Experiment 9.

Task: Create a virtual experiment of a heat transfer process with the data given in Exp. 5. The experiment should use the DAC to output a voltage proportional to the temperature. It should also use the ADC to read a voltage proportional to the applied heat. The process should be controlled by a PI or PID, reading temperature data and supplying heating power through the ADC/DAC channels.

Objective: Learn about modeling, discretization, feedback control and PID tuning, anti-windup modification, and host-target communications through the serial port.

Comments: A simple heat balance equation is

$$mc_p\dot{T} + hA(T - T_{amb}) + \sigma FA(T^4 - T_{amb}^4) = q$$

where m = mass, cp = specific heat, h =convection heat transfer coefficient, A = area, σ = Boltzmann constant, F = view factor, Tamb = ambient temperature in K ($^oK = {}^oC+273$), q = supplied heat.
Some ball-park numbers are (in SI) m = 0.034, cp = 1000, h = 2, A = 0.0063, F = 1, σ = 5.675e-8. A reasonable supplied heat is in the order of several hundreds. These data correspond roughly to a heating element heated by electric power.

The continuous-time Simulink model for this process is supplied in exp5.mdl. For its real time simulation, a fixed-step solver should be used and a discrete integrator should replace the continuous time one. For the discretization, the sampling time should be small enough, in the order of seconds or fraction of seconds so that the difference between continuous and discrete simulation is small. This means that the sampling time should be small relative to the system time constant. To ensure the realistic behavior of the simulated system make sure that the supplied power is saturated between 0 and qmax (e.g. 1000).

Design a PI controller for the linearized system: $\tilde{T}(s) = P(s)\tilde{q}(s)$, where, $\tilde{T} = T - T_{ss}$, $\tilde{q} = q - q_{ss}$ and ss denotes the constant steady-state of linearization. To derive the linearized transfer function P(s), consider the Taylor expansion of the nonlinear terms, keeping the linear and constant parts:

$$mc_p(\dot{\tilde{T}} + \dot{T}_{ss}) + hA([\tilde{T} + T_{ss}] - T_{amb}) + \sigma FA([\tilde{T} + T_{ss}]^4 - T_{amb}^4) = \tilde{q} + q_{ss}$$

$$mc_p(\dot{\tilde{T}} + 0) + hA\tilde{T} + hA(T_{ss} - T_{amb}) + \sigma FA([T_{ss}]^4 + 4T_{ss}^{\ 3}\tilde{T} - T_{amb}^4) = \tilde{q} + q_{ss}$$

The last equality is approximate and is valid for small $\tilde{T}$. Next, regroup terms and define q_{ss} so that all the constants add up to zero. (This is the power required to maintain the steady-state.)

$$mc_p\dot{\tilde{T}} + (hA + 4\sigma FAT_{ss}^{\ 3})\tilde{T} = \tilde{q} + \{q_{ss} - hA(T_{ss} - T_{amb}) - \sigma FA([T_{ss}]^4 + T_{amb}^4)\}$$

Use 600 degC as the desired steady-state temperature. Design the PI controller to achieve a closed-loop time constant of 6 sec (Bandwidth 1/6 rad/min), approximately. Use a sampling time of 0.5 sec. Implement the controller in SIMULINK by using a limited integrator so that you avoid integrator windup effects. Also, try a controller implementation with a regular integrator to visualize the integrator windup.

Use a square wave reference input, alternating between 550-650^{o}C, with period 4min. Include a serial block in the controller PC104 to report the values of the reference, control input, and output back to the host computer using the serial port, as in Exp. 3. Use the MATLAB plot command to plot the system response in the host computer.