1. Journal of King Saud University*,* ***14****,* Eng. Sci. *(2), 183-198, 2002*.

Online Tuning Strategy for Multi-loop SISO PI Control Algorithms in Multivariable Interactive Systems

Emad M. Ali

Chemical Engineering Department
King Saud University
P.O.Box 800, Riyadh 11421,
Saudi Arabia
Fax no.: ++(9661) 467-8770
Email address: amkamal@ksu.edu.sa

Abstract

Tuning of PI control algorithms for coupled multi input multi output (MIMO) systems is a challenging problem. This paper extends a previously developed model-based adaptive tuning method to handle the tuning problem of coupled multivariable systems. The performance of the proposed method is compared to those of existing methods such as Biggest Log Modulus (BLT), Sequential Loop Closing (SLC) and Multi-loop Modified Ziegler-Nichols (MMZN). The proposed method is not limited to any special form of process models and allows the use of a digital control law formulation. The tuning procedure is cast as a constrained least squares problem which can be easily solved enhancing its on-line implementation. Linear constraints are imposed on the least square problem to account for loop interactions. Application of the tuning algorithm to two simulated MIMO examples revealed its ability to perform as well as the other tested methods.

1. Journal of King Saud University*, **14***, Eng. Sci. *(2), 183-198, 2002*.

Introduction

Most control problems encountered in the chemical industries are of multivariable nature and known as Multi-input Multi-output (MIMO) systems. High interaction may also exist between the control loops making their controller tuning a challenging task. Most of the tuning efforts proposed for the multi-loop PID controllers were based on de-coupled single input single output (SISO) systems, i.e. each loop is tuned independently of the others. The multi-loop controller is then implemented in a decentralized structure in which a single input affects only one output. This de-coupled control structure has its merit of being simple, easy to maintain and having fewer tuning parameters than the cross-coupled controllers. However, when strong interaction is inevitable, the performance of a decentralized controller may degrade substantially.

For this reason, some tuning design methods that provide suppression of interaction have been reported in the literature. In these cases, the controller is still implemented in a decentralized form, but its tuning parameters are adjusted to account for interaction brought in by the other loops. Among these methods is the Biggest Log Modulus (BLT) proposed by Luyben (1986). The attractive feature of this procedure is simplicity since it includes designing only one parameter used as a de-tuning factor for all control loops. However, the resulted controller performance is conservative since the de-tuning factor is determined such that it provides tradeoff between stability and performance. Another design method is the Sequential Loop Closing (SLC) (Maciehowski, 1986). In this case, the loops are tuned individually but closed one after another so that interaction caused by closing a previous loop is accounted for during tuning the current loop. One drawback of such method is that interaction is taken care of in one direction only. This means that interaction brought by closing a current loop into all previous loops is not accounted for. Recently Wang *et al.* (1998) proposed a novel procedure to design multivariable PID systems based on the Modified Ziegler-Nichols method (MMZN). The method handles the loop interactions in more rigorous way than the others do. However, the design procedure is iterative and its extension to high order systems is very computationally involved.

Among the recent work is the neural PID controller proposed by Yeo and Kwon (1999). Although this method tunes the PID settings online, it is computationally involved. Bao *et al*. (1999) proposed a tuning method for multi-loop PID controller. The PID settings are determined by solving a nonlinear optimization problem. Thus it is considered as computationally involved algorithm. Another recent work is proposed by Chen *et al*. (1999). Their tuning method is based on frequency domain analysis, which requires special presentation of the process model. *Chien et al.* (1999) suggested a multi-loop tuning method for PID controllers. The PI settings for each loop are determined by the regular direct synthesis method. The diagonal elements of the relative gain array are used then to de-tune the controller. Another multi-loop PI controller is proposed by Jung *et al*. (1999). The method is based on minimization of a cost function, which is another computationally demanding method.

Despite the differences among these multi-loop tuning methods, they require linear models for the process. Thus, it may not be suitable for non-linear processes. Moreover, the design calculations are based on the frequency domain analysis and require graphical interpretations. This requirement restricts the on-line implementation of such techniques. In addition, the control law design is carried out on continuous time framework, which is not the common industrial practice. This work extends a previously developed model-based adaptive tuning procedure (Ali, 1999a) to handle multivariable interacting systems. The procedure solves online a constrained least squares problem to obtain new values for the PI tuning parameters such that the resulted closed-loop response fits desired performance specifications. The least squares problem is of small dimension and can be easily solved and implemented online. The proposed tuning method is used to tune a SISO control loop of an unstable ethylene polymerization reactor (Ali, *et al*., 1999). The method was also tested to tune SISO control loops of processes with changing gains (Ali, 1999b). This paper emphasizes the application of this method to MIMO interactive systems. The capability to handle loop interactions is approached by two means. First, the optimization problem is solved for the space of tuning parameters of all loops to steer the most active output inside its desired specifications. Secondly, linear constraints are induced into the optimization to ensure satisfaction of all other loop specifications.

The paper is organized as follows. In the following section, the proposed tuning algorithm is presented. Modification of the original algorithm to handle MIMO systems is also discussed in the same section. Next section demonstrates the simulated application of the proposed algorithm to a linear 2x2 example and to a non-linear 2x2 example. Performance comparison with other methods such as BLT, SLC and MMZN is also given in that section. Conclusions are drawn in the last section.

Online adaptive tuning algorithm

In this section, a summary of the proposed adaptive tuning (ATN) procedure, which was developed elsewhere (Ali, 1999a), is presented. Modification of the algorithm to handle MIMO systems is addressed in the end of the section. The algorithm is based on online adjustment of the PI settings so that the resulted closed-loop response satisfies a predefined performance specification. This is achieved through direct utilization of the closed-response sensitivity to the controller parameters. We consider the process model is given by the following differential equations:

$$\dot{z} = f(z,u,t) \qquad (1)$$

$$y = Cz \qquad (2)$$

And we assume the following decentralized PI control law:

$$\dot{z}_e = y_r - y \equiv e \qquad (3)$$

$$u = K_c e + K_I z_e \qquad (4)$$

Where K_c and K_I are diagonal matrices with their diagonal elements being the controller gain (k_{ci}) and the reciprocal integral time (τ_{Ii}) for each control loop respectively. Solution of equations (1-2) along with (3-4) gives the closed-loop simulation of the process under a standard PI control structure. Since the controller is implemented in decentralized framework, the sensitivity of the closed-loop response to the PI settings can be then computed from the numerical solution of the following adjoint differential equations:

$$\frac{d}{dt}\left(\frac{\partial z_j}{\partial k_{ci}}\right) = \frac{\partial f_j}{\partial z_j}\frac{\partial z_j}{\partial k_{ci}} + \frac{\partial f_j}{\partial u_j}\frac{\partial u_j}{\partial k_{ci}} \tag{5}$$

$$\frac{d}{dt}\left(\frac{\partial z_j}{\partial \tau_{Ii}}\right) = \frac{\partial f_j}{\partial z_j}\frac{\partial z_j}{\partial \tau_{Ii}} + \frac{\partial f_j}{\partial u_j}\frac{\partial u_j}{\partial \tau_{Ii}} \tag{6}$$

$$i = 1,\cdots,n_y;\quad j = 1,\cdots,n_x$$

where n_y represents the number of controlled variables and n_x is the number of state variables. The sensitivity of the measured output to the PI settings can then be extracted from Eqns. (5) and (6) according to equation (2). The gradients of the manipulated variables with respect to the PI settings are given as follows:

$$\frac{\partial u_i}{\partial k_{ci}} = e_i + \frac{1}{\tau_{Ii}} z_{ei} \tag{7}$$

$$\frac{\partial u_i}{\partial \tau_{Ii}} = -\frac{k_{ci}}{\tau_{Ii}^2} z_{ei} \tag{8}$$

The tuning algorithm is carried out as follows. First the user should provide a value for the prediction horizon, P_w, a value for the threshold within which the output variation is considered acceptable, and the desired performance specification as upper and lower bounds on the output response expressed in the form of vectors: y^u and y^l. Typical forms of the specification bounds are shown in the figures in the simulation section. The nominal bound vectors should have specific finite size. The tuner algorithm consists of two phases namely, observation phase and triggered phase. During the observation phase, the tuner function is monitoring the closed-loop predictions. If one of the predicted values violates its threshold, the algorithm will trigger the adaptation and scale the nominal bounds. During the triggered phase, the following steps are conducted:

Step 1: at time, k, solve equations (1) to (4) numerically up to $k+P_w$ to obtain the closed-loop prediction of the output responses and their sensitivity to k_c and τ_I.

Step 2: Determine the maximum bound violation, Δy, over all outputs and the horizon P_w.

Step 3: Solve the following least squared problem:

$$\min_{\Delta\phi_k} \left\| \Delta y - \nabla_{\phi} y^T \ \Delta\phi_k \right\|_2 \tag{9}$$

Subject to:

$$\Delta\phi_k^l \le \Delta\phi_k \le \Delta\phi_k^u$$

where $\Delta\phi$ is the change in the PI parameter space and $\nabla_{\phi} y$ is the sensitivity of the most active output to the parameter space at the maximum violation point.

Step 4: update the current parameter space:

$$\phi_k = \phi_{k-1} + \Delta\phi_k$$

Step 5: compute the new controller output and implement it. Set $k=k+1$ and go back to step 1.

When the sampling time exceeds the window size of the bounds, the tuner is disabled and switched back to the observation phase. The least squares problem in *step 3* is cast as Quadratic programming and solved by MATLAB software. It should be noted that the performance specifications can be determined by prudent plant experience and P_w can be used as an adjustable parameter. It should be emphasized here that the above algorithm is repeated every sampling time and that the control law and its tuning parameters are assumed to be step-wise continuous functions. More detailed information related to the proposed tuner such as derivation of the algorithm and automatic scaling of the nominal bounds can be found elsewhere (Ali, 1999a). Elaboration on the effect and selection of P_w is given in previous work (Ali, 1999a and 1999b).

According to the algorithm described above, loop interactions is been taken care of by simply tuning the PI parameters for all loops simultaneously to force the response of the most active output lies inside its desired specification (*step 3*). Moreover, additional linear constraints can be imposed on the least squares problem to ensure that the new parameters also satisfy the desired specification of the other outputs. The cost function (Eq. 9) is actually derived from the linear approximation of the relationship between process output and the PI parameters (Ali, 1999a):

$$\hat{y}_i(k+1) = y_i(k+1) + \nabla_{\phi_k} y_i^T(k+1)\Delta\phi_k \qquad (10)$$

for the most active output, say *j*, at the time instant at which the maximum violation of the performance occurs, say *k+m,* $1 \le m \le P_w$. Here, $\hat{y}$ denotes the expected output at the new value of ϕ. Based on equation (10), the new value of the parameter space should also satisfy the following constraints:

For the j_{th} output:

$$y_j^l(k+n) - y_j(k+n) \le \nabla_{\phi_k} y_j^T(k+n)\Delta\phi_k \le y_j^u(k+n) - y_j(k+n)$$
$$n = 1,\cdots,P_z,\ \ n \ne m$$

And for the other outputs:

$$y_i^l(k+\ell) - y_i(k+\ell) \le \nabla_{\phi_k} y_i^T(k+\ell)\Delta\phi_k \le y_i^u(k+\ell) - y_i(k+\ell)$$
$$\ell = 1,\cdots,P_z,\ \ i \ne j$$

The above two inequalities can be written in a matrix form:

$$A\Delta\phi_k \le b$$

P_z is the window size for the linear constraints. In this paper it is taken equal to P_w.

Setting P_z = 0 disables the constraints. Therefore, if performance improvement is required for highly interactive systems, then the above linear constraints can be imposed on the least squares problem in step 3 of the algorithm. The tuning method, as it stands, does not guarantee robust stability. However, nominal stability can be checked whenever new values for the PI settings are obtained. If the nominal stability condition is satisfied, the new settings are assumed safe to use, otherwise the old values are implemented.

Simulation Examples

In the following, the proposed tuning algorithm is applied to two simulated examples. The resulted performance is also compared to those obtained by other exiting methods.

Example 1:

Consider the well-known Wood/Berry Binary distillation column (Wood and Berry, 1973) which was also used by Luyben (1986) and Wang *et al*. (1998) to test their tuning algorithms:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

In this paper, the above transfer function is realized into state space formulation using MATLAB software. In this case, the dead time is approximated by first-order *Pade* approximation. This transformation introduces inverse response instead of time delay. For fair comparison, the BLT, SLC and MMZN are tested on the transformed model. The PI parameters for BLT, MMZN and regular Z-N (RZN) are taken from ref. 1 and ref. 3 and are also listed in Table 1.

The RZN parameters, given in Table 1, are the PI settings obtained by Ziegler-Nichols method (Ziegler and Nichols, 1943) for each individual loop without considering interaction. Table 1 also includes the SLC settings. The latter is obtained by carrying out the regular ZN method to the second loop with the first loop being closed using its corresponding PI parameters. The performance of the BLT, SLC and MMZN for a unit set point change in y_1 and a zero set point change in y_2 is shown in figure 1(a,b). A sampling time of 0.5 minutes is used in the simulation hereafter. Obviously, the BLT provided smoother response due to its conservative PI settings. However, this conservative settings lead to slow recovery of y_2 response to its set point. The feedback response using the RZN settings is shown in figure 1 (c,d) by the dotted line. The performance is quite oscillatory indicating how interaction can seriously affect the closed-loop performance of a coupled multivariable system.

The effectiveness of the proposed tuning method (ATN) for this specific control problem is shown in figure 1(c,d) by the solid line. The RZN settings are used as the initial guess. The adapted PI settings are shown in figure 2. The performance envelope shown in figure 1(c,d) is designed such that it limits the overshoot to 10% for the set point change case and 20% for the disturbance case and gradually brings the response to within 1% of its steady state value. The window size of the envelope is 56 samples (equivalent to time interval of 28 minutes) and its threshold value is ±1% of the output steady state value. P_w = 10 is used in the simulations. According to figure 1(c,d), the ATN method managed successfully to improve the feedback performance starting from somewhat aggressive PI settings, i.e., RZN. When compared to BLT performance, the ATN delivered less overshoot and faster response. Although the obtained performance does not lie completely inside the desired performance envelope, the latter was the main driving force to fire the tuning algorithm and to consequently improve the performance. Figure 2 illustrates that both integral times should be increased and k_{c2} be decreased to eliminate aggressiveness. k_{c1}, however, was decreased initially and then returned back to its initial value. Smaller simulation time interval is intentionally used for Figure 2(a) to make the initial variations in the controller gains clear and because the controller gains remain stationary beyond the time = 20 minute of the simulation.

The above tuning algorithms were also examined for disturbance rejection. A step change of magnitude of 0.1 is introduced to the states of the process model. This represents an internal disturbance and the purpose is to test their robustness. The feedback response is demonstrated in Figure 3. As expected, the BLT performance is sluggish compared to the others due to its conservative tuning parameter values. On the other hand, the RZN had the worst performance. Although the ATN method started with the unreasonable RZN settings, it was able to dramatically improve the closed-loop response of both outputs as shown in figure 3(c,d). The obtained performance is somewhat comparable to that of the MMZN if not better in the sense of less down-shoot in y_2 response. The shape and size of the performance envelopes are the same as in the previous case. The tuning algorithm was fired twice. Once at the beginning because the predicted values of the second output violated its specifications and then at time = 25 minute because of the specification violation by the first output. In both cases, the nominal envelopes were scaled and oriented automatically to suit the actual behavior of the process. The corresponding adaptation profile for the PI settings is depicted in figure 4. A smaller time interval is purposely used in part (a) of figure 2 to clarify the initial variation in the controller gains.

Our investigation revealed that inclusion of the linear constraints in *step 3* of the tuning algorithm was not necessary in both cases, i.e. set point change and disturbance rejection. In fact, it was found to be of little help. The magnitude of performance violation is large and the addition of linear constraints made the constrained least squares problem more stringent. A situation lead MATLAB to terminate the optimization with infeasible solution, which is almost similar to the unconstrained solution. It should also be pointed out that in all simulations, the controller gains were bounded between [0.1, –0.3] and [1, -0.01] and the reset times between [1, 2] and [10, 30]. It should be noted that in practice positive controller gains are used. In that case, if the process has a negative feedback behavior, which is the case in this paper, the definition of the error signal in Eq. (3) is reversed. However, since we are dealing only with simulation, negative gains are used instead of changing the definition of the error signal.

Example 2:

Consider a blending tank (Smith and Corripio, 1997) in which two streams of volumetric flow rates w_1 and w_2 and concentration x_1 and x_2 respectively are mixed adiabatically. The outlet stream has a volumetric flow rate w and concentration x.

Assuming constant fluid density and isothermal operation, the process model is given as follows:

$$\frac{dV}{dt} = w_1 + w_2 - w$$
$$V\frac{dx}{dt} = w_1(x_1 - x) + w_2(x_2 - x)$$
$$w = c_v\sqrt{V}$$

Where V is the liquid holdup and C_v is the valve characteristic constant, which equals to 1.414 $m^{1.5}/min$ in this paper. The process outputs are w and x. At nominal operation conditions, $w_1 = w_2 = 1$ m^3/min, $x_1 = 0.1$ and $x_2 = 0.8$. Consequently, the steady state values of the outputs are $w = 2$ m^3/min *and* $x = 0.45$. It should be noted here that only the dynamic model of V and x are taken from Smith and Corripio (1997). The outlet flow description and design parameter values are chosen arbitrarily in this paper. The control objective is to regulate w and x by manipulating both w_1 and w_2. To measure the severity of loop interactions, the relative gain array (Stephanopuolos, 1984) of the process model is computed. The diagonal element of the relative gain array is found to be 0.5, which indicates strong interaction. In this paper, the outlet flow rate, w is controlled by w_1 and the outlet concentration, x by w_2.

Iterative closed-loop simulations were used to determine the RZN, BLT, and SLC settings. Their numerical values are given in Table 2. For the BLT method a de-tuning factor of magnitude of 5 is used. In fact, the maximum closed-loop log modulus for the process before de-tuning is much less than 4, a reference value suggested by Luyben (1986) as a decision criteria for his tuning method. Nevertheless, the value used for the de-tuning factor in this paper is chosen such that it further reduces the log modulus value. A sampling time of 0.1 minute is used in all the following simulations.

Figure 5 illustrates the feedback response to set point change of [0, 0.05] using the RZN settings. The RZN settings of each individual loop provide excellent closed-loop performance when the other loop is open (not shown). However, when the two loops are closed, they interact and the overall closed-loop performance deteriorates as indicated by the persistent oscillation (Fig. 5). A better response can be obtained using the SLC settings as shown by Figure 6(a,b). Moreover, much smoother response, but at the expense of higher overshoot, can be achieved when using the BLT settings (Fig. 6(a,b)).

The result of testing the ATN method to the same set point change is shown in Figure 6(c,d). The nominal performance envelopes are designed such that they limit the overshoot to 5% for servo problem and 7% for regulatory problem and eventually bring the response to within 1% of its final steady state value. The window size of the envelope is 56 samples, i.e. 5.6 minutes, and the threshold value is ±2% of the output steady state value. P_w = 42 is used. The tuner is fired twice. First at the beginning due to the set point change and secondly at time = 5.6 minute because the first output, i.e., *w*, predicted response violated its specifications. Although the response of *w* lies inside the envelope for t > 5 minute, its future predicted values by the model (not shown) violate the performance specifications. In this simulation the RZN settings are used as the initial guess. Despite the output initial oscillation as shown in Figure 6(c,d), the tuner succeeded in steering the PI settings (Fig. 7) in such a way to smooth out the feedback response.

To investigate the robustness of the tuning methods, their feedback performance was tested for the rejection of disturbance on x_2 in the presence of modeling error. Specifically, -10% step change in x_2 is introduced into the plant and the value of C_v in the model is assumed to be 10% larger than that in the plant. The BLT and SLC will be only affected by the disturbance since they operate on the error signal coming form the plant. On the other hand, the ATN will be affected by both the disturbance and modeling error since it operates on the error signal coming from the plant and on the future predictions produced by the model. The simulation result is depicted in Figure 8. The shape and size of performance envelopes used are the same as those described in the previous paragraph for regulatory problem. P_w = 10 is employed in this case. The ATN algorithm is triggered twice at which the envelopes were sized and oriented automatically. Obviously, the ATN method delivered slightly inferior performance compared to those obtained by the BLT and SLC. This is attributed to the effect of modeling error associated with the model prediction of the output and its sensitivity to the tuning parameters. In fact, as shown by Figure 9, the modeling error influenced the adaptation of the PI settings for the first loop. This is because the parametric error affects directly the first output, i.e. *w*, dynamics. Nevertheless, the ATN was still able to provide somewhat good performance knowing the fact that it started from bad initial guess such as the RZN settings.

On the other hand, SLC outperformed the BLT in the regulatory control problem, but this is not the case in the servo control problem. This phenomenon is also observed in example 1. This can be attributed to the more aggressive tuning parameters (larger controller gains) used by SLC, which is desirable for disturbance rejection. The trade-off between good tuning for load-disturbance and for set point change responses is not surprising. It is well known that good tuning for load-disturbance response may often give too oscillatory response for set point changes (Astrom, *et al.*, 1993, Sung and Lee, 1996). For this reason most of the available PID tuning formulas provide two set of settings, one for load-disturbance and another for set point change. This situation highlights another advantage of the proposed tuning algorithm, which is being independent of the source of process dynamics.

In the entire ATN simulations, the controller gains were constrained between [5, 15] and [80, 200] and the reset times between [0.083, 0.0833] and [1, 1]. In the above two cases, the magnitude of performance violation was small. This in turn made the adaptation algorithm looses strength. For this reason, the inclusion of the linear constraints was essential to enhance the tuner performance. The reason for the small performance violation is partly due to over relaxed design of the performance bounds.

Conclusions

A novel approach to handle the tuning problem of multivariable interacting processes is presented and tested by two simulated examples. The method requires a process model from which future predictions of the output closed-loop response and its sensitivity can be estimated. The obtained information is used directly to obtain new values for the PI settings that minimizes the predicted output violation of its performance specification. The tuning algorithm thus solves a simple constrained least squares problem, which makes it appealing for online implementation. The simulation results revealed the capability of the proposed method to perform as well as other existing methods and may become slightly inferior in case of model uncertainty. It was also found that, in some cases, the inclusion of linear constraints on the least squares problem improves the tuner performance.

The main advantage of the proposed technique is its online adaptation feature. This makes it more suitable for process with time varying dynamics (Ali, 1999b). However, the method provides only nominal stability guarantees, but not robust stability. Robust stability conditions can be formulated as nonlinear constraints and introduced to the least squares problem, however, the algorithm will loose its simplicity as online tuner.

1. Journal of King Saud University, ***14***, Eng. Sci. *(2), 183-198, 2002*.

Nomenclature

A	Constant matrix for linear constraints
b	Constant vector for linear constraints
C	Constant matrix
C_v	Valve characteristic constant, ($m^{1.5}$/min)
e	Error signal
k	Sampling time
k_c	Controller gain
K_c	Diagonal matrix of controller gains
K_I	Diagonal Matrix of reset time
n_y	Number of measured outputs
n_x	Number of states
P_w	Prediction horizon
P_z	Window size for linear constraints
u	Manipulated variable
V	Tank volume, (m^3)
w_1, w_2	Volumetric flow rate of stream 1 and 2 respectively (m^3/min)
x_1, x_2	Concentration of stream 1 and 2 respectively
y	Measured output
y_r	Set point
y^u, y^l	Vector of desired specifications
z	Process states
z_e	Augmented states
ϕ	Space of PI tuning parameters
ϕ^u, ϕ^l	Upper and lower values of ϕ
τ_I	Controller reset time
Δy	Amount of bound violation

1. Journal of King Saud University, ***14***, Eng. Sci. *(2), 183-198, 2002.*

References

[1] Ali, E., On-line Tuning Strategy for PI Control Algorithms, *J. of King Saud University,* **11**, *Eng. Sci.,* (1), 49-70, (1999a).

[2] Luyben, W. L., Simple Method for Tuning SISO Controllers in Multivariable Systems. *Ind. Eng. Chem. Proc. Des. Dev*., **25**, 654-660, (1986).

[3] Maciehowski, J.M., *Multivariable feedback Design*, Addison Wesley, Workingham, UK, (1986).

[4] Wang, Q., T. Lee, and Y., Zhang, Multiloop Version of the Modified Ziegler-Nichols Method for Two Input Two Output Processes, *Ind. Eng. Chem. Res*., **37**, 4725-4733, (1998).

[5] Yeo Y. and T. Kwon, "A Neural PID Controller for the pH Neutralization Process", *Ind. Eng. Chem. Res*., **38**, 978-987 (1999).

[6] Bao, J., J.F. Forbes and P.J. McLellan, "Robust Multiloop PID Controller Design: A Successive Semidefinite Programming Approach", *Ind. Eng. Che,. Res*., **38**, 3407-3419 (1999).

[7] Chen, C., H. Huang and C. Hsieh, "Tuning of PI/PID Controllers Based on Specification on Maximum Closed-loop Amplitude Ratio". *J. Chem. Eng. Japan*, **32**, 783-788 (1999).

[8] Chien, I, H. Huang and J. Yang, "A Simple Multiloop Tuning Method for PID Controllers with No Proportional Kick", *Ind. Eng. Chem. Res*., 38, 1465-1468 (1999).

[9] Jung, J., J. Choi and Jietae Lee, "One-Parameter Method for a Multiloop Control system Design", Ind. Eng. Chem. Res., 38, 1580-1588 (1999).

[10] Ali, E. M., A. E. Abasaeed and S. M. Al-Zahrani; "Improved Regulatory Control of Industrial Gas Phase Ethylene Polymerization Reactors", *Ind. Eng. Chem. Res.* **38**(6), 2383-2390 (1999).

[11] Ali, Emad; ``Control of Non-linear Chemical Processes Using Adaptive PI Algorithms``, *Ind. Eng. Chem. Res*., **39**, 1980-1992 (2000).

[12] Wood, R. K., and M. W., Berry, Terminal Composition Control of a Binary Distillation Column, *Chem. Eng. Sci*., **28**, 1707-1717, (1973).

[13] Ziegler, J. G. and N. B., Nichols, Optimum Settings for Automatic Controllers, *Trans. ASME*, **65**, 433-444, (1943).

[14] Smith, C. A., and A. B., Corripio, *Principles and Practice of Automatic Process Control*, John Wiley & sons, New York, (1997).

1. Journal of King Saud University, ***14***, Eng. Sci. *(2), 183-198, 2002*.

[15] Stephanopuolos, G., *Chemical Process Control*, Prentic-hall, NJ, (1984).

[16] Astrom, K. J., T. Hagglund, C. C., Hang and W. K., Ho, Automatic Tuning and Adaptation for PID Controllers- A Survey, *Control Eng. Practice*, **1**, 699-714, (1993).

[17] Sung, S. W., and I. B. Lee, Limitations and Countermeasures of PID Controllers, *Ind. Eng. Chem. Res.*, **35**, 2596-2610, (1996).

Table 1: PI parameters for example 1

Parameter	BLT	SLC	MMZN	RZN
k_c	[0.375,-0.075]	[0.96,-0.09]	[0.94,-0.132]	[0.96,-0.19]
τ_I	[8.29,23.6]	[3.25,6.23]	[8.09,9.03]	[3.25,9.2]

Table 2: PI settings for example 2

Parameter	BLT	SLC	RZN
k_c	[7, 22]	[36, 45]	[36, 113]
τ_I	[0.83, 0.83]	[0.16, 0.33]	[0.16, 0.16]

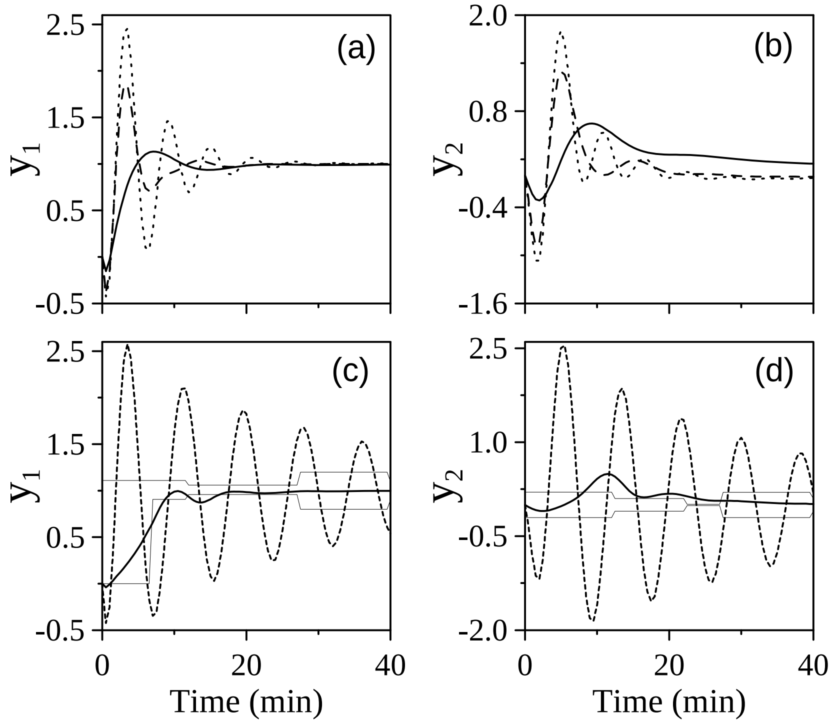

Figure 1: Time response to set point change of [1, 0]. (a,b) BLT: solid line; SLC: dotted line; MMZN: Dashed line, (c,d) RZN: dotted line; ATN: solid line; specification envelop: light solid line.

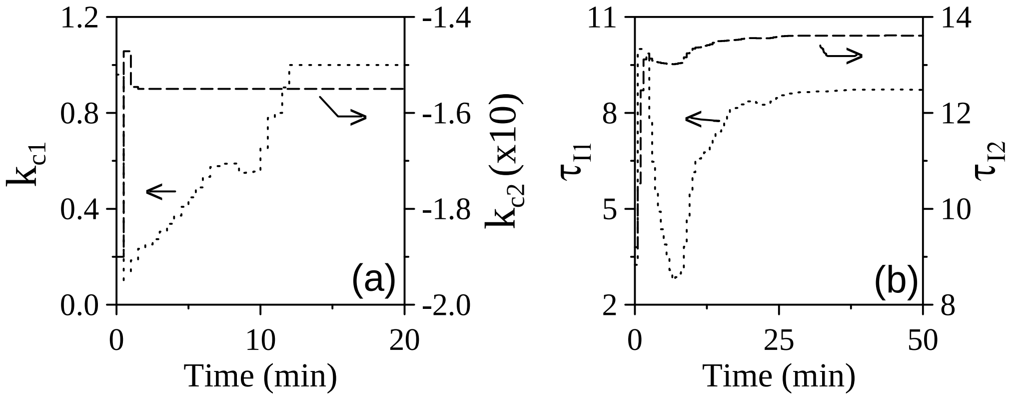

Figure 2: Adapted PI settings of example 1 for set point change of [1, 0], Dotted line: loop 1; Dashed line: loop 2.

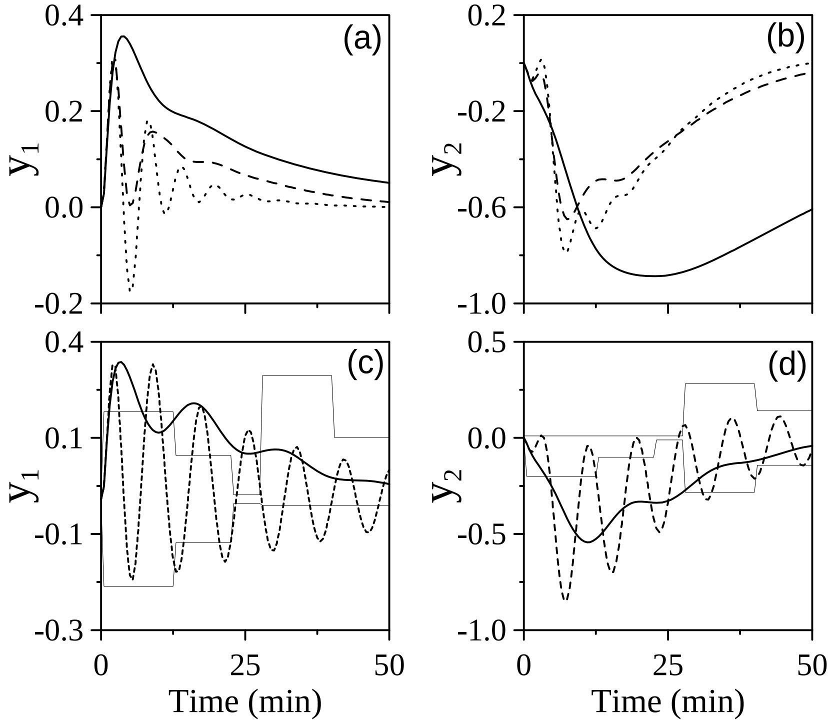

Figure 3: Time response for disturbance rejection. (a,b) BLT: solid line; SLC: dotted line; MMZN: Dashed line, (c,d) RZN: dotted line; ATN: solid line; specification envelop: light solid line.

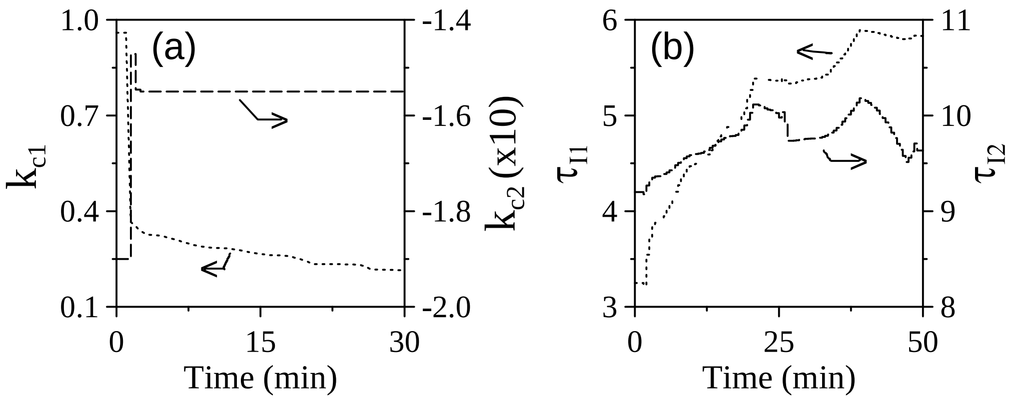

Figure 4: Adapted PI settings of example 1 for disturbance rejection, dotted line: loop 1; dashed line: loop 2.

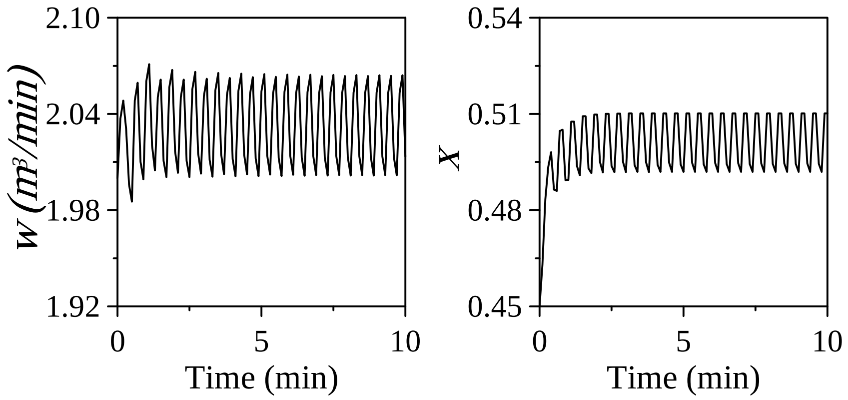

Figure 5: Time response to set point change of [0, 0.05] for example 2 using the RZN PI settings.

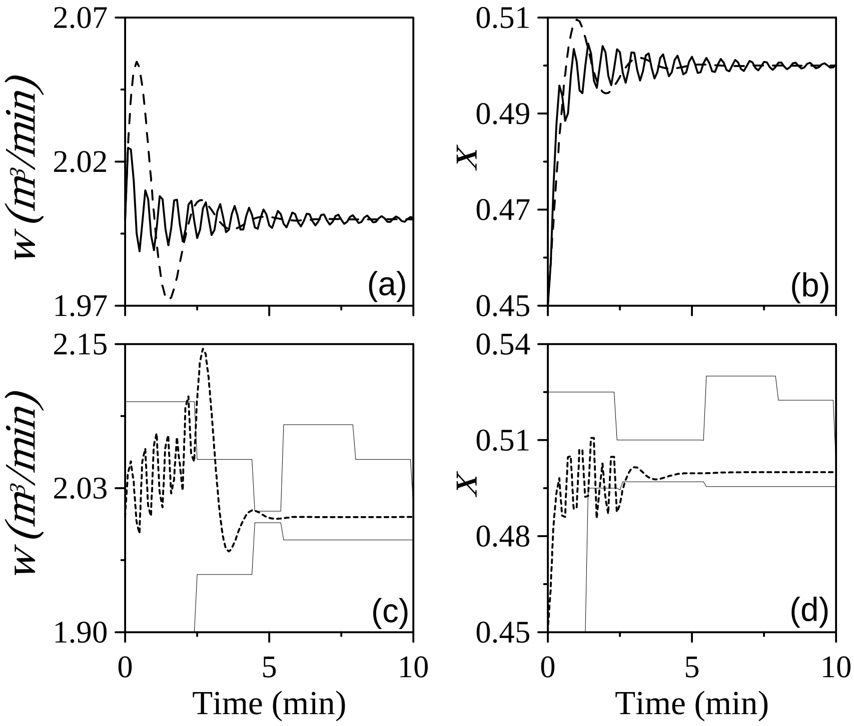

Figure 6: Time response to set point change of [0, 0.05] for example 2, (a,b) SLC: solid line; BLT: dashed line, (c,d) ATN: dashed line; performance envelope: solid line.

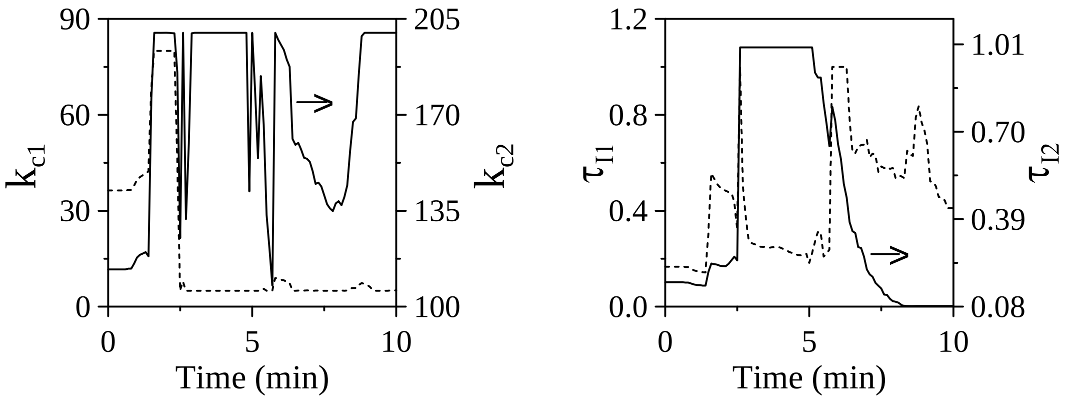

Figure 7: Adapted PI settings profile of example 2 for the servo problem, dashed line: k_{c1} & τ_{I1}; solid line: k_{c2} & τ_{I2}.

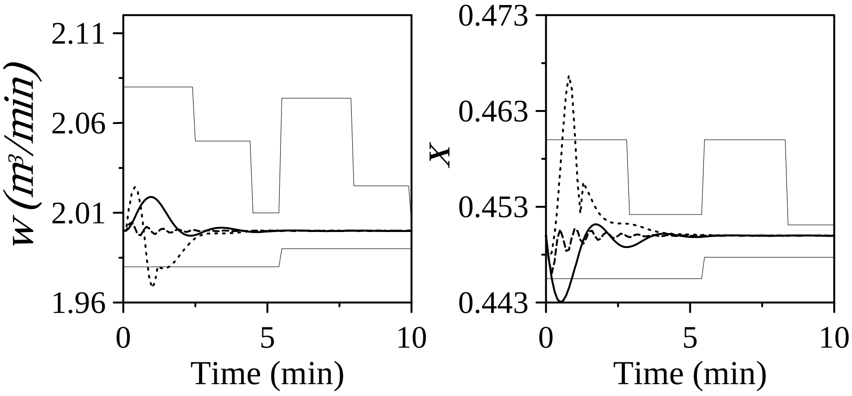

Figure 8: Time response of example 2 for disturbance rejection in the presence of modeling error, dotted line: ATN; dashed line: SLC, solid line: BLT.

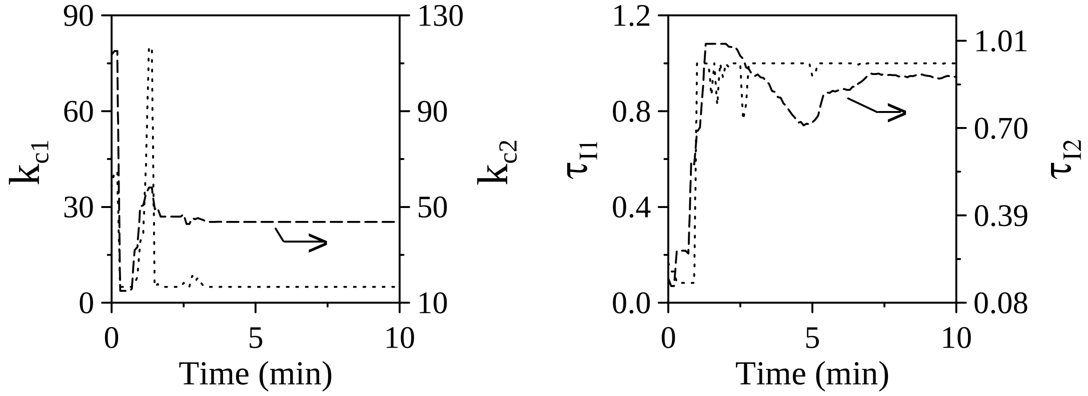

Figure 9: Adapted PI settings profile of example 2 for disturbance rejection in the presence of modeling error.