

## AUTOMATIC TUNING OF MODEL PREDICTIVE CONTROLLERS BASED ON FUZZY LOGIC

EMAD ALI

Chemical Engineering Department, King Saud University, P.O.Box 800, Riyadh 11421, Saudi Arabia.

Email: [amkamal@ksu.edu.sa](mailto:amkamal@ksu.edu.sa), fax:++(9661)467-8770

### ABSTRACT

This paper presents an online tuning strategy for model predictive control. Specifically, the tuning strategy adjusts automatically the prediction horizon,  $P$ , the diagonal elements of the input weight matrix,  $\Lambda$ , and the diagonal elements of the output weight matrix,  $\Gamma$ . The control horizon is left constant because its relative value with respect to  $P$  is more important. The MPC parameters are adjusted such that the resulted feedback response satisfies certain time-domain performance specification. The tuning algorithm is based on fuzzy logic. Predefined fuzzy rules that formulate the general tuning guidelines available in the literature and the performance violation measure in the form of fuzzy sets determine the new tuning parameter values. Therefore, the tuning algorithm is cast as a simple and straightforward mechanism with modest computational requirements. The effectiveness of the proposed tuning method is tested through simulated implementation on an evaporator example.

**Keywords:** Model Predictive Control, Online tuning, Fuzzy logic, performance specs.

### INTRODUCTION

Model Predictive Control (MPC) is a well-known control algorithm, which has received considerable acceptance in the industry due to its effectiveness in various industrial applications [1-3]. Detailed information on the history and progress of such a control algorithm can be found in a recent review paper [4]. In practice, tuning of MPC for good performance and/or stability is crucial. Basically, MPC is tuned via trial-and-error and sometimes with the aid of tuning guidelines available in the literature [5-8]. However this procedure is a cumbersome task due to the overlapping effect of the MPC tuning parameters and due to non-linearity brought by input constraints. The procedure is even more complicated for processes with large number of control loops. Consequently, advancement in MPC tuning is still an active area of research.

Recently, Shridhar and Cooper [9,10] developed an easy to use MPC tuning formula. However, their approach is limited to unconstrained MPC. Moreover, it requires representing the process by a first order plus dead-time (FOPDT) model, which may not work well for higher order and/or unstable processes. Ali and Zafiriou [11]

presented an off-line optimization-based tuning procedure of MPC algorithms that utilizes non-linear process models. Lately, Jiang and Jutan [12] have proposed another off-line tuning method. Their method solves an optimization problem with the objective function is cast as an Integral Squares Error (ISE). The two former procedures are computationally demanding and thus can not be implemented online. In addition, the proposed method in [12] tunes only two MPC parameters, which are the control horizon and a scalar multiplication factor for the diagonal elements of the move suppression matrix. Several other off-line tuning strategies for specific cases are reviewed by Shridhar and Cooper [9,10]. Most of the existing tuning methods is either general tuning guidelines based on sophisticated theoretical analysis or off-line procedures. Both can not be easily understood and implemented online by control operators. Therefore, devising a systematic and automatic tuning algorithm can be of great contribution.

An aspect of a tuning procedure that would be appealing from a practical point of view would be to allow the operator to give the performance specifications as *time-domain* envelopes, within which the closed-loop responses should lie for several different set point changes or disturbances [13]. For this purpose, Al-Ghazzawi et. al. [14] proposed an on-line adaptive strategy for constrained MPC. The algorithm utilizes the gradients of the closed-loop response with respect to the MPC parameters to automatically determine on-line the new values of the MPC parameters.

Here the above approach is modified, but without using the output sensitivity to the MPC tuning parameters. Instead, fuzzy logic will be used to determine the new values of the MPC tuning parameters. The proposed algorithm will be less computationally demanding than the previous one. Moreover, it will allow adjusting the prediction horizon, which is an integer tuning parameter. The prediction horizon can not be adjusted using the previous method because the latter is limited to real-valued variables. Therefore, the main objective of this paper is to develop a simple and straightforward *online* tuning method that can be easily understood and implemented by the practitioner. The method is computationally undemanding and thus can be solved

easily online. An added feature of the proposed method is that adjustment of the MPC parameters is based on preset time-domain desired performance.

### THE ON-LINE MPC ALGORITHM

The proposed on-line tuning strategy is developed for the well-established linear Model Predictive Controllers (LMPC) based on FIR models. A usual MPC formulation (notation follows that in [15] solves the following on-line optimization:

$$\min_{\Delta U(k)} \left\| \Gamma(Y_p(k+1) - R(k+1)) \right\|^2 + \left\| \Lambda \Delta U(k) \right\|^2 \quad (1)$$

subject to

$$Y_p(k+1) = M_p Y(k) + S_p^m \Delta U(k) \quad (2)$$

$$F^T \Delta U(k) \geq b \quad (3)$$

The symbol  $\| \cdot \|$  denotes the Euclidean vector norm and  $k$  denotes the current sampling point.  $\Gamma$  and  $\Lambda$  are diagonal weight matrices.  $R(k+1) = [r(k+1) \cdots r(k+P)]^T$  contains the desired output trajectories over horizon  $P$ .  $\Delta U(k) = [\Delta u(k+1) \cdots \Delta u(k+M-1)]^T$  is a vector of  $M$  future changes of the manipulated variable vector  $u$  to be determined by the on-line optimization.  $Y_p(k+1) = [y(k+1) \cdots y(k+P)]^T$  includes the predicted outputs over the future horizon  $P$ , where  $y$  is the output vector, assuming  $\Delta u(k+i) = 0$ ;  $i = M$ .  $Y(k) = [y(k) \cdots y(k+n-1)]^T$  includes the predicted outputs over the truncation horizon  $n$  (length of FIR) based on  $\Delta u(k+i) = 0$ ;  $i = 0$ . Equation (2) represents the output prediction based on the process model. A disturbance estimate should also be added in (2) or alternatively it can be absorbed in  $R(k+1)$ . The latter is assumed for simplicity. In this paper, the disturbance is assumed constant over the prediction horizon, and set equal to the difference between plant and model outputs at present time  $k$ . The matrices  $M_p$  and  $S_p^m$  are defined as in [15].  $M_p$  is a constant matrix consisting of ones and zeros and  $S_p^m$  is what is usually referred to as the Dynamic Matrix of step response coefficients. The MPC tuning parameters, that need to be adjusted, are the control horizon,  $M$ , the prediction horizon,  $P$ , the output weight,  $\Gamma$ , and the input weight (input-move suppression),  $\Lambda$ .

The above objective function (1) is minimized subject to constraints on the manipulated variables and on the change of manipulated variables represented by (3). No real-time dynamic output constraints over the prediction horizon will be imposed on the on-line optimization. One reason is that commercial MPC software packages do not utilize hard dynamic output constraints [3]. This is maybe because inclusion of hard constraints increases the online computational effort, which is not recommended in real-time application. Another reason is that imposing output constraints in the

on-line optimization may destabilize the feedback response [16]. In this paper, the output constraints are incorporated in the specifications of the on-line tuning strategy, so that the MPC parameters are tuned online to satisfy the output constraints without explicitly using them on-line.

### TUNING OF THE MPC PARAMETERS

The tuning technique adapts on-line the MPC parameters in order to steer the closed-loop response to satisfy preset time-domain specifications. Examples of time-domain specifications are shown in Figures 5-9 by the light solid line. The shape of the performance specs can be designed to reduce overshoot, reject disturbances, and maintain proper speed of the response. The user should provide the performance specifications in the form of vectors of upper and lower bounds,  $y^u$  and  $y^l$  respectively. The performance envelope should have a specific window size. The size and magnitude of the specs should be designed by the user according to his understanding of the process dynamics. Also, the user should define a value for the closed-loop prediction horizon,  $P_w$ . More elaboration on the utilization of  $P_w$  will be discussed later.

The tuning algorithm consists of two phases namely; observation and triggering phases. In the observation phase, the algorithm monitors the closed-loop prediction of the output. If the prediction violates the preset performance specs or if the output set point is changed the algorithm switch into the triggered phase. In the triggering phase, the magnitude of performance violation and its rate are calculated from which a new value for the MPC tuning parameters is determined. The new values of the MPC parameters will be determined by a fuzzy logic system. The fuzzy logic system consists of three consecutive stages, e.g. Fuzzification, Base rules and Defuzzification. Each stage is explained in the following section. Here, it is assumed that the reader is familiar with fuzzy logic terminology and concepts.

*Fuzzification:* In this stage, a specific measured variable is transformed into a member of a set of fuzzy membership functions. Two different input sets of membership functions and one output set of membership functions are used in this paper. The first input set is shown in Figure 1. The set consists of two membership functions namely; (G)ood denoted as  $\mu_G$  and (H)igh denoted as  $\mu_H$ . The universe of discourse of this set spans the possible values for the scaled value of the bound violation. The scaled bound violation is defined as follows:

If upper bound is violated:

$$A \equiv \frac{y_j(k+m) - y_j^u(k+m)}{y_j^u(k+m)} \quad (4)$$

If lower bound is violated:

$$B \equiv \frac{y_j^l(k+m) - y_j(k+m)}{y_j^l(k+m)} \quad (5)$$

where  $y_j$  is the predicted value of the  $j^{\text{th}}$  output. Also,  $k$  is the sampling time and  $m$  is the instant at which maximum violation occurs for the  $j^{\text{th}}$  output. The  $j^{\text{th}}$  output is determined such that it has the maximum violation over all other outputs. The definition of  $A$  and  $B$  in the above equations guarantees positive value when the corresponding bound is violated and negative otherwise. Note that the upper and lower bounds can not be violated in the same time by one specific output. Therefore, if the upper bound is violated, then  $A$  belongs to  $\mu_H$  and  $B$  to  $\mu_G$  and vice versa. If neither bound is violated, then both  $A$  and  $B$  belong to  $\mu_G$ . This argument applies for the case of set point change. It also applies to disturbance case with the exception that the belonging of  $B$  to the membership functions is reversed. The reason for reversing is that the lower bound in the disturbance case is always negative because deviation variables are used in this paper.

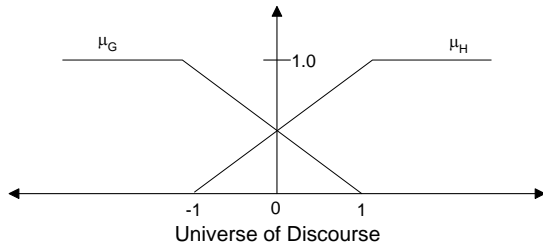


Figure 1: Fuzzy set for bound violation.

The second input set of membership functions is shown in figure 2. The set consists of three functions namely; (P)ositive, (Z)ero and (N)egative. These functions are denoted  $\mu_P$ ,  $\mu_Z$  and  $\mu_N$ , respectively. The input to this fuzzy set is the scaled violation rate, which is defined as follows:

$$C \equiv \frac{y_j(k+m) - y_j(k+m-1)}{y_j(k+m)} \quad (6)$$

The above expression for violation rate applies for both upper and lower bound violation. Note that scaled values are used for the bound violation measure ( $A, B$ ) and its rate ( $C$ ). The reason for that is to simplify the determination of the possible range for the universe of discourse in figure 1 and 2.

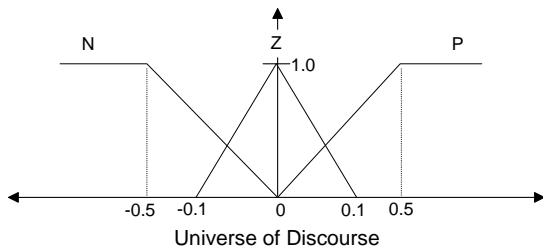


Figure 2: fuzzy set for bound violation rate

*Inference Engine:* The base rules governing the tuning guidelines is given in Table 1 for set point change and in Table 2 for disturbance rejection case. In the tables,  $\mu_\lambda$ ,  $\mu_\gamma$

and  $\mu_P$  represent the rule output for  $\lambda$ ,  $\gamma$  and  $P$ , respectively. LN, SN, ZE, SP and LP are the output membership functions as shown in figure 3 for  $\lambda$  and  $\gamma$  and  $P$ . These functions are denoted  $\mu_5$ ,  $\mu_4$ ,  $\mu_3$ ,  $\mu_2$  and  $\mu_1$  respectively. The rules given in Table 1 and 2 formulate the general understanding of the effect of  $\lambda$ ,  $\gamma$  and  $P$  on the closed-loop response. The general effect of these parameters is explained next.

It is known that increasing  $\lambda$  makes the response slower. Specifically, when  $\lambda$  is slightly increased from zero, while other parameters are fixed, the response becomes slower and starts overshooting. As  $\lambda$  is further increased, the response becomes slower and exhibits under-damped behavior. When  $\lambda$  is increased beyond a *critical* value, the under-damping behavior disappears and the response becomes very sluggish. The critical value for  $\lambda$  increases with increasing  $P$ . Building upon the above understanding, it is always preferable to reduce  $\lambda$  to speed up the response, i.e. when the lower bound of the performance specs is violated. Likewise, it is desirable to reduce  $\lambda$  in order to eliminate overshoot, i.e., when the upper bound is violated. Generally,  $\lambda$  is increased only when reducing it has no further effect. The above reasoning on  $\lambda$  is reflected in Tables 1 and 2.

Table 1: The base rules for set point case

no	Rule	Result for $\lambda$	Result for $\gamma$	Result for $P$
R1	If A is H and B is G	$\mu_\lambda$ is SN	$\mu_\gamma$ is SP	$\mu_P$ is SP
R2	If A is G and B is H	$\mu_\lambda$ is LN	$\mu_\gamma$ is SP	$\mu_P$ is SP
R3	If A is G and B is G	$\mu_\lambda$ is ZE	$\mu_\gamma$ is ZE	$\mu_P$ is ZE
R4	If A is H and C is P	$\mu_\lambda$ is LN	$\mu_\gamma$ is SP	$\mu_P$ is SP
R5	If A is H and C is Z	$\mu_\lambda$ is SP	$\mu_\gamma$ is SN	$\mu_P$ is SN
R6	If A is H and C is N	$\mu_\lambda$ is ZE	$\mu_\gamma$ is ZE	$\mu_P$ is ZE
R7	If B is H and C is P	$\mu_\lambda$ is SN	$\mu_\gamma$ is SP	$\mu_P$ is SP
R8	If B is H and C is Z	$\mu_\lambda$ is SP	$\mu_\gamma$ is SN	$\mu_P$ is SN
R9	If B is H and C is N	$\mu_\lambda$ is ZE	$\mu_\gamma$ is ZE	$\mu_P$ is ZE

Table 2: The base rules for disturbance case

no	Rule	Result of $\lambda$	Result for $\gamma$	Result for $P$
R1	If A is H and B is H	$\mu_\lambda$ is SN	$\mu_\gamma$ is SP	$\mu_P$ is SP
R2	If A is G and B is G	$\mu_\lambda$ is LN	$\mu_\gamma$ is SP	$\mu_P$ is SP
R3	If A is G and B is H	$\mu_\lambda$ is ZE	$\mu_\gamma$ is ZE	$\mu_P$ is ZE
R4	If A is H and B is G	$\mu_\lambda$ is ZE	$\mu_\gamma$ is ZE	$\mu_P$ is ZE
R5	If A is H and C is P	$\mu_\lambda$ is SN	$\mu_\gamma$ is SP	$\mu_P$ is SP
R6	If A is H and C is Z	$\mu_\lambda$ is SP	$\mu_\gamma$ is SN	$\mu_P$ is SN
R7	If A is H and C is N	$\mu_\lambda$ is ZE	$\mu_\gamma$ is ZE	$\mu_P$ is ZE
R8	If B is G and C is P	$\mu_\lambda$ is SN	$\mu_\gamma$ is SP	$\mu_P$ is SP
R9	If B is G and C is Z	$\mu_\lambda$ is SP	$\mu_\gamma$ is SN	$\mu_P$ is SN
R10	If B is G and C is N	$\mu_\lambda$ is ZE	$\mu_\gamma$ is ZE	$\mu_P$ is ZE

On the other hand, increasing  $P$  has a robust effect on the closed-loop response. According to previous

experience, it is found that, for several cases, increasing  $P$  at fixed non-zero value for  $\lambda$ , a slightly faster response with less overshoot can be obtained. Therefore, it is always desirable to increase  $P$  whenever an upper or a lower bound violation occurs.  $P$  is only reduced when increasing it has no further effect. This reasoning is also reflected on the base rules of Tables 1 and 2. The output weight  $\gamma$  is used to trade-off between different outputs. Therefore, it is preferable to increase  $\gamma$  that corresponds to the active output. It should be noted that the base rules are not limited to those listed in Table 1 and 2, thus any other well-known guidelines can also be incorporated.

*Defuzzification:* In this stage, the results of the second stage (inference engine) are combined in a special way to produce a crisp value for the output. The resulted output is the factor that will be used to adjust the MPC tuning parameters. This procedure of combining the results of the inference engine is known as defuzzification. Here the center of area (COA) defuzzification method [17] is adopted. The COA can be applied to find the output (factor) for  $\lambda$  and  $\gamma$  as follows:

$$w(z) = \frac{\sum_{j=1}^{n_R} \sum_{i=1}^{n_f} \mu_{j,i}(z) \delta_i A_{j,i}}{\sum_{j=1}^{n_R} \sum_{i=1}^{n_f} \mu_{j,i}(z) A_{j,i}} \quad (10)$$

Where  $n_R$  is the number of rules and equals 10 in this paper,  $n_f$  is the number of output membership functions and equals 5 in this paper, and  $\delta_i$  is value for the location of the center of  $\mu_i$ . The value of  $\delta_i$  is pre-calculated and fixed as shown in figure 3.  $A$  is  $n_R \times n_f$  pre-calculated matrix, which identifies which membership function is included in each Rule. For example, row 1 of matrix  $A$ , which is assigned for Rule 1, contains 1 at the first column and zeros elsewhere. The same logic is carried out over the remaining rows. The argument  $z$  denotes either  $\lambda$  or  $\gamma$ . For the prediction horizon,  $P$ , the output is calculated as follows:

$$w(P) = \sum_{j=1}^{n_R} \sum_{i=1}^{n_f} \mu_{j,i}(P) \delta_i \quad (11)$$

Since  $P$  is an integer value, the factor  $w$  is rounded to the nearest integer.

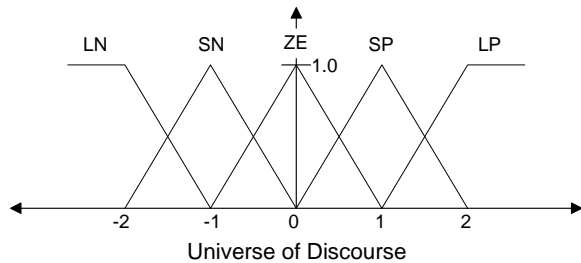


Figure 3: output fuzzy set

*MPC Parameters Adaptation Algorithm:*

The adaptation algorithm can be clearly understood by the following algorithm: At any on-line sampling point  $k$ , and before computing the control action:

**Step 1:** Predict the closed-loop response over the prediction horizon ( $P_w$ ) using equations (2) for fixed values of the tuning parameters and constant disturbance variables.

**Step 2:** Check whether the closed-loop prediction violates the specifications. If it does not, then go to step 5.

**Step 3:** Determine the output index and the sampling point at which the maximum violation of the specification occurs. Let this be for output  $j$  and point  $k+m$ .

**Step 4:** Calculate the bound violation measure ( $A, B$ ) and its rate ( $C$ ) using equations 4-6.

**Step 4.1:** Determine the degree of membership of  $A$  and  $B$  with respect to membership functions  $\mu_G$  and  $\mu_H$ . Also determine the degree of membership of  $C$  &  $B$  and  $C$  &  $A$  with respect to membership functions  $\mu_P$  and  $\mu_Z, \mu_N$ .

**Step 4.2:** Calculate the adjustment factor  $w$  using equation (10) and (11).

**Step 4.3:** Set  $\lambda_{k,i} = \lambda_{k,i} (1 + w_k(\lambda)/g_i)$ ,  $i=1, \dots, nu$ ;  $\gamma_j = \gamma_j (1 + w_k(\gamma))$ ;  $P = P + w_k(P)$ .

**Step 5:** Compute and implement the control action. Advance to the next sampling time in real-time operation and, set  $k=k+1$ . Go to Step 1.

In step 4.3 above, the correction factor for  $\lambda_i$  is scaled by the static gain of the  $i_{th}$  output. The gains are computed at a specific steady state and are not allowed to vary during the simulation. The reason for scaling is to make the resulted  $\lambda$ 's have the same order of effect on the output. The algorithm has one parameter, namely  $P_w$ . The prediction horizon is an important design parameter as it provides advance prediction of the behavior of the closed-loop response, which may result in earlier correction of the MPC parameters. The larger the value of  $P_w$ , the more robust is the tuning algorithm, but the more the computational load is. A straightforward guideline for selecting the value of  $P_w$  is given elsewhere [13,14]. Since the tuning algorithm works automatically, the manual adjustment of the MPC tuning parameters is replaced by adjusting manually a single parameter, which is  $P_w$ . Here  $P_w$  is fixed so that the tuning procedure becomes fully automatic.

It should be noted that the algorithm tunes the real variables,  $\Gamma$  &  $\Lambda$  and the integer variable  $P$ . A small fixed value for  $M$  is preferable to reduce the on-line computational requirements. Another reason for fixing  $M$  is that changing the value of  $P$  relative to  $M$  is more important and meaningful. The algorithm works for general MIMO systems.

## SIMULATION EXAMPLE

A forced circulation evaporator is shown in figure 4. The process is originally proposed by Newell and Lee [18] and is modeled as follows:

$$M_v \frac{dC_2}{dt} = F_1 C_1 - F_2 C_2$$

$$W \frac{dP_2}{dt} = F_4 - F_5$$

where  $C_1$  and  $C_2$  are the input and product compositions, respectively and  $P_2$  is the operating pressure (kPa).  $F_1$  and  $F_2$  are the feed and product flow rates (kg/min), respectively.  $F_4$  and  $F_5$  are the vapor and condensate flow rates, respectively (kg/min).  $M_v$  is the liquid holdup in the evaporator (20 kg) and  $W$  is a constant (4 kg/kPa). The liquid level in the separator is considered well controlled by manipulating the product flow rate. Therefore, the flow rates are given as follows:

$$F_2 = F_1 - F_4$$

$$F_4 = [0.16(F_1 + F_3)(-0.3126C_2 - 0.5616P_2 + 0.1538P_{100} + 41.57) - F_1 C_p (0.3126C_2 + 0.5616P_2 + 48.43 - T_1)] / \lambda_{s1}$$

$$F_5 = \frac{2UA(0.507P_2 + 55 - T_{200})C_p F_{200}}{\lambda_{s2}(UA + 2C_p F_{200})}$$

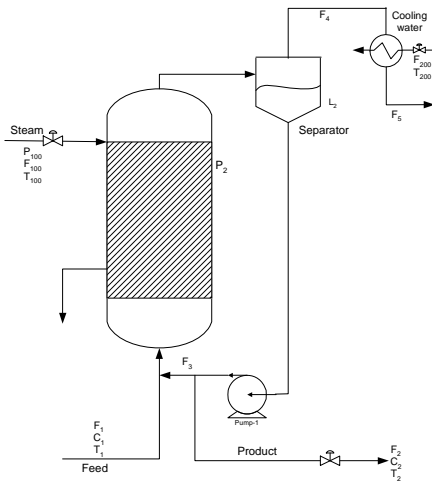


Figure 4: Evaporator process

where  $P_{100}$  is the steam pressure (kPa).  $F_{200}$  (kg/min) and  $T_{200}$  ( $^{\circ}$ C) are the flow and temperature of the cooling water, respectively.  $UA$  is the product of heat transfer coefficient and the transfer area (6.84 kW/K) and  $C_p$  is the heat capacity of the cooling water (0.07 kW/kg min).  $F_3$  is the recycle flow rate (20 kg/min).  $\lambda_{s1}$  is the latent of steam at saturation condition (36.6 kW/Kg min) and  $\lambda_{s2}$  is the latent heat of evaporation of water (38.5 kW/kg min).  $F_1$  is fixed at 10 kg/min and its temperature  $T_1$  is fixed at 40  $^{\circ}$ C. The cooling water enters the cooler at 25  $^{\circ}$ C. The initial steady state value for the outputs is  $C_2 = 0.1474$  and  $P_2 = 32.109$  kPa which corresponds to  $C_1 = 0.05$ ,  $P_{100} = 200$  kPa and  $F_{200} = 200$  kg/min. The control objective is to maintain the outputs within desired values using  $P_{100}$  and  $F_{200}$  as manipulated variables. This control problem is

selected because it has strong cross-loop interaction. In addition, because mass fraction is used for  $C_1$  and  $C_2$ , the process is ill conditioned, which makes the problem even more challenging.

A step response model is generated from the nonlinear process model using MATLAB software with truncation number,  $n = 30$  and sampling time  $T_s = 0.5$  min. The step response model is used in the MPC controller and in the tuning algorithm. The full nonlinear model is used to simulate the plant from which the output measurement will be obtained. This formulation generates model-plant mismatch, which makes the control problem more difficult. The nominal performance envelope for set point change is designed such that it limits the overshoot to 8% for the first 30 samples, brings the response to within  $\pm 5\%$  for the following 40 samples and finally to within  $\pm 1.1\%$  for the last 40 samples for the first output. For the second output the envelope limits the overshoot to 5% for the first 30 samples, brings the response to within  $\pm 2\%$  for the following 40 samples and finally to within  $\pm 1.1\%$  for the last 40 samples. The nominal performance envelope for disturbance rejection is designed such that it limit the overshoot to 2% for the first 30 samples, brings the response to within  $\pm 0.5\%$  for the following 40 samples and finally to within  $\pm 0.2\%$  for the last 40 samples for both outputs. The arbitrary initial values for the MPC parameters and those obtained using Shridhar and Cooper formula [10] are given in Table 3. Note that the diagonal elements of  $\Lambda$  are constrained between 0 and 1, the diagonal elements of  $\Gamma$  are constrained between 1 and 100, and that  $P$  is constrained between  $M$  and  $n$ . The closed-loop prediction horizon,  $P_w$  is fixed at 10 samples in all simulations. The response for the manipulated variables will be excluded in the following simulation plots for space limitation.

Table 3: Tuning parameters for evaporator example

Parameter	Arbitrary	Shridhar & Cooper
$M$	1	14
$P^0$	2	59
$\Gamma^0$	Diag [1,1]	Diag [1,1]
$\Lambda^0$	Diag [0.01,0.01]	Diag[0.02, 0.06]

Figure 5 shows the response of the process outputs to set point change of [0.0326,1] as deviation from the initial steady state. The light solid line represents the performance specs. The solid line represents the closed-loop response for fixed MPC parameters at the arbitrary values given in Table 3. The dashed line shows the closed-loop response using MPC with the proposed tuning method. In this case, the arbitrary values for the MPC parameters are used as the initial guess. It is clear from the figure that the adapted response outperforms the non-adapted one in the sense of faster response especially for the product composition. Although the pressure response is slightly degraded and that the composition

response is initially oscillatory, the overall response is considered acceptable.

Figure 5 also shows how the tuning parameters change to improve the feedback response. Obviously, the tuning method decreased  $\Lambda$  to speed up the response and increased  $P$  to maintain stable behavior. Figure 7 depicts the output behavior for disturbance rejection while the process is operating at the initial steady state. The disturbance is introduced as  $-5^\circ\text{C}$  step change in both  $T_1$  and  $T_{200}$  and occurs after 2.5 minutes from the start of the simulation. In this simulation, the adapted response uses the arbitrary MPC parameter values as the initial guess. Whereas the non-adapted response uses fixed arbitrary values for the MPC parameters. It is clear that the MPC response for the first output with fixed parameters is very sluggish and suffers from large overshoot. On the other hand, the tuning method managed to substantially improve the response of the first output (composition) to fit within the desired performance specs. This is achieved at smaller  $\Lambda$  and slightly larger  $P$ .

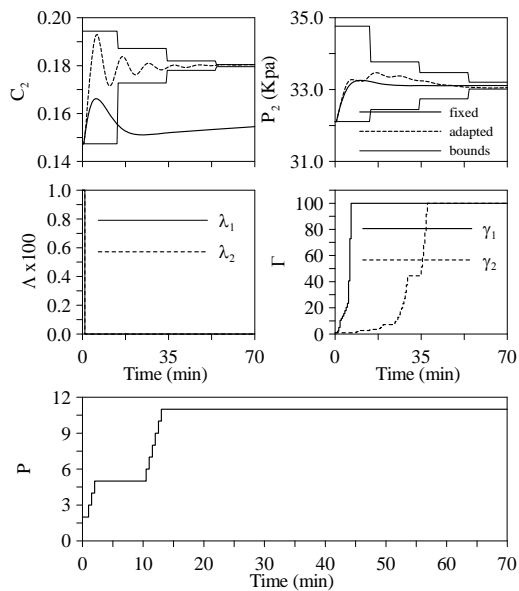


Figure 5: Response to set point change of [0.0326, 1].

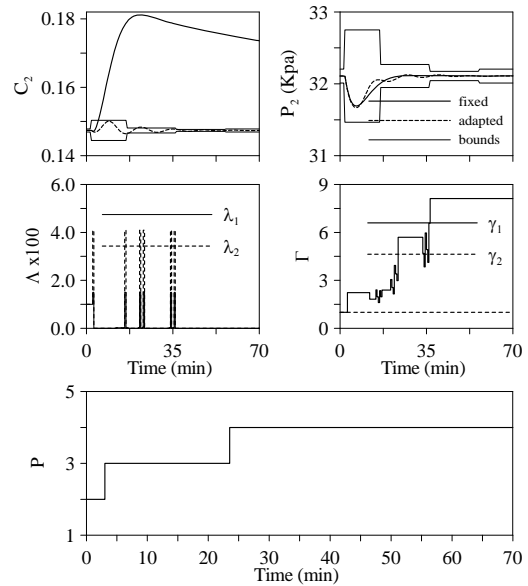


Figure 6: Response to sep change in  $T_1$  &  $T_{200}$  of [-5, -5].

Figure 7 demonstrates the feedback response to set point change of [0.0326,0], which is expected to create difficulty due to the cross-loop interaction. In addition, the MPC parameter values determined by Shridhar & Cooper (SC) formula listed in Table 3 are used as the initial guess. The solid line in Figure 7 shows the feedback response using fixed MPC parameters determined by the SC formula. In that case, the pressure response is perfect, while that for the composition is very sluggish. By the aid of the proposed tuning method the feedback performance for the product composition is enhanced drastically as shown by the dashed line in Figure 7. However, this improvement was at the cost of slight degradation in the pressure response. The overall performance can be further improved when larger  $P_w$  is used; however,  $P_w$  is fixed in all simulations for fair comparison. Moreover,  $P_w$  is fixed to keep the number of manually adjustable parameters to minimum.

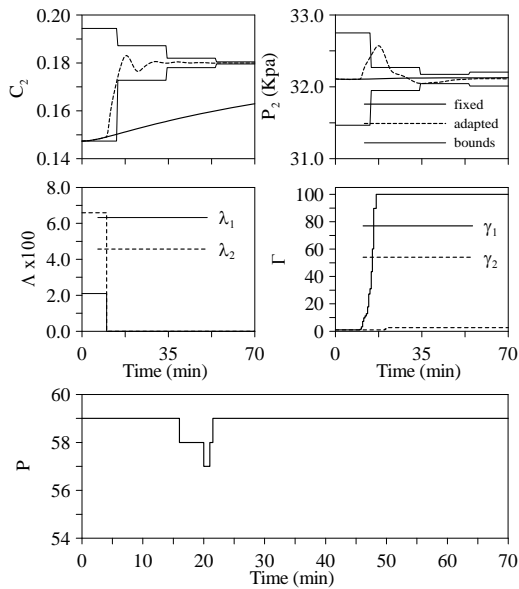


Figure 7: Response to set point change of [0.0326, 0].

Figure 8 illustrates the feedback response to step change of  $-5\text{ }^{\circ}\text{C}$  in  $T_1$  and  $+5\text{ }^{\circ}\text{C}$  in  $T_{200}$ . The MPC parameters obtained by the SC formula are used in this simulation for both the adapted and non-adapted response. As demonstrated in figure 8, the adapted response for both outputs delivered improved performance over that obtained with fixed MPC parameters. Despite the minor violation of the lower bound for the first output, the overall performance is acceptable and much better than that without re-tuning. The results shown in Figs. 7 & 8 showed that the adapted MPC parameters provide better performance than that obtained using the fixed SC parameters. This is because the SC formula provides conservative values for the MPC parameters to ensure robust stability.

Finally, Figure 9 depicts the feedback response for three consecutive set point changes, each of which spans 55 minutes. The first set point change is [0.0326,2], the second is [-0.02, -1] and the third is [0.02,1] as deviation from the initial steady state. The arbitrary values for the MPC parameters are used in this simulation. There is no doubt that the adapted response outperforms the non-adapted one. Despite some violation of the performance specs, the performance of the adapted MPC delivered faster response and the first output reached its set point within the given simulation time. On the other hand, the non-adapted response is so slow that it the first output can not reach its set point within the desired time. The profile for the MPC parameters is not shown here to save space.

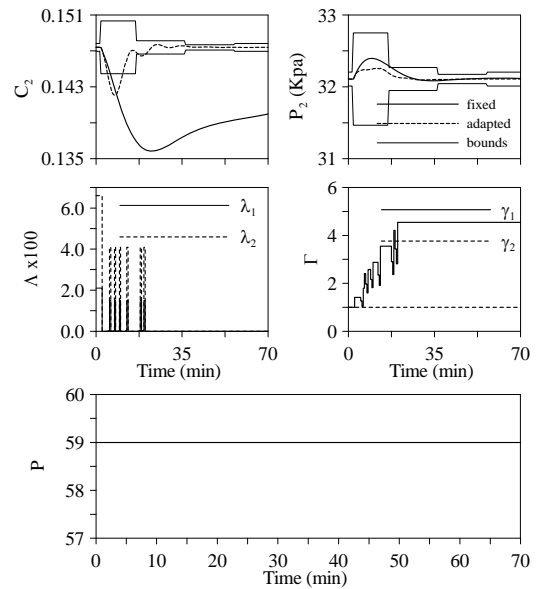


Figure 8: Response to sep change in  $T_1$  and  $T_{200}$  of [-5, +5]

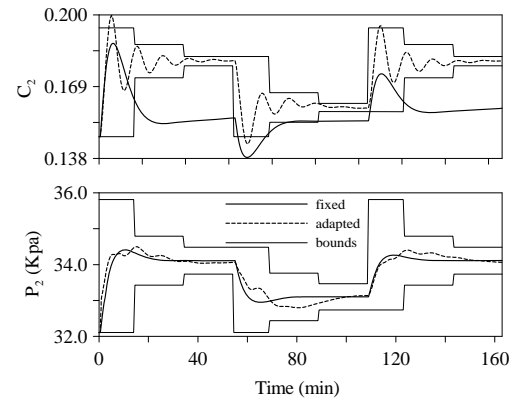


Figure 9: Response to a series of set point changes.

## CONCLUSION

This paper considers the problem of fine-tuning the model predictive controller for good performance. The tuning method formulates the general tuning guidelines available for MPC algorithms into an automatic mechanism. This mechanism adapts the MPC parameters automatically online such that the resulted closed-loop response fits certain time-domain performance specs. The mechanism is based on fuzzy logic, which requires minimal computational load. The effectiveness of the proposed tuning method is tested through a simulated implementation on an evaporator model. The simulations revealed that, whether arbitrary or rigorous values for the MPC parameters were used as initial guess, the proposed tuning method could enhance the feedback performance substantially. Note that the proposed tuning method has a single adjustable parameter namely  $P_w$ . This parameter is fixed in this paper to eliminate creating additional tuning parameters. This is obvious through the ability of the proposed method to provide good performance over different operating conditions for the same fixed value of

$P_w$ . However, the tuning method as it stands does not provide robust stability.

## REFERENCES

- [1] J. Richalet, Rault, A., Testud, J., and Papon, J., Model Predictive Heuristic Control: Application to Industrial Processes, *Automatica*, 14, 1987, 413-423.
- [2] C. R. Cutler, and Hawkins, R. B., Application of a Large Predictive Multivariable Controller to a Hydrocracker Second Stage Reactor, *Proc. of ACC*, Atlanta, GA, 1988, 284-291.
- [3] S. J. Qin, and T. A. Badgwell, An Overview of Industrial Model Predictive Technology, in *Fifth International Conference on Chemical Process Control*, AIChE Symposium Series 316, 93, Jeffrey C. Kantor, Carlos E. Garcia, and Brice Carnahan, eds., 232-256, 1997.
- [4] M. Morari, and Lee, J. Model Predictive Control: Past, Present, and Future, *Comp. Chem. Eng.*, 23, 1999, 667-682.
- [5] A. Morshedi, and Garcia, C., Quadratic Programming Solution of Dynamic Matrix Control (QDMC), *Chem. Eng. Comm.*, 46, 1986, 73-87.
- [6] C. Garcia, and Morari, M., Internal Model Control 1: A Unifying Review and Some New Results, *Ind. Eng. Chem. Proc. Des. Dev.*, 21, 1982, 309-323.
- [7] H. Ohno, H., Ohshima, M., and Hashimoto, I., A Study on Robust Stability of Model Predictive Control, *IFAC Model Based Process Control*, GA, USA, 1988.
- [8] E. S. Meadows, and Rawlings, J. B., *Model Predictive Control*, In: Henson, M. A., and Seborg, D. E., (Eds), *Nonlinear Process Control*. Prentice-Hall, Englewood Cliff, NJ, 1997.
- [9] R. Shridhar, and Cooper, D. J., A Tuning Strategy for Unconstrained SISO Model Predictive Control, *Ind. & Eng. Chem. Res.*, 36, 1997, 729-746.
- [10] R. Shridhar, and Cooper, D. J., A Tuning Strategy for Unconstrained Multivariable Model Predictive Control, *Ind. & Eng. Chem. Res.*, 37, 1998, 4003-4016.
- [11] E. Ali, and Zafiriou, E., Optimization-based Tuning of Non-linear Model Predictive Control with State Estimation, *J. of Process Control*, 3, 1993, 97-107.
- [12] A. Jiang, and Jutan A., Response Surface Tuning Methods in Dynamic Matrix Control of a Pressure Tank system, *Ind. Eng. Chem. Res.*, 39, 2000, 3835-3843.
- [13] E. Ali, Optimization-Based Tuning Strategies for Linear and Non-linear Model Predictive Controllers, Ph.D. thesis, University of Maryland, College Park, MD, 1995
- [14] A. Al-Ghazzawi, Ali, E., Nouh, A., and Zafiriou, E., On-line Tuning Strategy for Model Predictive Controllers, *J. of Process Control*, 11(3), 2001, 265-284.
- [15] J. H. Lee, and Yu, Z. H., Tuning of Model Predictive Controllers for Robust Performance, *Comp. and Chem. Eng.*, 18, 1994, 15-37.
- [16] E. Zafiriou, and Marchal, A., Stability of SISO Quadratic Dynamic Matrix Control with Hard Output Constraints, *AIChE J.*, 37, 199, 1550-1560.
- [17] J. Yen, and Langari, R. *Fuzzy Control*, Prentice Hall, New Jersey, 1999.
- [18] R. B. Newell, and P. L. Lee, *Applied Process Control - A case Study*, Prentice-Hall, Sydney, 1989