J. King Saud Univ., Vol. 11, Eng. Sci. (1), pp. 49-70 (A.H. 1419/1999)

On-line Tuning Strategy for PI Control Algorithms

Emad Ali
Department of Chemical Engineering, College of Engineering, King Saud University, P.O. Box 800, Riyadh 11421, Kingdom of Saudi Arabia

(Received 20 May 1997; accepted for publication 4 March 1998)

Abstract. On-line model-based tuning of the parameters of conventional PI control algorithms is considered. This is accomplished by requiring satisfaction of the performance specifications. The specifications are defined as time-domain response envelopes, which are appealing to the practitioner. The parameters adaptation at each sampling point is accomplished by using a linear relationship between the process output and the PI tuning parameters. Thus the adaptation strategy directly uses the sensitivity of the closed-loop response to the parameters. Nominal stability condition is also developed for the algorithm and used only as an indicator of the stability of the closed-loop response. The efficiency of the method is presented through implementation on a SISO time-delayed integrator process and on a MIMO non-linear CSTR process. Robustness of the method is also examined by considering modeling error in the MIMO example.

Introduction

Conventional PID controllers have been widely used in the chemical industry for its simplicity and successful practical applications. Many efforts dealing with tuning of such controllers for better performance have been reported[1-9]. These tuning methods can be classified under three major categories; reaction curve methods[2-3;7;9], multi-objective optimization methods[1;4], and frequency-domain methods[5;6]. Despite the variation of these methods, they provide excellent initial guess of the PID settings. However these settings are then kept constant over the entire simulation time or the actual controlled operation. Moreover, most of these methods do not consider the time-domain performance specifications directly in their algorithms. Recently, some efforts dealing with PI controller tuning were reported which are based on step testing methods which belongs to the first category, *i.e.*, reaction curve methods [10;11;12]. These proposed approaches require representing the process by linear model in the standard transfer function form which limit their application. Similarly Parasiliti [13] presented a tuning scheme for processes represented by SISO linear transfer function models. This method tunes only one variable by binary search technique which cannot be easily extended to multi-variable systems. Another recent tuning algorithm which is based on frequency domain analysis is reported by Ho *et. al.* [14]. The algorithm is also limited to SISO linear models.

In this present work a model-based on-line tuning of the PI algorithms is presented. The method is so simple and intuitive and does not require exhausting trial-and-error procedure nor time-consuming computations. The method is also easily implemented and understandable by the control engineer. The added features of this proposed method are:

1) The ability to deal with the changing operating conditions and behavior of the process due to its adaptive on-line implementation where other methods use one 'rigid' set of PI tuning parameters.
2) Enhanced multi-loop tuning as the PI settings of all loops are adjusted simultaneously where other methods tune each loop independently.
3) Full automation where no deep user understanding of controller tuning is necessary. The user needs only to provide the desired specs and the algorithm will be triggered and operates by itself.
4) The method is applicable to both linear and non-linear models.
5) The method is based on satisfaction of time-domain specifications which makes it more suitable for practical application.

Ha [15] proposed a fuzzy logic scheme for PI controller tuning that possess adaptive nature (first feature) only. The method is based on SISO systems and requires that the minimum and maximum values of the controller setting to be known in advance. Another adaptive tuning scheme was proposed by Jones *et. al* [16]. The method however, is based on frequency domain analysis and requires the Ziegler Nichols [9] settings to be used as initial guess. Yusof *et. al.* [17] addressed tuning scheme that deals with the first and second features. Their method, however, is based on linear models and requires initializing some polynomial matrices.

The proposed tuning method in this paper requires a mathematical (linear or non-linear) model for the process from which a prediction of the closed-loop response and its sensitivity to the PI settings can be inferred. The sensitivities obtained can be utilized along with the linear approximation of the relationship between the closed-loop response and the PI settings to determine new values for the latter. These values are determined such that the resulting closed-loop response satisfies a preset time-domain performance specifications. A similar tuning approach was proposed by Zhu and Lee[8]. However, their method was based on open-loop prediction of the output and its gradients. Moreover, the algorithm requires adjusting many parameters which leads to the loss of its automated feature.

Controller On-line Tuning

Sensitivity equations

The adaptation of the PI tuning parameters is achieved through the exploitation of the sensitivity of the closed-loop response to the tuning parameters.

Hence development of analytical expressions for the sensitivity of the closed-loop response with respect to the PI tuning parameters will be presented. The closed-loop dynamic is described by the following state-space model:

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

$$y = cz \tag{2}$$

$$\dot{z}_{ei} = y_i - r_i \tag{3}$$

$$u_i = kc_i(y_i - r_i) + \frac{kc_i}{\tau_{Ii}} z_{ei} \tag{4}$$

$$i = 1, \ldots, n_y$$

where z is the state vector, u is the manipulated variable vector, y is the process output vector, r is the set point vector, and n_y is the number of controlled outputs. It should be noted that although discrete PI controller will be used in the process, a continuous formulation is used in the model to facilitate the mathematical treatment for driving the sensitivity expressions. The gradients of the closed-loop output with respect to the tuning parameters, *i.e.*, kc and τ_I can be developed directly by taking the derivative of the above equations as follows:

$$\frac{\partial y}{\partial kc_i} = c\frac{\partial z}{\partial kc_i} \tag{5}$$

$$\frac{d}{dt}\frac{\partial z}{\partial kc_i} = \frac{\partial f}{\partial z}\frac{\partial z}{\partial kc_i} + \frac{\partial f}{\partial u}\frac{\partial u}{\partial kc_i} \tag{6}$$

$$\frac{\partial u_i}{\partial kc_i} = y_i - r_i + \frac{1}{\tau_{Ii}} z_{ei} \tag{7}$$

$$i = 1, \ldots, n_y$$

Similar expressions can be, of course, obtained for τ_I. However, the only difference would be in the last equation which will be given as follows:

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

It should be noted that in the standard PI control framework (multi-loop SISO control system), each controller loop connects one output to one manipulated variable. Thus, only the derivative of each output to its PI settings is required as the derivative of the output of one loop to the PI settings of another loop is zero. Closed-loop prediction of y and its gradients over a specific prediction horizon can then be obtained by numerical integration of the above augmented state equations. The initial conditions of the actual states and the manipulated variables are their corresponding initial steady state values, while that for the augmented states, *i.e.*, z_e and the gradients, are zeros. To reduce model-plant mismatch, the predicted output is corrected by disturbance estimates in the standard Internal Model Control framework[18]. The estimated disturbance is assumed constant over the prediction horizon and set equal to the difference between the plant and model outputs at the current sampling point, k.

PI settings tuning strategy

This section presents an on-line tuning technique that adapts the PI parameters in order to steer the closed-loop response to satisfy preset time-domain desired specifications. Typical time-domain specifications for set point tracking and disturbance rejection are shown by the solid lines in the simulation figures in the example section .

The on-line adaptation strategy centers around linear approximation of the relationship between the process output and the PI tuning parameters. Hence, at any on-line sampling point k, the *ith* closed-loop predicted output (y_i) at the sampling instant, $k+1$, can be related to the space of change in the tuning parameter, Δx_k, as follows:

$$\hat{y}_i(k+1) = y_i(k+1) + \nabla_{x_k} y_i^T(k+1)\Delta x_k \tag{9}$$

where $x_k = [kc_1(k) \ \ldots \ kc_{ny}(k) \ \ldots \ \tau_{I1}(k) \ \ldots \ \tau_{Iny}(k)]^T$, $\nabla x_k \, y_i$ is the gradient of the *ith* predicted closed-loop response as developed in the previous section, and $\hat{y}_i$ denotes the expected output at the new value of x. If $y_i(k+1)$ violates the desired specification, then an intuitive adaptation strategy would be to utilize equation 9 to determine, on-line, the minimum value of Δx_k that makes $\hat{y}_i(k+1)$ satisfy the specifications.

In order to ensure that the control action will not result in violation of the specifications, Δx_k can be adjusted on-line so that the output j, which has the maximum specification violation over all outputs, at the sampling instant k+m, at which the corresponding closed-loop response has the maximum specification violation over the closed-loop prediction horizon, P_w, lies within the desired specifications. If the system has dead time, then the prediction horizon P_w is equal to P_w + nd, where nd is the maximum dead time of the system. The optimum value of Δx_k is obtained by minimizing the 2-norm of Eq. (9), *i.e.*, by solving a least squares problem in the form of:

$$\min_{\Delta x_k} \left\| \nabla_{x_k} y_i^T (k+m) \Delta x_k - \Delta y_i (k+m) \right\| \tag{10}$$

Where Δy is the difference between $\hat{y}$ and y. Lower and upper bounds on Δx_k will be imposed on the minimization problem. It is desirable, when dealing with multiple variables, to use scaled values of the parameters especially when their corresponding initial values are not of the same order of magnitude. Thus, scaled value of x will be used in the optimization problem (10) which also involves scaling the gradients of the closed-loop response. The latter is formulated mathematically in step 4.2 of the adaptation algorithm discussed in the next paragraph. In addition, for fast dynamic processes, the tuning algorithm may produce large value of x in order to steer the closed-loop response back inside the performance envelope. Thus, in that course, the change in x should be relaxed to avoid aggressiveness. Conceptually, this will be achieved by limiting the rate of change of Δy of equation 10 to be less than the rate of change of the process over one sampling time point. The concept of relaxing x will be formulated mathematically in step 4.1 in the following algorithm. The adaptation algorithm can be clearly understood by the following steps:

At any sampling point k, and before computing the control action;

***Step** 1:* Predict the closed-loop response and its gradient over the prediction horizon P_w via numerically integrating Eqs. (1) to (8) for fixed values of the tuning parameters at x_k and constant estimated disturbance at k. Δx_k is initially equal to x_{k-1}.

Step 2: Evaluate the predicted violation of the specifications:

$$M_i(k+l) = \begin{cases} y_i^l(k+l) - y_i(k+l) & \text{if } y_i^l(k+l) > y_i(k+l) \\ y_i(k+l) - y_i^u(k+l) & \text{if } y_i^u(k+l) < y_i(k+l) \\ 0 & \text{if } y_i^u(k+l) > y_i(k+l) > y_i^l(k+l) \end{cases}$$

$$i = 1, \ldots n_y, \qquad l = 1 + nd_i, \ldots, P_w + nd_i$$

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

$$M_j(k+m) = \max_{1 \le i \le n_y} \max_{1+nd_i \le l \le P_w + nd_i} M_i(k+l)$$

Step 4: If $M_j(k+m) = 0$ go to step 5, otherwise:

Step 4.1: Compute the deviation from the desired specification and relax it if necessary:

$$\Delta y = \begin{cases} y_j^l(k+m) - y_j(k+m) & \text{if } y_j^l(k+m) > y_j(k+m) \\ y_j(k+m) - y_j^u(k+m) & \text{if } y_j^u(k+m) < y_j(k+m) \end{cases}$$

$$\text{if} \left|\Delta y_j\right| > \left|yp_j(k) - yp_j(k-1)\right| \quad \text{set: } \Delta y = \left|yp_j(k) - yp_j(k-1)\right| \ \text{sign}(\Delta y_j)$$

where *yp* is the output measurement.

Step 4.2: Define $a = \nabla x_k y_j^T(k+m)$, and scale it by post-multiplying with ψ which is a matrix that has the current values of the settings on the diagonal and zeros elsewhere.

Step 4.3: If $\|a\|_\infty \le \beta$ (a user defined number, equal to 10^{-5} in this paper) go to step 4.4, otherwise solve:

$$\min_{\Delta x_k} \|a\Delta x_k - \Delta y\|_2$$

subject to:

$$\Delta x^l \leq \Delta x_k \leq \Delta x^u$$

where $\Delta x^l = x^l - x_k$, $\Delta x^u = x^u - x_k$, and x^l & x^u are the lower and upper bounds on x respectively. Go to step 4.5

Step 4.4: *Set $\Delta x_k = 0$*

Step 4.5: *Set $x_k = x_k + \Delta x_k \psi$.*

Step 5: Compute and implement the control action. Shift to the next sampling time, set $k=k+1$. Go to Step 1.

The under-determined unconstrained least square problem of step 4.3 has infinite solutions. One solution is the minimum norm solution given as:

$$x_v = a^T (aa^T)^{-1} \Delta y \tag{11}$$

If x_v satisfies the bounds, then $\Delta x = x_v$ is an acceptable solution of the optimization problem. Otherwise, the solution is parameterized as $\Delta x = x_v + v_r x_r$ where v_r^T is an n-r × n lower portion of the matrix V^T. The latter is a unitary matrix related to the singular value decomposition of a, i.e., $a = U \Sigma V^T$, where U is m x n orthogonal matrix, V is an n × n orthogonal matrix and Σ is an n × n diagonal matrix with its first r diagonal elements are the singular values of a and the rest are zeros. m and n are the dimension of a and we assume $m \leq n$. r is the rank of a. Specifically, for our case, m = r = 1 and n equal to the number of tuning parameters. x_r is obtained as the solution of:

$$\min_{x_r} \| x_v + v_r x_r \|_2$$

subject to:

$$\Delta x^l - x_v \leq v_r x_r \leq \Delta x^u - x_v$$

The optimization problem is solved by MATLAB software. The above algorithm has one parameter, namely, P_w. The prediction horizon provides advance prediction of the behavior of the closed-loop response which may result in earlier correction of the PI parameters. The user should also provide nominal performance envelopes with specific window size for both servo and regulator problems. The servo performance envelope is designed for unit step while that for the regulator case should be designed in a way that reflect the user understanding of the process. In the regulator case, the user should also provide threshold values within which the output variation is considered tolerable. Nevertheless, the tuning algorithm will be automatically triggered, and the nominal envelopes will be automatically adjusted at the triggering instant. The algorithm triggering and envelope adjustment are conducted on-line according to the process transient behavior as discussed in the following section.

Activation of the algorithm

Process operating at steady state

The adaptation algorithm will be monitored every sampling interval to check whether the closed-loop predicted process output violate its corresponding threshold value or not. The adaptation algorithm will then be triggered only when any of the outputs violates its threshold value due to the influence of disturbance. The algorithm will be turned off again when the time exceeds the window size of the specs.

The tuning algorithm will scale the performance specs to properly suit the actual behavior of the process under disturbance. To achieve this, a scaling factor will be computed on-line at the triggering point based on an estimate of the actual effect of the disturbance on the output. At the triggering point say, (*k*), the estimate of the disturbance effect on the output *j*, which possesses the maximum violation of its threshold, is given by:

$$d_j = yp_j(k) - ym_j(k)$$

where *ym* is the model process output. Hence the scaling factor can be computed as follows:

$$S_j = \frac{\alpha d_j}{\max\limits_{1 \le l \le n_s} ybd(l,j)} \tag{12}$$

where ybd is the nominal disturbance specs and n_s is the size of specs window. α is a relaxation tuning parameter. A value of 2 for α means that S_j is equal to 1 when d_j is half of the maximum value of the specs. The nominal envelope specs for output j will be scaled by S_j. For the other outputs, the upper bound of their corresponding nominal specs will be scaled by the absolute value S_j, while the lower bound will be changed to the mirror image of the upper bound after scaling.

Process operating at changing set point

In this case, the adaptation is triggered at the time instant at which the set point change occurs. When the time exceeds the window size of the specs, the algorithm will then be turned off, and nominal disturbance specs are assigned to the output with set point change assuming that the output has settled at its new set point. In this case, scaling the nominal envelopes is straightforward. Each output with changing set point is scaled with the magnitude of its corresponding set point change. Outputs with fixed set points are scaled by the absolute value of the maximum set point change.

To avoid getting ambiguous scaling factor, the scaling factor for outputs with fixed set points in the above two cases is constrained between upper and lower limits (S^u, S^l) which are determined by the user.

Nominal stability conditions

A criteria for checking the nominal stability of the on-line tuning algorithm can be developed. The criteria is based on calculating the poles of the PI feedback response for the nominal plant. Given the values of the process model states at any sampling time, say k, and the new computed values for the PI tuning parameters, equations (1) and (4) can be linearized around that point, which will result in a linearized closed-loop state space model. Therefore, to achieve nominal stability for the on-line tuning, all the eigenvalues of the dynamic matrix of the linearized model should lie strictly in the left half plane. This condition does not guarantee robust stability. However, it can, at least, give some degree of confidence in the new computed values of the PI settings. If the nominal stability conditions is satisfied, then the new PI settings can be implemented. Otherwise the old values of the PI settings should be used instead.

SISO Example

In the following the effect of the proposed tuning procedure on a linear SISO control example is examined. The process is described by the following equation:

$$\dot{x} = 0.05u(t - D)$$

$$y = x$$

where D is dead time of 3 minutes. In all the following simulations, a sampling time of 1-min will be used, lower and upper bounds of 1 and 50 respectively will be imposed on k_c, and lower and upper bounds of one-tenth and fifteen times the sampling interval will be imposed on τ_i. The nominal envelop for the servo case is designed to allow 22% overshoot and to maintain the ultimate response within 11% of the new set point. The nominal envelop for the regulator case is designed to allow 22% overshoot and to maintain the ultimate response within 2.8% of the steady state value. Each performance envelop has a window size of 75-min and will be represented by thick solid line in all the illustrative figures. A threshold value of ?? 0.05 as deviation from steady state value is used. The tuning algorithm will be tested using two initial PI settings. One is obtained by Ziegler and Nichols [9] method denoted as (Z-N) settings in which *kc* and τ_I are set equal to 3.18 and 11.7, respectively. The other one is an arbitrary value (AV) of 7 and 14 respectively. The model constants are similar to those used by Tyreus and Luyben [6] while the values of the performance specs are chosen arbitrary but it depends on the user, understanding the underlying process.

Simulation of this process for a set point change of 0.5 using the Z-N settings as the initial value is shown in Fig. 1a. The light solid curve in that figure represents the closed-loop response for fixed Z-N settings. The closed-loop responses of the proposed tuning algorithm are shown in the same figure for three different values of P_w and are represented by the dashed, dash-and-dot, and dash-and-double dots curves. Since this is a servo case, the algorithm was triggered at the beginning of the set point change, and turned off when time is 75-min at which the envelop is switched to its threshold values.

It is clear from Fig. 1a that the adapted closed-loop responses are almost similar to that using fixed Z-N settings. The time profile of kc and τ_I for these three values of P_w is illustrated by Fig. 1(b-c). Similar parameters profiles with small difference in magnitude were obtained for all values of P_w. In these cases, about 1% increase in kc, which is the wrong action, and 35% increase in τ_I were made by the tuning algorithm in order to reduce the overshoot of the controlled variable which in turn caused slight difference in their corresponding closed-loop responses compared to that for fixed Z-N settings. Nevertheless, the adapted responses obtained were somewhat reasonable.

The strength of the tuning algorithm becomes clearer when we start with initial bad values for the PI settings such as the AV settings. The results for this case are demonstrated by Fig. 2a for the same three values of P_w as before. The adaptation algorithm is triggered twice. First, the algorithm is triggered due to set point change, then turned off at time 75-min at which it was triggered again because the closed-loop response was still violating the specs. Apparently, the resulted feedback responses for all values of P_w are more conservative than that for fixed AV settings which is represented by the light solid curve. Despite the larger overshoot of the controlled variable in the adapted cases, the responses are much less oscillatory than that of the fixed AV settings. Although the resulted responses are not as good as that resulted from using the fixed Z-N settings shown in Fig. 1a, they are reasonable and easily obtained without the trial and error procedure associated with the Ziegler and Nichols method [9]. The time profiles for the adapted PI settings are shown in Fig. 2(b-c). The transient

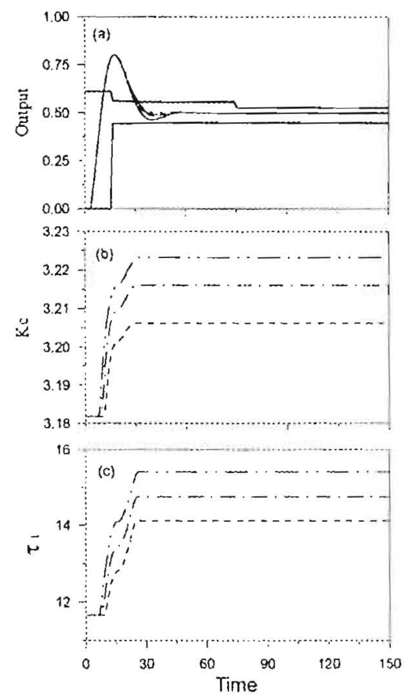

Fig. 1. Set point change response for the SISO example using Z-N settings as initial values; (a) output, (b) kc, (c) τ_I, thick solid lines: bounds; solid lines: Z-N settings; dashed lines: $P_w = 1$; dash-and-dot lines: $P_w = 3$; dash-and-double dots lines: $P_w = 5$.

behavior of kc and τ_I are almost the same for all values of P_w. In all cases, kc was reduced substantially, which is the expected action, to eliminate the aggressiveness in the controlled variable response. However, this reduction was associated with reduction in the values of τ_I which in turn increased the period of oscillation. Overall, the large drop in the value of kc resulted in smaller decay ratio than when no adaptation of the tuning parameters is done, i.e. when the PI settings are fixed at their arbitrary values. At $P_w = 5$ earlier prediction of the specs violation is detected by the tuning algorithm which lead to faster correction of the PI settings than for the smaller values of P_w as illustrated by Fig. 2(b-c). This explains the improved time response at $P_w = 5$.

The effectiveness of the proposed tuning algorithm is also tested for disturbance rejection simulation. The disturbance is a unit step change in a load variable that enters the process via a 4-min dead-time and a 10-min first-order lag. Similar simulation is also investigated by Tyreus and Luyben [6]. Figures 3a,b,c give the results for the case

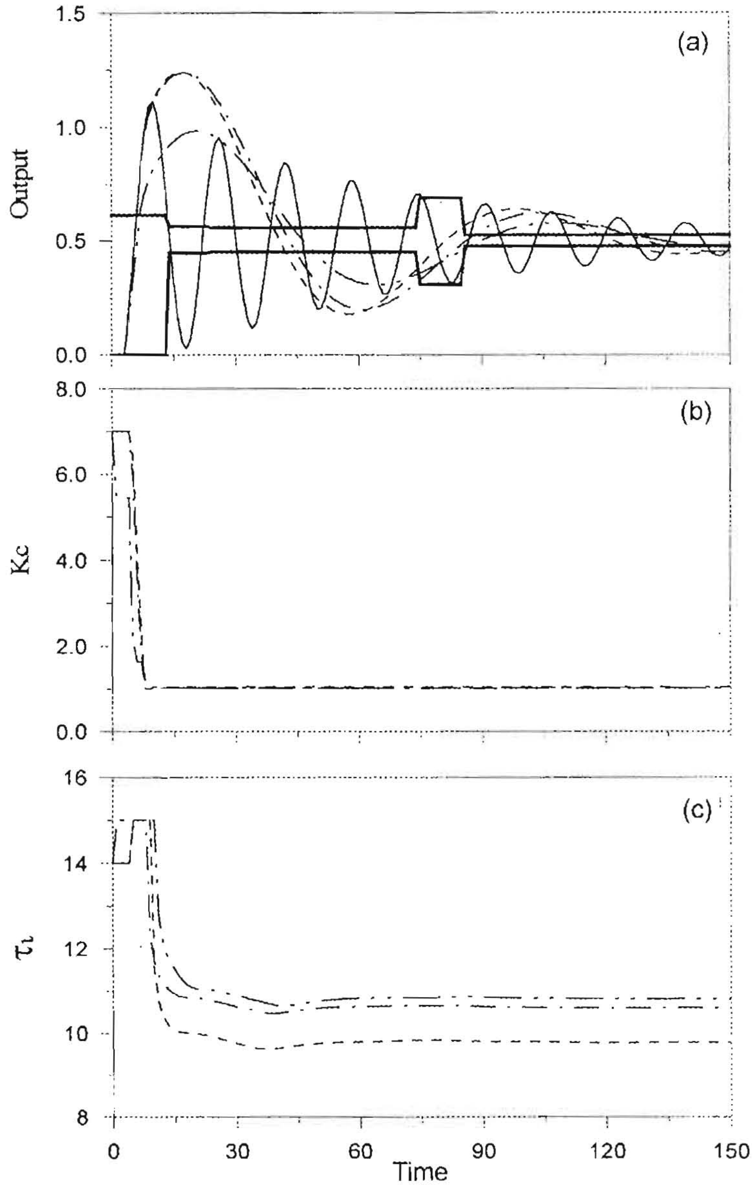

Fig. 2. Set point change response for the SISO example using arbitrary settings as initial values; (a) output, (b) kc, (c) τ_I , thick solid lines: bounds; solid lines: Z-N settings; dashed lines: $P_w = 1$; dash-and-dot lines: $P_w = 3$; dash-and-double dots lines: $P_w = 5$.

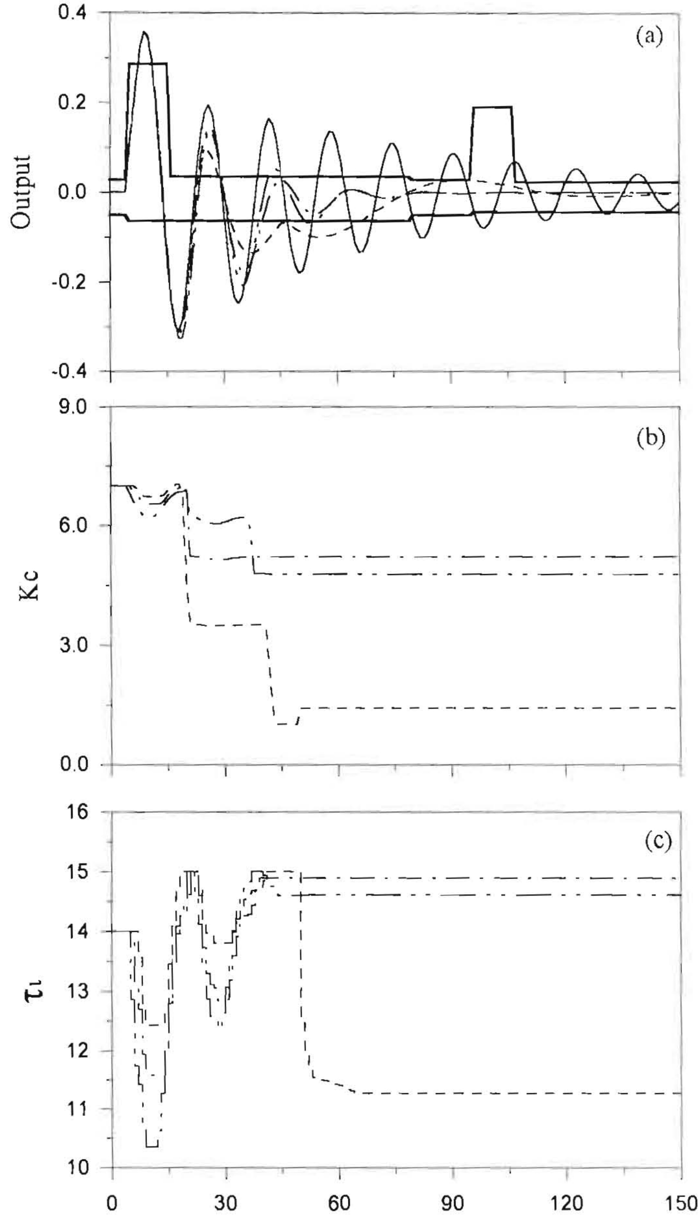

Fig. 3. Load change response for the SISO example using arbitrary settings as initial values; (a) output, (b) kc, (c) τ_I, thick solid lines: bounds; solid lines: Z-N settings; dashed lines: $P_w = 1$; dash-and-dot lines: $P_w = 3$; dash-and-double dots lines: $P_w = 5$.

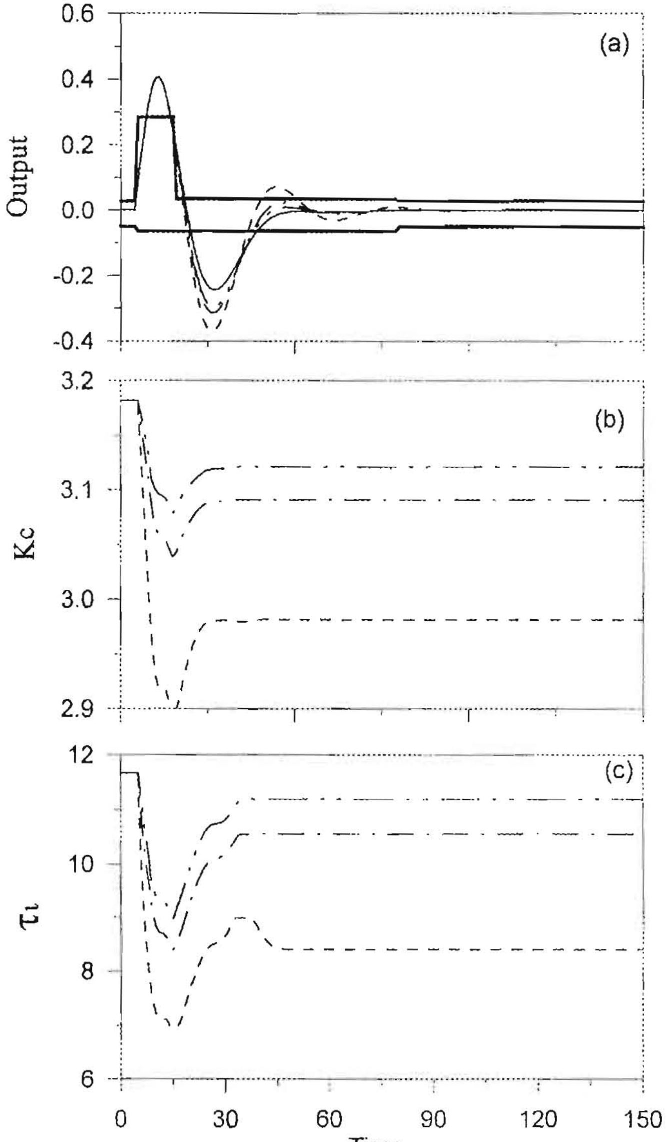

Fig. 4. Load change response for the SISO example using Z-N settings as initial values; (a) output, (b) kc, (c) τ_I, thick solid lines: bounds; solid lines: Z-N settings; dashed lines: $P_w = 1$; dash-and-dot lines: $P_w = 3$; dash-and-double dots lines: $P_w = 5$.

where the AV settings were used as initial values. The tuning algorithm is triggered after the dead time has elapsed and the threshold was violated, turned off at time 75 min and the envelope is switched to the threshold value. It is triggered again at 95 minutes, since the predicted closed-loop response still violating the specs. It is clear from Fig. 3a that the adapted responses for all values of P_w are superior to that of using fixed AV settings. They are also somewhat comparable to that using fixed Z-N settings which is shown by the light solid curve in Fig. 4a. Figure 3(b-c) demonstrates the corresponding time variation of the adapted PI settings. The PI settings transient behavior for $P_w = 1$ is different than those for the other values of P_w which explains the difference between their corresponding output feedback responses.

Testing the tuning algorithm for the same above load change but using the Z-N settings as the initial values is also examined. Successful reasonable results were obtained. Simulations of the controlled variable for fixed and varying PI settings are shown in Fig. 4a. The corresponding transient behavior of the PI settings is shown in Fig. 4(b-c).

It should be pointed out here that the feedback responses obtained using PI settings adaptation are not necessarily superior to those obtained by other tuning methods. In fact, our simulations illustrated that the adapted responses are not even as good as those obtained by Ziegler and Nichols method. However the major advantage of this method is that it is tractable and computationally undemanding. The nominal stability conditions for all the above simulations were examined and found to be satisfied. It should also be noted that plots of the manipulated variables for all the above simulations are omitted for brevity. Similar action will be taken in the next example.

MIMO Example

This control problem is adapted from Sand Bequette [19]. The objective in that paper is to control the temperature of an exothermic reaction taking place in a CSTR by manipulating the cooling jacket temperature. In this paper, we consider controlling both the reactor temperature and the product concentration through manipulation of the cooling jacket temperature and the cooling water flow rate. The dimensionless model equations are given by:

$$
\begin{aligned}
\dot{x}_1 &= -\phi\, k(x_2) + u_1(x_{1f} - x_1) \\
\dot{x}_2 &= \beta\phi\, x_1 k(x_2) - (u_1 + s)x_x + su_2 + u_1 x_{2f} \\
y_1 &= x_1 \\
y_2 &= x_2
\end{aligned}
$$

where $k(x_2) = \exp(x_2/(1+x_2/20))$, x_1 is the dimensionless concentration, x_2 is the dimensionless reactor temperature. u_1 is the cooling water flow rate and u_2 is the cooling jacket temperature. Initial conditions are $u_1 = 1$, $u_2 = 0$, $x_1 = 0.856$, and $x_2 = 0.8859$, which correspond to a stable steady state point. The nominal values of the model physical parameters are $\beta = 8.0$, $s = 0.3$, $\phi = 0.072$, $x_{1f} = 1.0$, and $x_{2f} = 0.0$. Upper bounds of 2, and lower bounds of 0 and -2 are imposed on u_1 and u_2 respectively.

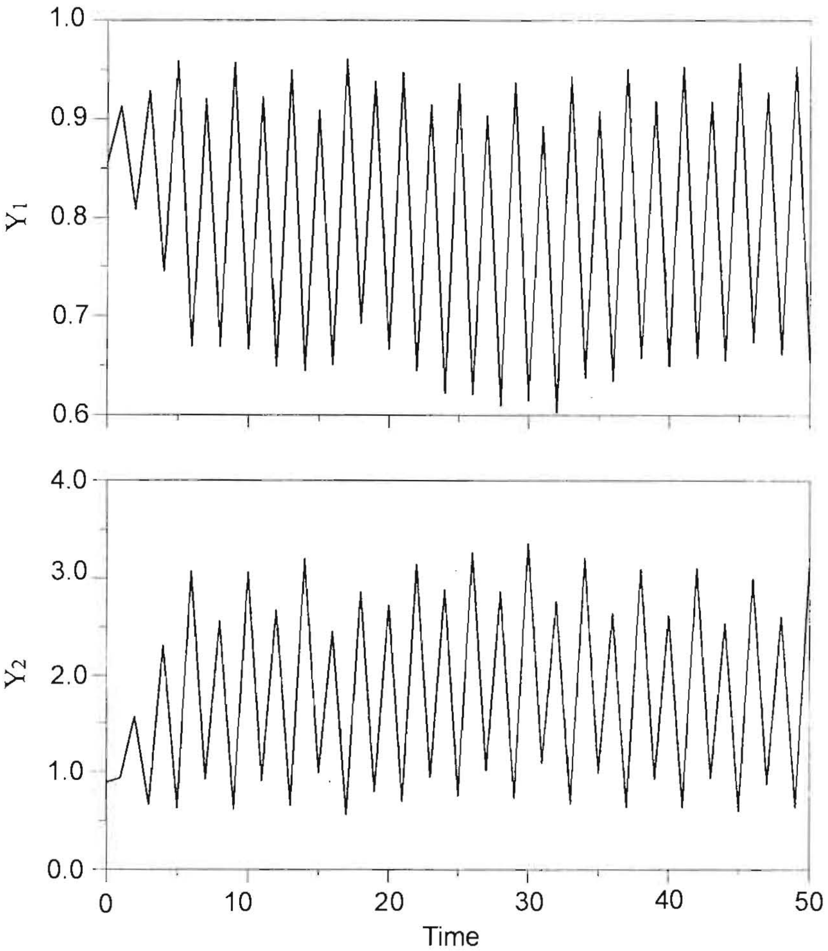

Fig. 5. Load change response for the MIMO example using fixed arbitrary settings.

Closed-loop simulation of the process for load change of 10% in the feed composition (x_{1f}) is shown in Fig. 5. Fixed arbitrary values for the controller settings used in that simulation are listed in Table 1. The poor unstable regulatory performance illustrated by Fig. 5 is indicative of badly tuned controller. Next the effectiveness of the tuning procedure to improve the controller performance will be examined. The previous example showed that the tuning problem is more challenging when arbitrary values for PI settings are used as the initial values in the tuning algorithm. Thus, the following tests will be carried out only for the case where arbitrary initial values of the tuning parameters are used. The nominal performance envelopes for the two controlled variables are designed such that they limit the overshoot to 10% and bring the asymptotic values of the outputs to within 1% of their corresponding steady state values. The window size of the performance envelopes is set to be 30 time-units, and the threshold values are set equal to 1% of the outputs steady state values. The controller gain for each controller loop will be bounded between 1 and 50. while the reset time for each loop will be bounded between one tenth and fifteen sampling times.

The feedback simulation for the same above load change using the proposed tuning procedure is shown in Fig. 6 for two values of P_w. The figures also include the controller performance using fixed Z-N settings which is represented by the solid curves. The latter performance is included for comparison purposes. The values of the Z-N settings are listed in the Table.

The thick solid lines represent the performance envelopes after adjustment which indicate that the tuning algorithms have been triggered twice. In both cases it was triggered by the response of the second output which had the maximum bound violation. The second triggering instant occurred because the predicted output violated the specs. This explains why the tuning algorithm was activated again even if the figure shows that the controlled outputs were within their bounds during the second triggering point. The dashed lines display the performance for the adapted tuning parameters. The resulted performance for the two values of P_w indicates much improvement over the one obtained for fixed arbitrary settings. In fact, the closed-loop response at $P_w = 5$ is much less aggressive than that for $P_w = 3$. Nevertheless, despite the initial excessive overshoot and undershoot responses, the tuning procedure managed to stabilize the response through successful on-line tuning of the settings as illustrated in Fig. 7(a-c). As shown by these figures, the controller gains decrease which is a spontaneous reaction to slow down the response and remove the aggressiveness. This also coincides with the tuning guidelines of traditional PI controller. Still, the improved performance is inferior to that of the Z-N settings. However, as mentioned earlier the improved performance obtained by using the proposed adaptation technique is automatically achieved on-line with little effort. While the excellent performance obtained by the Z-N settings is achieved off-line by trial-and-error procedure.

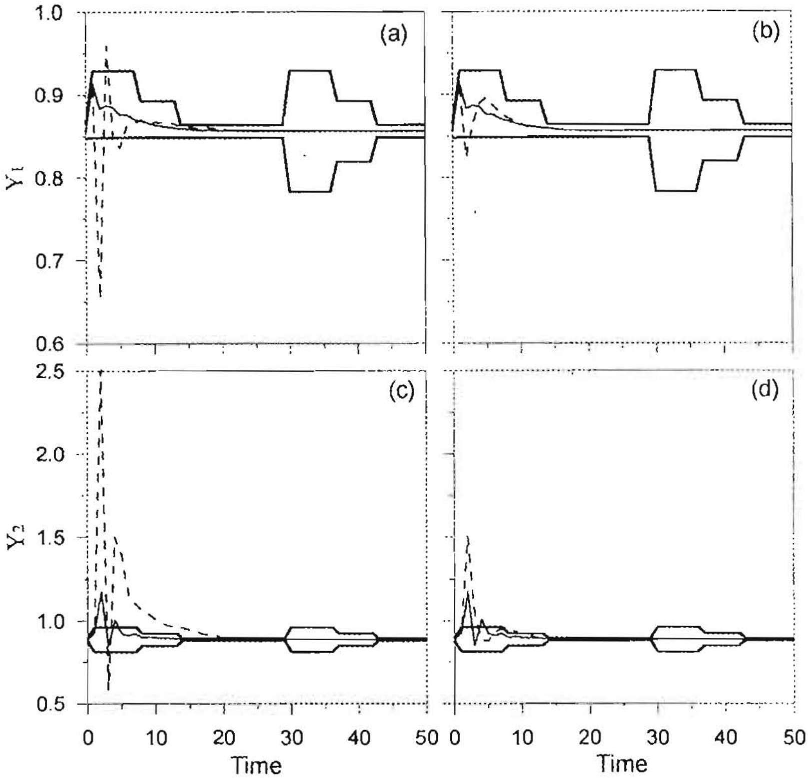

Fig. 6. Load change response for the MIMO example using adapted settings with arbitrary initial values, thick solid lines: bounds; solid lines: Z-N settings; dashed lines: adapted. (a,c) $P_w = 3$, (b,d) $P_w = 5$.

Table. PI controller settings

Parameter	Arbitrary	Ziegler Nichols
kc_1	7	4.5
kc_2	5	2
τ_{I1}	2	1.5
τ_{I2}	2	1.5

The adaptation algorithms when the process model contains some degree of parametric uncertainty were also examined. Specifically, the dimensionless heat transfer coefficient (s) is taken to be 0.2 for the model and 0.3 for the plant. This is also used by Sistu and Bequett [19] to examine the robustness of their control algorithm. The closed-loop test for the same above load change is repeated using the proposed tuning algorithm for two values of P_w and starting with an arbitrary value for the controller settings as listed in the Table. The results are shown in Fig. 8. which indicates satisfactory tuning of

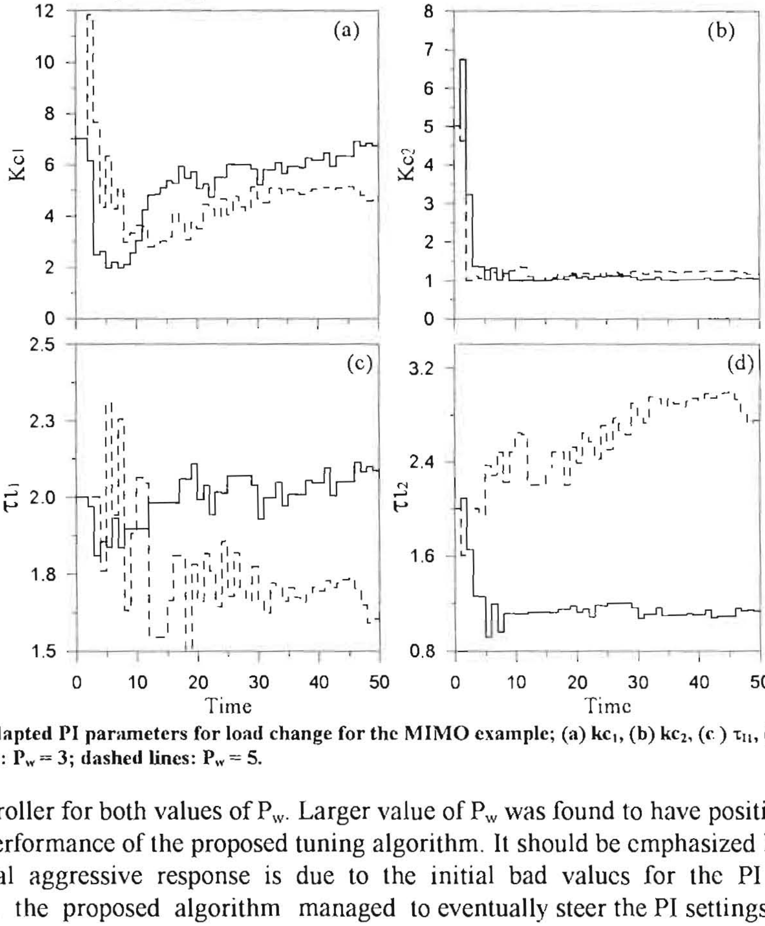

Fig. 7. Adapted PI parameters for load change for the MIMO example; (a) kc_1, (b) kc_2, (c) τ_{I1}, (d) τ_{I2}solid lines: $P_w = 3$; dashed lines: $P_w = 5$.

the controller for both values of P_w. Larger value of P_w was found to have positive effect on the performance of the proposed tuning algorithm. It should be emphasized here that the initial aggressive response is due to the initial bad values for the PI settings, however, the proposed algorithm managed to eventually steer the PI settings to other good values. The satisfactory result, even in the presence of the modeling error, is ascribed to the correction of the model output by adding the disturbance estimates to it as mentioned in the *controller on-line tuning* section. The tuning algorithm is triggered twice although in the second triggering point, the transit response is inside the envelope of the specs. This is because the triggering procedure is based on the model predicted output and not on the process output shown in the figures. Therefore, due to the model-plant mismatch induced by the modeling error, the model predicted output might violate the specs, which consequently initiate the algorithm, even if the actual process output does not. The time profiles for the adapted PI settings are omitted here for brevity. It should be clear that the nominal stability conditions were monitored and found to be satisfied.

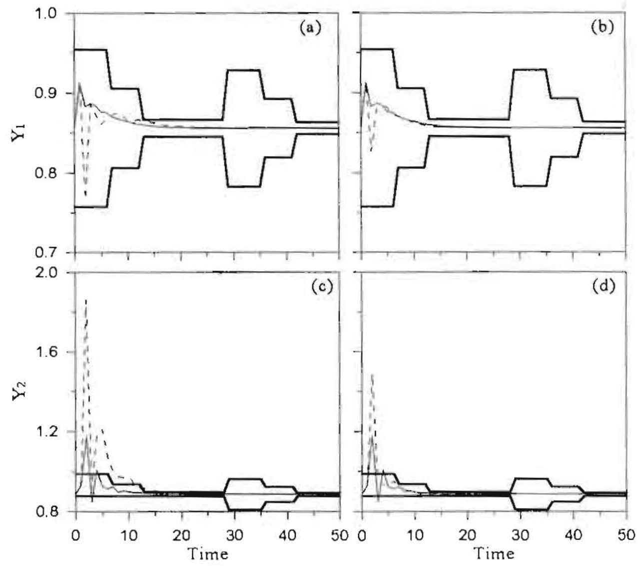

Fig. 8. Load change response for the MIMO example with imperfect model using adapted settings with arbitrary initial values, thick solid lines: bounds; solid lines: Z-N settings; dashed lines: adapted. (a,c) $P_w = 3$, (b,d) $P_w = 5$.

Conclusion

A simple on-line model-based tuning strategy for PI controllers is presented. The method directly uses the sensitivities of the closed-loop response to the PI settings to obtain new values of the latter in order to steer the feedback response inside a pre-defined time-domain performance specs. The effectiveness of the methods is tested on dead-time integrator process for set point and load changes and on MIMO non-linear process for load change. In both cases, whether starting from good or bad initial values for the PI settings, satisfactory and reasonable feedback response can be obtained with little effort and minor computations. Also, successful results were obtained even in the presence of modeling error. The investigation showed that increasing the closed-loop prediction horizon P_w improved the resulting feedback response. Nevertheless, the results were not very sensitive to the variation of P_w, thus, the tuning parameters of the proposed algorithm are kept at the minimum. The algorithm, however, considers only nominal stability conditions. Robust stability condition is an important issue that should be addressed in future work.

References

[1] Abbas, A. and Sawyer, P. ``A Multiobjective Design Algorithm: Application to the Design of SISO Control Systems." *Computers Chem. Eng.*, 19, 241(1995).

[2] Chen, C. ``A Simple Method for On-line Identification and Controller Tuning." *AIChE*, 35, 2037 (1989).

[3] Jietae, Lee. ``On-line PID Controller Tuning from a Single, Closed-loop Test." *AIChE*, 35, 329(1989).
[4] Seaman, C., Desrochers A., and List, G. ``Multiobjective Optimization of a Plastic Injection Molding Process" *IEEE Trans. Automatic Control*, 2, 157(1994).
[5] Semino, D. ``Automatic Tuning of PID Controllers for Unstable Processes," ACC, (1992).
[6] Tyreus, B. and Luyben, W. ``Tuning PI Controllers for Integrator/Dead Time Processes." *Ind. Eng. Chem. Res.*, 31, 2625 (1992).
[7] Yuwana, M. and Seborg, D.E. ``A New Method for On-line Controller Tuning." *AIChE J.*, 28, 434(1982).
[8] Zhu, W. and Lee, P. ``An Adaptive Strategy for Constrained Generic Model Control." *Chem. Eng. Comm.*, 97, 109(1990).
[9] Ziegler, J. G. and Nichols, N.B. ``Optimum Settings for Automatic Controllers." *Trans. ASME*, 64, 759 (1942).
[10] Kavdia, M. and Chidambaram, M. ``On-line Controller Tuning for Unstable Systems.`` *Comp. & Chem. Eng.* 20, 301 (1996).
[11] Park, H., S, S., Lee, I. and Lee, J. ``On-line Process Identification Using the Laguerre Series for Automatic Tuning of the Proportional-integral Derivative Controller.`` *Ind. Eng. Chem. Res.*, 36, 101 (1997).
[12] Srinivas, N. and Chidambaram, M. ``Comparison of On-line Tuning Methods for Unstable systems.`` *Process Control & Quality*, 8, 177 (1996).
[13] Parasiliti, F., Tursini, M. and Zhang, D. ``On-Line Self Tuning of PI Controllers for High Performance PMSM Drives.`` *IEEE Industry Applications Society*, Piscataway, NJ., 3 (1996).
[14] Khuen, Ho, W., Chieh Hang, C. and Cao, L. ``Tuning of PID Controllers Based on Gain and Phase Margin Specifications.`` *Automatica*, 3, 497 (1995).
[15] Ha, Q. ``Proportional-integral Controllers with Fuzzy Tuning.`` *Electronic Letters*, 32, 1043 (1996).
[16] Jones, K., Williams, D. and Goodier, A. ``Auto-tuning PID Control of Dissolved Oxygen in a Saccharomyces Cerevisiae Fermentation.`` *Measurement and Control*, 29, 266 (1996).
[17] Yusof, R., Omatu, S. and Khalid, M. `` Self-tuning PID Control: A Multi-variable Derivation and Application.`` *Automatica*, 30, 1975 (1994).
[18] Rawlings, J.B., Meaddows, E.S. and Muske, K.R. ``Nonlinear Model Predictive Control: A Tutorial Survey", *IFAC Advanced Control of Chemical Processes*, Kyoto, Japan (1994), 185-196.
[19] Sistu, P. and Bequette, B. ``Nonlinear Predictive Control of Uncertain Processes: Application to A CSTR.`` *AIChE J.*, 37, 1711 (1991).

خطه لتحديد معاملات الضبط لنظام التحكم النسبي التكاملي

عماد الدين علي

قسم الهندسة الكيميائية ،كلية الهندسة، جامعة الملك سعود ، ص . ب ٨٠٠ ،

الرياض ١١٤٢١ ، المملكة العربية السعودية

(استلم في ١٩٩٧/٥/٢٠م ؛ و قبل للنشر في ١٩٩٨/٣/٤م)

ملخص البحث. يدرس هذا البحث تصميم خطه مبنية على النمذجه لضبط عوامل نظام التحكم الكلاسيكي المعروف بالنسبي التكاملي. الغرض هو تحديد قيمة عوامل الضبط التي تجعل الاستجابة الزمنية المغلقة تتوافق مع الأداء المرغوب فيه المتمثل على شكل إطار زمني مما يجعله أكثر تقبلا من الناحية التطبيقية. تعتمد خطة الضبط على إيجاد علاقه خطية بين المعاملات و الاستجابة الزمنية المغلقة، و كذلك علاقه لحساسية الاستجابة الزمنية المغلقة مع معاملات الضبط. و من ثم تستخدم هذه العلاقات الرياضية مباشرة لإيجاد قيم جديدة لتلك المعاملات. كما تم تطوير ظروف التشغيل المستقر للنموذج الافتراضي و التي تستخدم كمؤشر لاستقرارية الاستجابة الزمنية المغلقة. درست كفاءة الخطة المقترحة عن طريق المحاكاة بمثالين. **الأول** هو نموذج خطي لعمليه غير مستقرة ذات متغير واحد. **الثاني** هو نموذج لاخطي لعملية متعددة المتغيرات. كما درست جودة الخطة عند وجود أخطاء نمذجية و ذلك في المثال الثاني فقط.