On-Line Elicitation of Mamdani-Type Fuzzy Rules via TSK-Based Generalised Predictive Control

M. MAHFOUF [*] , M.F. ABBOD AND D.A. LINKENS

[*] *Corresponding Author*

Department of Automatic Control and Systems Engineering

The University of Sheffield

Mappin Street, Sheffield S1 3JD

The United Kingdom

Tel: +44 114 222 5607

Fax: +44 114 222 5661

Email: M.Mahfouf@Sheffield.ac.uk

Abstract

Many synergisms have been described in the past between soft computing techniques such as Neural Networks (NN), Fuzzy Logic (FL) and Genetic Algorithms (GA) which have not only shown that such hybrid structures can work well but also add more robustness to the control system under consideration. In this paper, a new control architecture is proposed whereby the

on-line generated fuzzy rules relating to the Self-Organising Fuzzy Logic Controller (SOFLC) are obtained via integration with the popular Generalised Predictive Control (GPC) algorithm using a Takagi-Sugeno Kang (TSK) based CARIMA model structure. In this approach, GPC replaces the Performance Index (PI) table which, as an incremental model, is traditionally used to find new rules, delete rules and amend existing rules. Because the GPC sequence is computed using predicted future outputs, the new hybrid approach rewards the time-delay very well. The new generic approach, named Generalised Predictive Self-Organising Fuzzy Logic Control (GPSOFLC), is applied to a well-known non-linear chemical process, the distillation column, and is shown to lead to the elicitation of an effective fuzzy rule-base in both qualitative and quantitative terms.

1. Introduction

It is common knowledge that in the last 40 years most control designs have been based on linear models, despite the fact that most systems are inherently nonlinear, due to the difficulty associated with analysing nonlinear systems. One common approach has been to linearise the system around various operating points. Among the most popular model-based control approaches are three term PID controllers, Model-Based Predictive Control (MBPC) and robust control (H_∞). However, in recent years, there has been a move among system engineers towards intelligent control with a qualitative dimension due to the widespread dissatisfaction with quantitative engineering. One of the main attractions of intelligent system design is the possibility of multivariable system control without the need for extensive dynamic models of the process. The main difficulty in the multivariable case is the interaction between variables together with sensitivity to faults in various channels. Neural Networks (NN) [1], Fuzzy Logic Control (FLC) [2], [3], and Genetic Algorithms (GA) have been at the forefront of such methodologies and have proved to be strong contenders for other forms of control.

Various synergisms have been described between fuzzy logic, neural networks and genetic algorithms [4], [5] which not only showed that these intelligent structures can interact together but also can make the resultant overall structure more robust against model uncertainties as well as disturbances. In this present work, we show that a synergism between a mathematical model-based approach in the form of self-tuning Generalised Predictive Control (GPC) [6] and Self-Organising Fuzzy Logic Control (SOFLC) [7] is possible with the former being used as a mechanism to make the latter adjust itself to improve the overall system's performance. As will be expanded in later sections, both GPC and SOFLC are multi-level control designs; the former relying on a mathematical model formulation to derive the future control moves, whereas the latter is purely linguistic and relies on a fuzzy linguistic table (the Performance Index-PI) to issue control corrections to the low-level layer to allow it to adjust itself. In this paper the proposed architecture consists of replacing the above Performance Index (PI) table with the computed GPC moves as *the* corrections which will allow the generation of new fuzzy rules, deletion of redundant fuzzy rules, and alteration of the existing fuzzy rules. Both paradigms are adaptive (GPC runs with a Recursive Least-Squares- RLS estimation algorithm and in SOFLC the PI organises the fuzzy rules) and have been shown to work on a variety of systems, including distillation columns [8] and muscle relaxation [9], both systems being characterised by time-delays and nonlinearities. In addition, the GPC algorithm has been modified to reflect a fuzzy Takagi-Sugeno-Kang (TSK) Controlled Autoregressive Integrated Moving Average (CARIMA) model structure [10].

The paper is organised as follows: in Section 2 the TSK-based GPC and SOFLC algorithms are reviewed briefly. Section 3 presents the new modified algorithm which we call Generalised Predictive Self-Organising Fuzzy Logic Control (GPSOFLC). Simulation results using a distillation column are presented and discussed in Section 4. Finally, in Section 5, conclusions are drawn with regard to the new proposed algorithm.

2. Theoretical Background relating to TSK-based GPC and SOFLC

2.1 The Fuzzy TSK-based Generalised predictive Control Algorithm

One common factor in all Model Based Predictive Control strategies [11], which represents their *"raison d'etre"*, is their assumption of a model which has to be quite accurate. The modelling of real world systems, however, often presents problems. As processes increase in complexity, they become less amenable to direct mathematical modelling based on physical laws since they may be distributed, stochastic, non-linear and time-varying, uncertain, etc. According to Zadeh's Principle of Incompatibility [12], the closer one looks at a real world problem, the fuzzier becomes the solution.

An alternative to modelling the operator’s response is to use a fuzzy logic system to provide a computing paradigm for modelling the non-linear process dynamics when a sufficiently accurate model of the process to be controlled is unavailable. The modelling problem, instead of being posed within a strictly analytical framework, is based on empirically acquired knowledge regarding the operation of the process.

An alternative method of expressing fuzzy rules was proposed by Takagi and Sugeno [13] which include fuzzy sets only in the premise part and a regression[1] model as the consequent, i.e.

$$IF\ x_1\ is\ B^1\ and\ \ldots and\ x_n\ is\ B^n\ \ THEN\ \ y = c_0 + c_1 x_1 + \cdots + c_n x_n \qquad (1)$$

where $\underline{x} = (x_1, \ldots, x_n)^T$ and y are the input and output linguistic variables respectively, B^i are linguistic values characterised using membership functions, and c_i are real-valued parameters. It is considered that this fuzzy rule representation provides a convenient framework to incorporate human experts' knowledge.

A complex high dimensional non-linear modelling problem is decomposed into a set of simpler linear models valid within certain operating regimes defined by fuzzy boundaries. Fuzzy inference is then used to interpolate the outputs of the local models in a smooth fashion to get a global model. This modelling approach provides a good modelling accuracy [13], [14] and is free of the problems arising from model incompleteness. Also, as demonstrated by Takagi and Sugeno, standard identification

---

[1] This model can be either linear or non-linear.

techniques such as the method of least squares can be easily applied for determining the model parameters.

2.1.1 Fuzzy Process Model

Consider a single input single output (SISO) system which can be modelled using the method proposed by Takagi and Sugeno. Assuming that the input space is partitioned using *p* fuzzy partitions and that the system can be represented by fuzzy implications (one in each fuzzy sub-space), we can write the following [10]:

$$\begin{aligned} L^i : IF\ y(t)\ is\ B^i\ THEN\ \Delta y_m(t+1) = -a_1^i \Delta y(t) - \cdots - a_j^i \Delta y(t-n_a+1) \\ + b_1^i \Delta u(t) + \cdots + b_l^i \Delta u(t-n_b+1) \end{aligned} \qquad (2)$$

where $y(t)$ and $u(t)$ are the process and controller outputs at time t, $y_m(t+1)$ is the 1-step ahead model prediction at time t, B^i is a fuzzy set representing the fuzzy sub-space in which implication L^i can be applied for reasoning, and $\Delta = 1 - z^{-1}$, with z^{-1} being the backward shift operator.

Such model representation in the consequent part of the above implication is called a CARIMA[2] structure [16] which was found to be effective against offsets which can be present in the data.

[2] Controlled Auto-Regressive Integrated Moving Average

The model parameters can be expressed in the following matrix form:

$$\Theta = \begin{bmatrix} a_1^{\,1} \ldots a_{n_a}^{\;1} & b_1^{\,1} \ldots b_{n_b}^{\;1} \\ \vdots & \vdots \\ a_1^{\,p} \ldots a_{n_a}^{\;p} & b_1^{\,p} \ldots b_{n_b}^{\;p} \end{bmatrix} \tag{3}$$

The overall fuzzy model output (in incremental form) can be written as follows:

$$\Delta y_m(t+1) = \Theta' \, \Phi(t) \tag{4}$$

where,

$$\Phi(t) = \left[-\Delta y(t),\ -\Delta y(t-1), \cdots, -\Delta y(t-n_a+1), \Delta u(t), \Delta u(t-1), \cdots, \Delta u(t-n_b+1)\right]^T \tag{5}$$

Θ' represents a matrix of the β_i – weighted parameters of Θ

$$\beta = \left\lfloor \beta_1 \ \beta_2 \ \cdots \beta_i \ \cdots \beta_p \right\rfloor \tag{6}$$

and,

$$\beta_i = \frac{B^i\left[y(t)\right]}{\sum_{i=1}^{p} B^i\left[y(t)\right]} \tag{7}$$

$B^i\left[y(t)\right]$ is the grade of membership of $y(t)$ in B^i and β is a vector of the weights assigned to each of the p implications at each sampling instant.

It is worth noting that in order to design long-range predictive controllers using an analytical approach, one needs to linearise the above fuzzy model about the current operating point by weighting the fuzzy model parameters at each sampling instant as follows:

$$\Theta' = \begin{bmatrix} a_1'^1 \ldots a_{n_a}'^{\,1} & b_1'^1 \ldots b_{n_b}'^{\,1} \\ \vdots & \vdots \\ a_1'^p \ldots a_{n_a}'^{\,p} & b_1'^p \ldots b_{n_b}'^{\,p} \end{bmatrix} \qquad (8)$$

where Θ' denotes the vector with the weighted fuzzy model parameters such that:.

$$\begin{aligned} & a'^{\,i}_{j} = \beta_i . a_j^{\,i}; \quad b_k'^{\,i} = \beta_i . b_k^{\,i} \\ & i = 1, \ldots, p \\ & j = 1, \ldots, n_a \\ & k = 1, \ldots, n_b \end{aligned} \qquad (9)$$

The above modelling techniques will form the basis for the development within the long-range predictive control strategy.

2.1.2 Controller Formulation

The long-range predictive controller developed in this research study is based on the popular Generalised Predictive Control (GPC) strategy [6] whose theoretical background is briefly reviewed here:

Consider the following locally linearised discrete model in the backward shift operator z^{-1}:

$$A(z^{-1})\,\Delta y(t) = B(z^{-1})\,\Delta u(t-1) + C(z^{-1})\,\zeta(t) \qquad (10)$$

where:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{n_a} z^{-n_a}$$
$$B(z^{-1}) = b_1 + b_2 z^{-1} + b_3 z^{-2} + \ldots + b_{n_b} z^{-n_b+1}$$
$$C(z^{-1}) = c_0 + c_1 z^{-1} + c_2 z^{-2} + \ldots + c_p z^{-p}$$
$$\zeta(t) \textit{ is an uncorrelated random sequence.}$$
$$\Delta = 1 - z^{-1}$$

where $u(t)$ represents the control input and $y(t)$ is the measured variable. The controller computes the vector of controls using optimisation of a function of the form:

$$J_{GPC} = \sum_{j=N_1}^{N_2}\left[(P(z^{-1})\,\hat{y}(t+j) - \omega(t+j))^2\right] + \sum_{j=1}^{NU}\left[\lambda(j)(\Delta u(t+j-1))^2\right] \qquad (11)$$

where N_1 is the minimum costing (output) horizon, N_2 is the maximum costing horizon, NU is the control horizon, ω is the future set-point, $\lambda(j)$ is the control weighting sequence, and $P(z^{-1})$ is the inverse model in the model-following context with $P(1)=1$. Furthermore, the $C(z^{-1})$ polynomial in Equation (11) is replaced by a fixed polynomial $T(z^{-1})$ known as the observer polynomial for the predictions $P(z^{-1})\,\hat{y}(t+j)$.

The minimisation of the cost function described in Equation (11) leads to the following projected control increment:

$$\Delta u(t) = \bar{g}^T(\omega - \Psi), \quad \Psi = [\Psi(t+N_1), ..., \Psi(t+N_2)] \tag{12}$$

where $\bar{g}^T$ is the first row of the matrix $(G_d{}^T G_d + \lambda(j)I)^{-1} G_d{}^T$, G_d is the dynamic (step-response) matrix of the form given in [6], and Ψ is a vector of future output responses weighted by $P(z^{-1})$.

2.1.3 Adaptation Mechanism

For the purpose of application of parameter estimation techniques, equation (4) can be re-formulated in the following form:

$$\Delta y_m(t+1) = \theta(t).\varphi(t) \tag{13}$$

where $\varphi(t)$ is the regressor vector which is determined from $\Delta y(t), \Delta u(t)$ and $\beta(t)$ and is given by:

$$\varphi(t) = \begin{bmatrix} \beta_1 \Delta y(t) \\ \vdots \\ \beta_1 \Delta u(t - n_b + 1) \\ \vdots \\ \beta_p \Delta y(t) \\ \vdots \\ \beta_p \Delta u(t - n_b + 1) \end{bmatrix} \tag{14}$$

and $\theta(t)$ is the parameter vector determined from $\Phi(t)$ and is given by:

$$\theta(t) = \left[a_1^{\ 1} \dots a_{n_a}^{\ 1} \quad b_1^{\ 1} \dots b_{n_b}^{\ 1} \quad \dots \quad a_1^{\ p} \dots a_{n_a}^{\ p} \quad b_1^{\ p} \dots b^p{}_{n_b} \right] \tag{15}$$

The least-squares estimates of model parameters are found by minimising a certain cost function, which is not shown here due to lack of space. It should be stressed that in most circumstances the standard recursive least squares (RLS) in its *UD*-factorisation form is used [17].

Adopting the above control strategy to work with the fuzzy model requires the weighted model parameters determined at each sampling instant from the fuzzy model using the method proposed above, instead of the normal linear CARIMA model parameters. Hence, the overall control strategy consists of the following 5 simple steps at each sampling instant:

a) A process output is measured and fuzzified to obtain the vector of confidences β.

b) Calculate the weighted information vector $\varphi(t)$, using the vector of confidence β and the vector of information $\Phi(t)$ given by Equation (5)

c) Estimate the process parameters $\theta(t)$ using a RLS algorithm.

d) Calculate the weighted model parameters vector Θ' using Equations (8-9).

e) Calculate the optimal controller output sequence $\Delta u(t)$, using Equation (12).

Figure 1 is a schematic diagram representing the structure of the resultant fuzzy TSK-based GPC architecture.

2.2 A Qualitative Dimension from SOFLC

The first Self-Organising Fuzzy Logic Controller was proposed by Procyk and Mamdani [7] and includes a control policy that can change with respect to the process it is controlling and the environment it is operating in. This is what is often called an adaptive or learning controller to stress that its operation relies on the acquisition of past experience, i.e. a suitable combination of past control actions and the effects they produced. One interesting characteristic of this controller is that it strives to improve its performance until there is convergence to a predetermined quality. In doing this, the SOFLC performs two tasks at the same time which are a) observe the environment while ensuing the appropriate control action and b) use the results of these control actions to improve them even further. A considerable amount of work has been carried out using the SOFLC, originally in Queen Mary College, London, but, perhaps, the most interesting applications of the controller is that related to muscle relaxation control [9] and Sugeno's fuzzy car [18]. The car has the ability to learn how to park itself.

2.2.1 Description of the Controller

As illustrated in Figure 2 the SOFLC comprises two levels. The first level consists of a simple fuzzy controller, whereas the second level acts as a monitor and performance evaluator of the previous level and is usually called the self-organising mechanism. It includes four blocks: the performance index, the process reference model, the rules modifier, and the state buffer.

In the first level, the input signal to the controller is taken at each sampling instant in the form of error and change in error. Each signal is mapped to its corresponding discrete level by appropriate scaling factors and then sent to the SOFLC. According to control rules issued by the second level, the SOFLC calculates the output with respect to the inputs. The output signal is scaled to its actual value using the output-scaling factor and afterwards sent to the process being controlled.

The second level is basically the part which realises the adaptation referred to above. Based on the trajectory of the process being controlled, any deviation from the desired path is corrected by modifying the rule responsible for that particular undesirable deviation.

2.2.2 The Performance Index (PI)

The performance index measures the deviation from the path of the desired trajectory and issues the appropriate changes that are required at the output of the controller. It can be written in the form of a “look-up” table. If the antecedents of the performance index rules are the process output error and the change in the process output error, then the credit value of a particular process output *U* at a sample *nT* is given by:

$$P_i(nT) = f(e(t), \dot{e}(t)). \tag{16}$$

$$P_0(nT) = M.P_i(nT) \tag{17}$$

where f is the mapping performed by the Performance Index and M is a reference model [7].

The credit represents the required change in the system output to enable the rules modifications. It is worth noting that the performance rules are written using a qualitative feel for a general monotonic undamped process, and are intended to provide fast convergence coupled with the required damping around the equilibrium state to achieve high accuracy.

2.2.3 The Rules Modification Block

The rules modification procedure can be explained by the following:

Assume that the process has a time lag of m samples. This means that the control action at sampling instant $(nT - mT)$ has most contributed to the process performance at sampling instant nT. For a SISO process, $e(nt - mT)$ and $\dot{e}(nT - mT)$ would have been the error and change in error at that time, E and CE are the fuzzified sets relating to $e(t)$ and $\dot{e}(t)$ respectively, and $U(nT - mT)$ would have been the controller output. Consequently, the controller output that would have been required is $U(nT - mT) + P_0(nT)$ which needs to be scaled and fuzzified, in order to be registered as the new rule. Hence, the rule

$$E(nT - mT) \rightarrow CE(nT - mT) \rightarrow U(nT - mT) \qquad (18)$$

is modified to become:

$$E(nT - mT) \rightarrow CE(nT - mT) \rightarrow V(nT - mT) \qquad (19)$$

where,

$$V(nT - mT) = F\{U(nT - mT) + P_0(nT)\} \quad (20)$$

and F refers to the fuzzification operation.

2.2.4 The State Buffer

For a process with dead time, the control action should be rewarded an adequate amount of time earlier. The state buffer is a first in / first out register (FIFO) which records the values of the scaled error, scaled change in error, and the defuzzified output before scaling, and produces the registered values on the buffer output after a time equal to the Delay-in-Reward parameter.

2.2.5 Delay-in-Reward

The Delay-in-Reward mT is a parameter that reflects the time-delay present in the system. Only rules which include antecedents and consequents delayed by the amount mT must be modified. If this parameter is not taken into account, amplitudes of oscillations around the set-point will be too large for convergence to take place. Hence, this parameter introduces an amount of compensation for phase lag in the system.

As it stands, the SOFLC design includes only rules modification within specific definitions of the peak and width of each fuzzy set. Automatic selection of the scaling factors can also be added to the overall design [9]. For instance, as the process output approaches the set-point, the scaling factors can be chosen so to realise finer tuning.

One criticism, though, of the SOFLC scheme is that it starts from an empty rule-base (although this is not essential), which in sensitive areas of control applications such as medicine and aeronautics will not be favoured. However, because the SOFLC's ethos is to improve its performance as it learns about the process, it is always recommended to conduct "dry" runs on an approximate model of the process, build an initial rule-base and start the SOFLC with that particular initial rule-base in real-time. Moreover, because for the first mT samples, the SOFLC uses the rules directly from the PI table, then it is important to set an adequate reference trajectory, for example via the adaptive fuzzy TSK-based GPC algorithm proposed in this paper. The next Section reviews how this can be achieved within the new algorithm structure.

3 The Generalised Predictive Self-Organising Fuzzy Logic Control (GPSOFLC)

The idea behind the new algorithm lies in substituting a different mechanism for the PI table that will allow the Rules Modifier in Figure 2 to generate new fuzzy rules, delete existing but redundant rules and alter the fuzzy rules in a similar way to the SOFLC described in Section 2 but more effectively. This, we believe, will not only add more robustness to the self-learning architecture but will allow one to elicit, on-line, the fuzzy control rules of a Mamdani-type without the need for an off-line analysis.

The fuzzy rules which are included in the PI table are standard and are used for a wide range of systems and reflect characteristics of a system with adequate damping, overshoot, and settling-time. Since the index $P_i(nT)$ used for the Rules Modifier block is an incremental sequence and represents the 'best' correction that can be issued at time nT then we propose to replace it with the future control moves generated by the fuzzy TSK-based GPC described in Section 2, particularly $\Delta u(nT)$, whose parameters are adjusted via a Recursive Least-Squares (RLS) algorithm. It is worth noting that the adaptive predictive algorithm will only calculate the control increment based on 'gain' information as the fuzzy TSK-based GPC is not explicitly controlling the process. Hence, the RLS-based parameter estimates stem from a closed-loop identification operation which uses input-output data relating to the SOFLC controlling the process (see Figure 3). In all the simulations that we carried-out on a wide range of processes a second-order discrete-time model with no time-delay but using the filter polynomial $T(z^{-1})$ was sufficient to characterise the fuzzy TSK-based GPC layer. Moreover, in the same way in which the performance index output is scaled to be used as the *actual* fuzzy correction (through the block Model in Figure 3) the GPC increment is also scaled to map the real world onto the fuzzy world. Hence, in light of the above considerations Equations (18-20) become:

$$E(nT-mT) \rightarrow CE(nT-mT) \rightarrow U(nT-mT) \tag{21}$$

$$E(nT-mT) \rightarrow CE(nT-mT) \rightarrow V(nT-mT) \tag{22}$$

where,

$$V(nT-mT) = F\{U(nT-mT) + \Delta u(nT)\} \tag{23}$$

The algorithm hence obtained is named Generalised Predictive Self-Organising Fuzzy Logic Control (GPSOFLC). As it stands the algorithm allows transparency and interpretability of the fuzzy rules obtained given the fuzzy partitioning of the input output spaces. It is also worth noting that such fuzzy partitioning (in terms of the number of fuzzy partitions) does not have to be identical for 'the TSK-based GPC' layer and the 'Mamdani-type fuzzy system' layer. Hence, summarising the GPSOFLC algorithm leads to the formulation of the following steps:

1. A process output is measured and fuzzified to obtain the vector of confidences β.

2. Calculate the weighted information vector $\varphi(t)$, using the vector of confidence β and the vector of information $\Phi(t)$ given by Equation (5).

3. Estimate the process parameters $\theta(t)$ using a RLS algorithm.

4. Calculate the weighted model parameters vector Θ' using Equations (8-9).

5. Calculate the first control move $\Delta u(t)$, using Equation (12).

6. $\Delta u(t)$ is first scaled then sent to the 'Rules-Modifier' block to apply the correction (if necessary) to the rule (taking into account the value of the delay-in-reward) which was responsible for the current performance (see Equation (23)).

7. The values of error $e(t)$ and its derivative $\dot{e}(t)$ are used in 'the simple fuzzy logic control' layer to infer the actual process input.

8. All values are shifted back in time ready for the next sample.

The next sections will present and discuss a series of simulations results obtained using the new proposed algorithm on a difficult process, namely a distillation column [8].

4 Simulation Results

The simulation experiments involving the proposed adaptive control will use the binary distillation column as a test-bed. The distillation column is an extensively researched process in control engineering literature as it offers the most known challenges such as nonlinearities, offsets, time-delays, loop interactions, etc.

4.1 The Binary Distillation Column

Distillation is used in many chemical processes to separate feed streams and for purification of final and intermediate product streams. Figure 4 is a schematic representation of a binary distillation column. Feed is separated into an overhead product or "distillate" and a bottoms product or "bottoms". Heat is transferred into the process in the reboiler to vaporise some of the liquid from the base of the column. The vapour coming from the top of the column is liquified in another tube-and-shell heat exchanger called the condenser and liquid from the condenser drops into the reflux drum. The distillate is removed from this drum. In addition, some liquid, called "reflux", is fed back to the top of the column. A mathematical model of a 20-tray binary distillation column is provided in the text by Luyben [8]. A simplified version of this model, which neglects the dynamics introduced by tray fluid mechanics, has been considered in this study.

4.2 The Control Problem

The objective of the controller would be to control the distillate composition by manipulating the reflux flow-rate. Hence, the following fuzzy relations can be used for model identification:

$$\begin{aligned} L^i : IF\ x_D(t)\ is\ B^i\ THEN\ \Delta x_D(t+1) &= -a_1{}^i \Delta x_D(t) - a_2{}^i \Delta x_D(t-1) \\ &+ b_1{}^i \Delta L(t) + b_2{}^i \Delta L(t-1) \end{aligned} \tag{24}$$

where $x_D(t)$ and $L(t)$ are the distillate composition and reflux rate respectively and i is an index referring to the *ith* fuzzy partition.

Different fuzzy partitions (of triangular or Gaussian shapes) of the input space can be used. For the GPC algorithm, the following parameters were chosen to be identical for both channels:

$$N_1 = 1,\ N_2 = 5,\ NU = 3,\ P(z^{-1}) = \frac{(1-0.9z^{-1})}{0.1}$$

$$T_{co}(z^{-1}) = (1-0.7z^{-1})^2\ \ (\ for\ \ control)$$

$$T_{est}(z^{-1}) = \frac{(1-0.7z^{-1})^2}{0.09}(\ for\ \ estimation)$$

$$\lambda(j) = 0.$$

It is worth noting here that the observer polynomial $T(z^{-1})$ is usually taken to be identical for both calculating the control sequence ($T_{co}(z^{-1})$) and the Recursive Least Squares parameter estimation (RLS) ($T_{est}(z^{-1})$), but can sometimes include different dynamics and gains to enhance robustness in either case [9].

For parameter estimation, a second-order model with $2\hat{a}$'s and $2\hat{b}$'s was adopted throughout all experiments with an initial covariance matrix of $\text{cov}_i = 10^3.I$ and a forgetting factor of $\rho = 0.95$, with all parameter estimates initially set to zero except b_1 and b_2 which were set to 1.

The binary distillation column was simulated using the MATLAB-SIMULINK Toolbox with a sampling interval of 0.1 minute. A total time of 250 minutes was used for simulation with the initial set point of 0.98 for the distillate composition corresponding to the tuning phase, and from then on the set point was changed using decrements of 0.01 then was incremented with the same amount again.

For the purpose of this research, the low-level simple fuzzy controller assumed five fuzzy sets; Negative Big *(NB)*, Negative Small *(NS)*, Zero *(Z)*, Positive Small *(PS)*, and Positive Big *(PB)*, which have been defined in a universe linearly. It is worth noting that the choice of the shape relating to the Membership Functions (MF's), in this case Gaussian, is arbitrary and the same MF's can be chosen to be triangular also without any significant effect on performance.

The first experiment considered the use of the standard SOFLC algorithm with the modified Performance Index of Tables 1a,b which uses 5 fuzzy membership functions, in contrast to earlier proposed tables which used up to 13 fuzzy labels [7].

The simulation run produced the performance of Figure 5 where it can be seen that the output tracked the set-point quite well although low-magnitude limit cycles were reached between times 175 and 225 minutes. In turn, the control signal was very active displaying oscillatory modes. The run generated 20 fuzzy rules in the lower level of the control structure with a non-linear control surface of Figure 5c.

The second experiment considered the use of a linear CARIMA partition of the input-output space. The run produced the performance of Figure 6 where it can be seen that the output tracked the reference target very efficiently with a reasonably active control signal despite a control horizon of $NU = 3$. Also, the run generated 18 fuzzy rules out of a maximum of 25 rules with a control surface, shown in Figure 8c, which emphasises a non-linear mapping between inputs and output. Table 2 displays the corresponding fuzzy rules in the lower level obtained at the end of the run which also emphasises the free structure as opposed to the grid-like rule-base.

The third experiment considered the use of a 5-partition fuzzy model. The run produced the performance of Figure 7 which was better than those of Figures 5 and 6, in terms of the overshoots at times 200 and 250 minutes and also in terms of control activity where the ringing modes have been de-emphasised. Similarly to the previous run the number of generated fuzzy rules reached 18 out of a maximum of 25 rules.

The above experiments were conducted with the parameter estimation switched-on all the time which allowed the model to be updated on-line. In order to test the robustness of the proposed algorithm when the model is fixed the parameter estimation was switched off at a time of 150 minutes during the simulation run. Figure 8 shows the performance of the new algorithm when a linear model is considered. The performance obtained is similar to that of Figure 6 except that the number of fuzzy rules generated increased to 19 fuzzy rules (see also Table 3 for the individual rules obtained at the end of the run). When a 5-partition fuzzy TSK model was used the performance of Figure 9 was obtained where it can be seen that an equally good performance in terms of output tracking and control signal activity was obtained with the same number of fuzzy rules being generated as in the case of Figure 7, i.e. 18 rules; this number did not increase due to the fact that the relatively high number of fuzzy partitions compensated for the absence of the on-line parameter estimation, this being in contrast to the run of Figure 8 where the number of rules increased by 1 fuzzy rule.

5 Conclusions

Soft computing which includes the three intelligent systems approaches, neural networks, genetic algorithms, and fuzzy logic theory has been increasingly embraced by a wide section of systems engineers. Various synergisms are known to exist between the above three strands which can work well together by forming structures which can tackle complex problems in a robust way. In this study, a new algorithm was proposed in which a synergism was shown to be possible between a mathematical model-based approach (self-tuning GPC) and an intelligent control system with a qualitative dimension (fuzzy logic control). This substitutes the GPC algorithm for the performance index table used in the standard SOFLC algorithm to modify the rules of the low-level represented by the direct fuzzy controller. Whatever the structure of the process in question, a second order model was found to be sufficient for the GPC structure. This is compatible with the performance index table used in the standard SOFLC which reflects characteristics similar to a second order system with specific damping, overshoot, and settling-time. Using the distillation column as a test bed, the new proposed strategy named Generalised Predictive SOFLC (GPSOFLC) was shown to perform better than the standard SOFLC algorithm and to lead to a more transparent architecture by generating the necessary number of fuzzy control rules to control the process effectively. During an extensive simulation study which included also other processes, such as muscle relaxant anaesthesia, it transpired that the following guidelines for parameter tuning of the algorithm are recommended in order to achieve the best performance:

- A second-order linear or fuzzy TSK model with up to 5 fuzzy partitions is sufficient for the GPC algorithm to generate the necessary control moves for the Rule- Modifier block.

- A high control horizon ($NU \geq 2$) is recommended; this is to enable proper excitation of the 'Rules-Modifier' block.

- The use of a second-order observer polynomial $T(z^{-1})$ is also recommended to compensate for any unmodelled dynamics.

Finally, it is worth noting that these current findings are now the focus of extensive research which aims at demonstrating that superior performance need not be linked with a large number of rules but rather to the quality of these rules. It is planned to extend the work to include multivariable structures.

6 References

[1] K.S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. on Neural Networks*, **1**, 4-27. 1990.

[2] C.C. Lee, "Fuzzy logic in control systems- Part I and II," *IEEE Trans. Syst., Man & Cyber.*, **20**(2), 404-435,1990.

[3] C.C. Lee, "A self-learning rule-based controller employing approximate reasoning and neural net concepts," *International Journal of Intelligent Systems*, **5**, 71-93, 1993.

[4] R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. on Syst., Man, and Cyber.*, **23**(3), 665-685, 1993.

[5] M. Mahfouf, M.F. Abbod and D.A. Linkens, "Multi-objective genetic optimisation of GPC and SOFLC tuning factors- A new fuzzy-based ranking method," *Proc. of the IEE, Pt.D- Control Theory and Practice*, **147**(3), 344-354, 2000.

[6] D.W. Clarke, C. Mohtadi, and P.S. Tuffs, "Generalised predictive control - Parts I and II," *Automatica*, **23**, 137-160, 1987.

[7] T.J. Procyk and E.H. Mamdani, "A linguistic self-organising process controller," *Automatica*, **15**, pp 15-30, 1979.

[8] W.L. Luyben, "Process Modelling, Simulation and Control for Chemical Engineers," McGraw-Hill, 1990.

[9] M. Mahfouf. and D.A. Linkens, “Generalised Predictive Control and Bioengineering,” Taylor and Francis, ISBN 0-7484-0597-6, 1998.

[10] M. Mahfouf, D.A. Linkens, and M.F. Abbod, “Adaptive fuzzy TSK model-based predictive control using a CARIMA model structure,” *Transactions of the Institute of Chemical Engineers, Chemical Engineering Research and Design-Special Topic Issue*, Invited Paper, **78**(A), 590-596, 2000.

[11] D.W. Clarke (Eds.), “Model-Based Predictive Control,” Oxford University Press, ISBN 0-19-856292-6, 1994.

[12] L.A. Zadeh, “Outline of a new approach to complex systems and decision processes,” *IEEE Trans. on Systems, Man and Cybernetics*, **SMC 3**, 28-44, 1973.

[13] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its application to modelling and control,” *IEEE Trans. on Systems, Man, and Cybernetics*, **15**(1), 116-132, 1985.

[14] M. Sugeno and T. Yasukawa, “A fuzzy-logic based approach to qualitative modelling,” *IEEE Trans. on Fuzzy Systems*, **1**(1), 7-29, 1993.

[15] M. Sugeno and G.T. Kang, “Fuzzy modelling and control of multiplayer incinerator,” *Fuzzy Sets and Systems*, **18**, 329-346, 1986.

[16] P.R. Belanger, “On type-I systems and the Clarke-Gawthrop regulator,” *Automatica*, **19**, 91-94, 1983.

[17] G.J. Bierman, “Factorization Methods for Discrete Sequential Estimation,” Academic Press, New York, 1977.

[18] M. Sugeno and K. Murakami, “An experimental study on fuzzy parking control using a model car,” *in Industrial Application of Fuzzy Control, Sugeno, M. (Ed.), Amsterdam: North Holland*, 125-138, 1985.

7 Acknowledgements

The first and third authors acknowledge financial support for part of this work from UK EPSRC Research Grant GR/L8905.

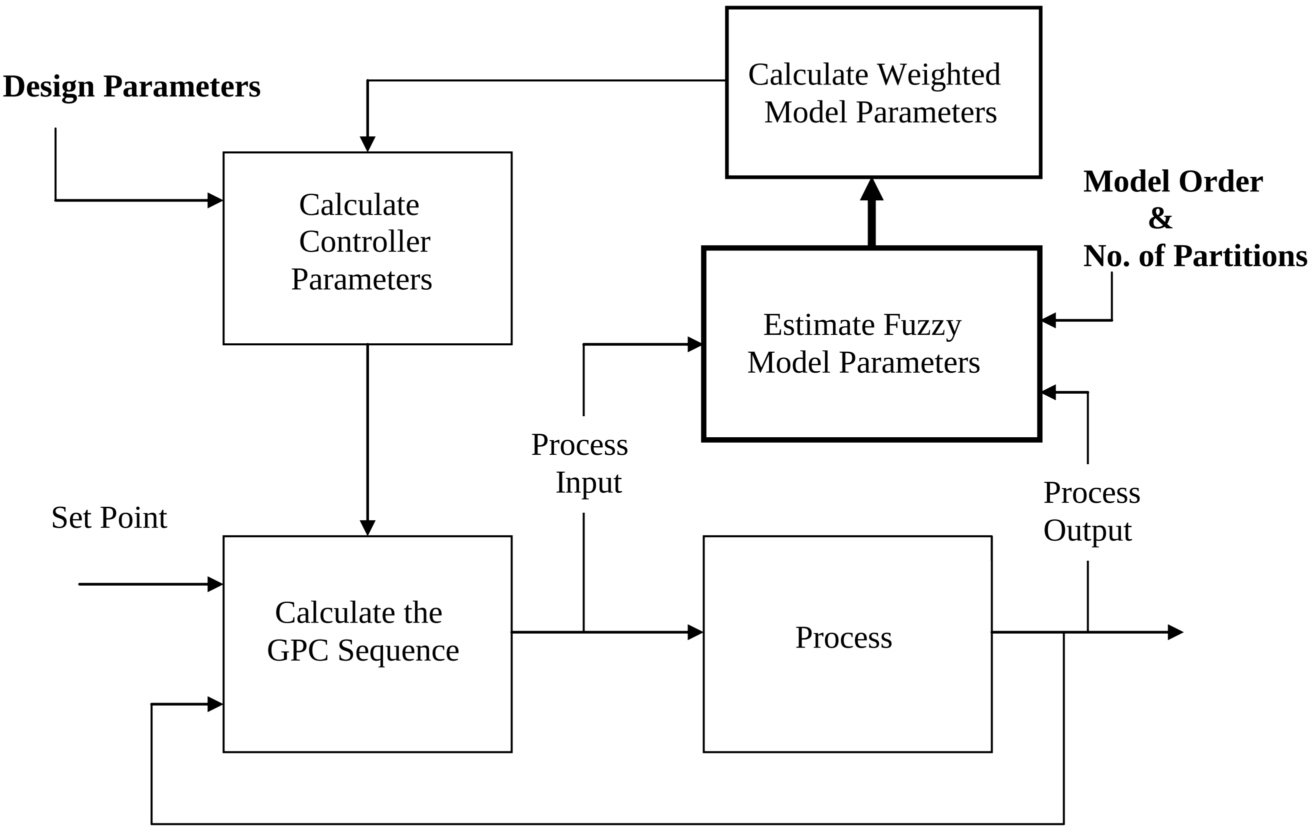


Figure 1 Adaptive Fuzzy TSK-based Generalised Predictive Control

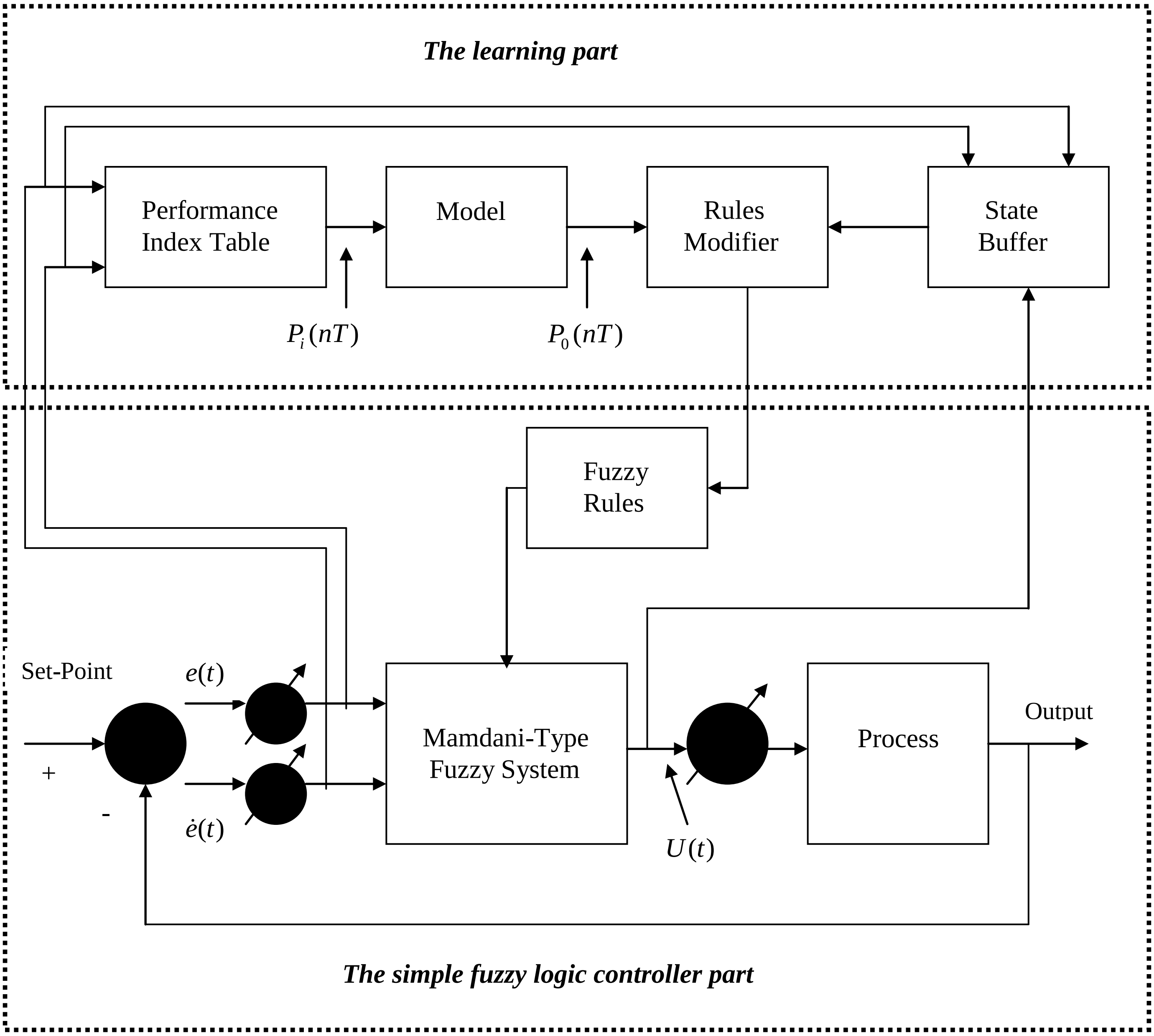


Figure 2 A schematic diagram depicting the SOFLC architecture.

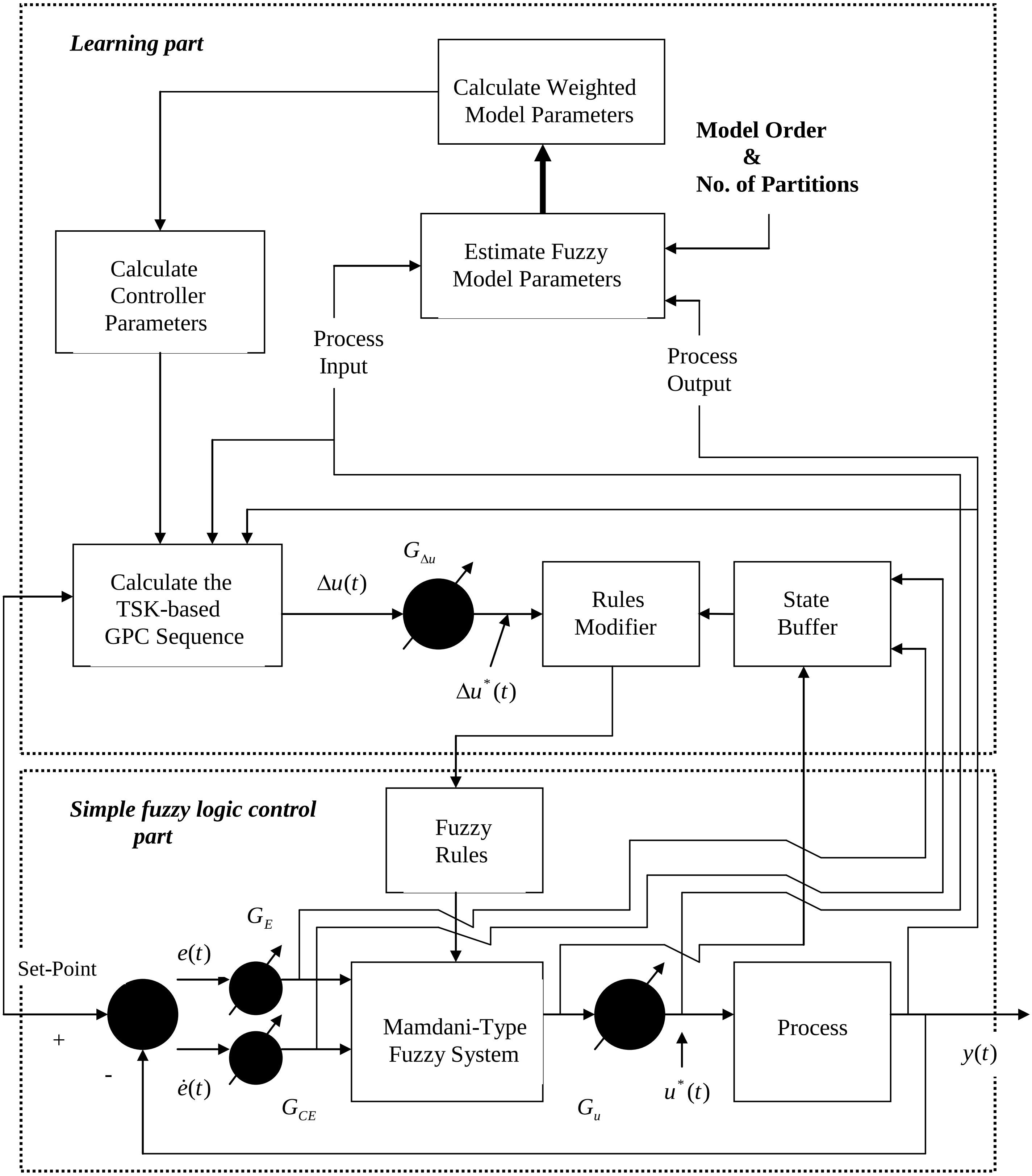


Figure 3 The GPSOFLC algorithm.

G_E : *Error Scaling Factor.*

G_{CE} : *Change in Error Scaling Factor.*

G_u : *Controller Output Scaling Factor.*

$G_{\Delta u}$: *GPC Sequence Scaling factor.*

$u^*(t)$: *Actual Process Input.*

$y(t)$: *Process Output.*

$\Delta u^*(s)$: *Scaled GPC move.*

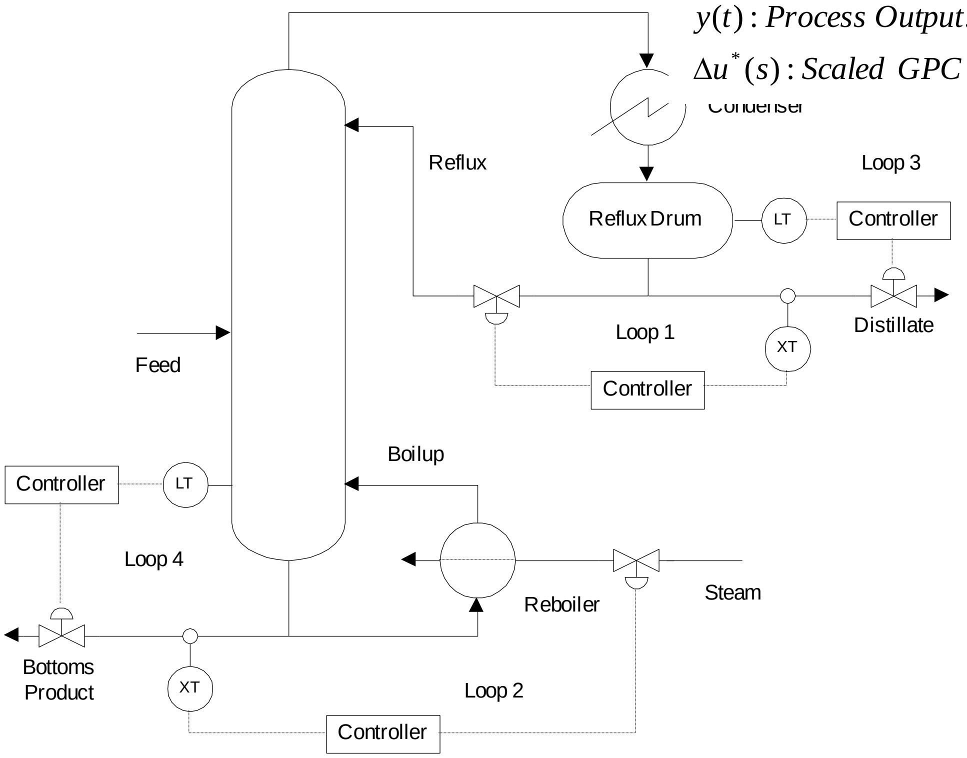
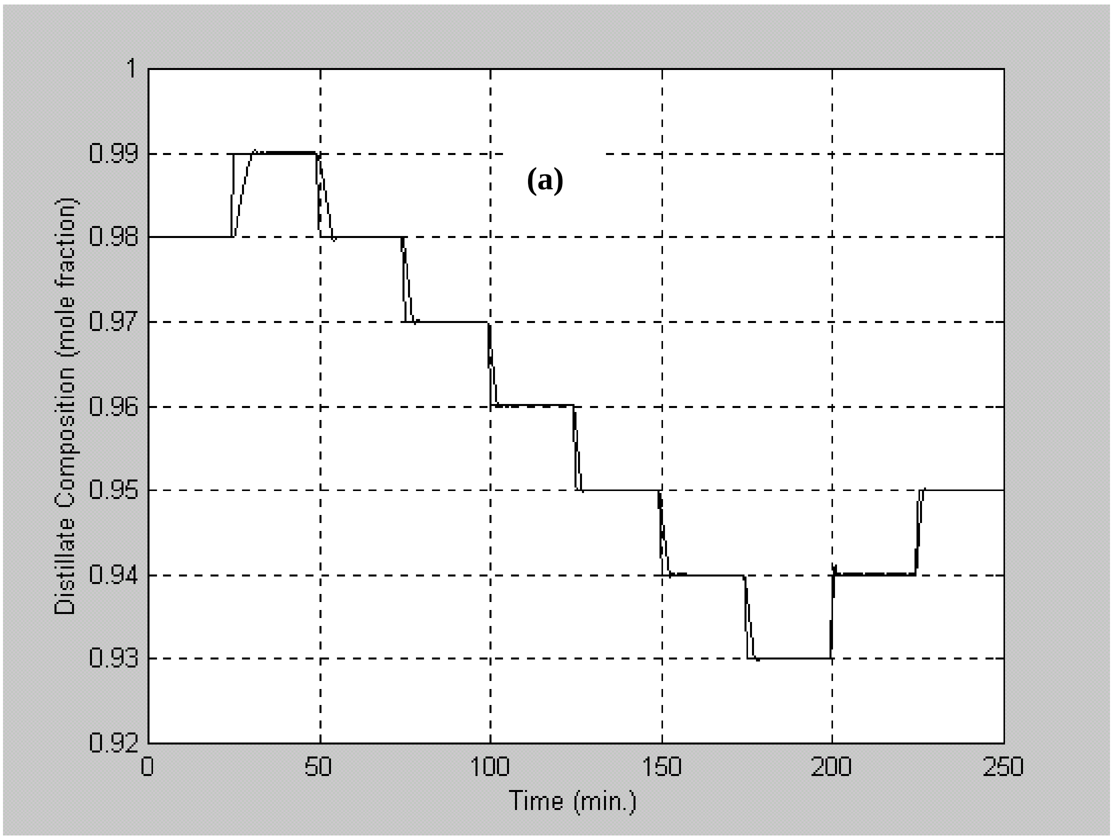
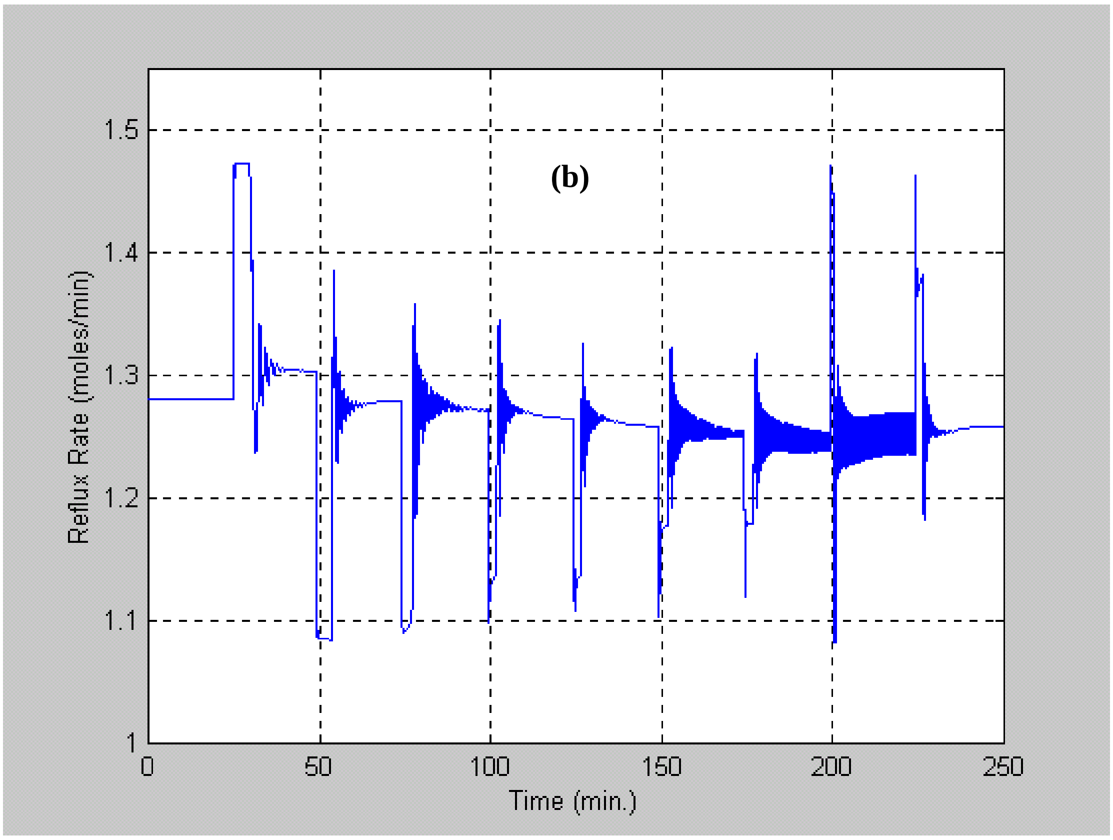


Figure 4 Control schematic for a binary distillation column.


(a)
Distillate Composition (mole fraction)
Time (min.)
1
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0.92
0
50
100
150
200
250


(b)
Reflux Rate (moles/min)
1.5
1.4
1.3
1.2
1.1
1
0
50
100
150
200
250
Time (min.)

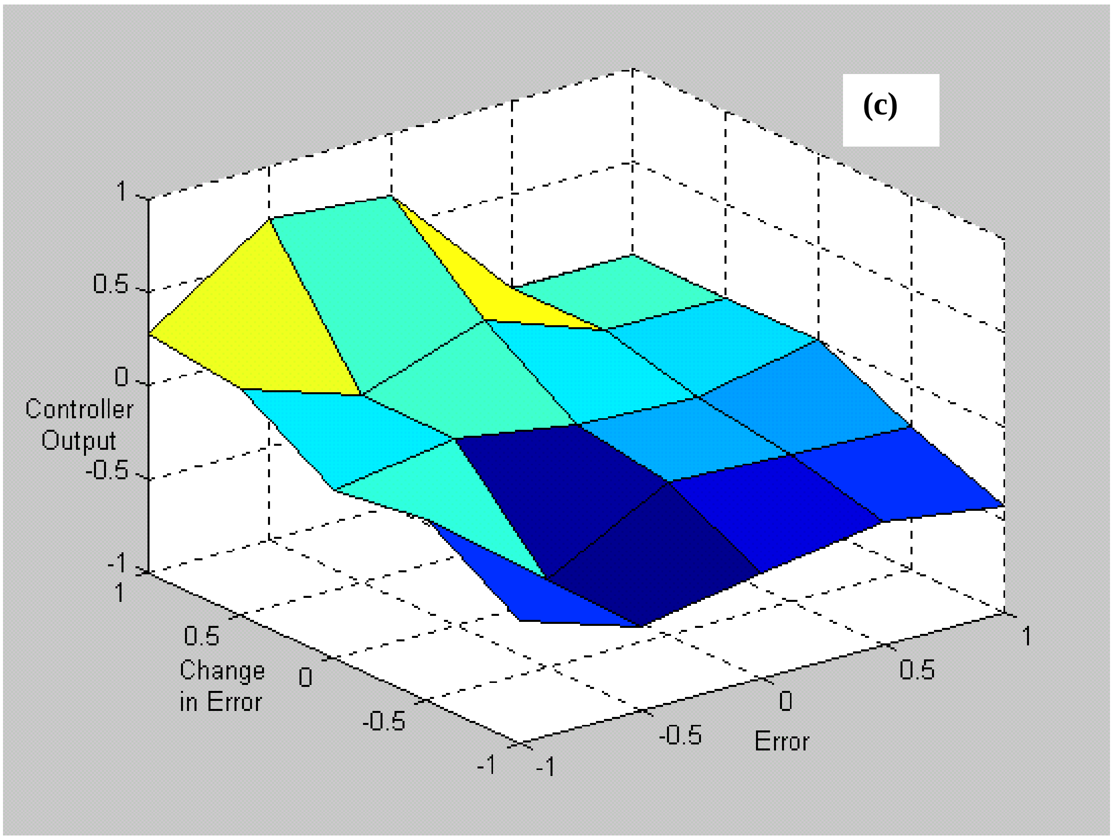
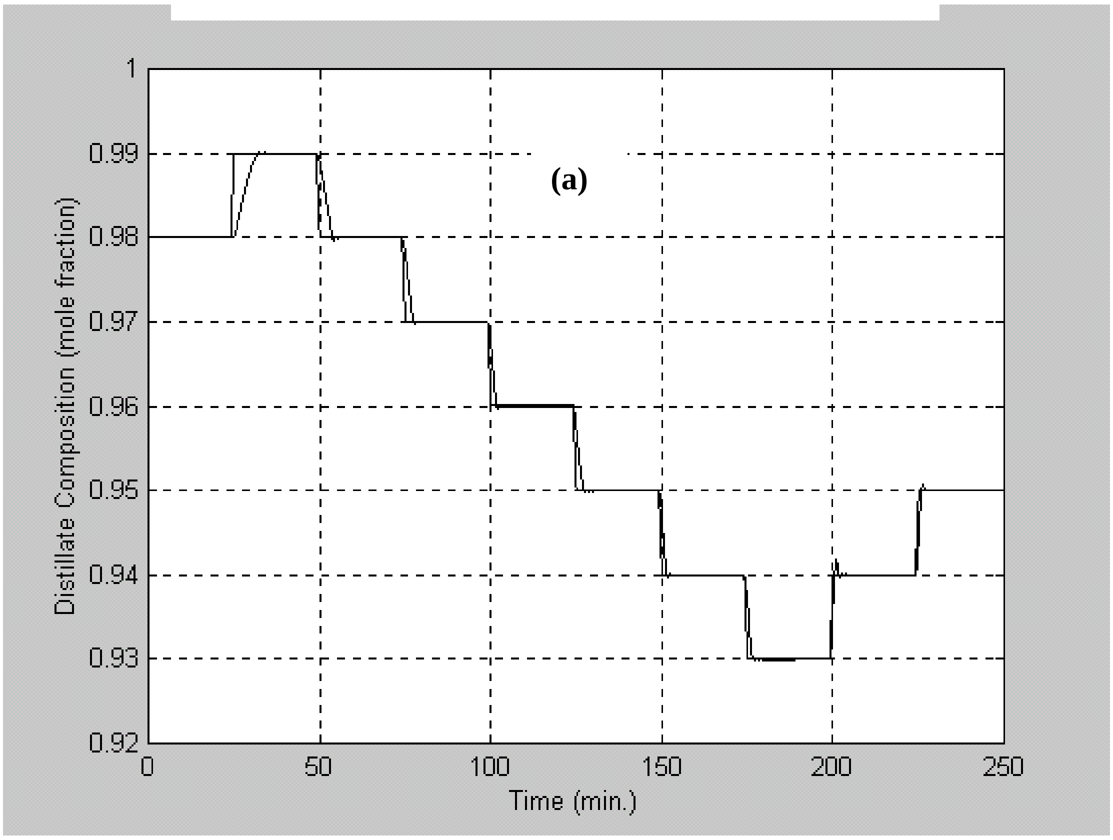
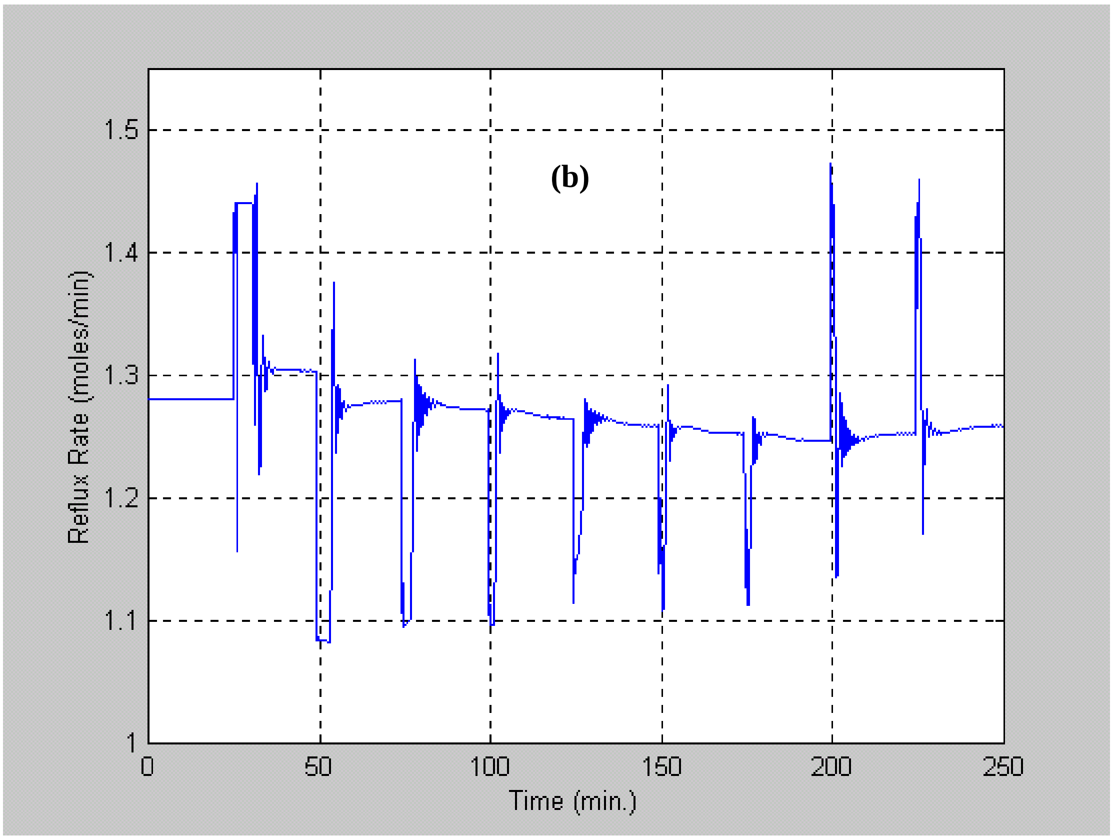

(c)
Controller Output
1
0.5
0
-0.5
-1
Change in Error
1
0.5
0
-0.5
-1
Error
-1
-0.5
0
0.5
1

Figure 5 Performance of the standard SOFLC algorithm
(a) Set-point and output.
(b) Controller output signal.
(c) 3D-surface relating to the fuzzy rule-base.




(b)
Reflux Rate (moles/min)
1.5
1.4
1.3
1.2
1.1
1
0
50
100
150
200
250
Time (min.)

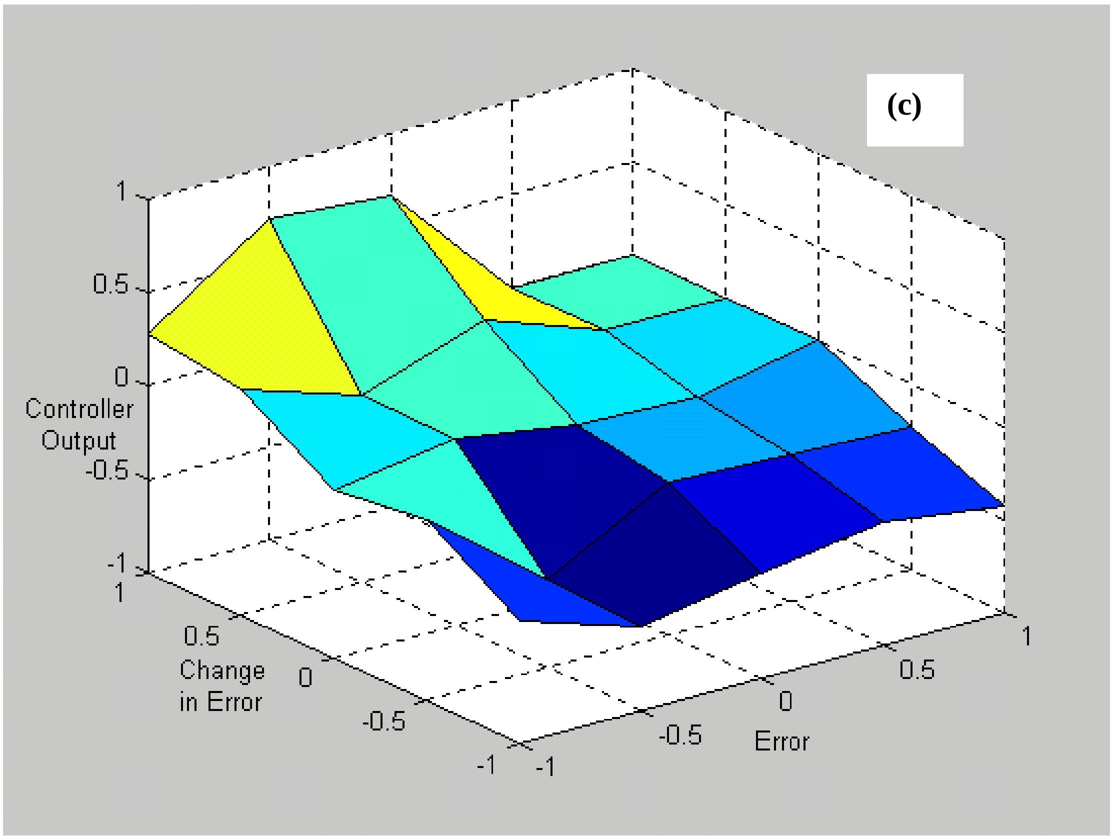
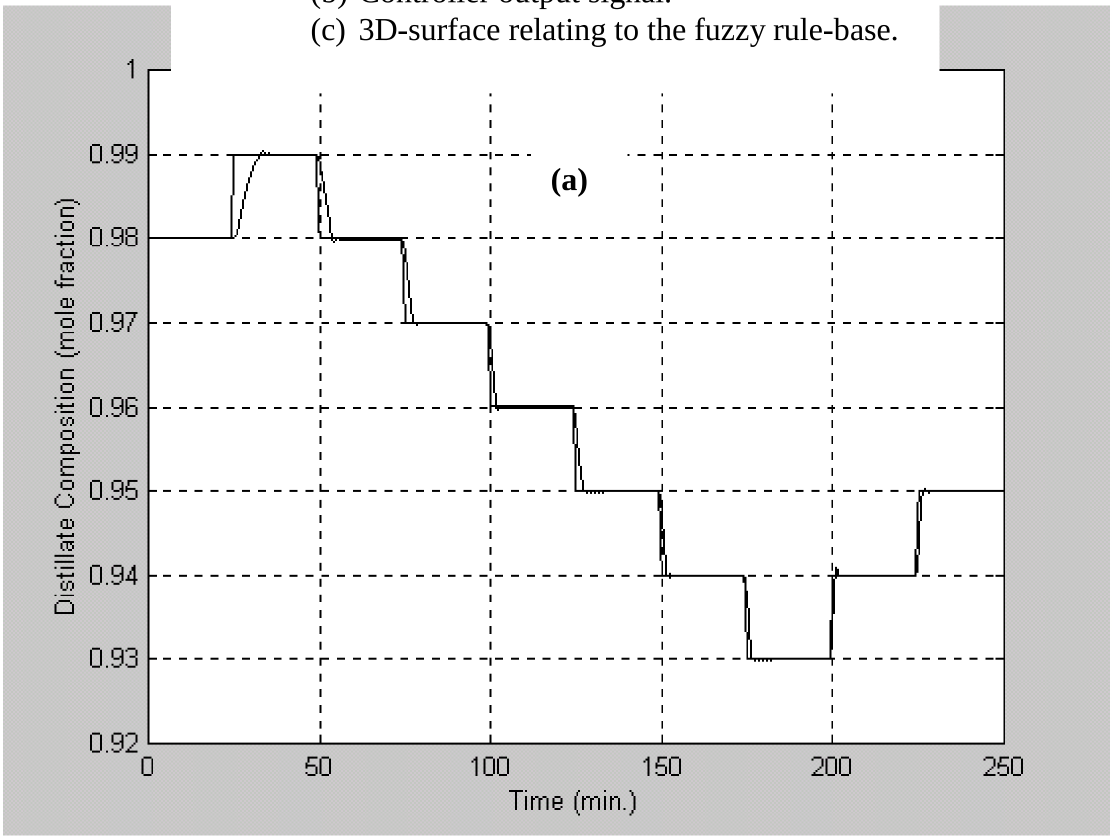
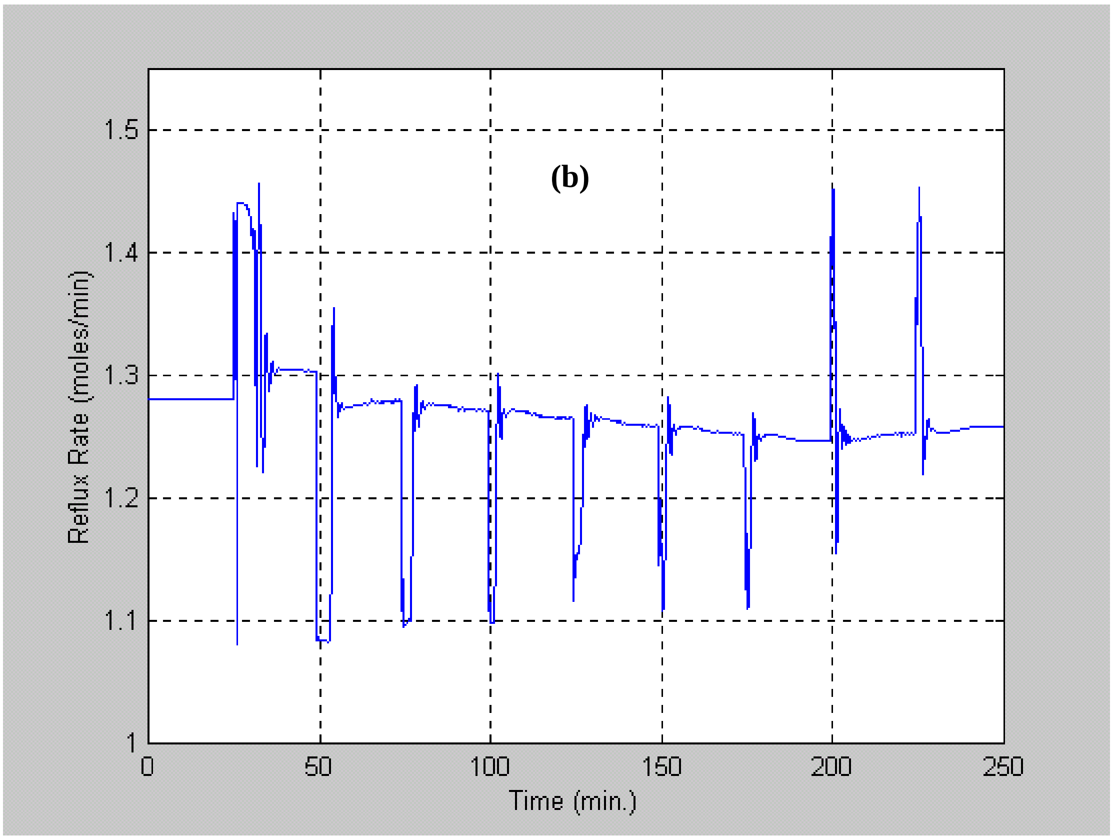

(c)
Controller Output
1
0.5
0
-0.5
-1
Change in Error
1
0.5
0
-0.5
-1
Error
-1
-0.5
0
0.5
1

Figure 6 Performance of the GPSOFLC algorithm using a linear CARIMA model;
(a) Set-point and output.
(b) Controller output signal.
(c) 3D-surface relating to the fuzzy rule-base.




(b)
Reflux Rate (moles/min)
1.5
1.4
1.3
1.2
1.1
1
0
50
100
150
200
250
Time (min.)

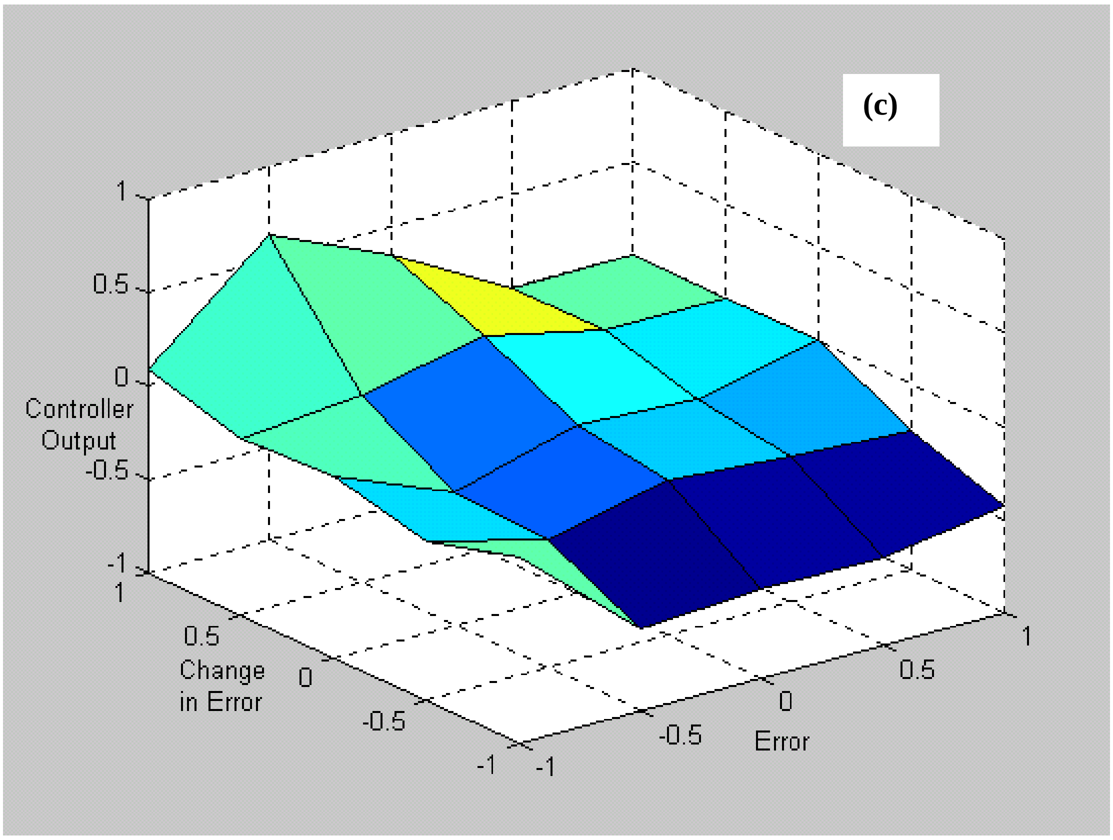
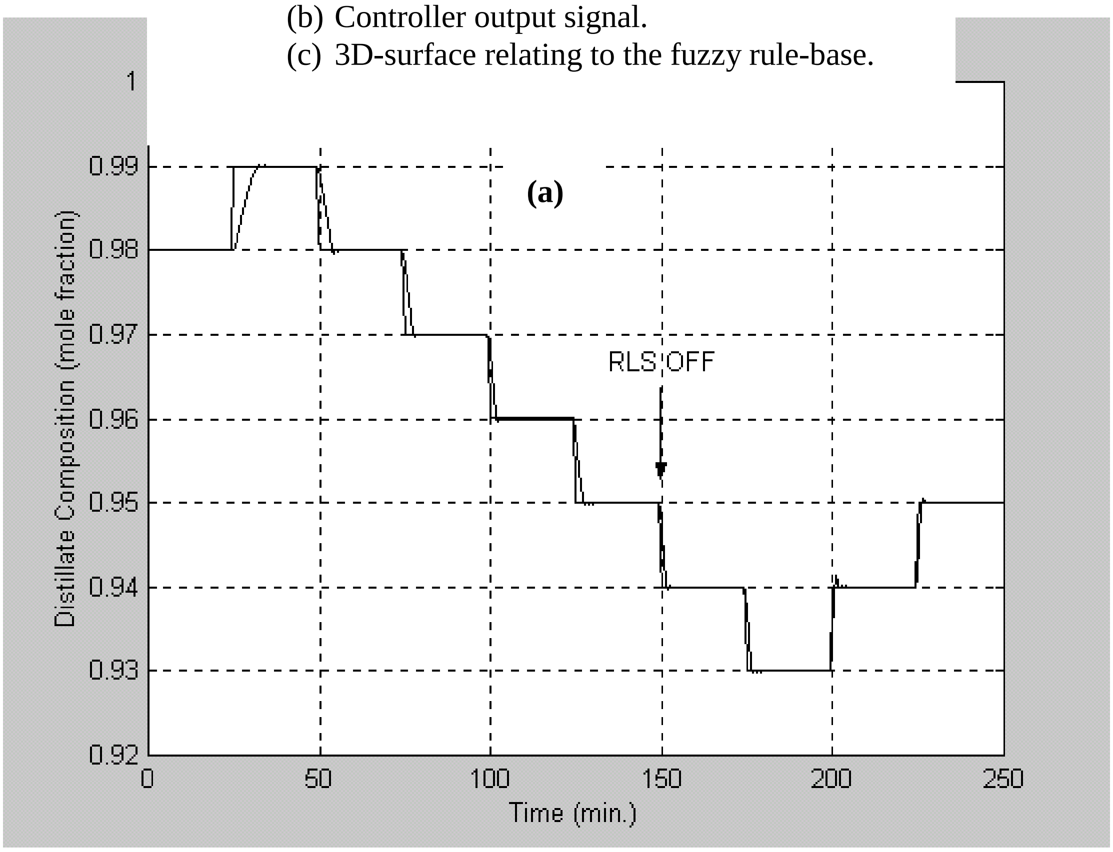
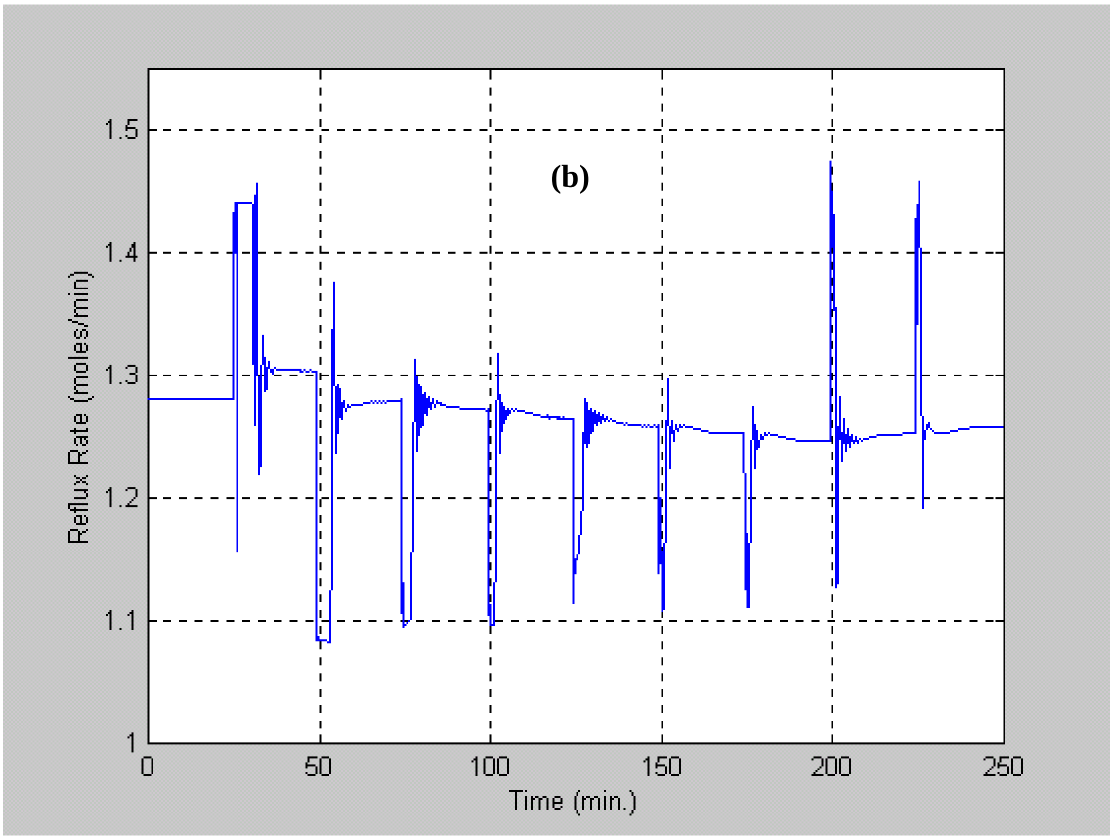

(c)
1
0.5
0
Controller Output
-0.5
-1
1
0.5
Change in Error
0
-0.5
-1
-1
-0.5
0
Error
0.5
1

Figure 7 Performance of the GPSOFLC algorithm
using a 5-partition fuzzy TSK based
CARIMA model;
(a) Set-point and output.
(b) Controller output signal.
(c) 3D-surface relating to the fuzzy rule-base.




(b)
Reflux Rate (moles/min)
1.5
1.4
1.3
1.2
1.1
1
0
50
100
150
200
250
Time (min.)

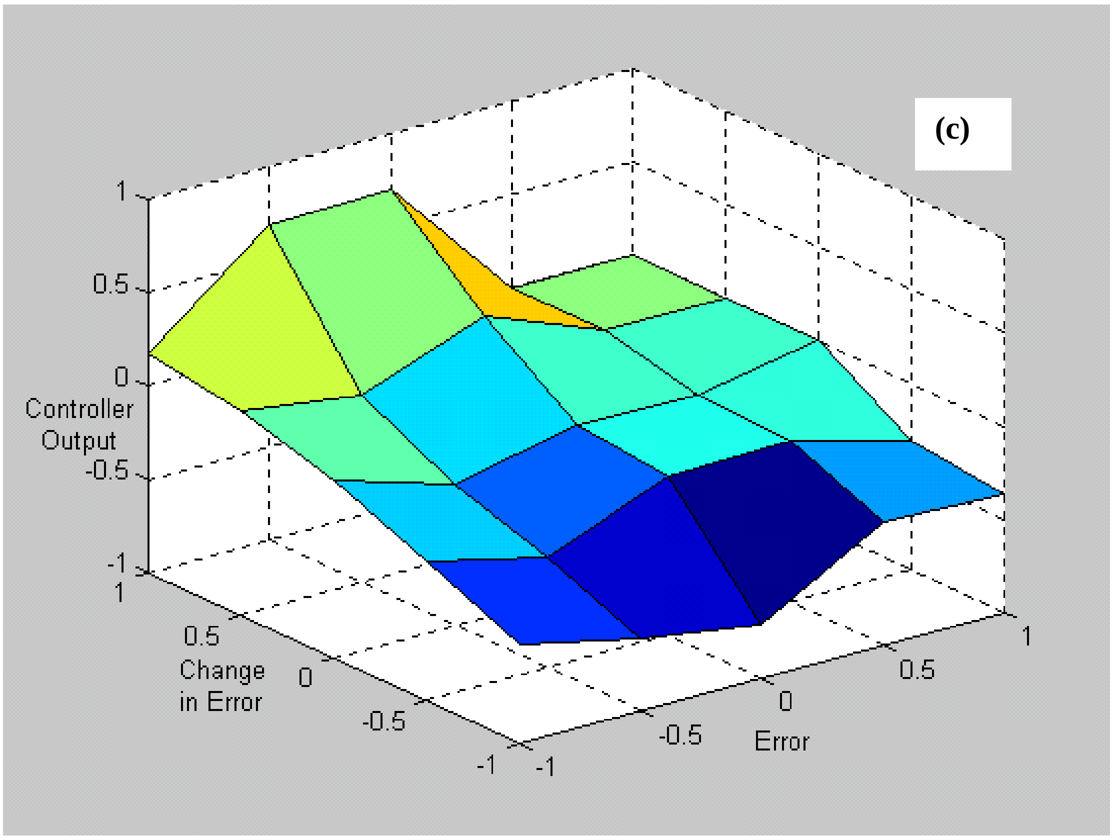
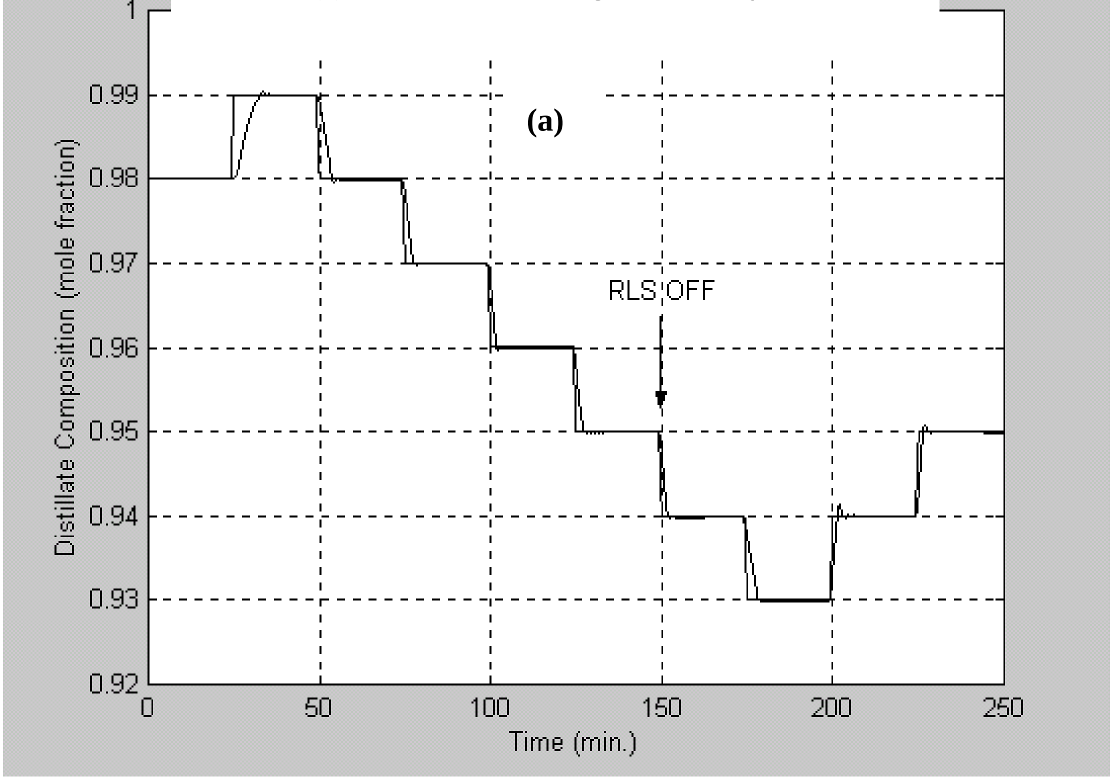
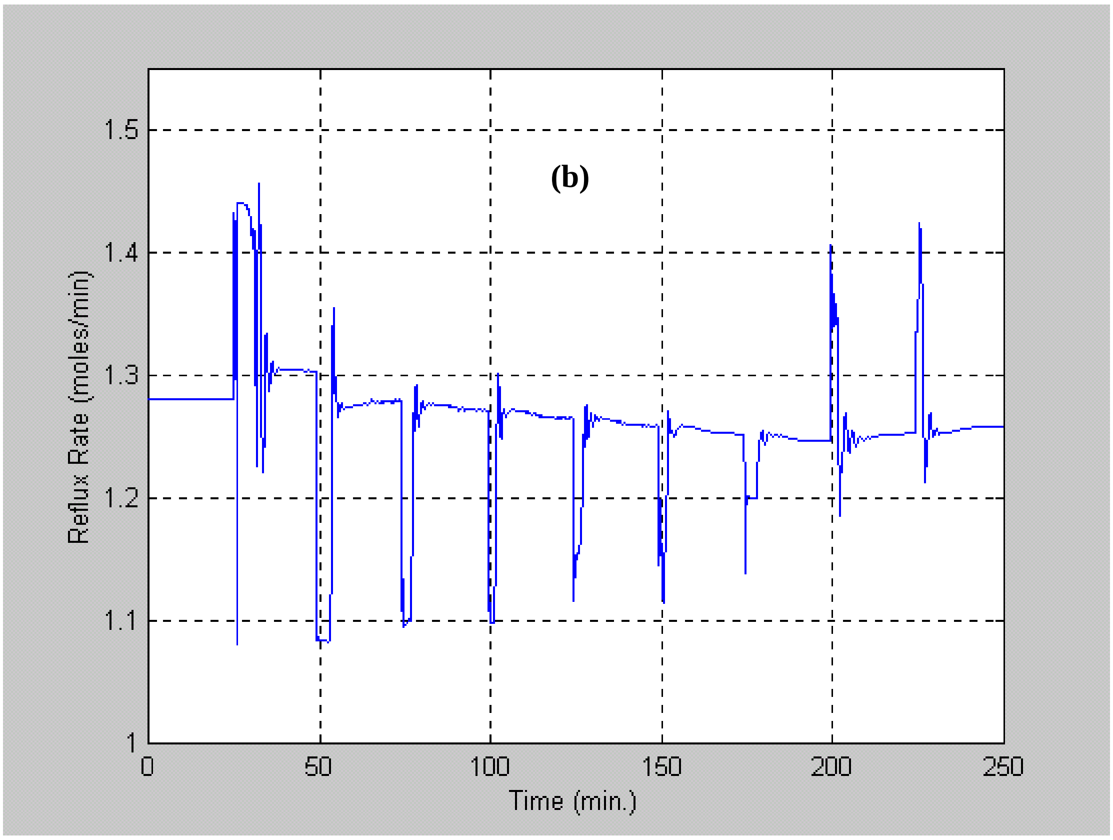

(c)
Controller Output
1
0.5
0
-0.5
-1
Change in Error
1
0.5
0
-0.5
-1
Error
-1
-0.5
0
0.5
1

Figure 8 Performance of the GPSOFLC algorithm using a linear CARIMA model when RLS is switched-off at time 150 minutes;
(a) Set-point and output.
(b) Controller output signal.
(c) 3D-surface relating to the fuzzy rule-base.




(b)
Reflux Rate (moles/min)
1.5
1.4
1.3
1.2
1.1
1
0
50
100
150
200
250
Time (min.)

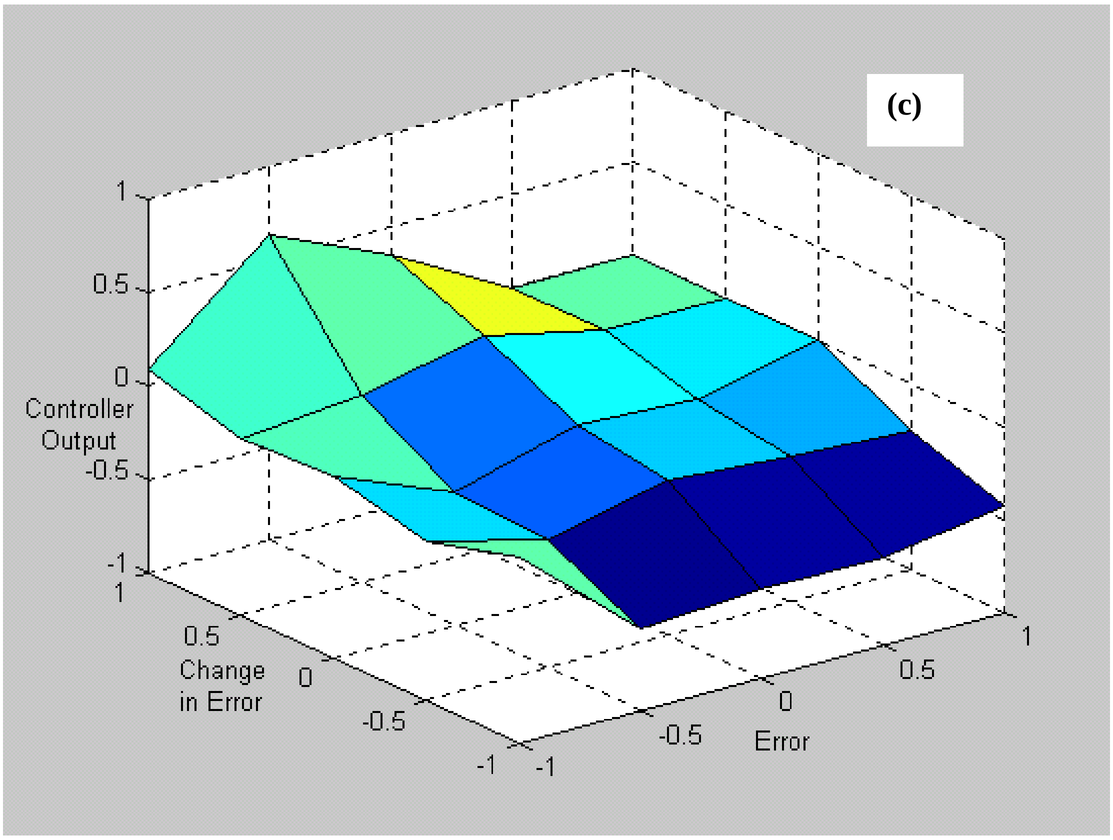

(c)
1
0.5
0
Controller
Output
-0.5
-1
1
0.5
Change
in Error
0
-0.5
-1
-1
-0.5
0
Error
0.5
1

Figure 9 Performance of the GPSOFLC algorithm using a 5-partition fuzzy TSK based CARIMA model when RLS is switched-off at time 150 minutes;

(a) Set-point and output.
(b) Controller output signal.
(c) 3D-surface relating to the fuzzy rule-base.

$\dot{e}$ \ e	NB	NS	ZO	PS	PB
NB	NB	NB	NS	NS	ZO
NS	NB	NS	NS	ZO	PS
ZO	NS	NS	ZO	PS	PS
PS	NS	ZO	PS	PS	PB
PB	ZO	PS	PS	PB	PB

Table 1a The modified performance index which uses 5 membership functions in a "rules" format;
N = Negative, P= Positive,
ZO = Zero, B = Big, S = Small,
M = Medium.

$\dot{e}$ \ e	NB	NS	ZO	PS	PB
NB	-1	-1	-0.5	-0.5	0
NS	-1	-0.5	-0.5	0	0.5
ZO	-0.5	-0.5	0	0.5	0.5
PS	-0.5	0	0.5	0.5	1
PB	0	0.5	0.5	1	1

Table 1b The modified performance index which uses 5 membership functions in a "look-up-table" format; N = Negative, P= Positive, ZO = Zero, B = Big, S = Small, M = Medium.

e \ $\dot{e}$	**NB**	**NS**	**ZO**	**PS**	**PB**
NB	-0.35	-0.55	-0.44	-0.34	-0.43
NS	-0.03	-0.53	-0.18	-0.21	-0.24
ZO	-0.10		-0.10	-0.13	
PS	0.21		0.23		
PB	0.28	0.72	0.68		

Controller Output values normalised between –1 and +1

No rules elicited

Table 2 The final free structure of the fuzzy rule-base relating to the run of Figure 6.
N = Negative, P= Positive, ZO = Zero, B = Big, S = Small.

e \ $\dot{e}$	**NB**	**NS**	**ZO**	**PS**	**PB**
NB	-0.47	-0.61	-0.71	-0.34	-0.37
NS	-0.25	-0.41	-0.14	-0.13	-0.30
ZO	-0.05	-0.24	-0.11	-0.11	
PS	0.10		0.26		
PB	0.17	0.68	0.70		

Table 3 The final free structure of the fuzzy rule-base relating to the run of Figure 8 **(RLS switched-off at time 150 minutes)**.
N = Negative, P= Positive, ZO = Zero, B = Big, S = Small.