AUTOTUNING AND ROBUSTNESS OF MULTI-LOOP PID-CONTROLLED MULTIVARIABLE PROCESSES

by

Ergin Levent Deva

BS. in M.E., İstanbul Teknik Üniversitesi, 1997

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Mechanical Engineering

Boğaziçi University

2000

ACKNOWLEDGMENTS

I would like to express my gratitudes to my thesis supervisor Eşref Eşkinat, not only for guiding, support and encouragement, but also for the extreme patience and flexibility throughout the study.

I would like to thank my family for patience and support.

I would also like to thank Turkish Military Forces for rearrangement of the date of my military service, and for accelerating my study.

ABSTRACT

In this study, decentralized PID controllers which are commonly used in process industry are applied to multi-input multi-output systems. A short literature survey is done and basic characteristics of these systems with their control procedure options are presented. A MATLAB based software to automate the previously developed multivariable decentralized PID tuning is developed. Autotuning is based on the multivariable version of relay feedback method. Structured singular value μ, μ–optimal tuning, μ–interaction measure and robustness properties of closed loop multivariable systems are also investigated. Autotuning of multivariable systems is done, and then PID controller parameters are found using different tuning methods. Five systems are simulated with three different tuning methods. Their time responses are compared with each other. Robustness of the controllers is investigated.

Results show that Biggest Log Modulus Tuning method does not give satisfactory responses, requires a model and is sensitive to uncertainties. Shen and Yu's tuning method does not require a model, is easy to apply to process systems, gives reasonable time responses and is robust against small uncertainties and perturbations. μ–optimal tuning gives best results, but requires extensive computation.

ÖZET

Bu çalışmada, proses endüstrisinde yaygın olarak kullanılan merkezden bağımsız PID kontrol organları, çok girişli çok çıkışlı sistemlere uygulandı. Kısa bir kaynak araştırması yapıldı ve bu sistemlerin ana karakteristikleri, kontrol prosedürleriyleriyle birlikte sunuldu. Önceden çok değişkenli diagonal PID kontrol organı ayarları için geliştirilmiş teknik, MATLAB kullanılarak otomatikleştirildi. Otomatik ayarlama, röle geri beslemesinin çok değişkenli tipine dayandırıldı. Kapalı çevrim çok değişkenli sistemlerin yapısal tekil değeri μ, μ-optimum ayarlaması, μ-girişim ölçümü ve dayanıklılık özellikleri araştırıldı. Çok değişkenli sistemlerin otomatik ayarlaması yapıldı, sonrasında çeşitli ayarlama metotlarına göre PID kontrol organı parametreleri bulundu. Beş değişik sistem, üç değişik kontrol yöntemiyle simüle edildi. Zaman cevapları birbirleriyle karşılaştırıldı. Kontrol organlarının dayanıklılığı araştırıldı.

Sonuçlar, En Büyük Logaritma Modülü Ayarı yönteminin tatmin etmeyen cevaplar verdiğini, sistemin bir modeline ihtiyaç duyduğunu ve belirsizliklere karşı hassas olduğunu gösterdi. Shen ve Yu'nun ayar metodu bir modele ihtiyaç duymadı, proses sistemlerine kolayca uygulandı, uygun zaman cevapları verdi ve ufak belirsizliklerde ve bozucu unsurlarda dayanıklı olduğunu gösterdi. μ-optimum ayarlarlaması en iyi sonuçları verdi, ama kapsamlı hesaplamalara ihtiyaç duydu.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a	amplitude of oscillation
D	time delay
E	error vector
e_i	error value on the *i*th loop
F	detuning factor
$G(s)$	transfer function (for SISO), transfer matrix (for MIMO)
$g_{ij,CL}$	(i,j) component of closed loop transfer matrix
GM	gain margin
h	amount of increase and decrease in input
I	identity matrix
$K(s)$	transfer function of the controller(for SISO), controller transfer matrix (for MIMO)
K_{Ci}	controller gain of the *i*th controller
K_{cr}	critical gain
K_i	controller on the *i*th loop
K_u	ultimate gain
$L(s)$	loop transfer function
L_c	log modulus
M	transfer function form the output to the input of perturbations
M_S	maximum peak of the sensitivity function
M_T	maximum peak of the complementary sensitivity function
n	Number of inputs and outputs of a MIMO system
P	generalized plant in robustness analysis
PM	phase margin
P_u	ultimate period
R	reference vector
r_i	*i*th reference value
S	sensitivity function

s_i	relative loop speed
T	complementary sensitivity function
T_d	derivative time
T_I	integral time
T_p	time constant of the plant
U	input vector (otherwise stated)
u_i	*i*th input value (otherwise stated)
W_i	normalization weight matrix
W_p	performance weight matrix
Y	output vector
y_i	*i*th output value
Δ	uncertainty matrix
γ	condition number
Λ	relative gain array
μ	structured singular value
σ_{max}	maximum singular value
σ_{min}	minimum singular value
ω	frequency
ω_{180}	phase crossover frequency
ω_c	gain crossover frequency
ω_{cr}	critical frequency
ω_u	ultimate frequency

1. INTRODUCTION

Proportional-Integral-Derivative (PID) control, especially decentralized PID control is one of the most common control structures for interacting multiple-input multiple-output (MIMO) plants in process and chemical industries. The main reason for this is its relatively simple structure, which is easy to understand and to implement. Even though adaptive-, intelligent-, self-tuning-, fuzzy-logic controllers and others can give better control options, decentralized PID control gives satisfactory performance in many control loops.

Decentralized PID control, applied to plants where the number of inputs is equal to the number of outputs, has some advantages against full matrix PID control. In decentralized PID control, the number of tuning parameters is $3n$, where n is the number of inputs and outputs, while in full matrix PID control there are $3n^2$ parameters. Considering moderately sized systems, this is a significant reduction. Also, in case of sensor or actuator failure, only one loop is affected by the failure, which means that the system can be stabilized relatively easily in manual mode.

Although decentralized PID control is popular, the number of applicable manual and automatic tuning methods are limited. Even for single-input single-output (SISO) systems, the tuning of a PID controller is not an easy task. If tuning is done by trial and error, the procedure becomes time-consuming. For MIMO systems, the situation becomes more serious.

According to a survey of the state of process control systems, done by the Japan Electric Measuring Instrument Manufacturers Association in 1989 (Yamamoto, 1991), more than 90% of controller loops are of the PID type. A previous work on a paper mill in Canada (Bialkowski, 1993) reports that the paper mill has more than 2000 control loops and that 97% use PI control. Another work (Ender, 1993) claims that only 20% of control loops work well. Of those that do not perform well, 30% were because of poor tuning. Ender's work claims that 30% of the installed process controllers operate in manual mode and 20% of the loops use "factory tuning", i.e., default parameters set by the manufacturer. We face the fact that PID controllers are very widely used but poorly tuned.

Any process like a manufacturing process or a refining process cannot operate successfully with a single control loop. Almost every process requires at least two control loops to maintain the desired production rate and product quality. A large number of real multiloop control systems consist of SISO controllers acting in a multiloop fashion.

Industrial processes, which mostly contain MIMO systems and which are poorly tuned, need to be tuned correctly in order to increase the efficiency of the process. This tuning procedure must be done in the easiest and most effective way.

To have a better idea of what kind of developments can be done, the implementation results of a previous work (Musch and Steiner, 1995) are considered. The advantages of autotuned over manually operated PID controllers for a distillation column are:

- more uniform product quality,
- better average product quality in column bottom,
- no need to install additional equipment like another distillation column,
- cost savings potential of $250000 annually.

1.1. The Problem

The aim of this thesis is to achieve the results as discussed above. Furthermore, an easy and effective way to obtain these results must also be found. We investigate various systems and tuning methods to decide which way is the best to reach our goals.

Although better control schemes than PID control can be applied to many systems, PID control structure is well-known, easy to apply, easily upgradable to one-button-push type of tuning with very little effort.

We face the problems of PID control in this study: The simple structure of this control scheme gives the controller limited behavior, limited bandwidth, limited input-output range. Control loops with large dead time or other kinds of complex dynamics such as highly nonlinear properties are hard to control with PID controllers.

The advantage of PID controllers over other advanced controllers is to be independent of the control of a computer. If the operator wants to have the possibility to retune the system, he does not have to possess computer knowledge, he just has to know how to push a button!

So, in this thesis, the methods of "Autotuning of Multivariable Systems" are investigated, which seem to be useful for saving energy and money, reducing the waste of time.

1.2. Approach to the Problem and Arrangement of the Thesis

The main characteristics of SISO systems should be investigated to interpret the simulation results which are obteained in this work. To make detailed comments about the MIMO simulation results, SISO time-domain and frequency-domain performance characteristics are given.

In Chapter 2, equations shared by SISO and MIMO systems are given. Many SISO equations can be applied to MIMO systems by changing scalars to vectors and matrices.

The next step is to look into the properties of MIMO systems. These properties are useful when performance and robustness are of concern. Robustness analyses are based on structured singular value (μ) analysis.

Moreover, because the subject of the thesis is "Autotuning", previous work on controller tuning and autotuning processes should be considered. The concepts, theories, procedures, advantages and disadvantages on "Automatic Controller Tuning" must be known.

In Chapter 3, robustness properties are given, the developed automatic tuning program (written in MATLAB) is presented, which builds a basis for the simulation work and produces simulation results. These results are checked by another program which tests robustness properties through the structured singular value analysis.

In Chapter 4, simulation results are given and comparison is made between various tuning methods. Results are given in tables and figures.

In Chapter 5, a summary of the study, its results, conclusions and the suggestions for futher work for the subject "automatic PID controller tuning" are given.

2. INTRODUCTION TO MULTIVARIABLE PROCESS SYSTEMS

Assume a combination of n SISO systems working in harmony. If such a system consists of n inputs and n outputs, it can be said that there exists an nxn multivariable (MV) system. But this is not exactly true. A MV system does not only consist of n SISO systems, all these SISO systems have interactions with each other more or less in a MV system structure. Every single input has effect on either corresponding or non-corresponding output.

Such systems exist in aerospace and process industries. Consider a distillation column. Assume that crude oil is fed from the bottom of the column and we want to obtain gasoline from a tray and fuel oil from another tray. In the real configuration of a distillation column, the inputs and outputs are determined by the structure and physical properties of the column (see Fig. 2.1). The inputs are reflux L, boilup V, distillate D, bottom flow B, overhead vapor V_T. The outputs are top composition y_D, bottom composition x_B, condenser holdup M_D, reboiler holdup M_B and pressure p. So, we have a 5x5 MV system. However, this system can be simplified to a 2x2 MV system by using some level (LC) and pressure controllers (PC).

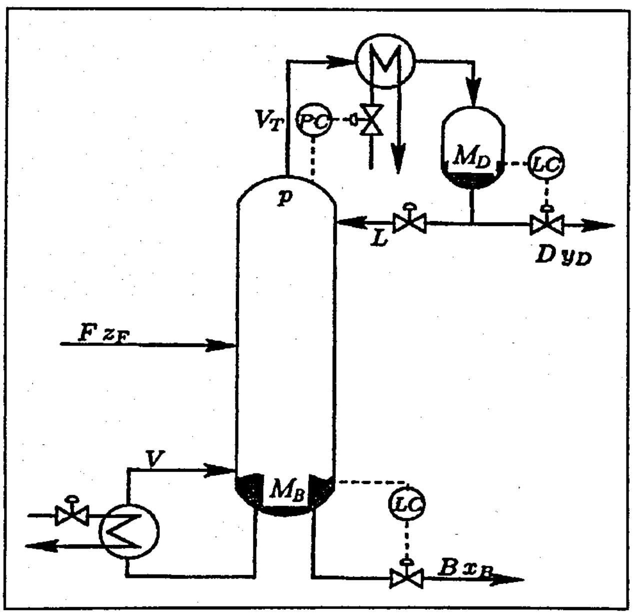

FIGURE 2.1. A distillation column, an example for a MIMO system (Skogestad, 1996)

In order to control such system, MV controllers are used (Figure 2.2). Because of the practical ease, PID controllers are chosen. A full matrix PID controller can be used which has n^2 single PID components. A simpler solution exists if that system is controlled with a decentralized controller (Figure 2.2.b), which controls only the corresponding outputs via the errors on the corresponding loops and which does not impose any interaction between the loops.

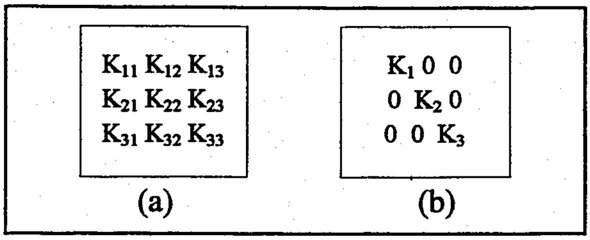

FIGURE 2.2. Structure of (a) a full MV controller, (b) decentralized MV controller

The basic properties of SISO systems are given and extended to MIMO systems, then the details of the subjects related with our work are briefly investigated.

2.1. Single Loop Feedback Design

2.1.1. Time Domain Performance and its Interpretation

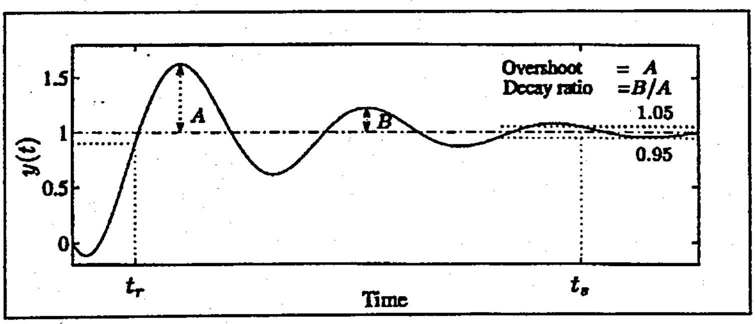

FIGURE 2.3. Step response analysis

In the step response analysis (Figure 2.3), time response for a given step input is mostly considered when evaluating the performance of a closed loop control system. Some characteristics are:

- Rise time (t_r): The time it takes for the output to first reach 90% of its final value. t_r is required to be small.
- Settling time (t_s): The time after which the output remains within $\pm 5\%$ of its final value, which is required to be small.
- Overshoot: Peak value divided by the final value, which should be 1.2 or less.
- Decay ratio: The ratio of second and first peaks should be 0.3 or less.
- Steady-state offset: The difference between the final value and the desired final value, which is required to be small.

2.1.2. Frequency Domain Performance and its Limitations

The frequency response of the loop transfer function; *L(jω)*, of various closed loop transfer functions, may also be used to characterize closed loop performance.

2.1.2.1. Gain and Phase Margins, Gain and Phase Crossover Frequencies. Let *L(s)* denote the loop transfer function of a system which is closed-loop stable under negative feedback. We will use the following information by tuning independent loops in the MIMO system theory. A typical Bode plot and a typical Nyquist plot of *L(jω)* illustrate the gain margin (GM) and phase margin (PM) in Figure 2.4. and Figure 2.5. respectively.

The gain margin is defined as

$$GM = 1/\left|L(j\omega_{180})\right| \tag{2.1}$$

where the phase crossover frequency ω_{180} is where the Nyquist curve of *L(jω)* crosses the negative real axis between -1 and 0. That is

$$\angle L(j\omega_{180}) = -180° \tag{2.2}$$

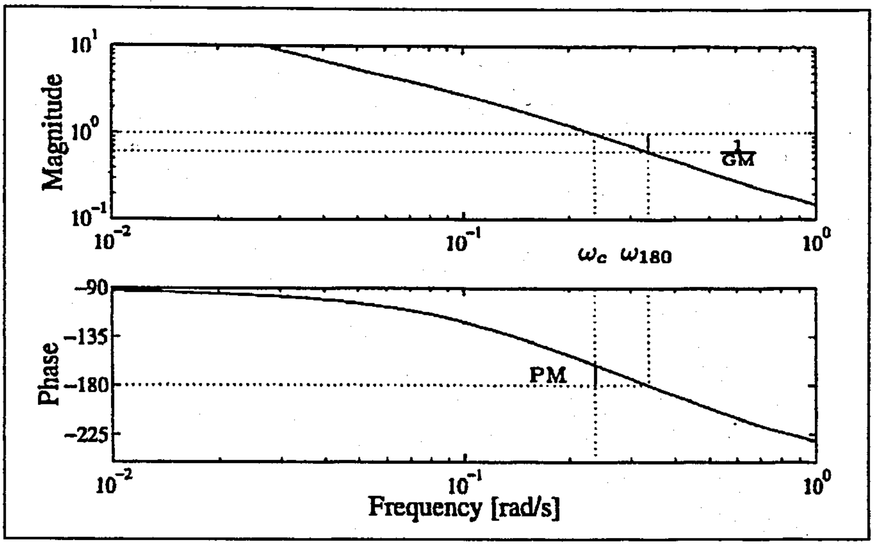

FIGURE 2.4. A typical Bode plot of a SISO system

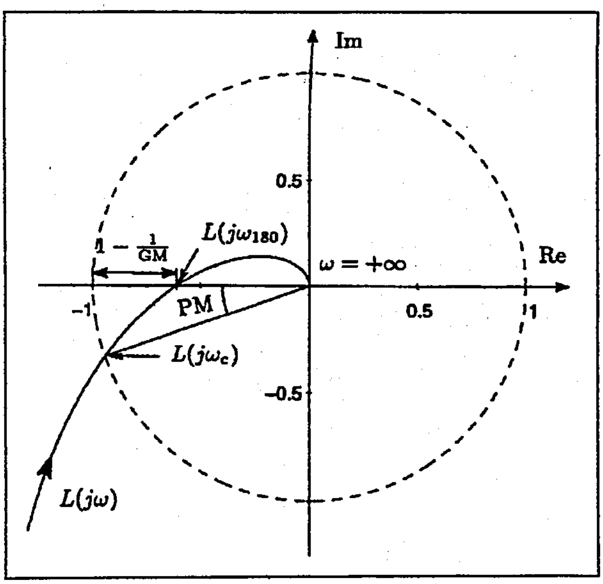

FIGURE 2.5. A typical Nyquist plot of a SISO system

If there is more than one crossing the largest value of $|L(j\omega)|$ is taken. The GM is the factor by which the loop gain $|L(j\omega)|$ may be increased before the closed-loop system becomes unstable. The GM is thus a safeguard against steady-state gain uncertainty and the required value is GM>2 (6dB). If the Nyquist plot (Fig. 2.5) of L crosses the real axis between -∞ and -1 then the gain reduction margin can be similarly defined from the smallest value of $|L(j\omega_{180})|$ of such crossings.

The phase margin is defined as

$$PM = \angle L(j\omega_c) + 180° \tag{2.3}$$

where the gain crossover frequency ω_c is where $|L(j\omega)|$ first crosses 1 form above, that is

$$|L(j\omega_c)|=1 \tag{2.4}$$

The phase margin indicates how much phase lag can be added to $L(s)$ at frequency ω_c before the phase at this frequency becomes -180° which corresponds to closed-loop instability. Typically, a PM of 30° or more is required. The PM is a direct safeguard against time delay uncertainty: The system becomes unstable if we add time delay of

$$D_{max}=PM/\omega_c \qquad \text{(PM and } \omega_c \text{ in rad/s)} \tag{2.5}$$

Decreasing the value of ω_c (lowering the closed-loop bandwidth, resulting in a slower response) the system can tolerate larger time delay errors.

2.1.2.2. Maximum Peak Criteria. The maximum peaks of the sensitivity M_S and complementary sensitivity functions M_T are defined as:

$$M_S \equiv \max_{\omega}|S(j\omega)|;\ M_T \equiv \max_{\omega}|T(j\omega)| \tag{2.6}$$

Typically, it is required M_S is less than about 2 (6 dB) and M_T is less than about 1.25 (2dB) for SISO systems. A large value of M_S or M_T (larger than about 4) indicates poor performance as well as poor robustness.

2.2. Common Equations for SISO and MIMO Systems

For the feedback system shown in Figure 2.6., we define $L=GK$ to be the loop transfer function as seen when breaking the loop at the outputs of the plant.

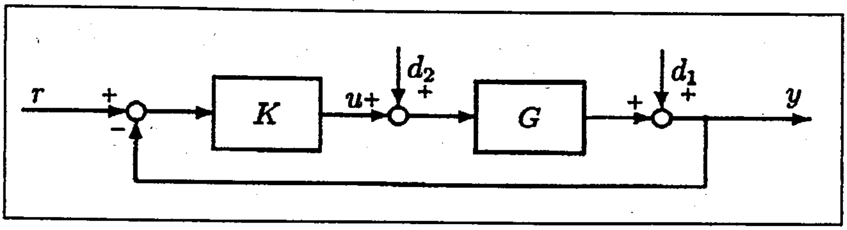

FIGURE 2.6. Conventional negative feedback control system

Thus, for the case where the loop consists of a plant G and a feedback controller K it can be written that

$$L=GK \tag{2.7}$$

and $L=GK$ is a square matrix (Skogestad, 1996).

The sensitivity and complementary sensitivity functions can be defined as:

$$S=(I+L)^{-1} \tag{2.8}$$

$$T=I-S=L(I+L)^{-1} \tag{2.9}$$

In Figure 2.6, T is the transfer function from r to y and S is the transfer function from d_1 to y.

Closed loop response of the system is:

$$y=L(I+L)^{-1}r+G_d(I+L)^{-1}d-L(I+L)^{-1}n \tag{2.10}$$

where G_d is the transfer function of the disturbance, d is the disturbance signal, n is the measurement noise, r is the reference value and y is the output.

S and T are sometimes called the output sensitivity and output complementary sensitivity, respectively, and to make this explicit one may use the notation $L_O=L$, $S_O=S$,

$T_O=T$. This is to distinguish them from the corresponding transfer functions evaluated at the input to the plant.

Define L_I to be the loop transfer function as seen when breaking the loop at the input to the plant with negative feedback assumed. In Figure 2.6.

$$L_I=KG \tag{2.11}$$

The input sensitivity and input complementary sensitivity functions can be defined as (Skogestad, 1996):

$$S_I=(I+L_I)^{-1} \tag{2.12}$$

$$T_I=I-S_I=L_I(I+L_I)^{-1} \tag{2.13}$$

The following relationships are useful (Skogestad, 1996):

$$(I+L_I)^{-1}+L_I(I+L_I)^{-1}=S+T=I \tag{2.14}$$

$$G(I+KG)^{-1}=(I+GK)^{-1}G \tag{2.15}$$

$$GK(I+GK)^{-1}=G(I+KG)^{-1}K=(I+GK)^{-1}GK \tag{2.16}$$

$$T=L(I+L)^{-1}=(I+(L)^{-1})^{-1} \tag{2.17}$$

2.3. Performance Analysis of MV Systems in Frequency Domain

The transfer function $G(s)$ is a function of the Laplace variable s and can be used to represent a dynamic system. However, if we fix $s=s_0$ then we may view $G(s_0)$ simply as a complex matrix, which can be analyzed using standard tools in matrix algebra. In particular, the choice of $s_0=j\omega$ is of interest since $G(j\omega)$ represents the amplitude ratio between the output and the input signals with respect to the frequency ω.

2.3.1. Singular Value Decomposition

Definition: A matrix $U \in C^{n \times n}$ is unitary if

$$U^H = U^{-1} \tag{2.18}$$

where U^H represents the complex conjugate transpose of U.

All the eigenvalues of a unitary matrix are equal to ±1, and all of its singular values are equal to 1.

Theorem: Given $A \in C^{l \times m}$, let $k=rank(A)$. Then, there exist unitary matrices $U \in C^{l \times l}$, $V \in C^{m \times m}$ and a diagonal matrix $\Sigma_1 \in R^{k \times k}$, with positive entries such that

$$A = U \Sigma V^H \tag{2.19}$$

where

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.20}$$

So,

$$\Sigma = \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix};\ l > m \quad \text{and} \quad \Sigma = \begin{bmatrix} \Sigma_1 & 0 \end{bmatrix};\ l \le m \tag{2.21}$$

when $rank(A)=min(l,m)$ (Morari and Zafiriou, 1989).

The matrix Σ_1 becomes:

$$\Sigma_1 = diag\{\sigma_1, \sigma_2, ..., \sigma_k\} \; ; k=min(l,m)$$

and

$$\sigma_{max} = \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_k = \sigma_{min}$$

SVD can be applied to the frequency response of a MIMO system $G(s)$ with m inputs and l outputs, and the numerical and graphical results can be interpreted physically.

Consider a fixed frequency ω where $G(j\omega)$ is a constant lxm complex matrix and denote $G(j\omega)$ by G for simplicity. Any matrix G may be decomposed into its singular value decomposition:

$$G=U\Sigma V^H \tag{2.22}$$

where Σ is an lxm matrix with $k=min\{l,m\}$ non-negative singular values σ_i, arranged in descending order along its main diagonal other entries being zero. Singular values are the positive square roots of the eigenvalues G^HG, where G^H is the complex conjugate transpose of G:

$$\sigma_i(G) = \sqrt{\lambda_i(G^H G)} \tag{2.23}$$

U is an lxl unitary matrix of output singular vectors, u_i, V is an mxm unitary matrix of input singular vectors, v_i.

The singular values are sometimes called the principal values or principal gains, and the related directions are called principal directions. In general, the singular values must be computed numerically.

2.3.1.1. Input and Output Directions. The column vectors of U, denoted u_i, represent the output directions of the plant, and u^H is the conjugate transpose of u. They are orthogonal and of unit length (orthonormal), that is

$$\|u_i\|_2 = \sqrt{|u_{i1}|^2 + |u_{i2}|^2 + \ldots + |u_{il}|^2} = 1 \tag{2.24}$$

$$u_i^H u_i = 1, \quad u_i^H u_j = 0, \quad i \neq j$$

Likewise, the column vectors of V, denoted v_i, are orthogonal and of unit length, and represent the input directions. These input and output directions are related through singular values. Noting that since V is unitary we have $V^H V=I$, so $G=U\Sigma V^H$ may be written as $GV=U\Sigma$, which for column i becomes

$$Gv_i = \sigma_i u_i \tag{2.25}$$

where v_i and u_i are vectors, whereas σ_i is scalar. That is if we consider an input in the direction v_i, that the output is in the direction u_i. Furthermore, since $\|v_i\|_2=1$ and $\|u_i\|_2=1$ we see that the i'th singular value σ_i gives directly the gain of the matrix G in this direction. In other words

$$\sigma_i(G) = \|Gv_i\|_2 = \frac{\|Gv_i\|_2}{\|v_i\|_2} \tag{2.26}$$

There are some advantages of the SVD over the eigenvalue decomposition for analyzing gains and directionality of MIMO plants (Skogestad, 1996). These are:

- The singular values give better information about the gains of a plant.
- The plant directions obtained from the SVD are orthogonal.
- The SVD also applies directly to non-square plants.

2.3.1.2. Maximum and Minimum Singular Values. The largest gain for any input direction is equal to the maximum singular value (Skogestad, 1996) (Let d be the input):

$$\sigma_{\max}(G)=\sigma_1(G)=\max_{d\neq 0}\frac{\|Gd\|_2}{\|d\|_2}=\frac{\|Gv_1\|_2}{\|v_1\|_2} \tag{2.27}$$

and the smallest gain for any input direction is equal to the minimum singular value

$$\sigma_{\min}(G)=\sigma_k(G)=\min_{d\neq 0}\frac{\|Gd\|_2}{\|d\|_2}=\frac{\|Gv_k\|_2}{\|v_k\|_2} \tag{2.28}$$

where $k=min\{l,m\}$. Thus, for any vector d we have:

$$\sigma_{\min}(G)\leq\frac{\|Gd\|_2}{\|d\|_2}\leq\sigma_{\max}(G) \tag{2.29}$$

Define $u_1=\overline{u}$, $v_1=\overline{v}$, $u_k=\underline{u}$, and $v_k=\underline{v}$. Then,

$$G\overline{v}=\sigma_{\max}\overline{u},\ G\underline{v}=\sigma_{\min}\underline{u} \tag{2.30}$$

The vector $\overline{v}$ corresponds to the input direction with largest amplification, and $\overline{u}$ is the corresponding output direction in which the inputs are most effective. The directions involving $\overline{v}$ and $\overline{u}$ are sometimes referred to as the "strongest", "high-gain" or "most important" directions. The next most important directions are associated with v_2 and u_2 and so on until the "least important", "weak" or "low-gain" directions are associated with $\underline{v}$ and $\underline{u}$.

2.3.1.3. Use of the Maximum Singular Value of the Plant. The minimum singular value of the plant, $\sigma_{\min}(G(j\omega))$, evaluated as a function of frequency, is a useful measure for evaluating the feasibility of achieving acceptable control. If the inputs and outputs are properly scaled, then with a manipulated input of unit magnitude (measured by the 2-norm), we can achieve an output magnitude of at least $\sigma_{\min}(G)$ in any output direction. $\sigma_{\min}(G)$ should be as large as possible and must be at least 1 to avoid input saturation over low frequencies (Skogestad, 1996).

2.3.2. Condition Number

The condition number of a matrix G is the ratio between its maximum and minimum singular values (assuming $\sigma_{max}\neq 0$),

$$\gamma(G) = \sigma_{\max}(G) / \sigma_{\min}(G) \tag{2.31}$$

A matrix with a large condition number (i.e., $\gamma(G)>10$) is said to be ill-conditioned. For a non-singular (square) matrix $\sigma_{max}(G)=1/\sigma_{min}(G^{-1})$, so $\gamma(G)= \sigma_{max}(G)\sigma_{max}(G^{-1})$.

The condition number γ has been used as an input-output controllability measure, and in particular it has been postulated that a large condition number indicates sensitivity to uncertainty (Skogestad, 1996). This is not true in general, but the reverse holds; if the condition number is small, then the effects of uncertainty are not likely to be serious.

If the $\gamma(G)$ is large, then this may indicate the following control problems:

- A large condition number may be caused by a small values of $\sigma_{min}(G)$, which is generally undesirable.
- A large condition number may mean that the plant has a large minimized condition number, or equivalently, it has large RGA-elements which indicate fundamental control problems.
- A large condition number does imply that the system is sensitive to unstructured input uncertainty but this kind of uncertainty often does not occur in practice. Therefore, 'a plant with a large condition number is sensitive to uncertainty' cannot be concluded generally.

2.3.3. Relative Gain Array (RGA)

The relative gain array (RGA) of a non-singular square matrix G is a square matrix defined as

$$\mathrm{RGA}(G)=\Lambda(G)=G\otimes(G^{-1})^T \tag{2.32}$$

where $\otimes$ denotes element-by-element multiplication.

In most cases, the value of the RGA at frequencies close to crossover is very important.

The RGA has many useful control properties and physical interpretations (Skogestad, 1996):

1. The RGA is a good indicator of sensitivity to uncertainty:

(a) Plants with large RGA elements around the crossover frequency are fundamentally difficult to control because of sensitivity to input uncertainty. In particular, decouplers or other inverse-based controllers should not be used for plants with large RGA elements.

(b) Large RGA elements imply sensitivity to element-by-element uncertainty. However, this is practically not possible because of physical couplings between the transfer functions elements. Therefore, diagonal input uncertainty is usually of more concern for plants with large RGA elements.

2. If the sign of an RGA element changes from $\omega=0$ to $\omega=\infty$, then there is a RHP-zero in G or in some subsystem of G (Hovd and Skogestad, 1992).

3. Extra inputs in non-square plants: If the sum of the elements in a column of RGA is small (much smaller than 1), then one may consider deleting the corresponding input. Extra outputs: If all elements in a row of RGA are small (much smaller than 1), then the corresponding output cannot be controlled (Lee et al., 1995).

4. The RGA can be used to measure diagonal dominance:

$$\text{RGA-number}=\|\Lambda(G)-I\|_{\text{sum}}$$

where the sum matrix norm is the sum of the element magnitudes.

For decentralized control, pairings for which the RGA-number at crossover frequencies is close to 1 are preferred. Similarly, for certain MV design methods, it is simpler to choose the weights and shape the plant to be diagonally dominant with a small RGA-number (Skogestad, 1996).

5. RGA and decentralized control:

(a) Integrity: For stable plants avoid input- output pairing on negative steady-state RGA elements. Otherwise, if the subcontrollers are designed independently each with integral action, then the interaction will cause instability either when all of the loops are closed, or when the loop corresponding to the negative gain becomes inactive. Interestingly, this is the only use of RGA directly related to Bristol's original definition (Bristol, 1966).

(b) Stability: Prefer pairings corresponding to an RGA-number close to 0 at crossover frequencies.

2.4. Previous Work on Controller Tuning

2.4.1. The Concept of Autotuning

An automatic tuning procedure mainly consists of two stages: (a) The identification phase; (b) the controller design phase. Aström-Hagglund autotuners are commonly used to identify the system (Aström and Hagglund, 1984). The main principle of this autotuner is based on the observation that a feedback system in which the output y lags behind the input u by $-\pi$ radians may oscillate with the ultimate period P_u. Instead of using a Z-N approach (Ziegler and Nichols, 1942) to the system with very big oscillations, a relay feedback test can be performed.

The advantage of the relay feedback is to keep the oscillation amplitudes as small as possible (Figure 2.7). At first, the input u which is a function of the error is increased by a small amount h. u can also be zero in some cases. As soon as the output is moving upwards by increasing the input amount h, the input is switched to lower the position by h amount. A transient response will occur until the steady state in which useful parameters can be obtained to identify the system. From the relay feedback test, ultimate gain K_u (2.33) and ultimate frequency ω_u (2.34) are available. By doing some approximation from the describing-function analysis (Aström, 1983), K_u and ω_u can be determined as:

$$K_u = \frac{4h}{\pi a} \tag{2.33}$$

$$\omega_u = \frac{2\pi}{P_u} \tag{2.34}$$

where a is the amplitude of oscillation, h is the amount of increase and decrease in the input and P_u is the ultimate period. Because the output is -π radians behind the input (let the input be positive when the output is negative and vice versa), a special point on the Nyquist curve is obtained by -π radians. This point is used to design the controller.

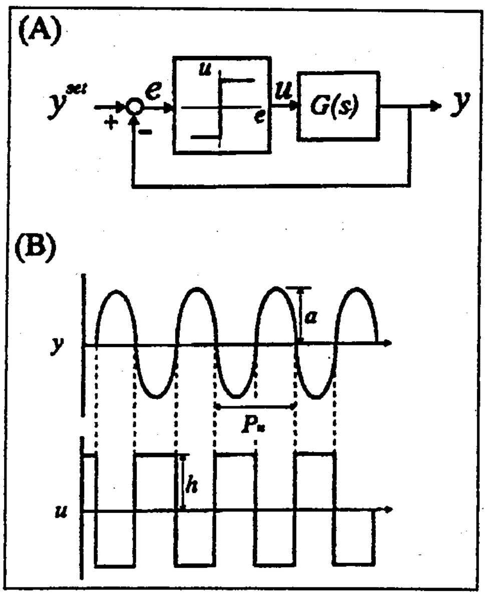

FIGURE 2.7. Amplitudes and periods (B) in relay-feedback systems (A)

For SISO systems, the classical and the most popular method is the Z-N method to tune the controller. The controller gain K_C and the integral time T_I of a PI controller can be found with the simple calculation:

$$K_C = K_u / 2.2 \tag{2.35}$$

$$T_I = P_u / 1.2 \tag{2.36}$$

For a PID controller, the Z-N tunings are:

$$K_C = 0.6K_u \tag{2.37}$$

$$T_I = P_u / 2 \tag{2.38}$$

$$T_d = P_u / 8 \tag{2.39}$$

where T_d is the derivative time.

Z-N tuning method is based on tuning the overdamped systems. There will be an overshoot which is four times higher than the second overshoot in the unit step response.

We will use PI control in our control schemes because of the following reasons:

- The process is "slow". Investigating the change in error with respect to time is not very useful.

- If the delay time of the process increases, it will be nearly impossible to set the differential time of the controllers to a constant value to increase the system performance.

- Uncertainties and disturbances are less dangerous in the control problem of the system because of the absence of T_d.

Tuning constants can also be obtained by other design methods such as gain or phase margin specifications according to Aström-Hagglund (Aström and Hagglund, 1984), dominant pole design according to Aström-Hagglund (Aström and Hagglund, 1988) and M-circle criterion (Shei, 1992). All of these methods used in designing the controller are based on a single point on the Nyquist curve of the frequency response of the plant.

2.4.2. Z-N Tuning Method

Z-N method (Ziegler and Nichols, 1942) is a simple, popular and easily implementable method in tuning PID type of controllers. But the problem with Z-N is that the method is suitable for SISO and overdamped systems. For MIMO systems and underdamped systems, the stability problem may occur. The sequential design procedure may produce an underdamped system (Shen and Yu, 1994). Furthermore, in the sequential design, the underdamped transfer function $g_{ii,CL}$ has to be tuned again with the simple closed-loop system. Therefore, modifications in Z-N tuning has to be made to avoid accumulating problems of underdamped poles.

2.4.3. Modified Z-N Tuning Methods

In our study, the Z-N method is modified in three major ways. According to:

(a) Biggest Log Modulus Tuning (BLT) method (Luyben, 1986): The system studied is an *nxn* MV process with the given transfer function $Y=G(s)U$ where Y is the vector of the controlled variables y_i (i=1,2,...n), U is the vector of manipulated variables u_j (j=1,2,...n) and $G(s)$ is the matrix of open-loop process transfer functions g_{ij}. These transfer functions are typically of the form

$$g_{ij} = \frac{K_{ij}(T_{1ij}s+1)e^{-D_{ij}s}}{(T_{2ij}s+1)(T_{3ij}s+1)(T_{4ij}s+1)} \tag{2.40}$$

and the controller structure is of the form:

$$K(s)=\begin{bmatrix} K_1(s) & & & 0 \\ & K_2(s) & & \\ & & \ddots & \\ 0 & & & K_n(s) \end{bmatrix} \tag{2.41}$$

where $$K_i(s)=K_{Ci}(1+\frac{1}{T_{Ii}s})$$

K_C is the controller gain and T_I is the controller integral time.

First, the Z-N settings are calculated for each loop. The ultimate gain K_u and the ultimate frequency ω_u of each diagonal transfer function $g_{ii}(s)$ are calculated in SISO way. In our work, relay feedback is used to find these ultimate gain and ultimate frequency.

Next, a factor F is assumed to modify the settings. In Luyben's study, typical values for F vary from 2 to 5. The gains of all feedback controllers are calculated by dividing the Z-N gain by the factor F and the integral times of all controllers are calculated by multiplying the Z-N integral times by the same factor F. So,

$$K_{Ci}=\frac{K_{ZNi}}{F} \tag{2.42}$$

where $$K_{ZNi}=\frac{K_{ui}}{2.2},$$

and

$$T_{Ii}=FT_{ZNi} \tag{2.43}$$

where $$T_{ZNi}=\frac{2\pi}{1.2\omega_{ui}}.$$

The factor F is called as the "detuning factor" which is applied to all loops. The larger the value of F, the more the system will be stable but the more sluggish will be the set-point and load disturbance responses.

Consider the input output relation of a closed loop stable system:

$$Y = GU = GK(R - Y) \tag{2.44}$$

where R is the reference values vector.

Solving for Y gives

$$Y = [I + GK]^{-1} GKR \tag{2.45}$$

Since the inverse of a matrix has the determinant of the matrix in the denominator, the closed-loop characteristic equation of the MV system is the scalar equation

$$\det(I + GK) = 0 \tag{2.46}$$

If the determinant of $I+GK$ is plotted as a function of frequency and the encirclements of the origin are observed, the number of the right half plane zeros of the closed-loop characteristic equation can be obtained. To make this plot look like the SISO scalar plot, one is subtracted from the equation (2.46) above and renamed as $W(s)$.

$$W(s) = -1 + \det(I + GK) \tag{2.47}$$

W is plotted as a function of frequency. The closer W approaches the point (-1,0), the more unstable the MV system becomes. The quantity $W/(1+W)$ will be similar to the closed loop servo-transfer function for a SISO loop $GK/(1+GK)$. Therefore, a multivariable closed-loop log modulus can be defined as:

$$L_c = 20\log\left|\frac{W}{1+W}\right| \tag{2.48}$$

The BLT tuning method is based on varying the factor F until the "biggest log modulus" $(L_c)_{max}$ is equal to some reasonable number. This reasonable value can be set to +2 dB for $(L_c)_{max}$ for SISO according to experience. This value gives reasonable time-domain responses for set point changes and load disturbances. To obtain good responses for nxn systems

$$(L_c)_{\max} = 2n \tag{2.49}$$

will be a reasonable choice, e.g. for a 2x2 system $(L_c)_{max}$ =4, for a 3x3 system $(L_c)_{max}$ =6. These empirical settings suggest that the higher the order of the system, the more underdamped the closed loop system must be to achieve reasonable responses.

This procedure guarantees that the system is stable with all controllers on automatic and also that each individual loop will be stable if all the others are on manual. However, further checks should be done to maintain stability for other combinations of manual/automatic operation.

The method weighs each loop equally. If it is important to keep tight control of one variable, different weighting factors can be applied to modify this rule. The less important loop could be detuned more than the important loop as described in a study (Palmor et al., 1995).

(b). BLT method with loop weighting: In SISO systems there are only a finite number of critical gains that brings the system to instability. In most cases, there is only one such gain. For a 2x2 system, there are infinitely many gains K_{C1}, K_{C2} that, when replacing $K_1(s)$ and $K_2(s)$, lead to neutral stability. The collection of these gains is the stability limit of the system. In Figure 2.8, three typical cases of instability limits are shown. The axes are $K_i g_{ii}(0)$ for i=1,2.

Each point on the curves corresponds to a pair of gains (K_{1cr}, K_{2cr}) and a critical frequency ω_{cr}. The points on the axes represent the situation where one loop is open. If the system does not have full interaction, i.e. $g_{12}(s)$ or $g_{21}(s)$ or both are zero, the stability limits take the rectangular form (curve 1 in Figure 2.8). The other two curves (curve 2 and

3) represent two typical cases of systems with interactions. The deviation from rectangular shape may be regarded as a crude measure of interaction (Palmor et al., 1995).

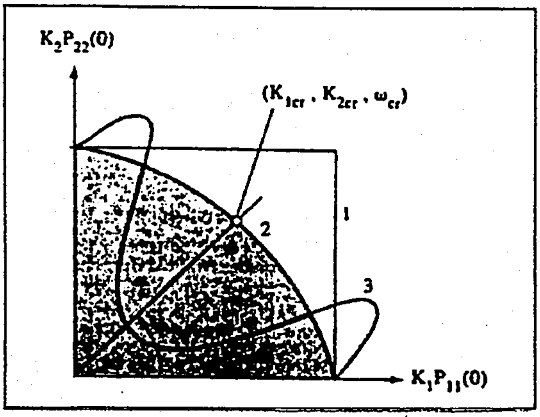

FIGURE 2.8. Stability limits: three typical cases (Palmor et al., 1995)

It is obvious that different critical points lead to different settings. Therefore, the critical point needs to be specified. So, the critical point is named as the desired critical point (DCP). The choice of the DCP depends on the relative importance of the loops. A particular point can be described by the ratio:

$$\tan\phi = \frac{W_f}{1} = \frac{K_{2cr}g_{22}(0)}{K_{1cr}g_{11}(0)} \tag{2.50}$$

where W_f is the weighting factor of the second loop with respect to the first loop. For a 2x2 MV system W_f can be chosen according to the importance of the loop, e.g. selecting $W_f>1$ means that loop two should be under tight control relative to loop one. But this weighting method cannot be easily applied to systems with higher numbers of inputs and outputs than 2x2. Matrices related with ϕ should be constructed, which is not suitable for one-button-push type of autotuning.

(c). Direct modification of Z-N rules (Shen and Yu, 1994): Any familiar SISO tuning method like gain margin, phase margin, biggest log modulus criterion can be applied for the PI controller design. However, when the identification phase is done according to relay feedback type and the K_u and ω_u are available, it is logical to choose the

Z-N type of method. It is obvious that any modification applied to the tuning constants is more conservative. The detuning procedure is very similar to the BLT technique. A single detuning factor is employed to find appropriate constants.

According to a reasonable number of tests on process industry applications, the value F is chosen approximately 2.5 in some studies (Luyben, 1986; Marino-Galarraga et al., 1987).

$$K_C = \frac{K_{C,ZN}}{F} \tag{2.51}$$

$$T_I = T_{I,ZN} F \tag{2.52}$$

The modified Z-N method for PI controller becomes:

$$K_C = K_u / 3 \tag{2.53}$$

$$T_I = P_u / 3 \tag{2.54}$$

Since the damping coefficient of $g_{22,CL}$ comes from $h_1 = g_{11}k_1[I + g_{11}k_1]^{-1}$ (for a 2x2 system), $(L_c)_{max}$ should be investigated of the modified Z-N type of tuning when applied to typical $g_{ii}(s)$. Figure 2.9 shows that damping coefficient of the Z-N method for the transfer functions of type $e^{-Ds}/(T_p s+1)$. Shen and Yu's modified method is less underdamped. In Figure 2.10, the $(L_c)_{max}$ plots show nearly the same kind of characteristics.

Another keypoint is to investigate the stability for any tuning method. If good performance is needed, the control system will not be very stable. Tan (Tan, 1985) made research on the third order processes of the form

$$G(s) = \frac{K_p}{T_p^2 s^2 + 2T_p \zeta s + 1}\left(\frac{T_{p2}s+1}{T_{p1}s+1}\right)e^{-Ds} \tag{2.55}$$

varying the constants in the following ranges: K_p=1, T_p=5, T_{p1}=0...10, T_{p2}=0...10, ζ=0.1...1 and D/T_p=0.02...0.2.

Figure 2.11 shows the stability regions for different tuning methods with D=0.1 and D=1.

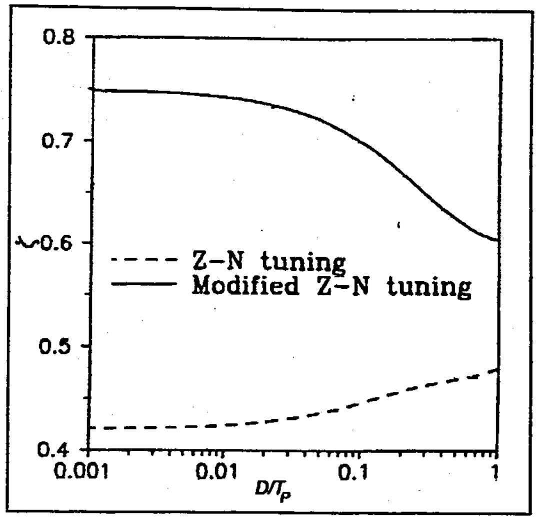

FIGURE 2.9. Damping coefficient for the transfer functions of type $e^{-Ds}/(T_ps+1)$ (Shen and Yu, 1994)

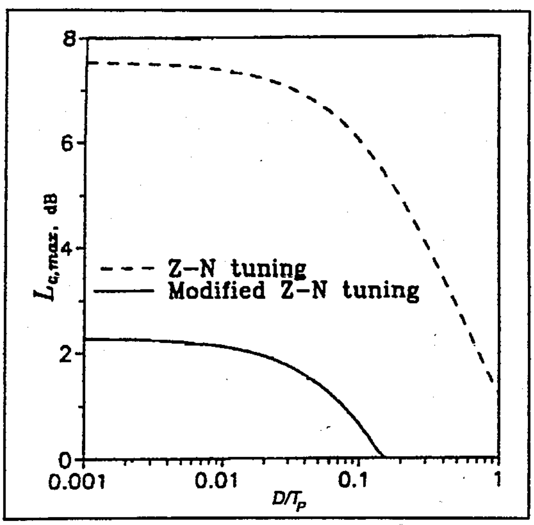

FIGURE 2.10. $(L_c)_{max}$ for the transfer functions of type $e^{-Ds}/(T_ps+1)$ (Shen and Yu, 1994)

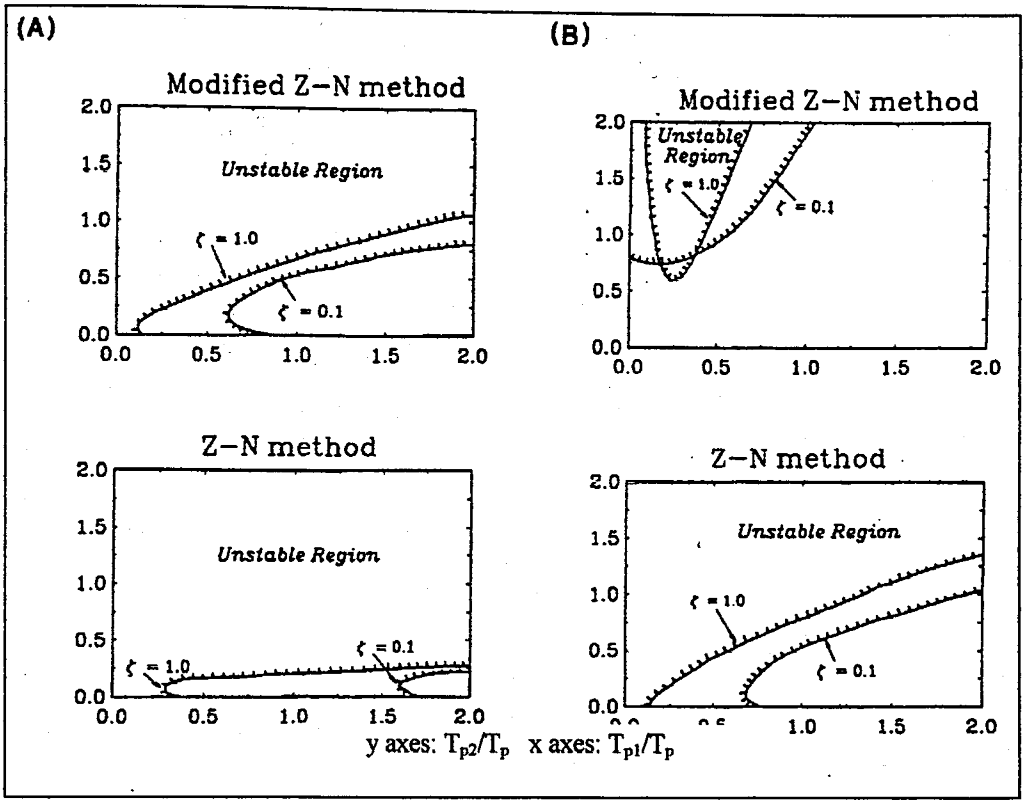

FIGURE 2.11. Stability regions for different tuning methods: (A) D=0.1, (B) D=1.0 (Shen and Yu, 1994)

It is clear that an instability region is often present in the region when $T_{p2}>T_{p1}$. The reason is the larger dead time constant in the last equation results in a resonant peak in the frequency response of the system, which can be interpreted as an enhancement of the underdamped behavior. Using a much larger F (detuning the controller further), stability can be achieved, but the responses will be more sluggish than normal.

The $(L_c)_{max}$ analyses and the stability for MIMO systems show that this modified Z-N tuning method works well with the control schemes we investigate. The tuning constants are directly obtained from the relay-feedback test, which shows that the method is very suitable for automation.

2.4.4. Autotuning in MIMO Systems

The use of relay feedback tests for the identification of the transfer function matrix $G(s)$ in a multivariable system is studied by Luyben (Luyben, 1987). He considers an nxn MV system $G(s)$ with the entry $g_{ij}(s)$. A relay feedback test is performed to each column in the transfer function matrix. This procedure is repeated n times. After the test each individual element in a column of the transfer function matrix can be found by fitting the corresponding point on the Nyquist curve to an assumed model. At the end, he obtains n^2 transfer functions in $G(s)$. Once the transfer function matrix is identified, independent design methods (Chiu and Arkun, 1992), for example, BLT method (Luyben, 1986) and μ (structured singular value) tuning criterion (Skogestad and Landström, 1990), can be employed to find the tuning constants. In the independent design, all controllers are designed independently.

All of these tuning methods are off-line controller design methods, they can be automated. By using independent design methods, extensive computational work and iteration techniques are needed to calculate the controller parameters from the n^2 transfer functions. Moreover, Luyben (Luyben, 1987) and Chang, Shen and Yu (Chang et al., 1992) state that the columnwise identification procedure can lead to inconsistency in non-linear MV processes.

2.4.5. Sequential Design

Another approach to the identification and autotuning problem is that there is no need to find all n^2 individual transfer functions for the controlled design. So, an attractive alternative approach occurred in MIMO autotuning: The sequential design. Sequential design has been widely investigated by some studies (Mayne 1973; Mayne, 1979; Bhalodia and Weber, 1979; Bernstein, 1987; O'Reilly and Leithead, 1991; Chiu and Arkun, 1992). What sequential design means is that each controller in a MIMO system is designed in sequence. That means, a MIMO process is divided into SISO systems and all the controllers of the SISO systems are designed in sequence.

Assume a 2x2 system (the method for a *nxn* system is the same) with a known pairing and under decentralized control (Figure 2.12). Initially, the second loop (which is "slower" than the first loop) is kept open. A relay is placed between y_1 and u_1 to produce oscillations in the first loop. Following the relay feedback test in the first loop, a controller can be designed from the ultimate gain and the ultimate frequency. Once the parameters are obtained for the controller on the first loop and the controller is tuned, the second loop is closed, a second relay is placed between y_2 and u_2. Another relay feedback test is performed while the first loop is on automatic. Then, a controller can be designed for the second loop with respect to the ultimate gain and ultimate frequency. Following this, both loops are closed, a relay is applied to the first loop while the second loop is on automatic. That leads us to a new pair of parameters for the controller on the first loop. This procedure is repeated until the controller parameters converge. Experience shows that the controller parameters converge after 3-4 relay feedback tests for 2x2 systems. If the convergence criteria are kept tighter, the parameters converge later than 3-4 relay feedback tests. This MIMO autotuning concept uses the "identification-and-design" sequence on SISO transfer functions. So, there is no need to identify all the n^2 independent transfer functions.

Sequential design has several advantages compared to independent design methods. First of all, sequential design approach treats the MIMO system as a sequence of SISO systems in which relay-feedback system is proven to be useful and reliable (Maciejowski, 1989). Secondly, the autotuner identifies not every single n^2 components of the transfer function matrix, which makes this approach an efficient one. Thirdly, compared to the independent identification, sequential identification and design finds the transfer function (can be a combination of $g_{ij}(s)$) which is needed for controller design at a single operating point, whilst independent identification finds linear $g_{ij}(s)$ at different operating points. Thus, it is not necessary to identify all components of the transfer function matrix to tune the controllers.

Another possibility is to use decentralized relay feedback (DRF) (Wang et al., 1997). Wang and his coworkers state that it is used as a test for the process frequency response matrix estimation. DRF is a complete closed loop test, meaning that for an *nxn* plant at any instant of the test, all the *n* outputs are simultaneously under feedback control, while IRF and SRF are only partial closed loop tests. For IRF, only one loop is closed, with the other *n*-1 open. For SRF, at the *i*th test, *i* loops are closed, with *n-i* loops open. Closed-loop testing is always preferred to open-loop testing (Aström and Hagglund, 1988), because a closed-loop test keeps outputs close to the setpoints, so that it causes less perturbation to the process and makes a linear model valid. DRF is used whenever an identification of multivariable process interaction is needed.

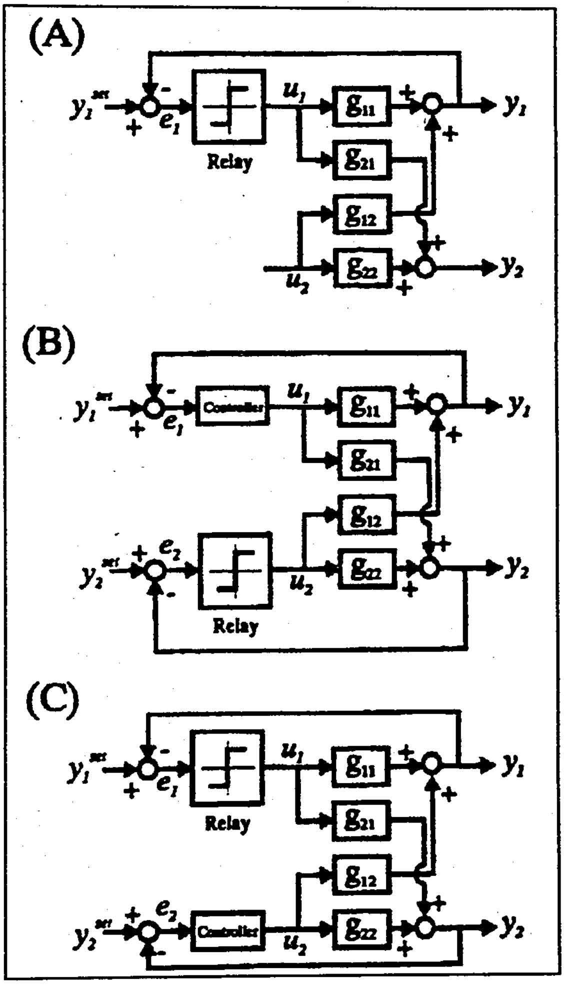

FIGURE 2.12. Application of relay feedback and sequential design (Shen and Yu, 1994)

If an *nxn* process is controlled by DRF, its outputs usually oscillate in the form of limit cycles after an initial transient. Each output has its own oscillation frequency, and they may be different, e.g. a 2x2 process consisting of two independent (or very little coupled) but different loops has different output oscillation frequencies. However, in a study (Atherton and Warwick, 1988) it says that for a typical coupled MV processes, n outputs have the same oscillation frequencies but with different phases. This frequency for a MV process is called as "process critical frequency" in Wang's work (Wang et al., 1997).

Even though there are many techniques to apply, the processes in the tested examples in this study must satisfy the following conditions:

- The process is open-loop stable.

- Interactions are on a reasonable level. The loops does not interact with each other or the loops have weak interactions with each other. So, decentralized controllers may be used.

- The process has low-pass characteristics. Most processes in the process and chemical industries satisfy this assumption.

2.4.5.1. Sequential Identification Procedure. It is very important that system identification plays a significant role for the proper tuning of controllers. Most of the time, independent identification is used. In the independent identification, identification of MIMO systems is done by manipulating the input one-by-one. That means the first column of the transfer function matrix is obtained for a change in u_1 while the rest of the inputs (u_j, $j \neq 1$) are kept constant (Figure 2.13). Signal flow is always from one input to the corresponding one output. However, independent identification brings some drawbacks to the identification scheme for the nonlinear MV processes (Luyben, 1987; Chang et al., 1992). Although the errors for each identified transfer function is in an acceptable range, the identified transfer function matrices cannot describe even the fundamental process characteristics.

So, transfer functions have to be consistent. The successful identification procedure must satisfy some consistency criteria. Therefore, the identification procedure must be renewed and satisfy the consistency relations.

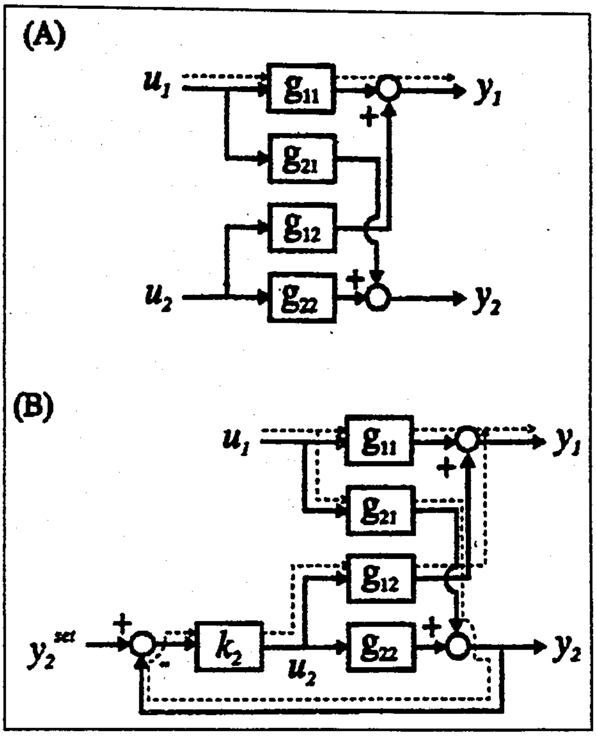

FIGURE 2.13. Signal flow in: (A) independent identification and (B) sequential design (Shen and Yu, 1994)

The actual transfer function, we need to consider in designing controllers for MV systems, consists of combinations of $g_{ij}(s)$. For instance, let us assume the closed loop transfer functions for a 2x2 MV process:

$$g_{ii,CL}(s) = g_{ii}(1-\kappa h_j)\ ,\ i=1,2 \text{ and } j \neq i \tag{2.56}$$

where $\kappa = \dfrac{g_{12}g_{21}}{g_{11}g_{22}}$ and $h_j = g_{jj}k_j\left[I + g_{jj}k_j\right]^{-1}$.

If the cross-diagonal elements of the transfer function matrix $g_{ij}(s)$ come directly from the independent identification (without checking consistency relations), the design can cause errors. The simplest way to meet consistency relations is to perform sequential identification (Figure 2.12), i.e., performing the identification procedure in a sequential manner. See Figure 2.12., which illustrates the sequential identification procedure with a relay feedback test.

Sequential identification has a major advantage against the independent identification. The sequential identification finds the essential element, $g_{ii,CL}$, for the controller design. Satisfying and finding this $g_{ii,CL}$ the consistency relations are achieved automatically.

2.4.5.2. Summary of Sequential Identification and Design Procedure.

(1a). Perform a relay-feedback test on y_1-u_1 while loop 2 is open. (see Figure 2.12.a)

(1b). Design a PI controller for the first loop based on K_{u1} and ω_{u1} according to the investigated rules.

(2a). Perform a relay feedback test on y_2-u_2 while loop 2 is closed and the controller k_1 is on-line. (Figure 2.12.b)

(2b). Design a PI controller k_2.

(3a). Perform another relay feedback test on y_1-u_1 while controller k_2 is on-line.

(3b). Design a PI controller for the loop 1.

(4). Repeat the procedure until all parameters converge.

In theory, further steps are required in autotuning, i.e., redesigning controllers for many times. However, parameters converge faster than expected so that the tuning sequence is not repeated more than 2-3 or 4 times.

2.4.6. Properties and Common Problems in MIMO System Tuning

Although it is easy to apply and to obtain good results, like every tuning process this autotuning method might have some unwanted properties or problems.

2.4.6.1. Convergence. Recall the autotuning steps (e.g., steps 2 & 3 in the section of performance evaluation of linear models). In step 2, the identification phase find K_{u2} and ω_{u2} while the first controller is on-line. ($h_2(j\omega)$ is known) and the controller design phase calculates the constants for the controller 2 from K_{u2} and ω_{u2}, consequently h_2. When we go back to loop 1 in step 3, our aim is to find K_{u1} and ω_{u1} with controller 2 on-line. Therefore, the problem can be formulated as follows:

Find ω_{u1} and ω_{u2} such that the following two non-linear equations converge:

$$f_1(\omega_{u1},\omega_{u2}) = \tan^{-1}\left[\frac{\mathrm{Im}\{g_{11,CL}(j\omega_{u1},j\omega_{u2})\}}{\mathrm{Re}\{g_{11,CL}(j\omega_{u1},j\omega_{u2})\}}\right] \tag{2.57}$$

$$f_2(\omega_{u1},\omega_{u2}) = \tan^{-1}\left[\frac{\mathrm{Im}\{g_{22,CL}(j\omega_{u1},j\omega_{u2})\}}{\mathrm{Re}\{g_{22,CL}(j\omega_{u1},j\omega_{u2})\}}\right] \tag{2.58}$$

where

$$g_{11,CL}(j\omega_{u1},j\omega_{u2}) = g_{11}(j\omega_{u1}) \times \left[1-\kappa(j\omega_{u1})\frac{g_{22}(j\omega_{u1})k_2}{1+g_{22}(j\omega_{u1})k_2}\right] \tag{2.59}$$

$$k_2 = \frac{1}{3\left|g_{22,CL}(j\omega_{u2})\right|}\left[1+\frac{1}{\frac{4\pi}{\omega_{u2}}j\omega_{u1}}\right] \tag{2.60}$$

and the similar expressions are also valid for the $g_{22,CL}(j\omega_{n1},j\omega_{n2})$.

These nonlinear equations must have been solved according to the sequentially, not simultaneously. That means: When ω_{u1} is solved in the kth iteration in the equation (2.57) with ω_{u2} taking constant values from previous iteration. In the linear equations, we call this Gauss-Seidel method for solving linear algebraic equation in sequential manner.

Consider a set of linear algebraic equations

$$Ax = B \tag{2.61}$$

where A is the coefficient matrix, x is the solution vector, and B is a vector constant. For this sequential equation solving to converge, the necessary and sufficient condition is

$$\rho\left[I - A_{diag}^{-1}A\right] < 1 \tag{2.62}$$

where ρ is the spectral radius (the largest absolute value of the eigenvalue of the expression in the brackets and A_{diag} is the matrix with only a_{ii} components. For a 2x2 system, the last inequality turns out to be

$$\frac{a_{12}a_{21}}{a_{11}a_{22}} < 1 \tag{2.63}$$

In case of a sequential-design problem, the equations take the following form (Shen and Yu, 1994):

$$\begin{pmatrix} \left(\frac{\partial f_1}{\partial \omega_{u1}}\right)_{\omega_{u2}} & \left(\frac{\partial f_1}{\partial \omega_{u2}}\right)_{\omega_{u1}} \\ \left(\frac{\partial f_2}{\partial \omega_{u1}}\right)_{\omega_{u2}} & \left(\frac{\partial f_2}{\partial \omega_{u1}}\right)_{\omega_{u2}} \end{pmatrix} \begin{pmatrix} \omega_{u1} \\ \omega_{u2} \end{pmatrix} = \begin{pmatrix} -\pi \\ -\pi \end{pmatrix} \tag{2.64}$$

Convergence is satisfied, if:

$$\frac{\left(\frac{\partial f_1}{\partial \omega_{u2}}\right)_{\bar{\omega}_{u1}} \left(\frac{\partial f_1}{\partial \omega_{u1}}\right)_{\bar{\omega}_{u2}}}{\left(\frac{\partial f_1}{\partial \omega_{u1}}\right)_{\bar{\omega}_{u2}} \left(\frac{\partial f_1}{\partial \omega_{u2}}\right)_{\bar{\omega}_{u1}}} < 1 \tag{2.65}$$

where the overbar stands for the solution of the nonlinear equations.

This phenomenon has a physical interpretation. From the equation for $g_{11,CL}$, it can be seen that the only way ω_{u2} can affect $g_{11,CL}(j\omega_{u1}, j\omega_{u2})$ is through the controller 2 (or through h_2). However, generally the magnitude of h_j is kept constant for a range of frequencies and the bandwidth of h_j is often larger than that of g_{ii}. That brings us to the point that the change of ω_{u2} has little effect on the solution of $g_{11,CL}$. Experiences show that no convergence problem occurred during the studies (Shen and Yu, 1994).

2.4.6.2. Obtaining the Tuning Sequence. The sequential design brings us a new problem: Which of the loops should be tuned first and how should the others follow the first loop? The previous work on this subject shows that the tuning sequence greatly influences the speed of convergence. But the faster the convergence, the sooner the autotuning procedure ends. In our study, the prior information of the process is mostly not known, so we have to perform a test to find the order of the speed of various loops. Performing relay feedback tests will solve the problem. Defining 'relative loop speed' s_i (Marino-Galarraga et al., 1987):

$$s_i = \frac{\omega_{ui}}{\sum_j \omega_{uj}} \tag{2.66}$$

where ω_{ui} is the ultimate frequency of g_{ii}. s_i will take values between 0 and 1. For a 3x3 system, if the loop speed of loop one is faster than the other two loops, then s_1>1/3. The relative loop speed is only determined through the diagonal components of the closed-loop transfer function matrix. This is a rough idea to estimate the speed of the loops, but it is the easiest and the most efficient way in the experiments. Once the relative loop speeds are determined, the order of the tuning sequence can be determined.

Faster convergence can be achieved if the fast loop is tuned first. For a system with very different loop speed, the system interaction has little impact on the fast loop dynamically. For instance, the effect from system interaction through the slow loop does not show on the fast variable until the transient (resulting from the transfer function of fast loop) almost dies out. On the contrary, the effect of the fast loop always acts in the slow loop. Mathematically expressed, the bandwidth of complementary sensitivity function for the slow loop is much smaller than that of the fast loop, so that the slow loop acts like a low-pass filter. Therefore, it is very likely that:

$$g_{ii,CL}(j\omega_{u1}) = g_{11}(j\omega_{u1})(1-\kappa(j\omega_{u1})h_2(j\omega_{u1})) \approx g_{11}(j\omega_{u1}) \tag{2.67}$$

The last equation points out that if the fast (loop 1) loop is tuned first, the solution will be closer to the exact solution. If we have the speed information for the loops, it will lead us to a more efficient autotuning procedure.

2.5. Uncertainty, Stability and Robustness Analysis Tools in MV Systems

Our main analysis tool will be the structured singular value (SSV), μ. μ analysis should be known to predict how the system will react when model uncertainties exist. μ analysis results also give an idea of the stability and robustness of a system under a prescribed performance weights.

2.5.1. General Control Configuration with Uncertainty

The starting point for robustness analysis is a system representation in which the uncertain perturbations are presented like below:

$$\Delta = diag\{\Delta_i\} = \begin{bmatrix} \Delta_1 & & & \\ & \cdot & & \\ & & \cdot & \\ & & & \Delta_i \\ & & & & \cdot \end{bmatrix} \tag{2.68}$$

where each Δ_i represents a specific source of uncertainty, e.g. input uncertainty, Δ_I, or parametric uncertainty δ_i, where δ_i is real. If we also pull the controller K, we get the generalized plant P (Figure 2.14). So, we obtain a structure which is suitable to analyze. This structure is called $N\Delta$-structure in Figure 2.15.

N is related to P and K by lower linear fractional transformation (LFT) (Skogestad, 1996).

$$N = F_l(P,K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \tag{2.69}$$

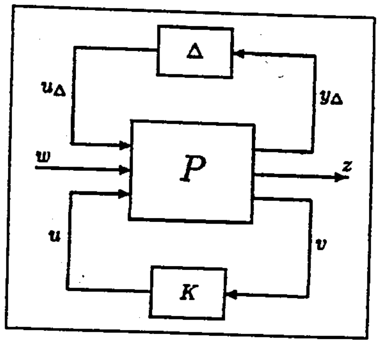

FIGURE 2.14. General control configuration

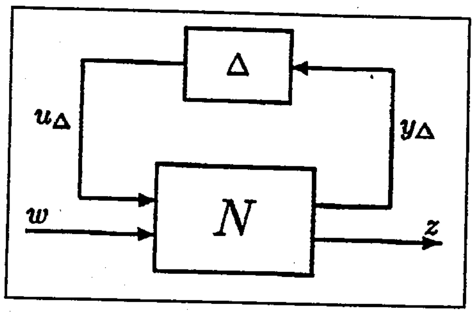

FIGURE 2.15. NΔ-structure

The uncertain closed-loop transfer function from w to z, $z=Fw$, is related to N and Δ by an upper LFT.

$$F=F_u(N,\Delta)=N_{22}+N_{21}\Delta(I-N_{11}\Delta)^{-1}N_{12} \tag{2.70}$$

To analyze the robust stability of F, the system can be arranged into a $M\Delta$-structure (see Figure 2.16) where $M=N_{11}$ is the transfer function from the output to the input of the perturbations.

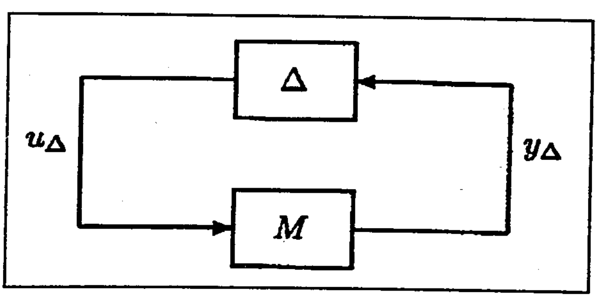

FIGURE 2.16. MΔ-structure for RS analysis

2.5.2. Definitions of Robust Stability and Robust Performance

In our study, stability and acceptable performance are checked for all tested examples. Basically, two main analysis methods are applied:

- Robust stability (RS) analysis: With a given controller K we determine whether the system remains stable for all plants in the uncertainty set.

- Robust performance (RP) analysis: If RS is satisfied, we determine how "large" transfer function from exogeneous inputs w to outputs z may be for all plants in the uncertainty set.

Considering the upper LFT and H_∞ norm, where H_∞ norm represents the peak value of the maximum singular values of a frequency response of a closed loop stable system, we will give some other properties of RS and RP. We use the H_∞ norm to define performance and require for RP that $\|F(\Delta)\|_\infty \leq 1$ for all possible Δ's. Generally, $F=w_P S_P$ is chosen, where w_P is the performance weight and S_P is the set of perturbed sensitivity functions.

So, the stability and performance requirements will be as follows:

- NS (nominal stability) ⇔ N is internally stable.
- NP (nominal performance) ⇔ $\|N_{22}\|_\infty<1$; and NS.
- RS (robust stability) ⇔ $F=F_u(N,\Delta)$ is stable for all Δ, $\|\Delta\|_\infty\leq 1$; and NS.
- RP (robust performance) ⇔ $\|F\|_\infty<1$; for all Δ, $\|\Delta\|_\infty\leq 1$; and NS.

In short, 'nominal stability' means that the system is stable with no model uncertainty, 'nominal performance' means that the systems satisfies the performance specifications with no model uncertainty, 'robust stability' means that the system is stable for all perturbed plants about the nominal model up to the worst-case model uncertainty, and 'robust performance' means that the system satisfies the performance specifications for all perturbed plants about the nominal model up to the worst-case model uncertainty.

2.5.3. The Structured Singular Value μ

The structured singular value (SSV or μ) is a function which provides a generalization of the singular value and the spectral radius, ρ. μ can be used to define necessary and sufficient conditions for RS and RP.

A simple definition for μ is: "Find the smallest structured Δ (measured in terms of $\sigma_{max}(\Delta)$) which makes $\det(I-M\Delta)=0$; then $\mu(M)=1/\sigma_{max}(\Delta)$."

$\mu(M)$ depends not only on M but also on the allowed structure for Δ.

More precisely, let M be a given complex matrix and let $\Delta=\text{diag}\{\Delta_i\}$ denote a set of complex matrices with $\sigma_{max}(\Delta)\leq 1$ and with a given block-diagonal structure (in which some of the blocks may be repeated and some may be restricted to be real). The real non-negative function $\mu(M)$, called the structured singular value, is defined as:

$$\mu_{\Delta}(M)=\frac{1}{\min\{\sigma_{max}(\Delta)\colon \Delta\in\mathbf{\Delta},\ \det(I-M\Delta)=0\}} \tag{2.71}$$

If no such structured Δ exists then $\mu(M)=0$.

A value $\mu=1$ means that there is a perturbation with $\sigma_{max}(\Delta)=1$ which is just enough to make $I-M\Delta$ singular (Skogestad, 1996). A larger value of μ is bad because smaller perturbations make $I-M\Delta$ singular, whereby a smaller value of μ is good.

2.5.4. P and N Matrices in RS and RP Analysis

We will use the structure below in our RS and RP analysis (Figure 2.17).

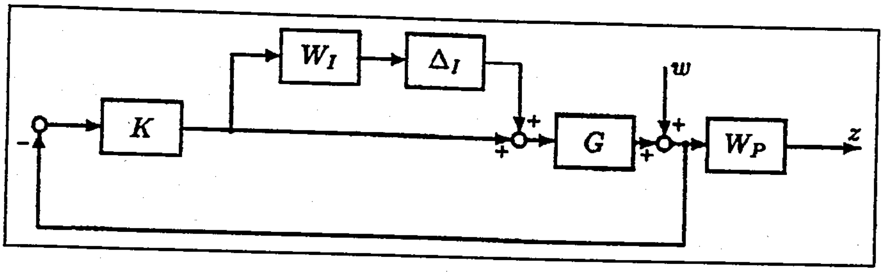

FIGURE 2.17. Structure to be used in RP and RS analysis

Moreover, our systems will have input uncertainties. Considering a feedback system with multiplicative input uncertainty Δ_I, a normalization weight w_I for the uncertainty and a performance weight w_P, we will use the equations below:

$$P = \begin{bmatrix} 0 & 0 & W_I \\ W_P G & W_P & W_P G \\ -G & -I & -G \end{bmatrix} \tag{2.72}$$

The transfer function from u_Δ to y_Δ is 0 because u_Δ has no direct effect on y_Δ except through K

$$P_{11} = \begin{bmatrix} 0 & 0 \\ W_P G & W_P \end{bmatrix} \quad , \quad P_{12} = \begin{bmatrix} W_I \\ W_P G \end{bmatrix}$$

$$P_{21} = \begin{bmatrix} -G & -I \end{bmatrix} \quad , \quad P_{22} = -G$$

and applying lower linear fractional transformation to P and K forms N as:

$$N = \begin{bmatrix} -W_I KG(I+KG)^{-1} & -W_I K(I+GK)^{-1} \\ W_P G(I+KG)^{-1} & W_P (I+GK)^{-1} \end{bmatrix} \tag{2.73}$$

2.5.5. Structured Singular Value Interaction Measure

Define:

$$\tilde{G} = diag\{g_{ii}\} \tag{2.74}$$

The magnitude of the off-diagonal elements in G (the interactions) relative to its diagonal elements are given by the matrix (Skogestad, 1996)

$$E = (G - \tilde{G})\tilde{G}^{-1} \tag{2.75}$$

and the $\tilde{S}$ and $\tilde{T}$ matrices are defined as:

$$\tilde{S} = (I + \tilde{G}K)^{-1} = diag\{\frac{1}{1 + g_{ii}k_i}\} \text{ and } \tilde{T} = I - \tilde{S} \tag{2.76}$$

Assume G is stable and that the individual loops are stable ($\tilde{T}$ is stable). Then the entire system is closed loop-stable ($\tilde{T}$ is stable) if

$$\sigma_{\max}(\tilde{T}) = \max_i \left|\tilde{t}_i\right| < 1/\mu(E) \text{ for all } \omega \tag{2.77}$$

Here $\mu(E)$ is the structure singular value interaction measure. It is desirable to have $\mu(E)$ small. Note that $\mu(E)$ is computed with respect to the diagonal structure of $\tilde{T}$ (a diagonal matrix), where $\tilde{T}$ may be viewed as the "design uncertainty".

A plant is (generalized) diagonally dominant at frequencies where $\mu(E(j\omega))<1$. From (2.77), $\sigma_{\max}(\tilde{T}) \geq 1$ can be allowed (tight control) at such frequencies and closed-loop stability is still guaranteed. (2.77) is a sufficient condition for the stability of the closed loop system. Violation of (2.77) does not always indicate instability but if (2.77) is satisfied, the system is always stable.

2.6. Details of Robustness Analysis and μ-optimal Tuning

μ interaction measure can be applied to the examples. First, μ-optimal settings are calculated through a controller synthesis algorithm. Then, uncertainties are formulated as matrices. Every single PID parameter is optimized to give appropriate results under model uncertainties and performance weights.

To find μ-optimal settings for the PID controllers a fixed structure is defined for K (Musch and Steiner, 1995):

$$K(\theta) = \arg \inf_{\substack{\text{K stabilizing} \\ \text{K with fixed structure}}} \left\| \mu_{\tilde{\Delta}}(M) \right\|_{\infty} \tag{2.78}$$

where the plant M incorporates the process models, the weighting functions, and the controller (see $M\Delta$-structure), $K(\theta)$ denotes the controller, θ the controller parameters, $\tilde{\Delta}$ the matrix of uncertainty blocks. The solution of the problem in (2.78) must be done with parameter optimization methods. To find the infinity norm in the equation (2.78), μ is calculated for a number frequency points to achieve a reasonable estimate of its maximum in the frequency range of interest. The estimated maximum can be very sensitive to the number of frequency points calculated and even small changes in the tuning parameters can cause large changes in the μ values. Therefore, estimated maximum can be a discontinuous function of the controller parameters θ. Numerical treatment is simplified, if the design objective is formulated as the summation of the cube of the μ for all k frequency points (Musch and Steiner, 1995):

$$\theta = \arg\_\inf_{\theta} \sum_{i=1}^{k} \mu_{\tilde{\Delta}}^{3} \left\{ F_l\left[P, K(\theta)\right] \right\} \tag{2.79}$$

The plant P incorporates the process models and the weighting functions (Beaven et al., 1996). This design objective is less sensitive to the number of frequency points calculated. In tested examples, 30 frequency points were sufficient for the convergence of the optimization problem in (2.79). Using this algorithm, computation time is reduced to approximately two hours for a set of μ-optimal parameters on a Pentium-166 based personal computer. This is approximately equal to 1.2×10^9 flops.

During parameter optimization, nominally unstable control loops may be generated. Therefore, some boundary conditions have to be applied or the process is checked manually. After a successful run of the program (μ is kept under 1), μ-optimal settings are found with the pre-defined intervals for the controller settings, uncertainties and perturbations.

μ interaction measure can then be applied to the system under given input uncertainties and under specific performance weights. All possible settings, included μ-optimal settings are used to find $1/\mu(E)$ for all frequencies in a given interval (see equations from 2.75 to 2.77) and then compared with the frequency response of the diagonal elements of the complementary sensitivity function. If $1/\mu(E)$ plot is higher than the singular values of the diagonal elements of the complementary sensitivity function, then the system is nominally stable.

3. THE DEVELOPED AUTOMATIC TUNING PROGRAM

In our study, one of our aims is to construct a program to tune the PID controllers for industrial process systems. This program considers many concepts explained in the previous chapter. The advantage of this program is to automate the tuning sequence and to make the sequence compatible for one-button-push type of tuning.

3.1. Objectives

The program must satisfy the following conditions:

- Identifying the system or the basic plant properties to tune the controllers.
- Obtaining system parameters and unknowns for a tuning sequence.
- Performing a relay feedback test according to the wanted relay feedback scheme.
- Finding a sequence to tune the loops.
- Obtaining and setting corresponding parameters according to tuning schemes.
- Repeating the sequence until a meaningful result appears.

3.2. The Program Algorithm

Initially, the sign of the diagonal elements in the transfer function are found. This is used to find the sign of the gain of the controller parameters. Then, the tuning sequence is obtained by comparing loop speeds. Loop speeds are found by evaluating the response of the diagonal elements of the transfer function to a relay feedback. The ranking of loop speeds from fast to slow are done and a tuning sequence is obtained. After that, the identification procedure begins. At beginning, all inputs to the plant are kept zero except the loop which has the highest loop speed. A relay is applied as input to this loop. Then, a controller is designed for this loop. Keeping this loop controlled with the tuned controller, the same procedure is applied for the second fastest loop until all loops are tuned and some controller parameters are obtained. The tuning scheme is continued until the controller parameters converge.

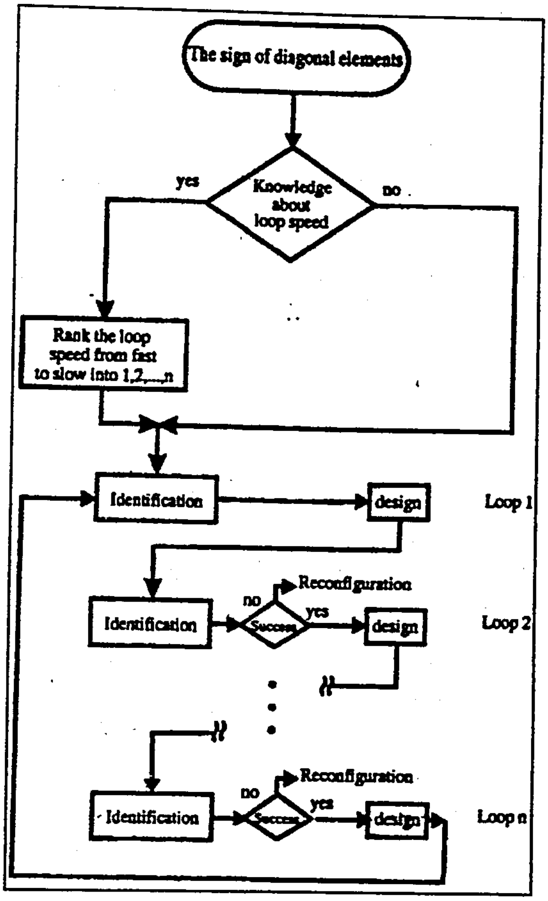

FIGURE 3.1. The algorithm of the developed autotuning program

For a 2x2 MV process, the identification and design procedures are given in 2.4.5.2.

3.3. Application of the Autotuning Program to a 2x2 MV Process.

The developed autotuning program is run to tune the Wood and Berry Distillation Column (Wood and Berry, 1973) with the tuning criteria developed by Shen and Yu. The picture of the example under the test is given is Figure 3.2. The "on-screen" appearance of the steps are given following the Figure 3.2. The variables and results are created by the simulink model in Fig. 3.2 and processed with the autotuning program "tun2c.m".

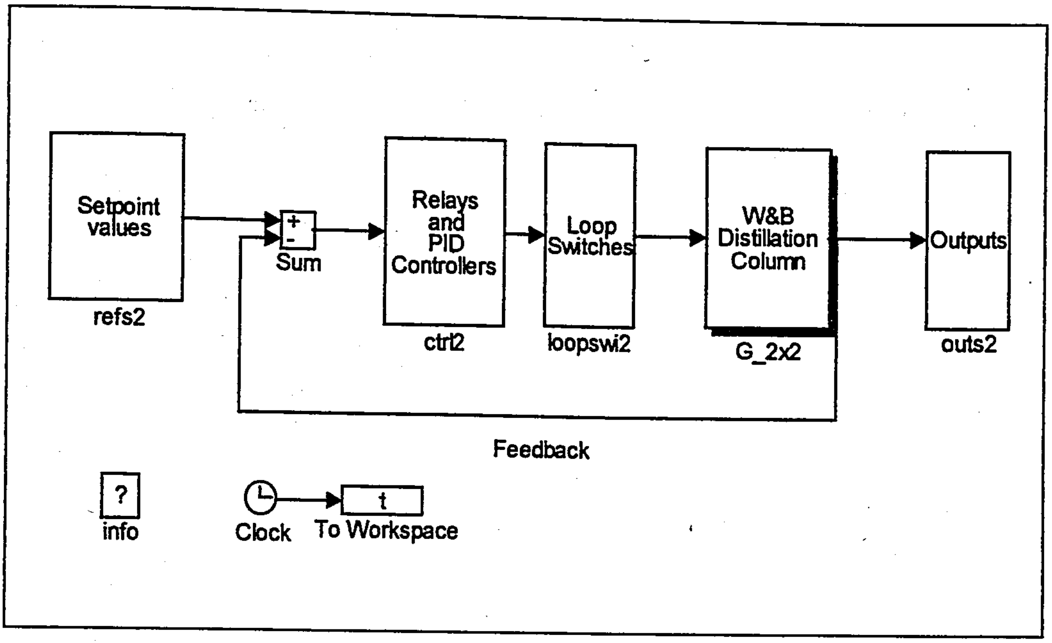

FIGURE 3.2. The model of the plant created in MATLAB

The tuning program runs and makes a connection with the variables in the simulink model, asks the method we want to tune the controllers according to. It takes the variables from the properties of applied relays, calculates the amplitudes of the relays , which will be used in the following controller tuning steps. On the screen, following lines appear:

```
» tun2c
Simulink model must be open.

To tune the system, write the name of the file as a string: 'dene19'

Choose 1 to tune according Ziegler-Nichols and modified ZN PID rules,
Choose 2 to tune according Ziegler-Nichols and modified ZN PI rules,
Choose 3 to tune according Shen and Yu and modified SY PI rules,

Which one? 3


Amplitude of the relay 1 is 1
Amplitude of the relay 2 is 1

     1

Run simulation in pause mode.
Press enter when done.
```

Following the instructions given in the program, we obtain the following pictures (Figure 3.3).

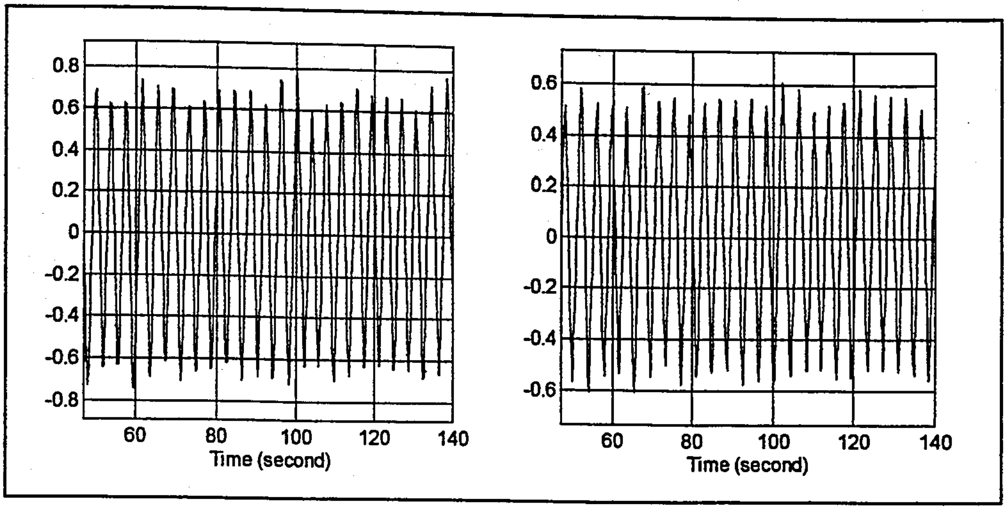

FIGURE 3.3. Time responses of the first and second loop when the first relay is applied to find the speed of the first loop

On the screen, the following lines and the responses (Figure 3.4) appear:

```
2

Run simulation in pause mode.
Press enter when done.
```

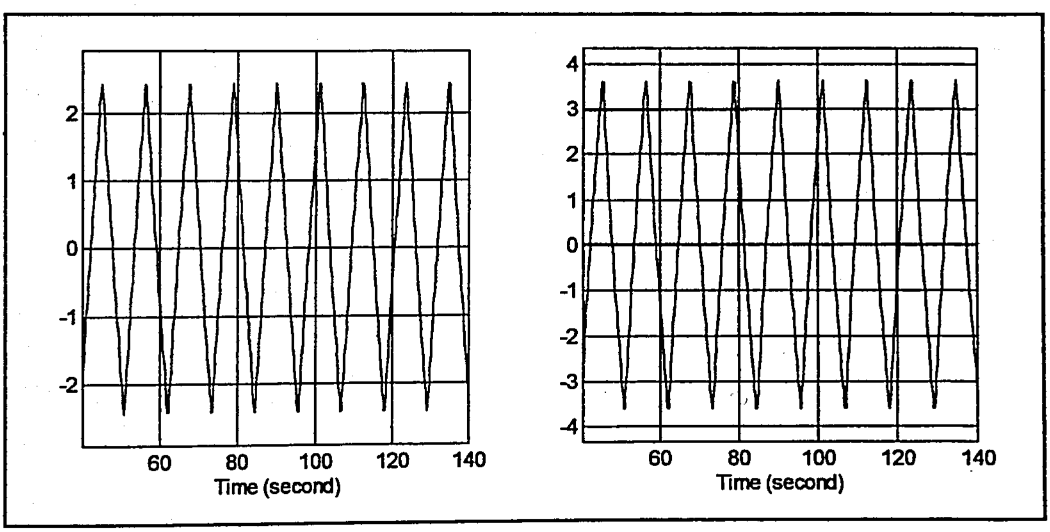

FIGURE 3.4. Time responses of the first and second loop when the second relay is applied to find the speed of the second loop

Loop speeds are found, tuning sequence is obtained. On the screen, the following lines and the responses (Figure 3.5) appear:

```
Frequency of the loop 1 is 1.698 rad/s
Frequency of the loop 2 is 0.561 rad/s

Tuning loop 1 first !!!


Run simulation in pause mode before tuning...
Then strike any key when done.
```

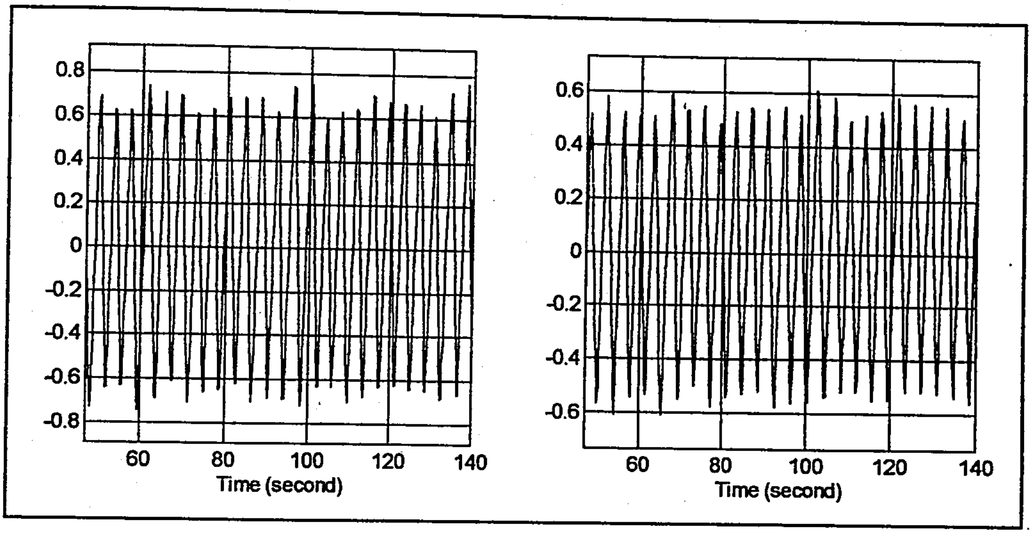

FIGURE 3.5. Time responses of the first and second loop when the first relay is applied to tune the controller on the fast loop

On the screen, the following lines and the responses (Figure 3.6) appear:

```
Tuning loop with number 1

PID-1 parameters are: Proportional... 0.6693
                      Integral....... 0.09045
                      Derivative..... 0

Enter a detuning factor (1 or greater than 1) 1

Detuned settings
PID-1 parameters are: Proportional... 0.6693
                      Integral....... 0.09045
                      Derivative..... 0
Press any key...

Run simulation in pause mode before tuning...
Then strike any key when done.
```

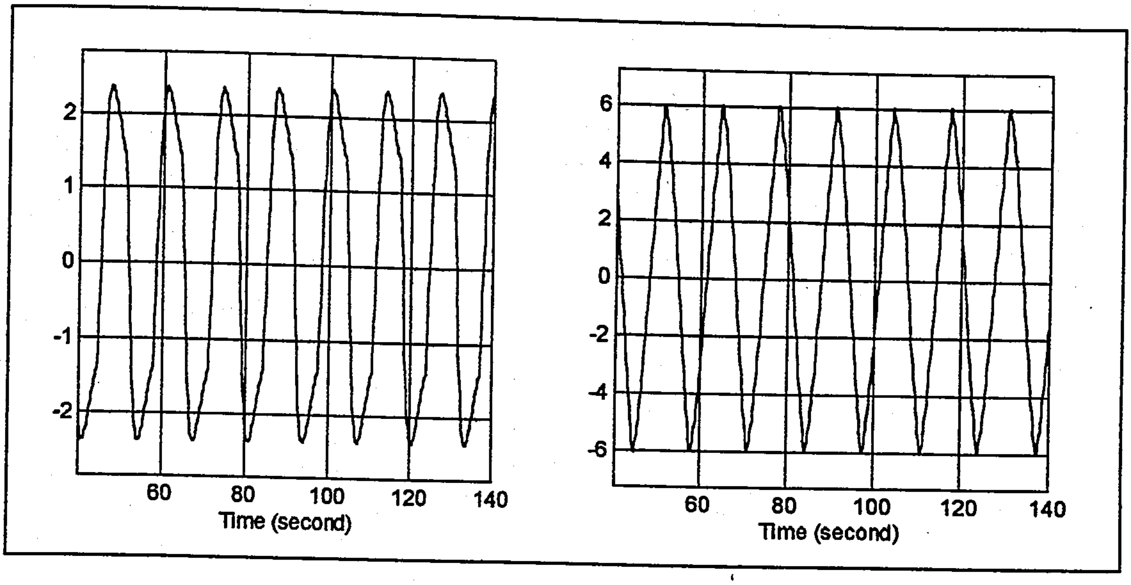

FIGURE 3.6. Time responses of the first and second loop when the second relay is applied to tune the controller of the slow loop, while the controller on the fast loop is on line

On the screen, the following lines appear:

```
Tuning loop with number 2

PID-2 parameters are: Proportional... -0.07036
                      Integral....... -0.002665
                      Derivative..... 0

Enter a detuning factor (1 or greater than 1) 1

Detuned settings
PID-2 parameters are: Proportional... -0.07036
                      Integral....... -0.002665
                      Derivative..... 0
Press any key...

Now you can simulate in pause mode,
Enter setpoint values and simulate,
Press any key when you are finished with looking at the graphs...
```

Following the instructions given by the program, setpoint values are given, simulation is run and the responses are observed. This is used to assure that the responses are "good enough" to operate the system. This is the manual part of the program. One can let the program run further until all controller parameters converge or one can stop the program if he decides that the responses are satisfactory.

Figure 3.7 shows the responses with respect to the reference values, both set to 1. If the responses are not satisfactory, the program continues running.

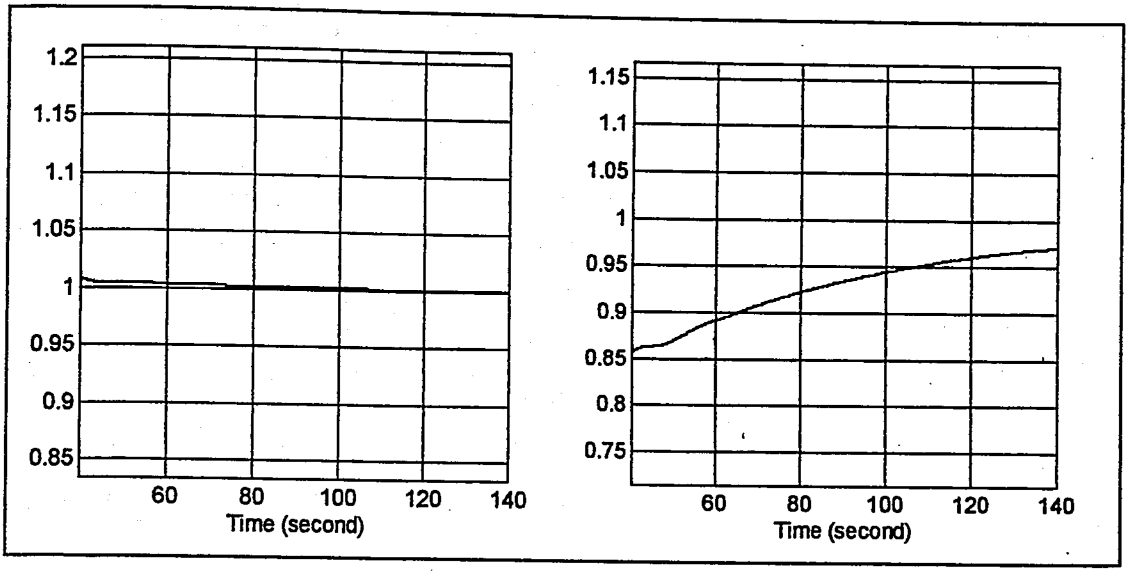

FIGURE 3.7. Time responses, when both references are set to 1, after the first run

The program waits for an entry:

```
Do you want to tune again? (1:yes, 0:no) 1
```

The autotuning procedure is repeated and the following results are finally obtained:

```
PID-1 parameters are: Proportional... 0.6378
                      Integral....... 0.08392
                      Derivative..... 0

Enter a detuning factor (1 or greater than 1) 1

Detuned settings
PID-1 parameters are: Proportional... 0.6378
                      Integral....... 0.08392
                      Derivative..... 0


PID-2 parameters are: Proportional... -0.07148
                      Integral....... -0.002707
                      Derivative..... 0

Enter a detuning factor (1 or greater than 1) 1

Detuned settings
PID-2 parameters are: Proportional... -0.07148
                      Integral....... -0.002707
                      Derivative..... 0
```

The parameters converge on the second tuning run. The tuning and the μ-analysis programs are given in Appendix A.

3.4. Common Problems in Autotuning Programs

Some problems may occur in autotuning programs. The most common ones are:

- If the plant has strong interactions, especially if the interacting components have different signs in gains, there may be a possibility to tune the controllers incorrectly. Then, the signs can be entered manually.

- If the loop speeds are very different from each other (e.g. 100 times), there may be a possibility to obtain normal setpoint responses on one loop and sluggish setpoint responses on the other loop. This is not very common in real life, but possible in theoretical trials of the program.

- If the plant has strong interactions and the interacting components of the transfer functions have extremely different time delays, there is a possibility to obtain "second bumps" in the responses when relay feedback is applied. "Second bumps" are sinusoidal peaks which occur right after the peak of the conventional sinusoidal response, so the sinusoidal responses appear to have double peaks, which leads the program to different settings. This may cause some problems in finding the ultimate period of the loop. Our program defeats such problems if the difference of the amplitudes between the first sinusoid and the corresponding "bump" does not exceed five per cent.

Tests show that there do not exist any problems in the examples used in our study.

4. RESULTS AND COMPARISON

In this chapter, the simulation results are investigated and are compared with each other. Our aim is to give an idea on how and which autotuning method is appropriate for the system needs. How "good" are the simulation results for various reference values with various systems tuned with various methods? So, the frequency responses should also be examined.

Five systems are thoroughly investigated. Some of the methods do not work under some unwanted conditions. Systems which are controllable with respect to the tuning criteria should be chosen and some assumptions on the average properties of these systems have to be made to simplify the tuning process.

4.1. System Characteristics and Properties

Because we are interested in chemical and process control systems, the transfer functions of those systems generally take the form:

$$G(s) = \frac{Ke^{-Ds}}{T_p s + 1} \tag{4.1}$$

where K is the gain, D is the time delay, T_p is the time constant of the plant; generally type of a first-order-plus-dead time transfer functions.

4.2. General Assumptions

In order to obtain some results or to begin with the tuning procedure, some assumptions are made about the process. These assumptions are:

- The process is open loop stable.

• Interactions are significant. If the process is decoupled or weakly coupled then a multiloop autotuner is not needed. So, the decentralized auto-tuner and controller put into the system, is capable of handling controllability.

• The process has low-pass characteristics. Fortunately, most processes in the chemical and process industries satisfy this assumption.

4.3. Plants and Applied Tuning Methods

The systems investigated in this study are shown in Table 4.1.

TABLE 4.1. Investigated plants and their descriptions

Name of the plant	Description
Wood and Berry Distillation Column (Wood and Berry, 1973)	2x2, chemical process model
Ogunnaike and Ray Distillation Column (Ogunnaike and Ray, 1979)	3x3, chemical process model
Doukas and Luyben Distillation Column (Doukas and Luyben, 1978)	4x4, chemical process model
Solid Fuel Boiler Plant (Johansson et al., 1984)	2x2, a system with small interactions
Quadruple Tank (min. phase) (Johansson et al., 1997)	2x2, an experimental system, set up by K. H. Johansson

Here are the transfer matrices for Wood and Berry Distillation Column:

$$G(s)=\begin{bmatrix} \frac{12.8e^{-s}}{1+16.7s} & \frac{-18.9e^{-3s}}{1+21s} \\ \frac{6.6e^{-7s}}{1+10.9s} & \frac{-19.4e^{-3s}}{1+14.4s} \end{bmatrix} \tag{4.2}$$

and for Solid Fuel Boiler Plant:

$$G(s)=\begin{bmatrix} \frac{-e^{-2s}}{1+10s} & \frac{-1}{1+10s} \\ 0 & \frac{e^{-10s}}{1+60s} \end{bmatrix} \tag{4.3}$$

for Johansson's Tank with minimum phase:

$$G(s)=\begin{bmatrix} \dfrac{2.6}{1+62s} & \dfrac{1.5}{(1+23s)(1+62s)} \\ \dfrac{1.4}{(1+30s)(1+90s)} & \dfrac{2.8}{1+90s} \end{bmatrix} \tag{4.4}$$

for Ogunnaike and Ray Distillation Column:

$$G(s)=\begin{bmatrix} \dfrac{0.66e^{-2.6s}}{6.7s+1} & \dfrac{-0.61e^{-3.5s}}{8.64s+1} & \dfrac{-0.0049e^{-s}}{9.06s+1} \\ \dfrac{1.11e^{-6.5s}}{3.25s+1} & \dfrac{-2.36e^{-3s}}{5s+1} & \dfrac{-0.01e^{-1.2s}}{7.09s+1} \\ \dfrac{-34.68e^{-9.2s}}{8.15s+1} & \dfrac{46.2e^{-9.4s}}{10.9s+1} & \dfrac{0.87(11.61s+1)e^{-s}}{(3.89s+1)(18.8s+1)} \end{bmatrix} \tag{4.5}$$

and for Doukas and Luyben Distillation Column:

$$G(s)=\begin{bmatrix} \dfrac{-9.811e^{-1.59s}}{11.36s+1} & \dfrac{0.374e^{-7.75s}}{22.22s+1} & \dfrac{-2.368e^{-27.33s}}{33.3s+1} & \dfrac{-11.3e^{-3.79s}}{(21.74s+1)^2} \\ \dfrac{5.984e^{-2.24s}}{14.29s+1} & \dfrac{-1.986e^{-0.71s}}{66.67s+1} & \dfrac{0.422e^{-8.72s}}{(250s+1)^2} & \dfrac{5.24e^{-60s}}{400s+1} \\ \dfrac{2.38e^{-0.42s}}{(1.43s+1)^2} & \dfrac{0.0204e^{-0.59s}}{(7.14s+1)^2} & \dfrac{0.513e^{-s}}{s+1} & \dfrac{-0.33e^{-0.68s}}{(2.38s+1)^2} \\ \dfrac{-11.3e^{-3.78s}}{(21.74s+1)^2} & \dfrac{-0.176e^{-0.48s}}{(6.9s+1)^2} & \dfrac{15.54e^{-s}}{s+1} & \dfrac{4.48e^{-0.52s}}{11.11s+1} \end{bmatrix} \tag{4.6}$$

The methods according to which controller parameters are set are in Table 4.2. The settings are shown in Tables 4.3 to 4.7.

4.4. Convergence Criteria

Sequential tuning methods are applied to all possible settings except μ-optimal settings. To obtain acceptable settings, sequential tuning must be applied more than once. The faster the settings are obtained, the better the method will be. The autotuning convergence criterion is to obtain reasonable settings. These settings should not vary more than three per cent in the next tuning run. Some control parameters, for various systems which have the same number of inputs and outputs, can converge at different number of trials. The three per cent criterion is not always met in experiments performed, sometimes the error in the tuning is slightly more than three per cent, and it is nearly equal to zero. It depends on the computational effort and number of trials (see Tables 4.8 to 4.13).

TABLE 4.2. Applied tuning methods and their descriptions

Name of the tuning method	Description
Ziegler-Nichols method (Z-N)	used generally for overdamped SISO systems
Luyben empirical	settings obtained by trial-and-error by Luyben; show reasonable results
BLT	biggest log modulus tuning
BLT4, BLT6, BLT8	biggest log modulus tuning, where the number adjacent to the method presents the amount of positive decibels for the corresponding system
Shen-Yu	settings obtained by modifying the Z-N controller setting according to rules described in the article of Shen and Yu.
μ-optimal	settings obtained by minimizing the structured singular value and keeping it close to 1 according to some uncertainty assumption. Transfer functions are used to simulate input uncertainties and performance weights.

The three per cent error criterion cannot be applied to μ-optimal tuning because of this method considers the input uncertainties acting on the system. In this type of settings, beginning from the initial settings ending in the final settings, the number of trials vary from minimum 200 for 2x2 systems to minimum 800 for 4x4 systems. Remembering that our aim is to set the controller parameters according to μ slightly smaller than 1 while considering the frequency responses of the example systems, μ-optimal tuning method needs very long computation. Each run of the program takes at least two hours on the Pentium 166 MHz based personal computer for a 2x2 system model. So, μ-optimal settings can be used to perform stability check and/or can be applied once they are obtained, but μ-optimal tuning method is not easily applicable in autotuning processes.

TABLE 4.3. Controller settings for Wood and Berry (WB) Distillation Column

	Luy.Emp.	Z-N	BLT	Shen-Yu	μ-optimal	μ-optimal-2
K_{C1}	0,2	0,96	0,375	0,6687	0,0473	0,5026
K_{C2}	-0,04	-0,19	-0,075	-0,07012	-0,0697	-0,0922
T_{I1}	4,44	3,25	8,29	7,64	6,38	41,34
T_{I2}	2,67	9,2	23,6	26,38	13,61	19,87

TABLE 4.4. Controller settings for Solid Fuel Boiler Plant (SFBP)

	Z-N	BLT	Shen-Yu	μ-optimal
K_{C1}	-3,68	-3,84937	-2,726	-1,7652
K_{C2}	4,041	4,226987	2,994	1,4964
T_{I1}	6,67	6,37	16,00	11,75
T_{I2}	30,00	28,68	72,01	50,97

TABLE 4.5. Controller settings for Johansson's Quadruple Tank Model (4 Tank)

	Z-N	BLT	Shen-Yu	μ-optimal
K_{C1}	27,40	9,1333	26,86	7,6897
K_{C2}	36,26	12,0867	20,27	9,5701
T_{I1}	1,67	5,01	4,00	10,8917
T_{I2}	1,67	5,01	4,00	20,0382

TABLE 4.6. Controller settings for Ogunnaike and Ray (OR) Distillation Column

	Luy.Emp.	Z-N	BLT 4	BLT 6	Shen-Yu	μ-optimal
K_{C1}	1,2	3,24	1,28	1,51	2,758	0,3353
K_{C2}	-0,15	-0,63	-0,251	-0,295	-0,2317	-0,3145
K_{C3}	0,6	5,66	2,24	2,63	3,907	4,805
T_{I1}	5	7,62	19,3	16,4	18,003	3,950
T_{I2}	10	8,36	21,1	18	25,408	11,829
T_{I3}	4	3,08	7,78	6,61	8,000	3,664

TABLE 4.7. Controller settings for Doukas and Luyben (DL) Distillation Column

	Luy.Emp.	Z-N	BLT 4	BLT 8	Shen-Yu	μ-optimal
K_{C1}	-0,12	-0,55	-0,084	-0,118	-0,261	-0,2994
K_{C2}	-15	-33,9	-5,16	-7,26	-20,19	-20,2028
K_{C3}	0,89	2	0,305	0,429	0,1733	0,1067
K_{C4}	0,2	3,47	0,529	0,743	2,1	2,446
T_{I1}	7,9	5,03	33	23,5	12,92	12,93
T_{I2}	30	2,36	15,5	11	5,72	5,49
T_{I3}	20	2,58	17	12,1	14,99	14,96
T_{I4}	20	1,7	11,2	7,94	4,20	5,96

TABLE 4.8. Convergence results for WB with Shen-Yu (SY) settings

Number of Trials	PI 1 K_{c1}	PI 2 K_{c2}	PI 1 T_{i1}	PI 2 T_{i2}
1	0,6651	-0,07067	7,39	26,33
2	0,6459	0,07229	7,49	26,33
3	0,6687	-0,07012	7,64	26,38

TABLE 4.9. Convergence results for SFBP with Shen-Yu settings

Number of Trials	PI 1 K_{c1}	PI 2 K_{c2}	PI 1 T_{i1}	PI 2 T_{i2}
1	-3,68	4,041	6,67	30,00
2	-3,68	4,041	6,67	30,00

TABLE 4.10. Convergence results for SFBP with Z-N settings

Number of Trials	PI 1 K_{c1}	PI 2 K_{c2}	PI 1 T_{i1}	PI 2 T_{i2}
1	-2,726	2,994	16,00	72,01
2	-2,726	2,994	16,00	72,01

TABLE 4.11. Convergence results for 4-Tank with Shen-Yu settings

Number of Trials	PI 1 K_{c1}	PI 2 K_{c2}	PI 1 T_{i1}	PI 2 T_{i2}
1	27,4	36,27	1,66	1,65
2	27,36	36,26	1,67	1,67

TABLE 4.12. Convergence results for OR with Shen-Yu settings

Number of Trials	PI 1 K_{c1}	PI 2 K_{c2}	PI 3 K_{c3}	PI 1 T_{i1}	PI 2 T_{i2}	PI 3 T_{i3}
1	2,511	-0,4192	3,909	17,04	19,84	8,00
2	2,808	-0,2504	3,906	14,88	26,00	8,00
3	2,711	-0,2313	3,908	17,75	26,00	8,00
4	2,758	-0,2317	3,907	18,00	25,41	8,00

TABLE 4.13. Convergence results for DL with Shen-Yu settings

Number of Trials	PI 1 K_{c1}	PI 2 K_{c2}	PI 3 K_{c3}	PI 4 K_{c4}	PI 1 T_{i1}	PI 2 T_{i2}	PI 3 T_{i3}	PI 4 T_{i4}
1	-0,2443	-29,79	0,1794	2,889	12,00	4,80	14,20	3,40
2	-0,2566	-24,73	0,1779	2,319	11,60	5,40	14,44	4,40
3	-0,26	-19,39	0,1755	2,17	12,60	6,40	14,80	4,60
4	-0,261	-20,19	0,1733	2,1	12,92	5,72	14,99	4,20

4.5. Weights Used in the Structured Singular Value Analysis

The plants are perturbed with an multiplicative input uncertainty and the performance of the plants are measured at the output (see Fig. 2.17). The input uncertainty weights and the performance weights are given in Table 4.14.

TABLE 4.14. μ– analysis results, input uncertainties and performance weights

Name	Setting	μ of RP	μ of NS	destabilizing *: yes	μ of RS	Explanation (w_i: input uncertainties, w_p: performance weights)
WB	LuyEmp	1,9953	0,8092		1,2078	Tested with $w_i=(0,8s+0,2)/(0,5s+1)$ $w_p=(0,33s+0,02)/s$
	Z-N	-	-	*	-	
	BLT	1,1628	0,4887		0,7383	
	Shen-Yu	1,4263	0,7538		0,8768	
	μ-optimal	0,9368	0,2396		0,7368	
SFBP	Shen-Yu	1,7581	0,4401		1,4425	Tested with $w_i=(s+0,2)/(0,5s+1)$ $w_p=(0,5s+0,05)/s$
	Z-N	2,3488	0,8633		1,8657	
	Blt	2,5566	0,9625		1,9717	
	μ-optimal	2,0578	0,2511		1,8527	
4 Tank	Shen-Yu	1,2242	0,8005		0,5	Tested with $w_i=(s+0,2)/(0,5s+1)$ $w_p=(0,5s+0,05)/s$
	Z-N	-	-	*	-	
	μ-optimal	0,7824	0,3105		0,5	
	BLT	0,9232	0,4383		0,5061	
OR	LuyEmp	6,6345	0,0088		6,5666	Tested with $w_i=(0,205s+0,05)/(0,5s+1)$ $w_p=(0,12s+0,012)/s$
	Z-N	10,5231	0,5717		9,9547	
	BLT 4	2,8935	0,1038		2,7865	
	BLT 6	3,7982	0,1414		3,5489	
	Shen-Yu	6,978	0,2495		6,6137	
	μ-optimal	0,9359	0,2428		0,8853	
DL	LuyEmp	-	-	*	-	Tested with $w_i=(0,205s+0,05)/(0,1s+1)$ $w_p=(0,125s+0,0125)/s$
	Z-N	-	-	*	-	
	BLT 4	2,2452	0,2099		2,0377	
	BLT 8	1,9104	0,3244		1,6884	
	Shen-Yu	1,052	0,4146		0,5759	
	μ-optimal	0,6708	0,3583		0,5862	
	continued on the next page					

	continued from the previous page					
WB-2	LuyEmp	2,8061	0,9117		1,9224	Tested with $w_i=(s+0,2)/(0,5s+1)$ $w_p=(0,5s+0,05)/s$
	Z-N	-	-	*	-	
	BLT	2,053	0,5712		1,8456	
	Shen-Yu	2,3955	0,9316		2,1917	
	μ-optimal	2,0418	0,2519		1,6759	
SFBP	Shen-Yu	1,0994	0,3622		0,7876	Tested with $w_i=(0,8s+0,2)/(0,5s+1)$ $w_p=(0,33s+0,02)/s$
	Z-N	1,6833	0,708		1,0392	
	BLT	1,8417	0,7848		1,0963	
	μ-optimal	0,9551	0,212		0,7613	
4 Tank	Shen-Yu	1,3842	0,8005		0,68	Tested with $w_i=(s+0,2)/(0,5s+1)$ $w_p=(0,68s+0,15)/s$
	Z-N	-	-	*	-	
	μ-optimal	0,9871	0,3105		0,681	
	BLT	1,1624	0,4383		0,7308	
OR	LuyEmp	27,0158	0,2958		26,7457	Tested with $w_i=(s+0,2)/(0,5s+1)$ $w_p=(0,5s+0,05)/s$
	BLT 4	11,8308	0,4518		11,6105	
	BLT 6	15,3167	0,6294		14,0862	
	Shen-Yu	-	-	*	-	
DL	LuyEmp	9,907	0,6879		9,2594	Tested with $w_i=(s+0,2)/(0,5s+1)$ $w_p=(0,5s+0,05)/s$
	BLT 4	8,3785	0,8082		7,5641	
	BLT 8	5,2499	0,9753		4,5661	
	Shen-Yu	-	-	*	-	

4.6. Simulation Results

Time and frequency responses are given in the following pages. Time responses are generated by applying combinations of unit steps to reference values. Plants are controlled with various settings of decentralized PID controllers. Some plots are generated by adding uncertainties given in Table 4.14.

Biggest log modulus and singular value plots for our plants with different settings and robustness analysis plots for our plants with given uncertainties (Table 4.14) are generated in the frequency domain. These results are also in the following pages.

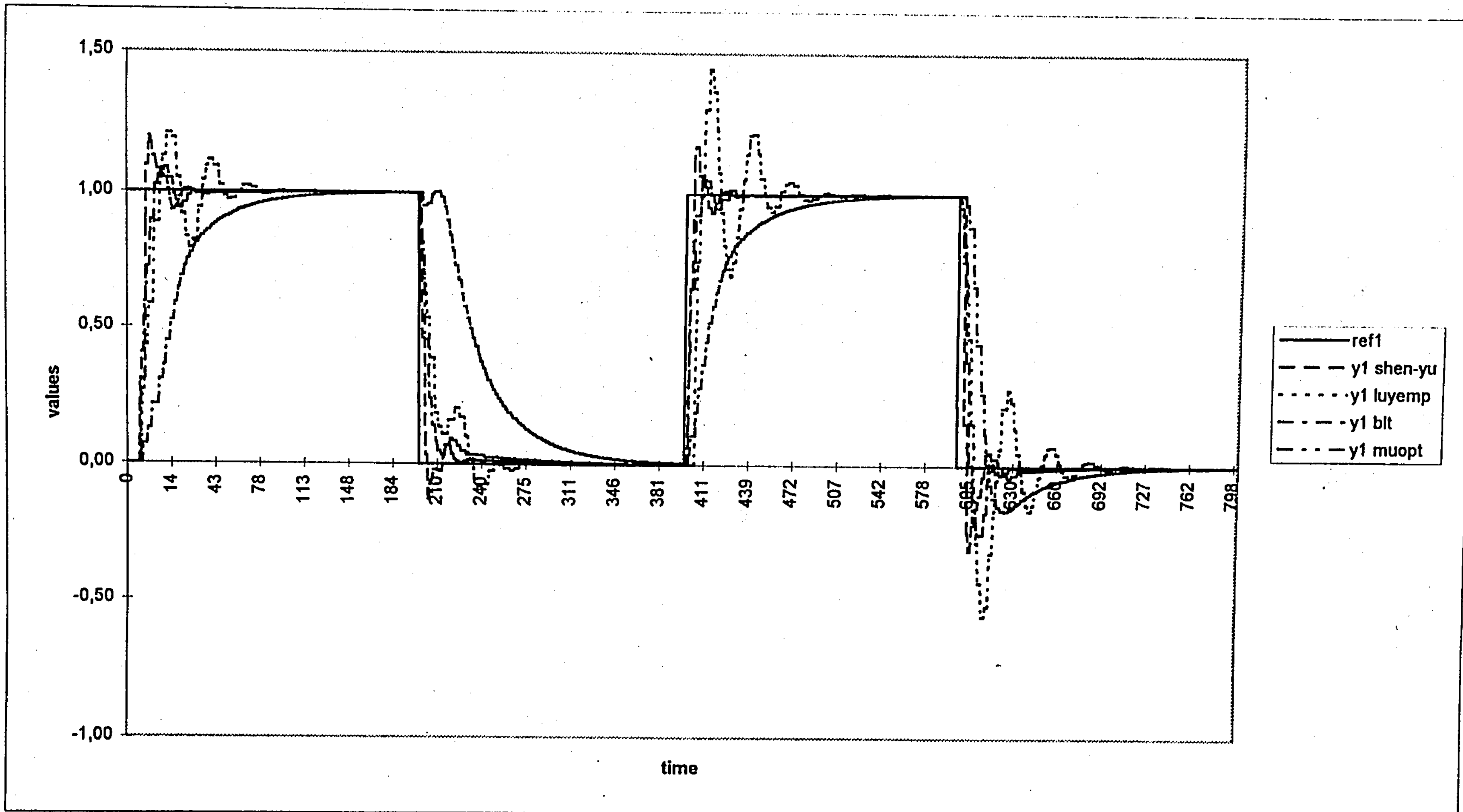

FIGURE 4.1. Wood and Berry Distillation Column, $y_1(t)$ according to various tuning methods

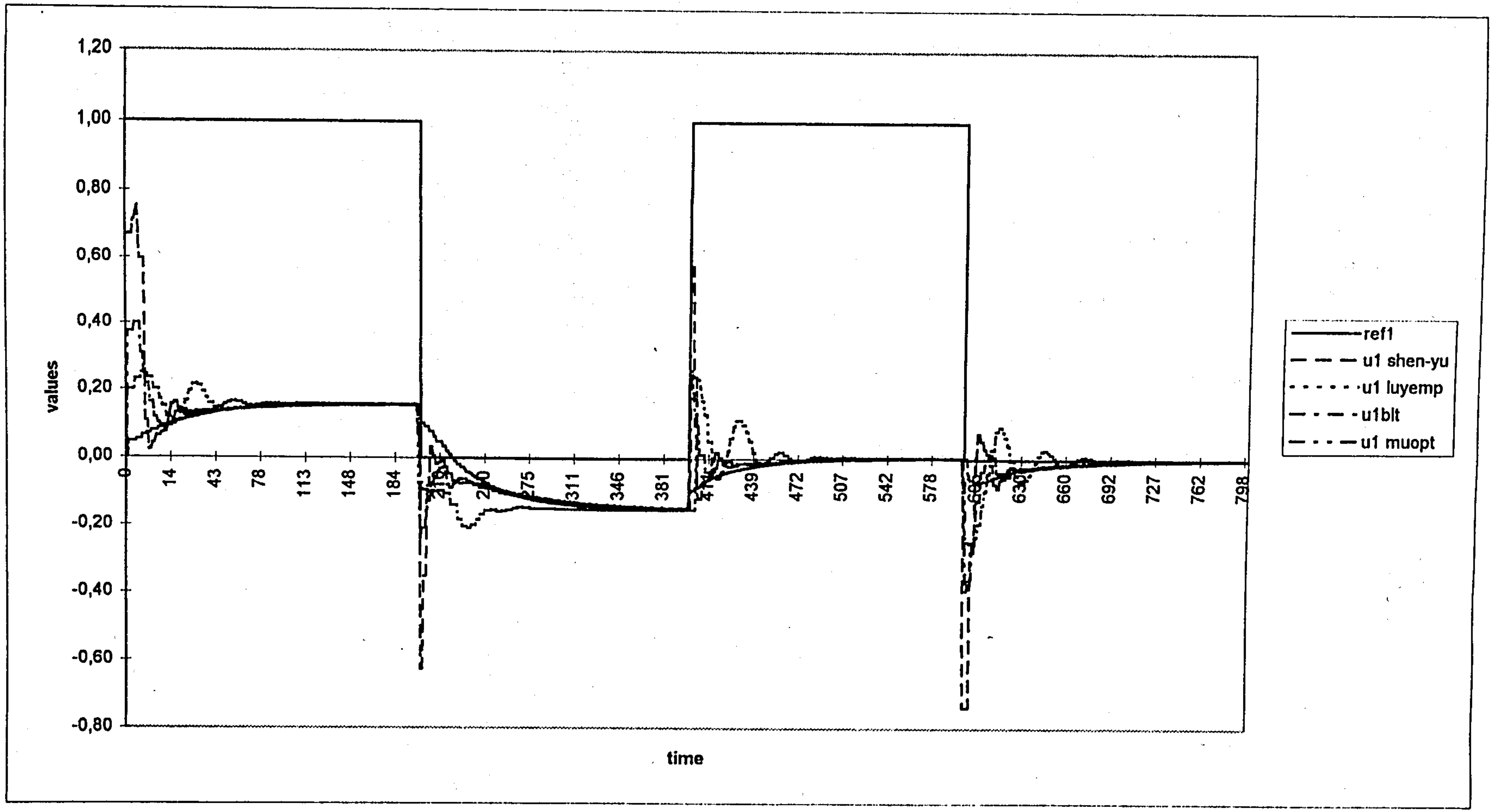

FIGURE 4.2. Wood and Berry Distillation Column, $u_l(t)$ according to various tuning methods

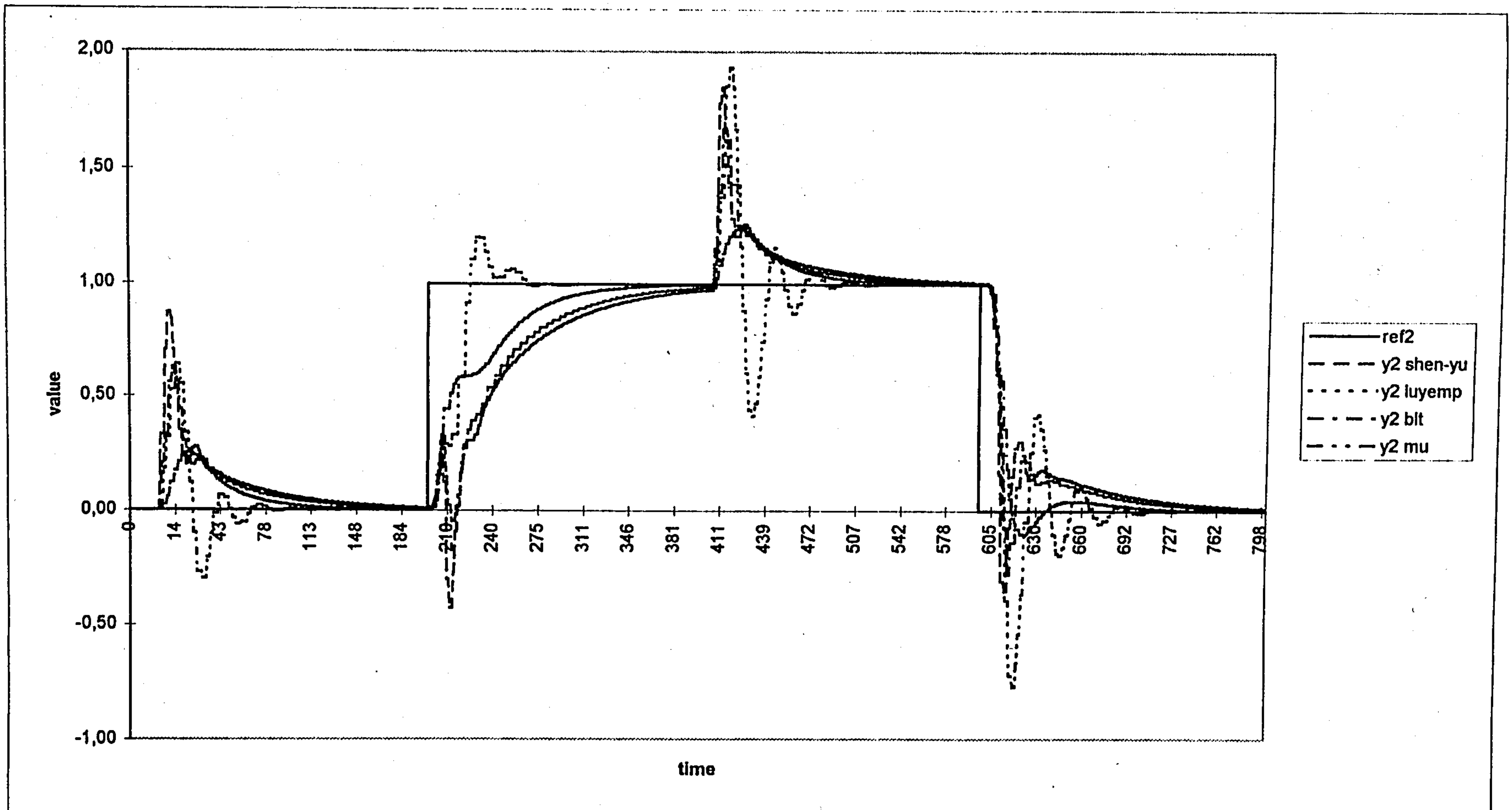

FIGURE 4.3. Wood and Berry Distillation Column, $y_2(t)$ according to various tuning methods

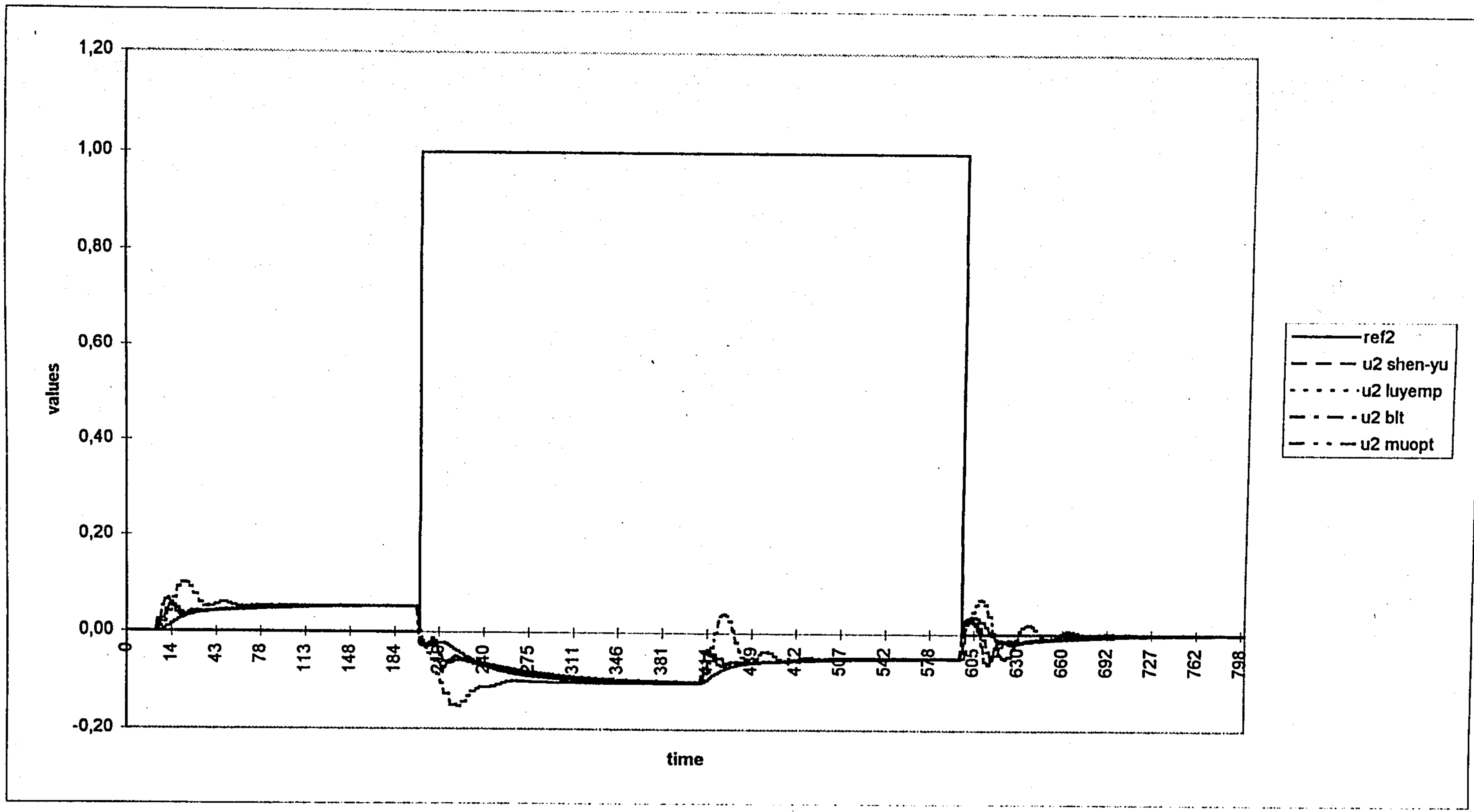

FIGURE 4.4. Wood and Berry Distillation Column, $u_2(t)$ according to various tuning methods

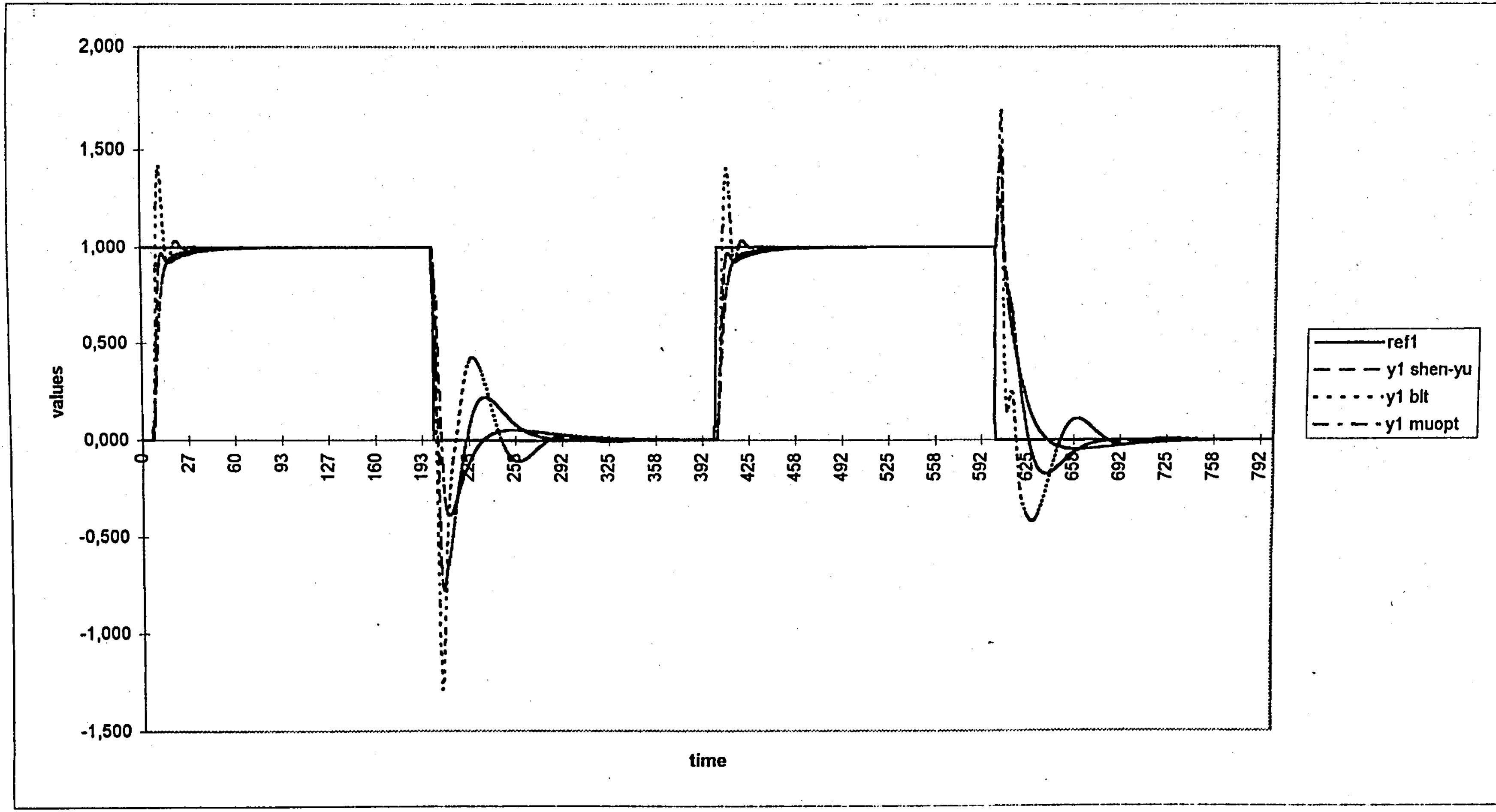

FIGURE 4.5. Solid Fuel Boiler Plant, $y_1(t)$ according to various tuning methods

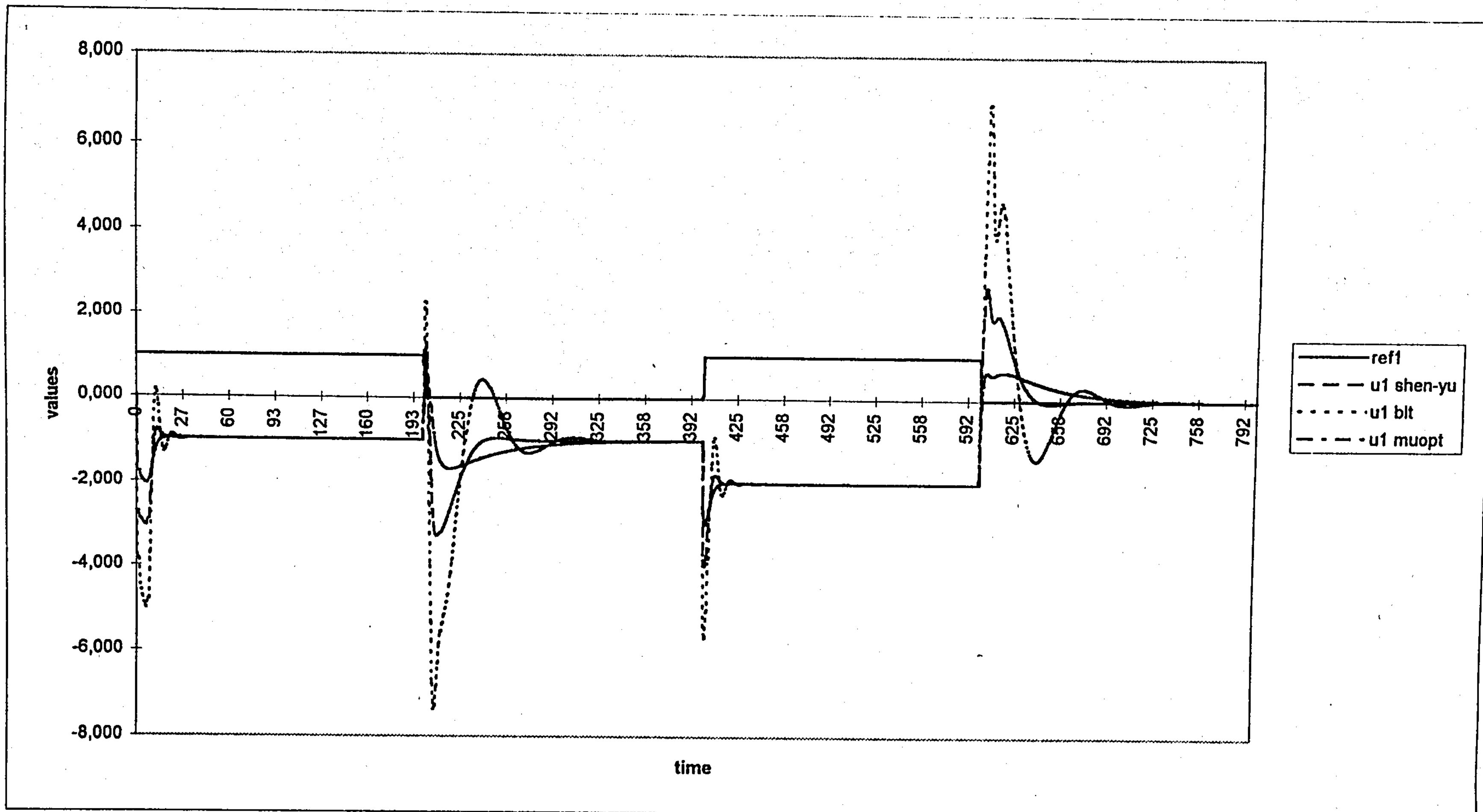

FIGURE 4.6. Solid Fuel Boiler Plant, $u_1(t)$ according to various tuning methods

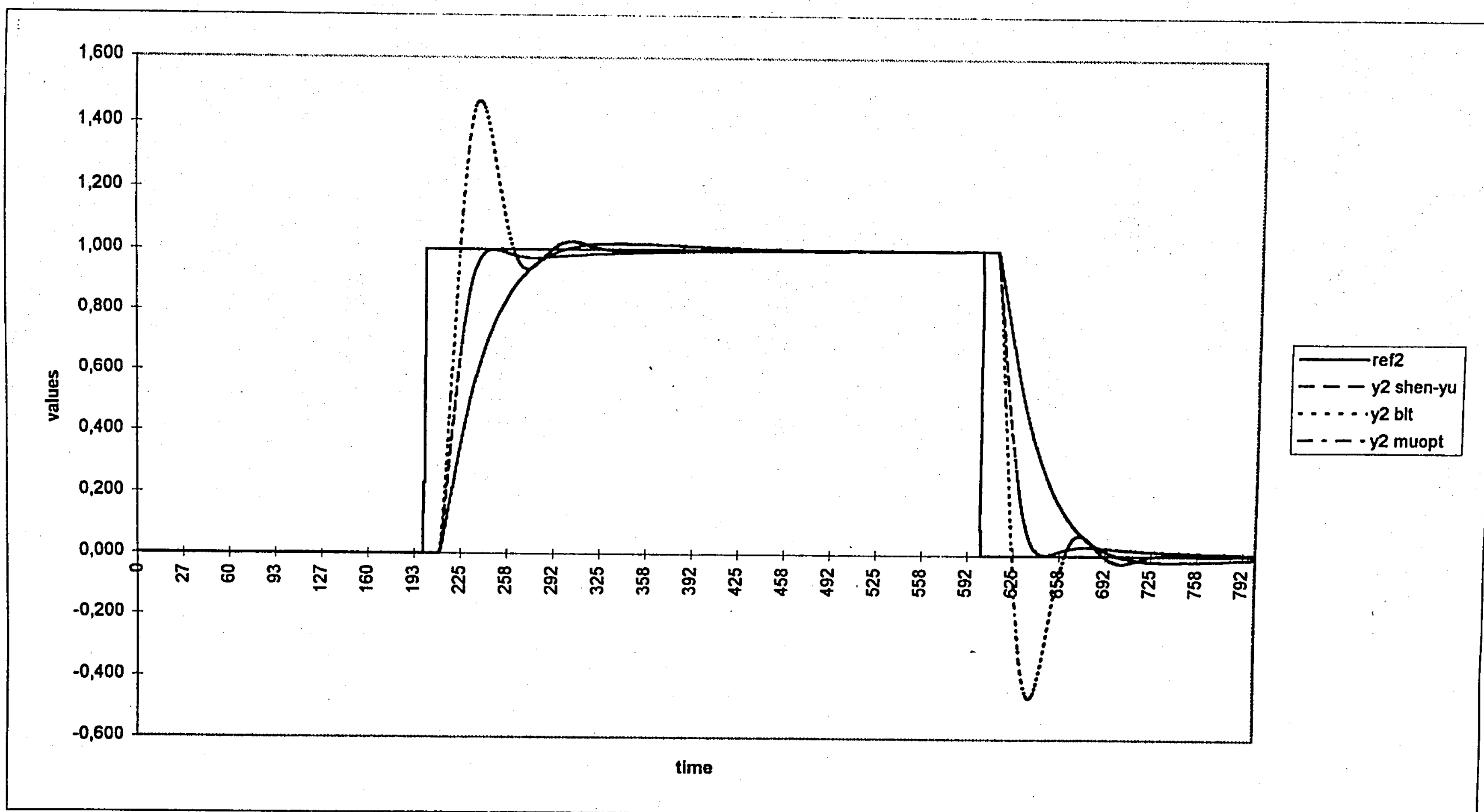

FIGURE 4.7. Solid Fuel Boiler Plant, $y_2(t)$ according to various tuning methods

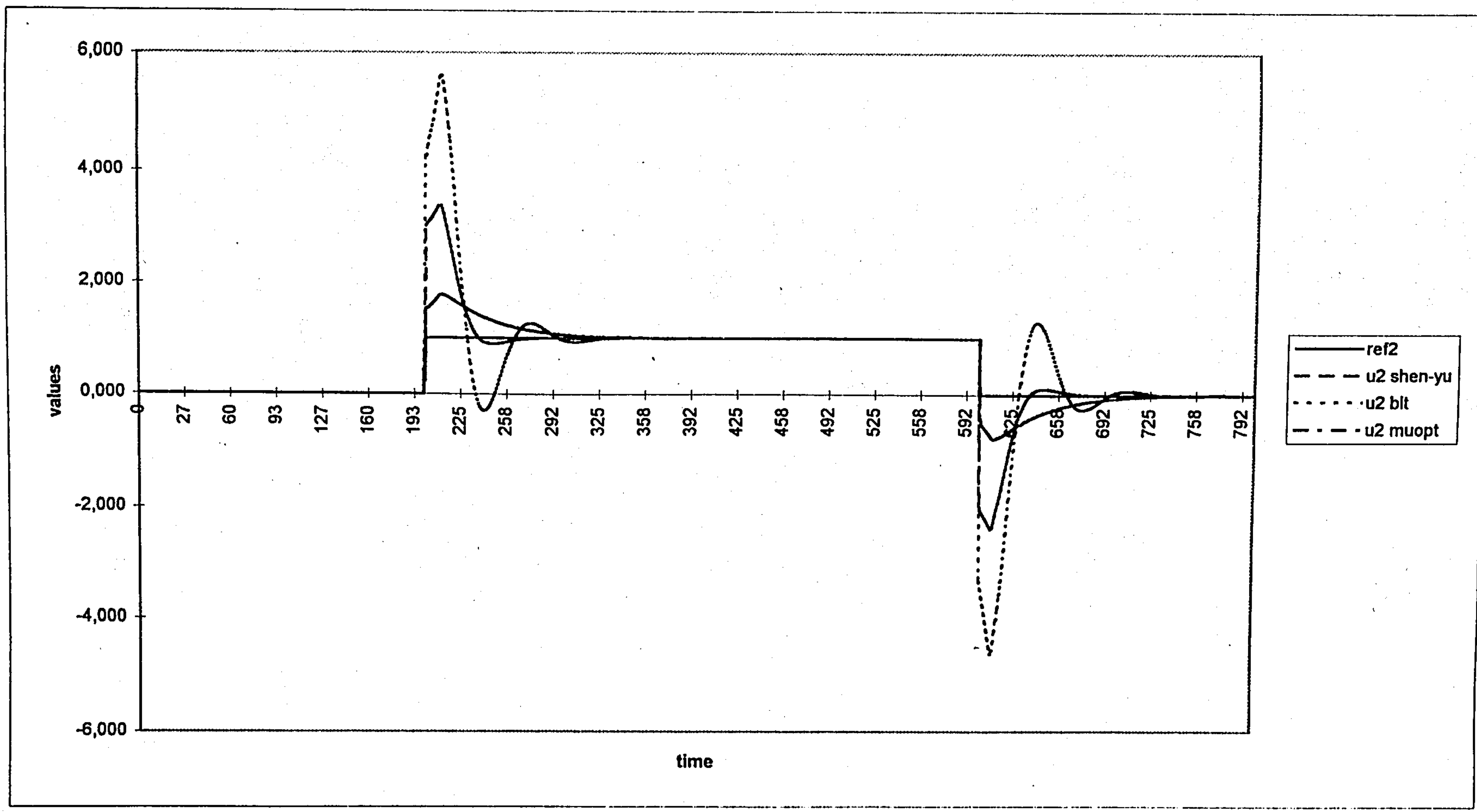

FIGURE 4.8. Solid Fuel Boiler Plant, $u_2(t)$ according to various tuning methods

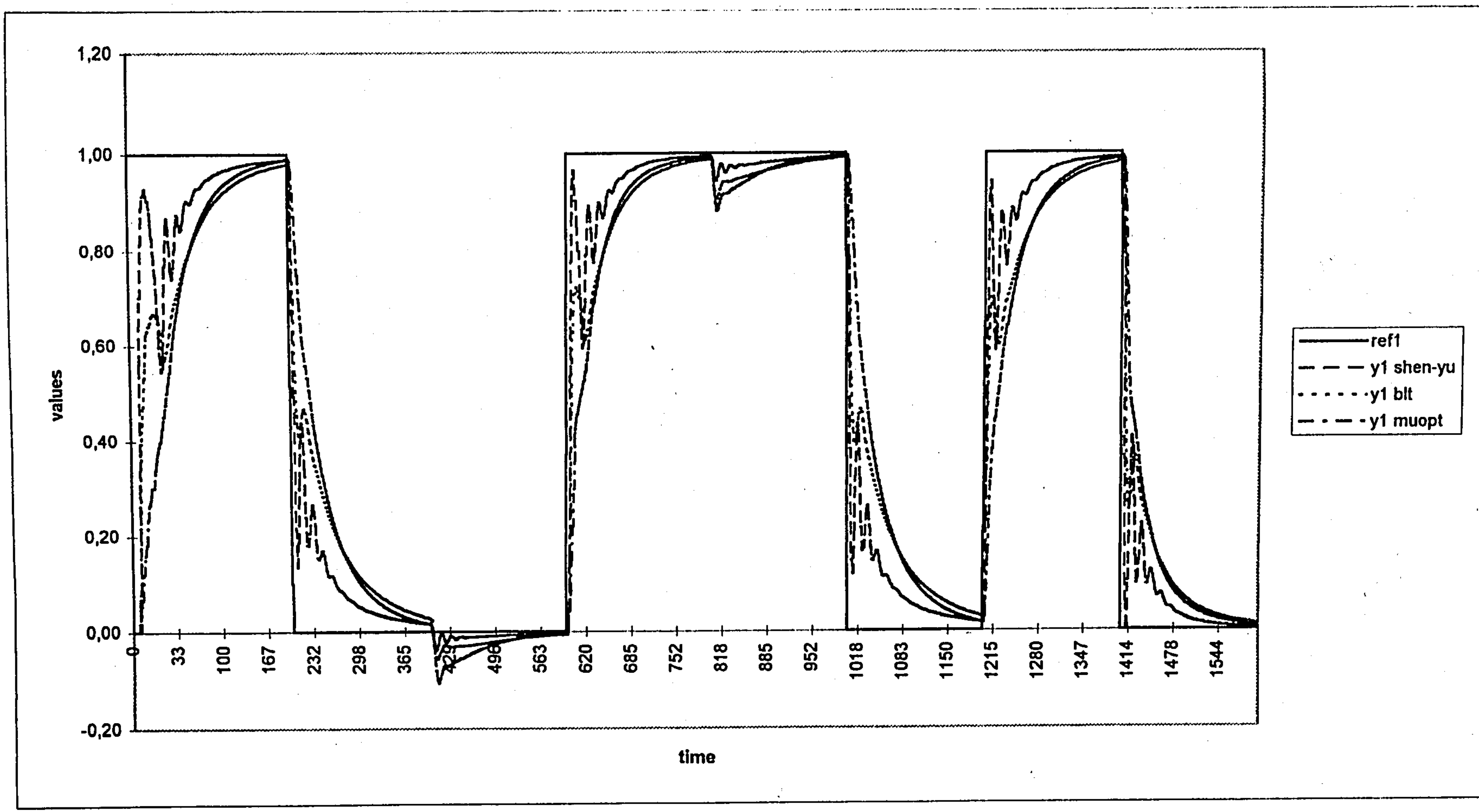

FIGURE 4.9. Ogunnaike and Ray Distillation Column, $y_1(t)$ according to various tuning methods

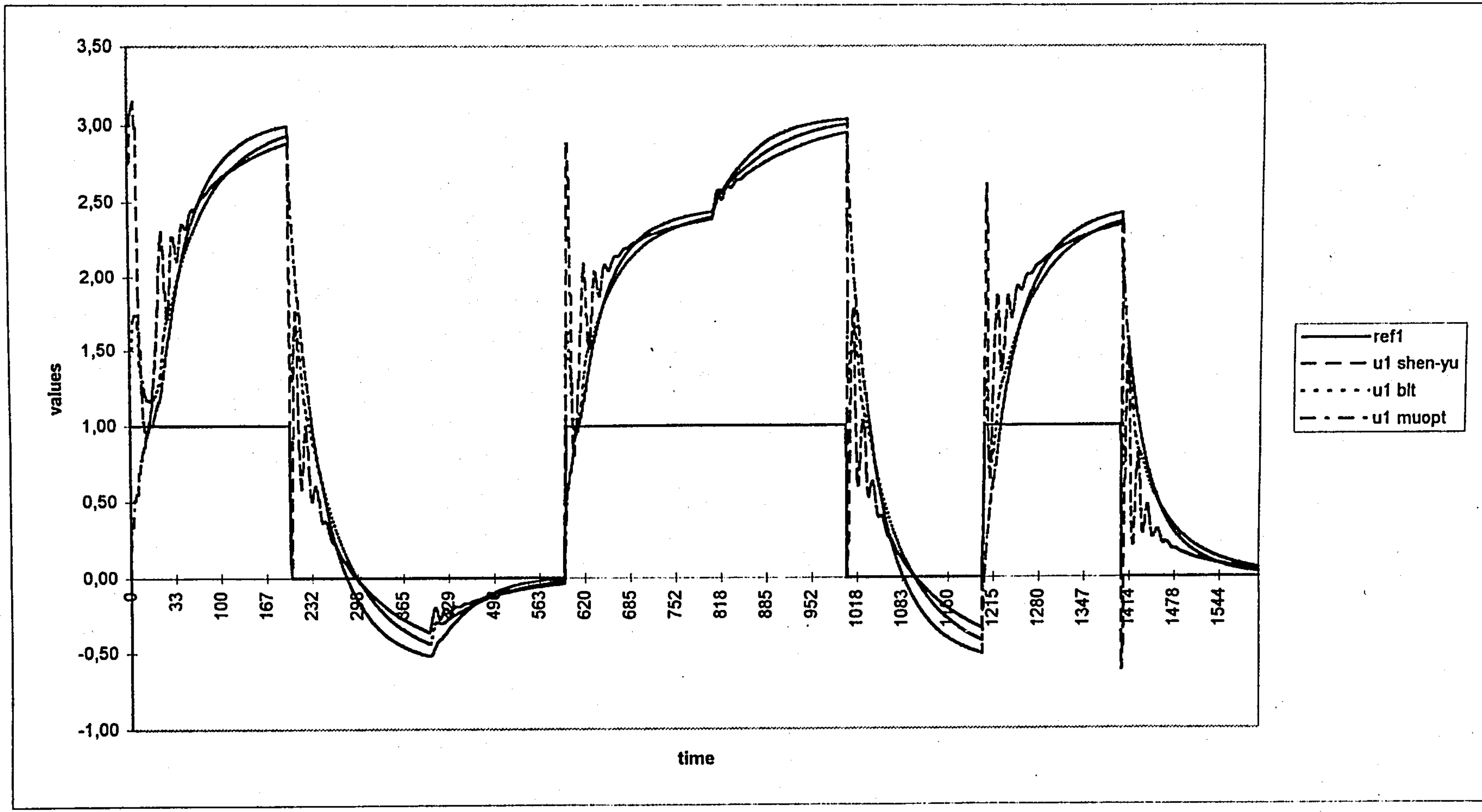

FIGURE 4.10. Ogunnaike and Ray Distillation Column, $u_1(t)$ according to various tuning methods

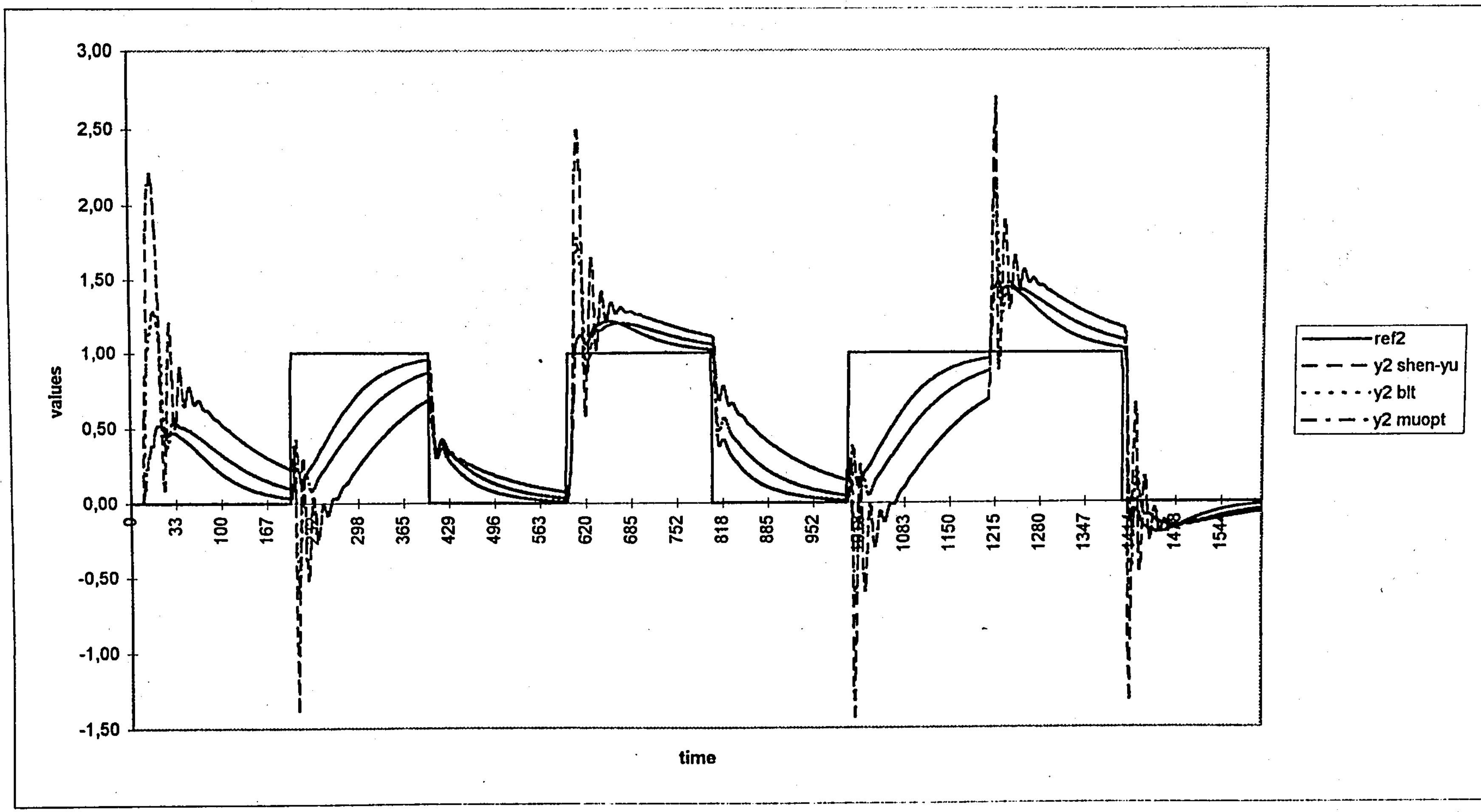

FIGURE 4.11. Ogunnaike and Ray Distillation Column, $y_2(t)$ according to various tuning methods

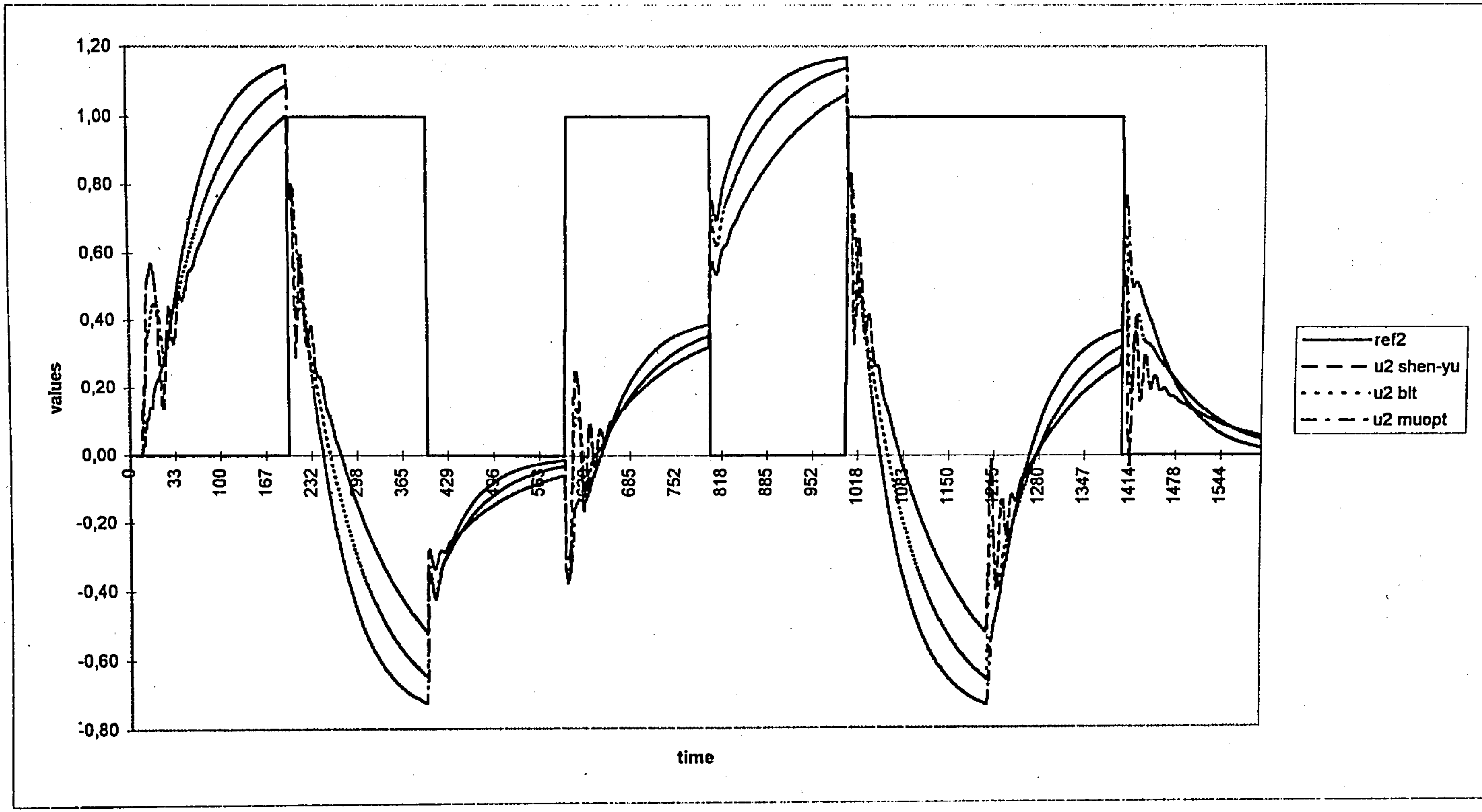

FIGURE 4.12. Ogunnaike and Ray Distillation Column, $u_2(t)$ according to various tuning methods

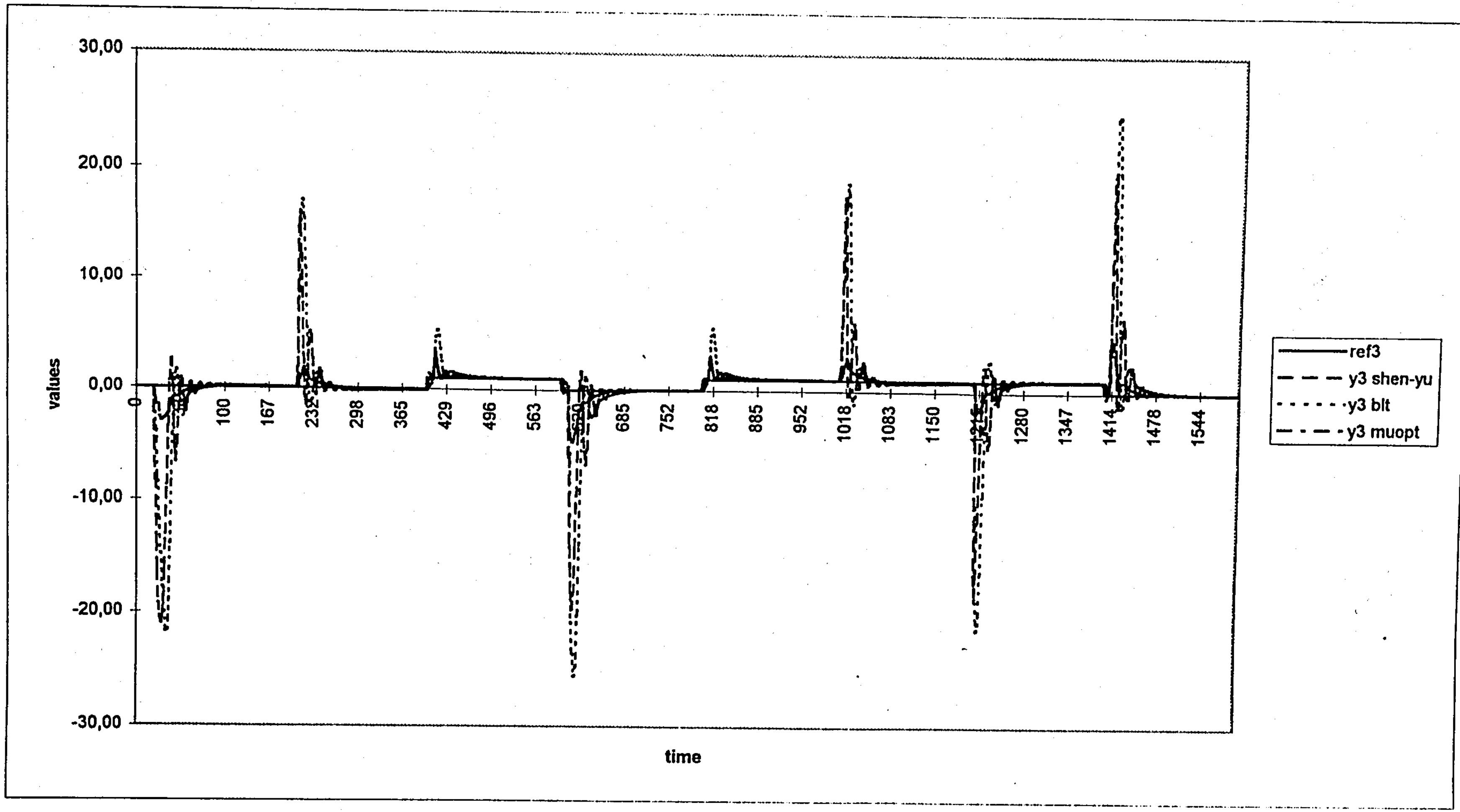

FIGURE 4.13. Ogunnaike and Ray Distillation Column, $y_3(t)$ according to various tuning methods

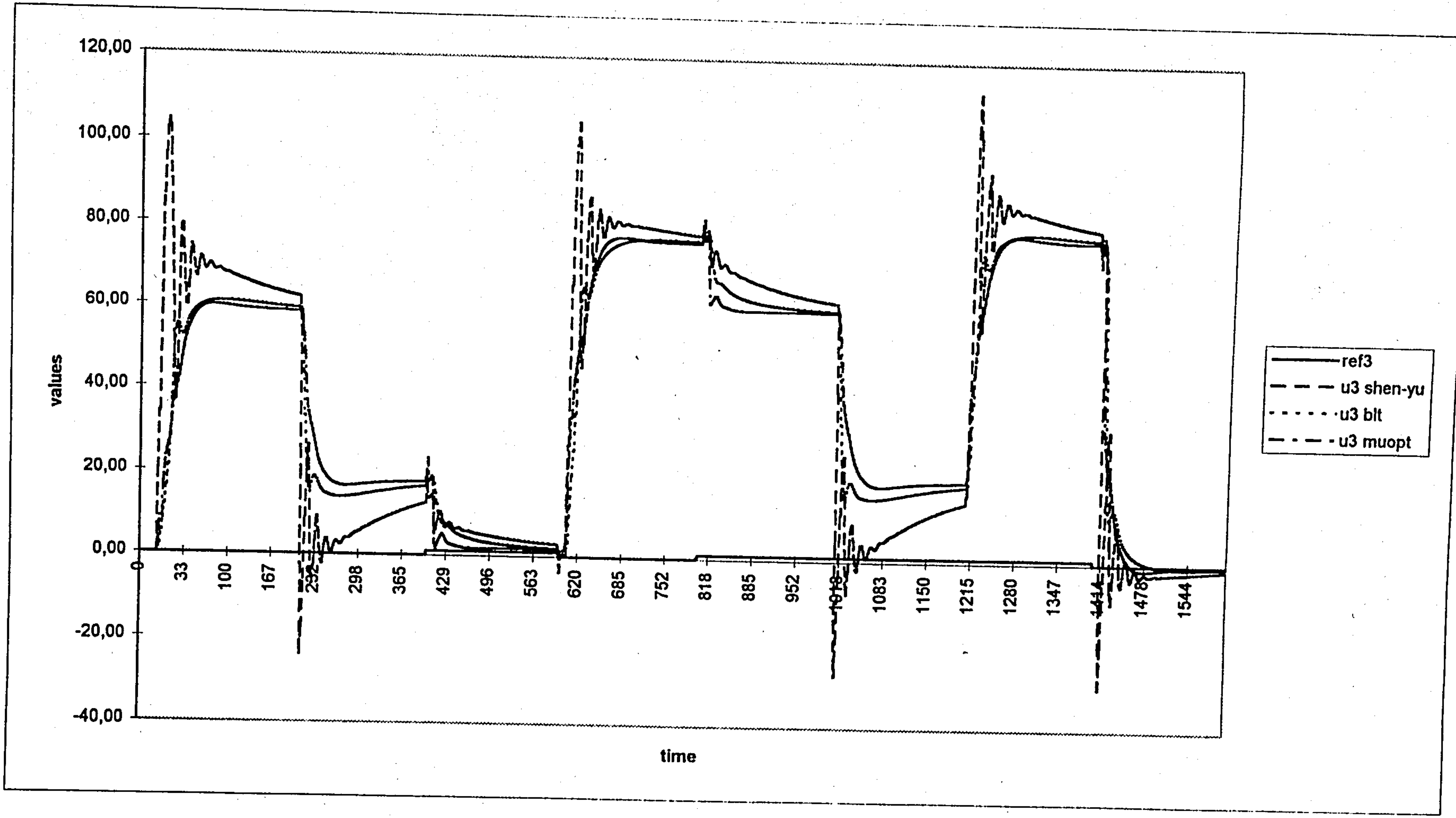

FIGURE 4.14. Ogunnaike and Ray Distillation Column, $u_3(t)$ according to various tuning methods

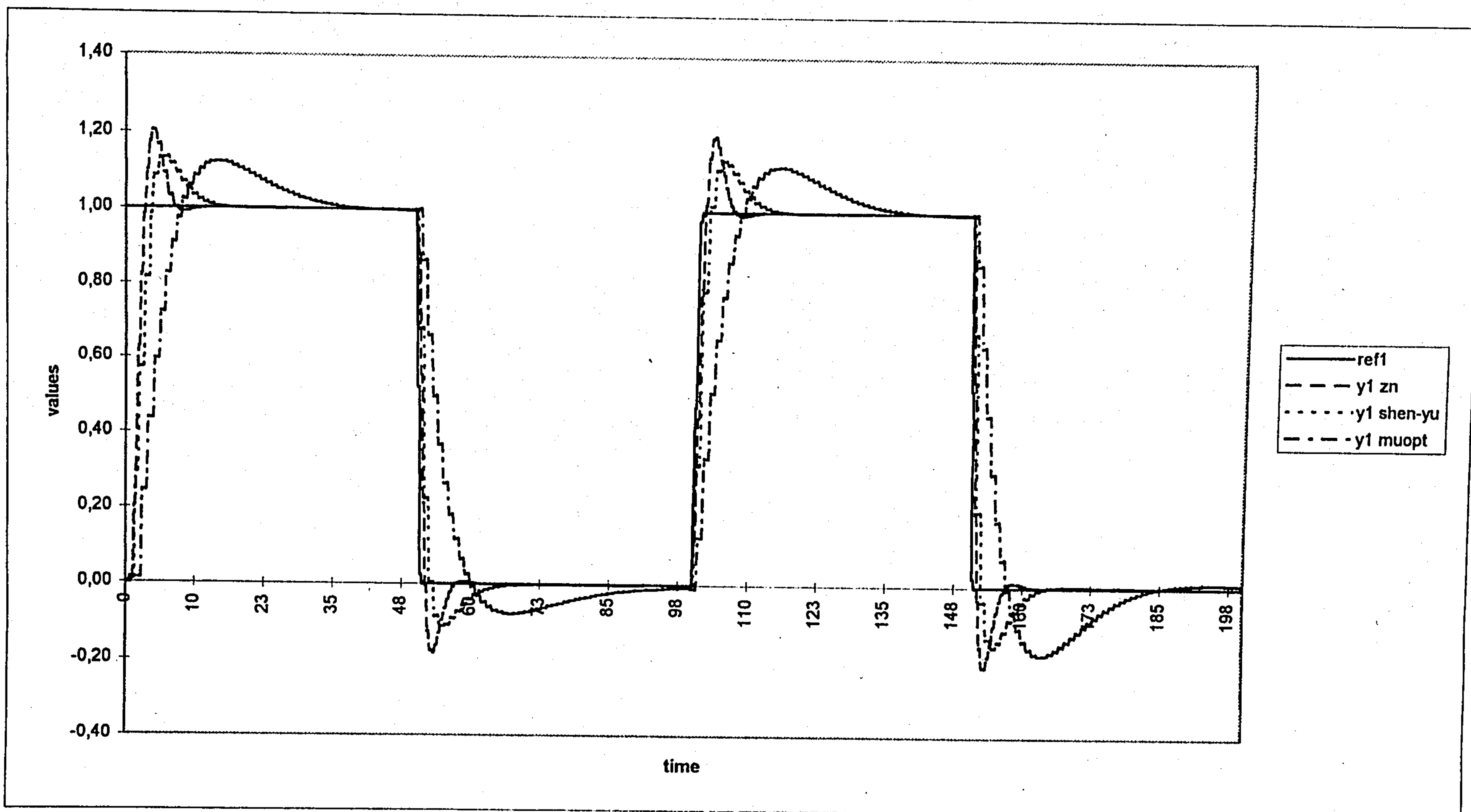

FIGURE 4.15. Johansson's Quadruple Tank, $y_1(t)$ according to various tuning methods

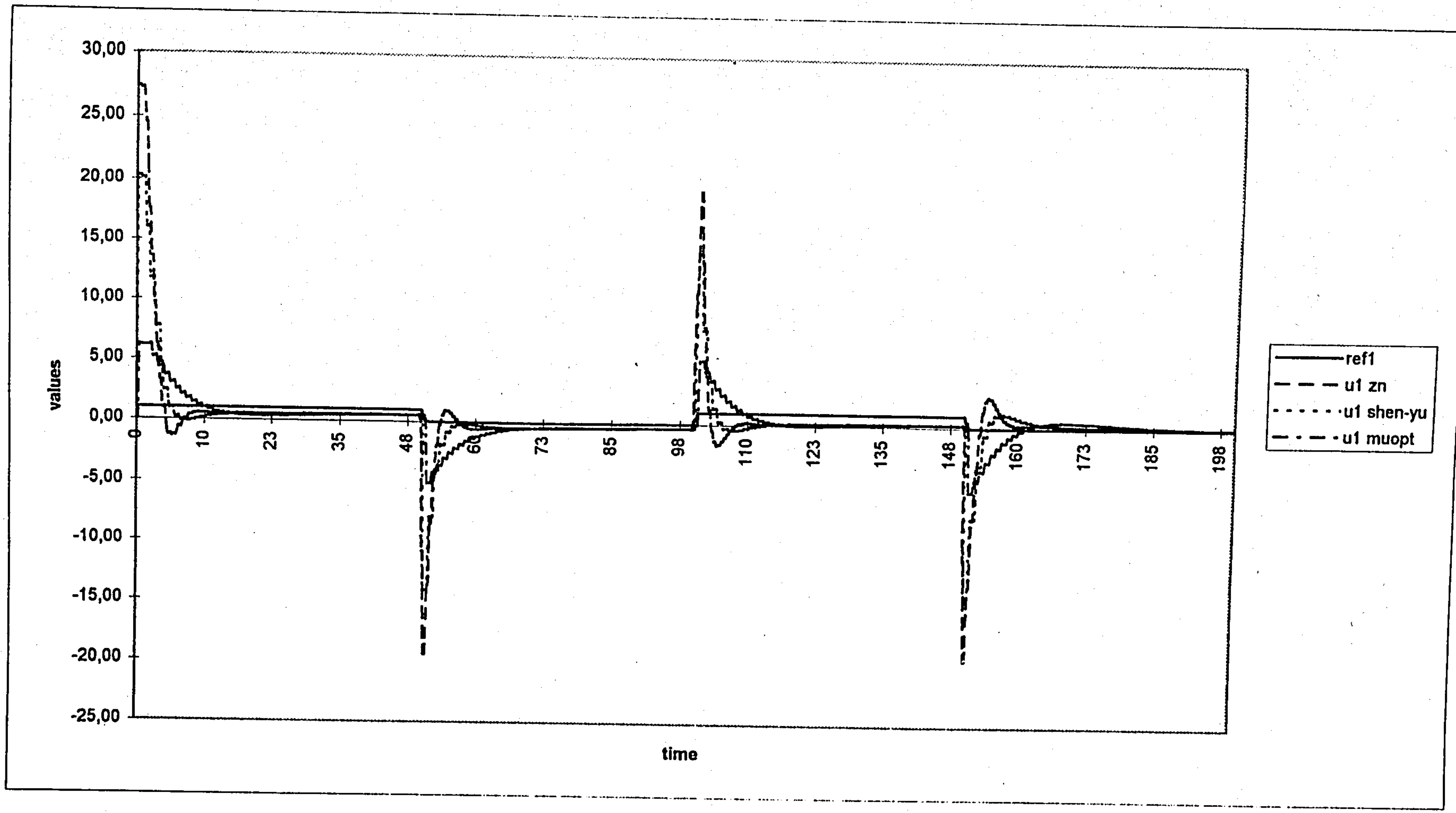

FIGURE 4.16. Johansson's Quadruple Tank, $u_l(t)$ according to various tuning methods

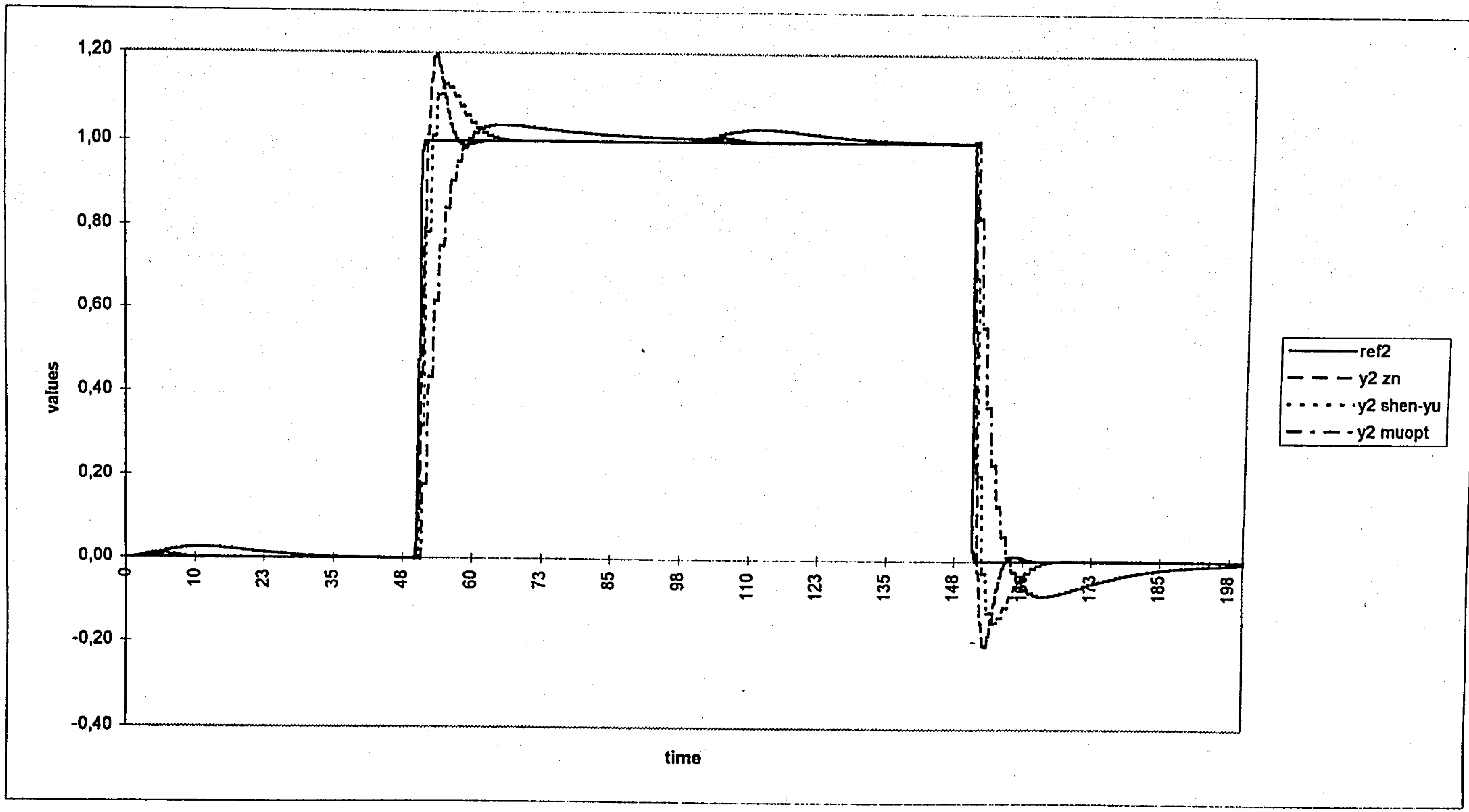

FIGURE 4.17. Johansson's Quadruple Tank, $y_2(t)$ according to various tuning methods

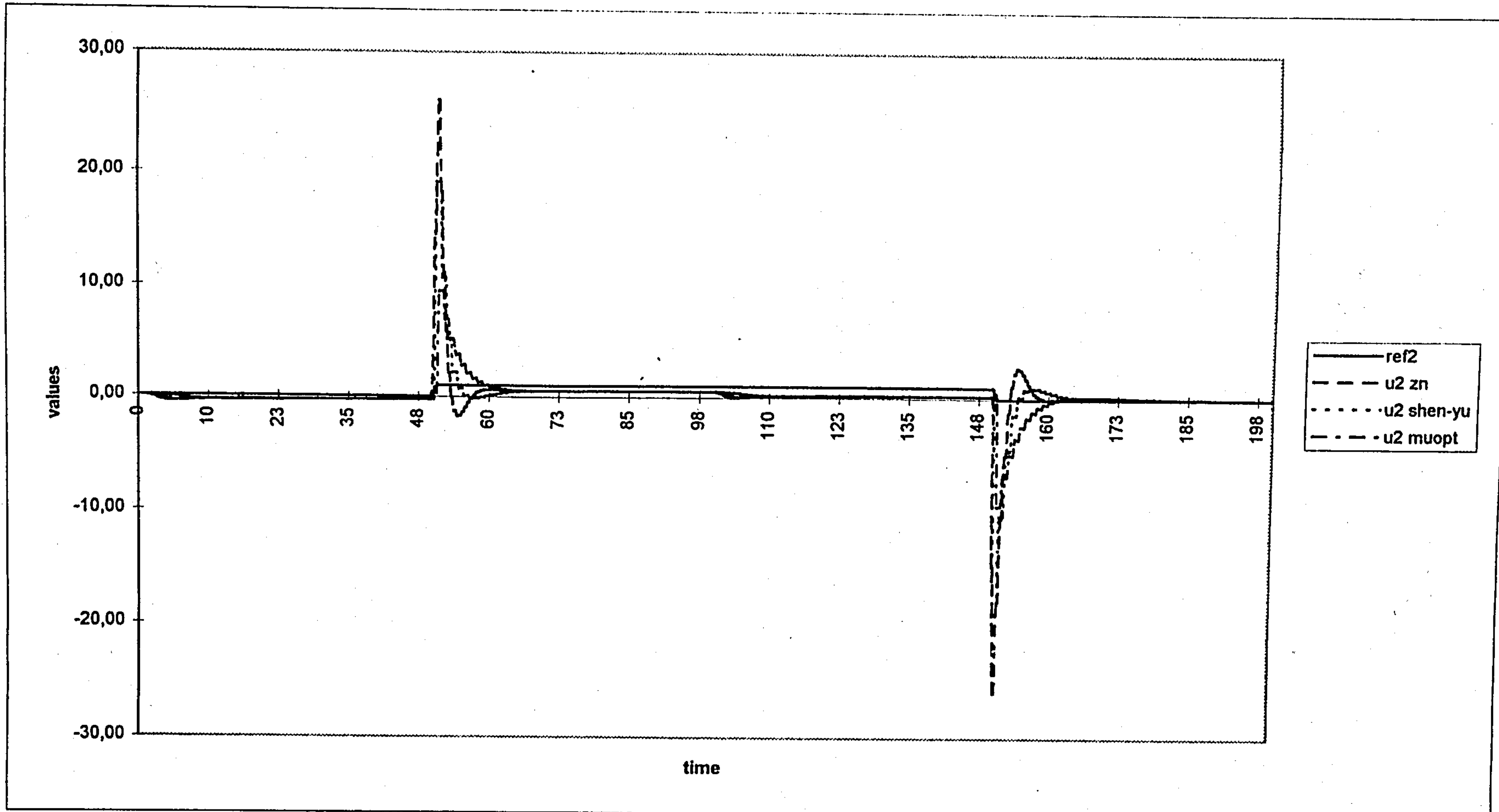

FIGURE 4.18. Johansson's Quadruple Tank, $u_2(t)$ according to various tuning methods

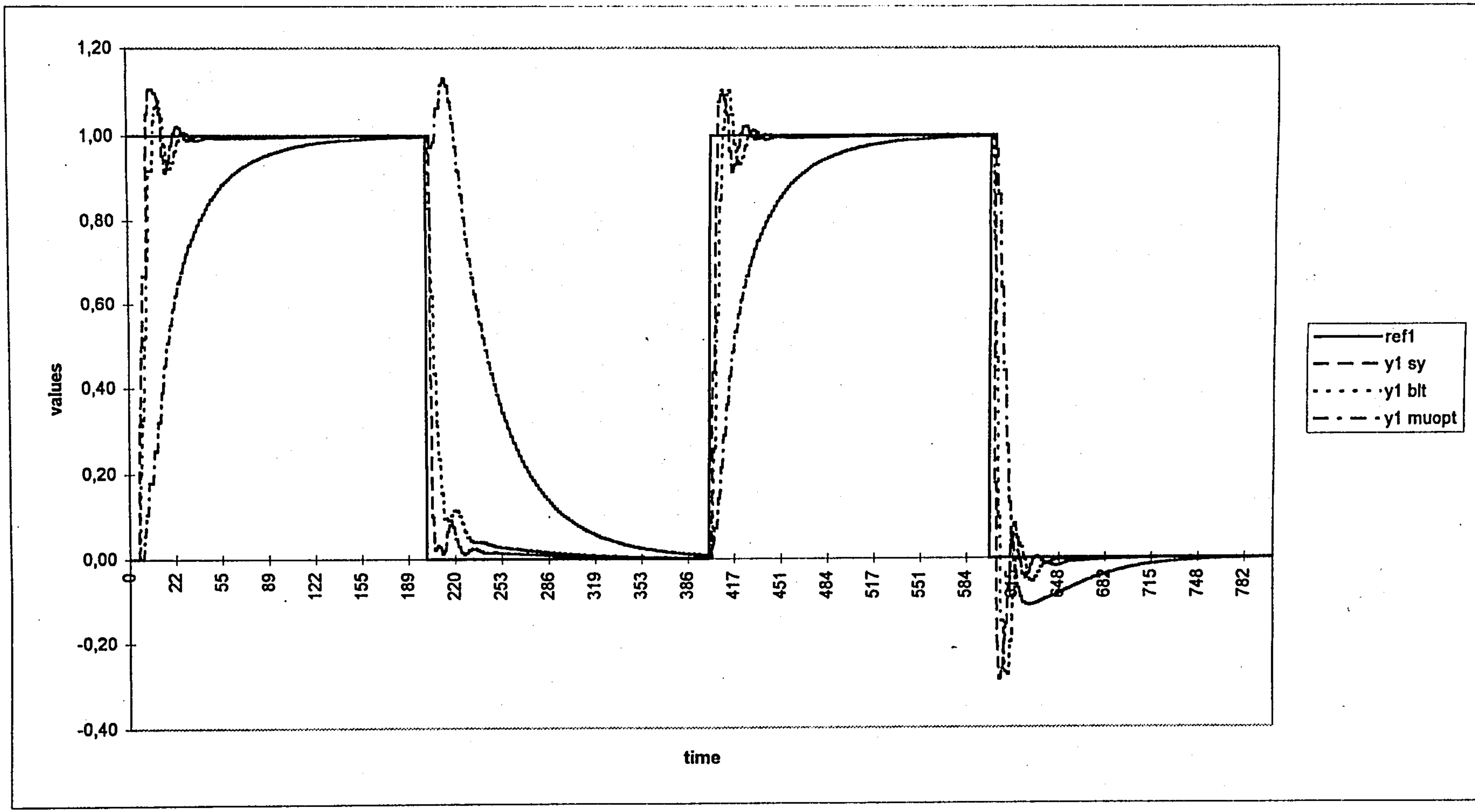

FIGURE 4.19. Wood and Berry Dist. Col. with input uncertainties, $y_1(t)$ according to various tuning methods

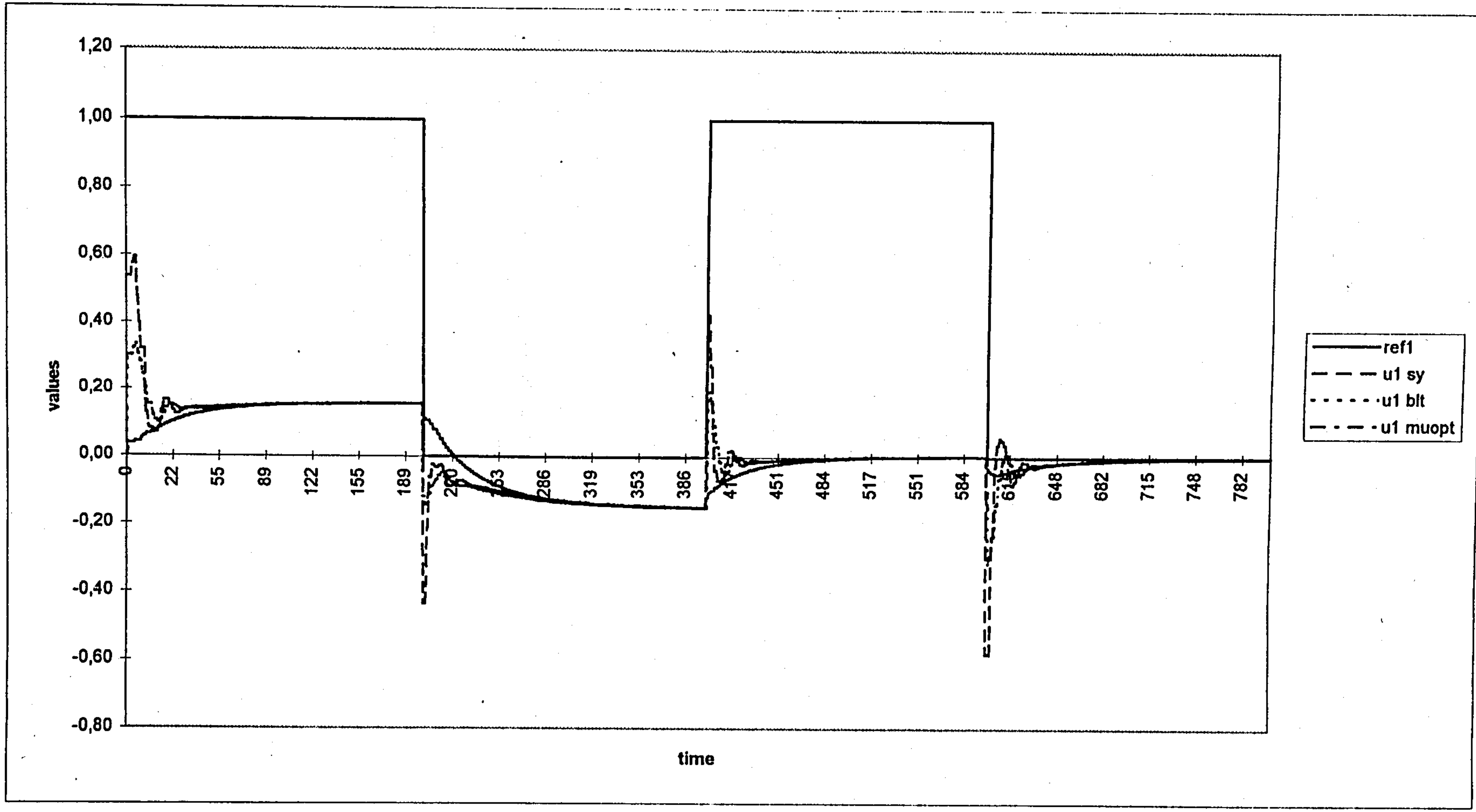

FIGURE 4.20. Wood and Berry Dist. Col. with input uncertainties, $u_1(t)$ according to various tuning methods

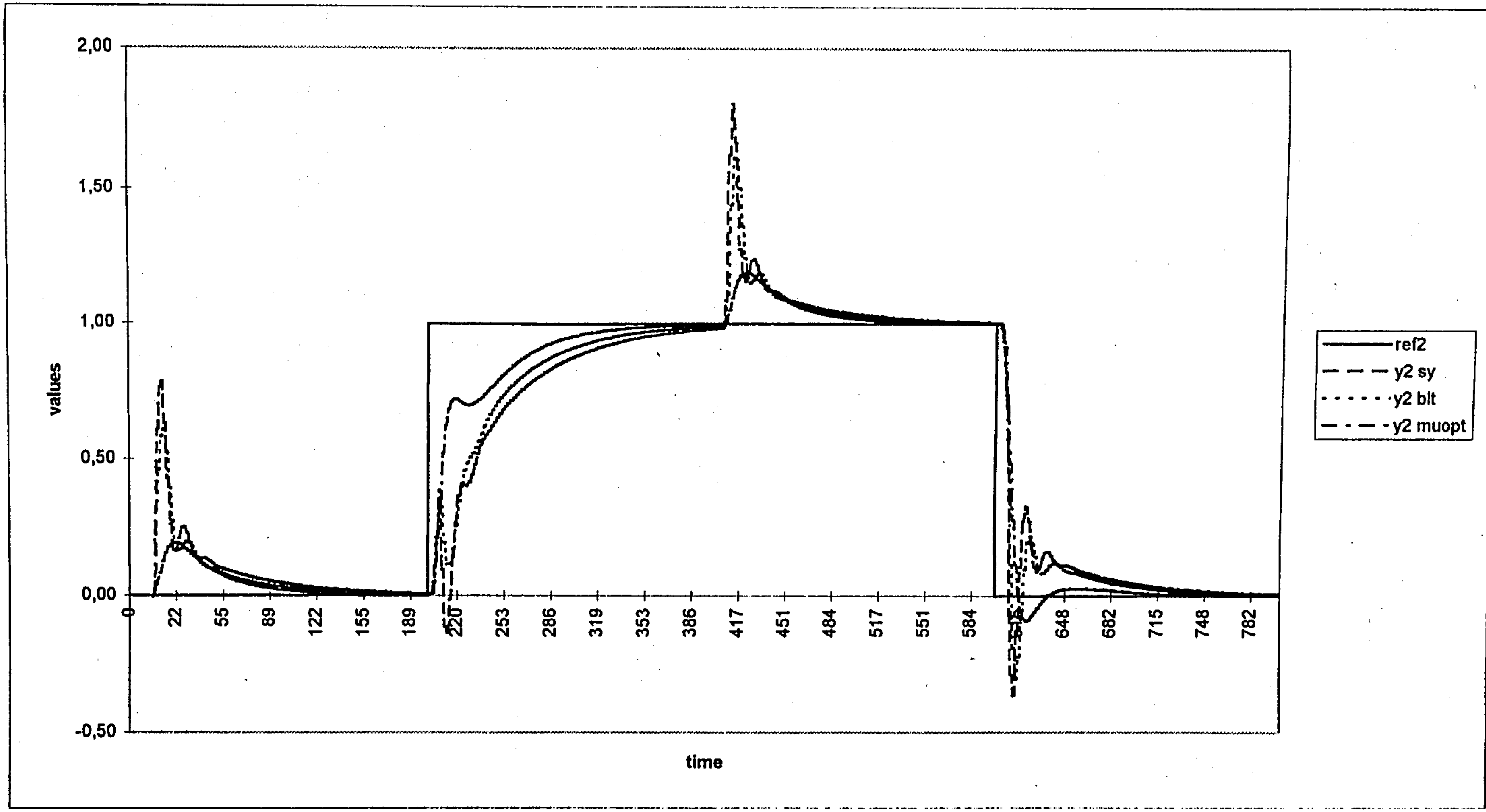

FIGURE 4.21. Wood and Berry Dist. Col. with input uncertainties, $y_2(t)$ according to various tuning methods

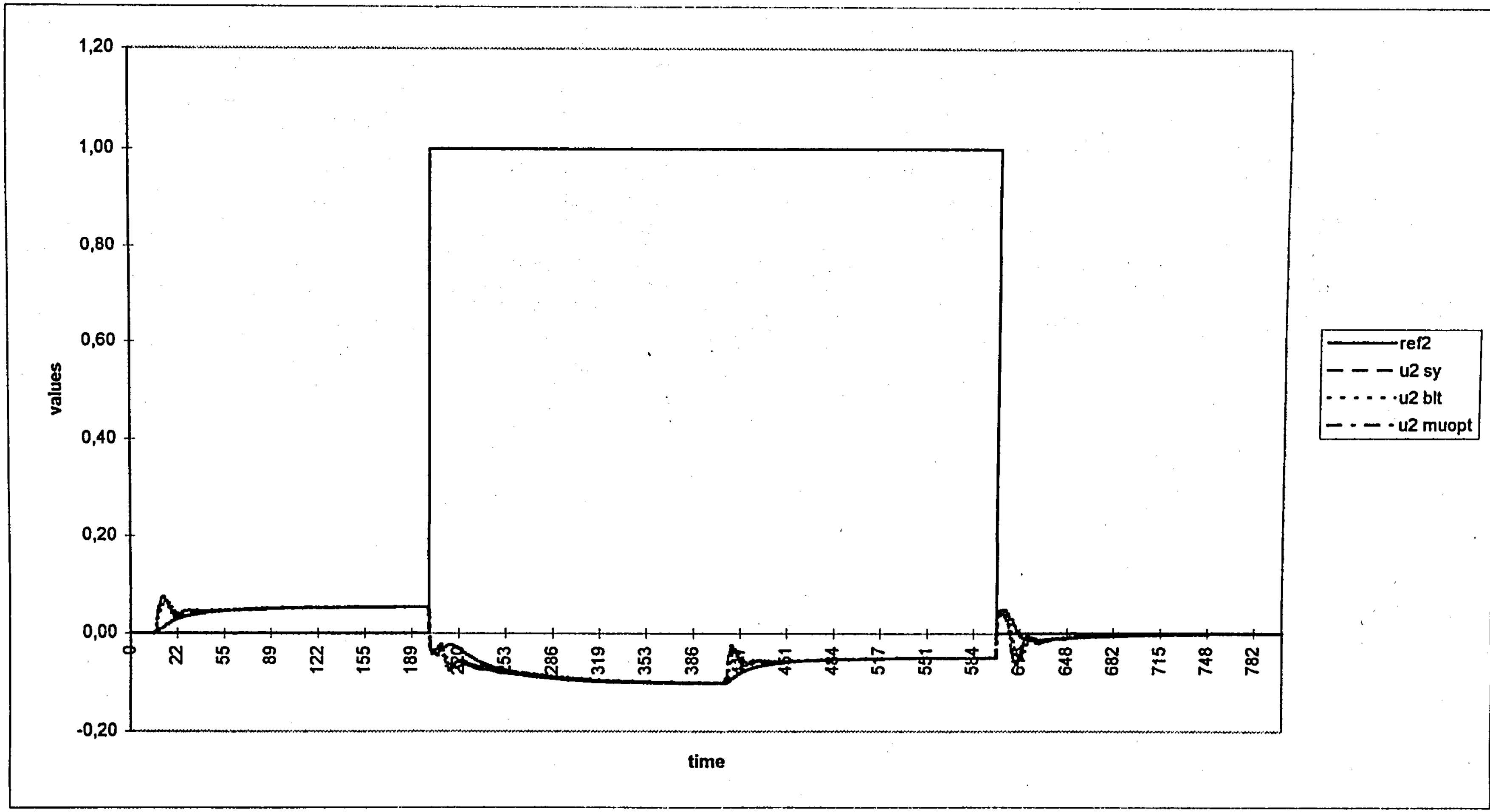

FIGURE 4.22. Wood and Berry Dist. Col. with input uncertainties, $u_2(t)$ according to various tuning methods

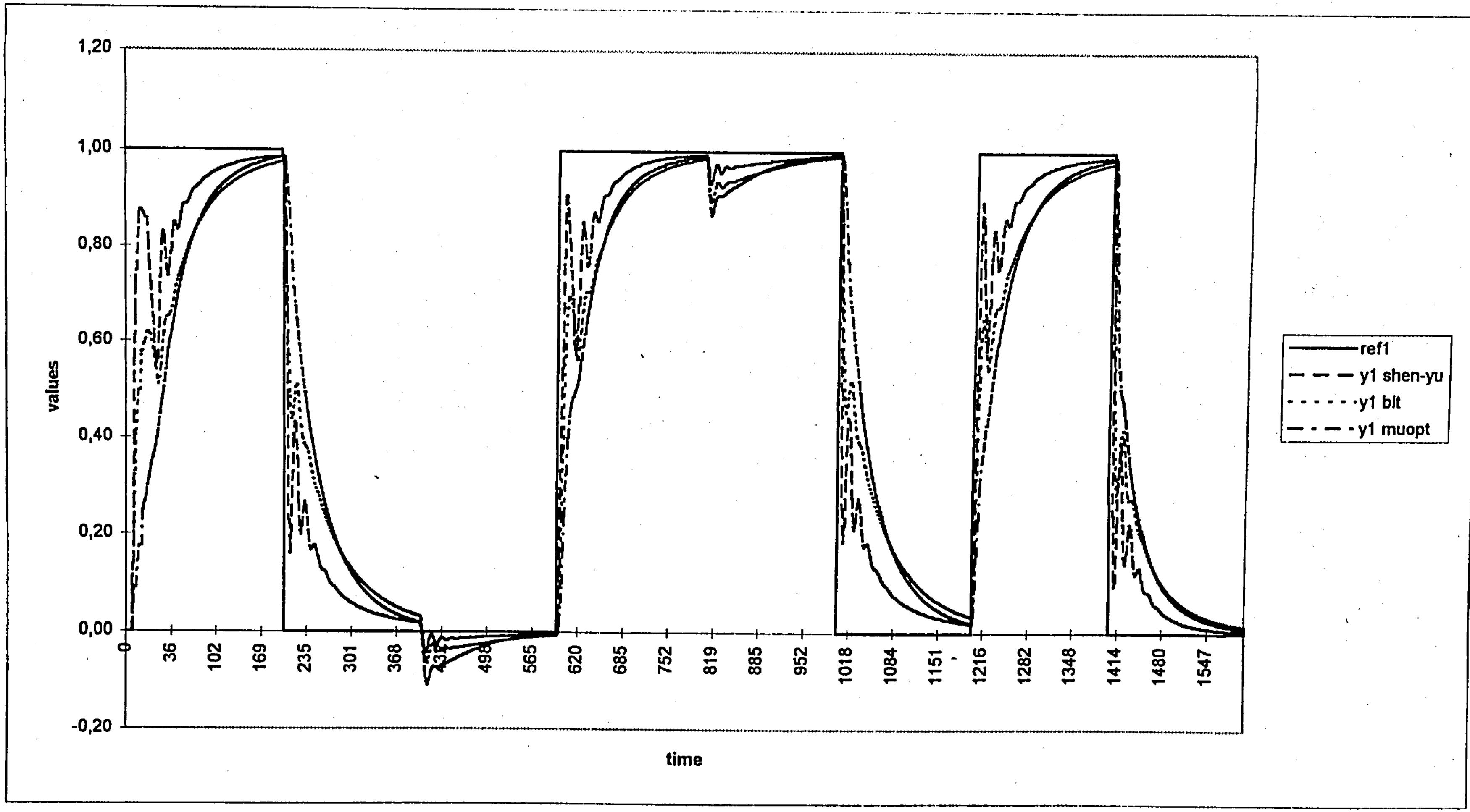

FIGURE 4.23. Ogunnaike and Ray Dist. Col. with input uncertainties, $y_1(t)$ according to various tuning methods

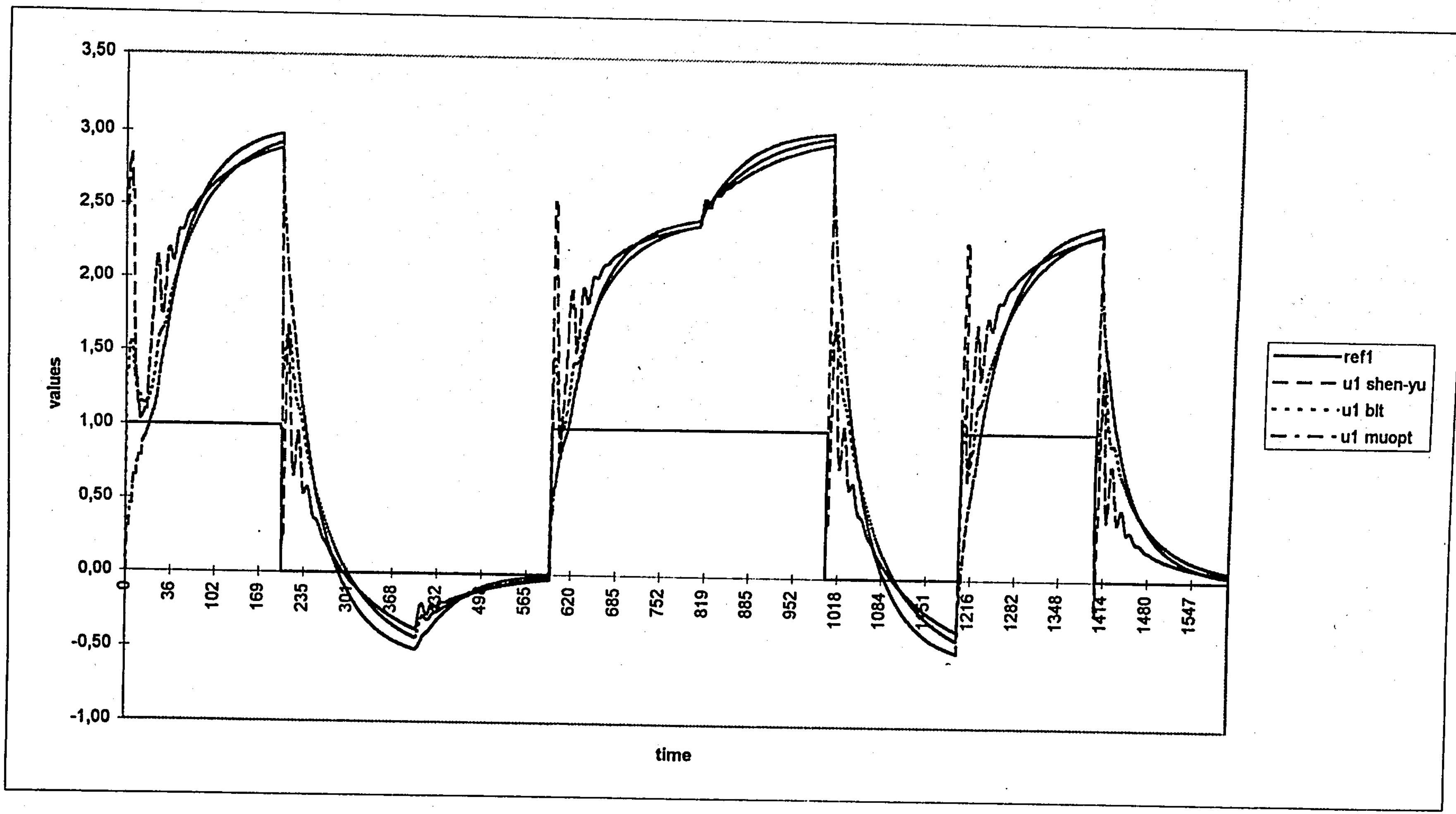

FIGURE 4.24. Ogunnaike and Ray Dist. Col. with input uncertainties, $u_1(t)$ according to various tuning methods

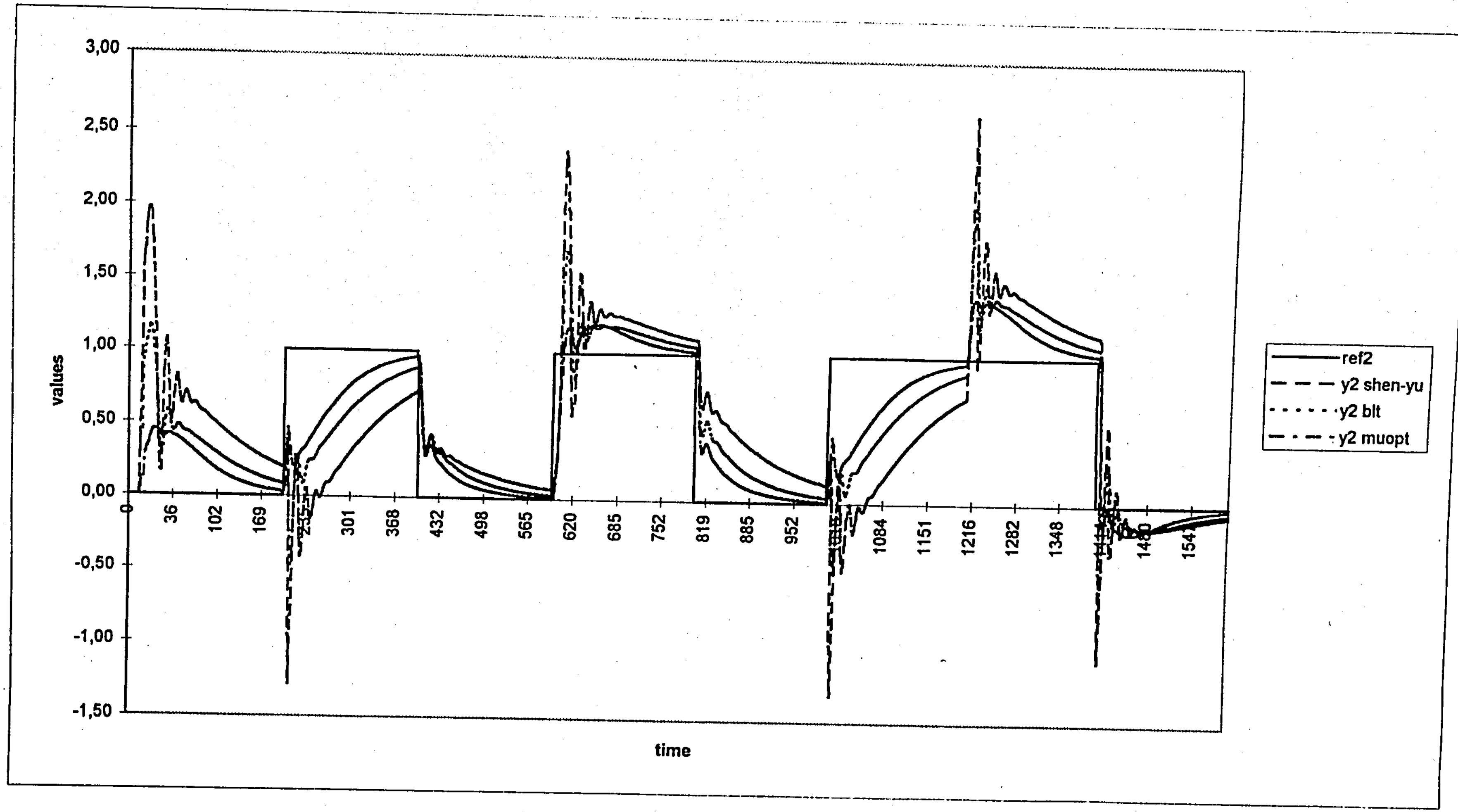

FIGURE 4.25. Ogunnaike and Ray Dist. Col. with input uncertainties, $y_2(t)$ according to various tuning methods

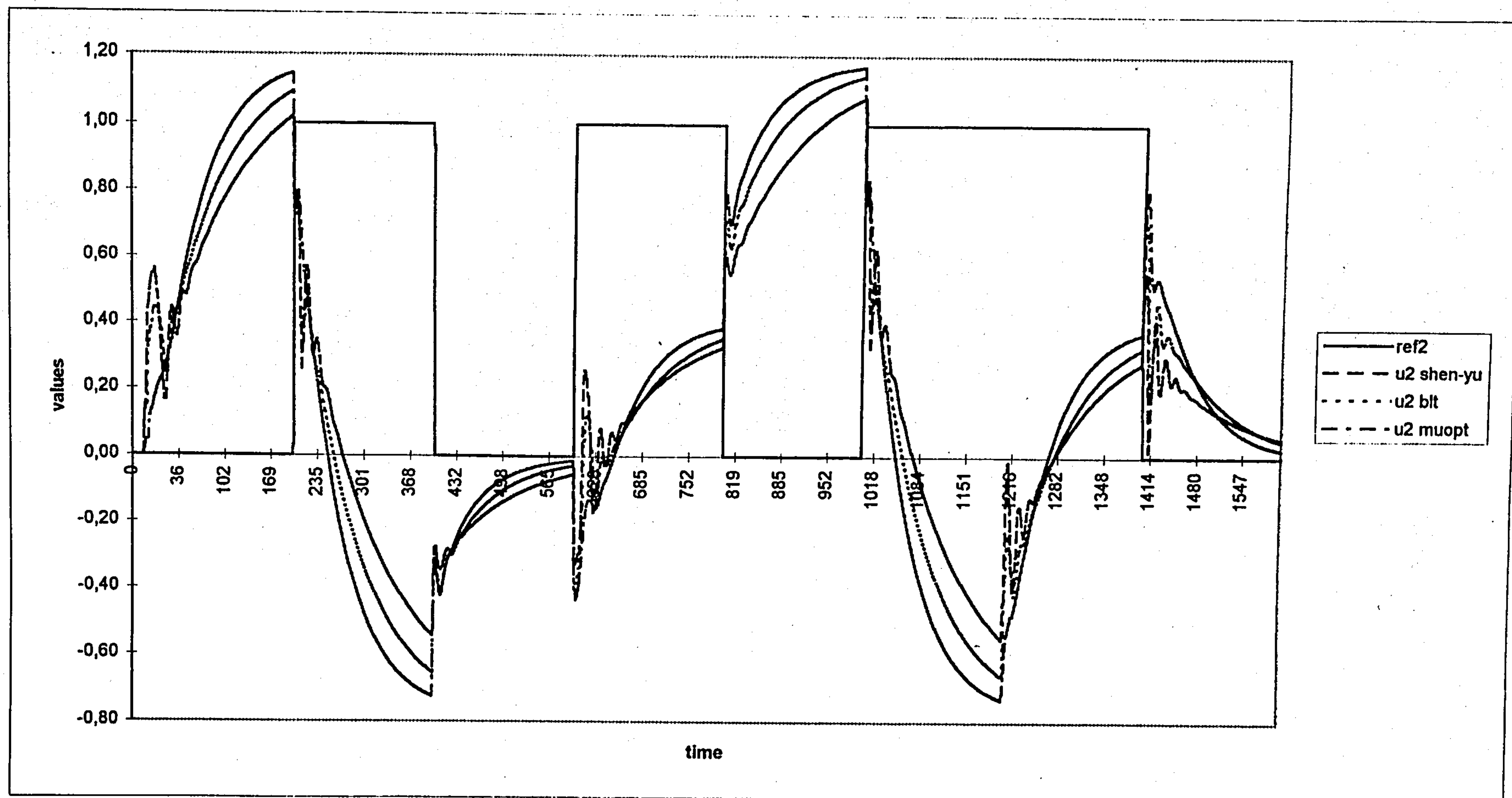

FIGURE 4.26. Ogunnaike and Ray Dist. Col. with input uncertainties, $u_2(t)$ according to various tuning methods

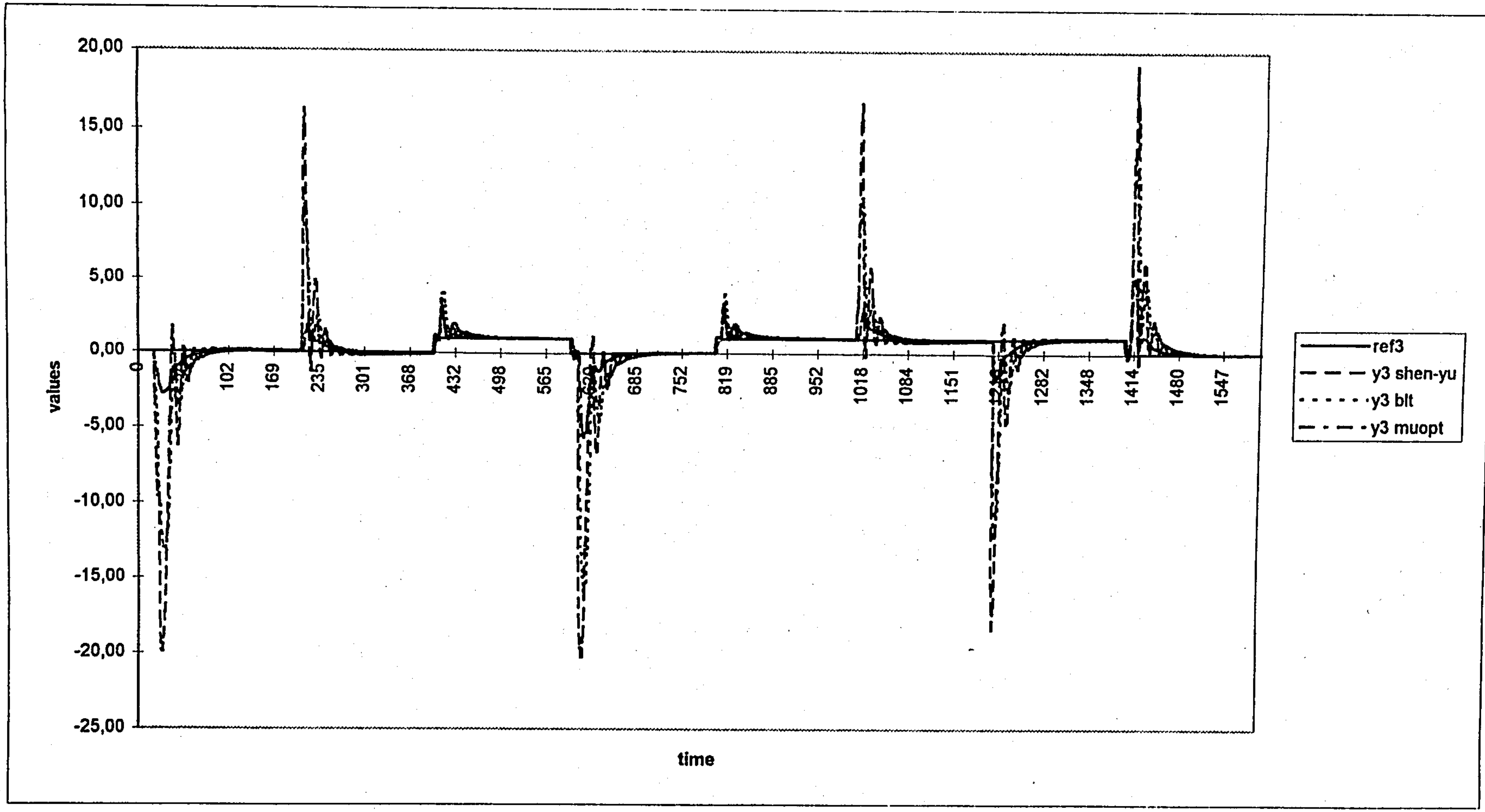

FIGURE 4.27. Ogunnaike and Ray Dist. Col. with input uncertainties, $y_3(t)$ according to various tuning methods

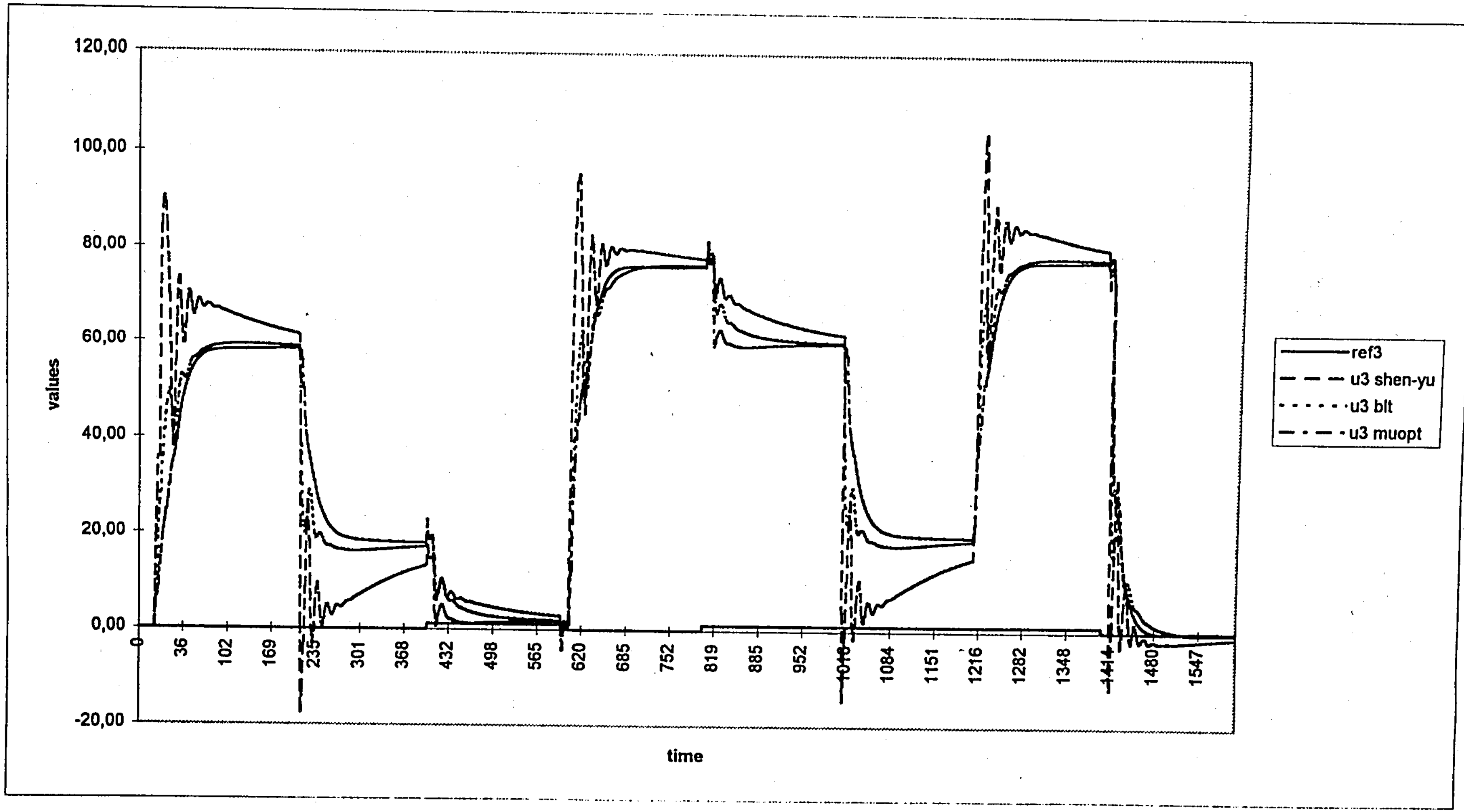

FIGURE 4.28. Ogunnaike and Ray Dist. Col. with input uncertainties, $u_3(t)$ according to various tuning methods

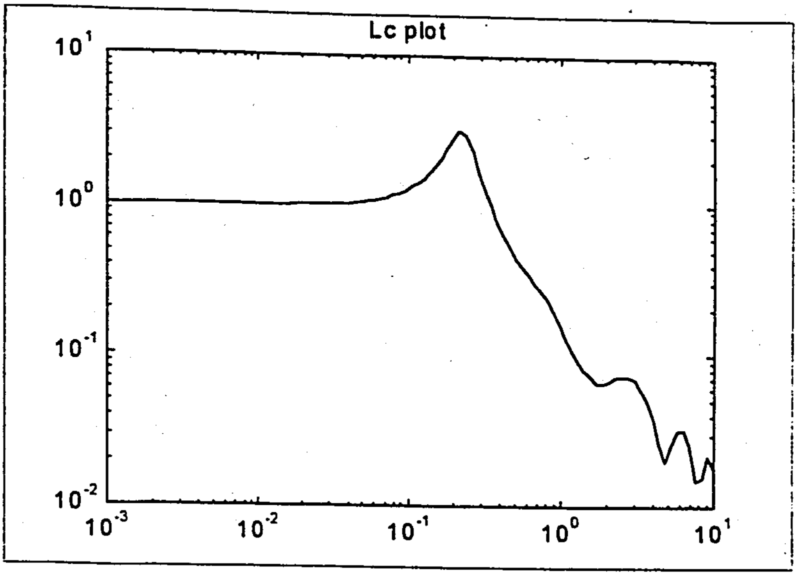

FIGURE 4.29. L_c vs. frequency for WB with LuyEmp settings

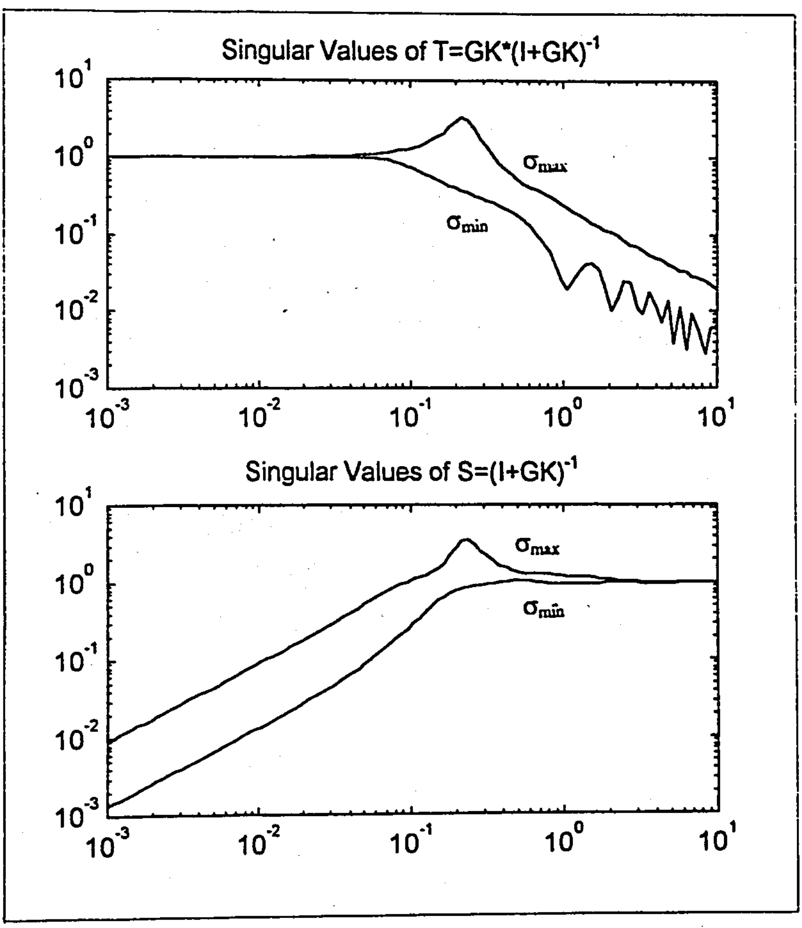

FIGURE 4.30. Singular values of the complementary sensitivity T and sensitivity functions S vs. frequency for WB with LuyEmp settings

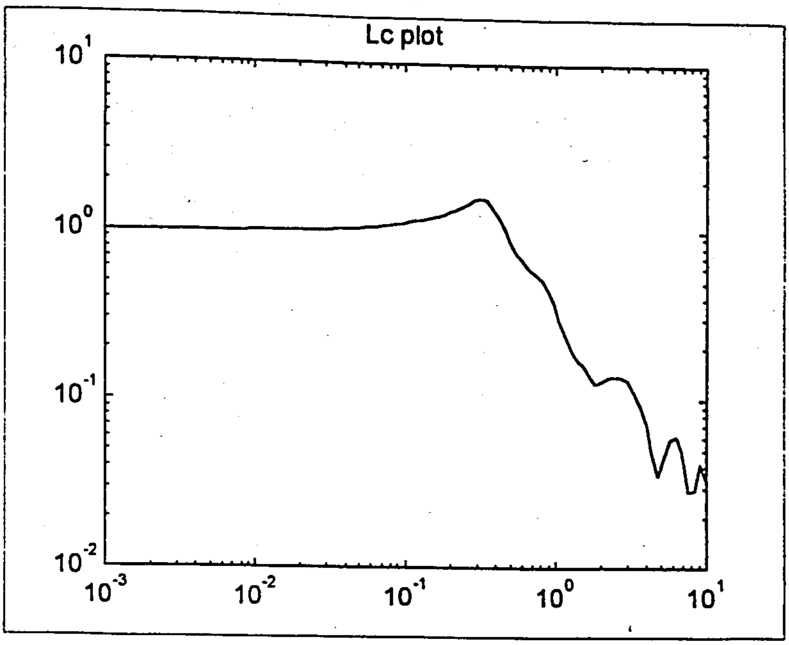

FIGURE 4.31. L_c vs. frequency for WB with BLT settings

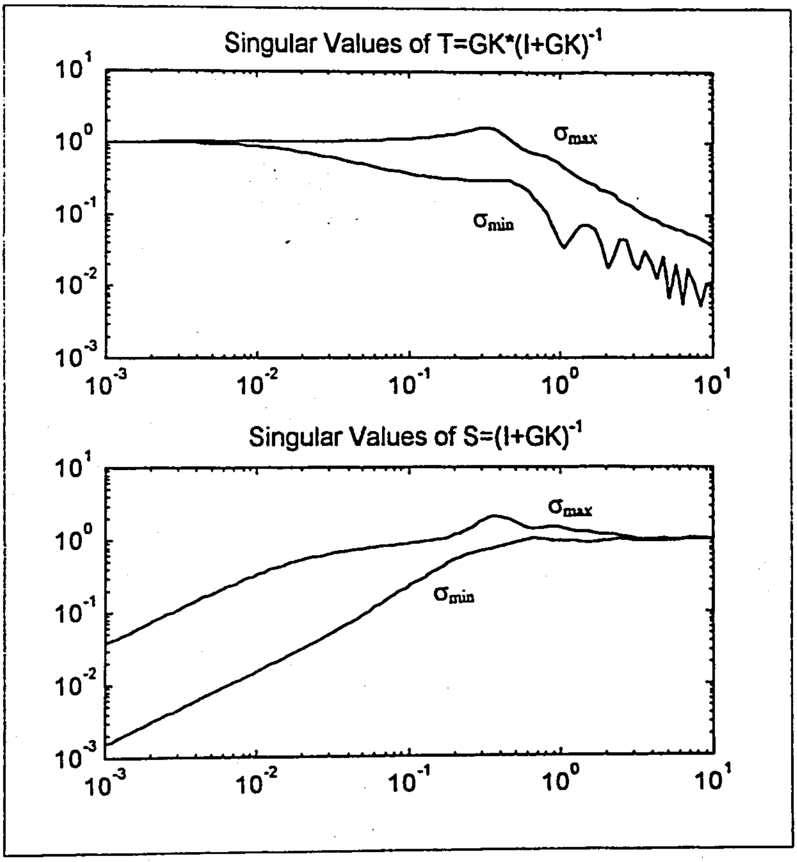

FIGURE 4.32. Singular values of the complementary sensitivity T and sensitivity functions S vs. frequency for WB with BLT settings

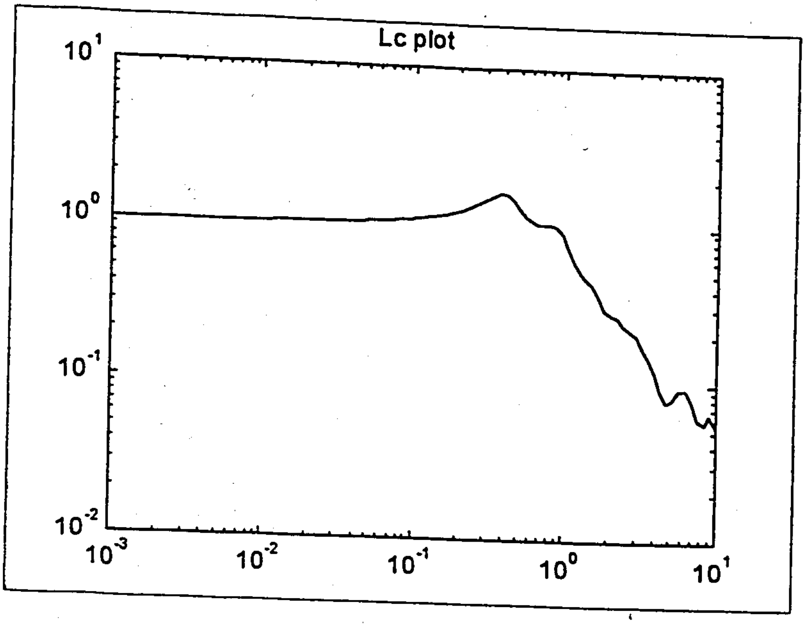

FIGURE 4.33. L_c vs. frequency for WB with SY settings

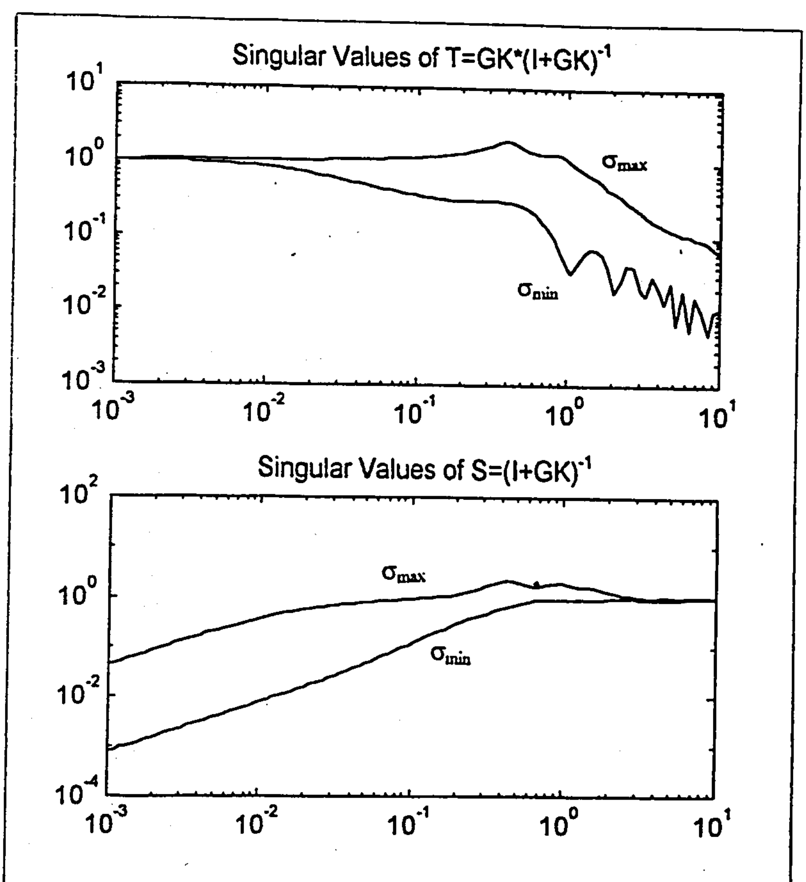

FIGURE 4.34. Singular values of the complementary sensitivity T and sensitivity functions S vs. frequency for WB with SY settings

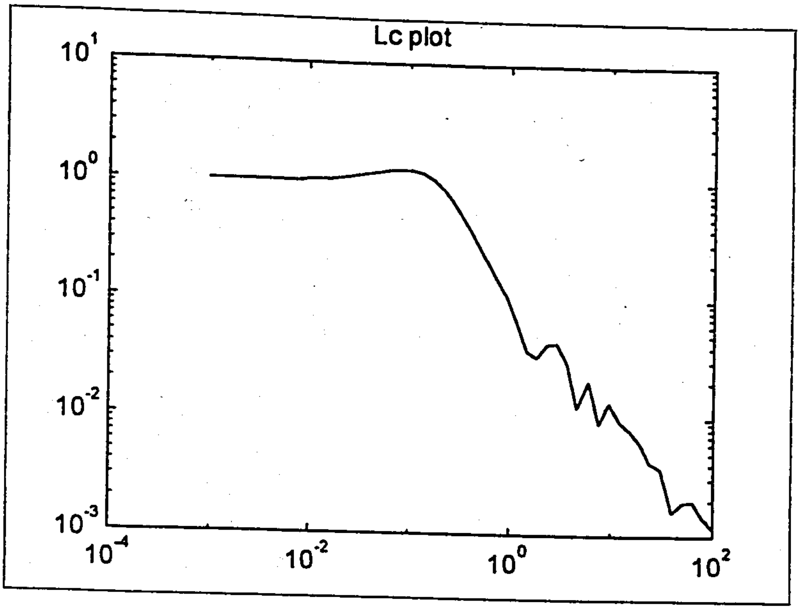

FIGURE 4.35. L_c vs. frequency for WB with μ-optimal settings

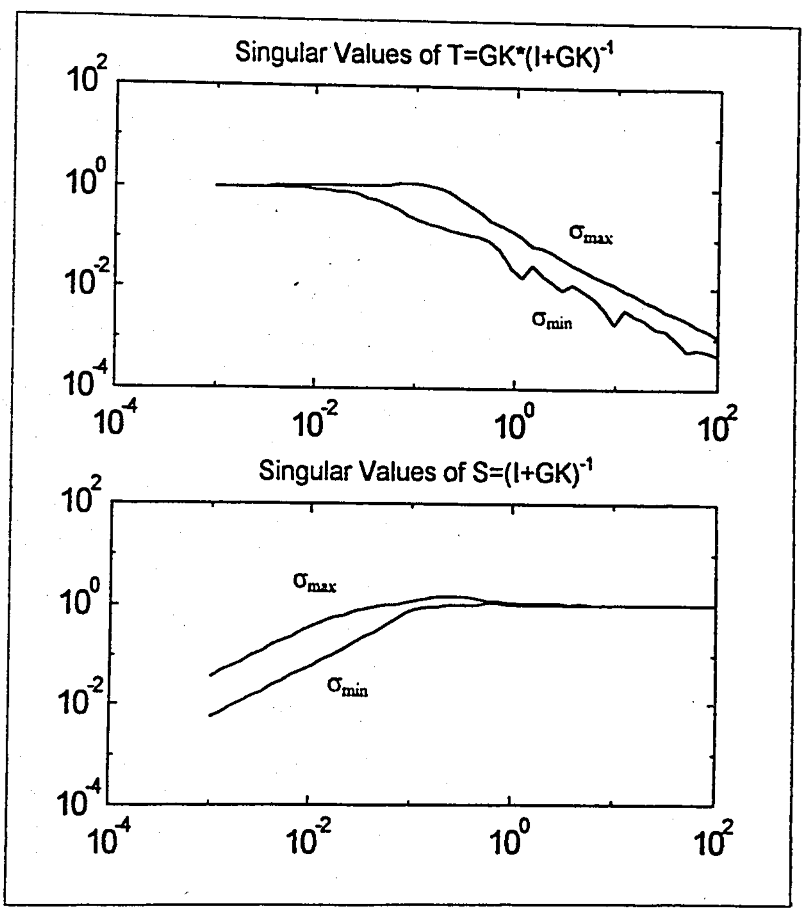

FIGURE 4.36. Singular values of the complementary sensitivity T and sensitivity functions S vs. frequency for WB with μ-optimal settings

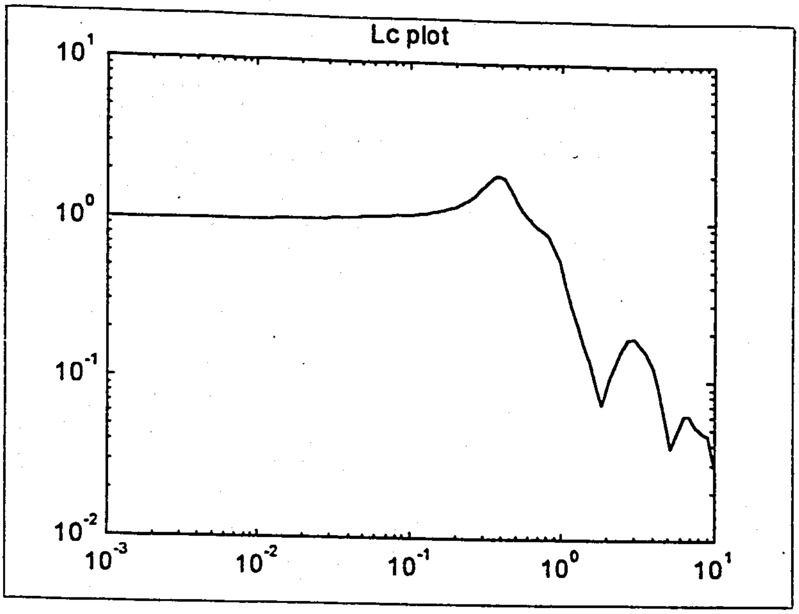

FIGURE 4.37. L_c vs. frequency for OR with BLT-6 settings

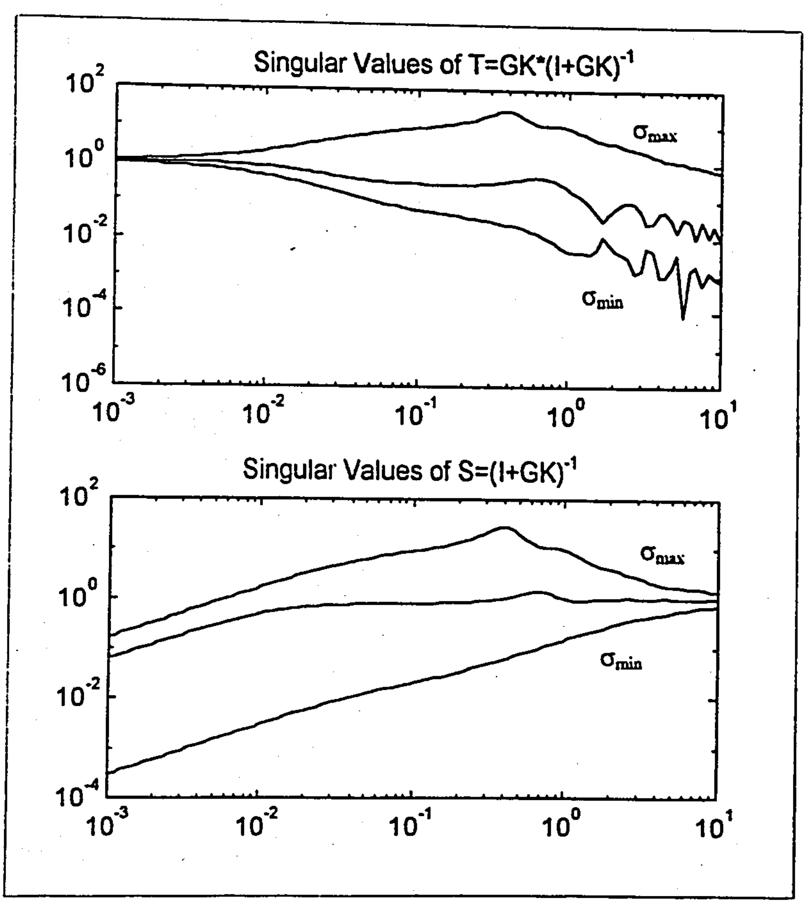

FIGURE 4.38. Singular values of the complementary sensitivity T and sensitivity functions S vs. frequency for OR with BLT-6 settings

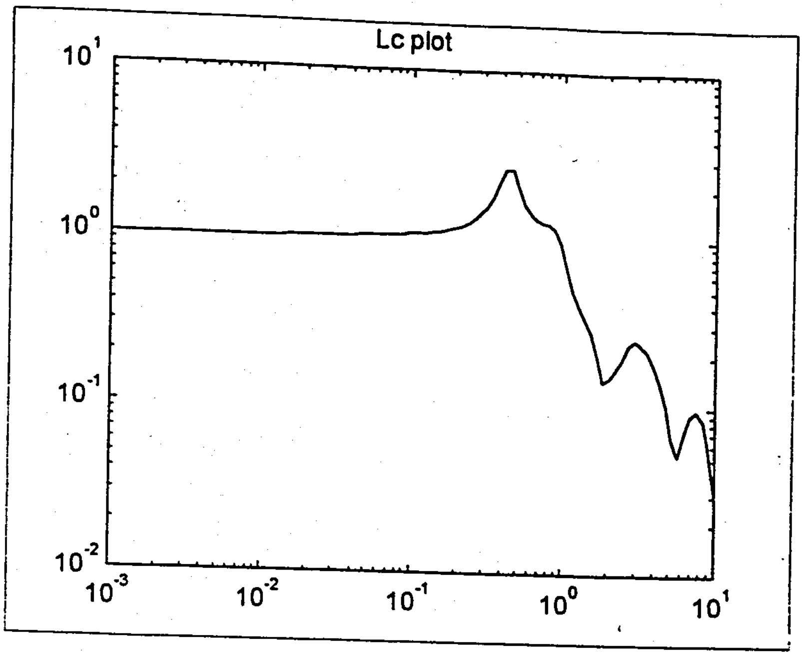

FIGURE 4.39. L_c vs. Frequency for OR with SY settings

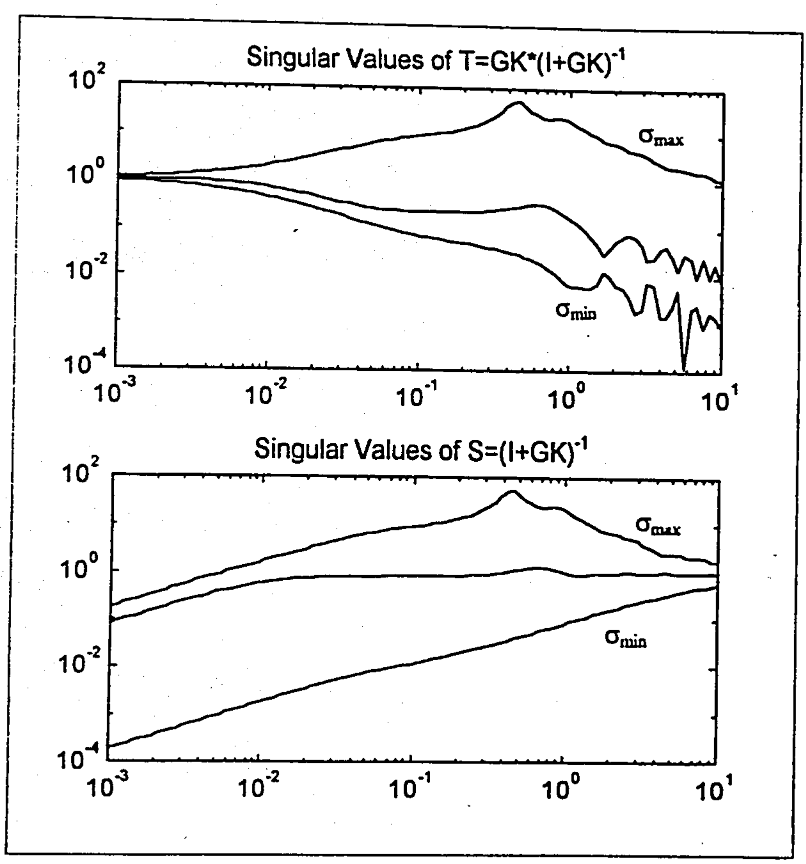

FIGURE 4.40. Singular values of the complementary sensitivity T and sensitivity functions S vs. frequency for OR with SY settings

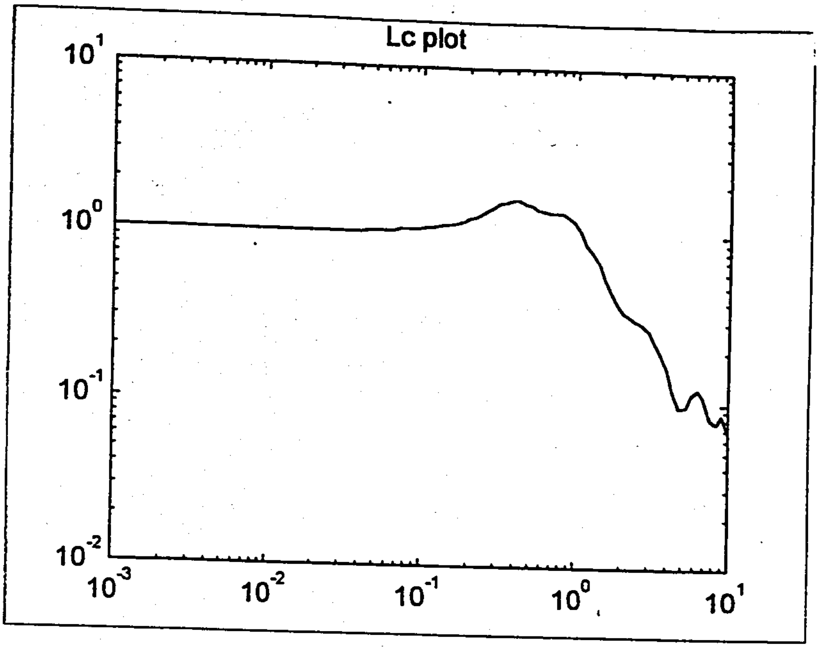

FIGURE 4.41. L_c vs. Frequency for OR with μ-optimal settings

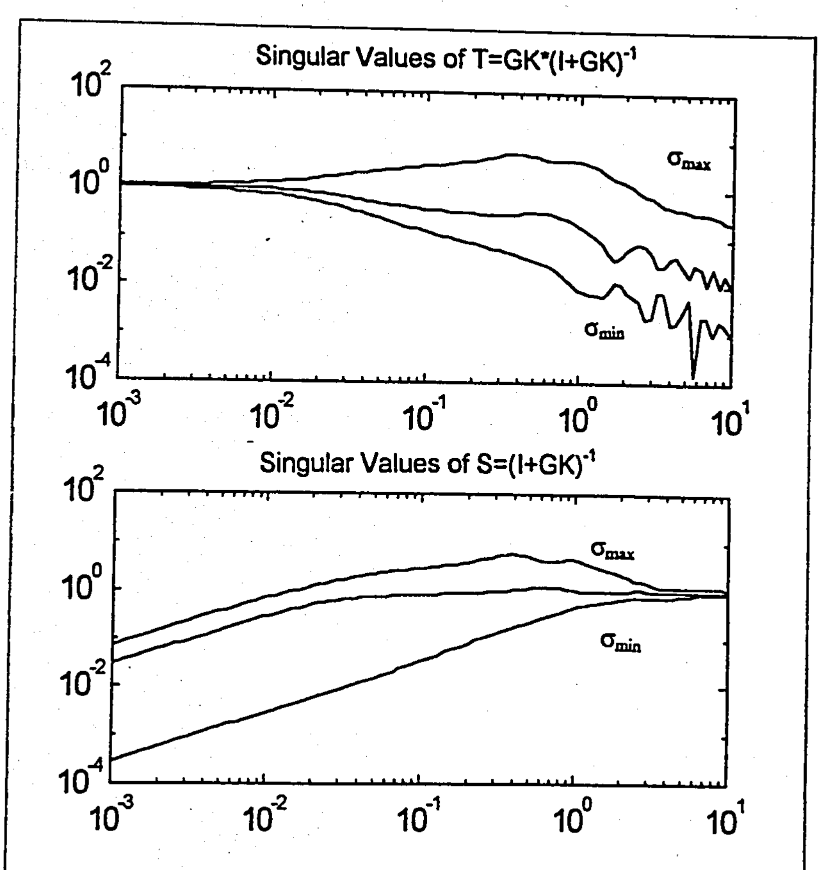

FIGURE 4.42. Singular values of the complementary sensitivity T and sensitivity functions S vs. frequency for OR with μ-optimal settings

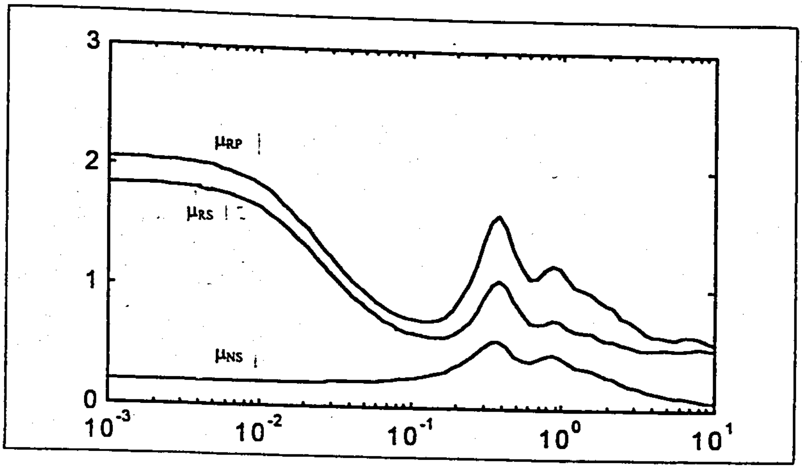

FIGURE 4.43. μ-plots for WB-2 with BLT settings

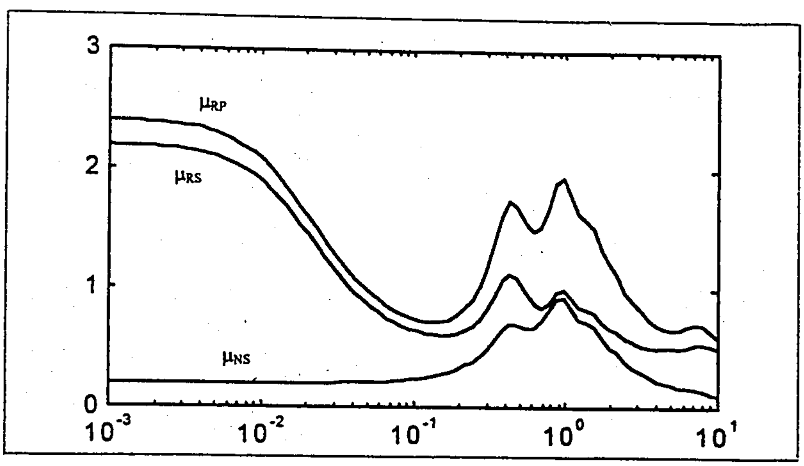

FIGURE 4.44. μ-plots for WB-2 with SY settings

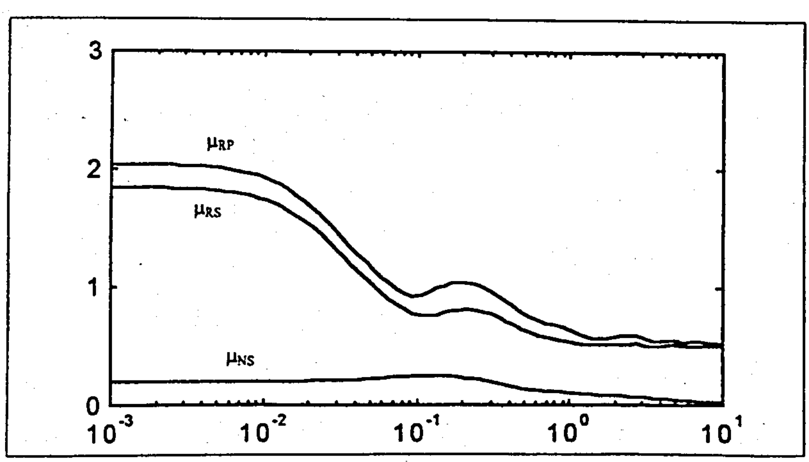

FIGURE 4.45. μ-plots for WB-2 with μ-optimal settings

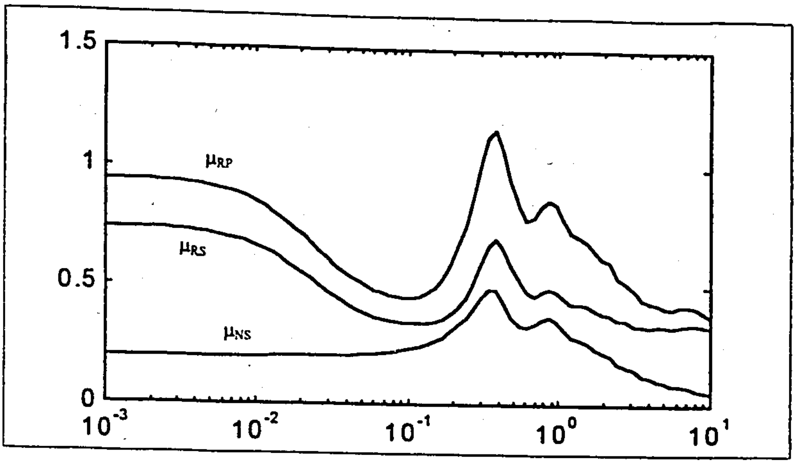

FIGURE 4.46. μ-plots for WB with BLT settings

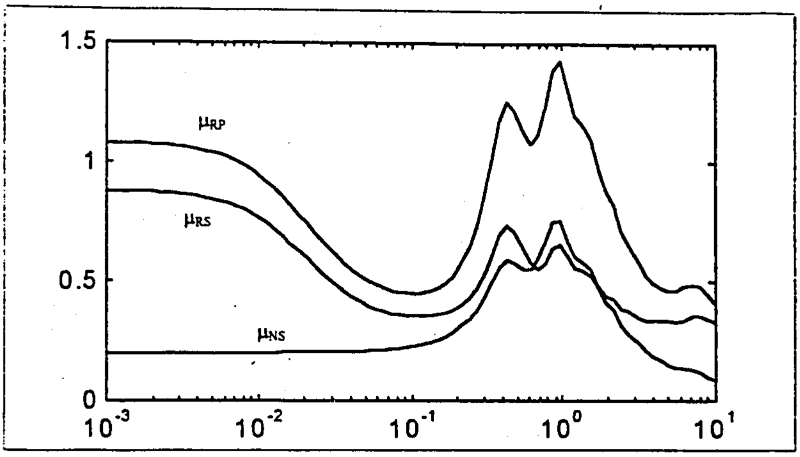

FIGURE 4.47. μ-plots for WB with SY settings

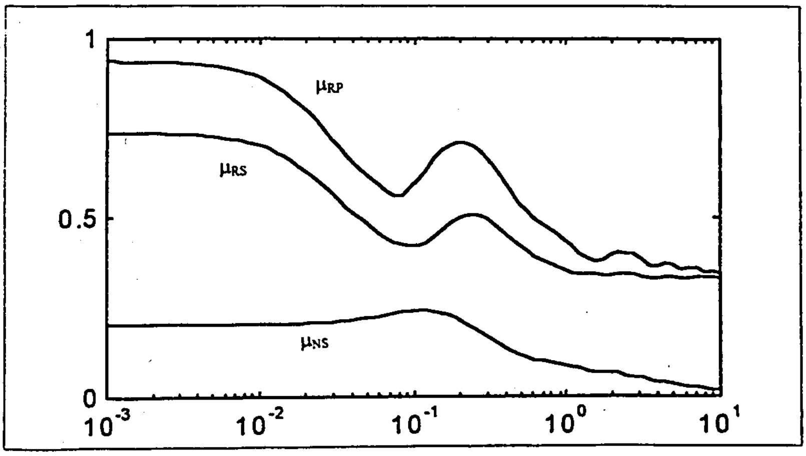

FIGURE 4.48. μ-plots for WB with μ-optimal settings

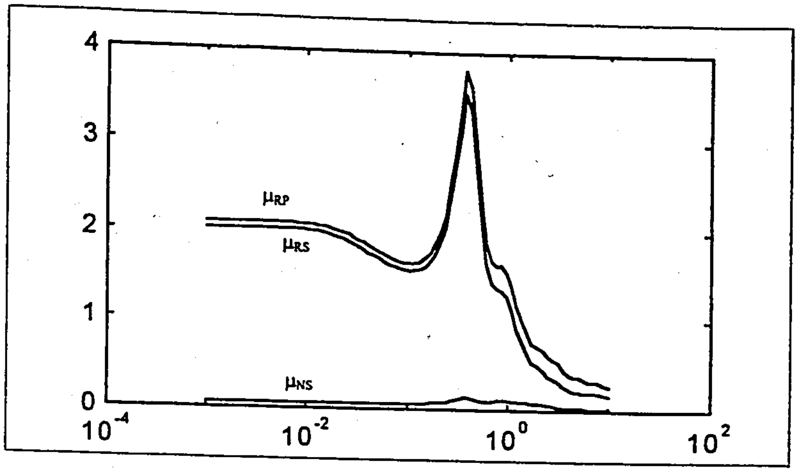

FIGURE 4.49. μ-plots for OR with BLT settings

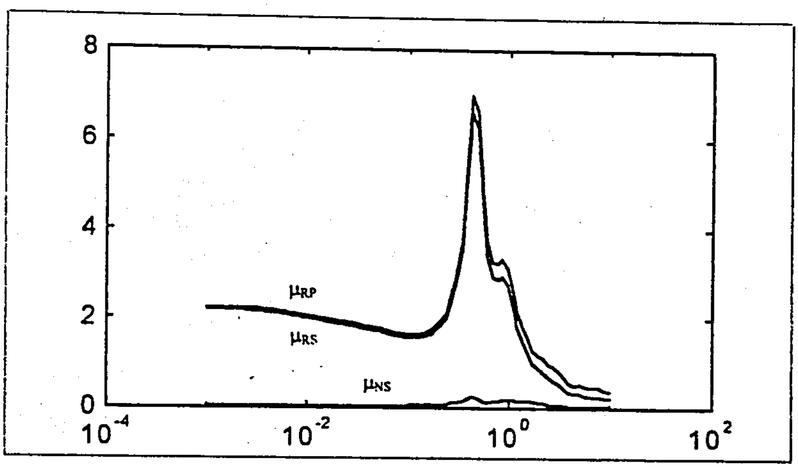

FIGURE 4.50. μ-plots for OR with SY settings

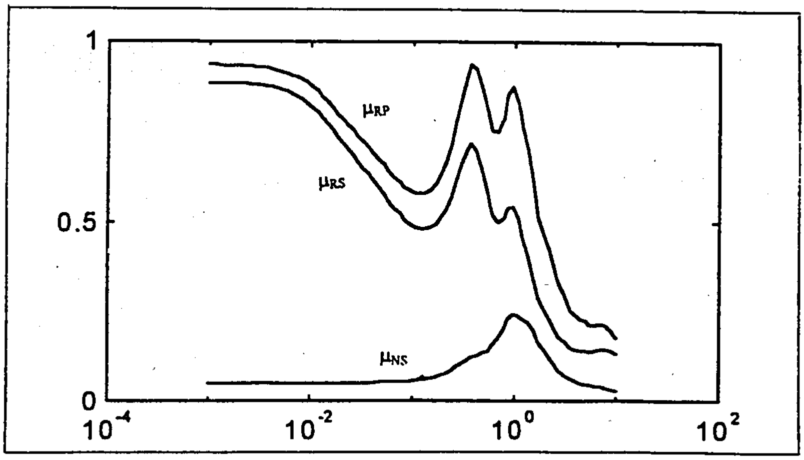

FIGURE 4.51. μ-plots for OR with μ-optimal settings

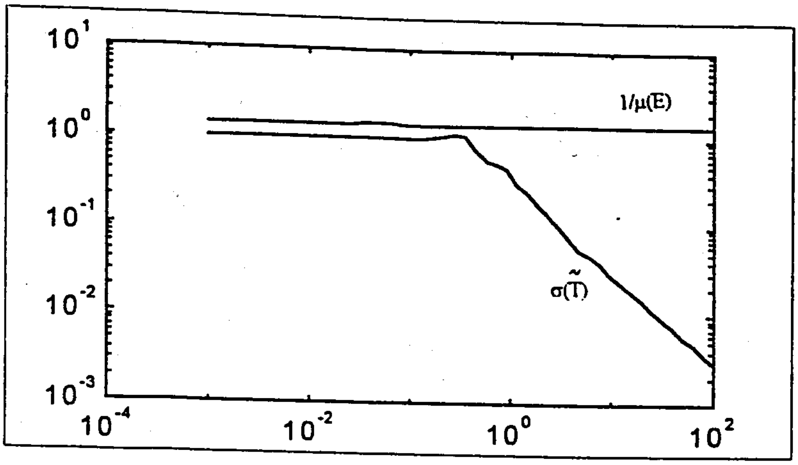

FIGURE 4.52. μ interaction plot for WB with BLT settings

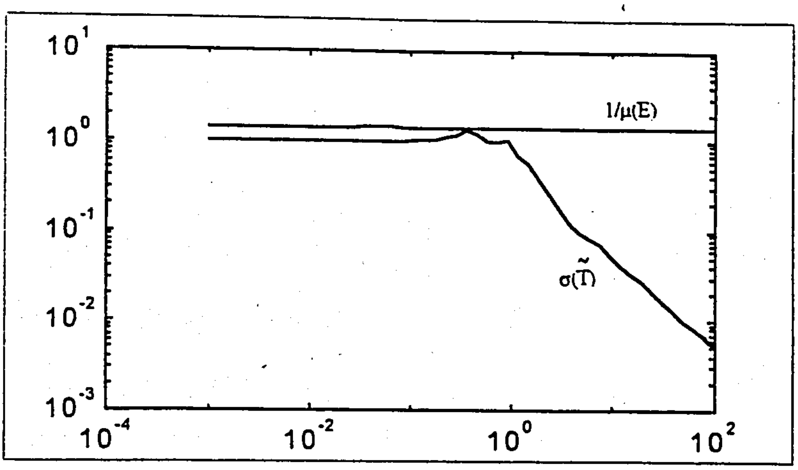

FIGURE 4.53. μ interaction plot for WB with BLT settings

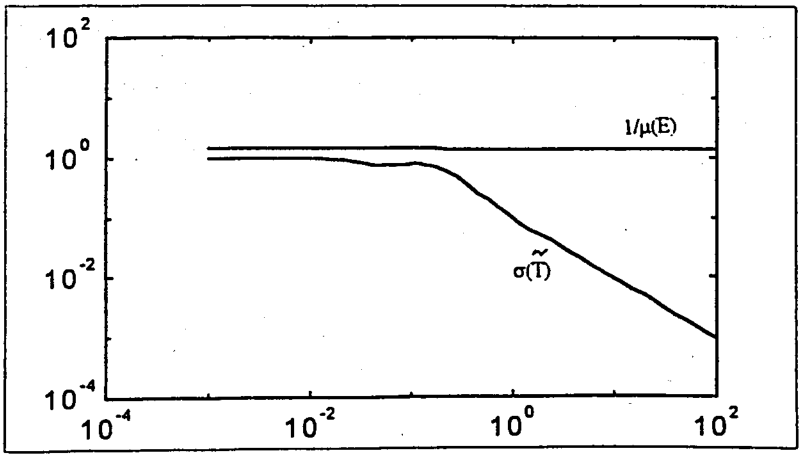

FIGURE 4.54. μ interaction plot for WB with μ-optimal settings

5. CONCLUSIONS

In this study, tuning of controllers used in process industries is investigated. The use of PID controllers is considered and decentralized PID control is taken to be an efficient solution to apply in the control for MIMO systems.

The basic characteristics and the similarities between SISO and MIMO systems are considered. The logic of the SISO systems is extended to MIMO systems. The minimum requirements for analyzing the responses of MIMO systems are given. Performance evaluation of multivariable systems is presented with details: singular value analysis and interpretation of condition number in practical use.

Previous work on controller tuning is detailed, because the basis of the autotuning procedure depends on a good knowledge in controller tuning. Three different tuning methods are presented together with the sequential relay feedback and automation of the tuning procedures.

Uncertainty, instability, robustness and their relation with the structured singular value, μ, are also given. μ-optimal tuning is used to obtain decentralized multivariable PID controller settings for a given system under a pre-defined performance and perturbation criteria.

An autotuning program is developed in MATLAB. This program identifies the basic system properties, sorts the loops according to descending loop speeds, identifies the necessary parameters for controller tuning in all loops. Then, it designs controllers for each loop until the parameters converge to their optimum values. Autotuning is based on the MV version of relay feedback method.

μ-optimal settings are found differently. Optimization Toolbox of MATLAB automates the procedure, and after a long optimization procedure μ-optimal settings are calculated for the decentralized multivariable PID controllers.

Another program is written in MATLAB to evaluate robustness. This μ-analysis program uses the controller parameter, input uncertainty and performance weight entries, and finds the upper bounds for μ in a frequency interval for a system.

All the above mentioned programs are run, simulation results are obtained and the results are compared with each other.

Our results indicate that the BLT tuning method can be applied to 2x2 systems, but as the number of input and outputs increase, the method gives unsatisfactory results. The systems become more oscillatory, the input values of these systems exceed reasonable levels, and overshoots in the time response rise. BLT method requires a model, which makes it difficult to apply if the exact mathematical model is not known. BLT method also needs calculations for $L_{c,max}$ at some frequency points. All these disadvantages make BLT method difficult to apply to every system in the process industries.

Shen and Yu's tuning method produces reasonable results. This tuning method does not require a model of the plant. The identification part of this method is based on identifying a single point on the Nyquist curve. The PID controller settings can be calculated with the help of this single point, which makes the identification and design easier than those of BLT. Shen and Yu's method is formulated as simple as Ziegler-Nichols method, but with an addition: a constant detuning factor.

In Shen and Yu's method, an increase in the number of inputs and outputs of the system does not affect the robustness. Input and output signals are on reasonable levels; but if model uncertainties and perturbations exist in the system, the method tends to produce unstable responses. This method requires the least computational time and effort among the other tuning methods. It is easy to apply and the simulation results show that it is efficient and appropriate for many industrial process applications.

μ-optimal method gives the best results as expected, because the PID controller settings are optimized to give optimum input signals for a system with input uncertainties and perturbations. The simulation results indicate that responses are stable for a wide range of uncertainties and perturbations. This method produces slower, more stable, nut more sluggish responses than the other two methods. Moreover, μ-optimal tuning method requires a system model and extensive computation. Autotuning with this method is easy only if a computer or a powerful processor is on-line with the system. So, μ-optimal tuning method should better be a method for comparison criterion for other tuning results, until it is further developed.

For further work, PID controllers on different loops can be detuned using different detuning factors for each loop. This can result in good performance and stability for the BLT and Shen-Yu methods even if strong interactions among the components of the transfer function matrix exist.

Another idea is to apply decouplers, pre- or post compensators and feedforward control, but the advantages and disadvantages of this idea should strictly be examined before application.

A logical idea can be an application of gain scheduling and/or adaptive algorithms with decentralized multivariable PID control for simple tuning methods. Different sets of controller parameters can prevent sluggish responses and/or unstable responses.

APPENDIX A

The developed autotuning program (written in MATLAB) for a 2x2 process:

```
% TUN2 is used to find the period of the system and the amplitude
% ratio of that system for a (unit) amplitude relay. It also sets
% the parameters of the PID controller of the specified file
% inside this program using Ziegler-Nichols or modified ZN or
% Shen and Yu's tuning rules.
%
% Version 0.452
% Written by L. Deva
%
% Sequential Relay Feedback Tuning for 2x2 systems
% (easily upgradable to mxm systems or DRF or IRF).
% Requires: srf2.mat
load srf2.mat
zer=0;
mone=-1;
disp('Simulink model must be open. ')
answ0=input('To tune the system, write the name of the file as a string: ');
disp(' ')
disp('Choose 1 to tune according Ziegler-Nichols and modified ZN PID rules, ')
disp('Choose 2 to tune according Ziegler-Nichols and modified ZN PI rules, ')
disp('Choose 3 to tune according Shen and Yu and modified SY PI rules, ')
answ2=input('Which one? ');
disp(' ')
for i=1:2
ro(i)=str2num(get_param([answ0 s02 s06 num2str(i)],'On_output_value'));
rf(i)=str2num(get_param([answ0 s02 s06 num2str(i)],'Off_output_value'));
end
disp(' ')
for i=1:2
d(i)=(abs(ro(i))+abs(rf(i)))/2;
disp(['Amplitude of the relay ' num2str(i) ' is ' num2str(d(i))])
end
disp(' ')
for i=1:2
        set_param([answ0 s03 s05 num2str(1)],s10,ls2i(i,1));
        set_param([answ0 s03 s05 num2str(2)],s10,ls2i(i,2));
disp(i)
disp('Run simulation in pause mode.')
disp('Press enter when done. ')
pause
disp(' ')
        clear y
        clear yout
        yout=[y1 y2];
y=yout(:,i);
%plot(t,y)
[amp,Pul]=ampper(t,y,answ0);
w(i)=2*3.1416/Pul;
end
for i=1:2
disp(['Frequency of the loop ' num2str(i) ' is ' num2str(w(i)) ' rad/s'])
end
disp(' ')
[ws,wpo]=sort(w);
% 2 shows that there are 2 frequencies...
wpos(1:2)=wpo(2:-1:1);
disp(['Tuning loop ' num2str(wpos(1)) ' first !!!'])
disp(' ')
rpx2=rp2;
lsx2=ls2;
for i=1:2;
 rp2(:,i)=rpx2(:,wpos(i));
 ls2(:,i)=lsx2(:,wpos(i));
end
j=0;
```

```
answ1=1;
 while answ1~=0;
   j=j+1;
                    for k=1:2
                    set_param([answ0 s01 s05 num2str(k)],s10,zer);
                    end
   for i=1:3;
          ii=3*j+i-3;
          set_param([answ0 s02 s05 num2str(1)],s10,rp2(ii,1));
          set_param([answ0 s02 s05 num2str(2)],s10,rp2(ii,2));
          set_param([answ0 s03 s05 num2str(1)],s10,ls2(ii,1));
          set_param([answ0 s03 s05 num2str(2)],s10,ls2(ii,2));
    if i~=3
          disp(' ')
          disp('Run simulation in pause mode before tuning...')
          disp('Then strike any key when done.')
          disp(' ')
          pause
          clear y
          clear yout
          yout=[y1 y2];
          y=yout(:,wpos(i));
%         plot(t,y)
          [Au,Pu]=ampper(t,y,answ0);
          disp(['Tuning loop with number ' num2str(wpos(i))])
          disp(' ')
          Ku=4*d(wpos(i))/3.1416/Au;
          w(wpos(i))=2*3.1416/Pu;
%         CONTROLLER TUNING
           if answ2==1
                  K=0.6*Ku;
                  Ti=Pu/2;
                  Td=Pu/8;
           end
           if answ2==2
                  K=0.45*Ku;
                  Ti=Pu/1.2;
                  Td=0;
           end
           if answ2==3
                  K=Ku/3;
                  Ti=Pu/0.5;
                  Td=0;
           end
         if sign(ro(wpos(i)))==-1;
         K=-1*K;
         end
         prop=K;
         inte=K/Ti;
         drvt=K*Td;
         disp(['PID-' num2str(wpos(i)) ' parameters are: Proportional... ' num2str(prop)])
         disp(['                  Integral....... ' num2str(inte)])
         disp(['                  Derivative..... ' num2str(drvt)])
         f=input('Enter a detuning factor (1 or greater than 1) ');
         disp(' ')
         disp('Detuned settings ')
         prop=prop/f;
         inte=inte/f;
         drvt=drvt/f;
         disp(['PID-' num2str(wpos(i)) ' parameters are: Proportional... ' num2str(prop)])
         disp(['                  Integral....... ' num2str(inte)])
         disp(['                  Derivative..... ' num2str(drvt)])
         set_param([answ0 s02 s07 num2str(wpos(i)) s15],s14,prop);
         set_param([answ0 s02 s07 num2str(wpos(i)) s16],s12,inte);
         set_param([answ0 s02 s07 num2str(wpos(i)) s17],s14,drvt);
         disp('Press any key... ')
 end
         if i==3;
         disp(' ')
         disp('Now you can simulate in pause mode, ')
         disp('Enter setpoint values and simulate,')
         disp('Press any key when you are finished with looking at the graphs... ')
         disp(' ')
         pause
```

```
            end
      end
         answ1=input('Do you want to tune again? (1:yes, 0:no) ');
            disp(' ')
   end
```

The μ-optimal tuning program for a 2x2 process:

```
% SSV-optimal tuning program for Wood and Berry
options=zeros(1,18);
options(1)=1;
options(2)=1;
options(3)=1;
options(4)=.1;
options(14)=250;
options(16)=.1;
options(17)=10;
x(1)=kc(1); x(2)=kc(2); x(3)=ti(1); x(4)=ti(2);
x=constr('muopi',x,options,[0 0 10 10],[2 10 500 500],[],num,den,td,w);
kc(1)=x(1); kc(2)=x(2);
ti(1)=x(3); ti(2)=x(4);
```

The "muopi" part of the μ-optimal tuning program for a 2x2 process:

```
function[Fmu,Gconst] = muopi(x,num,den,td,w)
%   Parameters
x
kc(1)=x(1); kc(2)=x(2);
ti(1)=x(3); ti(2)=x(4);
%Weights
numwi=[1 0.2;0 0;0 0; 1 0.2];
denwi=[.5 1;0 1;0 1; .5 1];
numwp=[0.5 0.02;0 0;0 0;0.5 0.02];
denwp=[1 0;0 1;0 1;1 0];
% Initializing and clearing
Nfrq=[];
Nfr=[];
Nf=[];
ome=w;
% Creating the frequency response of the plant
          Kg=[kc(1);0;0;kc(2)];
          Tin=[ti(1) 0;0 1;0 1;ti(2) 0];
   [m,n] = size(den);
   g = zeros(length(ome),m);
   s = sqrt(-1)*ome(:);
      for i = 1:m
       g(:,i) = polyval(num(i,:),s);
       g(:,i) = g(:,i).*exp(-s*td(i));
       g(:,i) = (polyval(den(i,:),s).\g(:,i));
           k(:,i) = polyval(Tin(i,:),s);
           k(:,i) = 1+1./k(:,i);
           k(:,i) = (polyval(Kg(i,:),s).*k(:,i));
       wi(:,i) = polyval(numwi(i,:),s);
       wi(:,i) = (polyval(denwi(i,:),s).\wi(:,i));
       wp(:,i) = polyval(numwp(i,:),s);
       wp(:,i) = (polyval(denwp(i,:),s).\wp(:,i));
      end
         I=eye(m/2);
   for i = 1:length(ome)
     G = [g(i,1) g(i,2); g(i,3) g(i,4)];
     K = [k(i,1) k(i,2); k(i,3) k(i,4)];
          Wi= [wi(i,1) wi(i,2); wi(i,3) wi(i,4)];
          Wp=    [wp(i,1) wp(i,2); wp(i,3) wp(i,4)];
          GKinv = inv(I+G*K);
     KGinv = inv(I+K*G);
       Nfr = [-Wi*K*G*KGinv -Wi*K*GKinv; Wp*G*KGinv Wp*GKinv];
```

```
            Nfrq = [Nfrq;Nfr];
        end
    % Creating compatible input matrix for mu.m
    omp=ome';
    Nf=[Nfrq;zeros(1,m)];
    [m1,n1]=size(Nf);
    Nf(m1,n1)=length(ome);
    addfrq=[omp;zeros(m1-length(ome),1)];
    Nf=[Nf addfrq];
    Nf(m1,n1+1)=inf;
    % mu for RP
    blk=[1 1;1 1;2 2];
    [mubnds,rowd,sens,rowp,rowg]=mu(Nf,blk,'c');
    muRP=sel(mubnds,':',1);
    maxmu=pkvnorm(muRP)
    % Function to be minimized
    Fmu = maxmu^3;
    % worst-case weighted sensitivity
    %[delworst,muslow,musup]=wcperf(Nf,blk,1);
    %musup
    % mu for RS
    Nrs=sel(Nf,1:2,1:2);
    [mubnds,rowd,sens,rowp,rowg]=mu(Nrs,[1 1;1 1],'c');
    muRS=sel(mubnds,':',1);
    % RS constraint
    Gconst(1) = pkvnorm(muRS)-1;
```

The μ-interaction measure program for a 2x2 process:

```
% MU INTERACTION MEASURE *** 2x2 ***
% numwi,denwi,numwp,denwp: weight info;
% w:frequencies created with logspace command,
%
% **** SPECIFY BLOCK STRUCTURE FOR MU ANALYSIS MANUALLY in this m file. ****
% Use kolonxx.m files to load system information needed.
% Usage: (Specify system transfer functions,
%         specify controller settings,)
%         specify input weights,
%         specify output weights,
%         run program.
%     (Num and Den contains numerators and denominators of the
%         transfer functions in the following order :
%         row 1 -> (1,1) ; row 2 -> (1,2) ; row 3 -> (2,1) ; row 4 -> (2,2)
%         Td is a vector containing time delays in the same order as above
%         w is the logarithmically spaced frequency points created by
%         logspace command.
%         kc and ti are 2x1 vectors containing the gains and integral times for
%         each loop controller.)
%
% Written by E. L. Deva - 1999
% Version: 0.09
disp('You must have run kolonXX.m to obtain a system and controller settings.')
disp('You may change w by using logspace command.')
disp('Press any key to continue')
pause
% Initializing and clearing
svfrq=[];
svfr=[];
svf=[];
svdiag=[];
Nfrq=[];
Nfr=[];
Nf=[];
Ediag=[];
ome=w;
% Creating the frequency response of the plant
        Kg=[kc(1);0;0;kc(2)];
        Tin=[ti(1) 0;0 1;0 1;ti(2) 0];
    [m,n] = size(den);
    g = zeros(length(ome),m);
    s = sqrt(-1)*ome(:);
```

```
for i = 1:m
  g(:,i) = polyval(num(i,:),s);
  g(:,i) = g(:,i).*exp(-s*td(i));
  g(:,i) = (polyval(den(i,:),s).\g(:,i));
       k(:,i) = polyval(Tin(i,:),s);
       k(:,i) = 1+1./k(:,i);
       k(:,i) = (polyval(Kg(i,:),s).*k(:,i));
end
    I=eye(m/2);
for i = 1:length(ome)
            i
  G = [g(i,1) g(i,2); g(i,3) g(i,4)];
  K = [k(i,1) k(i,2); k(i,3) k(i,4)];
       GKinv = inv(I+G*K);
  KGinv = inv(I+K*G);
       Gdiag = [G(1,1) 0;0 G(2,2)];
       E = (G-Gdiag)*inv(Gdiag);
       Te = G*K*GKinv;
       Tdiag = [Te(1,1) 0;0 Te(2,2)];
%      muofE = osborne(E,[1 1]');
       svTdiag = svd(Tdiag);
       svfrq = [svfrq;svTdiag(1)];
       Nfrq=[Nfrq;E];
end
% Creating compatible input matrix for mu.m
omp=ome';
Nf=[Nfrq;zeros(1,2)];
[m1,n1]=size(Nf);
Nf(m1,n1)=length(ome);
addfrq=[omp;zeros(m1-length(ome),1)];
Nf=[Nf addfrq];
Nf(m1,n1+1)=inf;
% Calculating 1/mu(E)
[mubnds,rowd,sens,rowp,rowg]=mu(Nf,[1 1;1 1],'c');
muuu=sel(mubnds,':',1);
[m2,n2]=size(muuu);
oneovermuofE=1./(muuu);
loglog(w,svfrq,w,oneovermuofE(1:m2-1,1))
```

REFERENCES

Aström, K. J., "Theory and Applications of Adaptive Control- A Survey," *Automatica*, Vol. 19, No. 5, pp. 471-486,1983.

Aström, K. J., and T. Hagglund, "Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins," *Automatica*, Vol. 20, No. 5, pp. 645-651, 1984.

Aström, K. J., and T. Hagglund, *PID Controllers: Theory, Design, and Tuning, Second Edition*, Instrument Society of America, Research Triangle Park, NC, 1985.

Aström, K. J., and A. B. Ostberg, "A Teaching Laboratory for Process Control," *IEEE Control Systems Magazine*, Vol. 6, pp. 37-42, 1986.

Aström, K. J., and T. Hagglund, *Automatic Tuning of PID Controllers*, Instrument Society of America, U.S.A., 1988.

Atherton, D. P., and K. Warwick, *Implementation of Self-tuning Controllers*, Peter Peregrinus Ltd., London, 1988.

Beaven, R. W., M. T. Wright, and D. R. Seaward, "Weighting Function Selection in the H_∞ Design Process," *Control Eng. Practice*, Vol.4, No. 5, pp. 625-633, 1996.

Bernstein, D. S., "Sequential Design of Decentralized Dynamic Compensators Using the Optimal Projection Equations," *Int. Journal of Control*, Vol. 46, p. 1569, 1987.

Bhalodia, M., and T. W. Weber, "Feedback Control of a Two-Input, Two-Output Interacting Process," *Ind. Eng. Chem. Process Des. Dev.*, Vol. 18, p. 599, 1979.

Bialkowski, W. L., "Dreams vs. Reality: A View from Both Sides of the Gap," in *Control Systems '92*, Whistler, B. C., Canada, 1992.

Bristol, E. H., "On a New Measure of Interactions for Multivariable Process Control," *IEEE Transactions on Automatic Control*, Vol. 11, pp. 133-134, 1966.

Chang, R., S. Shen, and C. Yu, "Derivation of Transfer Function from Relay Feedback Systems," *Ind. Eng. Chem. Res.*, Vol. 31, pp. 855-860, 1992.

Chiu, M., and Y. Arkun, "A Methodology for Sequential Design of Robust Decentralized Control Systems," *Automatica*, Vol. 28, No. 5, pp. 997-1001, 1992.

Doukas, N., and W. L. Luyben, "Quadruple Distillation Column," *Instrum. Technol.*, Vol. 25, p. 43, 1978.

Ender, D. B., "Process Control Performance: Not as Good as You Think," *Control Eng.*, Vol. 40, pp. 180-190, 1993.

Hang, C. C., A. P. Loh, and V. U. Vasnani, "Relay Feedback Auto-tuning of Cascade Controllers," *IEEE Trans. on Control Systems Technology*, Vol. 2, No. 1, pp. 42-45, 1994.

Hovd, M., and S. Skogestad, "Simple Frequency-dependent Tools for Control System Analysis, Structure Selection and Design," *Automatica*, Vol. 28, No. 5, pp. 989-996, 1992.

Johansson, K. H., B. James, G. F. Bryant, and K. J. Aström, "Multivariable Controller Tuning", 1997, *http://www.control.lth.se/publications/fulldocs/joh97dis.pdf*

Johansson, L. and H. Koivo, "Inverse Nyquist Array Technique in Design of a Multivariable Controller for a Solid Fuel Boiler," *Int. J. Control*, Vol. 25, pp. 1077-1088, 1984.

Lee, J. H., R. D. Braatz, M. Morari, and A. Packard, "Screening Tools for Robust Control Structure Selection," *Automatica*, Vol. 31, No. 2, pp.229-235, 1995.

Li, W. L., and J. H. Lee, "Frequency-Domain Closed-Loop Identification of Multivariable Systems for Feedback Control," *AIChE Journal*, Vol. 42, No. 10, pp. 2813-2827, 1996.

Li, W., Eşkinat, E., and W. L. Luyben, "An Improved Autotune Identification Method," *Ind. Eng. Chem. Res.*, Vol. 30, pp. 1530-1541, 1991.

Loh, A. P., C. C. Hang, C. K. Quek, and V. U. Vasnani, "Autotuning of Multiloop Proportional-Integral Controllers Using Relay Feedback," *Ind. Eng. Chem. Res.*, Vol. 32, pp. 1102-1107, 1993.

Luyben, W. L., "Simple Method for Tuning SISO Controllers in Multivariable Systems," *Ind. Eng. Chem. Process Des. Dev.*, Vol. 25, pp. 654-660, 1986.

Luyben, W. L., "Derivation of Transfer Functions for Highly Non-linear Distillation Columns," *Ind. Eng. Chem. Res.*, Vol. 26, p. 2490, 1987.

Maciejowski, J. M., *Multivariable Feedback Design*, Addison-Wesley, Reading, MA, 1989.

Mario-Galarraga, M., T. J. McAvoy, and T. E. Martin, "Short-Cut Operability Analysis. 2. Estimation of f_i Detuning Parameter for Classical Control Systems," *Ind. Eng. Chem. Res.*, Vol. 26, pp. 511-521, 1987.

Mayne, D. Q., "The Design of Linear Multivariable Systems," *Automatica*, Vol. 9, pp. 201-207, 1973.

Mayne, D. Q., "Sequential Design of Linear Multivariable Systems," *Proc. IEE*, Vol. 126, No. 6, pp. 568-572, 1979.

Morari, M., and E. Zafiriou, *Robust Process Control*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

Musch, H. E., and M. Steiner, "Robust PID Control for and Industrial Distillation Column," *IEE Control Systems,* August 1995.

O'Reilly, J., and W. E. Leithead, "Multivariable Control by Individual Channel Design," *Int. J. Control,* Vol. 54, p. 1, 1991.

Ogunnaike, B. A., and W. H. Ray, "Multivariable Controller Design for Linear Systems Having Multiple Time Delays," *AIChE Journal,* Vol. 25, p. 1043, 1979.

Palmor, Z. J., Y. Halevi, and N. Krasney, "Automatic Tuning of Decentralized PID Controllers for TITO Processes," *Automatica,* Vol.31, No: 7, pp. 1001-1010, 1995.

Shei, T. S., "A Method for Closed Loop Automatic Tuning of PID Controllers," *Automatica,* Vol. 28, No.3, pp. 587-591, 1992.

Shen, S., and C. Yu, "Use of Relay-Feedback Test for Automatic Tuning of Multivariable Systems," *AIChE Journal,* Vol. 40, No. 4, pp. 627-646, 1994.

Skogestad, S., and P. Lundström, E. Jacobsen, "Selecting the Best Distillation Control Configuration," *AIChE Journal,* Vol. 33, No. 5, pp. 753-764, 1990.

Skogestad, S., *Multivariable Feedback Control: Analysis and Design,* John Wiley & Sons, England, 1996.

Tan, L. Y., and T. W. Weber, "Controller Tuning of a Third Order Process under Proportional-Integral Control," *Ind. Eng. Chem. Process Des. Dev.*, Vol. 24, p. 1155, 1985.

Wang, Q., B. Zou, T. Lee, and Q. Bi, "Auto-tuning of Multivariable PID Controllers from Decentralized Relay Feedback," *Automatica,* Vol. 33, No. 3, pp. 319-330, 1997.

Wood, R. K., and M. W. Berry, "Terminal Composition Control of a Binary Distillation Column," *Chem. Eng. Sci.*, Vol. 2, p. 1707, 1973.

Yamamoto, S., and I. Hashimoto, "Present Status and Future Needs: The View from Japanese Industry," *Chemical Process Control- CPCIV: Proceedings of the Fourth International Conference on Chemical Process Control*; Arkun, Y., Ray, W. H., Eds., *AIChE*, NY, 1991.

Ziegler, J. G., and N. B. Nichols, "Optimum Settings for Automatic Controllers," *Transactions of ASME*, Vol. 65, pp. 433-444, 1942.

REFERENCES NOT CITED

Akesson, M., E. Gustafson, and K. H. Johansson, "Control Design for a Helicopter Lab Process," in IFAC'96, *Preprints 13th World Congress of IFAC*, San Francisco, CA, 1996.

Aström, K. J., and M. Lundh, "Lund Control Program Combines Theory with Hands-on Experience," *IEEE Control Systems Magazine*, Vol. 12, pp. 22-30, 1992.

Aström, K. J., C. C. Hang, P. Persson, and W. K. Ho, "Towards Intelligent PID Control," *Automatica*, Vol. 28, No. 1, pp. 1-9, 1992.

Aström, K. J., and B. Wittenmark, *Adaptive Control*, Addison-Wesley Publ., U.S.A., 1995.

Bryant, G. F., and L. F. Yeung, *Multivariable Control System Design Techniques: Dominance and Direct Methods*, Wiley, 1996.

Cluett, W. R., and E. Goberdhansingh, "Autotuning for Model-based Predictive Control," *Automatica*, Vol. 26, No. 4, pp. 691-697, 1990.

Davison, E. J., "Description of Multivariable Apparatus for Real Time-control Studies," *Technical Report 8514a*, Dept. of Electrical Engineering, Univ. of Toronto, Canada, 1985.

Desbiens, A., A. Pomerlau, and D. Hodouin, "Frequency Based Tuning of SISO Controllers for Two-by-Two- Processes," *IEE Proc. Control Theory Appl.*, Vol. 143, No. 1, pp. 49-56, 1996.

Devanathan, R., *Intelligent Tuning and Adaptive Control*, Pergamon Press, England, 1991.

Freudenberg, J., and D. Looze, *Frequency Domain Properties of Scalar and Multivariable Feedback Systems*, Springer Verlag, Berlin, Germany, 1988.

Friman, M., and K. V. Waller, "Autotuning of Multiloop Control Systems," *Ind. Eng. Chem. Res.*, Vol. 33, pp. 1708-1717, 1994.

Gawthrop, P. J., and P. E. Nomikos, "Automatic Tuning of Commercial PID Controllers for Single-Loop and Multi-Loop Applications," *IEEE Control Systems Magazine*, Vol. 10, No. 1, pp. 34-42, 1990.

Goodwin, G. C., "Defining the performance Envelope in Industrial Control," in *16th American Control Conference*, Albuquerque, NM. Plenary Session I, 1997.

Hagglund, T., and K. J. Aström, "Industrial Adaptive Controllers Based on Frequency Response Techniques," *Automatica*, Vol. 27, No. 4, pp. 599-609, 1991.

Hang, C. C., K. J. Aström, and W. K. Ho, "Relay Auto-tuning in the Presence of Static Load Disturbance," *Automatica*, Vol. 29, No. 2, pp. 563-564, 1993.

Harris, C. J., and S. A. Billings, *Self-tuning and Adaptive Control: Theory and Applications*, Peter Peregrinus Ltd., London, 1985.

Ho, W. K., C. C. Hang, and L. S. Cao, "Tuning of PID Controllers Based on Gain and Phase Margin Specifications," *Automatica*, Vol. 31, No. 3, pp. 497-502, 1995.

Ho, W. K., T. H. Lee, and O. P. Gan, "Tuning of Multiloop Proportional-Integral-Derivative Controllers Based on Gain And Phase Margin Specifications," *Ind. Eng. Chem. Res.*, Vol. 36, *pp.* 2231-2238, 1997.

Hsu, L., M. Chan, and A. Bhaya, "Automated Synthesis of Decentralized Tuning Regulators for Systems with Measurable DC Gain," *Automatica*, Vol. 28, No. 1, pp. 185-191, 1992.

Jussila, T. T., and H. N. Koivo, "Tuning of Multivariable PI-Controllers for Unknown Delay-Differential Systems," *IEEE Transactions on Automatic Control*, Vol. AC-32, No. 4, 1987.

Koivo, H. N., and S. Pohjolainen, "Tuning of Multivariable PI-Controllers for Unknown Systems with Input Delay," *Automatica*, Vol. 21, No. 1, pp. 81-91,1985.

Koivo, H. N., and J. T. Tanttu, "Tuning of PID Controllers: Survey of SISO and MIMO Techniques," in R. Devanathan (editor), *IFAC Intelligent Tuning and Adaptive Control*, pp. 75-80, Pergamon Press, England, 1991.

Mansour, M., and W. Schaufelberger, "Software and Laboratory Experiment Using Computers in Control Education," *IEEE Control Systems Magazine*, Vol. 9, No: 3, pp. 19-24, 1989.

Monica, T. J., C. C. Yu, and W. L. Luyben, "Improved Multiloop Single-Input/Single-Output (SISO) Controllers for Multivariable Processes," *Ind. Eng. Chem. Res.*, Vol. 30, pp. 1530-1541, 1991.

Morris, H. M., "How Adaptive Are Adaptive Process Controllers ?," *Control Engineering*, Vol. 34, No. 3, pp. 96-100, 1993.

Munro, M., *Modern Approaches to Control System Design*, Peter Peregrinus Ltd., England, 1979.

Niederlinski, A. "A Heuristic Approach to the Design of Linear Multivariable Interacting Control Systems," *Automatica*, Vol. 7, pp. 691-701, 1971.

Nishikawa, Y., N. Sannomiya, T. Ohta, and H. Tanaka, "A Method for Auto-tuning of PID Control Parameters," *Automatica*, Vol. 20, No.3, pp. 321-332, 1984.

Özdaş, M. N., A. T. Dinibütün, and A. Kuzucu, *Otomatik Kontrol Temelleri*, Birsen Yayinevi, Istanbul, 1995.

Seborg, D. E., T. F. Edgar, and D. A. Mellichamp, *Process Dynamics and Control*, Wiley, NY, 1989.

Seron, M. M., J. H. Braslavsky, and G. C. Goodwin, *Fundamental Limitations in Filtering and Control*, Springer Verlag, 1997.

Shinskey, F. G., *Controlling Multivariable Processes*, Instrument Society of America, Research Triangle Park, NC, 1981.

Shiu, S., and S. Hwang, "Sequential Design Method for Multivariable Decoupling and Multiloop PID Controllers," *Ind. Eng. Chem. Res.*, Vol. 37, pp. 107-119, 1998.

Stein, G., "Respect the Unstable," in *30th IEEE Conference on Decision and Control*, Honolulu, HI, 1990.

Toh, K. A., and R. Devanathan, "An Expert Autotuner for Multiloop SISO Controllers," *Control Engineering Practice*, Vol. 1, No. 6, pp. 999-1008, 1993.

Vasnani, V. U., "Towards Relay Feedback Auto-tuning of Multiloop Systems," *PhD Thesis*, National University of Singapore, 1994.

Visher, D., and H. Bleuler, "A New Approach to Sensorless and Voltage Controlled AMBs Based on Network Theory Concepts," *in 2nd International Symposium on Magnetic Bearings*, Institute of Industrial Science, Tokyo University, 1990.

Wang, Q., "Decoupling with Internal Stability for Unity Output Feedback Systems," *Automatica*, Vol. 28, No. 2, pp. 411-415, 1992.

Wellstead, P. E., and M. B. Zarrop, *Self-tuning Systems: Control and Signal Processing*, John Wiley & Sons, England, 1995.

Zgorzelski, P., H. Unbehauen, and A. Niederlinski, "A New Simple Decentralized Adaptive Multivariable Regulator and its Application to Multivariable Plants," in *IFAC 11th Triennial World Congress*, pp. 381-386, Talinn, Estonia, 1990.

Zuhang, M., and D. P. Atherton, "PID Controller Design for a TITO System," *IEE Proc. Control Theory Appl.*, Vol. 141, No. 2, pp. 111-120, 1994.