Heat Flow Experiment
First order modeling and control

QUANSER
INNOVATE. EDUCATE.

1   Objective

Design a simple PI control system to control the air temperature at selected points in a duct using a heating element and a blower. Develop a simple first order model and study the dynamics of the system in order to gain insight into developing more complex controllers.

This heat-flow experiment (HFE) system consist of a duct equipped with a heater and a blower at one end and three temperature sensors located along the duct. The power delivered to the heater is controlled using an analog signal. The fan speed can also be controlled using an analog signal. Fast settling platinum temperature transducers are used to measure the temperature. Fan speed is measured using a tachometer and can be used to design speed controllers.
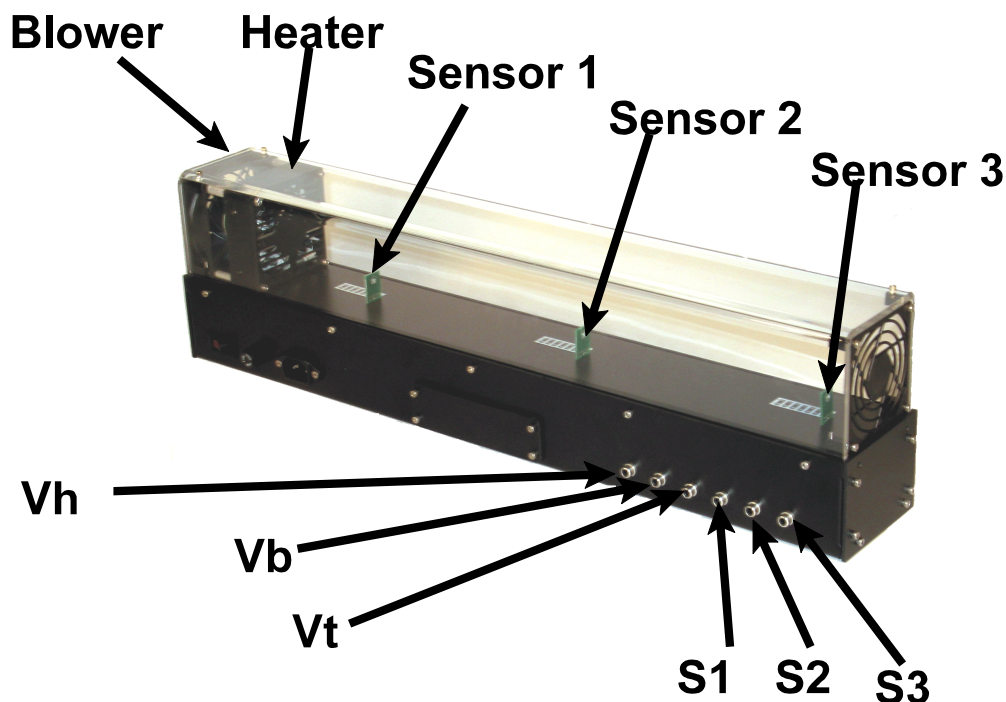

Figure 1 Heatflow apparatus

2   Open loop model

A complete thermodynamic model of the system is difficult to derive. In principle however the following equation applies:

$$\dot{T}_n = F(V_h, V_b, T_a, x_s)$$

Where

Tn	Temperature at Sensor n
Vb	Voltage applied to blower
Vh	Voltage applied to heater
Ta	Ambient temperature
Xn	Distance of sensor n from heater

This experiment is aimed at introducing you to the system in order to gain insight into the salient issues. You will develop a very simple first order model and an associated controller. In subsequent experiments you will investigate system identification techniques and control of systems with time delay.

3   First order model

In order to obtain a very simple first order model, we can apply step inputs to the heater and observe the temperature change at each sensor.

The Simulink model **open_loop.mdl** along with the script file **ol_data.m** are used to collect open loop data from the HFE. In order to run the script perform the following:

Start Matlab
Change the directory where you have the FHE models
In Matlab type open_loop - This opens the simulink model shown in Figure 2
In open_loop.mdl Click WinCon / Set WinCon Options - This sets the compile for the OS you have.
In Matlab run **init_ol** - This initializes parameters so you can compile the diagram
In the diagram click WinCon/ Build - this will build the realtime code and download to WinCon
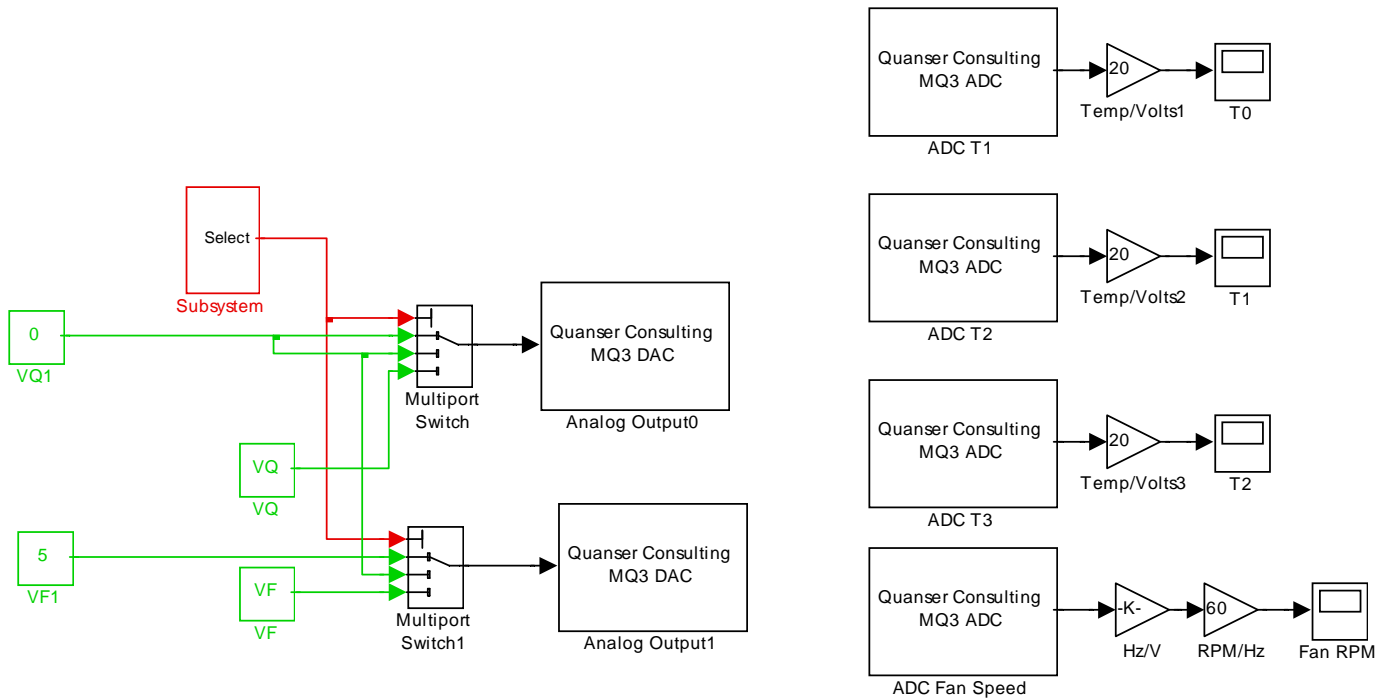
Figure 2 Simulink model ol_data.mdl

In WinCon click on Plot / New / Scope and select T0, T1 T2
In the WinCon plot, select Update/ Buffer and enter 60

Click run to ensure that the controller works. It should run for 5 seconds and then stop.

Now that you have a model built you can run the script file ol_data from matlab. The script performs the following:

Sets up the following parameters:

VF   Voltage to be applied to the blower
VQ   Voltage to be applied to the heater
T1   Duration required to flush the system of initial heat - VQ = 0, VF = 5 (full flow)
T2   Time with VF = VQ = 0
TE   Duration of experiment

Runs the controller until it stops at time TE
Saves the data into a mat file of the form ol_fhF**X**H**Z**.mat (where X = Blower voltage and Z = Heater voltage)
Runs the experiment again with zero heat and full flow to flush the contents
Repeats the above over X = 3 , 4 and 5 Volts and Z = 3 4 & 5 volts

When the script is completed, the files are available for display. Use **ol_plots** to display the traces. Figure 3 shows one of
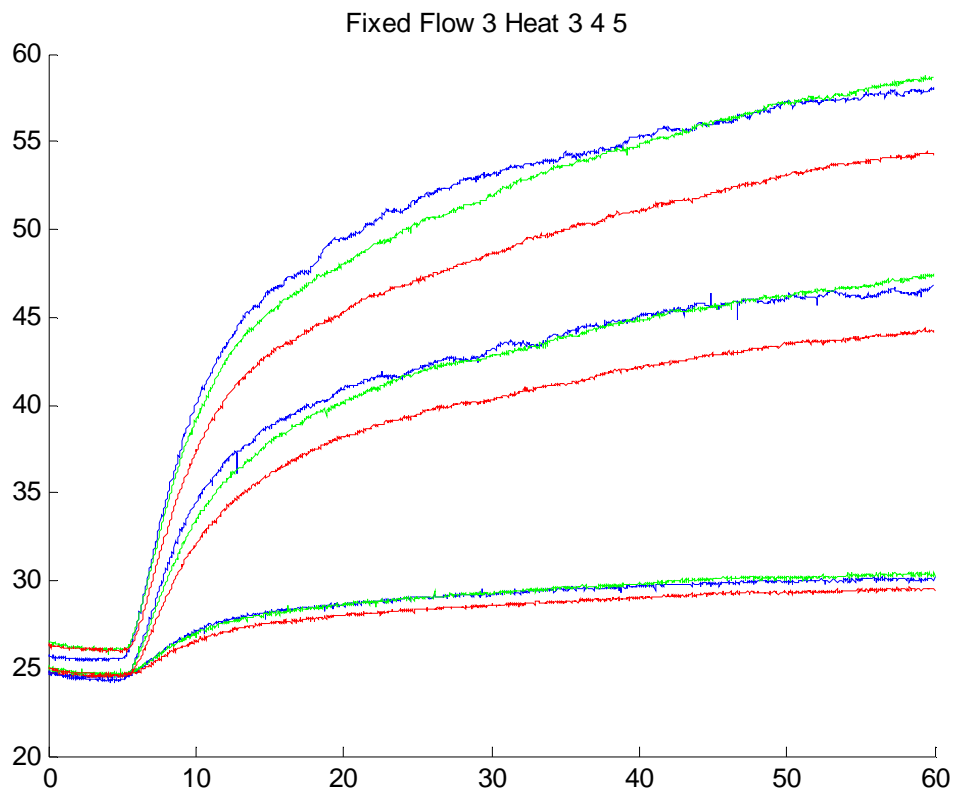
Figure 3 Step responses at the three sensors for VQ = 3 4 &5 and a fixed flow rate voltage of 3
volts.

the plots that are obtained. In this case we see the temperature in the three sensors (Blue = S1, Green = S2, red = S3)
with a fixed flow rate using VF = 3.0 Volts and three different voltages applied to the heater.

4   Control System Design

In order to design a feedback controller to control the temperature at a desired sensor location, we need to have a model
of the system. Examining Figure 3 we note that the step response can be approximated by a first order system of the form:

$$\frac{T_n}{V_Q} = \frac{G}{T_n\, s + 1}$$

No w suppose we want to control the temperature at S1 with fixed flow rate using Vf = 3.   Examining Figure 3 we note the following:

Applying VQ = 5 volts to the heater results in a temperature change of approximately 32 degrees (57 - 25). The time constant is the duration taken to reach 63% of the change. In this case that would be approximately 25+0.63*32 = approximately 45 degrees. Examining the figure we see that 45 degrees is reached in about 7.9 seconds from the time the step is applied (T = 5 seconds)

So a very simple first order approximation for the open loop model is

$$\frac{T_1}{V_Q} = \frac{6.4}{6.9\ s+1}$$

We now want to design a feedback controller that will regulate T1 with zero steady state error.

The controller is a PI controller of the form:

$$V_Q = K_p\ (T_{1d}-T_1)+\frac{Ki}{s}(T_{1d}-T_1)$$

Substituting into the first order model we obtain:

$$\frac{T_1}{T_{1d}}=\frac{G(K_p\ s+K_i)}{s^2\ \tau+(Kp\ G+1)\ s+G\ Ki}$$

whose denominator is rewritten as:

$$s^2+\frac{(K_p\ G+1)\ s}{\tau}+G\ \frac{K_i}{\tau} = s^2 +2\zeta w_o s +w_o^2$$

we can now obtain the gains Kp and Ki for a desired performance.

Suppose we want to specify a peak time and damping ratio

Then , with

$$T_{peak} = \frac{\pi}{wo\ \sqrt{1-\zeta^2}}$$

we can solve for the gains as:

$$K_p = \frac{2\ \zeta\ \pi\ \tau-T_{peak}\ \sqrt{1-\zeta^2}}{G\ T_{peak}\ \sqrt{1-\zeta^2}}$$

and

$$K_i = \frac{-\pi^2\ \tau}{G\ T_{peak}^2\ (-1+\zeta^2)}$$

5   Setting up

Wire up the system as described below. Use single RCA cables for each connection.

Inputs to system	Attach to Analog outputs	Range
Vh - Voltage to heater	MultiQ D/A # 0 - Control heat delivered	0 - 5 VDC
Vb - Voltage to blower	MultiQ D/A # 1 - Control fan speed (open loop)	0 - 5 VDC
Measurements	Attach to Analog inputs	
S1	MultiQ A/D #0	0 - 5 VDC
S2	MultiQ A/D #1	0 - 5 VDC
S3	MultiQ A/D #2	0 - 5 VDC
Vt	MultiQ A/D #3	0 - 5 VDC

6   Implementation and results

The controller is implemented using WinCon and Simulink. The Simulink controller is in the file q_hfe.mdl The controller measures the voltages from the three sensors and calibrates them to degrees. The feedback controller implements the equation

$$V_Q = K_p \ (T_{1d} - T_1) + \frac{Ki}{s}(T_{1d} - T_1)$$

depending on the value of the selector (Constant named Sensor in the diagram), we will control one of the three points along the duct. Note that we are also running a realtime simulation of the system modeled as a first order system.

In order to set the gains Kp & Ki, run the design file **d_hfe.m**. this sets the parameters as follows:

```
G = 6.4;
tau = 6.9;
zeta = 0.9;
T_peak = 10;
Kp = (2*zeta*pi*tau-T_peak*sqrt(1-zeta^2))/(G*T_peak*sqrt(1-zeta^2))
Ki = -pi^2*tau/(G*T_peak^2*(-1+zeta^2))
```
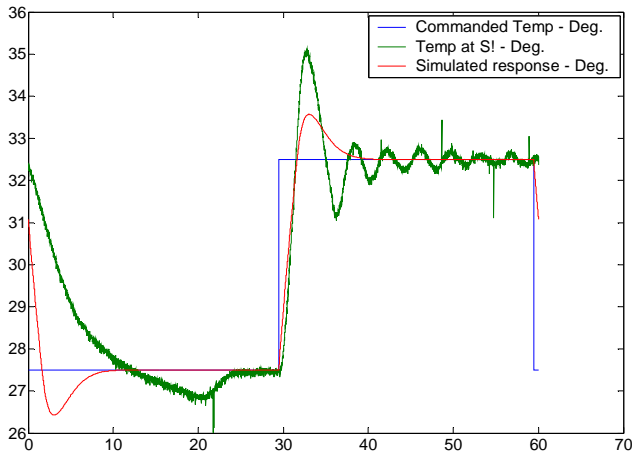
resulting in

Kp = 1.24 V / deg
Ki = 0.56 V / (deg - sec)
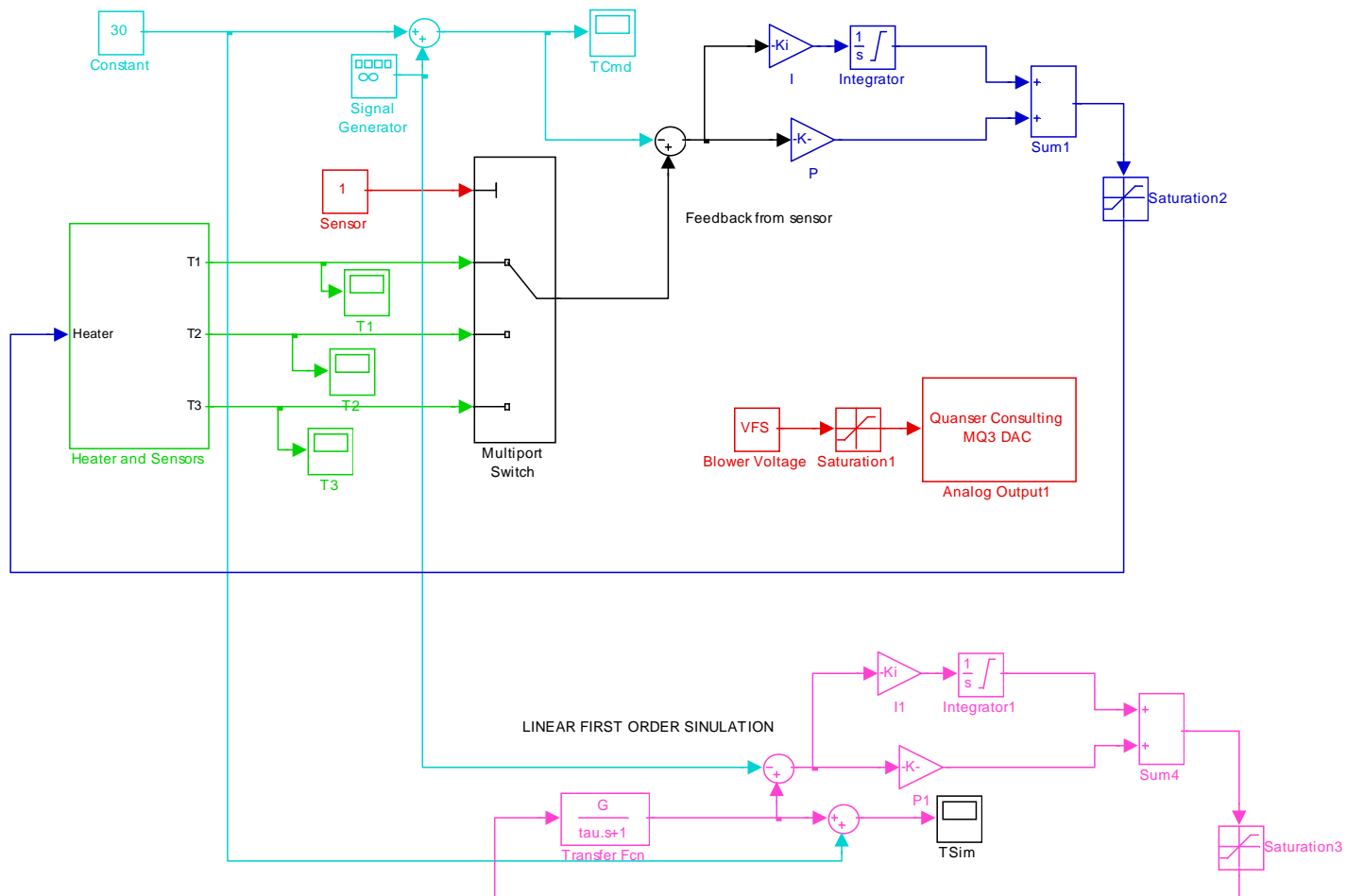
Figure 4 Simulink diagram of feedback controller q_hfe

By setting Sensor = 1, we control T1 and make it track a square wave of 5 deg p-p centered about 30 degrees. The response is shown in Figure 4.


Figure 5 Square wave input response at sensor S1. Red trace shows simulated response of first order model.

Examining the response we note the following:

1   The output tracks the desired temperature command with zero steady state error
2   The system has asymmetrical response. Heating behaves differently than cooling
3   The transient response, even during heating, does not match the simulation closely

Furthermore, we can examine the system response if we try to control the temperature at S3. We do this by setting Sensor = 3 in the diagram and running it again. This response is shown in Figure 6 . In this case, the response deteriorates and we note a maintained oscillation when heating the system. This is due to the time delay introduced due to the distance of the sensor from the heater.
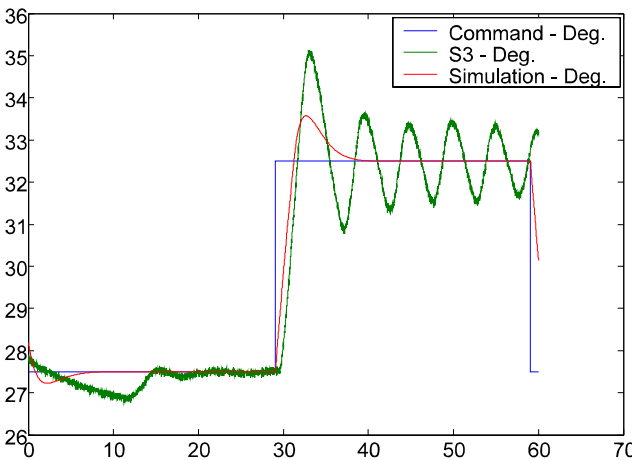
Figure 6 Response at S3 when closing the loop about S3

It is also interesting to see what happens when the blower speed is increased. Figure 7 shows the response when the controller for S1 is used to control S3 but with maximum airflow. We note that the natural frequency of the underdamped response increases and it does start dampening out - indicating that the time delay introduced into the system dynamics is shorter due to the increased airflow.
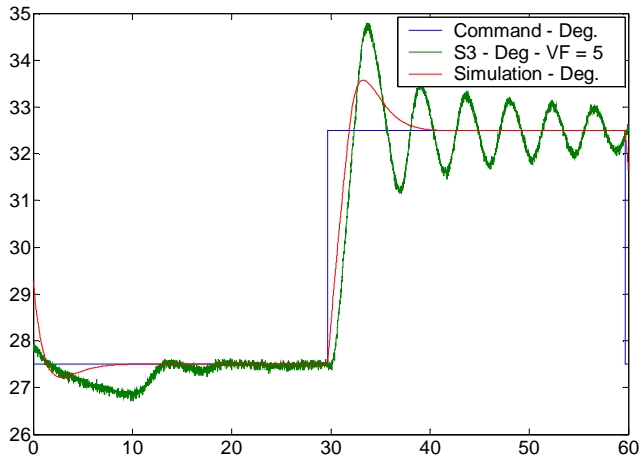
Figure 7 Controlling S3 with S1 controller using maximum airflow.

4   Conclusions

Although we were able to control th temperature at S1, the system model does not match the actual behavior closely. This is especially true when cooling.

The response from the farther sensor is not acceptable.

5   System Parameters

Parameter	Symbol	Value	Units
Heater maximum power (5 V Input)	P	400	Watt
Blower nominal Airflow (4 V Input)	B	32.1	CFM
Blower nominal Airflow (4 V Input)	B_{SI}	0.91	m^3/ min
Maximum blower speed (5 V Input)	Fmax	2580	RPM
Minimum blower speed (0 V Input)	Fmin	1580	RPM
Tachometer calibration	Ct	1063	RPM / V
Temperature Sensor calibration	Cs	20	Deg C / V
Duct cross section area	A	0.0064	m^2
Max wind speed	Ws	142	m/ min
Temp sensor settling time at max wind speed	Ts	4	seconds
Dimensions		**Power Requirements**	
50 x 15 x 10	cm	VAC	100 - 240
0.5	Kg	Amps Max	5
Data Acquisition Requirements			
Analog inputs	4	0 - 5 V DC	
Analog outputs	2	0 - 5 V DC	

6   Calibration

Occasionally you may need to calibrate the sensor measurements. You will need a measurement of ambient temperature and a DC Voltmeter. (Or you can use WinCon)

Power up the system
Open the service panel
Measure S1 and adjust potentiometer P1 such that the voltage is proportional to room temperature
Measure S2 and adjust potentiometer P2 such that the voltage is proportional to room temperature
Measure S3 and adjust potentiometer P3 such that the voltage is proportional to room temperature

The table below shows the voltages you should obtain at various temperatures:

Temp	19	20	21	22	23	24	25	26	27	28	29	30
Vs	0.95	1.00	1.05	1.10	1.15	1.20	1.25	1.30	1.35	1.40	1.45	1.50