Proceedings of the 7th Mediterranean Conference on Control and Automation (MED99) Haifa, Israel - June 28-30, 1999

MULTIPLE MODEL CONTROL OF A PILOT DISTILLATION COLUMN

J. A. Rodriguez [•][**], G. C. Goodwin [**], J. A. Romagnoli [•],

** CIDAC, Electrical and Computer Engineering, University of Newcastle, Australia.
• ICI Laboratory of Process Systems Engineering, Chemical Engineering, Sydney University, Australia.
E-mail: julio@enviro.chem.eng.usyd.edu.au

Abstract: This paper discusses a complete controller design strategy required for implementing Multiple Model Control (MMC), applied to nonlinear systems. It is shown that the multiple model design can be recast into a supervisory arrangement, where a global supervisor is utilised to select the appropriate controller from a fixed family set. Unlike current techniques where Fuzzy Validity Functions or Bayesian Estimators are utilised in the selection mechanism, the approach of a Multiple Model Observer (MMO) is employed for the selection architecture within the supervisor. This notion, is a natural extension of the MMC design. Switching between the individual controllers is accomplished bumplessly by using a Multiple Model Bumpless Transfer Mechanism, thus producing a smooth and continuous control signal as the plant passes through different operating regions. The above notion is applied to a Pilot Binary Distillation Column, which is nonlinear in nature. The principal nonlinearity of the process is strongly related to its operating point. This paper illustrates that, as the distillation column moves from one operating point to another, the MMC self-regulates according to the operative trajectory, consequently ensuring that global stability and performance is maintained at an optimal point.

1 INTRODUCTION

When faced with nonlinear systems, the control strategy required for such a plant has not yet been fully understood, since traditional control synthesis is essentially limited to linear processes. Techniques which have been adopted for nonlinear control include; Adaptive, Sliding mode control etc. While there has been a lot of work in regulatory control over the years, much less attention has been devoted to supervisory control issues. Dealing with nonlinear systems is obviously an inherently difficult problem. A consequence of this principle is that models and analysis of nonlinear systems will be less precise than for the simpler linear case. A further consequence is that one should, perhaps, look for other model representations and tools that utilise less precise system knowledge than the traditional approaches. This is indeed the trend in the area of intelligent control where Fuzzy Logic, Global Decomposition, Neural Networks, Expert systems and Probabilistic Reasoning are being explored. The current paper focuses on one approach to the decomposition of modelling and control problems that has recently attracted significant attention, namely operating regime decomposition.

The core of the operating regime approach is to make use of a partitioning of the operating range of the system in order to solve modelling and control problems. The operating regime approach thus leads to multiple-model or multiple controller synthesis, where different local models/controllers are applied under different operating conditions, see figure 1. One version of the above strategy is to represent the global system as a family of smaller local regions, where the supervisory controller alters the regulatory controller according to the current local region in which the process is operating. The mathematical representation of such a nonlinear system is then given as a Linear Parameter varying (LPV) system.

We emphasis, that this strategy only holds true for nonlinear systems which can be accounted for as an LPV system, however, it is also applicable for systems which move from one linear region to another, such as some continuous and batch operations.

1.1 System Description and Modelling.

An inherent problem with all function approximation approaches that are based on a partitioning of the function's domain is the curse of dimensionality. With an increasing number of variables on which the function depends, the number of partitions required (in a uniform partitioning) will increase exponentially. Consequently, a uniform partitioning is undesirable and unrealistic for anything other than low complexity problems. Fortunately, a uniform partitioning is usually not necessary, [6]. Development of a simple non-uniform decomposition of the operating range is often difficult. Prior knowledge about the system, or careful examination of large amounts of empirical data is the key to achieving this goal. In the sequel, we show that the curse of dimensionality can often be 'warded off', at least to some extent. Consider the development of a nonlinear model of the form

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

It is easy to see that such a model can always be written as

$$\dot{x} = a(x,u) + A(x,u).x + B(x,u).u , \tag{2}$$

where (a) is a steady state bias, A and B are linearised state space matrices, *x* is the state vector and *u* is the plant input. This form emphasises the close relationship to linear models of the form

$$\dot{x} = a_i + A_i x + B_i.u , \tag{3}$$

and quasi-linear models of the form

$$\dot{x} = a.(\lambda) + A.(\lambda).x + B(\lambda).u . \tag{4}$$

By comparing (2) and (4) we observe that the variable λ clearly should depend on *x* and *u*. Suppose we decompose the operating range into a number of operating regimes where the system is described by linear models of the form (3). From (2) it is evident that the linearised model's parameters will then depend on the operating point. However, by a possible change of variables it is clear that under some conditions, one can find a characterising vector λ satisfying the following;

$$\begin{aligned} &a(x,u) = a(\lambda), \\ &A(x,u) = A(\lambda), B(x,u) = B(\lambda), \end{aligned} \tag{5}$$

which contains fewer elements than the total number of elements in *x* and *u* [8]. Following the terminology of [15], LPV systems are linear time-varying plants whose state space matrices are fixed functions of some vector of varying parameters $\lambda(t)$. Hence LPV systems are described by state space equations of the form:

$$\begin{aligned} &\dot{x} = A(\lambda(t))x + B(\lambda(t))u \\ &y = C(\lambda(t))x + D(\lambda(t))u \end{aligned} , \tag{6}$$

where A, B, C and D are the state space matrices, (x) is the state vector and (u) is the plant input, [4].

1.2 Over-all Supervisory Structure.

The procedure which implements the proposed multiple model controller is based on observations of the past inputs and outputs of the system. A countable set of predictors, $O_i(\lambda), i \in n$, is used to predict the system output response. Using performance measures derived from the resultant prediction errors, a decision rule is designed to select a λ_i, $i \in n$. We allow the structure of the decision rules to converge to some *i*, such that the i^{th} prediction error has desirable properties, e.g. best fit to the real data. In a very general setting we give a positive result that there exists a stationary decision rule with minimal modelling error which converges within a finite time to a 'good' predictor. Based on the decision rule's selection at time *t*, a controller for the system is chosen from a family of pre-designed linear controllers, $C_i(\lambda_i)$, [10] and [11].

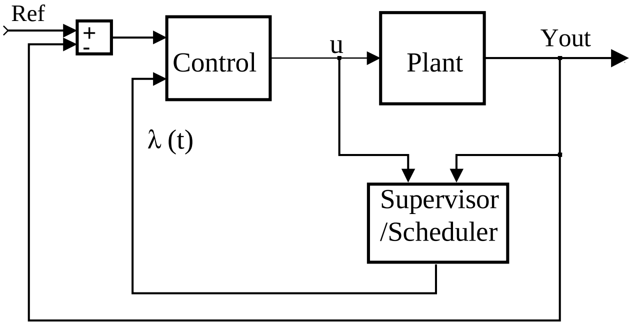

Figure.1 Multiple Model Closed Loop Structure

2. SUPERVISORY CONTROL

As described above, if a process can be expressed as an LPV system then the concept of Multiple Model Control can be utilised as a global controller for the nonlinear system. In such a scenario a supervisory controller is required to select the appropriate controller depending on the state of operation of the system. Based on this multiple representation of the process at different operative levels a corresponding controller is allocated. Optimal approaches are used to design the corresponding controller of each model. The first step is to locate which model of the process is appropriate and then to use the controller to drive the system back to the desired trajectory. The supervisor's unique feature, sharply distinguishing it from other logic which might be used for the same purpose, is that the controller selection is made not by going through an exhaustive search but rather by

1. continuously comparing in real-time suitably defined norm-squared output estimation errors or 'performance signals' determined by candidate nominal process model transfer functions and
2. by placing in the feedback-loop from time to time that candidate controller whose corresponding performance signal is the smallest. The paradigm is, of course, a manifestation of the idea of certainty equivalence from parameter adaptive control.

The process to be controlled is assumed to be modeled by a linear system whose transfer function is in the union of a number of given subclasses $G_i(\lambda)$. It is assumed that for each subclass $G_i(\lambda)$, there is a single, given loop-controller transfer function $C_i(\lambda)$. The loop-controller transfer functions are implemented by means of a linear system called a "multicontroller". Therefore, the system to be supervised is the feedback interconnection of the multicontroller and the process to be controlled $[G_i(\lambda), C_i(\lambda)]$.

2.1 Estimator-based Selection.

The proposed estimator can informally be explained in terms of the Multiple-estimator architecture given in figure 2. Here, the output estimation errors are of the form

$$e_i = y - y_i \tag{9}$$

where each y_i is a signal generated by the supervisor in such a way that it would be an asymptotically correct estimate of *y*, if *u* were the process input and there were no noise or disturbance signals. The i'th output estimation error π_i is a norm-squared value of e_i or some other performance signal which is used by the supervisor to assess the potential performance of a local controller, [9].

Hysteresis switching is accomplished by introducing switching logic whose function is to determine λ_i on the basis of the current values of π_i, [16].

Naturally this architecture can only be implemented as it stands if the number of output estimation errors is finite, i.e. if (λ) is a finite piece wise trajectory. It turns out, however, that such a supervisor can often be implemented using a simpler architecture, one, which permits (λ) to contain a continuum of points. The supervisor to which we are referring uses state-sharing not only to generate the all of the e_i signals but also all of the π_i's as well. Internally the supervisor consists of three subsystems: a state-shared estimator; a performance weight generator; and switching logic, [10], see figure 2.

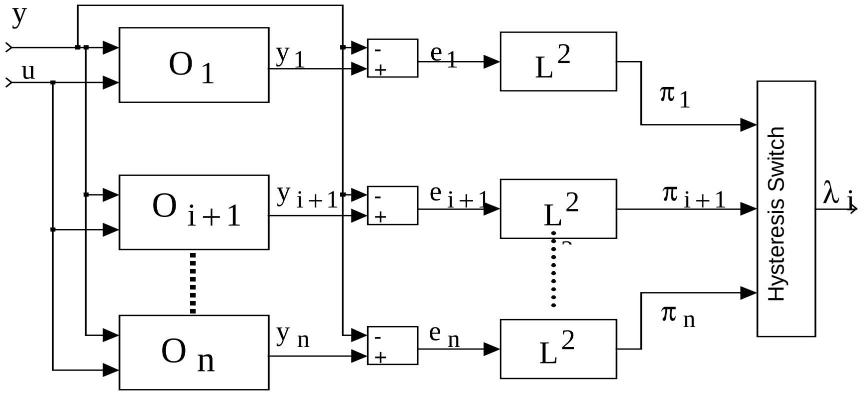

Figure.2 Multiple-Model-estimator Selector

3. GLOBAL CONTROLLER DESIGN.

Controller design is performed by assuming that the true plant lies in the family given by $G(\lambda) = M_i(\lambda) + \Delta_i(t)$, where $M_i(\lambda)$ is the LPV model given by (8) and $\Delta_i(t)$ is the corresponding additive uncertainty. The approach used for the global controller is a self-scheduled controller design, where the controller adjusts to variations in plant dynamics, so as to maintain stability and good performance during transitions. Since the control law has to include a logical choice, the controller is itself, an LPV system, of the form

$$\begin{aligned} \dot{x}_{c,i} &= A_{c,i}(\lambda).x_i + B_{c,i}(\lambda).e \\ u_i &= C_{c,i}(\lambda).x_i + D_{c,i}(\lambda).e \end{aligned} \qquad (10)$$

where $[A_{c,i}, B_{c,i}, C_{c,i}, D_{c,i}]$ are state space matrices, $x_{c,i}$ is the i'th controller state vector, *e* is the loop error and u_i is the local controller signal.

The feedback loop is next cast into this $M-\Delta$ form, in order to use the standard H_∞ design framework to obtain robust stability and performance for all possible values of a piece-wise vector λ_n, [1] and [2]. These requirements can be translated into a condition on performance of the form

$$\|W_1 S(\lambda)\|_\infty \le \beta, \qquad (11)$$

and a condition that the closed loop remains stable for all possible uncertainties, $\Delta = \Delta_2 W_2$, where

$$\|\Delta_2\|_\infty \le \beta. \qquad (12)$$

$S(\lambda) = (I + G(\lambda).C(\lambda))^{-1}$ is the sensitivity operator mapping *r* to *e*, W1 and W2 are suitable frequency dependent weighting matrices. Therefore, to ensure that the controller satisfies (11) and (12) it must also satisfy

$$\left\| \begin{pmatrix} -W_2.C(\lambda).S(\lambda) & W_2.C(\lambda).S(\lambda) \\ -W_1.S(\lambda) & W_1.S(\lambda) \end{pmatrix} \right\|_\infty \le \gamma, \qquad (13)$$

where $0 \le \gamma \le 1/\beta$.

However, the uncertainty region is parameter varying, and therefore Δ_2 is a parameter varying full complex matrix. To take this into account, the robust performance condition of (13) is modified to

$$\inf_{D_{sc}} \left\| \begin{pmatrix} D_{sc} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} -W_2.C(\lambda).S(\lambda) & W_2.C(\lambda).S(\lambda) \\ -W_1.S(\lambda) & W_1.S(\lambda) \end{pmatrix} \begin{pmatrix} D_{sc}^{-1} & 0 \\ 0 & I \end{pmatrix} \right\|_\infty \le \gamma \qquad (14)$$

where D_{sc} is a scaled identity matrix. For notational convenience $\hat{D}_{sc}$ is defined to be

$$\hat{D}_{sc} = \begin{pmatrix} D_{sc} & 0 \\ 0 & I \end{pmatrix}. \qquad (15)$$

In order to design a controller that satisfies these requirement a state space representation of the system must be obtained in the form of figure 3.

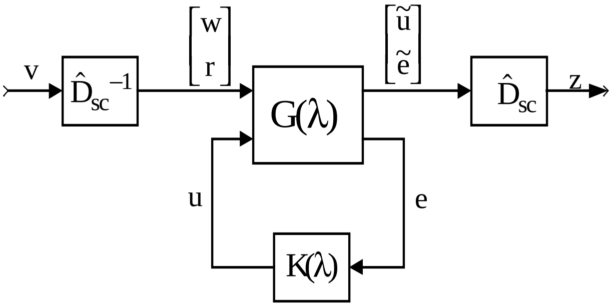
v
$\hat{D}_{sc}^{-1}$
$\begin{bmatrix} w \\ r \end{bmatrix}$
G(λ)
$\begin{bmatrix} \tilde{u} \\ \tilde{e} \end{bmatrix}$
$\hat{D}_{sc}$
z
u
e
K(λ)

Figure.3 Representation for controller design

In figure 3,

$$v = \hat{D}_{sc}\begin{bmatrix} w \\ r \end{bmatrix} \qquad z = \hat{D}_{sc}\begin{bmatrix} \tilde{u} \\ \tilde{e} \end{bmatrix}. \tag{16}$$

If the filters W_1 and W_2 are given in state-space form as $\left[\overleftrightarrow{A}_1, \overleftrightarrow{B}_1, \overleftrightarrow{C}_1, \overleftrightarrow{D}_1\right]$ and $\left[\overleftrightarrow{A}_2, \overleftrightarrow{B}_2, \overleftrightarrow{C}_2, \overleftrightarrow{D}_2\right]$ respectively, then the controller satisfying (14) may be transformed into that of figure 3. Note that the block $G(\lambda)$ represents the system:

$$\dot{x} = \left(\sum_i \lambda_i A_i\right)x + \hat{B}_1\begin{bmatrix} w \\ r \end{bmatrix} + \hat{B}_2 u \tag{17}$$

$$\begin{bmatrix} \tilde{u} \\ \tilde{e} \end{bmatrix} = \hat{C}_1 x + \hat{D}_{11}\begin{bmatrix} w \\ r \end{bmatrix} + \hat{D}_{12} u$$

$$e = \hat{C}_2 x + \hat{D}_{21}\begin{bmatrix} w \\ r \end{bmatrix} + \hat{D}_{22} u$$

where

$$\hat{A}_i = \begin{bmatrix} A_i & 0 & 0 \\ \overleftrightarrow{B}_1 C & \overleftrightarrow{A}_1 & 0 \\ 0 & 0 & \overleftrightarrow{A}_2 \end{bmatrix}$$

$$\hat{B}_1 = \begin{bmatrix} 0 & 0 \\ \overleftrightarrow{B}_1 & -\overleftrightarrow{B}_1 \\ 0 & 0 \end{bmatrix} \qquad \hat{B}_1 = \begin{bmatrix} B \\ \overleftrightarrow{B}_1 D \\ \overleftrightarrow{B}_2 \end{bmatrix}$$

$$\hat{C}_1 = \begin{bmatrix} -\overleftrightarrow{D}_1 C & -\overleftrightarrow{C}_1 & 0 \\ 0 & 0 & \overleftrightarrow{C}_2 \end{bmatrix} \qquad \overleftrightarrow{C}_2 = \begin{bmatrix} -C & 0 & 0 \end{bmatrix}$$

$$\hat{D}_{11} = \begin{bmatrix} -\overleftrightarrow{D}_1 & \overleftrightarrow{D}_1 \\ 0 & 0 \end{bmatrix} \qquad \hat{D}_{12} = \begin{bmatrix} -\overleftrightarrow{D}_1 D \\ \overleftrightarrow{D}_2 \end{bmatrix}$$

$$\hat{D}_{21} = \begin{bmatrix} -I & I \end{bmatrix} \qquad \hat{D}_{22} = -D$$

A The D-K iteration scheme, [5], has been employed to design a controller that satisfies the condition of equation (14), using the state space representation of equation (17). A comprehensive description of the algorithm is given in [4].

4. BUMPLESS TRANSFER SWITCHING

In the following, we assume that both the global controller and supervisory structure are predefined as above, where for each local controller a local selector is assigned. Also, within the supervisor the effect of redundant switching is removed and compensated by hysteresis. Global asymptotic stability of the combined supervisor and regulatory controller is guaranteed by the design synthesis of both systems. This assumption is possible since the dwell time T_c of the switching algorithm is made smaller than the selection time required by the supervisor.

For a smooth swap of control action to occur, the current controller output has to map to the latent controller prior to the switch. This is accomplished by forcing the states of the latent controller to track the output of the currently active controller. This concept of state matching is obtained by considering the memory element of the controller. From previous work [7], developed with respect to anti windup strategies, it has been shown that states of a bi-proper transfer function can be isolated. This means that the dynamics of the bi-proper transfer function i.e. memory can be encapsulated into one single node. This memory sub-section will then be a strictly proper function, therefore, given the controller function C(s) is bi-proper, we can rearrange C(s) to produce

$$C(s)^{-1} = c_\infty^{-1} + C(s)^{-1} - c_\infty^{-1}, \tag{18}$$

where c_∞^{-1} is the high frequency gain of $C(s)^{-1}$. Moreover, what is obviously required is that C(s) be implemented and not $C(s)^{-1}$, this can be corrected by re-inverting C(s) by feedback. With the above definition of $C(s)^{-1}$, the high frequency gain and the memory term from the controller inverse can be separated as

$$C(s)^{-1} = \gamma_0 + \Gamma(s), \tag{19}$$

where

γ_0 is the high frequency gain and

$\Gamma(s)$ is a strictly proper function, [13].

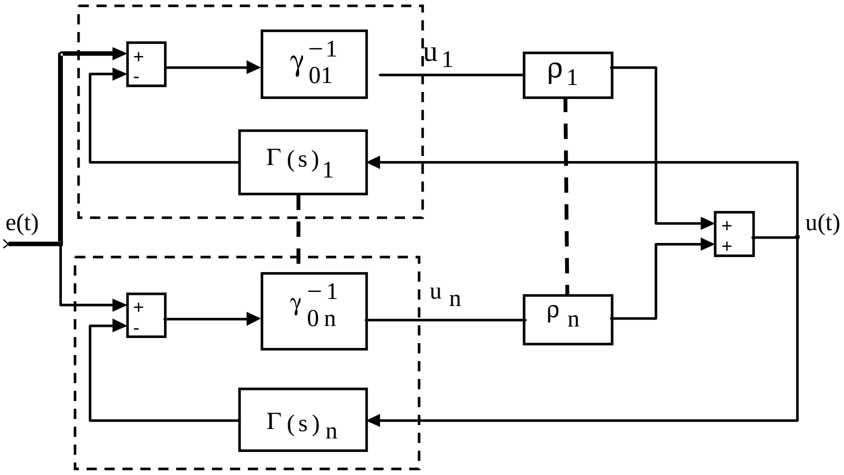

Figure.4 Controller Structure for (n) Controllers

Returning to our original requirement, namely that of a bumpless transfer, the states of the controllers have to be mapped. This is achieved by arranging all of the *n* controllers in the same structure to that of (19) and placing them as in figure 4. Note: the signal that is driving the memory element should be the actual final controller signal sent to the plant, u(t).

From figure 4 it can be seen that the, output u_i, of each individual controller is not equivalent, since each local controller is unique. However, we recall from section 3, that due to the controller and modelling design there is a small overlapping regime and during this period the switching occurs. During this regime the states of the relevant controller pair will be similar, since both controllers can stabilise the system at that particular time.

4.1 Combination of Controller Actions.

Figure 4 shows how that a new element ρ_i is introduced, this is a variable gain that basically selects which individual controller output is to be placed on-line with the process. Recalling from sections 2 and 3, that the selection of the controller from the family set is unique, i.e. only one of the controllers is utilised at only one time. This new gain can then be described as follows,

$$\rho_i \in \Re^1, \qquad \sum_{i=1}^{n} \rho_i = 1 \qquad (20)$$

The range of ρ_i is either 0 or 1, where

0 Ł indicates that controller (n) is in-active
1 Ł indicates that controller (n) is active

Instinctively, we could add the final control outputs from each local region to form

$$u = \rho_1 . u_1 + \rho_{i+1} . u_{i+1} + \ldots \rho_n . u_n .$$

However, further analysis indicates that it is preferable to use a parallel addition of controller outputs rather then a series connection, i.e.

$$u = \frac{1}{\frac{\rho_1}{u_1} + \frac{\rho_{i+1}}{u_{i+1}} + \cdots + \frac{\rho_n}{u_n}} . \qquad (21)$$

As implied previously the states of the relevant controllers before a switch occurs should be similar when an overlap regime occurs. However from figure 4 we notice that u_i and u_{i+1} are not identical, therefore producing a bump when the switching takes place. This is a direct result of the memory states being matched but not the controller outputs. Initially, the controller output of the latent controller was forced via feedback to equal the current controller output [7]. To correct this mismatch between controller outputs (u_i) a sliding switch was implemented. Consequently, even-though the switch between individual controllers is discrete the final control switching is not. The difference however between this approach and other fuzzy switching techniques is that the fuzzy nature of the switch is only limited here to the high frequency gain term and does not consider plant phenomena as has been traditional. Rather it is only utilised for the switch implementation. Hence instead of (ρ_i) and (ρ_{i+1}) behaving as in figure 5a, they respond as in figure 5b. The parallel combination as described earlier, (21) is crucial for success of this scheme.

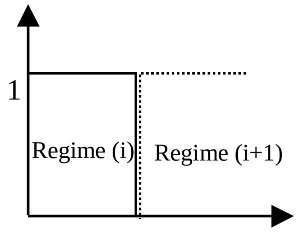

Fig. 5a Discrete Switching

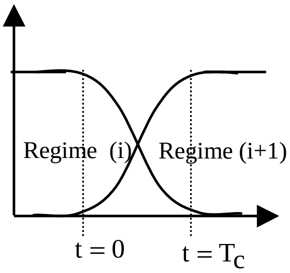

Fig.5b Fuzzy Switching

Obviously, the value of T_c is a fundamental parameter in the performance of the switching algorithm. If T_c is too large then the assumption that the model validity is the same during the transition period will no longer apply, added to this the plant dynamics might change, requiring that the current local controller be changed to another of the local controller from the family set. The choice of the largest possible T_c is defined by an upper limit, which is set by following two stability constraints, [14].

5. CASE STUDY: PILOT DISTILLATION COLUMN CONTROL

The theory developed above has been applied to a continuous pilot binary ethanol/water distillation column. The process is a 23inch-diameter bubble 12-plate column. The control of the column is structured as a decentralized controller, where the condenser duty is controlled by the distillate flow and the bottom reboiler duty is controlled by the residue flow. These two control loops control the material inventory of the system. Meanwhile, controlling the temperature at specific plates indirectly regulates the concentration of the residue and product.

Through experimental analysis it has been shown that an LV (reflux/steam), [14], configuration is the most desirable selection of control pairings for controlling the concentration. Therefore the concentration of the residue is controlled via the steam rate while the product concentration is controlled by manipulation of the reflux flow.

To produce the nonlinearity within the system, the operating point of the plant is altered. The manner in which the system is moved through its operating range is via the feed rate. Therefore, for this system the feed rate is set as the trajectory vector $\lambda(t)$. Unfortunately, the system is also nonlinear with respect to the steam rate. Figures 6a and 6b illustrate the nonlinearity of the system.

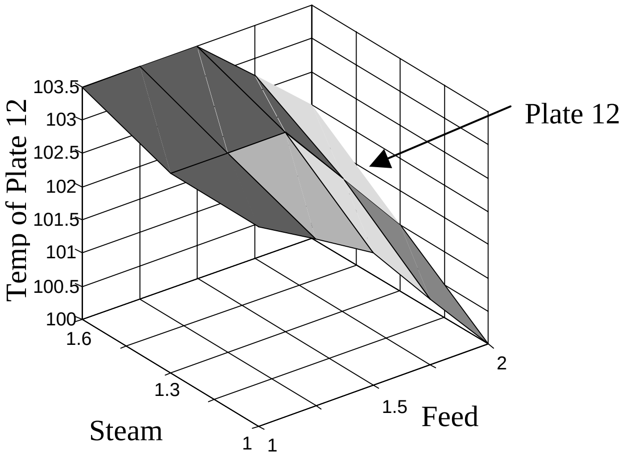

Figure 6a Bottom Plate Nonlinearity

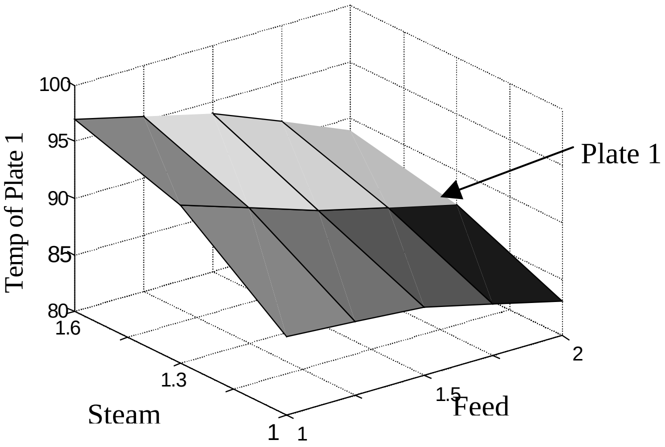

Figure 6b Top Plate Nonlinearity

The nonlinear system is then divided into two quasi-linear regions of operation, with respect to feed rate. Since the steam rate is a manipulated variable two local nonlinear controllers are designed with respect to the steam rate.

The global model obtained is described by

$$\begin{bmatrix} T_1 \\ T_{12} \end{bmatrix} = \begin{bmatrix} \dfrac{-13}{5s+1} & \dfrac{23.3}{71.6s^2+16.88s+1} \\ 0 & \dfrac{20\Phi(\text{steam})}{0.366s^2+0.866s+1} \end{bmatrix} \cdot \begin{bmatrix} \text{Reflux} \\ \text{Steam} \end{bmatrix} \tag{22}$$

and

$$\begin{bmatrix} T_1 \\ T_{12} \end{bmatrix} = \begin{bmatrix} \dfrac{-11.5}{5s+1} & \dfrac{13.3.e^{-20s}}{71.6s^2+16.18s+1} \\ 0 & \dfrac{33.6\Phi(\text{steam})}{2.25s^2+3s+1} \end{bmatrix} \cdot \begin{bmatrix} \text{Reflux} \\ \text{Steam} \end{bmatrix} \tag{23}$$

where $\Phi(\text{steam}) = \begin{cases} \text{steam} & \partial\text{steam}/\partial t \geq 0 \\ 0.33 \cdot \text{steam} & \partial\text{steam}/\partial t < 0 \end{cases}$ and (22) is for a feed rate of 1L/min and (23) for a rate of 2L/min.

5.1 Supervisory Multiple Model Control Design.

The complete controller design for the above nonlinear system was carried out as follows.

1. Once identifying the local regions, a test for global stability was verified (using the LMI toolbox in Matlab), fortunately, in this example two local models were enough to obtain global stability.
2. The multiple-model H_∞ controller was designed for both regions of operation, with the uncertainty of models represented by the weighting functions given by,

   $$W_1 = \frac{(s+6)^2}{100(s+0.0001)(s+0.6)} \qquad W_2 = \frac{10\sqrt{2}s}{s+35}.$$

   Each local controller was tuned such that a small region of overlap was present, this was to guarantee stability and performance during transient conditions, as well as, to satisfy the requirement for Bumpless Transfer.

3. In the supervisory section the LQ local observers were initially obtained with a unique global closed-loop gain. Unfortunately, the local models were reasonably similar in structure, as a result the observers continuously oscillated between themselves, even with presence of the hysteresis component To circumvent this problem the observer gain was reduced to zero, thus producing an open-loop model. Fortunately, the signal-to-noise ratio was large, therefore, there was no noise difficulty, and consequently justifying the open-loop arrangement.
4. In the Bumpless Transfer switching mechanism, the only tuning necessary was is the weightings on the smoothing functions as well as the maximum size of T_C .

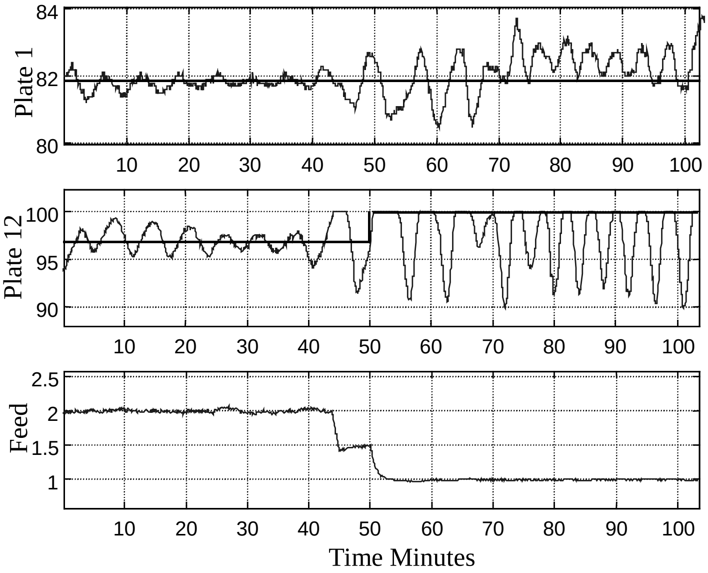

Figure 7 Time-invariant MIMO controller

5.2 Case Study Results.

To illustrate the performance improvement, a linear time-invariant MIMO controller was initially designed for the plant. Figure 7 shows the poor performance of the controller once the system moves outside the predefined operating range for which the controller was initially designed. While figure 8 illustrates the improved performance achieved when the MMC was utilised. Unfortunately, the performance did momentarily decrease during the transition region of operation. This is mainly due to the modelling error of the multiple model approach. A further observation can be seen in figure 9, where the selection of the local controllers is shown by the error signal produced by the local observers (models). The switch from one controller to the other did not occur simultaneous to the feed change.

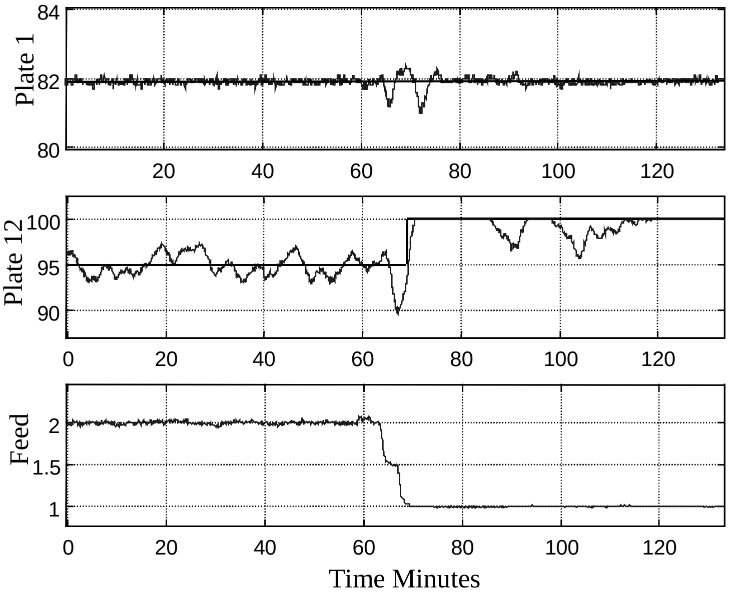

Figure 8 Multiple Model MIMO controller

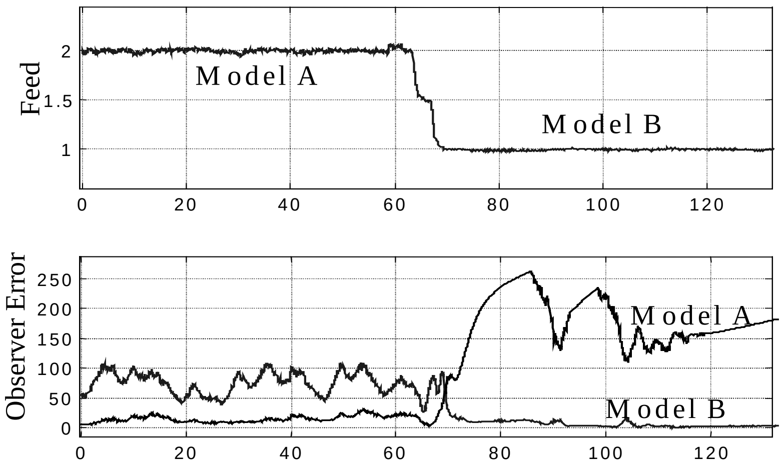

Figure 9 Selection Mechanism

This lag in control transfer is due to the response time between the feed change and the internal dynamics of the distillation column. We note that the response time will not be the same in the return path of the same trajectory, this is due to fundamental material balance phenomena, [14].

6. CONCLUSION

This paper has described the complete supervisory MMC design, aimed at controlling nonlinear systems. The result is valid for nonlinear systems, which can be approximated using an LPV formulation.

This assumption is extremely important. Fortunately, it appears valid in many practical cases including Hybrid aircraft (fighter planes), chemical plants etc.

The major advantage of the proposed structure is that it offers a bridge to linear control theory for nonlinear systems. The identification of each of the local regions is highly dependent on the process in hand. With respect to the case study it was found that the structure of the supervisor depends on the process.

If the noise element is increased the hysteresis component should be implemented. Here local linear controllers were utilised in the design. This should be contrasted with the use of a single nonlinear controller which would be considerably more difficult to conceptualize in design. It can be shown that local linear controllers can be identified with this system, however, the implementation is complicated by the fact that the local models are only valid in small operating regions. This design trade-offs are required in the MMC system. However, this is not unexpected since, fundamentally there are always compromises to be made between complexity and implementability.

7. REFERENCES

[1] Apkarian P, Gahinet P and Becker Greg. (1995). Self-schedule H_∞ Control of Linear Parameter varying Systems. A Design Example, *Automatica* Vol. 31 No 9 pp. 1251-1261.

[2] Apkarian P, Gahinet P and Becker Greg. (1994). Becker. Self-scheduled H_∞ control of linear parameter-varying systems. *In Proceedings of the American Control Conference,* p.p. 856-860, June.

[3] Banerjee A, Arkun Y and Ogunnaike B, Pearson R, (1997). Estimation of Nonlinear Systems using Linear Mulitple Models, *AIChE Journal*, May. Vol.43, No.5.

[4] Banerjee A, Arkun Y, and Pearson R, Ogunnaike B, (1995). H_∞ Control of Nonlinear Process using Multiple Linear Models, *ECC.*

[5] Doyle, J. (1985). Structured uncertainty in control system design. *In Proceeding of the IEEE Conference on Decision and Control*, p.p. 260-265.

[6] Friedman J.H (1991). Multivariate Adaptive Regression Splines, *Annals of Statitics* 19, 141.

[7] Graebe S.F and Anders A, (1995). Dynamic transfer among alternative controllers and its relation to anti-windup controller design, *IEEE Transactions on Control Systems Technology.*

[8] Johansen J and Foss B, (1993a). Constructing NARMAX models using ARMAX models. *Int J Control* 58, p.p. 1125-1153.

[9] Kulkarni S.R and Ramadge P, (1996). Model and Controller Selection Policies based on Output Prediction Errors, *IEEE Trans, Automatic Control,* Vol. 41 No. 11, Nov.

[10] Morse A.S, (1996). Supervisory Control of Families of Linear Set-point Controllers Part1 Exact Matching, *IEEE Transactions on Automatic Control* Vol.41 No.10 Oct.

[11] Morse A.S, (1995). Control using logic-based switching, *In Trends in Control, A Isidori,* Ed. New York Springer-Verlag, p.p. 69-113.

[12] Rodriguez J, Goodwin G.C and Romagnoli J (1997). Dynamic Bumpless Transfer Mechanism, *In Proceedings Control 97 Australia*, 497-502.

[13] Rodriguez J, Romagnoli J and Goodwin G (1998). Supervisory Multiple Model Control, *Dycops 5 Corfu, Greece*.

[14] Rodriguez J. Ph.D. Thesis, Chemical Engineering University of Sydney Australia. To be completed.

[15] Shamma J.S and Athans M, (1992). Gain-scheduling Potential Hazards and possible Remedies. *IEEE Control Systems Magazine* 12(3), p.p. 101-107.

[16] Weller S.R and Goodwin G.C, (1994). Hysteresis Switching Adaptive Control of Linear Multivariable Systems. *IEEE Trans, Automatic Control*, Vol. 39, No**. 7**, July.