Brazilian Journal
of Chemical
Engineering

ISSN 0104-6632
Printed in Brazil
www.abeq.org.br/bjche

Vol. 31, No. 01, pp. 131 - 144, January - March, 2014

MULTI-MODEL MPC WITH OUTPUT FEEDBACK

J. M. Perez[1*], D. Odloak[2] and E. L. Lima[3]

[1]PETROBRAS – Cenpes, Avenida Horacio Macedo, 950, C.P. 21941-915, Rio de Janeiro - RJ, Brazil.
E-mail: perez@petrobras.com.br
[2]Chemical Engineering Department, University of São Paulo,
C.P. 61548, 05424-970, São Paulo - SP, Brazil.
E-mail: odloak@usp.br
[3]Chemical Engineering Program/COPPE Federal University of Rio de Janeiro,
C.P. 68502, 21941-970, Rio de Janeiro - RJ, Brazil.
E-mail: enrique@peq.coppe.ufrj.br

(Submitted: February 21, 2012 ; Revised: April 3, 2013 ; Accepted: April 23, 2013)

Abstract - In this work, a new formulation is presented for the model predictive control (MPC) of a process system that is represented by a finite set of models, each one corresponding to a different operating point. The general case is considered of systems with stable and integrating outputs in closed-loop with output feedback. For this purpose, the controller is based on a non-minimal order model where the state is built with the measured outputs and the manipulated inputs of the control system. Therefore, the state can be considered as perfectly known and, consequently, there is no need to include a state observer in the control loop. This property of the proposed modeling approach is convenient to extend previous stability results of the closed loop system with robust MPC controllers based on state feedback. The controller proposed here is based on the solution of two optimization problems that are solved sequentially at the same time step. The method is illustrated with a simulated example of the process industry. The rigorous simulation of the control of an adiabatic flash of a multi-component hydrocarbon mixture illustrates the application of the robust controller. The dynamic simulation of this process is performed using EMSO – Environment Model Simulation and Optimization. Finally, a comparison with a linear MPC using a single model is presented.
Keywords: Model Predictive Control; Robust stability; Output feedback; Multi-model uncertainty.

INTRODUCTION

Model predictive control has achieved remarkable popularity, mainly in the oil refining and petrochemical industries, with thousands of practical applications (Qin and Badgwell, 2003). One of the main reasons for this industrial acceptance is the ability to incorporate inputs and output constraints in the formulation of the control problem. However, one desirable characteristic, the closed-loop robust stability, is still not attended by existing commercial MPC packages. Model uncertainty may occur when a linear model is used to approximate the model of a nonlinear system. This approximation may be satisfactory near the operating point where the linear model is obtained. However, if the operating point is moved due to disturbances or different product specifications, model uncertainty starts to play a critical role in the performance and stability of the control system.

Stable MPC approaches for the nominal system (without uncertainty) and state feedback have been proposed in the control literature since the seminal paper of Rawlings and Musk (1993) that proposed the infinite horizon MPC. Based on that work, Carrapiço and Odloak (2005) developed a MPC with nominal stability for systems with stable and integrating outputs for the case where the steady-state is not known. In that method, a state space model was considered in the incremental form and a modified control objective function that included slack variables to force the control objective function to be

*To whom correspondence should be addressed bounded. It was shown that the approach makes the MPC controller globally convergent for any control horizon. To deal with the case of output feedback, several authors (Perez, 2006; Wang and Young, 2005) proposed to design the MPC using the non-minimal state space model proposed in Maciejowski (2002). González *et al*. (2009) adopted the non-minimal model extended to the incremental form as in Perez (2006) to develop an infinite horizon MPC to the nominal case. Some of the ideas presented in González *et al*. (2009) and Carrapiço and Odloak (2005) will be followed in this work and applied to the development of the multi-model robust control of systems with integrating outputs.

When model uncertainty is considered in the synthesis of the predictive controller, many of the existing approaches lead to a computational effort that may be prohibitive for practical implementation (Lee and Cooley, 2000, Kothare *et al*., 1996). In fact we could not find in the control literature any application of a robust MPC in a real system. So, the implementation of a robust MPC is still an open problem.

In the area of robust MPC, the work of Badgwell (1997) can be considered an early breakthrough towards the practical implementation of a robust multi-model MPC. The main weakness of the work of Badgwell (1997) is related to the need that the system steady-state should be known. To circumvent this problem, Odloak (2004) adopted the Output Prediction Oriented Model (OPOM) (Rodrigues and Odloak, 2003), which is a state space model equivalent to the step response model in the analytical form, to develop the multi-model MPC for stable systems. González and Odloak (2007) extended the work of Odloak (2004) to systems of stable and integrating outputs. OPOM is a state space model, which is incremental in the input and, consequently, at any steady-state its input is null. Also, it can be shown that the state of OPOM is trivially related to the output steady state. Then, for a given output the steady state is uniquely defined even when the model is not exactly known. The robust multi-model MPCs mentioned above are all built on the assumption that the system state is perfectly known (or measured), which is not usually true. The straightforward solution to this problem is the adoption of a state observer that feeds the MPC with the estimated state. But, since the consideration of constraints makes the MPC a nonlinear controller, the separation principle cannot be applied to guarantee the robust stability of the closed-loop system with the multi-model MPC. A possible answer to this question is to adopt the non-minimal state space model (realigned model) considered in Perez (2006), where the state is built with the past values of the output and input increments. To simplify the presentation of the controller, it is assumed that we have integrating and stable outputs that are treated separately.

This paper is developed as follows: first, the realigned incremental model presented in Perez (2006) is briefly summarized and the optimization problems that define the multi-model MPC are presented. Then, it is shown that the proposed controller is able to stabilize the multi-model plant. Next, a simulation example is presented where the robust controller is applied in a flash drum where a complex hydrocarbon mixture is flashed. The process system is simulated through the dynamic simulator EMSO-Environment for Modeling Simulation and Optimization (Soares and Secchi, 2003). This example is used to illustrate the application potential of the new controller.

THE INCREMENTAL REALIGNED MODEL

The state space realigned model proposed in Maciejowski (2002) is equivalent to the following time difference model:

$$y(k)+\sum_{i=1}^{na} A_i y(k-i)=\sum_{i=1}^{nb} B_i u(k-i) \tag{1}$$

where na and nb define the order of the model, A_i and B_i are the model parameters. Perez (2006) proposed to write the realigned model in the incremental form as follows:

$$\begin{bmatrix} x_y(k+1) \\ x_{\Delta u}(k+1) \end{bmatrix}=\begin{bmatrix} A_y & A_{\Delta u} \\ \underline{0} & \underline{I} \end{bmatrix}\begin{bmatrix} x_y(k) \\ x_{\Delta u}(k) \end{bmatrix}+\begin{bmatrix} B_{\Delta u} \\ \overline{I} \end{bmatrix}\Delta u(k) \tag{2}$$

$$y(k)=\begin{bmatrix} C_y & C_{\Delta u} \end{bmatrix}\begin{bmatrix} x_y(k) \\ x_{\Delta u}(k) \end{bmatrix} \tag{3}$$

where:

$$x_y(k)=\begin{bmatrix} y(k)^T & y(k-1)^T & \cdots & y(k-na+1)^T & y(k-na)^T \end{bmatrix}^T \in \Re^{(na+1)ny}$$

$$x_{\Delta u}(k)=\begin{bmatrix}\Delta u(k-1)^T & \Delta u(k-2)^T & \cdots & \Delta u(k-nb+1)^T\end{bmatrix}^T \in \Re^{(nb-1)nu}$$

k is the present sampling instant, $\Delta u(k)=u(k)-u(k-1)$ is the vector of input increments ($\Delta u(k)\in\Re^{nu}$) and $y\in\Re^{ny}$ is the vector of controlled outputs.

The matrices involved in the model defined by (2) and (3) are:

$$A_y=\begin{bmatrix} I_{ny}-A_1 & A_1-A_2 & A_2-A_3 & A_3-A_4 & \cdots & A_{na-1}-A_{na} & A_{na}\\ I_{ny} & 0 & 0 & 0 & \cdots & 0 & 0\\ 0 & I_{ny} & 0 & 0 & \cdots & 0 & 0\\ 0 & 0 & I_{ny} & 0 & \cdots & 0 & 0\\ 0 & 0 & 0 & I_{ny} & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & 0 & \cdots & I_{ny} & 0 \end{bmatrix}, A_y\in\Re^{(na+1).ny\times(na+1).ny}$$

$$A_{\Delta u}=\begin{bmatrix} B_2 & \cdots & B_{nb}\\ 0_{ny\times nu} & \vdots & 0_{ny\times nu}\\ 0_{ny\times nu} & \vdots & 0_{ny\times nu}\\ \vdots & \ddots & \vdots\\ 0_{ny\times nu} & \cdots & 0_{ny\times nu}\end{bmatrix}, A_{\Delta u}\in\Re^{(na+1).ny\times(nb-1).nu},$$

$$\underline{0}=\begin{bmatrix} 0_{nu\times ny} & 0_{nu\times ny} & \cdots & 0_{nu\times ny} & 0_{nu\times ny}\\ 0_{nu\times ny} & 0_{nu\times ny} & \cdots & 0_{nu\times ny} & 0_{nu\times ny}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0_{nu\times ny} & 0_{nu\times ny} & \cdots & 0_{nu\times ny} & 0_{nu\times ny}\end{bmatrix}, \underline{0}\in\Re^{(nb-1).nu\times(na+1).ny}$$

$$\underline{I}=\begin{bmatrix} 0_{nu} & \cdots & 0_{nu}\\ I_{nu} & \cdots & 0_{nu}\\ \vdots & \ddots & \vdots\\ 0_{nu} & I_{nu} & 0_{nu}\end{bmatrix}, \quad \underline{I}\in\Re^{(nb-1)nu\times(nb-1)nu},$$

$$B_{\Delta u}=\begin{bmatrix}B_1\\ 0\end{bmatrix}=\begin{bmatrix} B_1\\ 0_{ny\times nu}\\ \vdots\\ 0_{ny\times nu}\end{bmatrix}, B_1\in\Re^{ny\times nu}, B_{\Delta u}\in\Re^{[(na+1).ny]\times nu},$$

$$\overline{I}=\begin{bmatrix} I_{nu}\\ 0_{nu}\\ \vdots\\ 0_{nu}\end{bmatrix}, \overline{I}\in\Re^{[(nb-1).nu]\times nu},\ C_y=\begin{bmatrix}\underbrace{I_{ny} \quad 0_{ny\times ny} \quad \cdots \quad 0_{ny\times ny}}_{na+1}\end{bmatrix}\ C_{\Delta u}=\begin{bmatrix}\underbrace{0_{ny\times nu} \quad 0_{ny\times nu} \quad \cdots \quad 0_{ny\times nu}}_{nb-1}\end{bmatrix}$$

Observe that the state of the system, defined in (2) and (3), has components $x_y(k)$, which depend on the last *na*+1 values of the output, and $x_{\Delta u}(k)$, which depend on the last *nb*-1 implemented input increments. This means that the proposed state can be built with the past values of the controlled outputs and manipulated inputs. Then, there is no need to include a state estimator to compute the present state that, as we will show in the next section, is necessary to solve the MPC problem that calculates the optimum control sequence. Also, the motivation for the designation of this sort of model as a realigned model becomes clear as the prediction of the future outputs is realigned with the past inputs and outputs.

Note also that, in the case of the multi-model system, the model matrices that appear in Equations (2) and (3) will be different for each of the models. This means that for each model, we will have a different set of matrices $\left(A_y, A_{\Delta u}, B_{\Delta u}\right)$. Now, if we associate a model with θ_p, where *p*=1,…,*L*, then each model will be defined as $\left(A_y(\theta_p), A_{\Delta u}(\theta_p), B_{\Delta u}(\theta_p)\right)$.

Since we are assuming that the system to be controlled has only stable and pure integrating outputs, and for each of these two classes of outputs the realigned model takes a different form, we find it appropriate to show the model in details for each of the two types of outputs.

For the integrating outputs, the model defined in (2) and (3) can be written as follows:

$$\underbrace{\begin{bmatrix} x_y^i(k+1) \\ x_{\Delta u}^i(k+1) \end{bmatrix}}_{x^i(k+1)} = \underbrace{\begin{bmatrix} A_y^i & A_{\Delta u}^i \\ \underline{0} & \underline{I} \end{bmatrix}}_{A^i} \underbrace{\begin{bmatrix} x_y^i(k) \\ x_{\Delta u}^i(k) \end{bmatrix}}_{x^i(k)} + \underbrace{\begin{bmatrix} B_{\Delta u}^i \\ \overline{I} \end{bmatrix}}_{B^i} \Delta u(k) \tag{4}$$

$$y^i(k) = \begin{bmatrix} C_y^i & C_{\Delta u}^i \end{bmatrix} \begin{bmatrix} x_y^i(k) \\ x_{\Delta u}^i(k) \end{bmatrix} \tag{5}$$

Analogously, for the stable outputs, the model defined in (2) and (3) corresponding to model θp can be written as follows:

$$\underbrace{\begin{bmatrix} x_{y,\theta_p}^s(k+1) \\ x_{\Delta u}^s(k+1) \end{bmatrix}}_{x_{\theta_p}^s(k+1)} = \underbrace{\begin{bmatrix} A_y^s(\theta_p) & A_{\Delta u}^s(\theta_p) \\ \underline{0} & \underline{I} \end{bmatrix}}_{A^s(\theta_p)} \underbrace{\begin{bmatrix} x_{y,\theta_p}^s(k) \\ x_{\Delta u}^s(k) \end{bmatrix}}_{x_{\theta_p}^s(k)} + \underbrace{\begin{bmatrix} B_{\Delta u}^s(\theta_p) \\ \overline{I} \end{bmatrix}}_{B^s(\theta_p)} \Delta u(k) \tag{6}$$

$$y_{\theta_p}^s(k) = \begin{bmatrix} C_y^s & C_{\Delta u}^s \end{bmatrix} \begin{bmatrix} x_{y,\theta_p}^s(k) \\ x_{\Delta u}^s(k) \end{bmatrix} \tag{7}$$

Since MPC always considers a finite control horizon *m*, it is obvious that $\Delta u(k+j\,|\,k)=0$ for any $j \ge m$, and, by simply applying the model defined in (2) at successive time steps, it is easy to show that if *n*= *m*+*nb*-1, the following relation will be true:

$$\begin{bmatrix} x_y(k+n+1\,|\,k) \\ x_{\Delta u}(k+n+1\,|\,k) \end{bmatrix} = \begin{bmatrix} A_y & A_{\Delta u} \\ \underline{0} & \underline{I} \end{bmatrix} \begin{bmatrix} x_y(k+n\,|\,k) \\ \underbrace{x_{\Delta u}(k+n\,|\,k)}_{=0} \end{bmatrix} + \begin{bmatrix} B_{\Delta u} \\ \overline{I} \end{bmatrix} \underbrace{\Delta u(k+n\,|\,k)}_{=0}$$

and, consequently

$$x_y(k+n+1\,|\,k) = A_y x_y(k+n\,|\,k) \tag{8}$$

This means that after time step *n*, the model defined in (2) will evolve according to the model defined in (8). Although the state matrix of the model defined in (2) is not of full rank, A_y is of full rank and allows the following Jordan eigenvalue-eigenvector decomposition

$$A_y V = V A_d \tag{9}$$

where A_d contains the eigenvalues of A_y in its diagonal and *V*, which is invertible, has the corresponding eigenvectors as its columns. It should be noted that in A_d we also have the integrating modes of the model resulting from the incremental form of the model.

It is now possible to define a reduced state, here designated *z*, that represents the system after the time step *n*. For instance, for the integrating system defined in (4) and (5), we can define

$$x_y^i(k) = V_i z^i(k)$$

$$x_y^i(k) = \begin{bmatrix} V_i^{\Delta} & V_i^i \end{bmatrix} \begin{bmatrix} z_i^{\Delta}(k) \\ z_i^i(k) \end{bmatrix} \quad (10)$$

where the state components z_i^{Δ} and z_i^i correspond to the integrating modes produced by the incremental input and the true integrating modes of the system, respectively. V_i is obtained from the Jordan decomposition of matrix A_y^i as follows:

$$A_y^i V_i = V_i A_d^i$$

If the state transformation defined in (9) is applied to the model defined in (4), which characterizes the integrating outputs, the following reduced state matrix is obtained:

$$A_d^i = \begin{bmatrix} I_{ny_i} & I_{ny_i} \\ 0 & I_{ny_i} \end{bmatrix} \quad (11)$$

where ny_i is the number of integrating outputs. Then, after time step n the model defined in (4) can be written in terms of the transformed state as follows:

$$\begin{bmatrix} z_i^{\Delta}(k+n+1) \\ z_i^i(k+n+1) \end{bmatrix} = \begin{bmatrix} I_{ny_i} & I_{ny_i} \\ 0 & I_{ny_i} \end{bmatrix} \begin{bmatrix} z_i^{\Delta}(k+n) \\ z_i^i(k+n) \end{bmatrix} \quad (12)$$

Analogously, for the model defined in (6), which corresponds to the stable outputs, the state matrix of the transformed state becomes:

$$A_{d,\theta_p}^s = \begin{bmatrix} I_{ny_s} & 0 \\ 0 & F_{\theta_p}^s \end{bmatrix} \quad (13)$$

where ny_s is the number of stable outputs, and the state transformation is defined as follows:

$$x_{y,\theta_p}^s(k) = V_{s,\theta_p} z_{\theta_p}^s(k)$$
$$= \begin{bmatrix} V_{s,\theta_p}^{\Delta} & V_{s,\theta_p}^s \end{bmatrix} \begin{bmatrix} z_{s,\theta_p}^{\Delta}(k) \\ z_{s,\theta_p}^s(k) \end{bmatrix} \quad (14)$$

where z_{s,θ_p}^{Δ} corresponds to the integrating states related with the incremental form of the model and z_{s,θ_p}^s is the state associated with the stable states of the system corresponding to the parameters represented as θ_p. V_{s,θ_p} is obtained from the Jordan decomposition of matrix $A_y^s(\theta_p)$ as follows:

$$A_y^s(\theta_p) V_{s,\theta_p} = V_{s,\theta_p} A_{d,\theta_p}^s$$

If the stable poles of the system are non-repeated, $F_{\theta_p}^s$ is a diagonal matrix containing the eigenvalues of $A_y^s(\theta_p)$. After time step $k+n$, the transformed state corresponding to the stable outputs will evolve according to the equation

$$\begin{bmatrix} z_{s,\theta_p}^{\Delta}(k+n+1) \\ z_{s,\theta_p}^s(k+n+1) \end{bmatrix} = \begin{bmatrix} I_{nys} & 0 \\ 0 & F_{\theta_p}^s \end{bmatrix} \begin{bmatrix} z_{s,\theta_p}^{\Delta}(k+n) \\ z_{s,\theta_p}^s(k+n) \end{bmatrix} \quad (15)$$

In the next section, the model equations presented in this section for both the integrating and stable output will be applied in the development of the robust multi-model MPC for systems with stable and integrating outputs.

ROBUST MULTI-MODEL MPC WITH REALIGNED MODEL

As explained in the previous section, the multi-model MPC presented here considers that although the same set of *nu* manipulated inputs should be used to control the system outputs, these outputs can be separated in two groups. In the first group, we have the integrating outputs and, in the second group, we consider the stable outputs. In the objective function of the robust controller, an infinite prediction horizon is considered. To avoid the objective function from becoming infinite, restrictions have to be included in order to force the integrating states to become null at the end of the extended control horizon. To guarantee that the optimization problem that defines the controller is feasible even when the desired steady-state is unreachable, slack variables are included as decision variables of the optimization problem. Also, besides these assumptions, in the robust controller proposed here, it is assumed that model uncertainty is related with the stable outputs. The justification for this assumption comes from the practical observation that, in industry, most of the integrating outputs are related to the level control of liquid in drums, bottom of distillation column, etc. In this kind of equipment the dynamics of the output (liquid level) are mainly related to the geometric configuration and dimension of the vessel, which are usually well known. So, uncertainty in these outputs is small or negligible. To illustrate how the controller is built, assume that the system is represented by a model whose parameters are defined by the set θ_p and we associate to this model the following objective function:

$$\begin{aligned}
J_k\left(\theta_p\right) = &\sum_{j=1}^{n} \left[y^i\left(k+j|k\right) - y_i^{sp} - \delta_{\Delta,k}^i - j\delta_{i,k}^i\right]^T Q_i \left[y^i\left(k+j|k\right) - y_i^{sp} - \delta_{\Delta,k}^i - j\delta_{i,k}^i\right] \\
&+ \sum_{j=n+1}^{\infty} \left[y^i\left(k+j|k\right) - y_i^{sp} - \delta_{\Delta,k}^i - j\delta_{i,k}^i\right]^T Q_i \left[y^i\left(k+j|k\right) - y_i^{sp} - \delta_{\Delta,k}^i - j\delta_{i,k}^i\right] \\
&+ \sum_{j=1}^{n} \left[y_{\theta_p}^s\left(k+j|k\right) - y_s^{sp} - \delta_{\Delta,k}^s\left(\theta_p\right)\right]^T Q_s \left[y_{\theta_p}^s\left(k+j|k\right) - y_s^{sp} - \delta_{\Delta,k}^s\left(\theta_p\right)\right] \\
&+ \sum_{j=n+1}^{\infty} \left[y_{\theta_p}^s\left(k+j|k\right) - y_s^{sp} - \delta_{\Delta,k}^s\left(\theta_p\right)\right]^T Q_s \left[y_{\theta_p}^s\left(k+j|k\right) - y_s^{sp} - \delta_{\Delta,k}^s\left(\theta_p\right)\right] \\
&+ \sum_{j=0}^{m-1} \Delta u\left(k+j|k\right)^T R \Delta u\left(k+j|k\right) + \left(\delta_{\Delta,k}^i\right)^T S_\Delta^i \delta_{\Delta,k}^i + \left(\delta_{\Delta,k}^s\left(\theta_p\right)\right)^T S_\Delta^s \delta_{\Delta,k}^s\left(\theta_p\right) + \left(\delta_{i,k}^i\right)^T S_i^i \delta_{i,k}^i
\end{aligned} \quad (16)$$

where $y^i\left(k+j|k\right)$ is the prediction of the integrating output at time $k+j$ computed at time k, $y_{\theta_p}^s\left(k+j|k\right)$ is the prediction of the stable output corresponding to model θ_p, y_i^{sp} is the vector of set-points to the integrating outputs, y_i^{sp} is the vector of set-points to the stable outputs, $\delta_{\Delta,k}^i$, $\delta_{\Delta,k}^s$ and $\delta_{i,k}^i$ are slack variables that, as it will be show next, are necessary to assure that the control problem is feasible and $J_k\left(\theta_p\right)$ is bounded. Q_i, Q_s, R, S_Δ^i, S_Δ^s and S_i^i are positive weighting matrices.

Observe that, using (5), the prediction of the integrating output can be written as follows:

$$\begin{aligned} y^i\left(k+n+j|k\right) = &C_y^i x_y^i\left(k+n+j|k\right) \\ &+ C_{\Delta u}^i x_{\Delta u}^i\left(k+n+j|k\right) \end{aligned} \quad (17)$$

But, since $x_{\Delta u}^i\left(k+n+j|k\right) = 0$, (17) becomes

$$y^i\left(k+n+j|k\right) = C_y^i x_y^i\left(k+n+j|k\right) \quad (18)$$

Then, introducing Equations (10) and (11) into (18), the following equation is obtained:

$$\begin{aligned} y^i\left(k+n+j|k\right) = &C_y^i V_i^\Delta \left[z_i^\Delta\left(k+n|k\right) + j z_i^i\left(k+n|k\right)\right] \\ &+ C_y^i V_i^i z_i^i\left(k+n|k\right) \end{aligned} \quad (19)$$

Now, Eq. (19) can be substituted into the infinite sum represented by the second term on the right hand side of (16). Then, it is easy to see that this term will be bounded if and only if the following constraints are satisfied:

$$\begin{aligned} &C_y^i V_i^\Delta z_i^\Delta\left(k+n|k\right) + C_y^i V_i^i z_i^i\left(k+n|k\right) \\ &- y_i^{sp} - \delta_{\Delta,k}^i - n\delta_{i,k}^i = 0 \end{aligned} \quad (20)$$

$$C_y^i V_i^\Delta z_i^i\left(k+n|k\right) - \delta_{i,k}^i = 0 \quad (21)$$

Now, consider the infinite sum represented by the fourth term on the right hand side of (16). In this term, the prediction of the stable output can be written as follows:

$$y_{\theta_p}^s\left(k+n+j|k\right) = C_y^s x_{y,\theta_p}^s (k+n+j\,|\,k) \quad (22)$$

Now, considering (14) and (15), Equation (22) becomes:

$$\begin{aligned} y_{\theta_p}^s\left(k+n+j|k\right) = &C_y^s V_{s,\theta_p}^\Delta z_{y,\theta_p}^\Delta (k+n\,|\,k) \\ &+ C_y^s V_{s,\theta_p}^s \left(F_{\theta_p}^s\right)^j z_{y,\theta_p}^s (k+n\,|\,k) \end{aligned} \quad (23)$$

Then, if (23) is substituted into (16), it is easy to see that the control objective function will be bounded only if the following constraint is satisfied:

$$C_y^s V_{s,\theta_p}^{\Delta} z_{y,\theta_p}^{\Delta}(k+n\,|\,k) - y_s^{sp} - \delta_{\Delta,k}^s(\theta_p) = 0 \tag{24}$$

Consequently, the optimization problem that produces the multi-model MPC proposed here will include the constraints defined in (20), (21) and (24). Particularly, constraint (24) will have to be written for all the models:

$$\theta_p,\ p = 1,...,L\,.$$

It should be noted that constraint (17) should be written in terms of the available states at time k and the vector of the future control moves. The desired expression is the following:

$$\begin{aligned}&\left[C_y^i V_i^{\Delta} N_1^i + C_y^i V_i^i N_2^i\right] V_i^{-1} N^i x^i(k+n\,|\,k)\\&-y_i^{sp} - \delta_{\Delta,k}^i - n\delta_{i,k}^i = 0\end{aligned} \tag{25}$$

where N_1^i is a matrix of ones and zeros that collects component z_i^{Δ} from z_i and matrix N_2^i collects component z_i^i from z_i. The model defined in (5) can be used to represent (25) in terms of the integrating state $x^i(k)$ that is measured as follows:

$$\begin{aligned}&\left[C_y^i V_i^{\Delta} N_1^i + C_y^i V_i^i N_2^i\right]\\&V_i^{-1} N^i \left((A^i)^{n-m} x^i(k) + B_m^i \Delta u_k\right)\\&-y_i^{sp} - \delta_{\Delta,k}^i - n\delta_{i,k}^i = 0\end{aligned} \tag{26}$$

where

$$\Delta u_k = \left[\Delta u(k\,|\,k)^T \quad \cdots \quad \Delta u(k+m-1\,|\,k)^T\right]^T$$

and

$$B_m^i = \left[(A^i)^{n-1} B^i \quad \cdots \quad (A^i)^{n-m} B^i\right]$$

Analogously, constraint (21) can also be represented in terms of the present state and the vector of control moves:

$$\begin{aligned}&C_y^i V_i^{\Delta} N_1^i V_i^{-1} N^i \left((A^i)^{n-m} x^i(k) + B_m^i \Delta u_k\right)\\&-\delta_{i,k}^i = 0\end{aligned} \tag{27}$$

Also, (21) can be written as follows:

$$\begin{aligned}&C_y^s V_{s,\theta_p}^{\Delta} N_1^s V_{s,\theta_p}^{-1}\\&N_2^s \left(A^s(\theta_p)^{n-m} x_{\theta_p}^s(k) + B_m^s(\theta_p)\Delta u_k\right)\\&-y_s^{sp} - \delta_{\Delta,k}^s(\theta_p) = 0\end{aligned} \tag{28}$$

where N_1^s captures the state component z_{s,θ_p}^{Δ} from state $z_{\theta_p}^s$ and N_2^s captures the component x_{y,θ_p}^s from state $x_{\theta_p}^s$. Considering the constraints discussed above, the control objective function defined in (16) can be written as follows

$$\begin{aligned}J_k(\theta_p) = &\sum_{j=1}^{n}\left[y^i(k+j|k) - y_i^{sp} - \delta_{\Delta,k}^i - j\delta_{i,k}^i\right]^T Q_i \left[y^i(k+j|k) - y_i^{sp} - \delta_{\Delta,k}^i - j\delta_{i,k}^i\right]\\&+\sum_{j=1}^{n}\left[y_{\theta_p}^s(k+j|k) - y_s^{sp} - \delta_{\Delta,k}^s(\theta_p)\right]^T Q_s \left[y_{\theta_p}^s(k+j|k) - y_s^{sp} - \delta_{\Delta,k}^s(\theta_p)\right]\\&+\ x_{\theta_p}^s(k+n\,|\,k)^T (N^s)^T (V_{s,\theta_p}^{-1})^T (N_3^s)^T \bar{Q}_s N_3^s V_{s,\theta_p}^{-1} N^s x_{\theta_p}^s(k+n\,|\,k)\\&+\sum_{j=0}^{m-1}\Delta u(k+j|k)^T R\Delta u(k+j|k) + \left(\delta_{\Delta,k}^i\right)^T S_{\Delta}^i \delta_{\Delta,k}^i + \left(\delta_{\Delta,k}^s(\theta_p)\right)^T S_{\Delta}^s \delta_{\Delta,k}^s(\theta_p) + \left(\delta_{i,k}^i\right)^T S_i^i \delta_{i,k}^i\end{aligned} \tag{29}$$

where $\bar{Q}_s$ is obtained from the solution to the following equation:

$$\bar{Q}_s - (F_{\theta_p}^s)^T \bar{Q}_s F_{\theta_p}^s = (F_{\theta_p}^s)^T (V_{s,\theta_p}^s)^T (C_y^s)^T Q_s C_y^s V_{s,\theta_p}^s F_{\theta_p}^s)$$

In González *et al*. (2007), it was shown that a MPC based on the minimization of an objective function similar to the one defined in (16), subject to the usual input constraints and the constraints that guarantee the feasibility of the controller and that the control objective is bounded, is stable if the control sequence is such that the integrating modes of the system are kept equal zero at the end of the control horizon. In the case considered here, this means that robust stability will happen if we can find a control sequence Δu_k that satisfies (27) with $\delta_{i,k}^i = 0$. Based on this information, at each time step *k*, we propose to obtain the control law of the multi-model robust MPC as the sequential solution to the following two problems:

Problem 1

$$\min_{\Delta u_{a,k},\delta_{i,k,a}^i} J_{k,a} = \left(\delta_{i,k,a}^i\right)^T S_i^i \delta_{i,k,a}^i$$

subject to

$$-\Delta u_{\max} \le \Delta u_a\left(k+j\middle|k\right) \le \Delta u_{\max}$$

$$u_{\min} \le u_a\left(k+j\middle|k\right) \le u_{\max}$$

$$C_y^i V_i^\Delta N_1^i V_i^{-1} N^i \left((A^i)^{n-m} x^i(k) + B_m^i \Delta u_{k,a}\right) - \delta_{i,k,a}^i = 0$$

where

$$\Delta u_{a,k} = \left[\Delta u_a(k\,|\,k)^T \;\cdots\; \Delta u_a(k+m-1\,|\,k)^T\right]^T$$

Problem 2

$$\min_{\substack{\Delta u_k,\delta_{i,k}^i,\delta_{\Delta,k}^i,\delta_{\Delta,k}^s(\theta_p)\\ p=1,\dots,L}} J_k\left(\theta_N\right)$$

subject to (26), (27) and

$$C_y^s V_{s,\theta_p}^\Delta N_1^s V_{s,\theta_p}^{-1} N_2^s\left(A^s(\theta_p)^{n-m} x_{\theta_p}^s(k) + B_m^s(\theta_p)\Delta u_k\right) - y_s^{sp} - \delta_{\Delta,k}^s(\theta_p) = 0 \qquad p = 1,\dots,L$$

$$-\Delta u_{\max} \le \Delta u\left(k+j\middle|k\right) \le \Delta u_{\max} \quad j = 0,\dots,m-1$$

$$u_{\min} \le u\left(k+j\middle|k\right) \le u_{\max} \quad j = 0,\dots,m-1$$

$$J_k\left(\theta_p\right) \le \tilde{J}_k\left(\theta_p\right) \quad p = 1,\dots,L \tag{30}$$

$$\delta_{i,k}^i = \delta_{i,k,a}^{i*} \tag{31}$$

where θ_N stands for the most probable or nominal model of the system and $\tilde{J}_k\left(\theta_p\right)$ is computed with $\Delta\tilde{u}_k = \begin{bmatrix}\Delta u^*\left(k\middle|k-1\right) & \Delta u^*\left(k+1\middle|k-1\right) & \cdots \\ \Delta u^*\left(k+m-2\middle|k-1\right) & 0\end{bmatrix}$ and $\tilde{\delta}_{\Delta,k}^i, \tilde{\delta}_{\Delta,k}^s(\theta_p), p = 1,\dots,L$, which are obtained from the solution to the following equations:

$$C_y^s V_{s,\theta_p}^\Delta N_1^s V_{s,\theta_p}^{-1} N_2^s\left(A^s(\theta_p)^{n-m} x_{\theta_p}^s(k) + B_m^s(\theta_p)\Delta\tilde{u}_k\right) - y_s^{sp} - \tilde{\delta}_{\Delta,k}^s(\theta_p) = 0 \qquad p = 1,\dots,L$$

$$\left[C_y^i V_i^\Delta N_1^i + C_y^i V_i^i N_2^i\right] V_i^{-1} N^i\left((A^i)^{n-m} x^i(k) + B_m^i \Delta\tilde{u}_k\right) - y_i^{sp} - \tilde{\delta}_{\Delta,k}^i - n\tilde{\delta}_{i,k}^i = 0$$

The inequality constraints defined in (30) represent the contraction of the control objective functions corresponding to each model of the set of models that characterize the multi-model uncertainty. Observe that $\Delta\tilde{u}_k$ is a feasible control sequence that is inherited from the optimal solution to Problem 2 at time step *k*-1.

At any time step *k*, Problem 1 is solved first and the optimal slack $\delta_{i,k,a}^{i*}$ is passed to Problem 2 that is solved at the same time step. From the solution of this second problem, the optimal control sequence Δu_k is obtained and the first control move $\Delta u(k\,|\,k)$ is injected into the true system and the controller is halted until time step *k*+1.

STABILITY OF THE MULTI-MODEL MPC

In this section, the stability of the multi-model robust MPC developed in the previous section is proved. We emphasize that the controller was developed for the quite general case where we have output feedback and systems with stable and integrating outputs. In the proofs that follows, it is assumed that the true plant model, which is here designated as θ_T, is one of the models of set $\Omega = \{\theta_1, \theta_2, ..., \theta_L\}$

Theorem 1

If the desired steady-state defined by the output set-points is reachable, then, for the undisturbed system, the sequential solution of problems 1 and 2 drives the slack of the integrating output $\delta^i_{i,k,a}$ to zero in a finite number of time steps.

Proof

Suppose that at time k, Problem 1 is solved and slack $\delta^{i*}_{i,k,a}$ is passed to Problem 2 that is solved at this same time step. From the vector of control moves $\Delta u(k \mid k)$ is implemented in the true process and we wait until time k+1 when Problem 1 is solved again. It is easy to show that at this time step $\left(\Delta \tilde{u}_k, \delta^{i*}_{i,k,a}\right)$ is a feasible solution to Problem 1 and corresponds to $\tilde{J}_{k+1,a} = J^*_{k,a}$. Consequently, solving Problem P1 will result in a control objective such that $J^*_{k+1,a} \le J^*_{k,a}$, where the equality will only hold if $\delta^{i*}_{i,k,a} = 0$. Then, the sequential solution of problems 1 and 2 will drive $J^*_{k,a}$ as well as $\delta^{i*}_{i,k,a}$ to zero. It is not difficult to show that this will happen in a finite number of time steps. This can be easily seen by observing the constraint $C^i_y V^{\Delta}_i N^i_1 V^{-1}_i N^i \left((A^i)^{n-m} x^i(k) + B^i_m \Delta u_{k,a}\right) - \delta^i_{i,k,a} = 0$. It is clear that, if there is no constraint on the input move, we can find $\Delta u_{k,a}$ that corresponds to $\delta^i_{i,k,a} = 0$. This means that Problem 1 will converge in a finite number of steps. If the input move is constrained but the desired steady-state is reachable, Problem 1 will still converge but with a larger number of time steps.

Theorem 2

Under the same hypothesis as in Theorem 1, after the convergence of slack $\delta^i_{i,k,a}$ to zero, the sequential solution of problem 1 and 2 will produce a sequence of control moves that will drive the system outputs to their set-points. If the output set-points are not reachable, the closed-loop system will converge to a steady-state where the outputs will lie at the minimum distance to the desired steady-state.

Proof

This theorem can be proved by following the same steps as the proof of the convergence of the robust MPC proposed in Gonzalez *et al*. (2007) for the case of integrating systems with state feedback.

APPLICATION OF THE ROBUST MULTI-MODEL MPC TO A FLASH DRUM SYSTEM

The robust multi-model MPC was tested by simulation using the EMSO-Environment for Modeling Simulation and Optimization (Soares and Secchi, 2003). The process considered here is a flash drum with two controlled variables: the level of liquid inside the drum (y_1), which is an integrating output, and the pressure in the drum (y_2), which is a stable output. The manipulated variables are the outlet gas flow rate (u_1) and the outlet liquid flow rate (u_2). Figure 1 shows the schematic representation of the flash drum considered here. The feed stream is a mixture of hydrocarbons. The Matlab routine ARX was used to identify the linear models that represent this process system around the normal operating point. These linear models were used in the multi-model MPC that controls the flash drum system. Basically, two different models were obtained corresponding to the regions where the drum pressure is above or below the normal operating pressure. Since there is no uncertainty related to the integrating output, the two models are the same as long as the integrating output is concerned. The two linear models that represent the flash drum system in the multi-model MPC are the following:

Model 1

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{-1.065\times 10^{-5}}{s} & \dfrac{-8.681\times 10^{-6}}{s} e^{-50s} \\ \dfrac{-0.09436s - 0.003964}{s^2 + 0.006119s + 1.576\times 10^{-5}} & \dfrac{-1.576\times 10^{-6}s - 3.176\times 10^{-6}}{s^2 + 0.0453s + 0.0001077} e^{-50s} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

Model 2

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{-1.065\times 10^{-5}}{s} & \dfrac{-8.681\times 10^{-6}}{s} e^{-50s} \\ \dfrac{-0.3749s - 0.01678}{s^2 + 0.01344s + 2.874\times 10^{-5}} & \dfrac{-1.576\times 10^{-6}s - 3.176\times 10^{-6}}{s^2 + 0.0453s + 0.0001077} e^{-50s} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

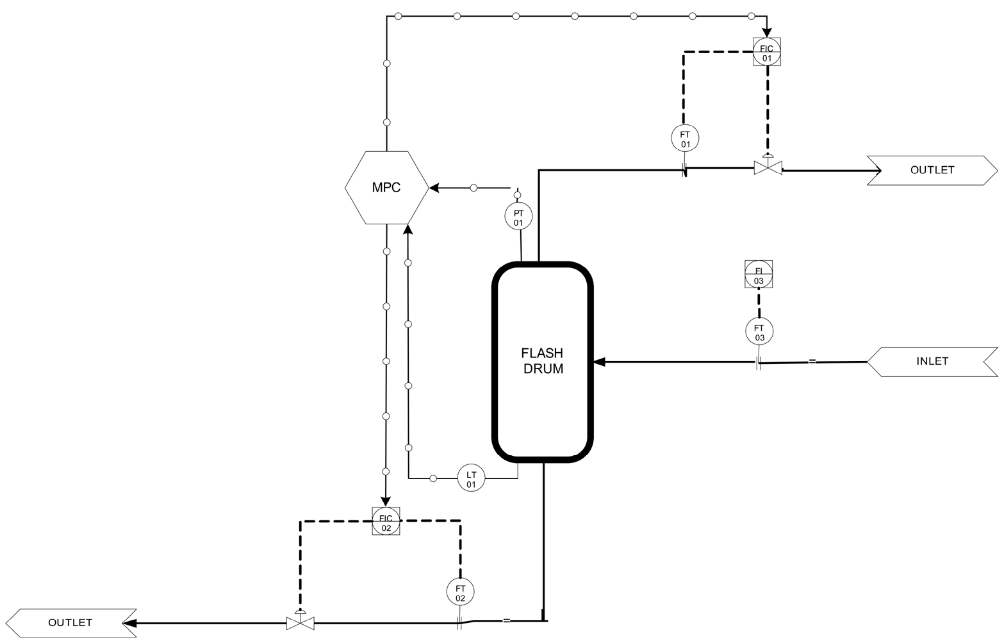

Figure 1: Schematic representation of the flash drum system

In the first simulation presented here, the set-point tracking of the drum liquid level and the drum pressure is studied. The tuning parameters used in the controller are the following: $T = 50s$; $m = 20$; $Q = diag[100 \quad 1]$; $R = diag[0.15 \quad 0.05]$, $S_\Delta^i = S_\Delta^s = S_i^i = 1\times 10^4$. The constraints of the manipulated inputs are $u_{max} = [1250 \quad 1250]$, $u_{min} = [-41 \quad -125]$ and $\Delta u_{max} = [20 \quad 40]$. The set-point to the liquid level was moved from the initial value, equal to 0.40m, to the new set-point, equal to 0.35m. Analogously, the set-point for the drum pressure was moved from 40 kPa to 50 kPa. Figure 2 shows the responses of the system outputs of the system in closed-loop with the multi-model MPC. Observe that both controlled outputs reach the new set-points quite smoothly and without offset. However, these responses are rather slow and this conservative behavior can be associated with the two models that are considered by the controller. The transfer functions corresponding to the pair (y_2,u_1) are quite different in the two models and result in a cautious controller. Figure 3 shows the inputs corresponding to this simulated case. In the second simulation case a disturbance (d) corresponding to an increase of 20% in the feed flow rate is introduced in the system. Figure 4 shows the output responses that are driven back to the desired values smoothly and without offset. Figure 5 shows that the manipulated inputs are moved to a new steady-state to compensate the effect of the unmeasured disturbance. In both simulated cases, it is shown that the new controller performs consistently with what is expected from a real commercial controller and its implementation in a real system should be attempted.

Figures 6 and 7 show a comparison between the robust multi-model MPC and a linear MPC based on a single model (Model 1) with the same tuning parameters as the multi-model MPC for the same problem. From Figure 6, it can be observed that the performance of the multi-model MPC is significantly better than the performance of the linear MPC. The output responses of the proposed controller converge to the set-points faster than the responses of the linear MPC, which indicates that model uncertainty is affecting the performance of the conventional MPC and the implementation of the multi-model MPC may result in better economic results.

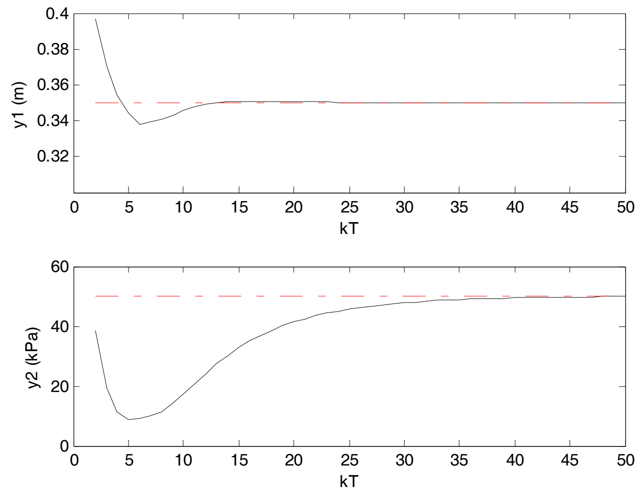

Figure 2: Controlled variables, output tracking case

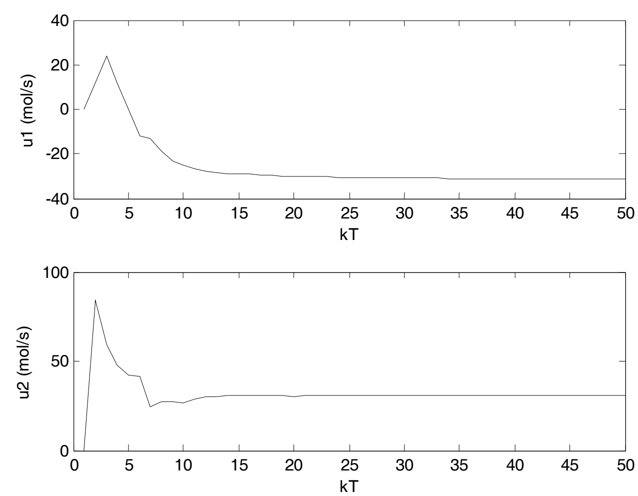

Figure 3: Manipulated variables, output tracking case

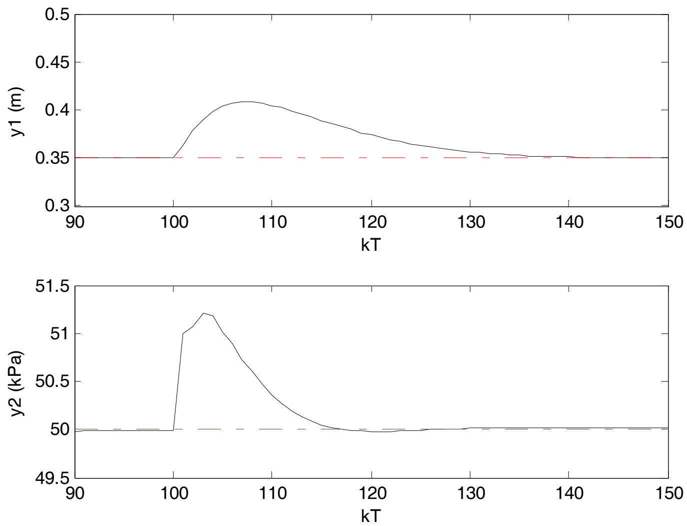

Figure 4: Controlled outputs and feed disturbance

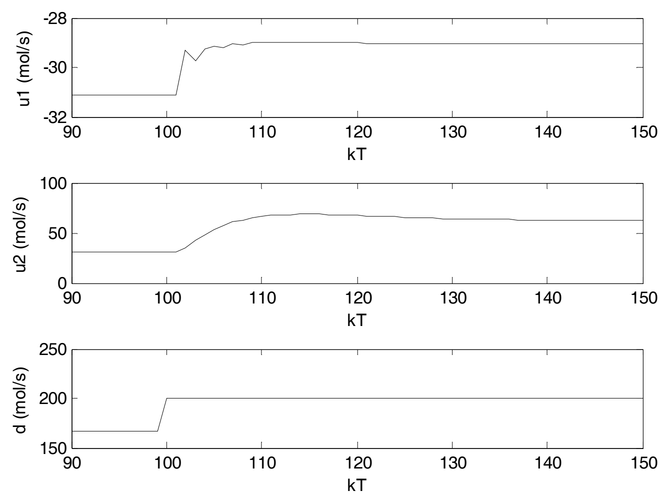

Figure 5: Input responses for the feed flow disturbance

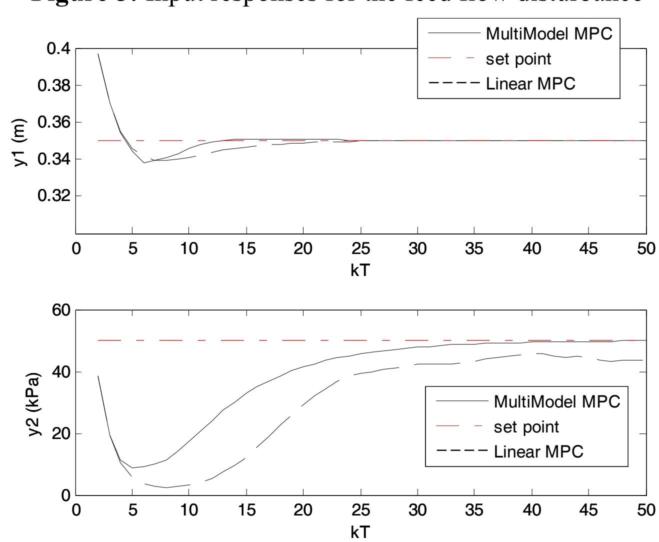

Figure 6: Controlled variables, output tracking case

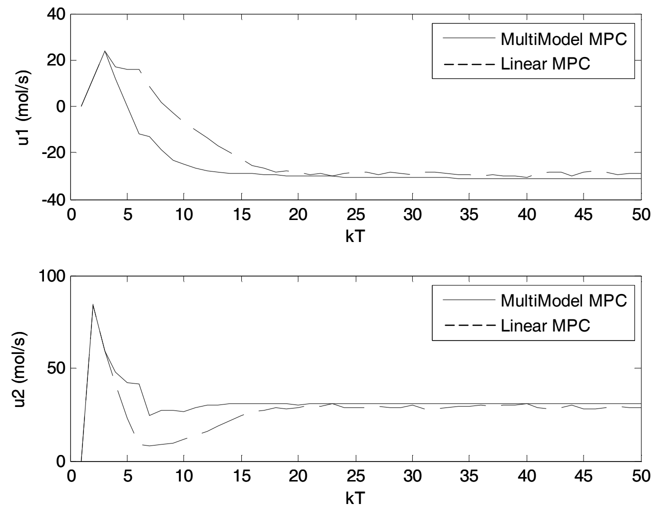

Figure 7: Manipulated variables, output tracking case

CONCLUSION

In this paper, a new version of the multi-model robust MPC with output feedback was developed. In the proposed method, assuming that the new set point is reachable, stability and convergence to the desired steady-state is assured even when the system model has uncertainties represented by a finite set of possible models. The controller was developed for the case of systems with stable and integrating outputs. The model uncertainty is assumed to be related to the stable outputs. The non-minimal state space model that is considered here defines the state as a sequence of past outputs and input moves. This eliminates the need for a state estimator, like the Kalman filter, which is usually included in commercial MPC packages. Consequently, the robust stability of other MPC controllers, which is only true when the state is measured, can be applied here and results in the first implementable robust MPC with output feedback. A simulation example shows that the implementation of the multi-model MPC can be achieved in practice at least for systems of small to medium dimension.

NOMENCLATURE

A_d	matrix of eigenvalues (Eq. 9)
A_i	model parameter matrix (Eq. 1)
A_y , $A_{\Delta u}$	state matrix components (Eq. 2)
A_y^i , $A_{\Delta u}^i$	state matrix components of the integrating system (Eq. 4)
A_d^i	matrix of eigenvalues of the integrating system (Eq. 10)
A_{d,θ_p}^s	matrix of eigenvalues of the stable system corresponding to θ_p (Eq. 13)
B_i	model parameter matrix (Eq. 1)
$B_{\Delta u}$	input matrix component (Eq. 2)
$B_{\Delta u}^i$	input matrix component of the integrating system (Eq. 4)
C_y , $C_{\Delta u}$	output matrix components (Eq. 3)
C_y^i , $C_{\Delta u}^i$	output matrix components of the integrating system (Eq. 5)
$C_y^s, C_{\Delta u}^s$	output matrix components of the stable system (Eq. 7)
$F_{\theta_p}^s$	component of the matrix of eigenvalues of the system θ_p (Eq. 15)
I_{nu}	identity matrix of dimension $nu \times nu$
I_{ny}	identity matrix of dimension $ny \times ny$
L	number of models considered in the robust MPC
na	order of the model defined in Eq. 1
nu	number of system inputs
ny	number of system outputs
ny_s	number of stable ouputs
Q_i, Q_s	weight matrices for the integrating and stable outputs (Eq. 16)
$\bar{Q}_s$	terminal weight of the stable output (Eq. 16)

R	weight matrix for the input move (Eq. 16)
$S_{\Delta}^{i}, S_{\Delta}^{s}, S_{i}^{i}$	weights for the slack variables (Eq. 16)
u	system input (Eq. 1)
V	matrix of eigenvectors of the state matrix A (Eq. 9)
V_i	matrix of eigenvectors of the integrating state matrix A_y^i
V_i^{Δ}, V_i^i	components of the matrix of eigenvectors of the integrating system (Eq. 5)
V_{s,θ_p}	matrix of eigenvectors of the stable system corresponding to θ_p
$V_{s,\theta_p}^{\Delta}, V_{s,\theta_p}^{s}$	components of the matrix of eigenvectors of system corresponding to θ_p
x_y	state vector component (Eq. 2)
$x_{\Delta u}$	state vector component (Eq. 2)
x_y^i	state component of the integrating system (Eq. 4)
$x_{\Delta u}^i$	state component of the integrating system (Eq. 4)
x_{y,θ_p}^s	state vector component of the stable model θ_p (Eq. 6)
$x_{\Delta u}^s$	state vector component of the stable models (Eq. 6)
y	system output (Eq. 1)
$y_{\theta_p}^s$	output of the stable system corresponding to model θ_p (Eq. 7)
y_i^{sp}	set point of the integrating output (Eq. 16)
y_s^{sp}	set point of the stable output (Eq. 16)
z^i	transformed state vector of the integrating system (Eq. 10)
z_i^{Δ}, z_i^i	components of the transformed state vector of the integrating system (Eq. 10)
$z_{s,\theta_p}^{\Delta}, z_{s,\theta_p}^{s}$	components of the transformed state of the stable model θ_p (Eq. 14)

Greek Symbols

θ_p	set of parameters that define the p^{th} model
$\delta_{\Delta,k}^{i}, \delta_{\Delta,k}^{s}, \delta_{i,k}^{i}$	slack variables for the integrating and stable outputs (Eq. (16))

REFERENCES

Badgwell, T. A., Robust model predictive control of linear systems. International Journal of Control, 68(4), 797-818 (1997).

Carrapiço, O. L. and Odloak, D., A stable model predictive control for integrating process. Computers and Chemical Engineering, 29(5), 1089-1099 (2005).

González, A. H., Marchetti, J. L. and Odloak, D., Extended robust model predictive control of integrating systems. AIChE Journal, 53(7), 1758-1769 (2007).

Gonzalez, A. H., Perez, J. M. and Odloak, D., Infinite horizon MPC with non-minimal state space feedback. Journal of Process Control, 19, 473-481 (2009).

Kothare, M. V., Balakrishnan, V. and Morari, M., Robust constrained model predictive control using linear matrix inequalities. Automatica, 32(10), 1361-1379 (1996).

Lee, J. H. and Cooley, B. L., Min-max predictive control techniques for linear system with bounded set of input matrices. Automatica, 36, 463-473 (2000).

Maciejowski, J. M., Predictive Control with Constraints. Prentice Hall, London (2002).

Odloak, D., Extended robust model predictive control. AIChE Journal, 50(8), 1824-1836 (2004).

Perez, J. M., Controle preditivo robusto com realimentação de saída. MSc. Dissertation, University of São Paulo, Brasil (2006). (In Portuguese).

Qin, S. J. and Badgwell, T. A., A survey of industrial model predictive control technology. Control Eng. Practice, 11(7), 733-764 (2003).

Rawlings, J. B. and Muske, K. R., Stability of constrained receding horizon. IEEE Transaction on Automatic Control, 38(10), 1512-1516 (1993).

Rodrigues, M. A. and Odloak, D., MPC for stable linear system model uncertainty. Automatica, 39(4), 569-583 (2003).

Rodrigues, M. A. and Odloak D., Robust MPC for systems with output feedback and input saturation. Journal of Process Control, 15(7), 837-846 (2005).

Soares, R. P. and Secchi, A. R., EMSO: A new environment for modeling, simulation and optimization. Computer Aided Chemical Engineering, 14, 947-952 (2003).

Wang, L. and Young, P. C., An improved structure for model predictive control using non-minimal state space realization. Journal of Process Control, 16(4), 355-371 (2005).