17

New PI Controller Tuning Methods Using Multi-Objective Optimization

Allan Vandervoort, Jules Thibault and Yash Gupta
Department of Chemical and Biological Engineering, University of Ottawa, Canada

17.1 Introduction

Efficient process control is an essential element in ensuring industrial chemical plants operate economically and optimally in a safe manner, while meeting product specifications and environmental regulations [1]. Despite tremendous advances in process control and the development of numerous control algorithms, the PI controller still remains the most commonly used control algorithm in industrial applications [2, 3, 4, 5]. The reason for its widespread industrial utilization is its simplicity and ease of implementation. In addition, when significant dead time is present the derivative term in a PID controller leads to an incorrect response such that a PI controller is favored [4, 5]. The development of efficient and robust tuning methods for PI controllers is therefore very important.

A properly configured controller for a chemical process should be robust, minimize excessive controller action, and produce a stable response with no final offset [6]. Many controller correlations have been developed for tuning PI controllers such as those proposed by Ziegler and Nichols [7], Cohen and Coon [8], Chien and Fruehauf [9], and Skogestad [10]. Although these and many other correlations have been implemented in process-control systems, no controller correlation can achieve all of the desired performance criteria simultaneously because they inherently involve conflicts and tradeoffs [6].

In recent years, the multi-objective optimization (MOO) of PI controllers has been studied in an attempt to better understand the tradeoff between various controller objectives. In many of these studies, MOO is performed for the control of specific processes [11, 12]. In

Multi-Objective Optimization in Chemical Engineering: Developments and Applications, First Edition.
Edited by Gade Pandu Rangaiah and Adrián Bonilla-Petriciolet.

other studies, general tuning rules have been developed by considering multiple objectives [13, 14] but in these methods only one set of controller parameters was suggested for a given process. Although the tradeoffs between each performance objective were considered by the authors during the development of these tuning methods, they cannot be considered by the decision maker when the tuning methods are applied to industrial control systems. In this investigation, new PI controller-tuning methods based on MOO are proposed. The tuning methods in this study, where multiple objectives are simultaneously optimized, were developed to improve the decision maker's understanding of the tradeoff associated with each objective, before optimum controller parameters are chosen.

17.2 PI Controller Model

In this study, a first-order plus dead time (FOPDT) process model was used for the development of the PI controller tuning methods. A FOPDT process transfer function is frequently used for controller optimization, as it adequately represents a large number of higher-order systems and industrial processes [15, 16, 17]. A FOPDT process can be represented in the Laplace domain by Equation (17.1) [18].

$$\frac{y(s)}{u(s)} = \frac{K_p e^{-\theta s}}{\tau s + 1} \tag{17.1}$$

Equation (17.1) is solved numerically using finite differences. The resulting expression is shown in Equation (17.2).

$$y_t = y_{t-\Delta t} + \frac{\Delta t}{\tau}\left[K_p u_{t-\Delta t-\theta} - y_{t-\Delta t}\right] \tag{17.2}$$

where u_t represents the manipulated variable at time t, K_p represents the process gain, τ represents the process time constant, Δt represents the small integration time step used in the calculation, y_t represents the controlled variable at time t, and θ represents the time delay.

For a PI controller the equation describing the response of the manipulated variable is shown in Equation (17.3) [18] for a bias term of zero.

$$u_t = K_c \varepsilon_t + \frac{K_c}{\tau_I}\int_0^t \varepsilon_t \, dt \tag{17.3}$$

Equation (17.3) is again solved numerically using finite differences as shown in Equation (X.4), which represents the velocity form of the PI controller.

$$u_t = u_{t-\Delta t} + K_c(\varepsilon_t - \varepsilon_{t-\Delta t}) + \frac{K_c}{\tau_I}\varepsilon_t \Delta t \tag{17.4}$$

where ε_t is the error at time t, K_c is the controller gain, and τ_I is the integral time.

17.3 Optimization Problem

Although several variables affect the controller performance, only the integral time τ_I and the controller gain K_c can be varied when tuning a PI controller. In this study, relative controller parameters were used to ensure that the optimization results could be generalized for all possible FOPDT processes. Specifically, the relative controller gain ($K_c K_p$) and relative integral time (τ_I/τ) were used as input variables. The feasible region for each input variable was set between 0.1 and 10.

For the tuning of a PI controller, many performance criteria can be used alone or as a weighted sum of two or more. In this investigation, MOO is performed for a FOPDT process subject to a unit set point change in the controlled variable. Three controller performance criteria were considered in the optimization: the integral of the time weighted absolute error (ITAE), the integral of the squares of the differences in the manipulated variable (ISDU) and the settling time. The mathematical expressions of ITAE and ISDU are given in Equations (17.5) and (17.6), respectively. The ITAE measures the cumulative deviation of the controlled variable from the set point, and penalizes deviations that are not resolved in a short period of time. ISDU measures the changes in the manipulated variable and favours a smooth response [6].

$$ITAE = \int_0^{t_f} t\ |\varepsilon_t|\, dt \simeq \sum_{k=1}^{t_f/\Delta t} t\ |\varepsilon_k|\ \Delta t \tag{17.5}$$

$$ISDU = \int_0^{t_f} \Delta u_t^2 dt \simeq \sum_{k=1}^{t_f/\Delta t} (u_k - u_{k-1})^2\ \Delta t \tag{17.6}$$

The settling time is defined in this investigation as the time that the process takes to stabilize within $\pm$ 5% of the final steady state value following a step change in the controlled variable. The settling time for a typical response from a unit step change in the controlled variable is demonstrated in Figure 17.1.

To further ensure that the controller optimization results can be generalized for all possible FOPDT processes, relative objective criteria were chosen for this optimization. The objectives chosen were the relative ITAE (ITAE/τ^2), the relative settling time (t_{set}/τ) and the ISDU. The ISDU objective did not directly depend on the response time of the controller and was not placed in relative form because the tuning methods in this study were developed for a unit step set point change. The optimization was performed for varying values of the relative dead time (θ/τ). The optimization problem is shown schematically in Figure 17.2.

17.4 Pareto Domain

Before ideal values of the relative integral time and relative controller gain are chosen by considering the tradeoffs between the three competing criteria specified in this study, the domain of potentially optimal solutions should be circumscribed so that the search for the optimal solution only needs to consider solutions that are located within this domain. This set of potentially optimal solutions or non-dominated solutions is known as the Pareto domain.

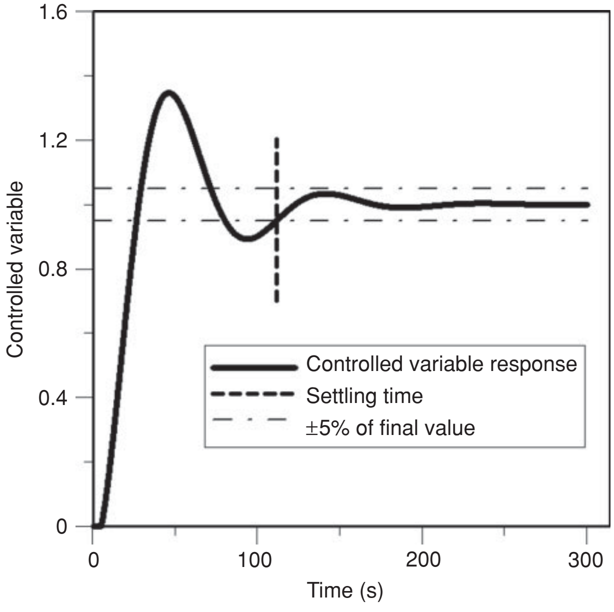

Figure 17.1 *Graphical illustration of the settling time for a typical response.*

17.4.1 Dominated and Non-Dominated Solutions

In Figure 17.3, a graphical representation of dominated and non-dominated solutions and an associated Pareto domain is presented using two objective functions, f_1 and f_2, to be maximized and corresponding to two input or decision variables, x_1 and x_2. Four solutions are used to illustrate the concept of dominance. The values for both objective functions for point A are lower than the other three points such that point A is dominated relative to these three points and, of course, point A will never be considered optimum. Point B dominates point A, both its objective function values are lower than those of points C and D such that point B is also a dominated solution relative to C and D. On the other hand, comparing solutions of points C and D, each solution is at least better for one objective such that points C and D are non-dominated solutions. Normally, to adequately define the Pareto domain, a similar comparison is performed on a large number of solutions and only non-dominated solutions are retained potentially optimal solutions.

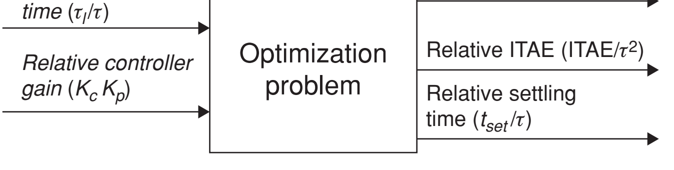

Figure 17.2 *Optimization problem showing input variables and objective functions.*

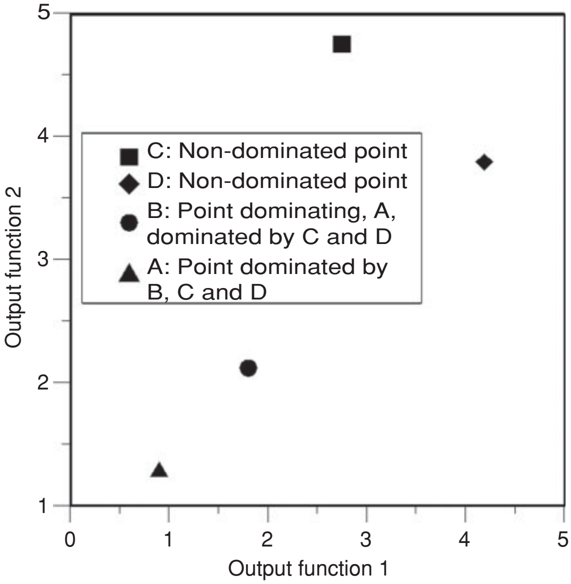

Figure 17.3 Graphical representation of the concept of dominance used to determine Pareto-optimal solutions.

For a description of the concept of dominance, consider two solutions, points P_1 and P_2, consisting of n decision variables (x_1, x_2, x_3 . . . x_n) and m values of objective functions or performance criteria (f_1, f_2, f_3 . . . f_m). Point P_1 dominates point P_2, if the following two conditions prevail [19]:

- All objective function values of P_1, f_1 to f_m, are not worse than the corresponding objective function values of P_2, f_1 to f_m. In other words, if all performance objective criteria need to be maximized then all objective function values of P_1 need to be equal to or greater than the corresponding objective function values of P_2.
- At least one performance criterion for P_1 is better than the corresponding performance criterion for P_2.

The Pareto domain is therefore the set of solutions within the set of all feasible solutions containing exclusively the non-dominated points. Solutions located outside of the Pareto domain are the set of dominated points, and therefore not Pareto-optimal.

17.4.2 Few Methods for Approximating the Pareto Domain

17.4.2.1 Non-dominated Sorting Genetic Algorithm II

The non-dominated sorting genetic algorithm II (NSGA-II) is an MOO technique developed by Deb et al. [20], based on a genetic algorithm. The first step in using NSGA-II is to create randomly an initial population of solutions. For successive generations, all solutions in the current population are assessed based on the number of times each solution is dominated and the most fit individuals of the population, in an attempt progressively to improve the degree of fitness of the population, are allowed to procreate to generate new individuals using a process based on random variation and combination of solutions. NSGA-II is one of the most popular MOO method and used in numerous applications [21, 22, 23, 24, 25].

17.4.2.2 Grid Search Approach

The GSA is a technique used to approximate the Pareto domain that involves the construction of a grid in the input space of an optimization problem. The objective functions are then calculated at the grid points, and the best points are chosen based on the concept of dominance. The search begins from a coarse grid, and is refined at each iteration. The full GSA procedure is described below.

1. Initially a grid is formed in the feasible region of the input space. The number of initial divisions for each input is specified before beginning the GSA, with five initial divisions being used for each input in this study.
2. The values of the objective functions are then calculated for each point in the grid. The procedure generates $\prod_{i=1}^{n} (M_i + 1)$ points, where M_i equals the number of divisions for input i, and n equals the total number of inputs in the specified optimization problem.
3. Next, all points in the grid are compared to other points in the grid one at a time, and the number of times a given point is dominated by another is determined.
4. Using the domination count, the best points are then selected to determine the range in the input variables for the construction of the next grid. The minimum and maximum values of the input variables, that contain the non-dominated points, are used as the range for the next grid. As the accuracy of the Pareto domain increases with each iteration, the number of divisions is also increased. In this study, the number of divisions was increased by 5 at each iteration.
5. Steps 1–4 are repeated, gradually producing a finer grid in the input space. Once the grid interval for an input variable reaches a predetermined minimum value, the number of divisions is no longer increased for that input variable. Once all of the input variables have reached the predetermined minimum grid interval, the approximation of the Pareto domain is complete.

17.4.3 Application of Principal Component Analysis to the Grid Search Approach

Of the MOO techniques used to approximate the Pareto domain, a grid-based search is one of the simplest and easiest to implement. The GSA also ensures a Pareto domain that evenly and fully spans the input space. However, the GSA can lead to an approximation of the Pareto domain with limited accuracy, requiring high computation time [26]. In the study by Vandervoort [27], modifications were made to the GSA using principal component analysis (PCA). These modifications were performed with the goals of increasing the accuracy of the Pareto domain obtained, and reducing the required computation time.

When the GSA is used to approximate the Pareto domain, a large number of dominated points can be generated at each iteration, even as the grid interval decreases and the procedure nears convergence. An example of an optimization problem where this is especially important is demonstrated in Figure 17.4, which shows the input space of the Pareto domain

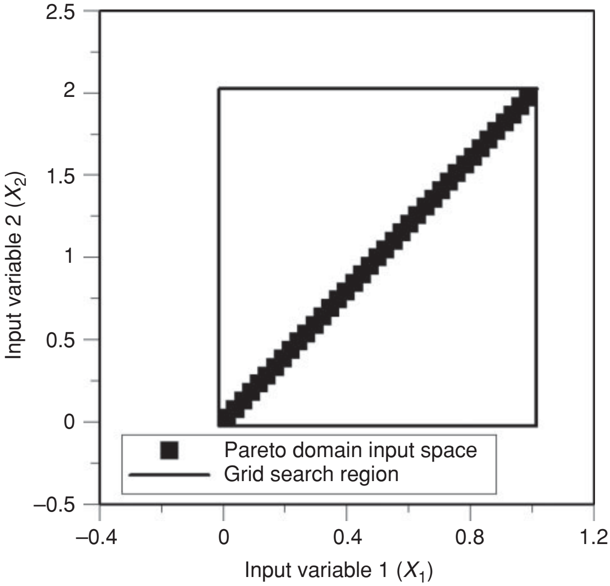

Figure 17.4 *Input space of the Pareto domain for the example problem.*

for a problem involving two input variables. The grid search region is also shown in the figure. In this example, the Pareto domain represents only a very small fraction of the grid search region. Decreasing the grid size will lead to a more accurate Pareto domain, but will also produce an increasing number of points in the grid search region that are not part of the Pareto domain. This limitation of the GSA leads to high computation time as each iteration generates many redundant points for which the objective functions must be calculated.

Principal component analysis (PCA) is a statistical tool that is most often used to convert a set of possibly correlated variables into a set of uncorrelated variables [28]. The example in Figure 17.4 is used below to demonstrate how PCA can benefit the GSA, and the required calculation procedure is described [28].

1. The first step in the procedure is to calculate the covariance matrix. The data set used for this calculation is the input space of the Pareto domain, i.e. the set of x_1 and x_2 values for the example problem. This calculation is given in Equation (17.7):

$$\begin{bmatrix} Cov(x_1, x_1) & Cov(x_1, x_2) \\ Cov(x_2, x_1) & Cov(x_2, x_2) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{N} \frac{\left(x_{1,j} - \bar{x}_1\right)^2}{N-1} & \sum_{j=1}^{N} \frac{\left(x_{1,j} - \bar{x}_1\right)\left(x_{2,j} - \bar{x}_2\right)}{N-1} \\ \sum_{j=1}^{N} \frac{\left(x_{2,j} - \bar{x}_2\right)\left(x_{1,j} - \bar{x}_1\right)}{N-1} & \sum_{j=1}^{N} \frac{\left(x_{2,j} - \bar{x}_2\right)^2}{N-1} \end{bmatrix} \tag{17.7}$$

where N is equal to the number of points in the input space and $\bar{x}_1$ and $\bar{x}_2$ are the average values of each input variable.

2. Next the eigenvalues are calculated. All optimization problems in this study consisted of two input variables. When two input variables are used, Equation (17.8) gives the eigenvalues of the dataset (λ_A and λ_B) using the quadratic formula:

$$\lambda_A = \frac{T + (T^2 - 4D)^{0.5}}{2}$$
$$\lambda_B = \frac{T - (T^2 - 4D)^{0.5}}{2} \tag{17.8}$$

where T is the trace of the covariance matrix, and D is the determinant of the covariance matrix.

3. Next the eigenvectors are calculated. The definition of an eigenvector V (with elements v_1 and v_2) corresponding to an eigenvalue λ for a system with two input variables is shown in Equation (17.9).

$$\begin{bmatrix} Cov(x_1, x_1) & Cov(x_2, x_1) \\ Cov(x_1, x_2) & Cov(x_2, x_2) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \lambda \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{17.9}$$

Equation (17.9) always yields two dependent equations for the eigenvector V. The process of finding vector V can therefore be simplified by setting v_2 to 1, and solving for v_1 from Equation (17.10):

$$v_1 = \frac{Cov(x_2, x_1)}{\lambda - Cov(x_1, x_1)} \quad \text{or} \quad v_1 = \frac{\lambda - Cov(x_2, x_2)}{Cov(x_1, x_2)} \tag{17.10}$$

The calculation described by Equations 17.9 and 17.10 is performed for both eigenvalues λ_A and λ_B, corresponding to the two eigenvectors V_A and V_B, each with distinct values for elements v_1 and v_2.

4. The principal component projection of the data set is then calculated. This calculation consists of multiplying the data set by the matrix composed of the eigenvectors V_A and V_B.

The calculation procedure outlined in steps 1–4 was performed on the data set shown in Figure 17.4. The projected dataset is shown in Figure 17.5.

In Figure 17.5, the data set is certainly ideal for the application of a grid. For this example the grid search region includes only a small fraction of redundant points, and the dataset has essentially been reduced from a two-variable system into a one-variable system. For a given optimization problem the number of dimensions may not be reduced, but unlike previous MOO studies involving PCA a significant reduction in the search space may still be obtained. This reduction in the search space demonstrates the benefit of projecting the input space using PCA before each grid is produced following the first standard grid search.

Once the grid has been formed, it must be projected back to the original frame of reference. This calculation consists of multiplying the grid by the inverse of the eigenvector matrix. The resulting grid for this problem is shown in Figure 17.6. This figure demonstrates a final resulting grid with a much smaller grid search region, and a smaller fraction of redundant points relative to the standard grid shown in Figure 17.4.

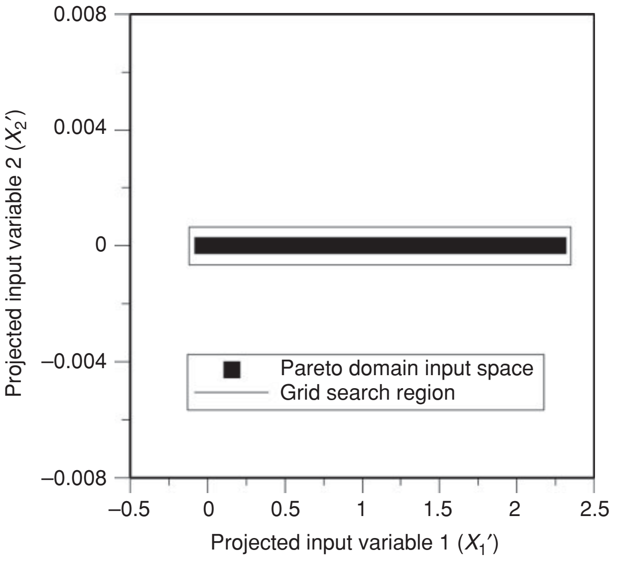

Figure 17.5 *Principal component projection of the input variable space for the example problem.*

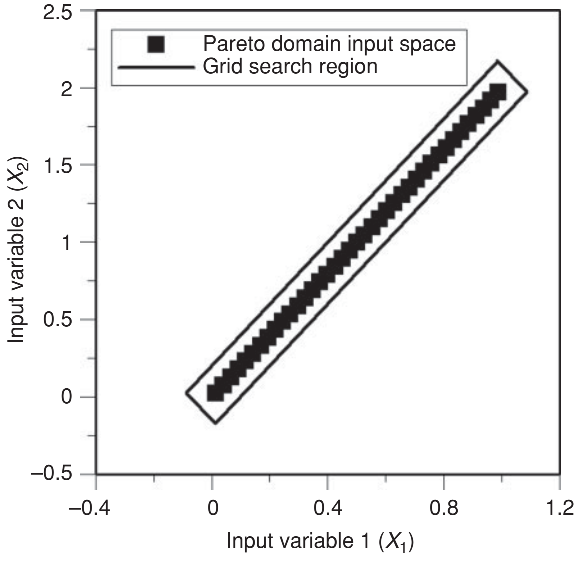

Figure 17.6 *Final grid generated using the PCA calculation procedure for the example problem.*

17.4.3.1 Principal Component Grid Algorithm

The general procedure for the algorithm proposed by Vandervoort [27], the principal component grid algorithm (PCGA), is shown below:

1. Initially a small number of iterations are conducted strictly using the procedure discussed in section 17.4.2.2 to generate a rough estimate of the Pareto domain. In this study, two iterations of the original GSA were used before PCA was implemented.
2. The next iteration begins by determining the eigenvalues and eigenvectors for the input variable space. Only the non-dominated points are used in this calculation as they represent the most current approximation of the Pareto domain.
3. The principal component projection is then calculated for the input-variable space, and a grid is formed with the projected data set. The grid is then projected back to the original frame of reference, and the corresponding objective functions are evaluated.

As was described in section 17.4.2.2, the number of divisions is increased at each iteration, and the calculation continues until all of the input variables have reached the predetermined minimum grid size.

In Vandervoort [27] it was demonstrated that for the wide range of optimization problems considered in the study, the PCGA led to a Pareto domain with higher accuracy than both the standard grid procedure [26] and NSGA-II [20], which is commonly used in MOO studies. The PCGA also led to a reduction in the required computation time relative to these two algorithms. As well as several theoretical and practical case studies, this high accuracy and increased efficiency were observed for MOO of a PI controller. The PCGA is therefore the MOO technique used in this study to approximate the Pareto domain.

17.5 Optimization Results

In order to determine ideal controller parameters, the Pareto domain for the generalized controller model discussed in section 17.3, subject to a unit step change in the set point, was approximated using the PCGA. The approximation was performed until the grid interval for both input variables was equal to 0.05. Multiple Pareto domains were approximated by varying the value of the relative dead time. Figure 17.7 shows the Pareto domains for varying values of the relative dead time, with Pareto domains corresponding to a relative dead-time of 0.2 to 2.0 shown. It should be noted that regardless of the specific values of the dead time, time constant and process gain, each value of the relative dead time led to one unique Pareto domain. For example, the Pareto domain for a time constant of 100 s, a dead time of 20 s and any value of the process gain was identical to the Pareto domain for a time constant of 5 s, a dead time of 1 s and any value of the process gain when relative variables as specified in section 17.3 were considered.

Figure 17.7 shows that the relative dead time has a significant effect on the input space of the generalized controller Pareto domain. Increasing the relative dead time leads to an increase in the slope of the Pareto domain input space and to changes in the range of the input variables defining the Pareto domain. Despite these variations, the input space of each Pareto domain is visually very narrow and follows closely a straight line. This is an important observation which suggests that if the relative controller gain is chosen, only a very narrow range in the relative integral time will lead to optimum controller performance, and vice versa. The strong correlation between the two controller parameters implies that

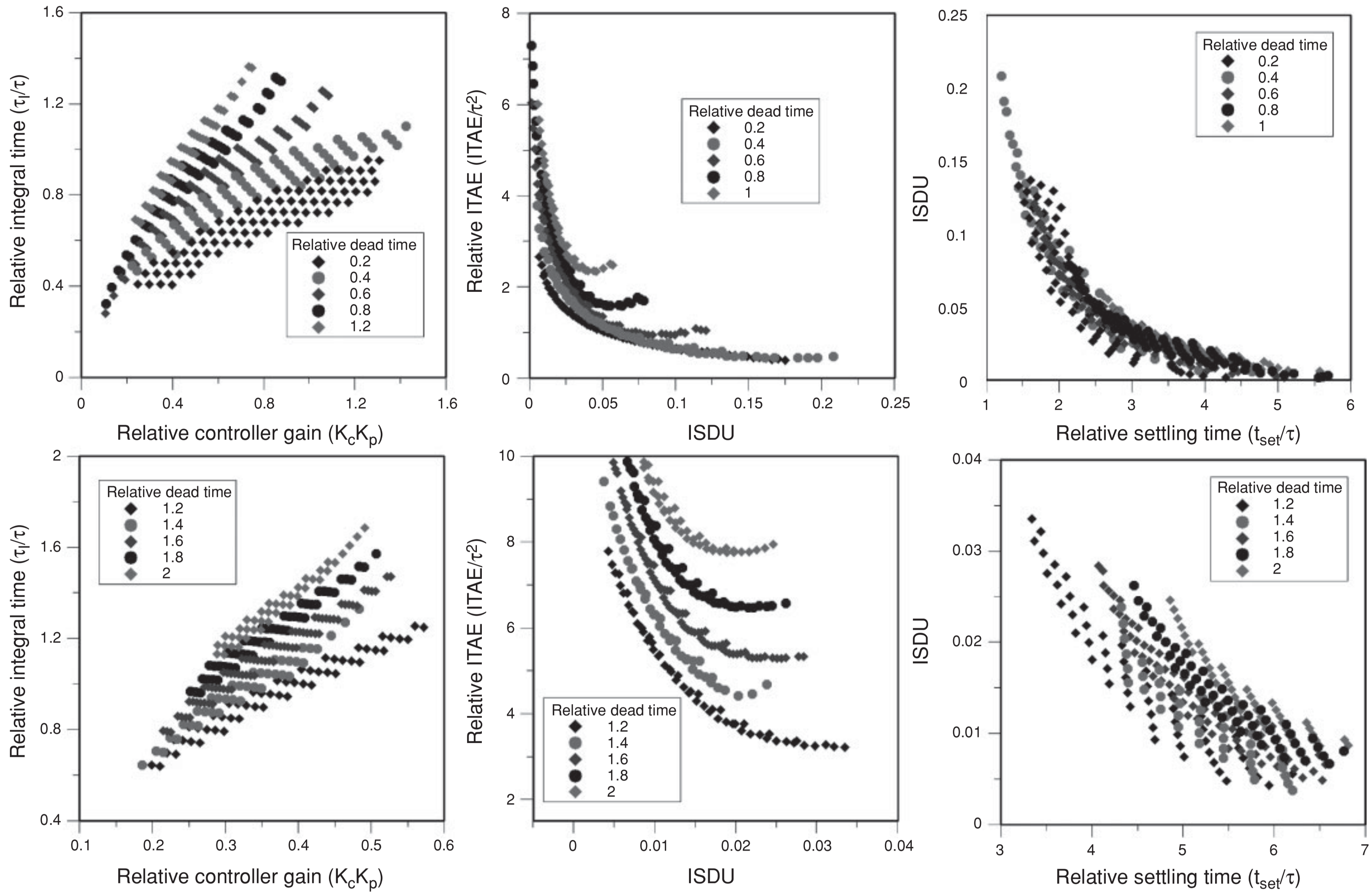

Figure 17.7 *The two decision variables and the three performance criteria for the Pareto domains of the generalized controller model for ten different relative dead times.*

Table 17.1 *Slope, intercept and input variable ranges for the generalized controller Pareto domain.*

θ/τ	Slope (η)	Intercept (b)	Optimum range of K_cK_p	Optimum range of τ_I/τ
0.2	0.325	0.317	0.247–1.30	0.407–0.952
0.4	0.479	0.447	0.223–1.42	0.466–1.10
0.6	0.841	0.351	0.106–1.09	0.280–1.26
0.8	1.13	0.335	0.109–0.876	0.321–1.31
1.0	1.30	0.382	0.236–0.748	0.679–0.748
1.2	1.57	0.373	0.199–0.572	0.638–1.25
1.4	1.87	0.294	0.186–0.483	0.645–1.33
1.6	2.09	0.370	0.214–0.527	0.790–1.47
1.8	2.30	0.392	0.252–0.507	0.961–1.57
2.0	2.55	0.351	0.289–0.491	1.13–1.68

when configuring a PI controller, only one of the two parameters needs to be specified, as the other can be obtained based on this strong correlation. Any selection of controller parameters outside of the narrow bands shown in Figure 17.7 will lead to a dominated point, which corresponds to deteriorated controller performance as all three performance criteria would be worse than at least one particular point within the Pareto domain.

Figure 17.7 also shows the effect of changing the relative dead time on the objective functions. It is apparent that both the relative ITAE and the relative settling time are highest at larger values of the relative dead time, whereas the ISDU is smallest at larger values of the relative dead time.

17.6 Controller Tuning

17.6.1 Method 1

Using the Pareto domain, optimal controller parameters can be determined for a given set of process parameters. The first tuning method involves setting one of the controller parameters, the gain or the integral time, such that the corresponding relative parameter falls within the Pareto domain. One Pareto domain was generated for each value of the relative dead time. The ranges of values for both relative controller parameters for each Pareto domain are presented in Table 17.1. The second controller parameter can then be specified graphically or using the linear relationship between the two relative controller parameters as defined by Equations (17.11)–(17.13). The values of both the slope and the intercept for each value of the relative dead time are given in Table 17.1. Equations (17.11)–(17.13) along with the information found in Table 17.1 allow for the calculation of the integral time for a prespecified value of the controller gain, and vice versa, to achieve optimum controller performance.

$$\frac{\tau_I}{\tau} = \left[\eta\left(K_c K_p\right) + b\right] \tag{17.11}$$

$$K_c K_p = \left(\frac{\tau_I}{\tau} - b\right)\frac{1}{\eta} \tag{17.12}$$

$$\eta = 1.256\frac{\theta}{\tau} + 0.06280 \tag{17.13}$$

17.6.2 Method 2

Method 1 ensures that an optimum set of controller parameters will be obtained, but the values of the objective functions are not specified a priori. An alternative method for configuring a PI controller is to first choose the desired values of the relative objective functions. For this tuning method, a well defined relationship must exist between each relative objective function and each relative controller parameter, to ensure that once a point is chosen in the objective space the corresponding input variables (controller tuning parameters) can be calculated. To confirm that this relationship does exist, the values of each objective performance criterion were plotted against the values of the two relative controller parameters. These plots are shown in Figures 17.8 and 17.9 for different values of the relative dead time. Figures 17.8 and 17.9 show that the relationship between each objective performance criterion and each relative controller parameter is well defined, and can therefore be used for controller tuning.

The first step in the second tuning method is choosing the desired values of the relative controller performance criteria for a given relative dead time. Using Figure 17.7, a desired point is first chosen in the output space. This point should be chosen to balance the tradeoff between each of the three objectives based on the preferences of the user. From the relative objective function values of the chosen point, the corresponding values of both the relative controller gain and the relative integral time can be determined from Figures 17.8 and 17.9. The optimum controller parameters are then calculated for the known process gain and time constant based on the relative input variables. Using this tuning method ensures that controller tuning parameters will be located within the Pareto domain and therefore optimal for the selected performance criteria. The two tuning methods use a different approach to achieve optimum controller performance.

17.7 Application of the Tuning Methods

17.7.1 First-Order Plus Dead Time System

The developed tuning method was first applied for a specific FOPDT system characterized by a process gain (K_p) of 1.5, a time constant (τ) of 5 and a dead time (θ) of 3. Results obtained were compared to several previously developed PI controller tuning methods.

The simulated FOPDT process corresponds to a relative dead time of 0.6. The Pareto domain for this specific value of the relative dead time was compared to the optimum controller parameters and objectives identified by several previously developed controller correlations. The controller correlations used for comparison along with the objective criteria used in each correlation are shown in Table 17.2. The results from the comparison are shown in Figure 17.10.

Figure 17.10 clearly demonstrates that the tuning method developed in this investigation provides a general framework for selecting PI controller parameters that would systematically be based optimally on the three performance indicators used to circumscribe the Pareto domain. The choice of the PI controller parameters depends on the performance specifications of control engineers. The PI control parameters of the two earlier control algorithms, Ziegler–Nichols and Cohen–Coon, lie outside the Pareto domain and all three performance criteria are therefore worse than a particular point located within the Pareto

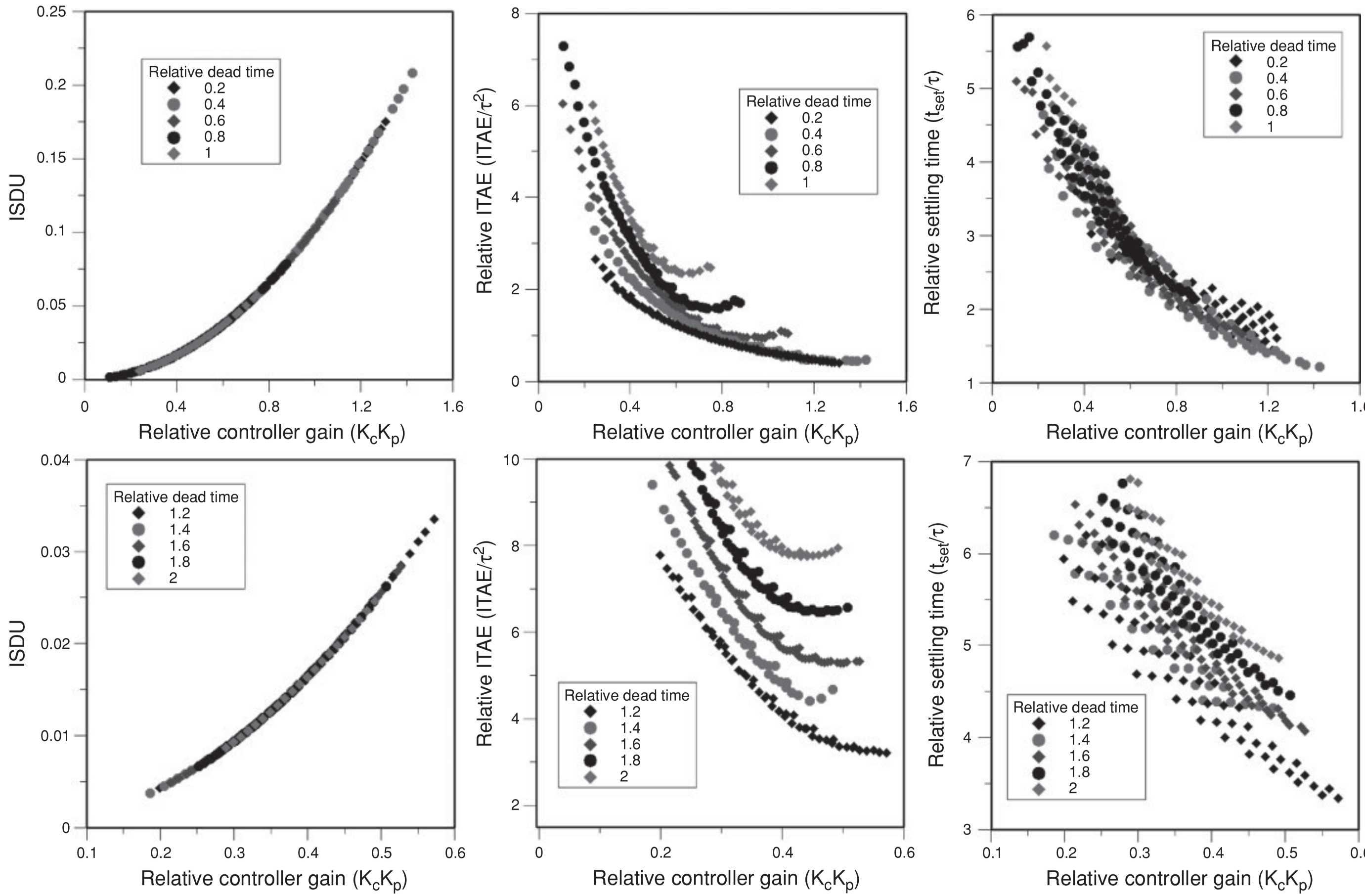

Figure 17.8 Graph of the three relative objective functions versus the relative controller gain for the generalized controller model for ten different relative dead times.

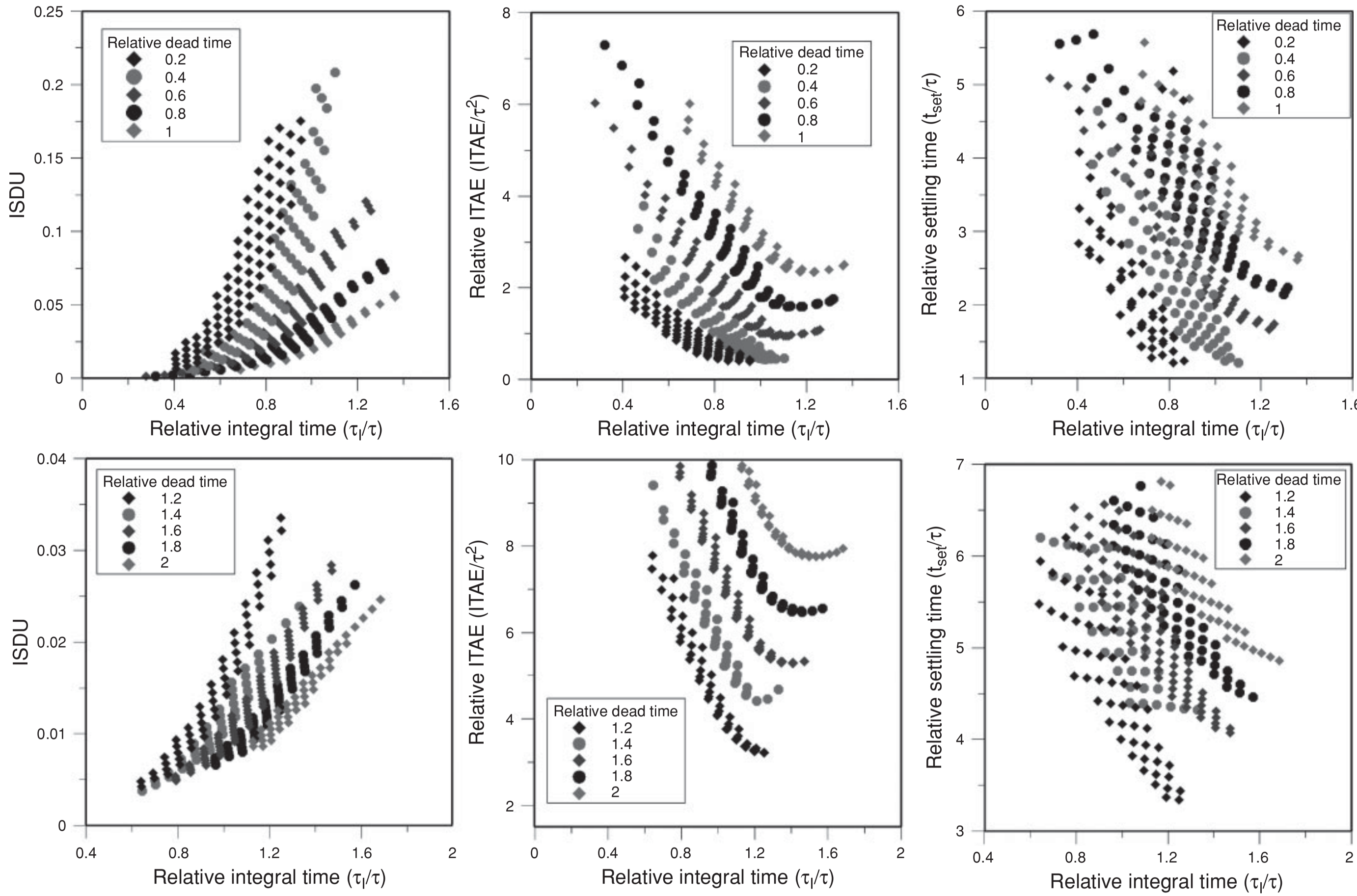

Figure 17.9 *Graph of the three relative objective functions versus the relative integral time for the generalized controller model for ten different relative dead times.*

Table 17.2 *Controller correlations used for comparison, and their objective criteria.*

Method	Objective criteria
Chien and Fruehauf [9]	IMC
Cohen and Coon [8]	One quarter decay ratio
Hägglund and Åström [29]	IAE
Skogestad [10]	IMC
Smith and Corripio [30]	ITAE
Tavakoli *et al.* [14]	MOO
Ziegler and Nichols [7]	One quarter decay ratio

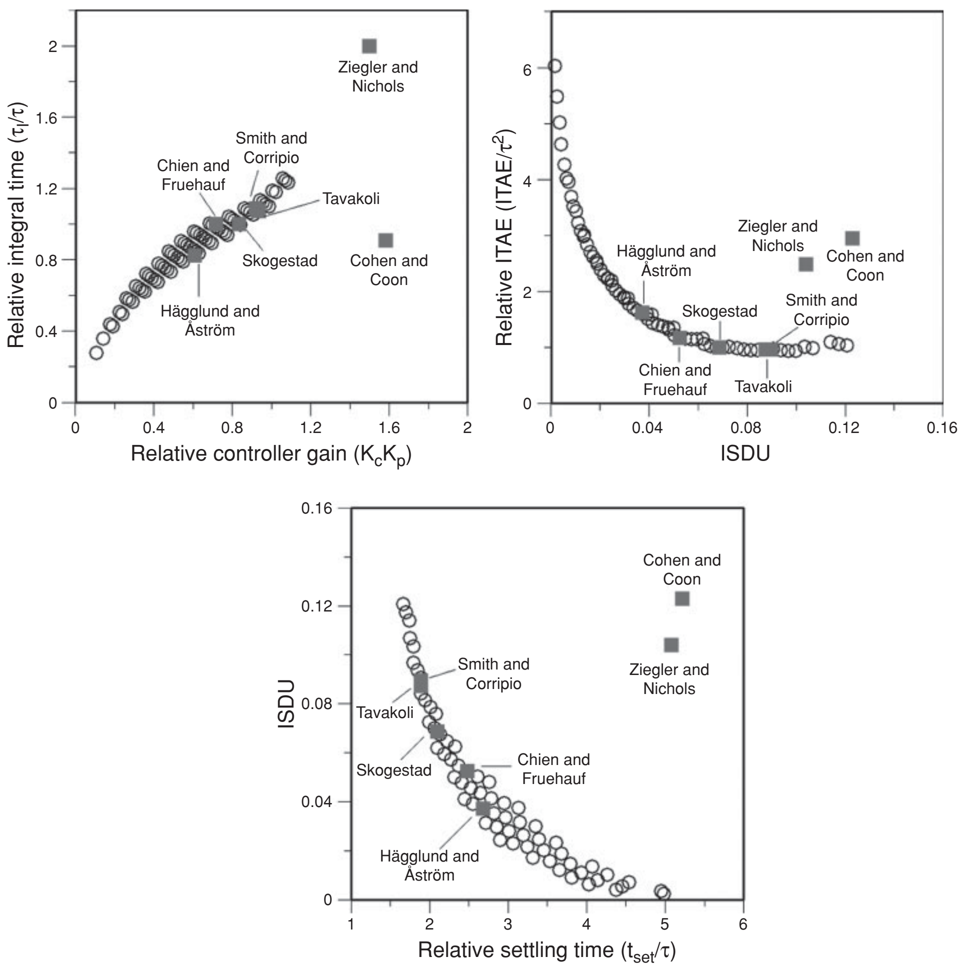

Figure 17.10 *Comparison of the PI controller Pareto domain with other PI controller tuning methods.*

domain. Of course, the control parameters from these earlier control algorithms were not obtained by minimizing one or more of the three criteria used in this investigation and it would have been surprising to find them located in the Pareto domain. The Ziegler–Nichols tuning method is a heuristic method of tuning a PID controller to achieve a response having a quarter decay ratio. The controller parameters for the Cohen–Coon tuning method are obtained from an open-loop process response curve. On the other hand, the other five tuning methods provide controller parameters that are located on the Pareto domain, as they minimize at least one of the objective criteria used in this investigation. These results clearly show that the Pareto domain offers an enhancement to previously developed controller correlations, because all possible optimum values for the PI controller parameters can be considered before choosing final controller parameters.

17.7.2 Fourth-Order Plus Dead Time System

To test the proposed tuning method for higher order systems, it was implemented for the control of an open-loop stable fourth-order system (with all real poles) with dead time. Although the tuning method was developed for a FOPDT system, a fourth-order system can be adequately approximated using a FOPDT model, and the tuning procedure developed in this study can be applied. The fourth-order system used in this study and the approximated FOPDT system are given in Equations (17.14) and (17.15), respectively. The FOPDT parameters were obtained by minimizing the squares of the differences between the FOPDT open-loop response and the fourth-order open loop response to a unit input step change. In this investigation, Solver in Excel was used to obtain the FOPDT parameters but any optimization software could be used. The responses of both systems are shown in Figure 17.11.

$$\frac{y(s)}{u(s)}=\frac{1.5e^{-s}}{(3.5s+1)(2.5s+1)(1.5s+1)(0.75s+1)} \tag{17.14}$$

$$\frac{y(s)}{u(s)}=\frac{1.51e^{-4.29s}}{(5.32s+1)} \tag{17.15}$$

Figure 17.11 clearly shows that the FOPDT system fits the fourth-order response very closely. The tuning method discussed in section 17.6.2 was next performed using the FOPDT process parameters. The relative dead time for this system is equal to 0.8. Two sets of PI controller parameters corresponding respectively to relative ITAE values of 3 and 5 were determined from Figures 17.8 and 17.9. To demonstrate more clearly how these controller parameters are determined, the middle upper panels of Figures 17.8 and 17.9 for a relative dead time of 0.8 have been reproduced in Figure 17.12. From these graphs, it is possible to estimate the relative controller gain (K_cK_p) and the relative integration time (τ_I/τ). For a relative ITAE value of 3, the corresponding values of K_c and τ_I were found to be 0.274 and 4.56 based on Figure 17.12 whereas, for a relative ITAE value of 5, these values are 0.156 and 3.16. It can be verified that these values are Pareto-optimal parameters as can be validated using in Figure 17.7. The closed-loop responses of both the fourth-order system and the corresponding FOPDT system were obtained using these controller parameters for a unit set point change. The closed-loop responses of both systems for relative ITAE values of 3 and 5 are presented in Figure 17.13(a) for the first 50 s and the

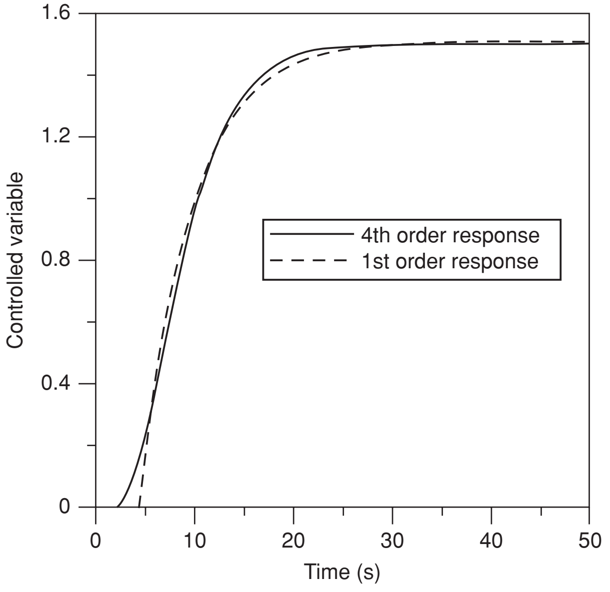

Figure 17.11 *Open loop response for the simulated fourth-order system and FOPDT system.*

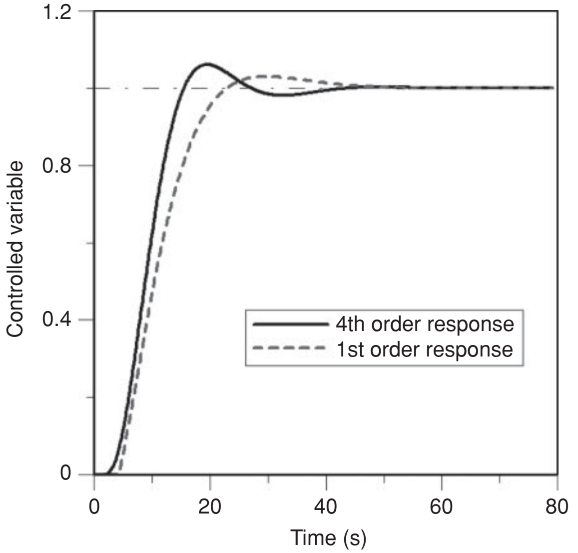

Figure 17.12 *Determination of the relative gain and relative integral time of a PI controller for a relative dead time of 0.8 and relative ITAE values of 3 and 5.*

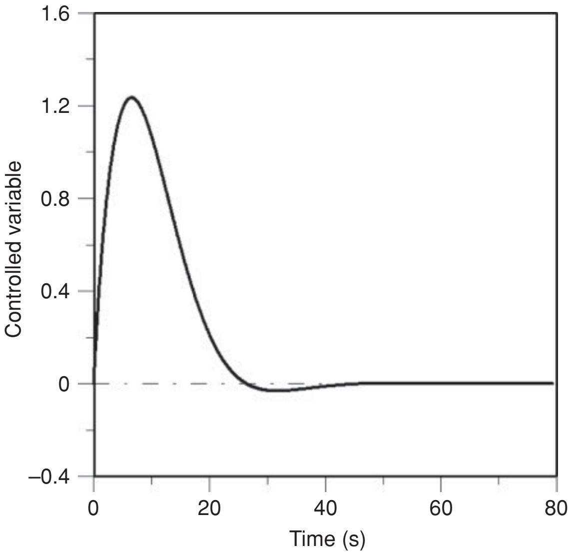

Figure 17.13 Closed-loop responses for the simulated fourth-order system (solid line) and FOPDT system (dotted line) for relative ITAE values of 3 and 5 for (a) a unit set point change and (b) a first-order disturbance.

corresponding objective function values are listed in Table 17.3. The dynamic responses were calculated for a total of 100 s using an integration step of 0.001 s.

Figure 17.13(a) and Table 17.3 clearly demonstrate that the controller configuration determined using the developed tuning method was effective in controlling the fourth-order system. Both systems showed very similar responses, with all three performance criteria very similar in value. These results show that the proposed tuning method can be applied to higher order systems as well as FOPDT systems. The obvious limitation is that the process can be represented adequately by a FOPDT and its open-loop behavior does not oscillate.

17.7.3 Application to a Process with a First-Order Disturbance

The proposed tuning method was also evaluated for the case of a disturbance. The controller parameters determined in section 17.7.2 were used to control the same fourth-order

Table 17.3 Performance criteria for the simulated fourth-order and FOPDT systems for a unit set point change corresponding to Figure 13.12(a).

Relative ITAE	System	ISDU	ITAE	Settling time
3	Fourth-order system	7.51×10^{-5}	78.62	19.95
	FOPDT system	7.51×10^{-5}	85.71	20.70
5	Fourth-order system	2.44×10^{-5}	132.13	25.28
	FOPDT system	2.44×10^{-5}	143.53	25.65

Table 17.4 Parameters of the first-order disturbance and the resulting objective functions.

Relative ITAE	System	ISDU	ITAE	Settling time
3	fourth-order system	8.80×10^{-8}	163.73	25.15
	FOPDT system	8.90×10^{-8}	175.83	25.83
5	fourth-order system	5.80×10^{-8}	256.84	30.24
	FOPDT system	5.84×10^{-8}	274.61	30.48

system subject to a unit set point change for a first-order disturbance. The first-order disturbance transfer function had a gain of 1.5 and a time constant of three time units as shown in Table 17.4. The closed-loop responses of the fourth-order and FOPDT systems are shown in Figure 17.13(b) and the associated performance criteria are presented in Table 17.4.

Figure 17.13(b) and Table 17.4 show that the developed tuning method performed well for the fourth-order process with a first-order disturbance. A similar settling time was realized for this simulation as for both responses shown in section 17.7.2. The ITAE was larger for the response to a disturbance, but the ISDU was reduced. All of the results from section 17.7.0 clearly show that the tuning method developed by approximating the Pareto domain leads to excellent controller performance, and is applicable to a wide variety of processes.

17.8 Conclusions

In this study, PI controller tuning methods were developed considering multiple objectives. The methods were developed by optimizing the ITAE, ISDU, and settling time for a FOPDT system. The Pareto domain identifying the region of optimal solutions was approximated using the PCGA due to its demonstrated high level of accuracy and efficiency. It was found that a strong correlation exists in the Pareto domain between the two controller input parameters, the relative controller gain, and the relative integral time. This implies that when configuring the controller, only one of the controller parameters needs to be specified, as the other is obtained via the strong correlation.

Using the controller optimization results, two methods were proposed for tuning the PI controller. The first tuning method allows for optimum controller performance to be obtained by initially specifying either one of the controller input parameters. The second tuning method involves first specifying the preferred relative objective function values from the Pareto domain, which correspond to specific values of the controller parameters. The developed controller tuning methods were compared to several previously developed controller correlations. It was found that all previously developed controller correlations showed equal or worse performance than that identified by the Pareto domain, but with the limitation of not allowing for enhanced understanding of the many optimal solutions and the tradeoff between each performance criterion. Finally, the tuning methods were applied to a fourth-order process and a process with a disturbance, and were shown to perform well for these two applications.

Acknowledgments

The authors would like to thank the Natural Science and Engineering Research Council (NSERC) for the financial assistance.

Nomenclature

Variable	Definition	Unit
b	Controller input space intercept	Dimensionless
Cov	Element in Covariance Matrix	Dimensionless
D	Determinant	Dimensionless
f	Objective function in optimization problem	Varies
ISDU	Integral of the squares of the differences in the manipulated variable	Varies
ITAE	Integral of the time-weighted absolute error	Varies
K_c	Controller gain	Units of u/y
K_p	Process gain	Units of y/u
M	Number of divisions in PCGA grid	Dimensionless
m	Number of objective criteria	Dimensionless
n	Number of input variables	Dimensionless
N	Number of points in Pareto domain	Dimensionless
P	Point in the Pareto domain	Dimensionless
T	Trace	Dimensionless
t_{set}	Settling time	s
u	Manipulated variable	Varies
x	Input function in optimization problem	Varies
y	Controlled variable	Varies

Greek Symbols

Δt	Time step	s
ε	Error	Varies
η	Controller input space slope	Dimensionless
θ	Dead time	s
τ	Time constant	s
τ_I	Integral time	s

Subscripts

Subscript	Definition
f	Final
t	Time
j	Point in the Pareto domain

Exercises

17.1. Starting with Equation (17.14), show that the FOPDT that best represents this fourth-order process is indeed given by Equation (17.15).

17.2. The transfer function of a chemical process can be accurately represented with the following equation in the Laplace domain:

$$\frac{y(s)}{u(s)} = \frac{0.82e^{-0.6s}}{(5s+1)(4s+1)(2s+1)} \tag{17.16}$$

Approximate this transfer function with a FOPDT and calculate the parameters K_c and τ_I of a PI controller using Method 2 for a relative ITAE of 2.5.

17.3. For the previous problem, using Method 1, determine the value of τ_I of a PI controller if the controller gain K_c is 0.43. What are the approximate values of the three performance criteria (ITAE, ISDU and settling time) associated with these controller parameters?

References

[1] G. Stephanopoulos, Chemical Process Control: An Introduction to Theory and Practice, Prentice-Hall, Inc. (1984).
[2] L. Desbourough and R. Miller, Increasing customer value of industrial control performance monitoring Honeywell's experience, in Proc. Sixth Int. Conference on Chem. Proc. Control, F. Rawlings, Ogunnaike, B. and Eaton, J., Eds., January, 2001, AIChE, pp. 172–192.
[3] S. Lee and J.H. Park, Performance improvement of PI controller with nonlinear error shaping function: IDA-PBC approach, Appl. Math. Comput., 215, 3620–3630 (2010).
[4] D.G. Koo, J. Lee, D.K. Lee, C. Han, L.S. Gyu, J.H. Jung, and M. Lee, A Tuning of the nonlinear PI controller and its experimental application, Korean J. Chem. Eng., 18, 451–455 (2001).
[5] K.K. Tan, T.H. Lee, and F.M. Leu, Predictive PI versus Smith control for dead-time compensation, ISA Trans., 40, 17–29 (2001).
[6] D.E. Seborg, T.F. Edgar, and D. Mellichamp, Process Dynamics and Control, John Wiley & Sons, Inc. (2004).
[7] J.G. Ziegler and N.B. Nichols, Optimum settings for automatic controllers, Trans. ASME., 64, 759–768 (1942).
[8] G.H. Cohen and G.A. Coon, Theoretical considerations of retarded control, Trans. ASME., 86, 827–834 (1953).
[9] I.-L. Chien and P.S. Fruehauf, Consider IMC tuning to improve controller performance, Chem. Eng. Progress, 86, 33–41 (1990).
[10] S. Skogestad. Simple analytic rules for model reduction and PID controller tuning, J. Proc. Cont., 13, 291–309 (2003).
[11] C. Fonseca, Multiobjective Genetic Algorithms with Application to Control Engineering Problems, PhD Thesis, University of Sheffield (1995).

[12] Y. Xue, L. Donghai, and F. Gao, Multi-objective optimization and selection for the PI control of ALSTOM gasifier problem, Cont. Eng. Prac., 18, 67–76 (2010).
[13] I.K. Kookos, PI controller tuning via multi-objective optimization, in Proc. Seventh Mediterranean Conference on Control and Automation, Z. Palmor, Ed., Israel, June 28–30, 1999, pp. 408-419.
[14] S. Tavakoli, I. Griffin, and P.J. Fleming, Multi-objective optimization approach to the PI tuning problem, in Proc. IEEE Congress on Evol. Comp., J. Xu, Ed., Singapore, September 25–27, 2007, pp. 3165–3171.
[15] J. Cvejn, Sub-optimal PID controller settings for FOPDT systems with long dead time, J. Proc. Cont., 19, 1486–1495 (2009).
[16] C.R. Madhuranthakam, A. Elkamel, and H. Budman, Optimal tuning of PID controllers for FOPTD, SOPTD and SOPTD with lead processes, Chem. Eng. and Process, 47, 251–264 (2008).
[17] A. Roy and K. Iqbal, PID controller tuning for the first-order-plus-dead-time process model via Hermite-Biehler theorem, ISA Trans. 44, 363–378 (2005).
[18] T. Marlin, Process Control: Designing Processes and Control Systems for Dynamic Performance, John Wiley & Sons, Inc., USA (2001).
[19] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Ltd (2001).
[20] K.A. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. on Evol. Comput. 6, 182–197 (2002).
[21] A. Agarwal and S.K. Gupta, Jumping gene adaptations of NSGA-II and their use in the multi-objective optimal design of shell and tube heat exchangers, Chem. Eng. Res. Des., 86 (2), 123–139 (2008).
[22] D. Mokeddem and A. Khellaf, Optimal Solutions of Multiproduct Batch Chemical Process Using Multiobjective Genetic Algorithm with Expert Decision System. J. Automated Methods and Management in Chemistry, 1–9 (2009).
[23] A. Tarafder, G.P. Rangaiah, and A.K. Ray, Multiobjective optimization of an industrial styrene monomer manufacturing process. Chem. Eng. Sci., 60 (2), 347–363 (2005).
[24] F. Logist, P.M. Van Erdeghem, and J.F. Van Impe, Efficient deterministic multiple objective optimal control of (bio)chemical processes, Chem. Eng. Sci. 64, 2527–2538 (2009).
[25] C. Li, X. Zhang, S. Zhang, and K. Suzuki, Environmentally conscious design of chemical processes and products: Multi-optimization method, Chem. Eng. Res. Des., 87 (2), 233–243 (2009).
[26] H. Halsall-Whitney and J. Thibault, Multi-objective optimization for chemical processes and controller design: Approximating and classifying the Pareto domain, Comput. Chem. Eng., 30, 1155–1168 (2006).
[27] A. Vandervoort., Multi-Objective Optimization Techniques and their Application to Complex Engineering Problems, M.A.Sc. Thesis, University of Ottawa (2010).
[28] I.T. Jolliffe, (1986) Principal Component Analysis. Springer-Verlag.
[29] T. Hägglund and K.J. Åström, Revisiting the Ziegler–Nichols tuning rules for PI control, Asian J. Control, 4, 364–380 (2002).
[30] C.A. Smith and A.B. Corripio. Principles of Automatic Control, John Wiley & Sons, Inc. (1997).