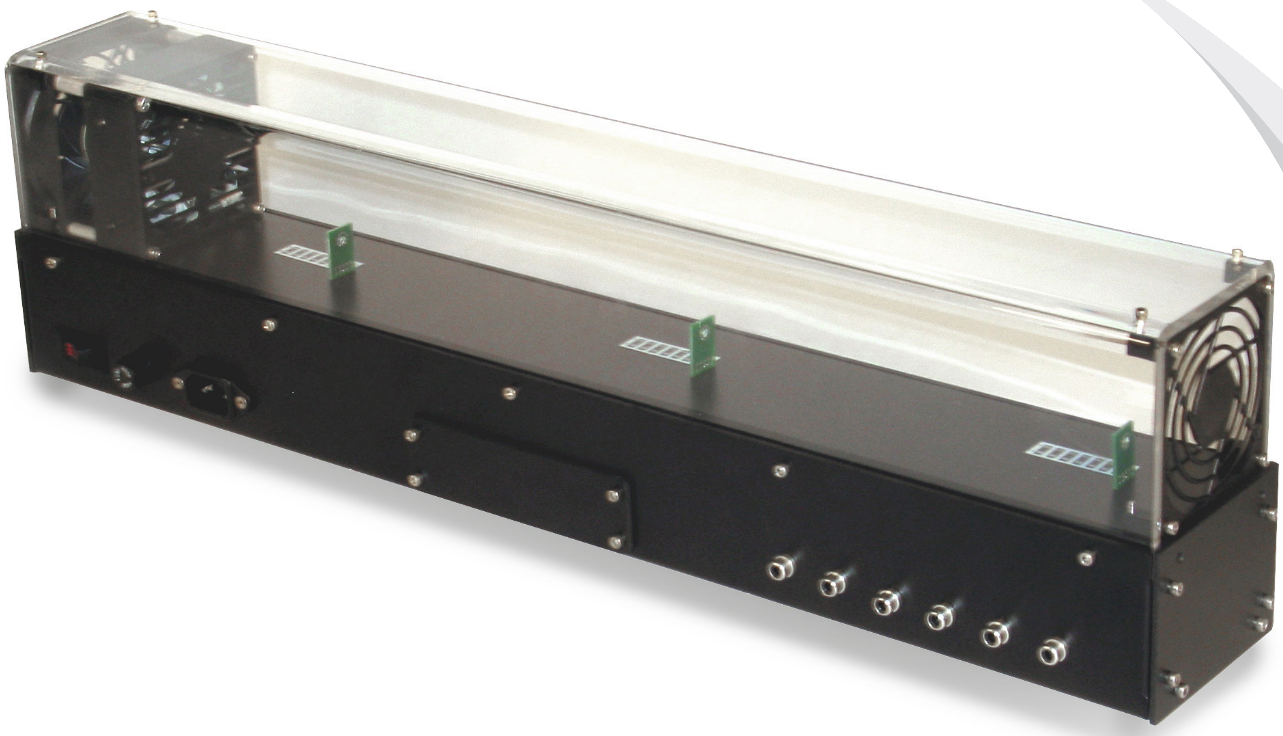
QUANSER
INNOVATE · EDUCATE

STUDENT WORKBOOK

Heat Flow Experiment for MATLAB®/Simulink® Users

Developed by:
Jacob Apkarian, Ph.D., Quanser
Hervé Lacheray, M.A.Sc., Quanser
Michel Lévis, M.A.Sc., Quanser



CAPTIVATE. MOTIVATE. GRADUATE.

Table of Contents

1. Introduction

The Quanser Heat Flow Experiment (HFE) is the process control plant shown in Figure 1. This system consist of a chamber equipped with three temperature sensors located equidistantly. There is a coil-based heater and a blower at one end chamber that is used to transfer heat conductively. The HFE has a built-in amplifier to deliver power to the heater and blower and the amount of power is controlled using analog signals. There is also a tachometer mounted on the blower to measure the speed of the fan.

The purpose of the experiment is to model the system using the "bumptest" method. This is a simple technique used to find the first-order model of the system (for a particular temperature sensor). The second main topic is temperature control. The temperature of the chamber is regulated by using an on-off control and a PID-based compensator. The on-off control is a more basic approach that does not require any knowledge of the plant. The PID algorithm is designed using the model previously found, and can obtain better performance. More practical process control related concepts are discussed, such as set-point point and integral anti-windup.

Figure 1: Quanser Heat Flow experiment.

The following topics are covered in this laboratory:

- Derive a first-order transfer function model of the system using the Bumptest method.
- Implement an on-off scheme using a relay switch to control the temperature.
- Design a Proportional-Integral controller to regulate the temperature using process control related strategies such as set-point weight and anti-windup.

2. Prerequisites

In order to successfully carry out this laboratory, the user should be familiar with the following:

- Wiring and operating procedure of the Heat Flow plant with the data acquisition device, as discussed in Heatflow User Manual.
- Transfer functions.
- Designing a PID controller.

3. Overview of Files

Table 1 below lists and describes the various files supplied with the experiment.

File Name	***Description***
Heat Flow Laboratory – Student Manual.pdf	This manuals goes through some pre-lab and in-lab exercises to model and control the Quanser Heat Flow device.
Heat Flow – User Manual	Describes the Heat Flow hardware, gives its specifications, and explains how to setup and wire the system to perform labs.
setup_hfe.m	The main Matlab script that sets the sensor calibration gains of the Heat Flow. **Run this file only to setup the laboratory.**
hfe_lib.mdl	Heat Flow Simulink library that has the *Heat Flow* subsystem. This subsystem contains QUARC block that interface to the Heat Flow hardware, i.e. heater and blower and temperature sensors.
q_hfe_open_loop.mdl	Simulink file that can feed open-loop voltages to the heater and blower and measure the temperatures using QUARC.
q_hfe_on_off.mdl	Simulink file that implements the on-off controller on the Heat Flow system using QUARC.
q_hfe_pi.mdl	Simulink file that implements the PI controller on the Heat Flow system using QUARC.

Table 1: Files supplied with the Heat Flow experiment.

4. Pre-Lab

Section 4.1 explains how to model the Heat Flow plant. In Section 4.2, the percentage overshoot, steady-state error, and peak time desired response is discussed. Two controllers to regulate the temperature in the chamber are then investigated in Section 4.3: the on-off control and the proportional-integral compensator.

4.1. Modeling

4.1.1. Dynamics of the Heat Flow Experiment

The complete thermodynamic model of the Quanser Heat Flow system is a challenge beyond the scope of this manual. The block diagram shown in Figure 2 depicts the inputs and outputs of the system. The blower voltage, V_b, and heater voltage, V_h, applied affects how the chamber temperature changes with respect to the ambient (i.e. room) temperature, denoted by variable T_a.

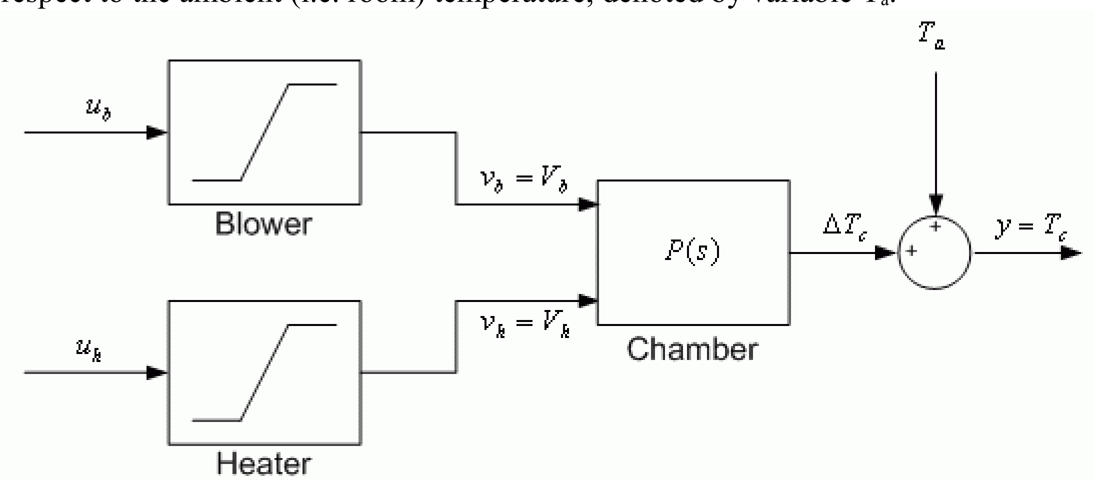


Figure 2: Dynamics of Heat Flow system.

The temperature of the chamber at sensor *n* can be described by

$$\frac{d}{dt} T_n(t) = f(V_h, V_b, T_a, x_n) \quad [1]$$

,

where x_n is the distance between sensor *n* and the heater. For the purposes of designing a temperature controller, a simple first-order transfer function model of the system will suffice. Consider the transfer function

$$T_n(s) = \frac{K_n V_h(s)}{\tau_n s + 1} \quad [2]$$

,

where for temperature sensor *n,* K_n is the steady-state gain and τ_n is the time constant.

4.1.2. Bumptest

The bumptest is a simple test based on a step response for a stable system. It is carried out in the following way. A constant input is chosen. A stable system will then reach an equilibrium. The input is then changed rapidly to a new level and the output is recorded.

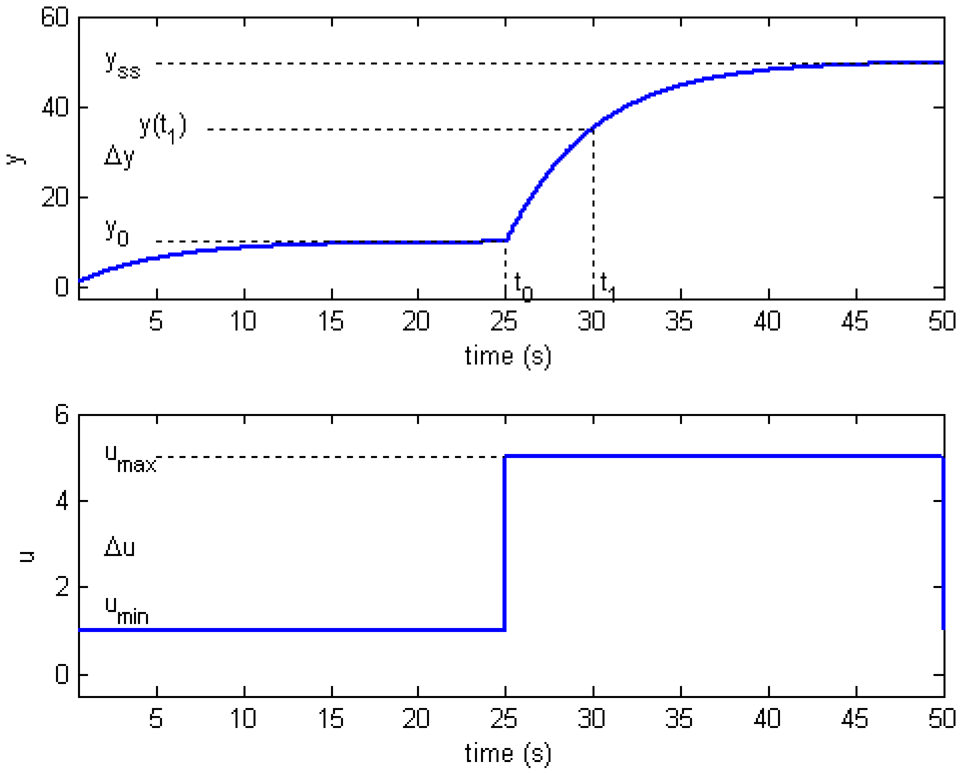


Figure 3: Input and output signal used in the bumptest method.

The step response shown in Figure 3 is generated using the transfer function

$$\frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1} \quad [3]$$

with the parameters

$$K = 10.0 \quad [4]$$

and

$$\tau = 5.0 \, [s] \quad [5]$$

The input signal, u, is a step that begins at time t_0. The input signal has a minimum value of u_{min} and a maximum value of u_{max}. The resulting output signal is initially at y_0. Once the step is engaged, the output eventually settles to its steady-state value y_{ss}. From the output and input signals, the steady-state gain is

$$K = \frac{\Delta y}{\Delta u} \quad [6]$$

where

$$\Delta y = y_{ss} - y_0 \quad [7]$$

and

$$\Delta u = u_{max} - u_{min} \quad [8]$$

In order to find the model time constant, τ, the output signal at $y(t_1)$ must be measured. It is defined

$$y(t_1) = (1 - \mathbf{e}^{(-1)})\,\Delta y + y_0 \quad [9]$$

and the time is

$$t_1 = t_0 + \tau \quad [10]$$

From this, the model time constant is

$$\tau = t_1 - t_0 \quad [11]$$

4.2. Desired Response

4.2.1. Second-Order System

The block diagram shown in Figure 4 is a general unity feedback system with a compensator C(s) and a transfer function representing the plant, P(s). The measured output, Y(s), is supposed to track the reference signal, R(s), and the tracking has to satisfy certain specifications.

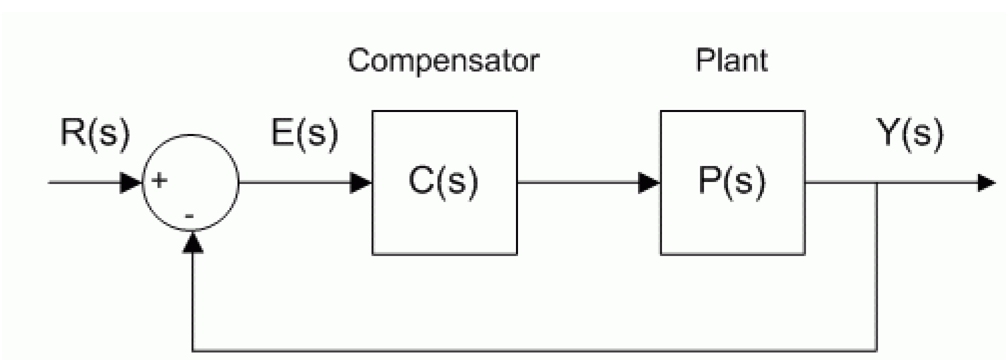


Figure 4: Unity feedback system.

The output of the system shown in Figure 4 is

$$Y(s) = C(s)\,P(s)\,(R(s) - Y(s)) \quad [1]$$

and by solving for Y(s) the resulting closed-loop transfer function

$$\mathrm{Y}(s) = \frac{\mathrm{C}(s)\,\mathrm{P}(s)\,\mathrm{R}(s)}{1 + \mathrm{C}(s)\,\mathrm{P}(s)} \qquad [2]$$

is obtained. Recall that the HFE voltage-to-temperature transfer function

$$\mathrm{P}(s) = \frac{K}{\tau\, s + 1} \qquad [3]$$

.

As will be seen in the control design section, using a proportional-integral (PI) compensator with a first-order plant, such as in [3], the closed-loop transfer function in [2] has the form

$$\frac{\mathrm{Y}(s)}{\mathrm{R}(s)} = \frac{\omega_n^2}{s^2 + 2\,\zeta\,\omega_n\, s + \omega_n^2} \qquad [4]$$

where ω_n is the natural undamped frequency and ζ is the damping ratio. This is called the *standard second-order* transfer function and the properties of its response depend on the values of ω_n and ζ parameters.

4.2.2. Peak Time and Overshoot

Consider when a second-order system as shown in Equation [4] is subjected to a reference step

$$\mathrm{R}(s) = \frac{R_0}{s} \qquad [5]$$

with an step amplitude of $R_0 = 1.5$. The obtained response is shown in Figure 5, below, where the red trace is the output response, y(t), and the blue trace is the reference step, r(t).

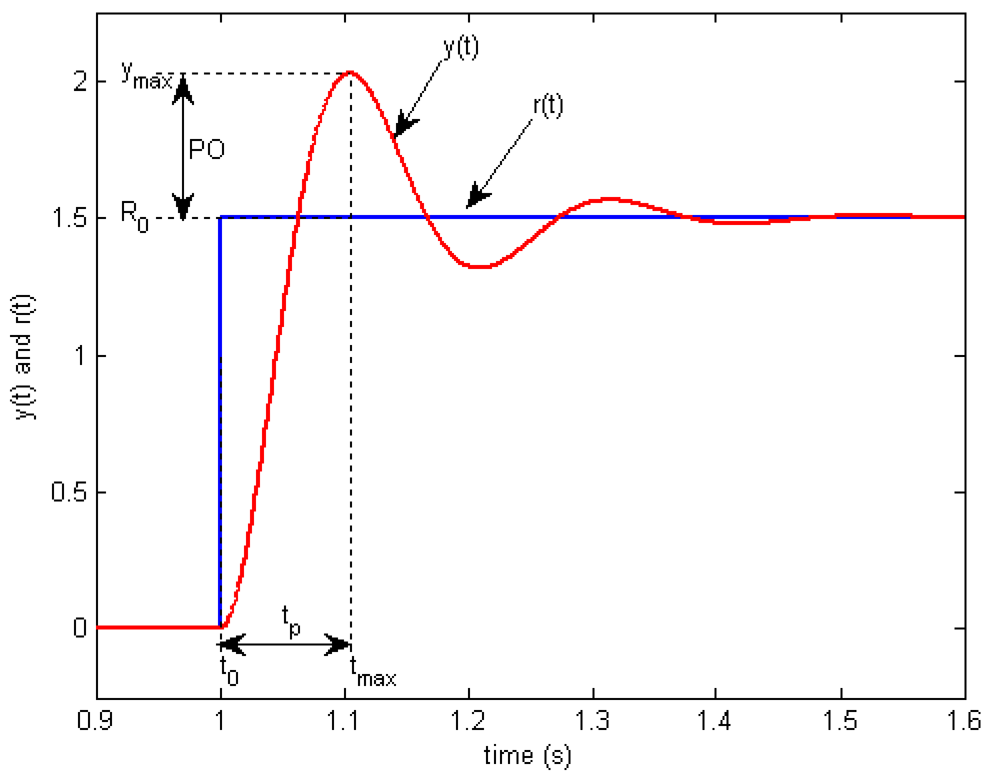


Figure 5: Standard second-order step response.

The maximum value of the response is denoted by the variable y_{max} and it occurs at a time t_{max}. For a response similar to Figure 5, the *percentage overshoot* is found using the equation

$$PO = \frac{100\,(y_{max} - R_0)}{R_0} \quad [6]$$

From the initial step time, t_0, the time it takes for the response to reach its maximum value is

$$t_p = t_{max} - t_0 \quad [7]$$

This is called the *peak time* of the system.

In a second-order system, the amount of overshoot depends solely on the damping ratio parameter and it can be calculated using the equation

$$PO = 100\,\mathbf{e}^{\left(-\frac{\pi\,\zeta}{\sqrt{1-\zeta^2}}\right)} \quad [8]$$

The peak time depends on both the damping ratio and natural frequency of the system and it can be derived that the relationship between them is

$$t_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \quad [9]$$

Generally speaking then, the damping ratio affects the shape of the response while the natural frequency affects the speed of the response.

4.2.3. Steady-State Error

Steady-state error is the difference between the reference and output signals after the system response has settled. Thus for a time *t* when the system is in steady-state, the steady-state error equals

$$e_{ss} = r_{ss}(t) - y_{ss}(t) \quad [10]$$

where $r_{ss}(t)$ is the value of the steady-state reference and y_{ss} is the steady-state value of the process output.

Assume the compensator is .

1. Find the error transfer function E(s), shown in Figure 4, in terms of the reference R(s), the plant P(s), and the compensator C(s).

0	1	2

2. Recall the final-value theorem

$$e_{ss} = \lim_{s \to 0} s\,E(s) \quad [11]$$

Given that the compensator equals

$$C(s) = k_p \quad [12]$$

the reference is the step defined in [5], and the process model is the transfer function in [3], find the steady-state error of the system.

0	1	2

3. Based on the steady-state error result from a step response with a proportional gain, what *Type* of system is the Heat Flow when performing temperature control (0, 1, 2, or 3)?

0	1	2

4.2.4. Heat Flow Specifications

The time-domain specifications for controlling the temperature of the HFE are:

$$e_{ss} = 0 ,\tag{13}$$

$$t_p = 15.0\ [s] \text{, and}\tag{14}$$

$$PO = 15.0 .\tag{15}$$

Thus when tracking the desired temperature, the transient response should have a peak time less than or equal to 15.0 seconds, an overshoot less than or equal to 15 %, and the steady-state response should have no error.

Calculate what the maximum peak of the response, in degrees Celsius, when the desired temperature is

$$T_d(t) = T_{d,\,off} + R_0\, u_{step}(t - t_0) .\tag{16}$$

The step function is defined as

$$u_{step}(\tau) = \begin{cases} 1 & 0 \le \tau \\ -1 & \tau < 0 \end{cases} ,\tag{17}$$

the offset temperature is

$$T_{d,\,off} = 47.5\ [degC] ,\tag{18}$$

and the amplitude of the step is

$$R_0 = 5.0\ [degC] .\tag{19}$$

Thus when the step goes positive to 1, the desired temperature will be at 50.0 °C.

0	1	2

4.3. Temperature Control

4.3.1. On-Off Control

An on-off control can be implemented using a variety of switches and its design requires no model of the plant. The block diagram shown in Figure 6 shows the on-off controller that will be used with the Heat Flow. The on-off control is implemented using a relay switch. Also shown in the diagram, is the nonlinear dynamics of the heater actuator, which is limited between 0 V and 5.0 V.

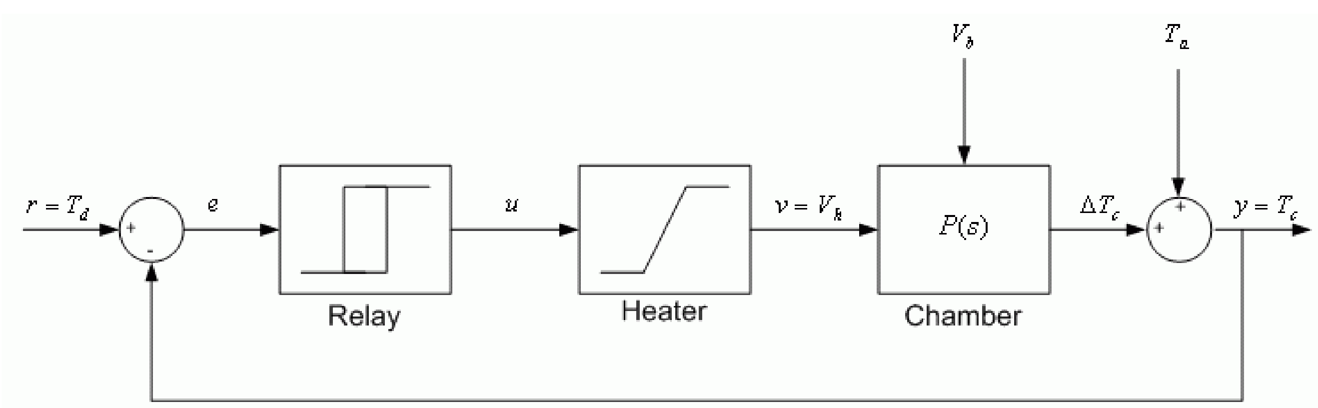


Figure 6: On-off control with relay.

The relay characteristics is illustrated in Figure 7. When on, the relay switch outputs a voltage of $V_{h,on}$ to the heater. When off, its outputs a value of $V_{h,off}$ to the heater (typically this is set to 0 V). The hysteresis width, ΔT_h, can be adjusted according to the desired goal. Changing the width of this hysteresis affects the performance of the control, as will be investigated in the laboratory.

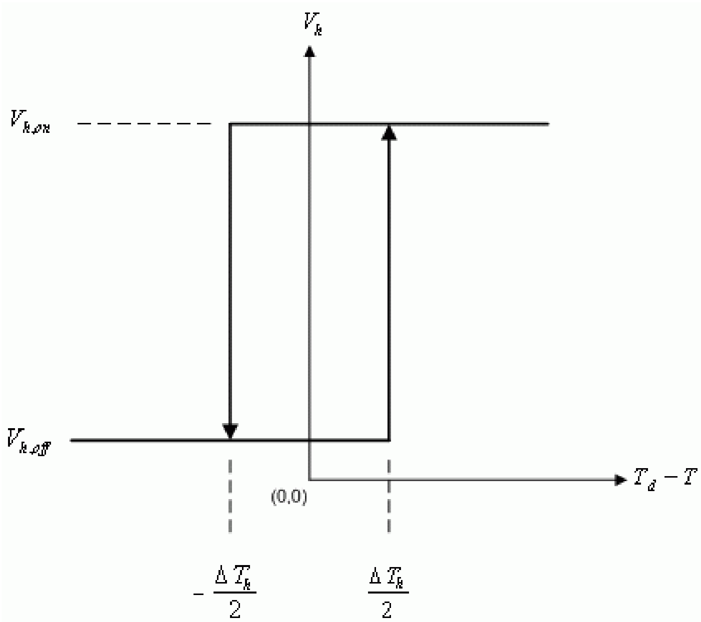


Figure 7: Relay switch.

4.3.2. Proportional-Integral Control

4.3.2.1. Closed-Loop Transfer Function

The proportional-integral (PI) compensator used to control the temperature of the Heat Flow has the structure

$$V_h(t) = k_p\,(b_{sp}\,T_d(t) - \mathrm{T}(t)) + k_i \int T_d(t) - \mathrm{T}(t)\,dt \tag{20}$$

where k_p is the proportional control gain, b_{sp} is the set-point weight, k_i is the integral control gain, $T_d(t)$ is the desired or setpoint temperature, T(t) is the measured chamber temperature at a particular sensor, and $V_h(t)$ is the heater control voltage. The block diagram of the PI control is illustrated in Figure 8.

Remark: The temperature is controlled about a single sensor – either 1, 2, or 3. From this point on, the T_n notation for measured temperature at sensor *n* will be dropped. The *T* denotes the temperature measured at a particular sensor. Typically this will be sensor 1 but it can be chosen for any.

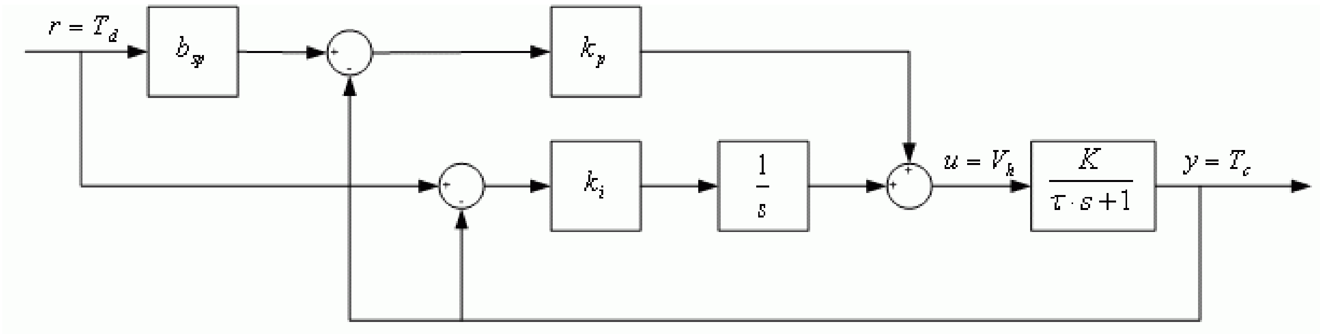


Figure 8: Heat Flow PI controller.

1. Find the closed-loop temperature control transfer function, $T(s)/T_d(s)$, using the time-domain PI control in Equation [20], the block diagram in Figure 8, and the process model in [3]. Assume the zero initial conditions, thus $T(0^-) = 0$.

0	1	2

4.3.2.2. Find PI Gains

1. The resulting PI Heat Flow closed-loop transfer function has the same structure as the *standard second-order system* given in [4]. The denominator of [4] is called the *standard characteristic equation,*

$$s^2 + 2\,\zeta\,\omega_n\,s + \omega_n^2 \quad [21]$$

Find the control gains k_p and k_i that map the characteristic equation of the HFE closed-loop system to the standard characteristic equation given above. With these two equations, the control gains can be designed based on a specified natural frequency, ω_n, and damping ratio, ζ.

0	1	2

2. Using the equations described in Section 4.2.2, express the natural frequency and the damping ratio in terms of percentage overshoot and peak time specifications.

0	1	2

3. Calculate the minimum damping ratio and natural frequency required to meet the specifications given in Section 4.2.4.

0	1	2

The PI gains will be computed in an in-lab exercise once the model parameters have been found.

4.3.2.3. Steady-State Error

1. Find the error transfer function of the closed-loop Heat Flow system when using the PI compensator.

0	1	2

2. Evaluate the steady-state error. Is there any benefit of adding integral action?

0	1	2

4.3.2.4. Anti-Windup

The heater actuator on the Heat Flow can be modeled by the function

$$\mathrm{u}(t) = \begin{cases} 5 & 5 < v \\ v & -v \le 0 \textbf{ and } -5 + v \le 0 \\ 0 & v < 0 \end{cases} \quad [22]$$

where *v(t)* is the ideal control voltage and *u(t)* is the actual voltage applied. As described, the heater operates linearly in the 0-5.0 V range but saturates when it goes below 0 V or beyond 5.0 V.

This non linearity can cause issues when used with a controller that has integral action. If the controller saturates, the integrator may begin to drift and effectively break the feedback loop. This is called integrator wind up.

An integrator anti-windup scheme such as the one shown in Figure 9 can be used to mitigate this issue. If the control signal does not saturate, then the extra feedback loop with the time constant T_r is inactive because $u = v$. When the controller output saturates, the extra feedback loop drives the saturation error, e_s, to zero and causes the integrator to output a value just at the saturation limit. This means that the control signal will decrease from the saturation limit as soon as the control error goes negative.

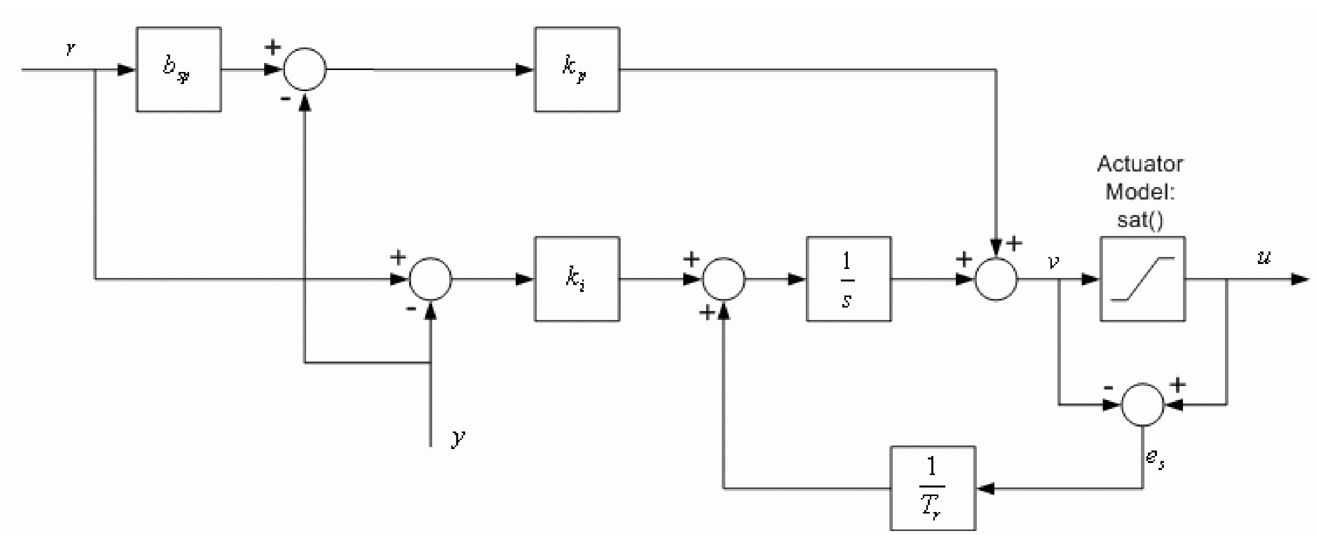


Figure 9: PI with anti-windup scheme.

The windup protection is governed by the integrator reset time, T_r. There is less protection against windup if T_r is made large (e.g. none if $T_r = \infty$). If $T_r = 1$ second, then the integrator is reset in one sampling period.

In Figure 10, the effect of using an anti-windup scheme is shown. The dashed blue line represents the response without windup protection and the solid red line is the response with anti-windup. When wind up occurs, the integrator builds up a lot of energy and causes there to be a larger overshoot in the response. However, with anti-windup the input to the integrator is decreased when the controller saturates and the overshoot is decreased significantly. Anti-windup becomes especially important in slow systems with large time constants, such as the Heat Flow, because the integrator has more time to wind up.

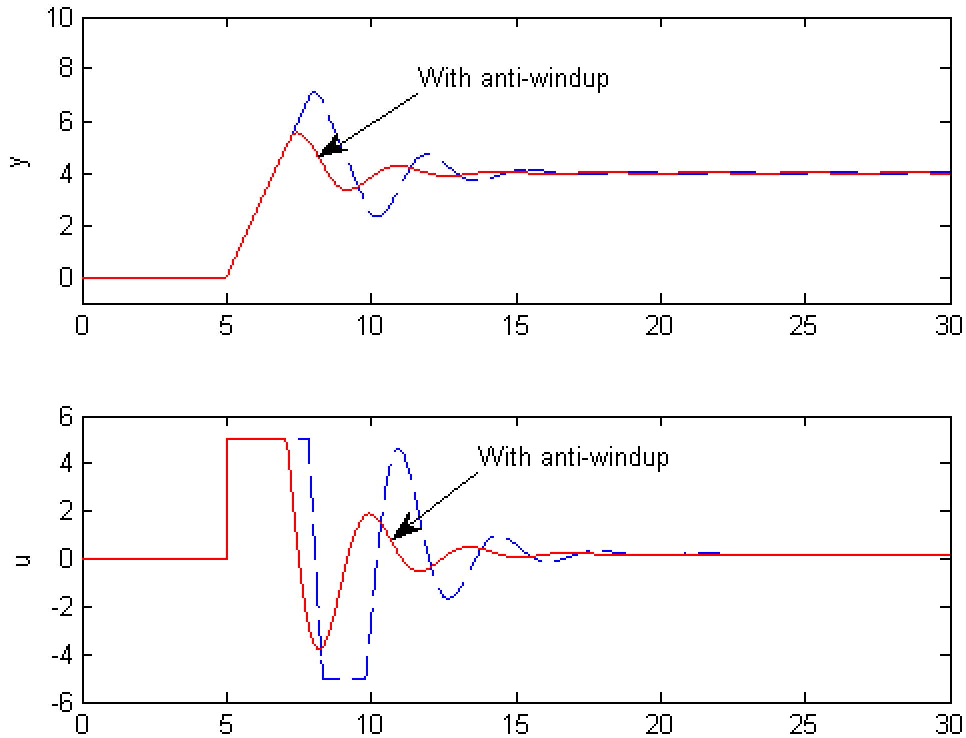


Figure 10: Effect of using integrator anti-windup on response.

5. In-Lab Procedures

5.1. Bumptest Modeling

The *q_hfe_open_loop* Simulink diagram shown in Figure 11 is used to feed open-loop voltages to the heater and blower and measure the corresponding chamber temperatures and fan speed. The *Heatflow* subsystem contains QUARC blocks that interface with the Heat Flow hardware. The *To Host File* block is used to save the temperature sensors readings and the blower and heater input voltages to a Matlab MAT file. The experimental data saved in this file can then be accessed in Matlab to plot the response and take measurements.

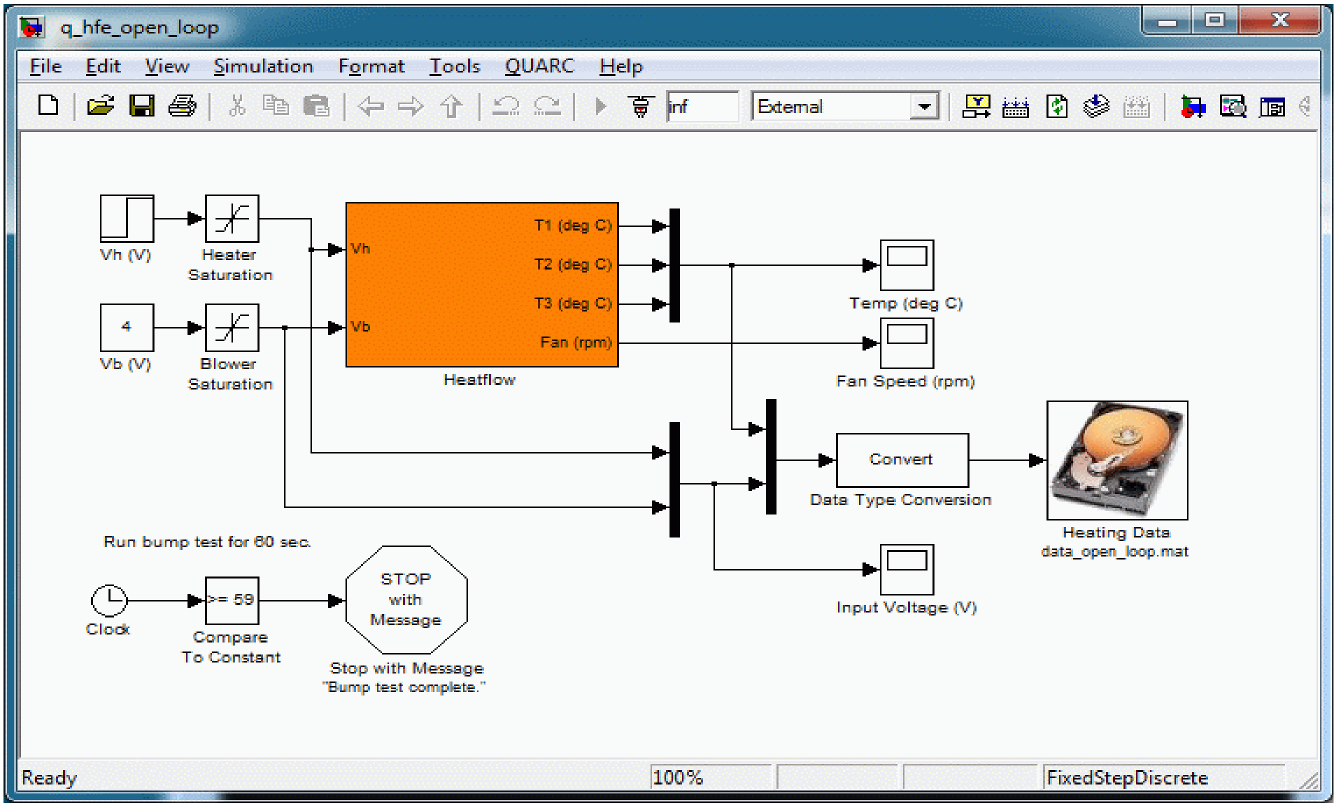


Figure 11: Simulink diagram used to run HFE in open-loop using QUARC.

5.1.1. Getting Started

Follow this procedure to run the open-loop controller on the Heat Flow device:

1. Load the Matlab software.
2. Browse through the *Current Directory* window in Matlab and find the folder that contains the lab files, e.g. *q_hfe_open_loop.mdl*.
3. Double-click on the *q_hfe_open_loop.mdl* file to open the Simulink diagram shown in Figure 11.
4. Go to the *Current Directory* window and open the *setup_hfe.m* file.
5. Run the script by selecting the Debug | Run item from the menu bar or clicking on the *Run* button in the tool bar. This loads the appropriate sensor calibration gains in the Matlab workspace.
6. The *Heatflow* subsystem in this model is linked to a Simulink library called *hfe_lib.mdl*. To access the library, right-click on the *Heatflow* subsystem and select *Link Options* | *Go To Library Block*. Double-click on the block to open it. The interior of the Heatflow subsystem is shown in Figure 12, below.

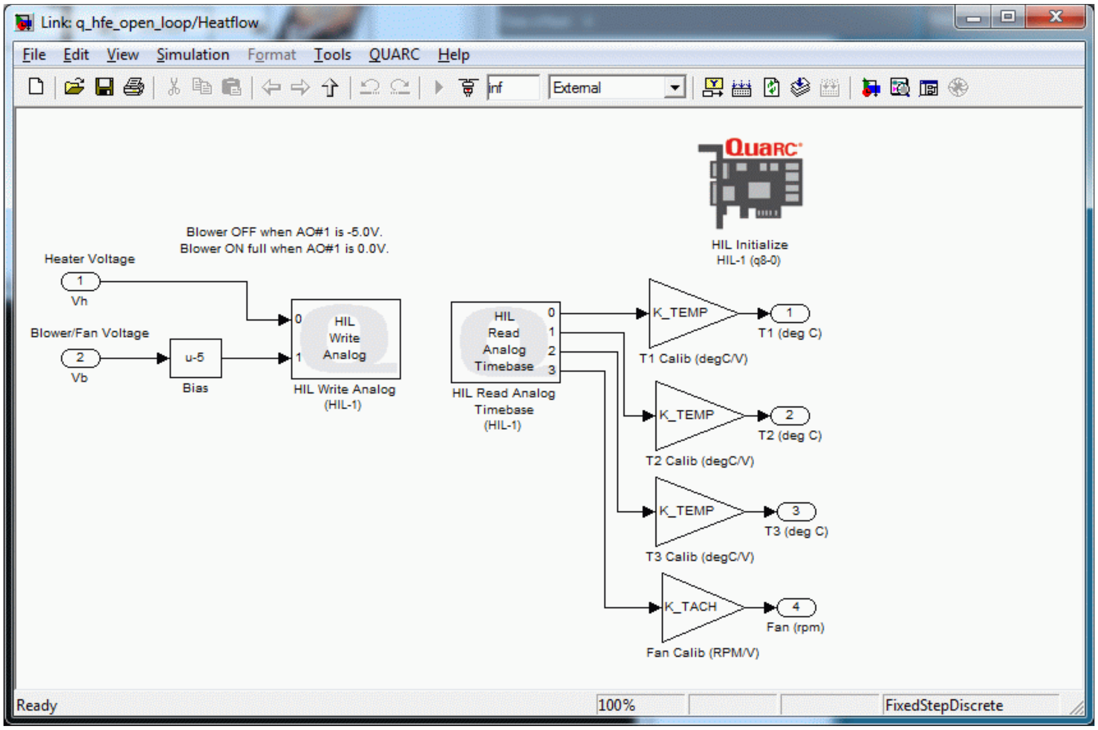


Figure 12: Heatflow subsystem.

7. **Configure DAQ**: Double-click on the HIL Initialize block inside the *Heatflow* subsystem (from *hfe_lib*) and ensure it is configured for the DAQ device that is installed in your system. By default the block is setup for the Quanser Q8 hardware-in-the-loop board. Before making changes, make sure you **unlock** the Simulink library by clicking on *Edit* | *Unlock Library* in the *hfe_lib* menu bar.
8. Save any changes and close *hfe_lib*. All the Heat Flow QUARC controllers are linked to this block so any changes only have to be applied once.
9. Ensure all the connections have been made as instructed Reference Heatflow User Manual.
10. Power on the Heat Flow.
    Remark: The blower should start when the power is engaged.
11. You are now ready to build and run the controller.

5.1.2. Running the Bumptest

This procedure demonstrates how to perform the bumptest with the heater:

1. Setup the Matlab workspace and the HFE Library as discussed in Section 5.1.1.
2. Click on *Quarc* | *Build* to compile the *q_hfe_open_loop* Simulink diagram.
3. Go to Quarc | Start to begin running the controller.
4. A 4V step is applied to the heater 5 seconds after the control begins running.
5. Make sure Vb = 4 in the Simulink diagram.
6. The scopes should be displaying responses similar to figures 13, 14, and 15. When the heater step is applied the temperature in the chamber begins to increase.

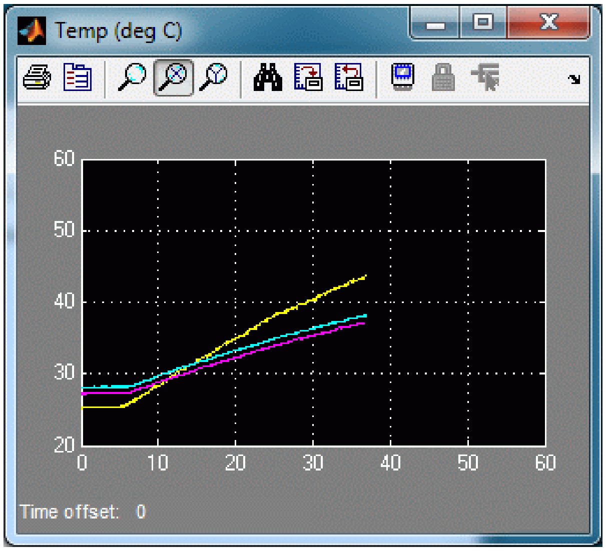


Figure 13: Temperature bumptest response. T1 is yellow, T2 is purple, and T3 is cyan.

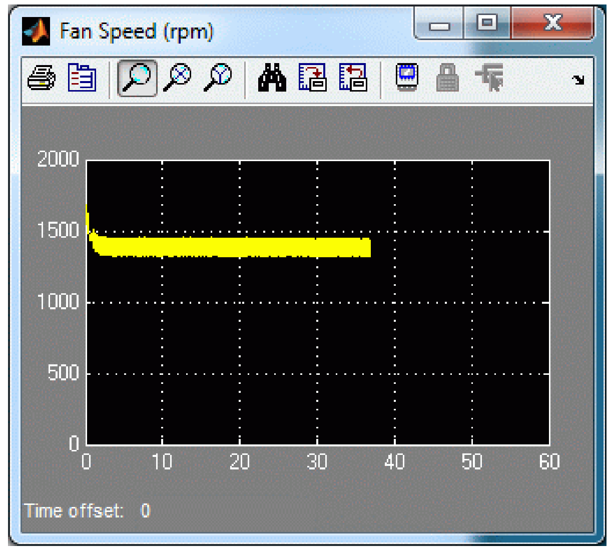


Figure 14: Fan speed in blower.

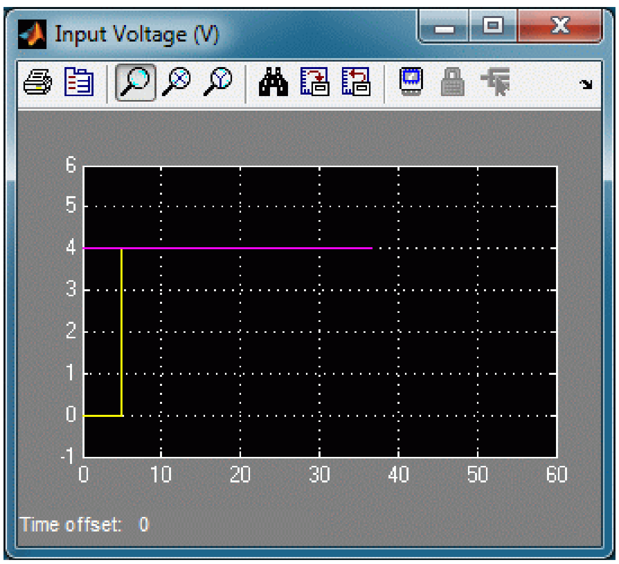


Figure 15: Heater (yellow) and blower (purple) voltage.

7. The controller is automatically stopped after 59 seconds.
8. The temperature sensor readings and blower and heater voltages are saved in the MAT file *data_open_loop.mat* and it can be be accessed using the Matlab commands shown in Text 1, below.

```
% load raw data from MAT file
load('data_open_loop.mat');
% save into variables
t = data_hfe_ol(1,:);
T1 = data_hfe_ol(2,:);
T2 = data_hfe_ol(3,:);
T3 = data_hfe_ol(4,:);
Vh = data_hfe_ol(5,:);
Vb = data_hfe_ol(6,:);
```

Text 1: Loading data from data_open_loop.mat.

9. Does the temperature in different locations along the duct go up at the same rate? Explain.

0	1	2

10. Using the commands shown in Text 1 and Matlab plotting commands, create a plot that shows the temperature response at the T_1 sensor as well as the blower and heater voltages used. Attach this to your report.

0	1	2

11. The Heat Flow step response resembles a first-order transfer function. Using the bumptest method discussed in Section 4.1.2, find the steady-state gain observed at the T_1 sensor.

0	**1**	**2**

12. Find the time constant from the obtained response.

0	1	2

13. Turn off the Heat Flow device if no more experiments will be conducted in this session.

Other things to try...

- Modeling using a random input (e.g. using the Matlab System Identification toolbox).

5.2. On-Off Temperature Control

The *q_hfe_on_off* Simulink model shown in Figure 16, below, is used to run the on-off controller discussed in Section 4.3.1 on the Heat Flow device. The *Heatflow* subsystem is linked to the Heat Flow library, *hfe_lib*, and the on-off controller is implemented using the Simulink *Relay* block.

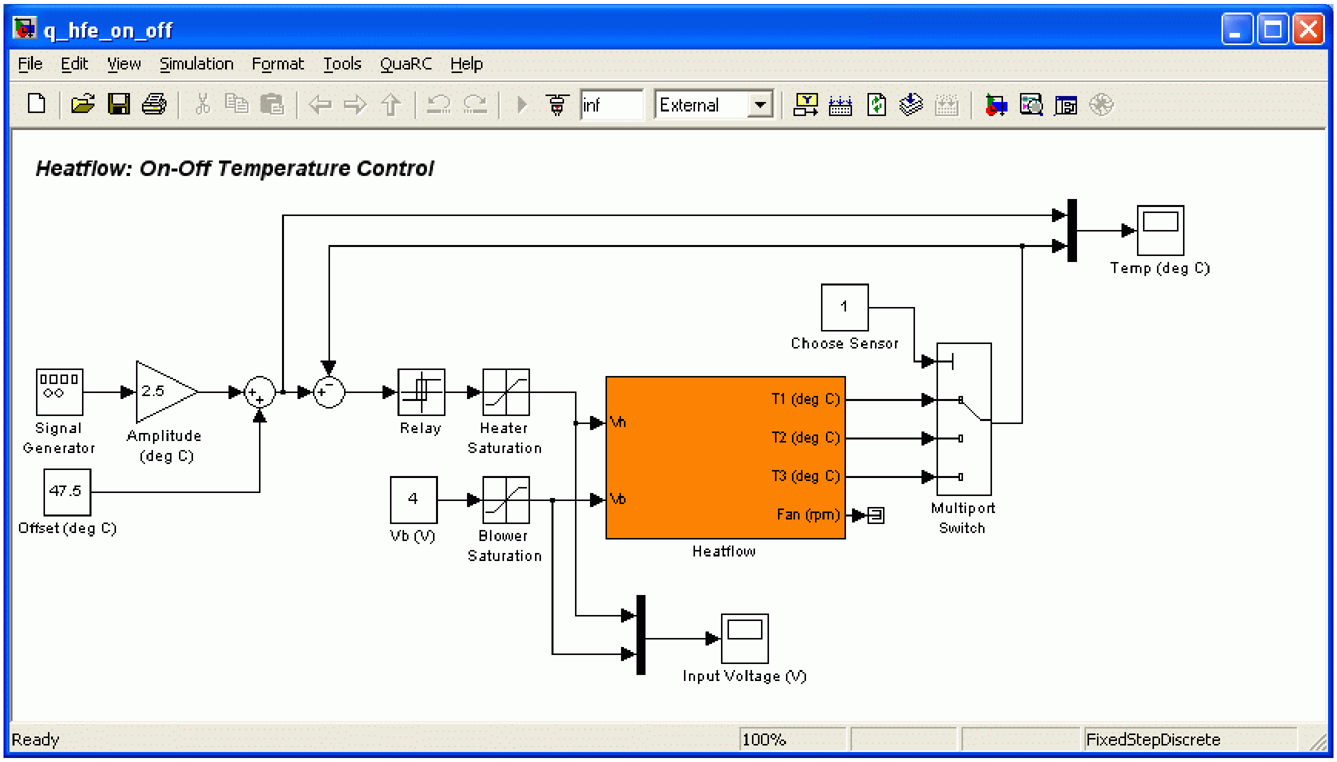


Figure 16: Simulink diagram to run on-off controller on Heat Flow using QUARC.

This procedure demonstrates how to run the on-off on the HFE:

1. Setup the Matlab workspace and the HFE Library for the *q_hfe_on_off* Simulink model (shown in Figure 16, above) as discussed in Section 5.1.1.
2. Click on *Quarc* | *Build* to compile the QUARC controller.
3. Set the *Amplitude (deg C)* block to 2.5 and the *Offset (deg C)* to 47.5. This square setpoint will vary between 45.0 °C and 50.0 °C.
4. In the *Signal Generator* block, set the *Frequency* of the square wave to 0.02 Hz.
5. Set the *Vb (V)* block to 4.0 V. The blower will be running at a constant rate while we vary the voltage to the heater, using the on-off control, to regulate the temperature to the setpoint.
6. For this run, set the *Choose Sensor* block to 1 in order to control the temperature about the T_1 sensor. However, you can select whichever sensor (1, 2, or 3) to control.
7. Go to Quarc | Start to begin running the controller.
8. The typical on-off control temperature response for sensor T_1 is shown in Figure 17 and the heater and blower voltages are shown in Figure 18. The heater voltage is the yellow trace and the purple plot is the blower voltage.

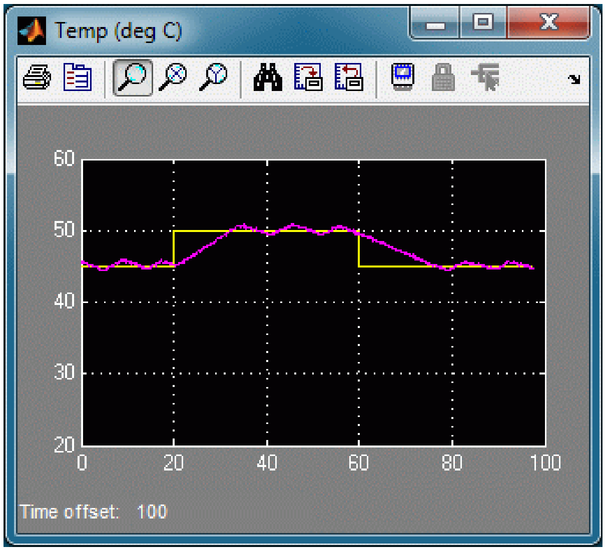


Figure 17: T1 on-off controller response.

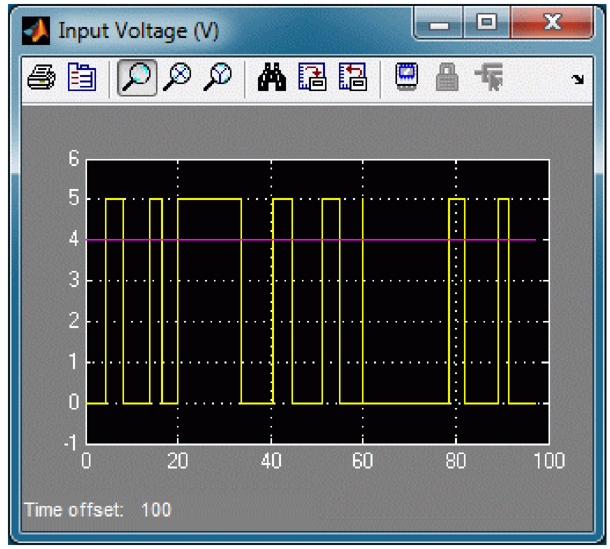


Figure 18: On-off control signals.

9. After each run, the last 100 seconds of the desired and measured temperature readings and the heater and blower voltages are saved to the Matlab workspace under the variables *data_temp* and *data_u*, respectively. You can access the data using the the commands shown in Text 2, below.

```
% save into variables
t = data_temp(:,1);
T1d = data_temp(:,2);
T1 = data_temp(:,3);
Vh = data_u(:,2);
Vb = data_u(:,3);
```

Text 2: Loading data from workspace variables data_temp and data_u.

Using the commands outlined in Text 2, above, and Matlab plotting commands, create a plot that shows the temperature response at the T_1 sensor as well as the blower and heater voltages used when running the on-off controller with a hysteresis width of 1.0 °C, $V_{h,off} = 0$, and $V_{h,on} =$ 5.0 V. Attach the resulting plot to your report.

0	1	2

10. Give one advantage and one disadvantage of the on-off controller.

0	1	2

11. Reduce the hysteresis width by a desired amount. Make sure you document what on-off control parameters were used and attach a plot of the response (showing the temperature and heater and blower voltages).

\

0	1	2

12. How does decreasing the hysteresis width affect the response? Do you see any potential drawbacks to making the width too small?

0	1	2

13. Click on QUARC | Stop to turn off the controller.
14. Power off the Heat Flow if no more experiments will be ran in this session.

Other things to try...

- Vary the on-off voltages of the heater relay.
- Add an on-off controller for the blower.
- Smooth out the output voltages using a sigmoid or rate limiter.

5.3. PI Temperature Control

The Simulink model called *q_hfe_pi,* depicted in Figure 19, is used to implement the PI compensator on the Heatflow using QUARC. The setpoint is set to the same temperature as in the on-off controller.

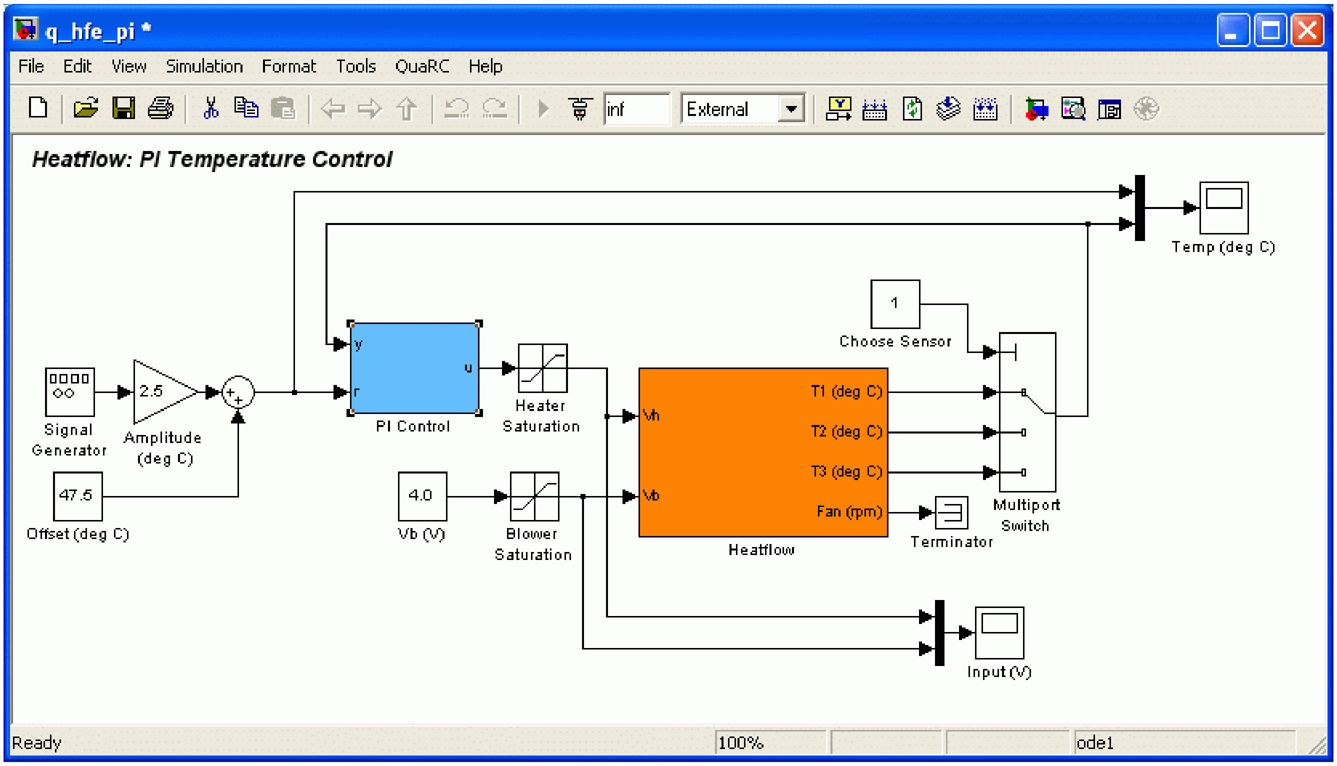


Figure 19: Simulink diagram used to run PI control on Heat Flow using QUARC.

The *PI Control* subsystem is shown in Figure 20. As depicted, the *kp* and *ki* Slider Gain blocks are used to change the proportional and integral gains, respectively. You can also vary the set-point weight and the integral anti-windup of the compensator. Set-point weight hasn't been discussed but its affect on the response will be investigated in the laboratory session.

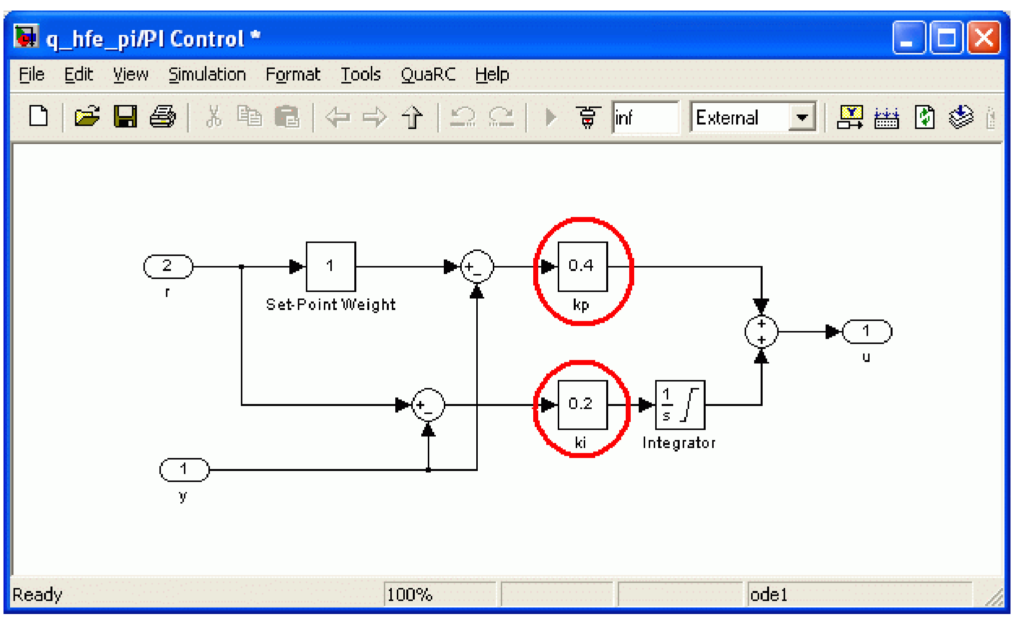


Figure 20: Inside the PI Control subsystem in q_hfe_pi model.

5.3.1.1. Qualitative Proportional Control

In this procedure, the user can see the effect of having integral control.

1. Setup the Matlab workspace and the HFE Library for the *q_hfe_pi* Simulink model as discussed in Section 5.1.1.
2. Click on *Quarc | Build* to compile the QUARC controller.
3. Set the *Amplitude (deg C)* block to 2.5 and the *Offset (deg C)* to 47.5. This square setpoint will vary between 45.0 °C and 50.0 °C.
4. In the *Signal Generator* block, set the *Frequency* of the square wave to 0.02 Hz.
5. Set the *Vb (V)* block to 4.0 V. The blower will be running at a constant rate while we vary the voltage to the heater, using the on-off control, to regulate the temperature to the setpoint.
6. For this run, set the *Choose Sensor* block to 1 in order to control the temperature about the T_1 sensor. However, in the future you can select whichever sensor (1, 2, or 3) to control.
7. In the menu of the Simulink model, go to *QUARC | Start* to begin running the controller.
8. The control temperature response for sensor T_1 is shown in Figure 21 with the PI gains set to $k_p = 0.7$ and $k_i = 0.05$. These, of course, are just the initial gains and can be changed to alter the closed-loop behaviour. The heater and blower voltages are given in Figure 22. The heater voltage is the yellow trace and the purple plot is the blower voltage. Again the blower is held at a constant 4.0 V.

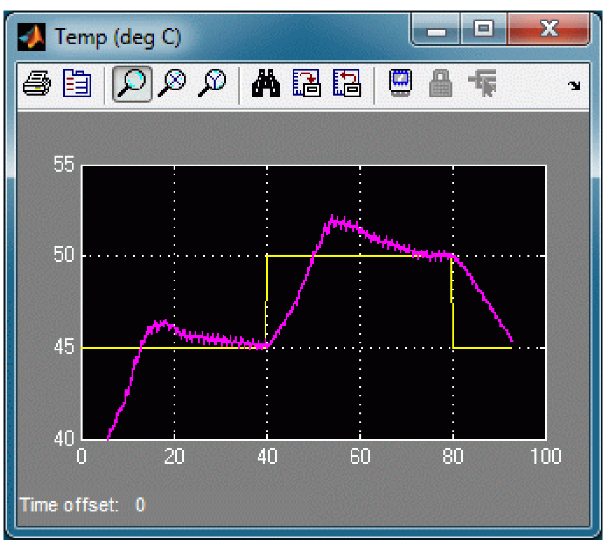


Figure 21: PI temperature response.

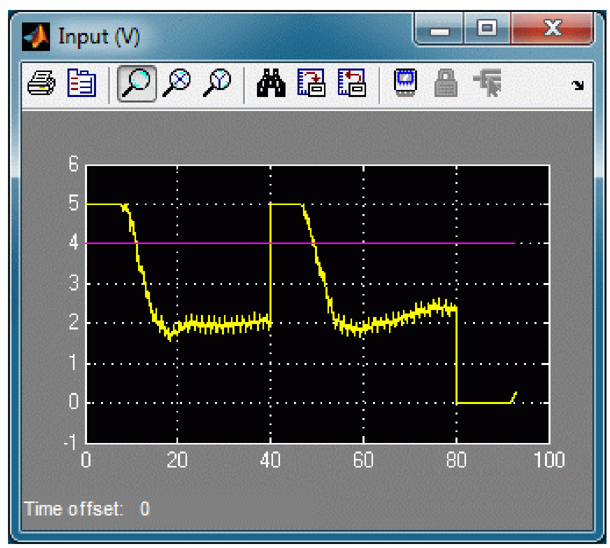


Figure 22: PI blower and heater voltages.

9. To observe the affect of a pure proportional control, set *kp* = 0.3 and *ki* = 0.
10. Allow the temperature to settle to 45.0 °C. When the step goes up to to 50.0 °C, let the controller run until the step goes back down again. Save this response in a plot and attach it to your report.

0	1	2

11. Measure the steady-state error. Is the response as expected?

0	1	2

12. Using the equation developed in Section 4.2.3, evaluate the steady-state error numerically given the amplitude of the step, the control gain 0.3, and the model parameters found in Section 5.1.2. How does it compared to the measured value (taking the sensor resolution into account, if necessary)?

0	1	2

13. Click on QUARC | Stop to stop running the controller.
14. If no more experiments will be performed in this session, turn off the Heat Flow.

5.3.1.2. Qualitative Integral Control

1. Run the *q_hfe_pi* QUARC controller with *kp* = 0.5 and *ki* = 0.
2. Slowly begin increasing the integral gain. Attach a sample response. Discuss some of the affects of using integral action in the controller.

0	1	2

3. The integral anti-windup scheme is currently being used, as identified by the saturation marks on the *Integrator* block inside the *PI Control* subsystem (pictured in Figure 20). The heater saturation is 0-5 V, so the lower and upper saturation limits of the integrator are set to 0 and 5, as shown in Figure 23, below. The integrator resets every sample, so the reset time is automatically set to T_r = 1 when using this block.

4. To turn the anti-windup off, first turn off the controller. Then go into the *PI Control* subsystem, double-click on the *Integrator* block, and un-select the *Limit Output* parameter as illustrated in Figure 23.

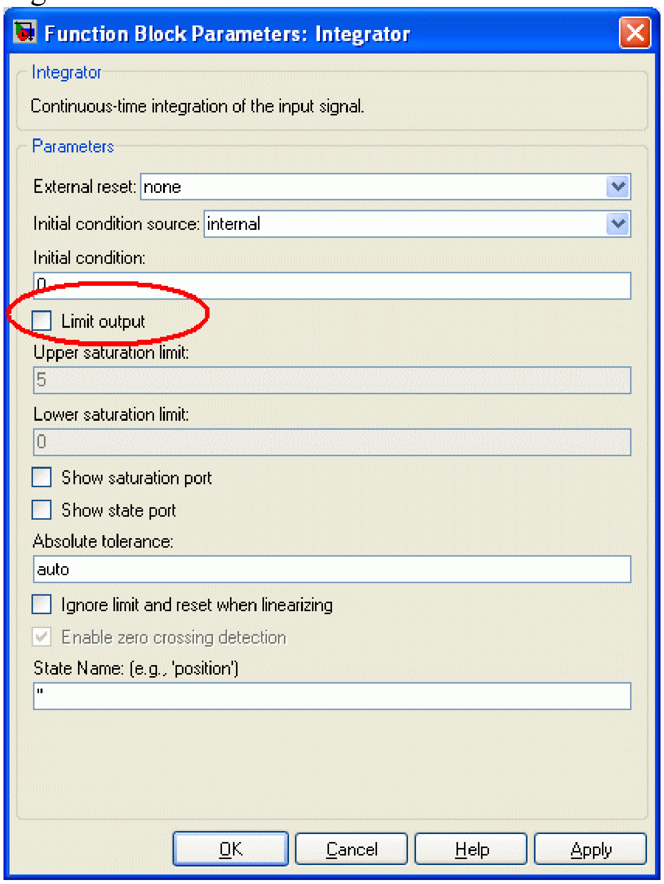


Figure 23: Integrator block parameters.

5. Rebuild and start the *q_hfe_pi* controller.
6. Set the gains to the kp = 1 and ki = 0.4 and run through a step. Attach the resulting response in a plot (make sure the temperature as well as the heater and blower voltages are given).

0	1	2

7. Re-engage the windup with the same gains and attach that response. Is the response improved?

0	1	2

8. Click on QUARC | Stop to stop running the controller.
9. If no more experiments will be performed in this session, turn off the Heat Flow.

5.3.1.3. PI Control According to Specifications

Go through this procedure to run the PI controller with gains that satisfy the specifications:

1. Recall the peak time and overshoot specifications given in Section 4.2.4. Based on the model parameters, K and τ, found in Section 5.1.2, as well as the natural frequency and damping ratio that need to be met in order to satisfy the time-domain response requirements, calculate the control gains k_p and k_i.

0	1	2

2. Run the *q_hfe_pi* QUARC controller with the PI gains and attach the corresponding closed-loop step response.

0	1	2

3. Measure the peak time and percentage overshoot. Do they satisfy the specifications?

0	1	2

4. If the response did not satisfy the requirements or to improve it, try to do some control tuning by adjusting the value of the set-point weight parameter via the Slider Gain block called *Set-Point Weight,* found in the *PI Control* subsystem. It may also be helpful to adjust the PI gains slightly. Attach any response along with the PI gains and the set-point weight used. How does the set-point weight parameter affect the response?

0	1	2

5. If adjustments were made, measure the peak time and percentage overshoot. Are the requirements now satisfied?

0	**1**	**2**

6. Stop the QUARC controller.
7. Shut off the Heat Flow if no more experiments will be conducted in this session.

Other things to try...

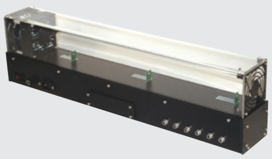
- Try using a PID controller.
- Add an PI controller for the blower.
- Using another type of controller to regulate the temperature, e.g. lead-lag.

6. References

Data Acquisition Device User Manual
Heatflow User Manual

Process control plants for teaching and research

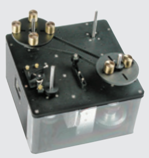
Heat Flow Experiment



Magnetic Levitation

Coupled Tanks

Industrial Mechatronic Drives Unit (IMDU)



IMDU Web Winding

IMDU Multi DOF Torsion

Quanser experimental plants can emulate complex mechatronic systems. With their precision , robust and open architecture design, they present highly suitable and affordable options for study and research of control-related challenges encountered in process control, such as fluid level control, temperature control and web winding. For more information please contact info@quanser.com

QUANSER
INNOVATE · EDUCATE

INFO@QUANSER.COM +1-905-940-3575 QUANSER.COM

Solutions for teaching and research. Made in Canada.