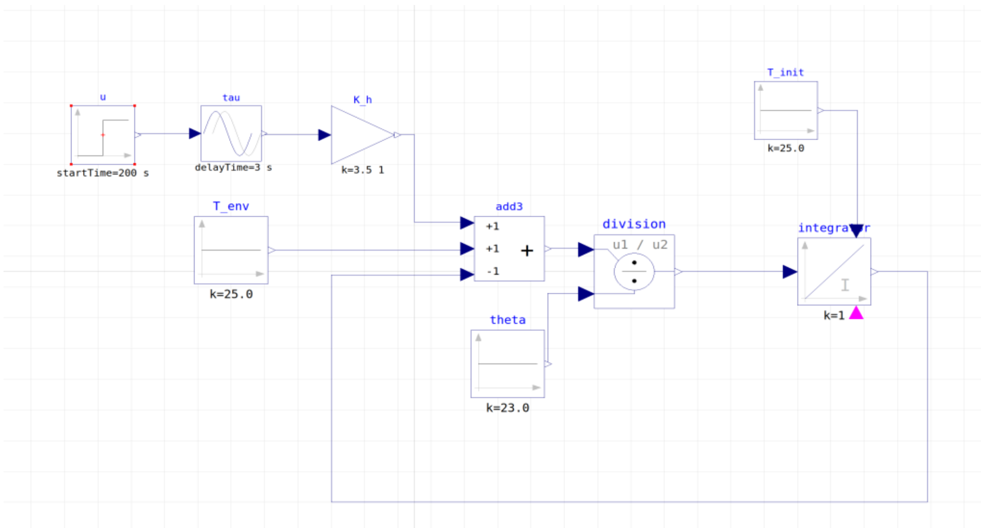
$$T'(t) = \frac{K_h[u(t-\tau)] + [T_{env}(t) - T(t)]}{\theta}$$

Integrating the differential equation gives,

$$T(t) = T_{init} + \int_0^t T'(\theta)\, d\theta$$

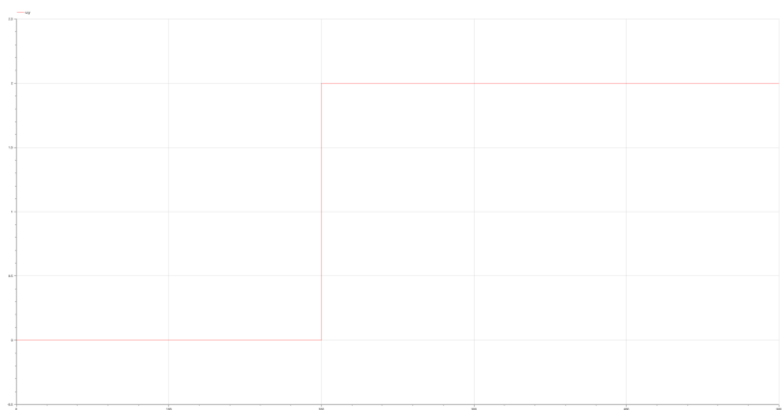
This mathematical model and its antiderivative can now be implemented in OpenModelica to simulate the air heat controller as shown in Fig.1.

Fig.1 (OpenModelica class for AirHeater)



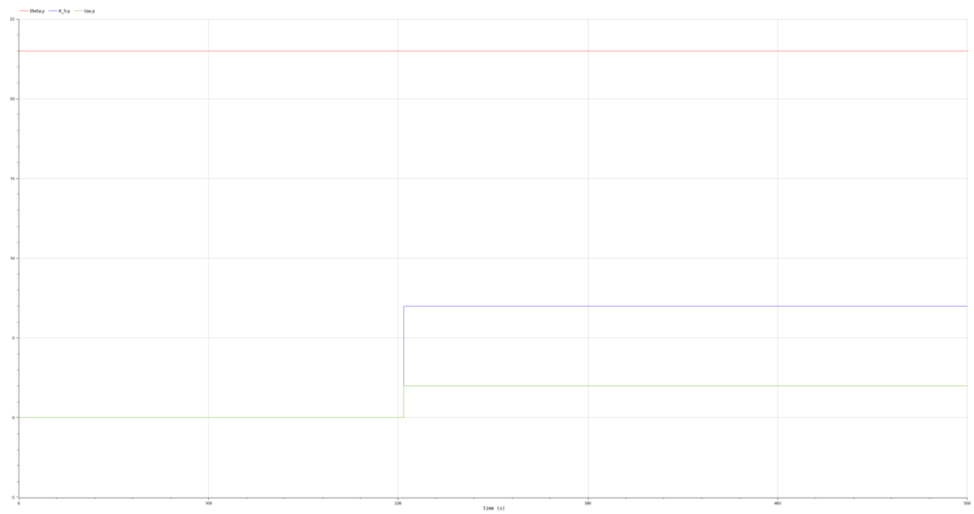
There is a step in the control signal as implemented in the AirHeater class, this can be read of in the plot shown in Fig.2.

Fig.2 (Control signal u with a step from 0 V to 2 V)



The gain, time constant and time delay can also be plotted, and their values correspond to those implemented in the AirHater class. This can be seen in Fig.3 below.

Fig.3 (Plot of gain, time constant and time delay)

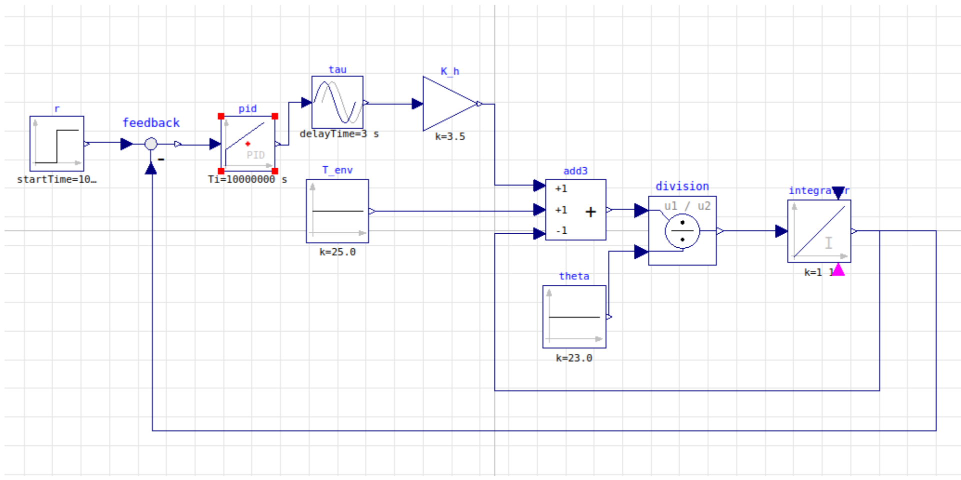


Since there the control signal is initially set to zero then the product of the gain and signal starts out at zero and is then increased with the step of the signal as expected. The time delay is also visible as the step does not occur exactly when the time is two hundred seconds but rather with a delay of three seconds.

Including a PID temperature controller

Using the same model as shown in Fig.1, but now with an included PID controller from the OpenModelica library. This controller is tuned to be a PI controller by setting the derivative part to zero. This model also has a feedback loop and a reference setpoint for measuring error control as shown in Fig.4.

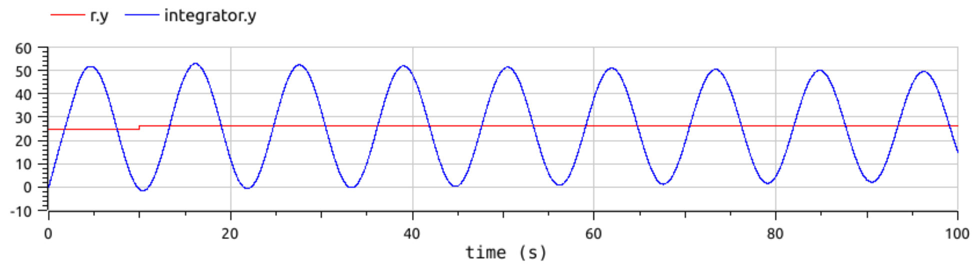
Fig.4 (AirHeater class with a PID controller and feedback loop)



Tuning procedure with relaxed Zieglers Nichols' method

To tune the PI controller, we set the integral part of the controller to zero, this can be done by making the time constant very large. This results in a P controller. Starting with a gain at zero and increasing it until there are sustained oscillations in the process measurement (see Fig.5).

Fig.5 (Sustained oscillation with)



Knowing the ultimate gain and the ultimate period which are, $K_{c,u} = 3.6$ and $P_u = 11$ respectively. It is now possible to apply the relaxed Zieglers Nichols' method and tune it as a PI-controller.

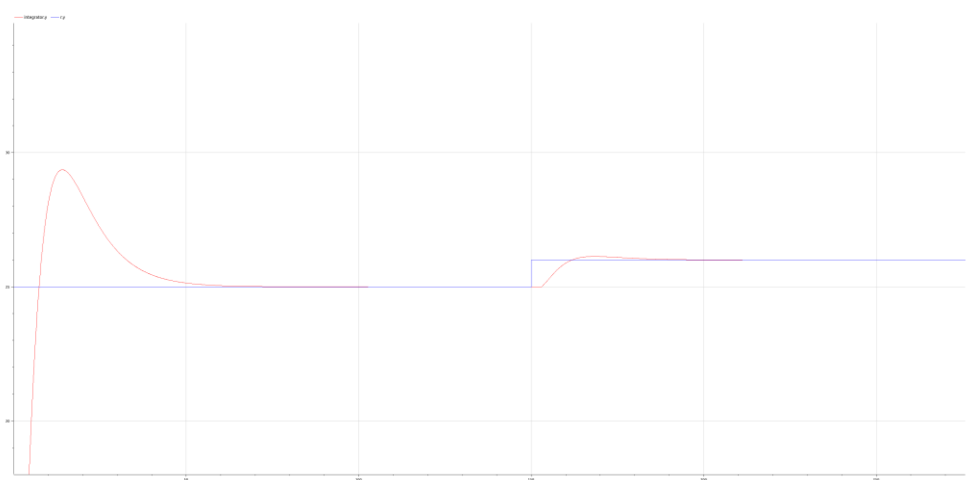
$$K_c = 0.25K_{c,u} = 0.25(3.6) = 0.9$$

and

$$T_i = 1.25P_u = 1.25(11) = 13.75$$

Using these values to tune the PI controller and make the system reach stability and stabilize with a step response as seen in Fig.6.

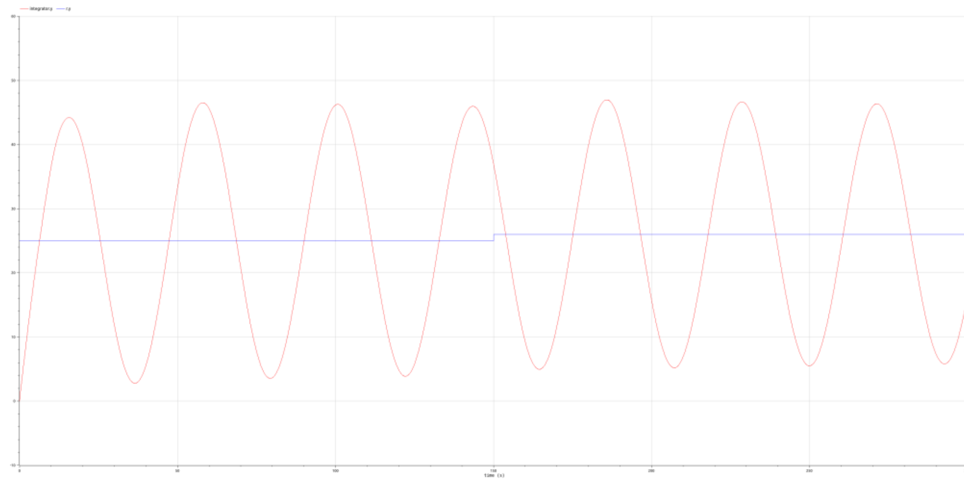
Fig.6 (Stabilized system with relaxed Zieglers Nichols' method)



As seen from the graph the process variable from the integrator reaches the setpoint and stabilizes, there is also a step in the setpoint from 25 to 26 degrees Celsius where the process variable stabilizes accordingly with a short delay.

Increasing the delay time to 9.5 seconds seems to give undamped oscillation as shown in Fig.7.

Fig.7 (Oscillations due to delay increase)



Making the control system marginally stable.