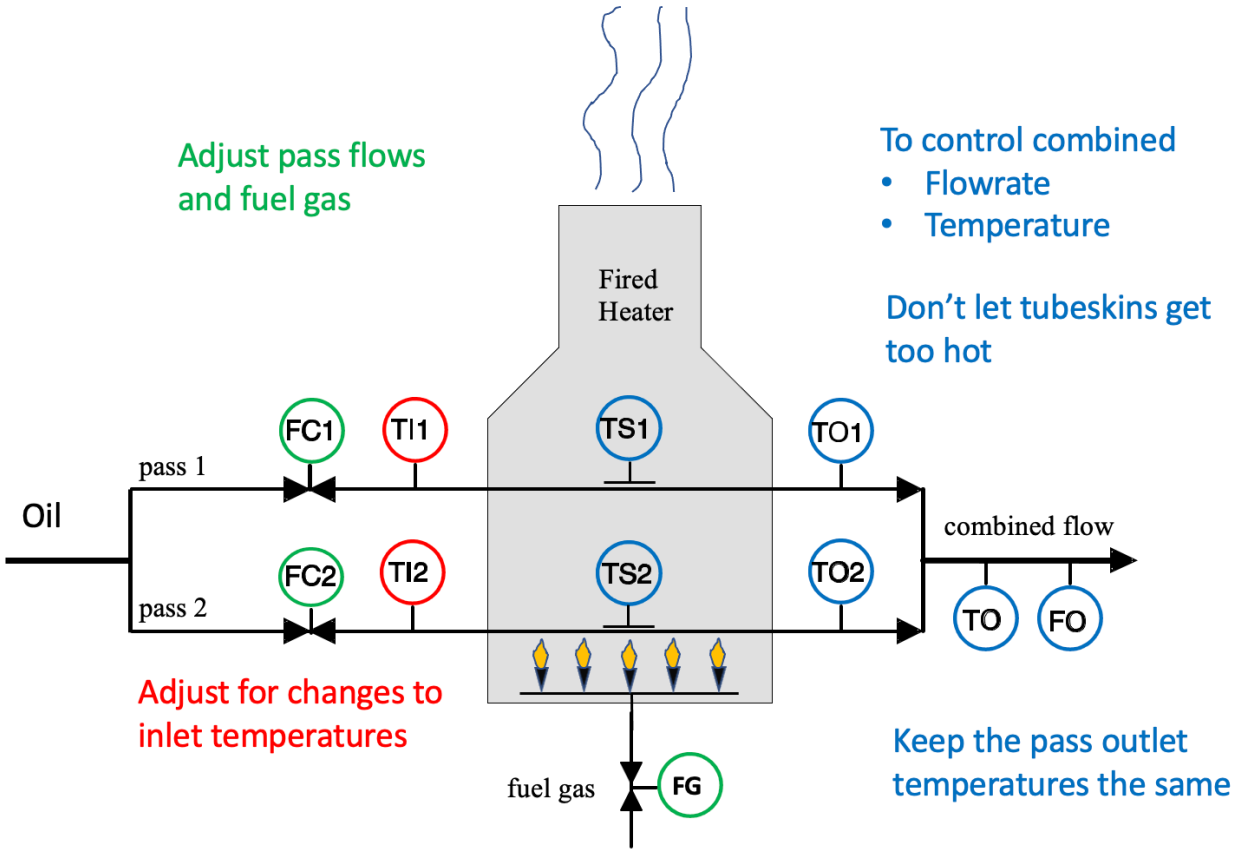
**ChBE 4412 Design Project Phase 2**

**Fall 2021**

**Due Friday December 3**

This case study was contributed by Dr. Tom Badgwell:




**Resources**

- Read "Model Predictive Control in Practice" by Badgwell and Qin, in the Reading Folder.
- Go through the "Fired Heater MPC Simulation Tutorial" in the Reading folder, including installation of the software.
- Watch the video by Tom Badgwell in the Design Project folder.

*Notes:*

- You can engage with this material in any order that you choose, but all three are valuable resources for you.
- Even though you read these materials for Phase 1, you should read them again, as they will contain specific information needed for Phase 2.

| **Table 1:** Fired heater MPC design specification | | | | |
|---|---|---|---|---|
| **Variable** | **Description (Units)** | **Type** | **Priority** | **Specification** |
| TS1 | Tube 1 skin temperature (DEGF) | CV | 1 | Max. limit |
| TS2 | Tube 2 skin temperature (DEGF) | CV | 1 | Max. limit |
| TO | Combined outlet temperature (DEGF) | CV | 2 | Setpoint |
| FO | Combined outlet flowrate (BPH) | CV | 3 | Setpoint |
| TO1 | Tube 1 outlet temperature (DEGF) | - | - | - |
| TO2 | Tube 2 outlet temperature (DEGF) | - | - | - |
| DT | Delta temperature TO1 – TO2 (DEGF) | CV | 4 | Setpoint (0) |
| FC1.SP | Flow controller tube 1 setpoint (BPH) | MV | - | Max/Min/ROC |
| FC2.SP | Flow controller tube 2 setpoint (BPH) | MV | - | Max/Min/ROC |
| FG.SP | Fuel gas flow controller setpoint (MSCFH) | MV | - | Max/Min/ROC |
| TI1 | Inlet temperature tube 1 (DEGF) | DV | - | - |
| TI2 | Inlet temperature tube 2 (DEGF) | DV | - | - |

**Assignment**

The state-space model of the fired heater is located in Software/mpcsim: the MATLAB command `load('HeaterModelCont.mat')` will read in *A, B, C,* and *D* for the system of five inputs and five outputs. Note that this model is in deviation variables. You can use the State Space block in Simulink to simulate this system. Given the difficulty of constructing a model based on balance equations (topic of Phase 1), we will use an empirical model for Phase 2.

Consider the scenario in which

i. The system is initially operating at steady state under closed-loop control with a temperature setpoint of 750°F, which corresponds to inputs of FC1 = FC2 = 100 BPH, FG = 95 MSCFH, and TI1 = TI2 = 540°F.
ii. The temperature setpoint for TO is suddenly changed from 750 to 755°F.
iii. After reaching the new steady state, the setpoint for FO is suddenly changed from 200 to 220 BPH.
iv. After reaching the new steady state, the setpoint for FO is suddenly changed from 220 to 250 BPH.

Part A:

i. Calculate the steady-state gain matrix *K* and the Bristol relative gain array Λ for the 5x5 system of inputs and outputs.
ii. In words, explain how the outputs are dependent on the inputs. Based on these matrices, which MVs would you pair with each CV?
iii. Design a SISO PID-type controller that manipulates fuel gas to control the outlet temperature to a setpoint. Explain your design methodology and show your calculations.
iv. Design a SISO PID-type control that adjusts the total flow (FC1+FC2) to control the outlet flow. Explain your design methodology and show your calculations.
v. How would you ensure that the tubeskins do not exceed their constraint temperature? Incorporate elements into your control strategy for this important safety priority.

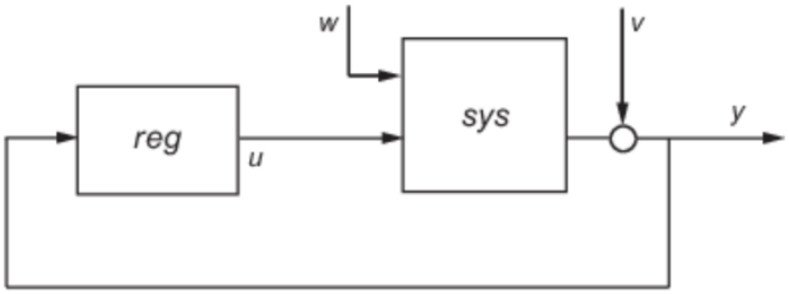
vi. How would you balance the passes to keep ΔT close to zero? Incorporate elements into your control strategy to reduce the risk of coking and clogging associated with large ΔT.

vii. In Simulink, run your controller for the scenario described above. Plot the MVs and the CVs. Describe the characteristics of the response, including whether or not the controller performs well.

Part B: Linear quadratic Gaussian (LQG) control

The linear quadratic Gaussian (LQG) controller is a model-based multivariable feedback controller that calculates the adjustments to all MVs, using the tracking error for all CVs. Type `help lqg` in MATLAB to get some description of the inputs and outputs for this function.

Here is some description taken directly from the MATLAB documentation:




The LQG regulator minimizes the cost function

$$J = E\left\{\lim_{\tau\to\infty}\frac{1}{\tau}\int_0^{\tau}[x^T, u^T]Q_{xu}\begin{bmatrix}x\\u\end{bmatrix}dt\right\}$$

subject to the plant equations

$$dx/dt = Ax + Bu + w$$
$$y = Cx + Du + v$$

where the process noise $w$ and measurement noise $v$ are Gaussian white noises with covariance:

$$E\left(\begin{bmatrix}w\\v\end{bmatrix}\cdot\begin{bmatrix}w' & v'\end{bmatrix}\right) = QWV$$

In order to calculate the optimal linear controller (which is another state-space system), you must specify the weighting matrices $Q_{xu}$ and $Q_{wv}$. (You will use these same matrices for Part C on MPC.) The idea is to minimize the cost function $J$, so that the state deviations are small, but the input action $u$ is also small. The weighting matrix $Q_{xu}$ is what helps you balance these two objectives, depending on your preference to have low tracking error versus to keep the input action small. The matrix $Q_{wv}$ represents how large is the noise in the process and the sensor, based on the expected variance of each quantity.

Here is some code to help you calculate the LQG controller:

```
% Calculate the LQG controller
B_MV = B(:,1:3);  % only use the 3 MV's as inputs here, not the DV's
D_MV = D(:,1:3);
n = size(A,1);      % state dimension
m = size(B_MV,2);   % input dimension (MVs only)
p = size(C,1);      % output dimension
sys = ss(A,B_MV,C,D_MV);
% Weightings on tracking error and on input (MV) for LQG and MPC
Qc = diag([100, 1, .001, .001, .001]);    % Weighting on state
Rc = diag([.001, .001, .01]);             % Weighting on input
% The Qc for MPC is for output control, so the Q weight for the LQG
calculation is
Q = C'*Qc*C;
% Weights for the state estimation part
Qe = eye(n);
Re = diag([0.0001, 0.0001, 0.0001, 0.0001, 0.0001]);
Qwv = [Qe zeros(n,p); zeros(p,n) Re];
Qxu = [Q zeros(n,m); zeros(m,n) Rc];
% Final step is to call the LQG function, which returns a structure
Klqg = lqg(sys,Qxu,Qwv);
```

i. In Simulink, run your controller for the scenario described above. Plot the MVs and the CVs. Describe the characteristics of the response, including whether or not the controller performs well.

Part C: Model Predictive Control (MPC)

Model predictive control is similar to LQG, in that the optimal action is calculated based on a model of the process, to minimize the user-specified cost function *J*. However, with MPC we can also incorporate constraints, which are not a part of the LQG approach.

Using the graphical user interface from Part 1, turn on the closed-loop function with the third button, having the circular arrow. You can change the setpoint using the CV menu.

i. Simulate the system according to the scenario described above. Plot the MVs and the CVs. (Ok to take a screen shot.) Describe the characteristics of the response, including whether or not the controller performs well.
ii. Flip the priorities for TO and FO by changing the weights in the Q's. Document your work and interpret the plots.
iii. Stop the pass flows from moving for a TO setpoint change by increasing their Rc weights. Document your work and interpret the plots.
iv. Slow down the TO and FO setpoint responses by increasing the Sc weights on the three MVs. (Note: the Sc weight is for the steady-state behavior, see video.) Document your work and interpret the plots.