MPC Case Studies

Temperature Control of Fluid in a Tank

- The temperature of the fluid contained in a tank with a constant flow rate in and out is to be controlled.
- The control variable is the temperature of the incoming fluid which is adjusted by a mixing valve that regulates the relative amounts of hot and cold fluid supply, as indicated in the diagram below.

Fluid temperature inside the tank is governed by the following first-order differential equation:

\[ \dot{T}_e = \frac{1}{cM}(q_i - q_o) \]

where
\[ c \triangleq \text{specific heat of fluid} \]
\[ M \triangleq \text{fluid mass in tank} \]
\[ T_e \triangleq \text{tank temperature} \]
\[ q_i = c \dot{m}_i T_{ei} \]
\[ q_o = c \dot{m}_o T_e \]
\[ \dot{m} \triangleq \text{mass flow rate} \]

- The temperature at the tank input at time \( t \) is the control temperature, \( T_{ec} \)

- A fluid transport delay of \( \tau_d \) seconds is present between the mixing valve outlet and the tank input:

\[ T_{ei}(t) = T_{ec}(t - \tau_d) \]

- Substituting we get

\[
\dot{T}_e = \frac{1}{c M} q_i - \frac{1}{c M} q_o \\
= \frac{1}{c M} c \dot{m}_i T_{ei} - \frac{1}{c M} c \dot{m}_o T_e \\
= \frac{\dot{m}_i}{M} T_{ei} - \frac{\dot{m}_o}{M} T_e
\]

which allows us to write

\[
\dot{T}_e + \left( \frac{\dot{m}_o}{M} \right) T_e = \left( \frac{\dot{m}_i}{M} \right) T_{ec}(t - \tau_d)
\]

- Let

\[
a = \frac{\dot{m}_o}{M} = \frac{\dot{m}_i}{M} = \frac{\dot{m}}{M}
\]
and we have

$$\dot{T}_e(t) + a T_e(t) = a T_{ec}(t - \tau_d)$$

- We can now generate a transfer function between the output variable $T_e(s)$ and the input $T_{ec}(s)$:

$$\frac{T_e(s)}{T_{ec}(s)} = \frac{e^{-\tau_d s}}{s/a + 1} = G(s)$$

- Converting to discrete-time we generate the $Z$-Transform, assuming a zero-order hold,

$$Z\left\{ \frac{1 - e^{-T_s}}{s} \cdot \frac{e^{-\tau_d s}}{s/a + 1} \right\}$$

- Assume

$$\tau_d = \ell T - mT, \quad 0 < m \leq 1$$

Then,

$$G(z) = Z\left\{ \frac{1 - e^{-T_s}}{s} \cdot \frac{e^{-\ell T_s} e^{mT_s}}{s/a + 1} \right\}$$

$$= (1 - z^{-1})z^{-\ell} \cdot Z\left\{ \frac{e^{mT_s}}{s(s/a + 1)} \right\}$$

$$= (1 - z^{-1})z^{-\ell} \cdot Z\left\{ \frac{e^{mT_s}}{s} - \frac{e^{mT_s}}{s + a} \right\}$$

$$= \left( \frac{z - 1}{z} \right) \cdot \left( \frac{1}{z^{\ell}} \right) \cdot \left( \frac{z}{z - 1} - \frac{e^{-amT}}{z - e^{-amT}} \right)$$

$$= \frac{(1 - e^{-amT})z + e^{-amT} - e^{-aT}}{z - e^{-aT}}$$
This gives
\[ G(z) = \frac{1 - e^{-a m T}}{z^\ell} \cdot \frac{z + \alpha}{z - e^{-a T}} \]

where we’ve defined
\[ \alpha = \frac{e^{-a m T} - e^{-a T}}{1 - e^{-a m T}} \]

**CASE 1**

- Let \( \dot{m} = 1000 \text{ kg/s} \) and \( M = 1000 \text{ kg} \)
  - This means that a volume equal to the entire content of the tank (i.e., \( 1.0 \text{ m}^3 \)) will flow in and out each second
  - This gives \( a = \frac{\dot{m}}{M} = 1 \)
  - Further assume that the sampling time \( T = 1 \text{ s} \) and the transport time delay is \( \tau_d = 1.5 \text{ s} \); which means \( \ell = 2.0 \text{ s} \) and \( m = 0.5 \text{ s} \).

- Substituting into the expression for \( G(z) \) above yields
  \[ G_1(z) = \frac{Y(z)}{U(z)} = \frac{(0.3935) \cdot (z + 0.6065)}{z^2(z - 0.3679)} = \frac{0.3935z^{-2} + 0.2387z^{-3}}{1 - 0.3679z^{-1}} \]

- In difference equation form this gives,
  \[ y(k) = 0.3679y(k-1) + 0.3935u(k-2) + 0.2387u(k-3) \]
  or in general terms
  \[ y(k) = e^{-aT} y(k-1) + (1 - e^{-a m T})u(k-2) + (e^{-a m T} - e^{-a T})u(k-3) \]

- Note that the zero location varies considerably as \( m \) varies throughout its range:
  \[ \alpha \to 0 \text{ as } m \to 1 \]
  \[ \alpha \to \infty \text{ as } m \to 0 \]

- Let’s now examine the unit pulse response for Case 1
Now if we assume a time delay $\tau_d = 1.9 \text{s}$, then this gives $\ell = 2.0 \text{s}$ and $m = 0.1 \text{s}$.

The resulting transfer function is

$$G_2(z) = \frac{(0.0952) \cdot (z + 5.6425)}{z^2(z - 0.3679)}$$

indicating a non-minimum phase zero.

- We plot both impulse responses below and note the different dynamics.
- More realistically, we let the flow rate $\dot{m} = 100 \text{ kg/s}$ which gives a ratio $\dot{m}/M = 0.1$

- Keeping the sampling interval the same, we generate two new transfer functions for $m = .1$ and $m = .5$ respectively,

$$G_3(z) = \frac{(0.01)(z + 8.5639)}{z^2(z - 0.9048)}$$

$$G_4(z) = \frac{(0.0488)(z + .9512)}{z^2(z - .9048)}$$

- The corresponding impulse responses are plotted below
For one more examination, we’ll increase the sampling interval to \( T = 5 \text{s} \).

We assume again a flow rate \( m = 100 \text{ kg/s} \) and compute the transfer functions for time delays of 7.5 s and 9.5 s, respectively (corresponding to \( m = .1 \) and \( m = .5 \)).

This time we obtain

\[
G_5(z) = \frac{(0.0488)(z + 7.0678)}{z^2(z - 0.6065)}
\]
\[
G_6(z) = \frac{(0.2212)(z + 0.7788)}{z^2(z - 0.6065)}
\]

with corresponding unit pulse responses,
At this point, let’s attempt to control the tank temperature using classical feedback control techniques.

For this, we’ll first assemble a feedback control configuration for the system.

- Note that we have implicitly assumed all quantities to be sampled discrete-time.
You'll notice here that for generality, we have indicated a temperature sensor (since one must somehow exist) and a valve controller.

We won't develop this here, but we must be mindful of the practical need for a mechanism that will convert our control input signal, $U(z)$, into a valve action that will result in the correct output temperature, $T_{ec}$, for the mixed fluid.

Note also that we have absorbed the transport delay into the tank transfer function $G(z)$.

For purposes of this case study we will focus on transfer function $G_5(z)$:

$$G_5(z) = \frac{(0.0488)(z + 7.0678)}{z^2(z - 0.6065)}$$

**Physical Interpretation**

Let's take a closer look at the system under study to gain a deeper understanding of its behavior.
- Transfer function $G_5(z)$ relates the output (tank) fluid temperature $T_e$ to the input (control) fluid temperature $T_{ec}$, where both quantities are expressed in °K.

- Without loss of generality, we can consider

$$T_e = T - T_o$$

where $T$ is the actual tank temperature and $T_o$ is a nominal (equilibrium) state.

  - In this sense, $T_e$ refers to the relative temperature difference from equilibrium and may be equivalently expressed in °C.

- Therefore, a reference demand on the system is a request to change the temperature from the current value by the amount of the reference.

  - For example, $T_{ref} = 10\, C$ indicates we wish to raise the internal tank temperature by $+10\, C$.

  - Conversely, $T_{ref} = -5\, C$ commands a reduction in temperature from the nominal by $5\, C$.

- Our interest in the magnitude of the control effort is driven by the physical reality that places real limits on the available temperature of the hot and cold feed supplies.

  - In our problem configuration, constraints placed on control magnitude reflect how much hotter (or colder) the supply temperatures must be relative to the nominal.

  - We assume fluid temperature mixing takes place according to the following formula

$$T_{ec} = \gamma T_h + (1 - \gamma) \, T_c$$

assuming constant specific heat and mass flow rates.
Here, \( T_h \) and \( T_c \) represent hot and cold supply temperatures, and \( 0 \leq \gamma \leq 1 \) is the mixing ratio and is controlled by the mixing valve.

- Our analysis below will utilize the discrete unit-step input for comparison of control performance.
  - Strictly speaking, a unit-step demand requests a change in temperature of 1°C from the nominal value.
  - Whereas this may be practical for some applications, it does not represent the full range of operation and is shown here for analysis purposes.

**Proportional-Integral (PI) Control**

- The uncompensated transfer function exhibits a finite value DC gain \( G(1) = 1 \) which will result in a constant offset error in steady-state.
  - Additionally, modeling errors will further affect accurate set-point tracking.

- For a baseline, we’ll design a simple PI controller using the Ziegler-Nichols tuning rules in order to introduce integral action.

- A discrete-time PI controller has the following form:

\[
K_{PI}(z) = K_p + \frac{K_p T z}{T_l(z - 1)}
\]

\[
K_{PI}(z) = K_p + \frac{K_p T z}{T_l(z - 1)} = \frac{T_l K_p (z - 1) + K_p T z}{T_l (z - 1)}
\]
\[
\frac{(T_I + T) K_p z - T_I K_p}{T_I (z - 1)}
\]
\[
\frac{(T_I + T) K_p}{T_I} \left( z - \frac{T_I}{(T_I + T)} \right)
\]
\[
\frac{1}{(z - 1)}
\]
where we’ll have to compute appropriate values for \( K_p \) and \( T_I \) using the tuning rules

- Our first step will be to generate an open-loop discrete-time unit step response for our plant \( G_S(z) \)
  - This is depicted via Matlab simulation below.

- Measuring from the plot we can determine the two process parameters \( L \) and \( R \) to be
\[ L = \tau_d = 2.0T = 10 \text{ sec} \]
\[ R = \frac{1}{\tau} = \frac{1}{2.5T} = 0.08s^{-1} \]

From these, the Ziegler-Nichols tuning parameters are found to be

\[ K_p = \frac{0.9}{RL} = \frac{0.9}{(0.08)(10)} = 1.125 \]
\[ T_I = 3L = (3)(10) = 30.0 \text{ sec} \]

- Substituting these values gives our compensator as

\[ K_{PI}(z) = \frac{(30 + 5)(1.125)}{(30)} \left( z - \frac{30}{(30 + 5)} \right) \]
\[ = \frac{1.3125(z - 0.8571)}{z - 1} \]

- Here, it is useful to examine the root locus plot in order to gain some insight into closed-loop system behavior:
The closed-loop poles are indicated by the blue ’x’ s in the figure
- Performance is governed by the complex conjugate pair of poles located at \( s_{1,2} = 0.6356 \pm j.5941 \) combined with the ’slow’ pole at \( s_3 = 0.9031 \)

Note that since the complex conjugate roots are not strictly dominant (because of the real slow pole), we cannot directly apply the damping specifications
- Computing the closed-loop step response of the PI-compensated system, we obtain the following:
The system exhibits an oscillatory response with a maximum peak overshoot of nearly 20%, which is typical of Ziegler-Nichols PI compensation.

PI compensation gives the following time-domain measures:

\[ t_r = 1.8 T = 9.0 \text{ sec}, \quad t_s = 35 T = 175 \text{ sec}, \quad M_{PO} = 19\% \]

Examining the corresponding control magnitude, we obtain the following:
• Control effort is likewise oscillatory, with a maximum magnitude of 
  \( u_{\text{max}} = 1.6 \)

• Depending on the application, it may be the case that the process is sensitive to such persistent fluctuations in temperature

• Let’s examine the closed-loop performance we would obtain with a range of gains \( K_p \)
• Clearly, by reducing the gain substantially, we can avoid excessive overshoot, but at the expense of slowing down the overall response

• So, whereas PI control eliminates steady-state error, the response is unsatisfactory in terms of excessive oscillation and slow dynamics

• With additional effort, we could extend the classical control approach with lead-lag compensation and perhaps improve on this response

• Let’s now investigate what we can achieve with MPC...

**Model Predictive Control: Unconstrained Case**

• We’ll first design an unconstrained model predictive controller to see if we can improve overall system response

• Computing an equivalent state-space representation from the transfer function $G_5(z)$ we obtain

\[
A = \begin{bmatrix}
0 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0.6065
\end{bmatrix}
\]
We’ll begin by computing the MPC step response for a two-step control horizon $N_c = 2$ and range of prediction horizons between $N_p = 2$ and $N_p = 10$ while setting the control weighting $R = 0.1$

- Responses for $N_p = 2$ and $N_p = 3$ are too oscillatory, but responses converge to a more satisfactory result for $N_p = 4$ and greater.
Let’s now fix $N_p = 10$ and vary the control horizon to assess its affect on performance.

Here we see the response is largely insensitive to values of control horizon beyond $N_c = 2$.

Now, selecting values $N_p = 10$ and $N_c = 2$, we’ll vary the control weighting $\bar{R}$,
– Clearly, the output response is highly sensitive to relative amount of control weighting
– From this parametric study, it appears that good response characteristics are obtained for $0.1 \leq \bar{R} \leq 1.0$
– Examining this region of the parameter space more closely,

– We see that a good compromise between speed and overshoot is achieved for $\bar{R} = 0.6$
- For the selected design tuning parameters, $N_p = 10$, $N_c = 2$, and $\bar{R} = 0.6$, we obtain the following performance measures:

<table>
<thead>
<tr>
<th>Measure</th>
<th>MPC-uncon</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_r$</td>
<td>13.2 sec</td>
</tr>
<tr>
<td>$t_s$</td>
<td>50.0 sec</td>
</tr>
<tr>
<td>$M_{PO}$</td>
<td>6.7%</td>
</tr>
<tr>
<td>$u_{max}$</td>
<td>1.30</td>
</tr>
</tbody>
</table>

- A comparison with our previous PI design appears below,

Comparing performance measures, we obtain:

<table>
<thead>
<tr>
<th>Measure</th>
<th>PI</th>
<th>MPC-uncon</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_r$</td>
<td>9.1 sec</td>
<td>13.2 sec</td>
</tr>
<tr>
<td>$t_s$</td>
<td>175 sec</td>
<td>50.0 sec</td>
</tr>
<tr>
<td>$M_{PO}$</td>
<td>17.7%</td>
<td>6.7%</td>
</tr>
<tr>
<td>$u_{max}$</td>
<td>1.60</td>
<td>1.30</td>
</tr>
</tbody>
</table>

**Model Predictive Control: Constrained Case**

- Perhaps the greatest advantage to using MPC is its ability to enforce hard constraints on problem variables
For the given example, we shall enforce hard limits on the control magnitude such that \(-1.1 \leq u[k] \leq 1.1\)

- The hope here is that by relaxing the control penalty - and enforcing the constraint - we can achieve a fast and acceptable response.

We introduce these bound by way of the linear constraint equation

\[ M \Delta U \leq \gamma \]

with corresponding matrices:

\[
M = \begin{bmatrix}
1 & 0 \\
0 & 1 \\
-1 & 0 \\
0 & -1 \\
1 & 0 \\
1 & 1 \\
-1 & 0 \\
-1 & -1
\end{bmatrix}, \quad
\gamma = \begin{bmatrix}
0.0 \\
0.0 \\
0.0 \\
0.0 \\
1.1 \\
1.1 \\
-1.1 \\
-1.1
\end{bmatrix}
\]

- The resulting output and control plots are presented below for the range of control weighting \(0.2 \leq \bar{R} \leq 1.0\),
– It is clear to see that MPC held the constraint tightly, which had the effect of reducing the sensitivity of the output response to $\bar{R}$

– It is also evident we’ve given up some performance on account of the constraint; a comparison of the $\bar{R} = 0.6$ case with prior results appears below

<table>
<thead>
<tr>
<th>Measure</th>
<th>PI</th>
<th>MPC-uncon</th>
<th>MPC-con</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_r$</td>
<td>9.1 sec</td>
<td>13.2 sec</td>
<td>17.3 sec</td>
</tr>
<tr>
<td>$t_s$</td>
<td>175 sec</td>
<td>50.0 sec</td>
<td>50.0 sec</td>
</tr>
<tr>
<td>$M_{PO}$</td>
<td>17.7%</td>
<td>6.7%</td>
<td>2.2%</td>
</tr>
<tr>
<td>$u_{max}$</td>
<td>1.60</td>
<td>1.30</td>
<td>1.10</td>
</tr>
</tbody>
</table>

• All three cases are plotted together in the following figure
Model Predictive Control: Laguerre

- Finally, we’ll re-run the same case using an unconstrained version of the Laguerre expansion form of the MPC algorithm.
- For this run, we’ll choose the number of Laguerre functions to be $N = 3$ and select the Laguerre pole as $a = 0.8$; all other problem parameters remain the same.
- The resulting output is shown below:
• For purposes of comparison, we’ll compute the $\tilde{R} = 0.6$ performance parameters for the Laguerre case as well,

<table>
<thead>
<tr>
<th>Measure</th>
<th>PI</th>
<th>MPC-uncon</th>
<th>MPC-con</th>
<th>MPC-Laguerre</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_r$</td>
<td>9.1 sec</td>
<td>13.2 sec</td>
<td>17.3 sec</td>
<td>16.3 sec</td>
</tr>
<tr>
<td>$t_s$</td>
<td>175 sec</td>
<td>50.0 sec</td>
<td>50.0 sec</td>
<td>40.0 sec</td>
</tr>
<tr>
<td>$M_{PO}$</td>
<td>17.7%</td>
<td>6.7%</td>
<td>2.2%</td>
<td>1.1%</td>
</tr>
<tr>
<td>$u_{max}$</td>
<td>1.60</td>
<td>1.30</td>
<td>1.10</td>
<td>1.16</td>
</tr>
</tbody>
</table>

• It appears that the Laguerre form of MPC performed on par with the best performance we obtained from constrained MPC for this example
(mostly blank)