CHAPTER 4

PERFORMANCES OF VARIOUS DEAD TIME COMPENSATING CONTROLLERS

In this chapter, performances of various dead time compensating controllers with constant controller parameters are analyzed through simulation for the effects of process parameter variations. A new double control scheme is proposed and its performance is compared with those of the various dead time compensating controllers like the conventional Smith predictor (Smith 1957, 1959), double control scheme (Tian and Gao 1998), and modified Smith predictor scheme (Vrecko et al 2001). Servo and regulatory responses are obtained through simulation.

4.1 SMITH PREDICTOR

The conventional Smith predictor is usually preferred for processes that contain a large transport lag and hence difficulties in control because a disturbance in set point or load does not reach the output of the process until $T_d$ units of time have elapsed. The control strategy, known as dead time compensation, may be visualized as an attempt to reduce the deleterious effect of transport lag. The transfer function of the above system shown in figure 4.1 is

$$H(s) = \frac{G_c(s)G_p(s)}{1+G_c(s)G_p(s)}$$

(4.1)
The process transfer function $G_p(s)$ shown in figure 4.1 is modeled as

$$G_p(s) = G_p^*(s)e^{-T_d s} \quad (4.2)$$

The right hand side of equation (4.2) is the product of a transport lag $e^{-T_d s}$ and a transfer function $G_p^*(s)$ which has minimum phase characteristics. If a step change is made in the input $R(s)$, the disturbance will not break through and appear at $H(s)$ until $T_d$ units of time have elapsed. Upto time $T_d$, no control action occurs, with the result that the overall closed loop response will be sluggish and generally unsatisfactory. To overcome this difficulty, Smith has suggested that $G_p(s)$ be modeled according to equation (4.2) and that an additional feedback path be inserted into the system as shown in figure (4.2).

![Figure 4.1 General Feedback System](image)

![Figure 4.2 Feedback system with dead time compensation](image)
The transfer function of the feedback system shown in figure 4.2 is

\[ \frac{H(S)}{R(S)} = \frac{G_c G_p}{1 + G_c G_m} \frac{1}{1 + G_c G_p} \]

(4.3)

The characteristic equation of the system shown in figure 4.2 is

\[ 1 + G_c G_m + G_c G_p = 0 \]

(4.4)

But the characteristic equation of a delay free system is

\[ 1 + G_c G_p^* = 0 \]

To nullify the effect of dead time the characteristic equation of the system shown in figure 4.2 must have a characteristic equation \( 1 + G_c G_p^* = 0 \)

Therefore the required condition is:

\[ G_m + G_p = G_p^* \]

\[ G_m + G_p^* e^{-\tau \omega} = G_p^* \]

\[ G_m = G_p^* (1 - e^{-\tau \omega}) \]

(4.5)

Here an additional block \( G_m \) is introduced parallel to the controller and is designed so as to mask the effect of dead time from the process. The above derivation show that the dead time is does not appear in the characteristic equation. The system is now much easier to control because the transport lag is not present in the loop. Of course, in the real system, the transport lag is still present; one has eliminated it in a mathematical sense from the feedback path by the additional feedback path of figure
4.2 and the assumption that the process transfer function $G_p(s)$ can be modeled exactly as shown in equation (4.5).

By the above dead time compensation method, one has moved the dead time out of the process loop as shown in the figure 4.3. This is feasible only if the model of the plant is exactly known. The dead time compensation does not depict any physical reality since the feedback signal in the equation (4.5) is not measurable in a real system with dead time.

![Figure 4.3 Result of dead time compensation](image)

The dead time compensator predicts the delayed effect that the manipulated variable will have on the process output. This prediction is possible only if the model for the dynamics of the process is available. In most process control problems, the model of the process is not perfect. This introduces a large modeling error and less effective compensation. The error in estimating the dead time is crucial for ideal compensation as the dead time appears as an exponential function.

### 4.2 CONVENTIONAL SMITH PREDICTOR

In terms of controller hardware implementation figure 4.2 can be redrawn as figure 4.4. Until the appearance of microprocessor based controllers, it was very difficult to realize the Smith predictor (figure 4.4). The transfer function of the
The conventional Smith predictor shown in figure 4.4 consists of a PI controller $G_{PI}$, a delay-free process model $G_p^*$, and an estimated time delay term $e^{-sT_{	ext{est}}}$. In equation (4.6), $H(s)$, $R(s)$ and $D(s)$ denote the process output, reference and disturbance respectively. If the process model $G_p^* e^{-sT_{	ext{est}}}$ is identical to the process $G_p e^{-sT}$, then the time delay disappears from the characteristic equation of the closed loop system. But if there is mismatch between the process and the model it gives rise to an additional error.

Figure 4.4 Closed loop system with the Smith predictor
The transfer function of the feedback system shown in figure 4.4 is

\[
\frac{H(s)}{R(s)} = \frac{G_p G_p^* e^{-sT_m}}{1 + G_p (G_p^* + G_p^* e^{-sT_m} - G_m e^{-sT_m})}
\]

(4.6)

Consequently, when the process time delay is significantly greater than the process dominant time constant, faster reference tracking and disturbance rejection are easily achieved compared to an ordinary PI controller. However, the robustness to variations in process parameters deteriorates in comparison to the PI controller (Vrecko et al. 2001).

### 4.3 DOUBLE CONTROLLER SCHEME

The main drawback in the conventional Smith predictor is that the robustness to variations in process parameters is poor. To overcome this, a Double Controller Scheme (DCS) is proposed by Tian and Gao (1998) as shown in figure 4.5. The DCS consists of two controllers: one to reject the load and the other to track the set point. The closed loop transfer function of the DCS is shown in equation (4.7). It is seen from figure 4.5 that when the model \( G_m e^{-sT_m} \) perfectly matches the process \( G e^{-sT_m} \), the servo and regulatory responses are controlled independently by the PI controllers \( G_{pi1} \) and \( G_{pi2} \) respectively. It is also seen from figure (4.5) that the mismatch between model and process is interpreted as an additional disturbance. Hence, the robustness for this scheme is the same as for a conventional PI controller. Unfortunately, the disturbance rejection deteriorates in comparison to Smith predictor (Vrecko et al. 2001).
The transfer function of the feedback system shown in figure 4.5 is

\[
H(s) = \frac{G_{P1} G_e^{-sT_d} (1 + G_{P21} G_m e^{-sT_{av}})}{(1 + G_{P11} G_m)(1 + G_{P12} G_m)}
\] (4.7)

To get better disturbance rejection, a Smith predictor can be used and to get a good reference tracking, a DCS can be used. To achieve a compromise between the two, certain modifications in Smith predictor are proposed (Vrecko et al 2001) which is called as a Modified Smith PI (MSPI) control scheme and it serves as a special control scheme to get a trade off between reference tracking and disturbance rejection.
4.4 MODIFIED SMITH PI CONTROL SCHEME

The conventional Smith predictor and DCS have their own disadvantages with respect to reference tracking and disturbance rejection. In the DCS, the reference tracking and disturbance rejection are designed independently through two PI controllers. Unfortunately, disturbance rejection deteriorates in comparison to conventional Smith predictor. To overcome this deficiency, the MSPI control scheme is proposed by (Vrecko et al 2001) and its main contributions are two fold. Firstly, compared to the DCS, the MSPI is richer in structure. This in turn provides an extra freedom in deciding between disturbance rejection and robustness. This can easily be achieved in a transparent manner using a single free parameter. Secondly, the MSPI structure entails two PI controllers. The tuning of MSPI controller is based on the tuning procedure of Vrancic (Vrecko et al 2001). This procedure combines magnitude optimum tuning criterion with process description using multiple integrals of step response. The result of this combination yields simple tuning rule.

Figure 4.6 shows the MSPI controller and the closed loop transfer function is given in equation (4.8). It consists of two controllers as in the case of the DCS but the main difference is that an additional loop is introduced parallel to the secondary controller. In other words, an additional Smith predictor is used instead of a single controller $G_{pi}$. It also consists of a delay free model $G_{pi}'$ and two pure delay terms i.e., $e^{-\tau}$ and $e^{-\tau'}$. Here $e^{-\tau}$ is fixed and the value of $l_x$ in $e^{-\tau'}$ is varied to achieve the required trade off between reference tracking and disturbance rejection.

If the process model $G_{pi}' e^{-\tau}$ is the same as $G_{pi}' e^{-\tau'}$ then the reference tracking of the MSPI controller becomes practically independent of $l_x$. On the other hand, by varying $l_x$ in the interval $0 \leq l_x \leq T_{sm}$ while keeping $l_m=T_{sm}$, disturbance rejection and
robustness can be made adjustable. If \( l_x = 0 \) and \( G_{ph} = G_{pi2} \) then the MSPI controller reduces to conventional Smith PI controller featuring fast disturbance rejection but poor robustness. On the contrary, \( l_x = T_{em} \) results in slower disturbance rejection in DCS on account of better robustness. Thus in a very transparent manner, trade-off between disturbance rejection and robustness can be achieved using a single free parameter.

The transfer function of the feedback system shown in figure 4.6 is

\[
G(s) = \frac{G_{pi1}G_p e^{-sT}}{(1+G_{pi1})G_p e^{-sT} + G_{pi1}G_{pi2} e^{-sT}}
\]  

(4.8)
4.5 PROPOSED DCS

In the DCS the error between the process variable and model output is given as error to the second controller and to find predicted model output, a delay is introduced in the output of the delay free model. If the delay in the output of the model is removed then the output of model becomes the predicted process variable ahead of dead time. If one can analytically predict the future variable ahead of dead time from the output of process variable then the model mismatch error ahead of dead time can be predicted. Therefore there is a scope for improvement in the response (Chidambaram et al 2003a). The simulation responses show the improvement.

The process variable ahead of dead time can be extrapolated by using Newton’s extrapolation formula. This formula is used in statistics to predict future population, future sales in a company etc.

Process variable ahead of dead time \( h(t+T_d) = \) present value \( h(t) + \) predicted future increment in \( h(t) \) ahead of dead time.

This can be written as:

\[
h(t + T_d) = h(t) + \frac{dh}{dt} * T_d
\]

(4.9)

The \( dh/dt \) can be visualized as the velocity of the process. The velocity of the process variable is not uniform under transients and becomes zero at steady state. In a stable process, the process variable will converge towards set point for increase or decrease in set point or load. As the process variable converges towards the set point, the velocity gradually decreases and ceases to zero at steady state. Therefore, under
transients there will be a definite decrease in velocity after duration of dead time. To consider this decrease in velocity, a tuning factor \( \alpha \) is introduced in the above equation. The equation (4.9) can be rewritten by including tuning parameter\( \alpha \) as follows:

\[
h(t + T_d) = h(t) + \frac{dh}{dt} * T_d * \alpha
\]

(4.10)

where \( \alpha \) is a constant and selected such that \( 0 < \alpha < 1 \). In the present study it is taken to be 0.5. To get the future values, a PID controller can be employed as a predictor as follows: When \( K_c=1, K_i=0 \) and \( K_d=\text{dead time} \times 0.5 \) then the PID controller will give the predicted future value. At steady state, \( \frac{dh}{dt} = 0 \) and therefore the equation (4.10) becomes \( h(t) = h(t+T_d) \). This shows that the controller tracks the set point without any steady state error.

![Figure 4.7 Proposed Double Controller Scheme (PDCS)](image-url)
4.6 RESULTS AND DISCUSSION

The servo and regulatory responses (Figures 4.8 to 4.15) using ZN-PI, Smith PI, MSPI controllers and DCS are obtained to show the effectiveness of the proposed controller for the transfer function model \( G(s) = \frac{1.74e^{-38s}}{25.8s + 1} \). The process considered for simulation has dominant time delay with \( T_d/T_p = 1.5 \) where \( T_d \) is the dead time of the process and \( T_p \) is the time constant of the process. All dead time compensating controllers, including the proposed controller are tuned using Haalman’s tuning rule suggested by Tian and Gao (1998) but the ZN-PI controller is tuned using Ziegler-Nichols rule. The tuning factor \( \alpha \) is assumed as 0.5. Figures 4.8 and 4.9 show the servo and regulatory responses with perfect model match. Both servo and regulatory responses with proposed controller show superior performance over other dead time compensating controllers. Similarly the servo and regulatory responses (Figures 4.10 and 4.11) with model mismatch for a 50% increase in process time constant show that the proposed controller performs better than other dead time compensating controllers. Figures 4.12 and 4.14 showing the regulatory responses with model mismatch due to 50% increase in process gain and 50% increase in dead time respectively validate the superiority of proposed controller. But the servo responses as shown in figures 4.13 and 4.15 with proposed controller and model mismatch due to 50% increase in process gain and 50% increase in dead time respectively fail to give better ISE and IAE. Under a mismatch due to 50% increase in process gain, the Smith PI gives better dynamics for servo response. Similarly under a mismatch due to 50% increase in dead time, the MSPI gives improved servo response. The simulation results of transfer function model and performance indices show that the proposed controller outperformed all compensating controllers and the ZN-PI controller except for two servo responses (figures 4.13 and 4.15) with model mismatch.
Figures 4.16 to 4.29 show the servo and regulatory responses of the conical tank level process at different operating points (39%, 24% and 54%) but all of them tuned at the nominal operating point of 39%. From the simulation results of transfer function model (Figures 4.8 to 4.15), it is found that the performances of the dead time compensating controllers vary with respect to the model mismatch. Performances of the dead time compensating controllers are better than that of the ZN-PI controller when the process has $T_d/T_p$ ratio far greater than unity (O’Dwyer, 1999). Therefore the performance of the proposed controller is tested for very small load change (-2%) and very small set point change (-1%) as shown in figures 4.16 and 4.17 so that the process variable is in the approximate linear range and the model mismatch is negligible. For analysis of performance of the controllers, the respective ISE and ISE of the ZN-PI controllers are assumed as 100%. The proposed controller rejects the load better than the ZN-PI controller (5% decrease in ISE and 4% decrease in IAE) even though the $T_d/T_p$ ratio is less than one. The ZN-PI controller gives (figure 4.17) better servo response than all dead time compensating controllers including the proposed one.

Regulatory responses for 15% decrease in load at nominal operating point 39% as in figure 4.18 show that the proposed controller gives an improved response (10% lesser ISE) while other controllers (MSPI (90% higher ISE), DCS (20% higher ISE) and conventional Smith predictor (150% higher ISE)) give a poor performance than the ZN-PI controller.

Regulatory responses for 15% increase in load at nominal operating point 39% as in figure 4.19 show that the proposed controller gives an improved response (6% lesser ISE) while other controllers (MSPI (100% higher ISE), DCS (23% higher ISE) and conventional Smith predictor (172% higher ISE)) give a poor performance than the ZN-PI controller.
Regulatory responses for 15% decrease in load at 24% operating point but tuned at 39% as in figure 4.20 show that the proposed controller gives improved response (58% lesser ISE) than MSPI (12% lesser ISE), DCS (30% lesser ISE) and conventional Smith predictor (19% higher ISE). The ZN-PI controller gives an oscillatory response.

Regulatory responses for 15% increase in load at 24% operating point but tuned at 39% as in figure 4.21 show that the proposed controller gives an improved response (30% lesser ISE) than MSPI (20% higher ISE), DCS (12% lesser ISE) and conventional Smith predictor (62% higher ISE). The ZN-PI controller gives an oscillatory response.

Regulatory responses for 15% decrease in load at 54% operating point but tuned at 39% as in figure 4.22 show that the proposed controller gives a slightly poor performance (same ISE but 12% higher IAE) than the ZN-PI controller but improved response than MSPI (120% higher ISE), DCS (27% higher ISE) and conventional Smith predictor (205% higher ISE).

Regulatory responses for 15% increase in load at 54% operating point but tuned at 39% as in figure 4.23 show that the proposed controller gives slightly poor performance (1% higher ISE) than the ZN-PI controller but improved response than MSPI (124% higher ISE), DCS (28% higher ISE) and conventional Smith predictor (209% higher ISE).

Servo responses for 20% decrease in set point at nominal operating point 39% as in figure 4.24 show that the proposed controller gives an improved response (46% lesser ISE) than the MSPI (11% lesser ISE), DCS (38% lesser ISE) and
Conventional Smith predictor (39% lesser ISE). The ZN-PI controller gives oscillatory response.

Servo responses for 20% increase in set point at nominal operating point 39% as in figure 4.25 show that the proposed controller gives a poor response (22% higher ISE) than the ZN-PI controller but gives an improved response than the MSPI (126% higher ISE), DCS (44% higher ISE) and conventional Smith predictor (50% higher ISE). Servo responses for 20% decrease in set point at 24% operating point but tuned at 39% as in figure 4.26 show that the proposed controller improve the response (5% lesser ISE) than the ZN-PI controller, MSPI (44% higher ISE), DCS (12% higher ISE) and conventional Smith predictor (11% higher ISE).

Servo responses for 20% increase in set point at 24% operating point but tuned at 39% as in figure 4.27 show that the proposed controller gives a poor response (21% higher ISE) than the ZN-PI controller but gives an improved response than the MSPI (119% higher ISE), DCS (48% higher ISE) and conventional Smith predictor (43% higher ISE).

Servo responses for 20% decrease in set point at 54% operating point but tuned at 39% as in figure 4.28 show that the proposed controller gives a poor response (4% higher ISE) than the ZN-PI controller but gives an improved response than the MSPI (85% higher ISE), DCS (21% higher ISE) and conventional Smith predictor (25% higher ISE). Servo responses for 20% increase in set point at 54% operating point but tuned at 39% as in figure 4.29 show that the proposed controller gives a poor response (21% higher ISE) than the ZN-PI controller but gives an improved response than the MSPI (112% higher ISE), DCS (40% higher ISE) and conventional Smith predictor (49% higher ISE).
Figure 4.8 Regulatory responses of \( G(s) = \frac{1.74 e^{-1.38s}}{25.8s + 1} \)
Figure 4.9 Servo responses of \( G(s) = \frac{1.74 e^{-3s}}{25.8s + 1} \)

(The responses 2, 3, 4 are super imposed)

<table>
<thead>
<tr>
<th></th>
<th>ISE</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88</td>
<td>173</td>
</tr>
<tr>
<td>2</td>
<td>51</td>
<td>64</td>
</tr>
<tr>
<td>3</td>
<td>51</td>
<td>64</td>
</tr>
<tr>
<td>4</td>
<td>51</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>49</td>
<td>61</td>
</tr>
</tbody>
</table>
Figure 4.10 Regulatory responses of $G(s) = \frac{1.74 e^{-3t}}{25.8s + 1}$ for model mismatch (50% increase in process time constant).
Figure 4.11 Servo responses of $G(s) = \frac{1.74 e^{-0.3 s}}{25.8s + 1}$ for model mismatch (50% increase in process time constant).
Figure 4.12 Regulatory responses of \( G(s) = \frac{1.74 e^{-38s}}{25.8s + 1} \) for model mismatch (50% increase in process gain).
Figure 4.13 Servo responses of $G(s) = \frac{1.74 \cdot e^{-25.8s}}{25.8s + 1}$ for model mismatch
(50% increase in process gain)
Figure 4.14 Regulatory responses of $G(s) = \frac{1.74 e^{-0.8s}}{25.8s + 1}$ for model mismatch (50% increase in dead time)
Figure 4.15 Servo responses of $G(s)=\frac{1.74 e^{-4s}}{25.8s + 1}$ with model mismatch (50% increase in dead time).
Figure 4.16 Regulatory responses of conical tank level process for very small (-2%) load change.
Figure 4.17 Servo responses of conical tank level process for a small step change of -1%.
Figure 4.18 Regulatory responses of conical tank level process for -15% load change at nominal operating point 39%
Figure 4.19 Regulatory responses of conical tank level process for +15% load change at nominal operating point 39%.
Figure 4.20 Regulatory responses of conical tank level process for -15% load change at 24% operating point.
Figure 4.21 Regulatory responses of conical tank level process for +15% load change at 24% operating point.
Figure 4.22 Regulatory responses of conical tank level process for -15% load change at 54% operating point.
Figure 4.23 Regulatory responses of conical tank level process for 15% load change at 54% operating point.
Figure 4.24 Servo responses of conical tank level process for -20% set point change at nominal operating point 39%.
Figure 4.25 Servo responses of conical tank level process for +20% set point change at nominal operating point 39%. 

<table>
<thead>
<tr>
<th></th>
<th>ISE</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23670</td>
<td>1895</td>
</tr>
<tr>
<td>2</td>
<td>53610</td>
<td>4350</td>
</tr>
<tr>
<td>3</td>
<td>34080</td>
<td>2689</td>
</tr>
<tr>
<td>4</td>
<td>35480</td>
<td>3029</td>
</tr>
<tr>
<td>5</td>
<td>29029</td>
<td>2321</td>
</tr>
</tbody>
</table>
Figure 4.26 Servo responses of conical tank level process for -20% set point change at 24% operating point.
Figure 4.27 Servo responses of conical tank level process for +20% set point change at 24% operating point.
Figure 4.28 Servo responses of conical tank level process for -20% set point change at 54% operating point.
Figure 4.29 Servo response of conical tank level process for 20% set point change at 54% operating point.
4.7 CONCLUSION

The performances of existing dead time compensating controllers for the chosen process do not give satisfactory results than the ZN-PI controller since the performance of dead time compensating controllers depend on the $T_d/T_p$ ratio, model mismatch and controller parameters (O’Dwyer 1999). The $T_d/T_p$ ratio of the chosen process varies form 0.11 to 2.05 as the level varies from top to bottom. Similarly the time constant varies from 16.6 to 316 seconds and the gain varies from 0.78 to 1.58. Therefore the variation in the process parameters (as the level varies) introduces model mismatch. The compensating controllers give satisfactory results only when $T_d/T_p$ ratio is greater than 1. Therefore the performance degrades for operation near the top of the tank. The model obtained at 39% of the level is used for the compensating controllers. The compensating controllers are tuned using this model only. Therefore the poor tuning parameters and model mismatch affect the performance for 24% operating point. Similarly small $T_d/T_p$ ratio affects the performance at 54% and 39% operating points.

The proposed controller anticipates the model mismatch ahead of dead time and hence gives superior performance compared with all other existing dead time compensating controllers. The performances are not improved as expected compared with the ZN-PI controllers for some servo responses at 54% and 39% operating points due to very small $T_d/T_p$ ratio. Therefore it is necessary to retune the controller to provide satisfactory response at newer operating points. Hence there is a need to consider model nonlinearity. One of the ways is to update the linear model parameters and retune the controller (gain scheduling of PI controller). The other way is using a nonlinear model based controller. In the next chapter, an attempt has been made to utilize the local gain and local time constant to auto tune the controller by gain scheduling approach.