EXPERIMENTAL APPLICATION OF A NEURAL CONSTRAINED MODEL PREDICTIVE CONTROLLER BASED ON REFERENCE SYSTEM

A. G. MONTANDON[1], R. M. BORGES[2] and H. M. HENRIQUE[3,4]
School of Chemical Engineering/Federal University of Uberlandia. Av. João Naves de Ávila, 2121
Phone number: +55-34-3239-4292 Fax number: +55-34-3239-4188
POB 38408-100 Uberlandia - MG, Brazil
[4]Author to whom all correspondence should be addressed
E-mails: [1]amontandon@mic-br.com.br, [2]raquelmb@cefetes.br, [3]humberto@ufu.br

***Abstract* — The proposed constrained model predictive control (MPC) is based on a successive linearization of a neural model at each sampling time and the closed loop response is subject to a first order reference system as set of equality constraints. In addition the system inputs are subject to hard constraints. In order to satisfy both types of constraints simultaneously it was needed to include a slack vector in the equality constraints. This slack vector provides more flexibility in the control moves in order to render the solution of the optimization problem feasible. The proposed MPC was implemented in an experimental pH neutralization plant. Results showed a very satisfactory performance of the proposed strategy.**

***Keywords* — Model Based Control, Neural Control, Neural network Models, pH control, Real-Time Control Systems.**

I. INTRODUCTION

Global industrial competition has revealed the importance of the automatic control for industrial profitability and safety. In this fashion advanced control strategies are used to assure that processes can be operated safely in regions of the high product quality with low consumption of raw materials and energy. However the development of advanced control strategies is a very hard task mainly due to nonlinear behavior of the chemical processes.

Nonlinear processes have been controlled by linear controllers in spite of the fact that the vast majority of chemical processes is inherently nonlinear. The advantage of this approach is that an easy analytical solution of the control problem can be found and a low computational effort is demanded by linear controllers. However, the linear approach can be very limiting for highly nonlinear processes and it can drive the system to unstable solution. The use of nonlinear process models within the control strategy has been shown to provide the potential for significant improvement over linear controllers for nonlinear processes (Bequette, 1991; Henson and Seborg, 1997). Nonlinear model predictive control (NMPC) (Garcia and Morshedi, 1986; Garcia *et al.*, 1989; Gattu and Zafiriou, 1992) and input-output linearizing control (IOLC) are the most widely studied nonlinear control techniques for process control problems. NMPC offers many of the appealing features of linear model predictive control, including explicit compensation for input and output constraints (Meadows *et al.*, 1995). As compared to NMPC, IOLC offers several important advantages including transparent controller tuning and low computational requirements (Kravaris and Kantor, 1990). However, conventional feedback linearization techniques have neither constraint handling (Rawlings *et al.*, 1994) nor predictive capabilities. This has motivated the development of several modifications of the basic input-output linearization approach (Balchen and Sandrib, 1995; Kendi and Doyle, 1995). On the other hand, the nonlinear approach can result in a large computational effort that limits its use in practical applications.

The scope of this paper is to deal with nonlinear process by using a control technique which is computationally feasible for industrial and practical implementation. Feasible for industrial implementation means the controller must have low computational effort and, the most important, the solution of the optimization problem must be guaranteed. When we linearized the model we can transform the optimization problem into a quadratic programming problem and this type of optimization problem has convergence guaranteed in a finite number of iteration steps (low computational effort). If we use a nonlinear model directly in the optimization problem we can guarantee neither convergence nor feasibility of the solution and this is unacceptable for industrial and practical applications. This is one of prime problems in nonlinear MPC. In addition the computational effort is generally very large in a fully nonlinear approach and all MPC calculations must be done during a sampling time (10 seconds in the present case). In the present application we deal with process nonlinearities by using a successive linearization which showed to be effective in an experimental application.

The aim of this work is to present a nonlinear control technique which is computationally feasible for industrial implementation. The proposed strategy is a model predictive control technique (MPC) based on a successive linearization of the model via Taylor's series expansion at each sampling time following the Gattu and Zafiriou (1992) idea.

The cost function of the optimization problem is subject to a first order reference system and upper and lower limits in the inputs. In order to satisfy both

constraints simultaneously and to provide a feasible solution, it is necessary to include a slack variable (λ) in the cost function of the optimization problem. The prime advantage of the proposed algorithm is that it does not need be re-tuned for different operating points. An experimental study was carried out in a pH neutralization plant.

II. THE DYNAMIC SYSTEM

Consider a general forced nonlinear dynamic system described by:

$$\frac{d\hat{\mathbf{w}}(t)}{dt} = \mathbf{f}(\hat{\mathbf{w}}(t), \mathbf{m}(t)) \quad \text{and} \quad \hat{\mathbf{z}}(t) = \mathbf{h}(\hat{\mathbf{w}}(t)), \qquad (1)$$

where $\mathbf{m}(t) \in \Re^n$ is the vector of manipulated variables, $\hat{\mathbf{w}}(t) \in \Re^m$ is the estimated state vector and $\hat{\mathbf{z}}(t) \in \Re^n$ is the estimated output vector. It is well known that the use of Eq. (1) in a MPC strategy can result in a large computational effort and the convergence of the involved optimization problem can not be guaranteed in a finite number of iteration steps. These problems have limited the use of models based on Eq. (1) directly into MPC formulations for industrial applications.

The scope of this research is to deal with a MPC strategy that can be implemented industrially, *i.e.*, MPC with convergence guaranteed and low computational cost. The basic idea is similar to the Gattu and Zafiriou´s (1992) idea which uses a successively linearized model. The use of a linear model in constrained MPC calculations allows transforming the involved optimization problem into a quadratic programming problem (convergence guaranteed in a finite number of iteration steps). The system nonlinearity is considered by updating the linear model at each sampling instant. This update is computationally inexpensive. However we have included a reference system in MPC calculations to force the closed-loop output to be as linear as possible. The main advantage of this formulation is that re-tuning of the controller is no need when setpoint tracking is desirable. This result will be shown later. So, in order to get a linear model, Eq. (1) is linearized via Taylor's series expansion around a general point ($\mathbf{m}_0$, $\mathbf{w}_0$, $\mathbf{z}_0$) but a measured point. The following equation is obtained:

$$\begin{aligned} \frac{d\hat{\mathbf{w}}(t)}{dt} &= \mathbf{f}(\mathbf{w}_0, \mathbf{m}_0) + \mathbf{A}(\hat{\mathbf{w}}(t) - \mathbf{w}_0) + \mathbf{B}(\mathbf{m}(t) - \mathbf{m}_0), \\ \hat{\mathbf{z}}(t) &= \mathbf{h}(\mathbf{w}_0) + \mathbf{C}(\hat{\mathbf{w}}(t) - \mathbf{w}_0), \end{aligned} \qquad (2)$$

where $\mathbf{A} \in \Re^{m\times m}$, $\mathbf{B} \in \Re^{m\times n}$, $\mathbf{C} \in \Re^{n\times m}$, $\mathbf{f}(\mathbf{w}_0, \mathbf{m}_0)$: $\Re^{n+m} \rightarrow \Re^m$ and $\mathbf{h}(\mathbf{w}_0)$: $\Re^m \rightarrow \Re^n$ are given by:

$$\begin{aligned} &\mathbf{A} = \left(\partial \mathbf{f} / \partial \hat{\mathbf{w}}\right)\Big|_{\hat{\mathbf{w}}=\mathbf{w}_0, \mathbf{m}=\mathbf{m}_0}, \quad \mathbf{B} = \left(\partial \mathbf{f} / \partial \mathbf{m}\right)\Big|_{\hat{\mathbf{w}}=\mathbf{w}_0, \mathbf{m}=\mathbf{m}_0}, \\ &\mathbf{C} = \left(\partial \mathbf{h} / \partial \hat{\mathbf{w}}\right)\Big|_{\hat{\mathbf{w}}=\mathbf{w}_0}, \quad \mathbf{h}(\mathbf{w}_0) = \mathbf{h}(\hat{\mathbf{w}})\Big|_{\hat{\mathbf{w}}=\mathbf{w}_0} \\ &\mathbf{f}(\mathbf{w}_0, \mathbf{m}_0) = \mathbf{f}(\hat{\mathbf{w}}, \mathbf{m})\Big|_{\hat{\mathbf{w}}=\mathbf{w}_0, \mathbf{m}=\mathbf{m}_0}, \end{aligned} \qquad (3)$$

The maps $\mathbf{f}(.)$, $\mathbf{h}(.)$ and the general point ($\mathbf{m}_0$, $\mathbf{w}_0$, $\mathbf{z}_0$) must be known in order to evaluate the matrices **A**, **B** and **C**. In order to get accurate system predictions by using the linearized model, the point ($\mathbf{m}_0$, $\mathbf{w}_0$, $\mathbf{z}_0$) about which we form the Taylor expansion should be as close as possible to the point ($\mathbf{m}(t)$, $\hat{\mathbf{w}}(t)$, $\hat{\mathbf{z}}(t)$). Thus, in control fashion, the output vector $\hat{\mathbf{z}}(t)$ and state vector $\hat{\mathbf{w}}(t)$ are available (measured values) at current sampling time "t_k" ($\mathbf{z}_k$ and $\mathbf{w}_k$ respectively) but the input vector $\mathbf{m}_k$ will still be calculated by the MPC. The freshest actual input, state and output information available about the system is the information at previous sampling instant "t_{k-1}". Therefore, we will use the previous actual point ($\mathbf{m}(t_{k-1})$, $\mathbf{w}(t_{k-1})$, $\mathbf{z}(t_{k-1})$) to perform the linearization of the system:

$$\begin{aligned} \frac{d\hat{\mathbf{w}}(t)}{dt} &= \mathbf{f}_{k-1} + \mathbf{A}_{k-1}(\hat{\mathbf{w}}(t) - \mathbf{w}(t_{k-1})) + \\ &\quad + \mathbf{B}_{k-1}(\mathbf{m}(t) - \mathbf{m}(t_{k-1})) \\ \hat{\mathbf{z}}(t) &= \mathbf{h}_{k-1} + \mathbf{C}_{k-1}(\hat{\mathbf{w}}(t) - \mathbf{w}(t_{k-1})), \end{aligned} \qquad (4)$$

where $\mathbf{A}_{k-1} \in \Re^{m\times m}$, $\mathbf{B}_{k-1} \in \Re^{m\times n}$, $\mathbf{C}_{k-1} \in \Re^{n\times m}$, $\mathbf{f}_{k-1}$: $\Re^{n+m} \rightarrow \Re^m$ and $\mathbf{h}_{k-1}$: $\Re^m \rightarrow \Re^n$ are given by:

$$\begin{aligned} \mathbf{A}_{k-1} &= \left(\partial \mathbf{f} / \partial \hat{\mathbf{w}}\right)\Big|_{\hat{\mathbf{w}}=\mathbf{w}(t_{k-1}), \mathbf{m}=\mathbf{m}(t_{k-1})}, \\ \mathbf{B}_{k-1} &= \left(\partial \mathbf{f} / \partial \mathbf{m}\right)\Big|_{\hat{\mathbf{w}}=\mathbf{w}(t_{k-1}), \mathbf{m}=\mathbf{m}(t_{k-1})}, \\ \mathbf{C}_{k-1} &= \left(\partial \mathbf{h} / \partial \hat{\mathbf{w}}\right)\Big|_{\hat{\mathbf{w}}=\mathbf{w}(t_{k-1})}, \\ \mathbf{f}_{k-1} &= \mathbf{f}(\hat{\mathbf{w}}, \mathbf{m})\Big|_{\hat{\mathbf{w}}=\mathbf{w}(t_{k-1}), \mathbf{m}=\mathbf{m}(t_{k-1})}, \\ \mathbf{h}_{k-1} &= \mathbf{h}(\hat{\mathbf{w}})\Big|_{\hat{\mathbf{w}}=\mathbf{w}(t_{k-1})}. \end{aligned} \qquad (5)$$

Now, we define the following deviation variable:

$$\begin{aligned} \hat{\mathbf{x}}(t) &= \hat{\mathbf{w}}(t) - \mathbf{w}(t_{k-1}), \\ \hat{\mathbf{y}}(t) &= \hat{\mathbf{z}}(t) - \mathbf{z}(t_{k-1}), \\ \mathbf{u}(t) &= \mathbf{m}(t) - \mathbf{m}(t_{k-1}). \end{aligned} \qquad (6)$$

Substituting Eq. (6) into Eq. (4) and taking the constant matrices $\mathbf{A} = \mathbf{A}_{k-1}$, $\mathbf{B} = \mathbf{B}_{k-1}$ and $\mathbf{C} = \mathbf{C}_{k-1}$ and constant vectors $\mathbf{f} = \mathbf{f}_{k-1}$ and $\mathbf{h} = \mathbf{h}_{k-1}$ in order to make the nomenclature cleaner, Eq. (4) can be rewritten as follows:

$$\begin{aligned} \frac{d\hat{\mathbf{x}}(t)}{dt} &= \mathbf{f} + \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t), \\ \hat{\mathbf{y}}(t) &= \mathbf{C}\hat{\mathbf{x}}(t). \end{aligned} \qquad (7)$$

It is important to keep in mind that $\mathbf{u}(t)$, $\hat{\mathbf{x}}(t)$ and $\hat{\mathbf{y}}(t)$ are deviation variable in relation to their values at previous sampling instant and that vector **f** and matrices **A**, **B** and **C** are kept constant during all MPC calculation (prediction phase) but they must be updated after all MPC calculations was performed and a new sampling time is going to be implemented in a receding horizon fashion. Now, considering matrix **A** nonsingular, Eq. (7) can be integrated from "t" to "t+Δt" or, in the other words, from "t_k" to "t_{k+1}" assuming $\mathbf{u}(k)$ constant during the sampling instant. The result is as follows:

$$\begin{aligned}\hat{\mathbf{x}}(k+1) &= \boldsymbol{\Phi}\hat{\mathbf{x}}(k) + \boldsymbol{\Psi}\mathbf{B}\mathbf{u}(k) + \boldsymbol{\Psi}\mathbf{f},\\ \hat{\mathbf{y}}(k+1) &= \mathbf{C}\hat{\mathbf{x}}(k+1),\end{aligned} \tag{8}$$

where the matrices $\boldsymbol{\Phi} \in \Re^{m\times m}$, $\boldsymbol{\Psi} \in \Re^{m\times m}$ are given as follows:

$$\begin{aligned}\boldsymbol{\Phi} &= e^{\mathbf{A}\Delta t},\\ \boldsymbol{\Psi} &= \mathbf{A}^{-1}\left(e^{\mathbf{A}\Delta t} - \mathbf{I}\right).\end{aligned} \tag{9}$$

Eq (7) can also be integrated analytically from "t_{k-1}" to "t_k" assuming $\mathbf{u}$(k-1) constant during the sampling instant and taking $\hat{\mathbf{x}}(k-1) = \mathbf{x}(k-1)$. The result is as follows:

$$\begin{aligned}\hat{\mathbf{x}}(k) &= \boldsymbol{\Phi}\mathbf{x}(k-1) + \boldsymbol{\Psi}\mathbf{B}\mathbf{u}(k-1) + \boldsymbol{\Psi}\mathbf{f},\\ \hat{\mathbf{y}}(k) &= \mathbf{C}\hat{\mathbf{x}}(k).\end{aligned} \tag{10}$$

Now, subtracting Eq. (8) from Eq. (10), taking the $\Delta\hat{\mathbf{x}}(k)$ vector as $\Delta\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k)$-$\mathbf{x}$(k-1) and the $\Delta\mathbf{u}(k)$ vector as $\Delta\mathbf{u}(k) = \mathbf{u}(k)$-$\mathbf{u}$(k-1), we can rearrange the resulting equation to obtain:

$$\hat{\mathbf{y}}(k+1) = \hat{\mathbf{y}}(k) + \mathbf{C}\boldsymbol{\Phi}\Delta\hat{\mathbf{x}}(k) + \mathbf{C}\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k). \tag{11}$$

If this procedure is continuously carried out up to instant k = P where P is the prediction horizon, M is the control horizon with P ≥ M and $\Delta\mathbf{u}(k+j) = \mathbf{0}$ to M < j ≤ P, the following set of equations is obtained:

$$\begin{aligned}\hat{\mathbf{y}}(k+2) = \hat{\mathbf{y}}(k) &+ \mathbf{C}\left(\boldsymbol{\Phi}^2 + \boldsymbol{\Phi}\right)\Delta\hat{\mathbf{x}}(k) +\\ &+ \mathbf{C}\left(\boldsymbol{\Phi} + \mathbf{I}\right)\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k) + \mathbf{C}\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+1),\end{aligned} \tag{12}$$

$$\begin{aligned}\hat{\mathbf{y}}(k+3) = \hat{\mathbf{y}}(k) &+ \mathbf{C}(\boldsymbol{\Phi}^3 + \boldsymbol{\Phi}^2 + \boldsymbol{\Phi})\Delta\hat{\mathbf{x}}(k) +\\ &+ \mathbf{C}(\boldsymbol{\Phi}^2 + \boldsymbol{\Phi} + \mathbf{I})\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k) +\\ &+ \mathbf{C}(\boldsymbol{\Phi} + \mathbf{I})\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+1) + \mathbf{C}\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+2),\\ &\vdots\end{aligned} \tag{13}$$

$$\begin{aligned}\hat{\mathbf{y}}(k+M+1) = \hat{\mathbf{y}}(k) &+ \mathbf{C}\left(\sum_{i=1}^{M+1}\boldsymbol{\Phi}^i\right)\Delta\hat{\mathbf{x}}(k) +\\ &+ \mathbf{C}\left(\sum_{i=1}^{M+1}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k) +\\ &+ \mathbf{C}\left(\sum_{i=1}^{M}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+1) +\\ &+ \mathbf{C}\left(\sum_{i=1}^{M-1}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+2) +\\ &+ \cdots + \mathbf{C}\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+M),\end{aligned} \tag{14}$$

$$\begin{aligned}\hat{\mathbf{y}}(k+M+2) = \hat{\mathbf{y}}(k) &+ \mathbf{C}\left(\sum_{i=1}^{M+2}\boldsymbol{\Phi}^i\right)\Delta\hat{\mathbf{x}}(k) +\\ &+ \mathbf{C}\left(\sum_{i=1}^{M+2}\boldsymbol{\Phi}^{i-1}\right)\mathbf{B}\Delta\mathbf{u}(k) +\\ &+ \mathbf{C}\left(\sum_{i=1}^{M+1}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+1) +\\ &+ \cdots + \mathbf{C}\boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+M),\\ &\vdots\end{aligned} \tag{15}$$

$$\begin{aligned}\hat{\mathbf{y}}(k+P) = \hat{\mathbf{y}}(k) &+ \mathbf{C}\left(\sum_{i=1}^{P}\boldsymbol{\Phi}^i\right)\Delta\hat{\mathbf{x}}(k) +\\ &+ \mathbf{C}\left(\sum_{i=1}^{P}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k) +\\ &+ \mathbf{C}\left(\sum_{i=1}^{P-1}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+1) +\\ &+ \cdots + \mathbf{C}\left(\sum_{i=1}^{P-M}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+M).\end{aligned} \tag{16}$$

It is important to keep in mind that the prediction instant from j > M to j = P $\Delta\mathbf{u}(k+j) = \mathbf{0}$ because $\mathbf{u}(k+j)$ remains constant. The Eqs. (11)-(16) can be put in a matrix form:

$$\hat{\mathbf{y}} = \boldsymbol{\Gamma}\cdot\Delta\mathbf{u} + \boldsymbol{\gamma}, \tag{17}$$

where:

$$\hat{\mathbf{y}} = \left[\hat{\mathbf{y}}^T(k+1) \quad \cdots \quad \hat{\mathbf{y}}^T(k+M) \quad \cdots \quad \hat{\mathbf{y}}^T(k+P)\right]^T, \tag{18}$$

$$\Delta\mathbf{u} = \left[\Delta\mathbf{u}^T(k) \quad \Delta\mathbf{u}^T(k+1) \quad \cdots \quad \Delta\mathbf{u}^T(k+M)\right]^T, \tag{19}$$

$$\boldsymbol{\gamma} = \begin{bmatrix} \hat{\mathbf{y}}(k) + \mathbf{C}\boldsymbol{\Phi}\Delta\hat{\mathbf{x}}(k)\\ \hat{\mathbf{y}}(k) + \mathbf{C}\left(\sum_{i=1}^{2}\boldsymbol{\Phi}^i\right)\Delta\hat{\mathbf{x}}(k)\\ \vdots\\ \hat{\mathbf{y}}(k) + \mathbf{C}\left(\sum_{i=1}^{M}\boldsymbol{\Phi}^i\right)\Delta\hat{\mathbf{x}}(k)\\ \vdots\\ \hat{\mathbf{y}}(k) + \mathbf{C}\left(\sum_{i=1}^{P}\boldsymbol{\Phi}^i\right)\Delta\hat{\mathbf{x}}(k)\end{bmatrix} \tag{20}$$

$$\boldsymbol{\Gamma} = \begin{bmatrix} \mathbf{C}\boldsymbol{\Psi}\mathbf{B} & [\mathbf{0}]_{n\times n} & \cdots & [\mathbf{0}]_{n\times n}\\ \mathbf{C}(\boldsymbol{\Phi}+\mathbf{I})\boldsymbol{\Psi}\mathbf{B} & \mathbf{C}\boldsymbol{\Psi}\mathbf{B} & \cdots & [\mathbf{0}]_{n\times n}\\ \vdots & \vdots & & \vdots\\ \mathbf{C}\left(\sum_{i=1}^{M}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B} & \mathbf{C}\left(\sum_{i=1}^{M-1}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B} & \cdots & \mathbf{C}\boldsymbol{\Psi}\mathbf{B}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{C}\left(\sum_{i=1}^{P}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B} & \mathbf{C}\left(\sum_{i=1}^{P-1}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B} & \cdots & \mathbf{C}\left(\sum_{i=1}^{P-M}\boldsymbol{\Phi}^{i-1}\right)\boldsymbol{\Psi}\mathbf{B}\end{bmatrix}, \tag{21}$$

A. The Reference System

The controller is designed to transform the closed loop output into a first order system as follows:

$$\frac{d\hat{\mathbf{y}}(t)}{dt} = \mathbf{K}\left(\mathbf{y}^{SP}(t) - \hat{\mathbf{y}}(t)\right), \tag{22}$$

where $\mathbf{K} \in \Re^{n\times n}$ is a tuning parameter, $\mathbf{y}^{SP}(t) \in \Re^{n}$ is the setpoint vector and $\hat{\mathbf{y}}(t) \in \Re^{n}$ is the estimated system output vector. However, infeasible solutions of the MPC optimization problem can occur when equality constraints represented by Eq.(22) and hard constraints (upper and lower bounds for inputs) must be satisfied simultaneously. To overcome this problem, a slack variable ($\boldsymbol{\lambda}$) is introduced into Eq.(22) in order to allow the system to deviate from reference system and to satisfy the hard constraints. This approach is based on the generic model control approach by Lee and Sullivan (1988). The use of a slack variable to handle infeasibility problems in MPC has been used in several publications (Lee, 1993; Lopes, 2000; Kalra *et. al.*, 2002). Therefore, introducing slack variable $\boldsymbol{\lambda}(t) \in \Re^{n}$ into Eq. (22) and substituting Eq. (7) into Eq. (22) we obtain:

$$\mathbf{C}\,\mathbf{f} + \mathbf{C}\mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{C}\mathbf{B}\mathbf{u}(t) + \boldsymbol{\lambda}(t) = \mathbf{K}\left(\mathbf{y}^{SP}(t) - \hat{\mathbf{y}}(t)\right). \tag{23}$$

Equation (23) can be discretized at the current instant $t = t_k$. In addition, $\Delta\mathbf{u}(k) = \mathbf{u}(k)$ and $\Delta\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}(t)$ because the system variables in Eq. (23) are written in deviation variable. Thus, rearranging Eq. (23) we obtain:

$$\begin{aligned}\mathbf{C}\mathbf{B}\Delta\mathbf{u}(k) + \boldsymbol{\lambda}(k) = \mathbf{K}(\mathbf{y}^{SP}(k) - \hat{\mathbf{y}}(k)) \\ - \mathbf{C}\mathbf{A}\Delta\hat{\mathbf{x}}(k) - \mathbf{C}\,\mathbf{f}\,.\end{aligned} \tag{24}$$

Equation (23) is now evaluated at instant $t = t_{k+1}$ and the resulting equation is subtracted from Eq. (23). After an algebraic manipulation we obtain:

$$\begin{aligned}(\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k) + \mathbf{C}\mathbf{B}\Delta\mathbf{u}(k+1) + \boldsymbol{\lambda}(k+1) - \\ - \boldsymbol{\lambda}(k) = \mathbf{K}\Delta\mathbf{y}^{SP}(k+1) - (\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}\Delta\hat{\mathbf{x}}(k)\,.\end{aligned} \tag{25}$$

If this procedure is continuously carried out up to instant k = P where P is the prediction horizon, M is the control horizon with $P \geq M$ and $\Delta\mathbf{u}(k+j) = \mathbf{0}$ to $M < j \leq P$, the following set of equations is obtained:

$$\begin{aligned}&(\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{B}\,\Delta\mathbf{u}(k) + (\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Psi}\mathbf{B}\,\Delta\mathbf{u}(k+1) \\ &\quad + \mathbf{C}\mathbf{B}\,\Delta\mathbf{u}(k+2) + \boldsymbol{\lambda}(k+2) - \boldsymbol{\lambda}(k+1) = \\ &\quad + \mathbf{K}\,\Delta\mathbf{y}^{SP}(k+2) - (\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}^{2}\Delta\hat{\mathbf{x}}(k)\,,\end{aligned} \tag{26}$$

$$\vdots$$

$$\begin{aligned}&(\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}^{M-1}\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k) + \cdots + \\ &\quad + (\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+M-1) + \\ &+ \mathbf{C}\mathbf{B}\Delta\mathbf{u}(k+M) + \boldsymbol{\lambda}(k+M) - \boldsymbol{\lambda}(k+M-1) = \\ &\quad \mathbf{K}\Delta\mathbf{y}^{SP}(k+M) - (\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}^{M}\Delta\hat{\mathbf{x}}(k)\,,\end{aligned} \tag{27}$$

$$\begin{aligned}&(\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}^{M}\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k) + \cdots + \\ &+ (\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+M) + \boldsymbol{\lambda}(k+M+1) - \\ &\quad - \boldsymbol{\lambda}(k+M) = \mathbf{K}\Delta\mathbf{y}^{SP}(k+M+1) - \\ &\quad - (\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}^{M+1}\Delta\hat{\mathbf{x}}(k)\,,\end{aligned} \tag{28}$$

$$\begin{aligned}&(\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}^{M+1}\boldsymbol{\Psi}\mathbf{B}\,\Delta\mathbf{u}(k) + \cdots + \\ &+ (\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k+M) + \boldsymbol{\lambda}(k+M+2) - \\ &\quad - \boldsymbol{\lambda}(k+M+1) = \mathbf{K}\Delta\mathbf{y}^{SP}(k+M+2) - \\ &\quad - (\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}^{M+2}\Delta\hat{\mathbf{x}}(k)\,,\end{aligned} \tag{29}$$

$$\vdots$$

$$\begin{aligned}&(\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}^{P-1}\boldsymbol{\Psi}\mathbf{B}\Delta\mathbf{u}(k) + \cdots + \\ &+ (\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}^{P-M}\boldsymbol{\Psi}\mathbf{B}\,\Delta\mathbf{u}(k+M) + \boldsymbol{\lambda}(k+P) - \\ &\quad - \boldsymbol{\lambda}(k+P-1) = \mathbf{K}\Delta\mathbf{y}^{SP}(k+P) - \\ &\quad - (\mathbf{C}\mathbf{A} + \mathbf{K}\mathbf{C})\boldsymbol{\Phi}^{P}\Delta\hat{\mathbf{x}}(k)\,.\end{aligned} \tag{30}$$

The Eqs. (24)-(30) can be put in a matrix form:

$$\mathbf{D}\;\mathbf{z} = \mathbf{b}\,, \tag{31}$$

where:

$$\mathbf{D} = \begin{bmatrix}
\mathbf{C}\mathbf{B} & [\mathbf{0}] & \cdots & [\mathbf{0}] & [\mathbf{I}] & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] & \cdots & [\mathbf{0}] \\
(\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Psi}\mathbf{B} & \mathbf{C}\mathbf{B} & \cdots & [\mathbf{0}] & [-\mathbf{I}] & [\mathbf{I}] & [\mathbf{0}] & [\mathbf{0}] & \cdots & [\mathbf{0}] \\
(\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{B} & (\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Psi}\mathbf{B} & \cdots & [\mathbf{0}] & [\mathbf{0}] & [-\mathbf{I}] & [\mathbf{I}] & [\mathbf{0}] & \cdots & [\mathbf{0}] \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
(\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Phi}^{M-1}\boldsymbol{\Psi}\mathbf{B} & (\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Phi}^{M-2}\boldsymbol{\Psi}\mathbf{B} & \cdots & \mathbf{C}\mathbf{B} & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] & \cdots & [\mathbf{I}] \\
(\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Phi}^{M}\boldsymbol{\Psi}\mathbf{B} & (\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Phi}^{M-1}\boldsymbol{\Psi}\mathbf{B} & \cdots & (\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Psi}\mathbf{B} & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] & \cdots & [\mathbf{I}] \\
(\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Phi}^{M+1}\boldsymbol{\Psi}\mathbf{B} & (\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Phi}^{M}\boldsymbol{\Psi}\mathbf{B} & \cdots & (\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Phi}\boldsymbol{\Psi}\mathbf{B} & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] & \cdots & [\mathbf{I}] \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
(\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Phi}^{P-1}\boldsymbol{\Psi}\mathbf{B} & (\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Phi}^{P-2}\boldsymbol{\Psi}\mathbf{B} & \cdots & (\mathbf{C}\mathbf{A}+\mathbf{K}\mathbf{C})\,\boldsymbol{\Phi}^{P-M}\boldsymbol{\Psi}\mathbf{B} & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] & \cdots & [\mathbf{I}]
\end{bmatrix}, \tag{32}$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{K}(\mathbf{y}^{SP}(k) - \hat{y}(k)) - \mathbf{CA}\Delta\hat{\mathbf{x}}(k) - \mathbf{Cf} \\ \mathbf{K}\Delta\mathbf{y}^{SP}(k+1) - (\mathbf{CA}+\mathbf{KC})\boldsymbol{\Phi}\Delta\hat{\mathbf{x}}(k) \\ \mathbf{K}\Delta\mathbf{y}^{SP}(k+2) - (\mathbf{CA}+\mathbf{KC})\boldsymbol{\Phi}^{2}\Delta\hat{\mathbf{x}}(k) \\ \vdots \\ \mathbf{K}\Delta\mathbf{y}^{SP}(k+M) - (\mathbf{CA}+\mathbf{KC})\boldsymbol{\Phi}^{M}\Delta\hat{\mathbf{x}}(k) \\ \mathbf{K}\Delta\mathbf{y}^{SP}(k+M+1) - (\mathbf{CA}+\mathbf{KC})\boldsymbol{\Phi}^{M+1}\Delta\hat{\mathbf{x}}(k) \\ \mathbf{K}\Delta\mathbf{y}^{SP}(k+M+2) - (\mathbf{CA}+\mathbf{KC})\boldsymbol{\Phi}^{M+2}\Delta\hat{\mathbf{x}}(k) \\ \vdots \\ \mathbf{K}\Delta\mathbf{y}^{SP}(k+P) - (\mathbf{CA}+\mathbf{KC})\boldsymbol{\Phi}^{P}\Delta\hat{\mathbf{x}}(k) \end{bmatrix}, \quad (33)$$

$$\mathbf{z} = \left[\Delta\mathbf{u}^{T}(k) \quad \cdots \quad \Delta\mathbf{u}^{T}(k+M) \quad \boldsymbol{\lambda}^{T}(k) \quad \cdots \quad \boldsymbol{\lambda}^{T}(k+P)\right]^{T}. \quad (34)$$

B The proposed MPC Design

The aim of the proposed controller is to do the system output to follow a first order linear system. This aim would be achieved if the slack variable $\boldsymbol{\lambda}(k)$ was zero for all time instant ($\boldsymbol{\lambda}(k) = \mathbf{0}$ for $\forall\ k \in [1, P]$). This requirement would yield a more aggressive and less robust controller. In addition, infeasible solutions of the MPC optimization problem can also occur when hard and reference system constraints are both present in the optimization problem. To avoid infeasible solutions and to improve the robustness of the proposed controller, it is allowed $\boldsymbol{\lambda}(k)$ to being different to zero. However this is done in an optimal fashion, i.e., minimizing the quadratic norm of the vector $\boldsymbol{\lambda}(k)$. It is also well know from MPC literature that the penalization of the quadratic norm of the control effort vector ($\Delta\mathbf{u}(k)$) improves the MPC stability. Thus, the proposed cost function is a dual cost function defined as:

$$\min_{\substack{\Delta\mathbf{u}(k),\ldots,\Delta\mathbf{u}(k+M) \\ \boldsymbol{\lambda}(k),\ldots,\boldsymbol{\lambda}(k+P)}} \mathbf{J} = \frac{1}{2}\left(\Delta\mathbf{u}^{T}\mathbf{R}\,\Delta\mathbf{u} + \boldsymbol{\lambda}^{T}\mathbf{S}\boldsymbol{\lambda}\right),$$

subject to :

$$\mathbf{D}\,\mathbf{z} = \mathbf{b}$$

$$\mathbf{u}_{min}(k+j) \le \mathbf{u}(k+j) \le \mathbf{u}_{max}(k+j)$$

$$-\left|\Delta\mathbf{u}_{max}(k+j)\right| \le \Delta\mathbf{u}(k+j) \le \left|\Delta\mathbf{u}_{max}(k+j)\right| \quad (35)$$

$$j = 0,\cdots,M$$

where the matrices $\mathbf{R}$ and $\mathbf{S}$ are positive definite matrices, the lambda vector $\boldsymbol{\lambda}$ is given by $\boldsymbol{\lambda} = [\boldsymbol{\lambda}^{T}(k) \ldots \boldsymbol{\lambda}^{T}(k+P)]^{T}$ and the control vector $\Delta\mathbf{u}$ is given by $\Delta\mathbf{u} = [\Delta\mathbf{u}^{T}(k) \ldots \Delta\mathbf{u}^{T}(k+M)]^{T}$. It is very easy to show that this cost function can also be put as a cost function of a quadratic programming problem:

$$\min_{\mathbf{z}(k),\ldots,\mathbf{z}(k+M+P+2)} \mathbf{J} = \frac{1}{2}\mathbf{z}^{T}\mathbf{H}\,\mathbf{z}, \quad (36)$$

subject to :

$$\mathbf{D}\,\mathbf{z} = \mathbf{b}$$

$$\mathbf{u}_{min}(k+j) \le \mathbf{u}(k+j) \le \mathbf{u}_{max}(k+j)$$

$$-\left|\Delta\mathbf{u}_{max}(k+j)\right| \le \Delta\mathbf{u}(k+j) \le \left|\Delta\mathbf{u}_{max}(k+j)\right|$$

$$j = 0,\cdots,M$$

where $\mathbf{z}$ is given by Eq.(34) and matrix $\mathbf{H} \in \Re^{(P+M+2)\times(P+M+2)}$ is given by:

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & [\mathbf{0}] \\ [\mathbf{0}] & \mathbf{S} \end{bmatrix}. \quad (37)$$

Borges (2001) showed the proposed controller as presented here can not eliminate offset unless that the following change in Eqs. (21) and (33) is made:

$$\hat{\mathbf{y}}(k) = \mathbf{y}(k), \quad (38)$$

$$\Delta\hat{\mathbf{x}}(k) = \Delta\mathbf{x}(k), \quad (39)$$

$$\mathbf{Cf}_{k-1} \approx \frac{\mathbf{y}_k - \mathbf{y}_{k-1}}{\Delta t}, \quad (40)$$

where $\mathbf{y}_k$ and $\mathbf{y}_{k-1}$ are the system outputs at k^{th} and $(k-1)^{th}$ sampling instants respectively. Remember the actual system state and system output ($\mathbf{x}(k)$ and $\mathbf{y}(k)$, respectively) are available at time instant "t_k". The basic structure of this MPC can be summarized by Fig. (1).

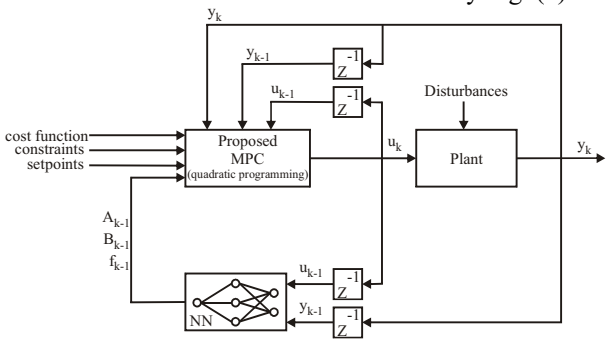

Figure 1: Basic structure of the proposed MPC.

C. The Classical MPC design

A MPC is designed in order to compare results with the proposed controller. A classical dual cost function is used where the quadratic norms of setpoint deviation vector and control effort vector are penalized:

$$\min_{\Delta\mathbf{u}(k),\ldots,\Delta\mathbf{u}(k+M)} J = \frac{1}{2}\left(\mathbf{e}^{T}\mathbf{Q}\mathbf{e} + \Delta\mathbf{u}^{T}\mathbf{R}\Delta\mathbf{u}\right), \quad (41)$$

Subject to :

$$\mathbf{u}_{min}(k+j) \le \mathbf{u}(k+j) \le \mathbf{u}_{max}(k+j)$$

$$-\left|\Delta\mathbf{u}_{max}(k+j)\right| \le \Delta\mathbf{u}(k+j) \le \left|\Delta\mathbf{u}_{max}(k+j)\right|,$$

$$j = 0,\cdots,M$$

where the matrices $\mathbf{Q}$ and $\mathbf{R}$ are positive definite matrices. In this formulation the predictions are calculated by using Eq. (17)-(21) modified by Eqs. (38) and (39). The vector $\mathbf{e}$ is the setpoint deviation vector and it is given by:

$$\mathbf{e} = -\boldsymbol{\Gamma}\Delta\mathbf{u} + \mathbf{e}', \tag{42}$$

$$\mathbf{e}' = (\mathbf{y}^{SP} - \boldsymbol{\gamma}), \tag{43}$$

$$\mathbf{y}^{SP} = \left[\mathbf{y}^{SP}(k+1)^T \quad \cdots \quad \mathbf{y}^{SP}(k+P)^T\right]^T. \tag{44}$$

In this case the optimization problem represented by Eq. (41)-(44) can also be put as a quadratic programming problem. If the modifications represented by Eq. (38)-(39) are also included in the MPC formulation represented by Eq. (17)-(21) it is very easy to show that this MPC will also eliminate offset. In this case the nonlinearities are considered by updating the matrices **A**, **B** and **C** next sampling instant after all MPC calculations are performed. Nevertheless, the constraints used by this MPC are different from the proposed MPC because they do not include the reference system constraints. The basic structure of this MPC can also be summarized by Fig. (1). But in this case, the cost function and constraints are different from the proposed MPC.

III. EXPERIMENTAL SETUP

The pH system was chosen because it is used as benchmark in control applications mainly due to its very strong nonlinear behavior. Figure 2 presents schematically the system. It consists of a continuous stirred tank reactor fed by a base flowrate (NaOH) and an acid flowrate (HNO_3). Both flowrates are measured by using infrared turbine meters and can be manipulated by using magnetic pump with external control. The system pH is measured by using a pH probe in the reactor output and sent to a data acquisition system. All input and output signals were manipulated in a low cost hardware (Pentium III 500 MHz IBM/PC computer) by using LabVIEW® software (version 6.1).

IV. SYSTEM MODELING

A. The physical model

Consider a neutralization process that occurs in the CSTR shown in Fig.2. The system has an output (pH) and two inputs (Q_1 and Q_2, acid stream and base stream respectively). The liquid level is kept constant and the chemical reactions involved are:

$$\begin{aligned} &H_2O \Leftrightarrow OH^- + H^+, \\ &HNO_3 \Leftrightarrow H^+ + NO_3^-, \\ &NaOH \Leftrightarrow Na^+ + OH^-. \end{aligned} \tag{45}$$

Following the approach of Gustafsson and Waller (1983), a reaction invariant can be defined for each stream as follows:

$$W_a = [H^+] - [OH^-]. \tag{46}$$

The quantity W_a is called reaction invariant because it is not affected by the extent of the reactions. It is assumed that the reactions are fast enough and the system can be considered in equilibrium. Then the equilibrium relations can be used to determine the hydrogen ion concentration from the reaction invariant. The water equilibrium constant is given by

$$K_W = [H^+][OH^-]. \tag{47}$$

Equations (46) and (47) can be combined and an implicit algebraic relation between $[H^+]$ and W_a can be derived:

$$W_a = [H^+] - \frac{K_W}{[H^+]}. \tag{48}$$

Now, the dynamic of the process is given by mass balance for the invariant W_a (Montandon, 2005):

$$V\frac{d(W_a)}{dt} = Q_1(W_{a1} - W_a) + Q_2(W_{a2} - W_a), \tag{49}$$

$$W_a(0) = \overline{W}_a. \tag{50}$$

Equation Eq.(50) represents initial steady state condition for the reactor and is given by:

$$\overline{W}_a = \frac{(Q_1 W_{a1} + Q_2 W_{a2})}{(Q_1 + Q_2)}. \tag{51}$$

$W_a(t)$ can be obtained by integrating of Eq.(49) subject to the initial condition represented by Eq. (50), $[H^+]$ is obtained by substituting $W_a(t)$ into Eq.(48) and pH(t) is obtained by substituting $[H^+]$ into Eq. (52):

$$pH = -\log[H^+]. \tag{52}$$

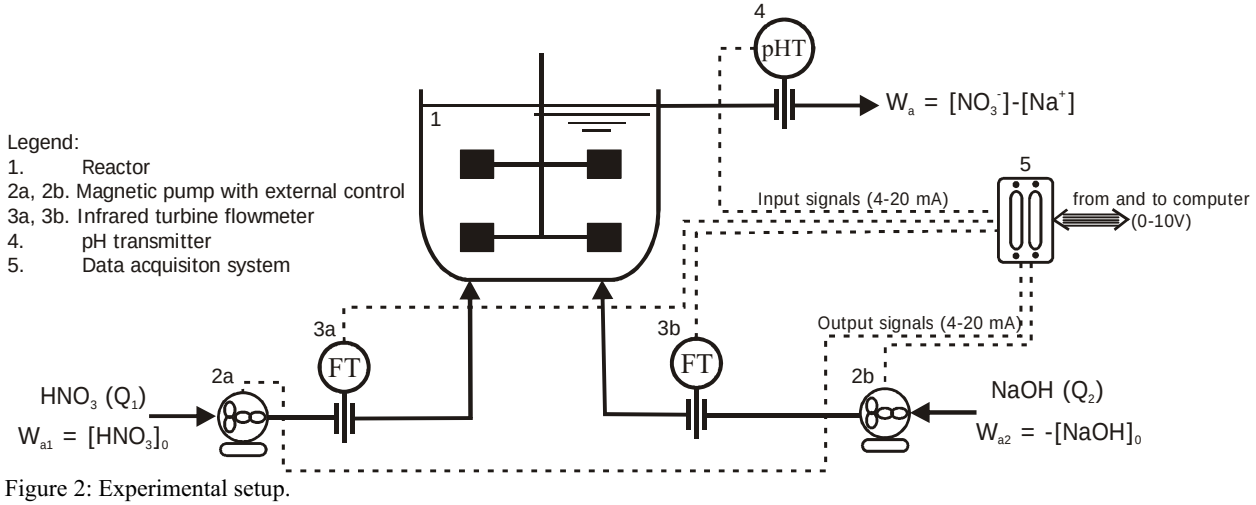

Figure 2: Experimental setup.

Table 1: Nominal values of the system parameters.

Variable	Symbol	Nominal values
Volume	V	4459.94 cm3
Acid flowrate	Q_1	12.0 mL/s
Base flowrate	Q_2	12.0 mL/s
pH	pH	7.0
Acid conc. in Q_1	$[HNO3]_0$	3.611e-03 M
Base conc. in Q_2	$[NaOH]_0$	3.611e-03 M
W_a in Q_1	W_{a1}	3.611e-03 M
W_a in Q_2	W_{a2}	-3.611e-03 M
W_a in output	W_a	0.0
Water equil. const.	K_W	10^{-14}

Table 1 gives the nominal values of system parameters. This system is interesting from control point of view because it is strongly nonlinear (see Fig. 3). This figure also reveals that the predicted titration curve obtained from the physical model was stepper over the operating range than the experimental titration curve. This result is due to water source used to prepare the solutions. The water is provided by the public water supplier of the Uberlandia city. Therefore, a buffering effect took place probably due to minerals presented in the water used to prepare the acid and base solutions. A more nonlinear curve would be expected to result if deionized or distillated water was used to prepare the solutions. Water provided from the public water supplier was used in order to reduce the costs with utilities and to render the system more realistic.

In order to validate the physical model an open loop prediction using this model was carried out. The experimental system was excited using a random uniform step sequence for base flow rate with $Q_2 \in [8 - 13,3$ mL/s] with a step probability (probability of a step change occurring at any given sampling instant) equal to 0.8 (Bomberger and Seborg, 1998). Q_1 was kept constant in its nominal value. The sampling time was chosen based on the dynamic of the system and the noise level of the experimental data. A sampling time of 10s was considered satisfactory based on several experimental tests. Figure 4 shows the experimental results and the physical model prediction.

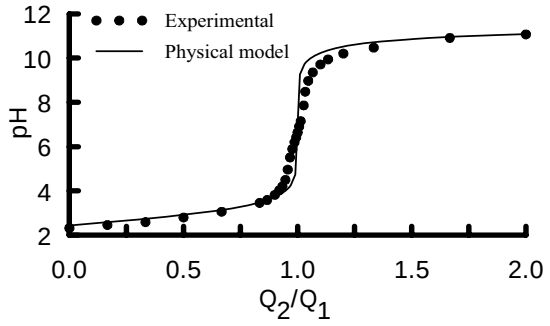

Figure 3: Predicted and experimental titration curves.

Results from Fig. 4 reveal modeling errors are well distributed around zero but with very large amplitude (standard deviation) as for one step as for fifteen ahead prediction. Modeling errors of this magnitude can deteriorate the performance of the MPCs. Due to the bad performance of the physical model a neural network model was tested. The bad performance of the physical model is likely due to presence of other ions in the water used to prepare the solutions. Performance of the neural model will be presented in next section.

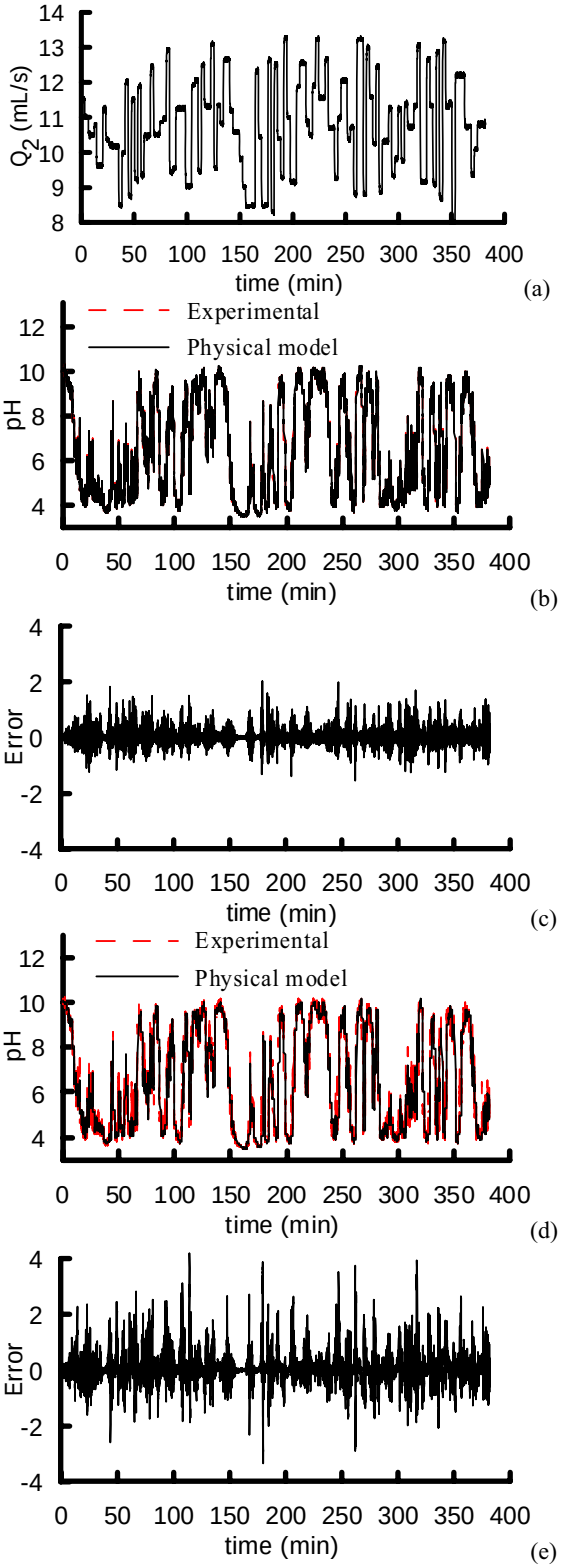

Figure 4: Physical model response. a). system input. b). system output and one step ahead prediction. c). one step ahead prediction error. d). system output and fifteen step ahead prediction. e). fifteen step ahead prediction error.

B. The neural network model

Input/output data are generally available in industrial application. To take advantage of this fact, it was developed a version of proposed controlled based on a neural network model. Before introducing the neural network (NN) model, it is convenient to discuss some practical aspects of the representation of this type of dynamic system. The natural way to represent the dynamic system represented by Eq. (1) is to use neural networks with neurons with dynamic characteristics (You and Nikolaou, 1993). This approach has the advantage of producing models of small dimension. For a single input single output system (SISO), the resulting neural network will have just one input. The prime disadvantage of this type of neural network is the training phase. It is very time consuming and hard to converge. A popular alternative is to consider a neural network with static neurons representing a discrete approximation of the dynamic system in the form of a NARX model (Su *et al*, 1992). In this case, the prime advantage is associated with the simplicity of the training phase. The disadvantage is that the number of required network inputs increases with input and output lags causing a huge increase in network structure. Another problem is that the determination of the input and output lags requires very often a tedious iterative process. For these reasons, in this work we consider a different alternative (Henrique *et al.*, 2000) that consists of the direct representation of Eq. (1) with a static neural network followed by a numerical integration to recover **y**(t+1). Figure 5 shows the neural network topology schematically.

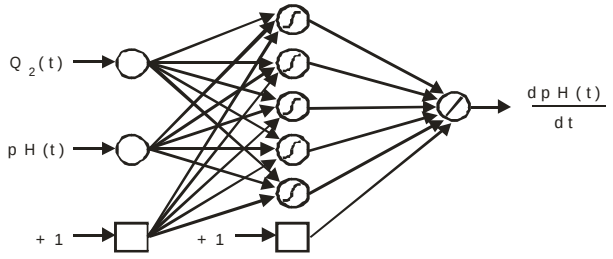

Fig. 5: Neural network topology used.

If the **y**(t) from plant is used as initial condition for obtaining **y**(t+1) by integration then one step ahead prediction is obtained. But if the **y**(t) is obtained from previous integration step then multiple step ahead prediction is obtained. For the pH neutralization process studied, the feedforward neural network model (FNN) predicts the time derivatives of the pH as a function of the base flow rate and the system pH. A sampling period of 10s was used and data from the time interval [0 to 256 min] were used to train the feedforward neural and from the interval [256 to 400 min] to validate the neural model. The network inputs were $Q_2(t)$, pH(t) and the network output was d[pH(t)]/dt. The derivatives of the pH were calculated numerically by finite difference schemes of the filtered pH values.

However, the structure determination of the neural network still remains. It is well known that large neural networks often have large number of redundancies that increase the network complexity without significantly increasing the mapping accuracy. In this paper we used the algorithm proposed by Henrique *et al.* (2000) to determine the network structure. This algorithm is able to identify and eliminate redundant and insignificant network parameters in an efficient fashion. It is based on an orthogonal least-squares pruning method. Results obtained by us and by those authors indicated that the algorithm is very efficient and accurately determines redundant and insignificant network parameters allowing parsimonious feedforward neural network (FNN) models. Statistical criteria used by Henrique *et al.* (2000) confirmed that pruned NN models are more accurate than full models for predicting data not used in the training phase. An extensive study about neural networks is beyond of the aim of this paper. But additional results about real applications can be found in Henrique *et al.* (2000). In the present case, a neural network with five nonlinear hidden neurons (hyperbolic tangent activation function) and one linear output neuron was selected by using the algorithm proposed by Henrique *et al.* (2000) and trained until convergence using the Levenberg-Marquardt method.

Figure 6 shows the experimental input/output and the one step/fifteen step ahead prediction performed by the neural model. This figure reveals the neural model yielded acceptable results as for one step ahead as for fifteen ahead predictions. Figure 7 shows a statistical analysis of the modeling errors. The mean and standard deviation of the one step ahead prediction error are 2.0671e-04 and 0.0398 respectively in training phase and -0.0029 and 0.0422 respectively in the validation phase. For the fifteen ahead prediction the mean and standard deviation of the error are -0.0018 and 0.2215 respectively in training phase and -0.0291 and 0.2293 respectively in the validation phase. These results support the hypothesis that the modeling errors are normally distributed with zero mean and with finite standard deviation (white noise). Therefore, the neural network is sufficiently accurate for using in a MPC strategy. Next section will show the closed loop results.

We have used a neural network to model the system because the physical model did not yield acceptable predictions. The Figures 4, 6 and 7 confirm this result. In addition, when we use the physical model in the MPC formulation we have an additional problem because the state $W_a(t)$ is unmeasured experimentally. In this case it would be needed to design a state estimator for $W_a(t)$. In the neural network modeling fashion we do not have this problem because we can always model the system in an input-output fashion. As a consequence, state estimator is not needed. In the proposed MIMO formulation we have used the vector C as a unitary vector ($C = [1 \ \ldots \ 1]^T$). In the pH application C is a scalar value equal to one. Only the input/output information from system is sufficiently to model and to control the system experimentally. No further information about the state is required.

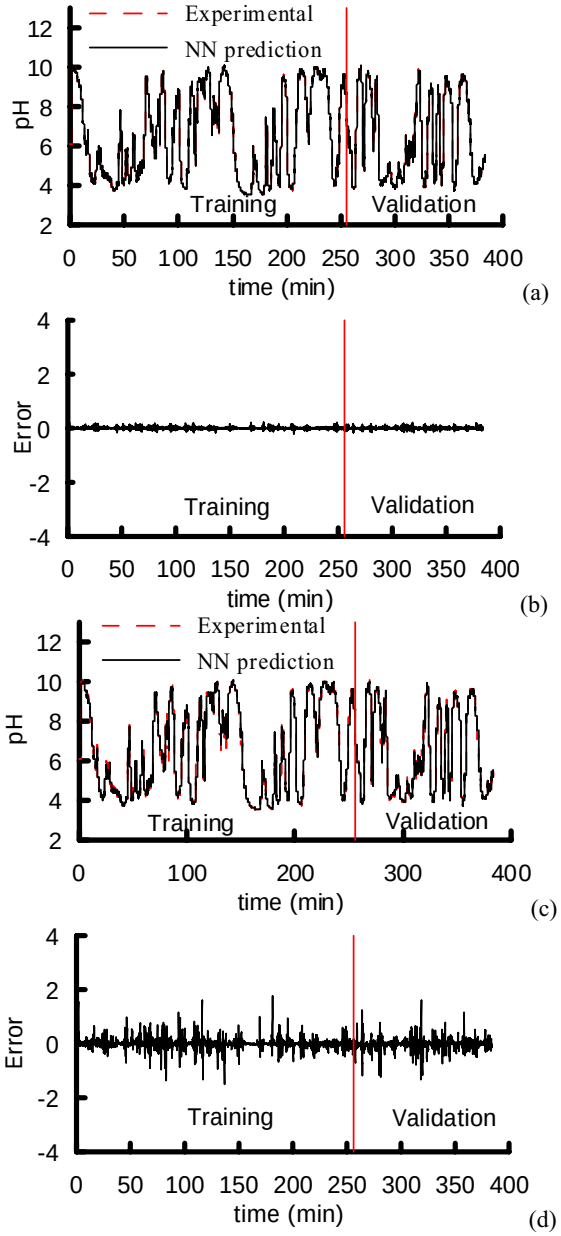

Figure 6: Neural network model response. a). system output and one step ahead prediction. b). one step ahead prediction error. c). system output and fifteen step ahead prediction. d). fifteen step ahead prediction error.

V. CLOSED LOOP RESULTS

In order to illustrate the advantage of proposed technique over classical PID and MPC technique, a digital PID and a classical constrained predictive controller (Eq. 41-44) were implemented experimentally and compared to the proposed controller (Eq. 35-40) for servo and regulator problems (unmeasured perturbation). The neural model developed was used in both MPC controllers. All controllers were first tuned by simulation tests using the NN model to represent the experimental plant. In following a fine field tuning were performed for all three controllers using preliminary parameters in order to get the best controller parameters experimentally adjusted (fine tuning). The tuned PID parameters are Δt = 10 s, K_c = 0.5 s/mL, τ_I = 90 s and τ_D = 0 s. For classical MPC the tuned parameters are Δt = 10 s, M = 10, P = 20, Q = 1, R = 150, $|\Delta Q_{2\ max}|$ = 1 mL/s, $Q_{2\ max}$ = 30 mL/s and $Q_{2\ min}$ = 5 mL/s. For proposed MPC the tuned parameters are Δt = 10 s, K=5e-03 s^{-1}, M = 10, P = 20, R = 150, S = 100, $|\Delta Q_{2\ max}|$ = 1 mL/s, $Q_{2\ max}$ = 30 mL/s, $Q_{2\ min}$ = 5 mL/s. Figure 8 shows results of the controllers in servo problem.

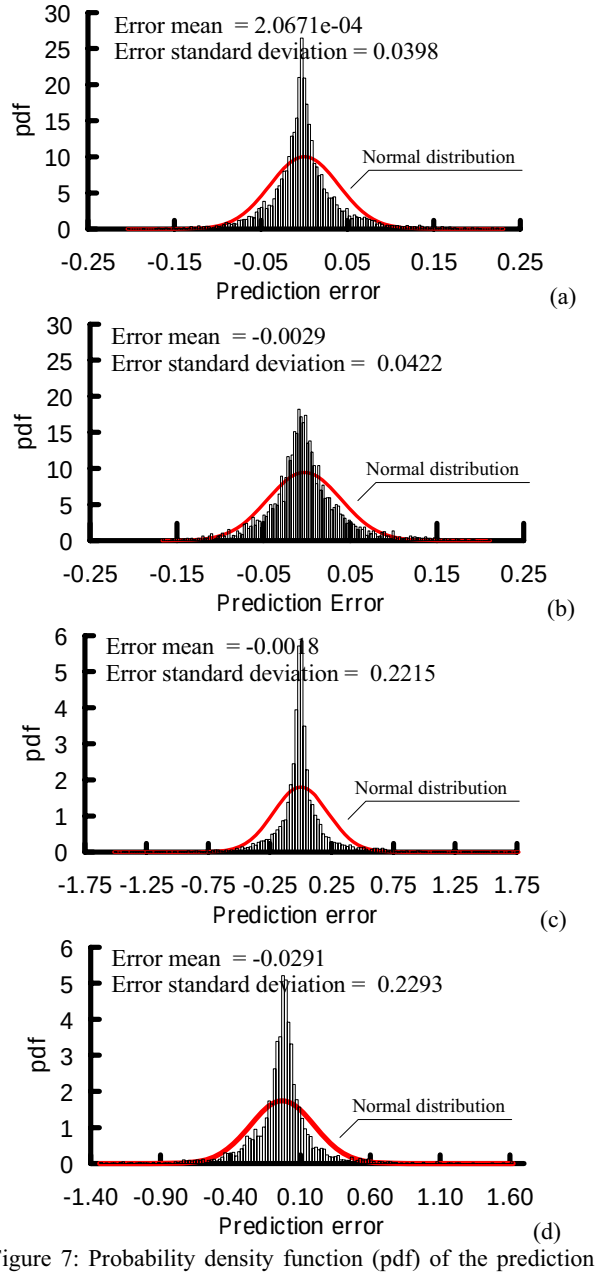

Figure 7: Probability density function (pdf) of the prediction errors: a). One step ahead prediction error of the training data; b). One step ahead prediction error of the validation data; c). Fifteen step ahead prediction error of the training data; d). Fifteen step ahead error of the validation data.

Results from this figure reveal that the proposed MPC yielded an almost symmetric response for setpoint changes. This result is a consequence of the system

reference which was put as an equality constraint in the quadratic programming of the control problem. As a consequence of this the closed loop response of the real system is basically a first order response. Therefore, controller retuning was not needed when new operation points are required. On the other hand, the Fig. 8 shows also that the PID and classical MPC performance are acceptable for system operation at low and high pH values, but the responses deteriorated considerably for system operation around of pH = 7. In order to improve the PID and classical MPC closed loop responses around pH = 7 a new set of the controller parameters is needed. However, if these new set of the controller parameters was used it would deteriorate the closed loop response for low and high pH values (around 4 and 10). Consequently, a different set of the controller parameters must be required for good performance at different operational conditions. This is clearly not a desirable situation in any application since it greatly increases the maintenance needs of the controller. Figure 8 shows also that the control actions for the three controllers. This figure reveals that the PID yielded more aggressive control moves than MPC approaches. The more conservative control moves obtained by MPC approaches are because of the presence of hard constraints in the manipulated variable.

On the other hand, it can be possible to get commercially available pH controllers based on known titration curve that can have performed as good as the proposed controller. But the aim this paper is to present a general control methodology that can be applied to any dynamic system not only for pH system with known titration curve. This actual dynamic system served only as practical example of a hard control system. By the way, the known titration curve was not used to design the proposed MPC controller.

Next, the capacity of unmeasured perturbation rejection of the controllers was tested. Acid flowrate was chosen as system load and the controller parameters were kept the same for all controllers. The run was started with Q_1 = 12 mL/s. It was changed to Q_1 = 13.5 mL/s at instant t = 40 min, to Q_1 = 12 mL/s at instant t = 80 min, to Q_1 = 10.5 mL/s at instant t = 120 min and to Q_1 = 12 mL/s at instant t = 160 min. After t = 160 min the acid flowrate was kept in its initial value Q_1 = 12 mL/s. The initial condition of the system is pH =7. In this region the magnitude of the change in Q_1 are too severe because of the high value of the system static gain. Figure 9 shows the controller performances for load changes. This figure reveals that the PID controller yielded unstable response, very oscillatory with increasing amplitudes. The MPC approaches yielded stable and acceptable responses around of the operation

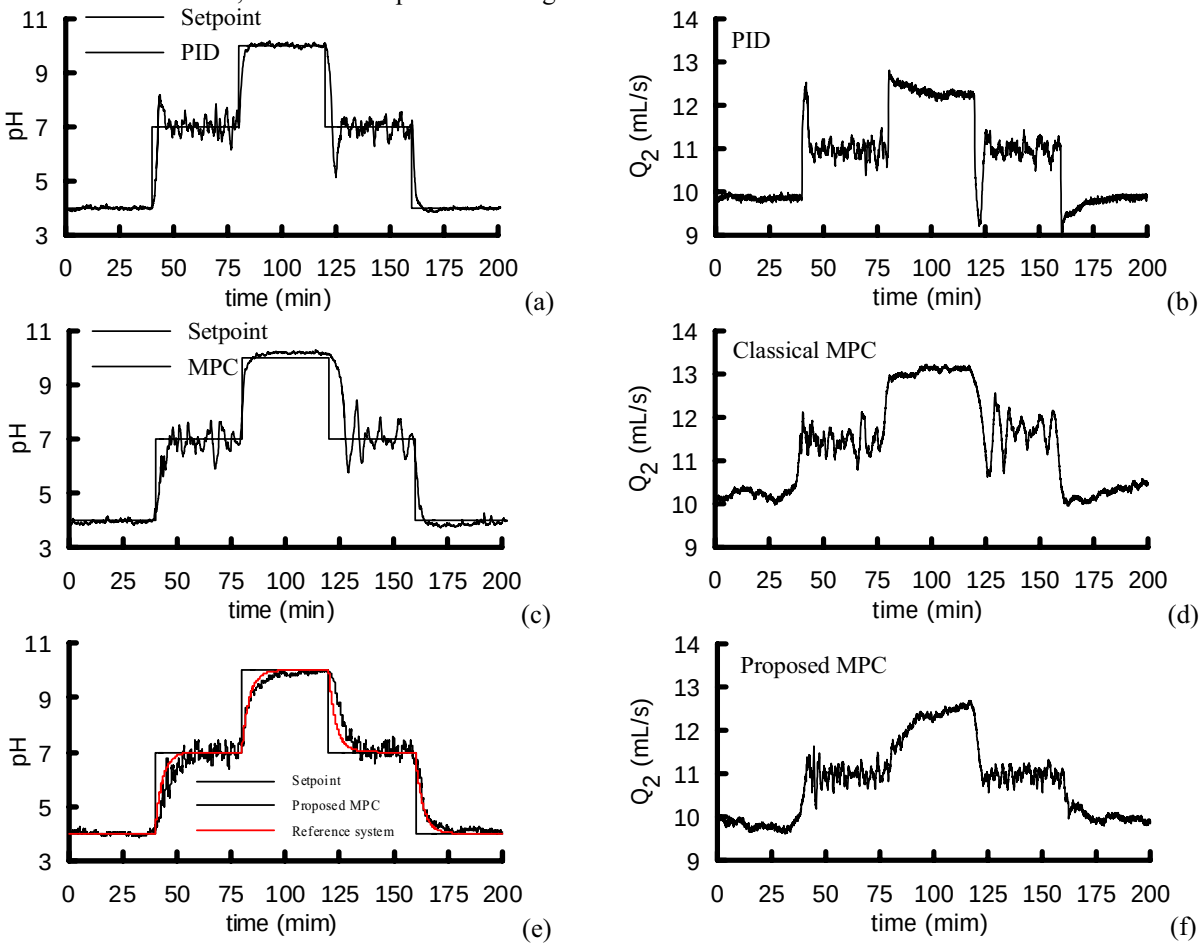

Figure 8: Closed loop responses (input and output) for setpoint changes: a) and b). PID; c). and d). Classical MPC; e). and f). Proposed MPC.

point (pH = 7). These behaviors are remarkable because the NN network was only trained to Q_1 = 12 mL/s. Due to high sensibility of the static gain of the system around of pH =7 (see Fig. 3) these changes in Q_1 have a drastic impact in accuracy of the NN model. In spite of this fact the MPCs controlled the system in a stable fashion. The PID control moves were very aggressive and the MPC control moves were acceptable and no violation of the limits occurred.

VI. CONCLUSION

This paper introduces a MPC strategy based on a continuously linearized neural model obtained from actual input/output data. The closed loop output of the system behaviors as a first order system because of the equality constraint added to the quadratic programming involved. As a result, controller retuning for different operating regions is not necessary. This is clearly a desirable situation in any practical application since it decreases the maintenance needs of the controller. The proposed MPC has one more tuning parameter than the classical MPCs. However this parameter is easily tuned because it has physical meaning. It is the reciprocal of the closed loop time constant for SISO system. This parameter controls how fast or how slow will be the closed loop response.

The proposed control algorithm was developed and implemented experimentally. Results of a pH neutralization process showed the proposed controller was clearly superior to the PID and the classical MPC for servo and regulator problem. The proposed method retains the computational simplicity while providing some desirable features from model predictive control, such as constraint handling, incorporating future setpoint changes, penalizing large control move increments by selecting appropriate weighting parameters in the objective function. It was also verified the proposed technique yielded promising results in a real control problem confirming its good potential for practical implementation due to low computational requirements, good closed loop performance as well as transparent controller tuning.

ACKNOWLEDGEMENTS

The authors thank the National Council for Scientific and Technological Development (CNPq) and the Foundation of Support to the Research of the State of Minas Gerais (FAPEMIG) for providing financial support for this research.

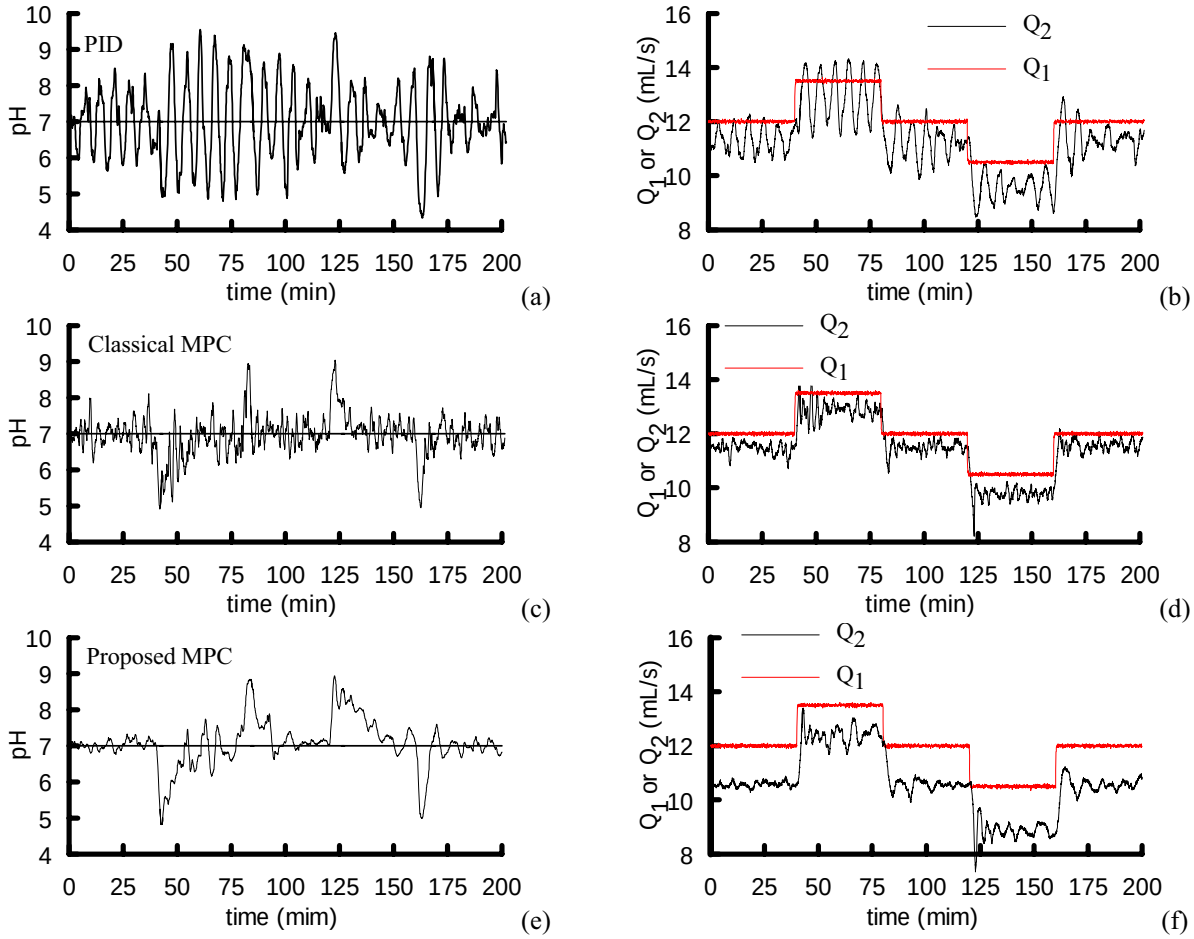

Figure 9: Closed loop responses (input and output) for load changes: a) and b). PID; c). and d). Classical MPC; e). and f). Proposed MPC.

REFERENCES

Balchen, J.G. and B. Sandrib, "Input saturation in nonlinear multivariable processes resolved by nonlinear decoupling," *Model. Ident. Control*, **16**, 95-106 (1995).

Bequette, B.W., "Nonlinear Control of Chemical Process: A Review," *Ind. Eng. Chem. Res.*, **30**, 1391-1413 (1991).

Bomberger, J.D. and D.E. Seborg, "Determination of Model Order for NARX Models Directly from Input-Output data", *J. Process Control*, **8**, 459-468 (1998).

Borges, R.M., *Controle Preditivo Basedo em Sistema de Referência*, Master science thesis, Uberlandia, Brazil (2001).

Garcia, C.E. and A.M. Morshedi, "Quadratic Programming Solution of Dynamic Matrix Control (QDMC)," *Chem. Engng. Commun.*, **46**, 73-87 (1986).

Garcia, C.E., D.M. Prett and M. Morari, "Model Predictive Control – A Survey," *Automatica*, **25**, 335-348 (1989).

Gattu, G. and E. Zafiriou, "Nonlinear Quadratic Dynamic Matrix Control With State Estimation," *Ind. and Eng. Chem. Res.*, **31**, 1096-1104 (1992).

Gustafsson, T.K. and K. V. Waller, "Dynamic modeling and reaction invariant control of pH," *Chem. Eng. Sci.*, **38**, 389-398 (1983).

Henrique, H.M., E.L. Lima and D.E. Seborg, "Model Structure Determination in Neural Network Models," *Chem. Eng. Sci*., **55**, 5457-5469 (2000).

Henson, M.A. and D.E. Seborg, *Nonlinear Process Control*, Prentice Hall, New Jersey (1997).

Kalra, L., C. Georgakis and L.C.O. Lopes, "Reference System Model Predictive Control I- Continuous Time Formulation and Case Studies on Performance," *Industrial & Engineering Chemical Research*, **41**, 3199-3212 (2002)

Kendi, T.A. and F. J. Doyle, "An anti-windup scheme for input-output linearization," *Proc. European Control Conf.*, Rome, Italy, (1995).

Kravaris, C. and J.C. Kantor, "Geometric methods for nonlinear process control: 2. controller synthesis," *Ind. Eng. Chem. Res.*, **29**, 2310-2323 (1990).

Lee, P.L. and G.R. Sullivan, "Generic Model Control (GMC)," *Comp. Chem. Eng.*, **12**, 573-580 (1988).

Lee, P.L., "Nonlinear Process Control: Applications of Generic Model Control," Sprong-Verlag, London (1993).

Lopes, L.C., "*Reference System Nonlinear Model Predictive Control*," PhD thesis, Lehigh University, Bethlehem, Pennsylvania (2000).

Meadows, E.S., M.A. Henson, J.W. Eaton and J.B. Rawlings, "Recending horizon control and discontinuous state feedback stabilization," *Int. J. Control*, **62**, 1217-1229 (1995).

Montandon, A.G., *Controle Preditivo em Tempo Real com Trajetória de Referência Baseado em Modelo Neural para Reator de Neutralização*, Master science thesis, Uberlandia, Brazil (2005). (in Portuguese)

Rawlings, J.B., E.S. Meadows and K.R. Muske, "Nonlinear model predictive control: A tutorial and survey," *Proc. IFAC Symposium on Advanced Control of Chemical Processes*, Kyoto, Japan, 203-214 (1994.).

Su, H., T.J. McAvoy and P. Werbos, "Long Term Predictions of Chemical Processes Using Recurrent Neural Networks: A Parallel Training Approach," *Ind. Chem. Res.*, **31**, 1338-1352 (1992).

You, Y. and M. Nikolaou, "Dynamic Process Modeling with Recurrent Neural Networks," *AIChE Journal*, **39**, 1654-1667 (1993).

Received: August 21, 2006.
Accepted: June 5, 2007.
Recommended by Subject Editor Julio Braslavsky.