ISA Transactions 42 (2003) 227–240

ISA
TRANSACTIONS®

Model predictive control with amplitude and rate actuator saturation

Leonardo L. Giovanini*

Electronic and Control Research Group, Universidad Tecnológica Nacional, Regional School at Villa Maria, Av. Universidad 450, 5900 Villa María, Argentina

(Received 19 July 2001)

Abstract

In this work we show that the anti-wind-up-bumpless-transfer controller emerges from the structure of model predictive control (MPC) with quadratic objective and input constrains. The key to establish that relationship is the application of optimality conditions to the equivalent optimal control problem. The proposed framework employs a model of physical constraints as part of the controller architecture to ensure that the commands sent to the actuator do not exceed their specific limits and the internal states of the controller are well updated. Numerical examples are presented for illustrating the proposed control design methodology. © 2003 ISA—The Instrumentation, Systems, and Automation Society.

Keywords: Optimal control; Minimum principle; Anti-wind-up control

1. Introduction

Controller design for linear process with actuator saturation nonlinearities has long been studied within various contexts. In general, there are two different classes of structures that handle input saturations:

1. On-line optimization based control structures, such us *model predictive control* (MPC) which modifies its structure every time that the optimization problem is solved in the way of obtaining the best possible performance [1], and

2. *Variable structure controllers* that have a closed form and do not perform on-line optimization. They change their structure with a predetermined logic, avoiding the solving of an optimization problem [2–4]. If they are correctly designed, they can provide an optimal and robust performance of the closed-loop system.

Controllers in the first class rely on a process model and on-line solution of a constrained optimization problem that minimizes an objective function over a future horizon [1,5]. If properly designed, such controllers can provide optimality, robustness, and other desirable properties [6]. However, because of the time needed to perform the on-line optimization, these controllers are usually implemented on relatively slow processes. For fast processes the implementation of MPC does not use on-line optimization, tunable parameters such as prediction and control horizon length and weighting factors in the objective function can be adjusted to achieve desirable closed-loop properties in the presence of constrains [7–9].

Controllers in the second class completely bypass on-line optimization, therefore they are inherently faster and can be used on faster processes. The anti-wind-up bumpless transfer (AWBT) controller design approach is based on the following two-step design paradigm [2]: First, a linear controller is designed ignoring input constraints. In

*Tel.: 54-353-4537500; fax: 54-353-4535498. *E-mail address*: lgiovani@fich1.unl.edu.ar

the next step, an anti-wind-up scheme is added to compensate the adverse effects of the constrains on the closed-loop performance. There are many heuristic techniques for designing AWBT controllers, which can be summarized into a structure that includes a saturation nonlinearity in the forward path and a linear transfer function in the feedback path. Kothare *et al.* [4] unified all the existing AWBT schemes and developed a general framework for studying stability and robustness issues. The importance of that work lies in that the model uncertainties can be taken into account systematically.

Although MPC and AWBT control methods can be applied to the same problem, no connections between these control techniques has appeared in literature. In this work we show that there is a relationship between MPC and AWBT control. In fact, we show that the general structure of AWBT controller emerges naturally from the structure of MPC controller with quadratic objective, input constraints, and plant model structure affine in input variables. Closed-loop stability, regulatory performance, and sensitivity analysis to model mismatch issues will be considered in a future work. The outline of this paper is as follow. In Section 2, we show the relationship between MPC and optimal control. The general structure of optimal control problems emerges naturally from the structure of MPC. In particular, we rigorously show that the *linear quadratic regulator* is derived from MPC with quadratic objective function. From this point of view, we analyze the effect of the tuning parameters on the closed-loop behavior. Section 3 concentrates on providing an analytical solution to the optimal-constrained control problem, equivalent to MPC controller. The key tools use for obtaining such solution are the optimality conditions and Pontryagin's minimum principle. Then, the general solution is applied to free final state and the closed-form structures are obtained. The proposed framework employs the model of saturations as part of the controller architecture to ensure that no rate and no amplitude commands are sent to the actuators that exceed their specific limits and the internal states of the controller are correctly up dated. In Section 4, we illustrate the application of this framework by showing two simulation examples on SISO linear systems. Finally, we present the conclusions and discuss future works on the topics in Section 5.

2. Relationship between MPC and optimal control

Given a linear system, a MPC controller solves at each sample time step the following optimization problem:

$$\min_{\Delta u(k+j)\, j=0,1,\ldots,U-1} \sum_{i=0}^{V} w_y \hat{e}^2(k+i) + \sum_{j=0}^{U-1} w_u \Delta u^2(k+j),$$

$$\text{st.} \tag{1}$$

$$x(k+i+1)=Ax(k+i)+Bu(k+i) \quad x(k),$$

$$r(k+i)=A_r x_r(k+i)+B_r u_r(k+i) \quad x_r(k),$$

where V is the prediction horizon and $U \leqslant V$ is the control horizon along with the output weigthing matrix w_y and the input weighting matrix w_u are the user specified tuning parameters. The states measured at the present time $[x(k)$ and $x_r(k)]$ are the initial conditions of the optimization problem. Employing the extended space-state model (37) (see Appendix A) and defining the weights as

$$Q=C_{\mathcal{X}}^T w_y C_{\mathcal{X}}, \tag{2a}$$

$$R=w_u, \tag{2b}$$

where

$$C_{\mathcal{X}}=[-C \quad 0 \quad C_r],$$

we can write the MPC controller (1) like the following optimal control problem:

$$\min_{\mathcal{U}(k+j)\, j=0,1,\ldots,U-1} \sum_{i=0}^{V} \mathcal{X}^T(k+i) Q \mathcal{X}(k+i) + \sum_{j=0}^{U-1} \mathcal{U}^T(k+j) R \mathcal{U}(k+j), \tag{3a}$$

$$\text{st.}$$

$$\mathcal{X}(k+i+1)=A_{\mathcal{X}}\mathcal{X}(k+i)+B_{\mathcal{U}}\mathcal{U}(k+i) + B_{\mathcal{R}}\mathcal{R}(k+i) \quad \mathcal{X}(k). \tag{3b}$$

Due to the receding horizon control philosophy using in the MPC controllers, the problem (3) is solved at each sample with the initial condition $\mathcal{X}(k)$ given by

$$\mathcal{X}(k) = [x(k) \;\; u(k-1) \;\; r(k)]^T.$$

Splitting the predicted errors that are inside the control horizon from those that are outside the control horizon, the objective function (3a) can be written as

$$\sum_{i=U}^{V} \mathcal{X}^T(k+i) Q \mathcal{X}(k+i) + \sum_{i=0}^{U-1} [\mathcal{X}^T(k+i) Q \mathcal{X}(k+i) + \mathcal{U}^T(k+i) R \mathcal{U}(k+i)]. \tag{4}$$

The first term of this equation measures the system performance outside of the control horizon. It can be written as function of the system states at time $k+U$ and the control action computed by the control problem $\mathcal{U}(k+i)$, $i=0,1,\ldots,U-1$ (Appendix B). The result is

$$\sum_{i=U}^{V} \mathcal{X}^T(k+i) Q \mathcal{X}(k+i) = \mathcal{X}^T(k+U) S_{\mathcal{X}} \mathcal{X}(k+U) + \sum_{i=0}^{U-1} \mathcal{U}^T(k+i) S_{\mathcal{U}} \mathcal{U}(k+i), \tag{5}$$

where

$$S_{\mathcal{X}} = \sum_{i=0}^{V-U} A_{\mathcal{X}}^{i\,T} Q A_{\mathcal{X}}^{i}, \tag{6a}$$

$$S_{\mathcal{U}} = \sum_{i=0}^{V-U} B_{\mathcal{U}}^T \left[\sum_{j=0}^{i} A_{\mathcal{X}}^{j\,T} \right] Q \left[\sum_{j=0}^{i} A_{\mathcal{X}}^{j} \right] B_{\mathcal{U}}. \tag{6b}$$

Before we continue our analysis, a remark about these expressions must be made. Note that Eq. (5) measures the cost of the uncontrolled portion of state trajectory ($U \leqslant i \leqslant V$). So it acts like a Lypunov function of the closed-loop system that could ensure the system stability if the controller parameters are properly selected. From Eqs. (6) we can see that the controller parameters which can be manipulated to guarantee the closed-loop stability, for a given U, are the prediction horizon V and the state weight w_y. Now, defining the matrix

$$R_{\mathcal{U}} = R + S_{\mathcal{U}}, \tag{7}$$

the MPC controller (1) can be written as the following optimal control problem:

$$\min_{\substack{\mathcal{U}(k+i) \\ i=0,1,\ldots,U-1}} \mathcal{X}^T(k+U) S_{\mathcal{X}} \mathcal{X}(k+U) + \sum_{i=0}^{U-1} [\mathcal{X}^T(k+i) Q \mathcal{X}(k+i) + \mathcal{U}^T(k+i) R_{\mathcal{U}} \mathcal{U}(k+i)],$$

$$\text{st.} \tag{8}$$

$$\mathcal{X}(k+i+1) = A_{\mathcal{X}} \mathcal{X}(k+i) + B_{\mathcal{U}} \mathcal{U}(k+i) + B_{\mathcal{R}} \mathcal{R}(k+i) \mathcal{X}(k).$$

Observe that the resultant optimal control problem has fixed initial condition, the states measured at present time $\mathcal{X}(k)$, and does not specify final state. Notice that the control law computed by the predictive controller (1) is the same as that designed by the optimal control problem (8). The only difference is the tools used to solve them: dynamic programing for the optimal control problem (8) and a parametric optimization program for the predictive controller (1).

2.1. Effect of tuning parameters on closed-loop behavior

Many authors show that the choice of MPC tuning parameters have a significant effect on the closed-loop stability and performance. Now based on the results of the previous section we want to analyze the effect of each parameter over the system response. The parameters that must be selected are: prediction horizon V, control horizon U, and penalty weight matrices w_u and w_y. Eqs. (4)–(7) show a strong relationship between them, which is shown by a couple effect on the closed-loop behavior [1,11].

- **Prediction horizon**: It is well known that longer prediction horizon produces a more robust controller, which result in a poor closed-loop performance [11]. These facts can be easily seen from Eqs. (4)–(7), where an increment of V increases the number of terms employed to computed $S_{\mathcal{X}}$ and $S_{\mathcal{U}}$ penalizing both: (i) the behavior outside the control horizon and (ii) the control actions employed to control the system. So, the controller tends to control the system smoothly and improve the closed-loop stability.

- **Control horizon**: Linear systems results indicate that longer control horizon relative to prediction horizon produces more aggressive controllers that achieve better closed-loop performance, employing more control energy. This effect is the same of reduce the prediction horizon V. Numerical results show that control horizon greater than five samples has a similar performance than controllers designed with this control horizon value [11].
- **Penalty weights**: For the choice of weighting matrices there are few rules; it is an active research area yet. Rahul and Cooper [12] derive an analytical expression for control weight w_u. This expression depends on the control horizon, sampling time, and the condition number of the controller.

The effect of the trajectory weight w_y on the closed-loop response is clear from Eqs. (6): it has a direct effect on the performance and stability of the closed-loop system. It influences both the state trajectory inside and outside the control horizon. On the other hand, the control weight w_u only acts on the performance. In practice, it is common to use only one value of w_y for the whole predicted trajectory and define the performance using the control weight w_u [11], due to its direct effect. However, it can be used different weights for the predictions inside ($0 \leqslant i \leqslant U-1$) and outside ($U \leqslant i \leqslant V$) the control horizon called w_{yu} and w_{yy}, respectively. Under this design condition we can use w_{yy} to guarantee the closed-loop stability [see Eqs. (4)–(6)] of all plants of the family employed to describe the system [13].

3. MPC anti-wind-up scheme

Most practical control problems are dominated by inequality constrains. In general, there are two types of constrains:

- **output constraints**, imposed by the operative conditions, and
- **input bounding constraints**, imposed by physical limitations of actuators.

The last ones cannot be ignored under any condition, since they may lead to significant deterioration in the closed-loop performance and even closed-loop instability.

In the case of MPC, as well as in optimal control, constrains are explicitly accounted for, and the controller action is a solution to a constrained optimization problem. So, the optimal control problem derived in the previous section [Eq. (8)] should be written as

$$\min_{\mathcal{U}(k+i)\quad i=0,1,\ldots,U-1} \mathcal{X}^T(k+U)S_{\mathcal{X}}\mathcal{X}(k+U) + \sum_{i=0}^{U-1}[\mathcal{X}(k+i)^T Q\mathcal{X}(k+i) + \mathcal{U}(k+i)^T R_{\mathcal{U}}\mathcal{U}(k+i)],$$

st.

$$\mathcal{X}(k+i+1) = A_{\mathcal{X}}\mathcal{X}(k+i) + B_{\mathcal{U}}\mathcal{U}(k+i) + B_{\mathcal{R}}\mathcal{R}(k+i)\mathcal{X}(k),$$

$$f(\mathcal{X}(k+i),\mathcal{U}(k+i)) \leqslant 0, \tag{9}$$

where $f(\mathcal{X}(k),\mathcal{U}(k))$ is a function that describes the constraints present in the manipulated variable. The solution of this problem must satisfy the *minimum principle* or the optimality conditions [14],

$$\mathcal{X}(k+i+1) = \frac{\partial H[\mathcal{X}(k+i),\mathcal{U}(k+i),\lambda(k+i+1)]}{\partial \lambda(k+i)} \mathcal{X}(k), \tag{10a}$$

$$\lambda(k+i) = \frac{\partial H[\mathcal{X}(k+i),\mathcal{U}(k+i),\lambda(k+i+1)]}{\partial \mathcal{X}(k+i)} \lambda(k+U), \tag{10b}$$

$$0 = \frac{\partial H[\mathcal{X}(k+i),\mathcal{U}(k+i),\lambda(k+i+1)]}{\partial \mathcal{U}(k+i)}, \tag{10c}$$

where $H(:)$ is the Hamiltonian function of the system, $\lambda(:)$ is the Lagrange multiplier or system costates, and $\mathcal{X}(k)$ and $\lambda(k+U)$ are the boundary conditions. These conditions are certainly not intuitively obvious, so we should discuss them a little. Writing Eq. (10) explicitly in terms of its components we can see that Eq. (10a) is the constrain that describes the dynamic behavior of the system. Eq. (10b), known as the *adjoint system*, describes the dynamic behavior of the costates $\lambda(k)$. The system (10a) and the adjoint system (10b) are coupled difference equations that describe the dynamical behavior of states as function of costates. They define a two-point boundary problem, since the boundary conditions required for the solution are the initial state $\mathcal{X}(k)$ and the final costate $\lambda(k+U)$. Finally, Eq. (10c) is called the *stationary condition*, it allows us to express the control $\mathcal{U}(k)$ as function of costates and guarantees the optimality of control sequence.

To solve this dynamical recursion we have to specify the split boundary conditions. They are given by

$$\left[\frac{\partial H[\mathcal{X}(k),\mathcal{U}(k),\lambda(k+1)]}{\partial \mathcal{X}(k)}\right] d\mathcal{X}(k)=0, \tag{11a}$$

$$\left[\frac{\partial \phi[\mathcal{X}(k+U),U]}{\partial \mathcal{X}(k+U)}-\lambda(k+U)\right] d\mathcal{X}(k+U)=0. \tag{11b}$$

Eq. (11a) holds only at initial time k and it depends on the initial states. In our application the system starts at known initial value: the measured state $\mathcal{X}(k)$. Thus $d\mathcal{X}(k)=0$ and Eq. (11a) is held regardless of the value of $H(k)$. On the other hand, Eq. (11b) holds at the final time $k+U$ and it depends on the behavior of the system outside the control horizon.

Applying these expressions to problem (9) gives

$$\mathcal{X}(k+i+1)=A_{\mathcal{X}}\mathcal{X}(k+i)+B_{\mathcal{U}}\mathcal{U}(k+i)\mathcal{X}(k), \tag{12a}$$

$$\lambda(k+i)=Q\mathcal{X}(k+i)+A_{\mathcal{X}}^{T}\lambda(k+i+1)$$

$$\lambda(k+U), \tag{12b}$$

$$\mathcal{U}(k+i)=-R_{\mathcal{U}}^{-1}B_{\mathcal{U}}\lambda(k+i+1). \tag{12c}$$

The control actions are given by

$$u(k+i)=P_U\mathcal{X}(k)+\sum_{l=0}^{i}\mathcal{U}(k+l),$$

$$i=0,1,\ldots,U-1, \tag{13}$$

where P_U is the matrix,

$$P_U=[\mathbf{0}\ \ 1\ \ \mathbf{0}].$$

The actual control action $u(k)$ will be given by

$$u(k)=\mathcal{P}U(k), \tag{14}$$

where $U(k)$ is the vector of future control actions

$$U(k)=[u(k)\quad u(k+1)\ \ \cdots\ \ u(k+U-1)]^T$$

and $\mathcal{P}$ is projection matrix

$$\mathcal{P}=[\mathbf{1}\ \ 0\ \ \cdots\ \ 0].$$

The presence of constrains in the actuators forces us to modify Eq. (12c), applying Pontryagin's minimum principle over the admissible solution set [15]. Since MPC computes the changes of control action instead of control actions, we should translate the amplitude saturation into a rate saturation. So, we can define the upper and lower admissible instantaneous changes $\underline{\mathcal{U}}(k+i)$ and $\overline{\mathcal{U}}(k+i)$ as

$$\underline{\mathcal{U}}(k+i)=u_{\min}-P_U\mathcal{X}(k+i), \tag{15a}$$

$$\overline{\mathcal{U}}(k+i)=u_{\max}-P_U\mathcal{X}(k+i). \tag{15b}$$

These constraints are time varying and will be active when the computed control action $u(k+i)$ overcomes the amplitude saturation limits. They are the maximum change in the control signal allowed at time $k+i$ by the amplitude saturation. When amplitude and rate saturation are simultaneously presenting the control $\mathcal{U}(k+i)$ action should be written as follows:

$$\mathcal{U}(k+i)=\max\{\mathcal{U}_{\min},\underline{\mathcal{U}}(k+i),$$

$$\min\{\mathcal{U}_{\max},\overline{\mathcal{U}}(k+i),-R_{\mathcal{U}}^{-1}B_{\mathcal{U}}\lambda(k+i+1)\}\}. \tag{16}$$

Replacing the costates $\lambda(k+i+1)$ and the admissible instantaneous changes $\underline{\mathcal{U}}$ and $\overline{\mathcal{U}}$ by their expressions [Eqs. (12c) and (15)], the constraint (16) is equivalent to (Fig. 1)

$$u(k+i)=f_u\{u(k+i-1)$$

$$-f_{\Delta}[R_{\mathcal{U}}^{-1}B_{\mathcal{U}}\lambda(k+i+1)]\},$$

$$i=0,1,2,\ldots,U-1, \tag{17}$$

$$=f_u\{u(k+i-1)-f_{\Delta}[\Delta u(k+i)]\}. \tag{18}$$

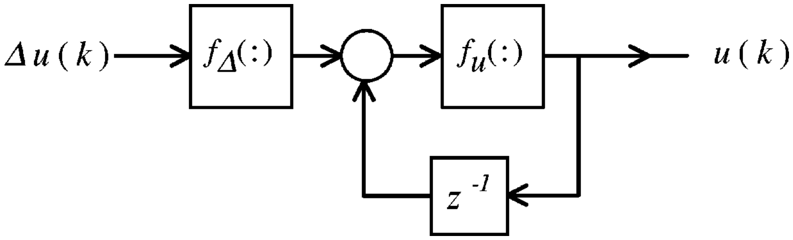

Fig. 1. Structure of the actuator's constraints.

Notice that Eqs. (16) and (17) mean that we have to apply all the control action allowed by the actuator such that the information employed by the controller to compute the next control action would be well updated.

At this point, we must note that Eqs. (12)–(16) correspond to a static system, where the knowledge of the actual state $\mathcal{X}(k)$ and the behavior outside the control horizon are sufficient to compute everything else. Fig. 2 shows a block diagram interpretation of Eqs. (12)–(16), where block $\mathcal{M}$ is the set of algebraic equations corresponding to the system dynamic

$$\mathcal{X}(k+i)=\mathcal{A}_{\mathcal{X}}^{i}\mathcal{X}(k)+\sum_{l=0}^{i-1}\mathcal{U}(k+l),$$

$$i=1,2,\ldots,U,$$

block $\mathcal{L}$ is the set of algebraic equations corresponding to costates behavior

$$\lambda(k+i-1)=Q\mathcal{X}(k+i-1)+A_{\mathcal{X}}^{T}\lambda(k+i),$$

$$i=1,2,\ldots,U,$$

$f(:)$ is the function (16), $\Delta U(k)$ is the vector of future changes in control actions,

$$\begin{aligned}\Delta U(k)&=U(k)-U(k-1)\\&=[\mathcal{U}(k)\quad \mathcal{U}(k+1)\ \cdots\ \mathcal{U}(k+U-1)]^{T},\end{aligned}$$

and S is the following matrix:

$$S=\begin{bmatrix}1&0&0&&0\\1&1&0&&0\\&\vdots&&\ddots&\vdots\\1&1&1&&1\end{bmatrix}.$$

In the following section we will apply the expressions derived in this section to a MPC controller and we will derive the expressions and the closed-form structure.

3.1. The controller

When we consider the standard MPC, the behavior of the system outside the control horizon is free [1]. So, the equivalent optimal control problem (9) has a free final state. In this case $d\mathcal{X}(k+U)\neq 0$ and the boundary condition (11b) is only held if

$$\lambda(k+U)=\frac{\partial\phi[\mathcal{X}(k+U),U]}{\partial\mathcal{X}(k+U)}.$$

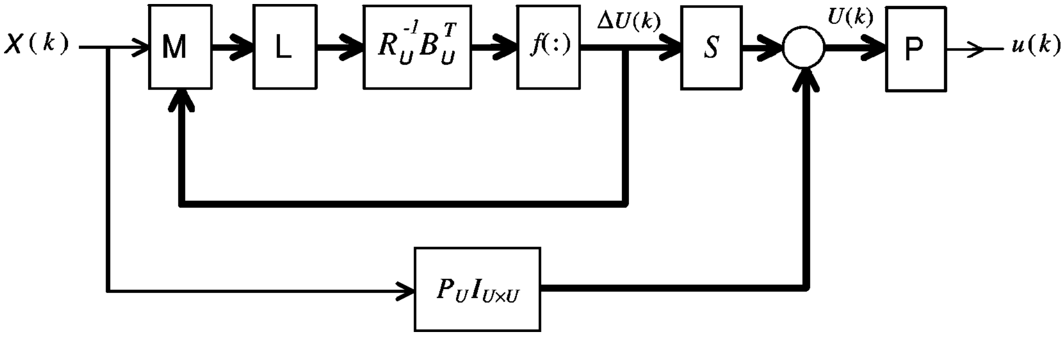

Fig. 2. Structure of the generic MPC controller.

Under this design condition, the final costate $\lambda(k+U)$ is given by

$$\lambda(k+U)=S_{\mathcal{X}}\mathcal{X}(k+U). \tag{19}$$

To solve the optimal control problem we employ the *swapping method*, such that we can write the costates as the boundary condition (19),

$$\lambda(k)=S_{\mathcal{X}}\mathcal{X}(k). \tag{20}$$

Employing the extended system model, the costates can be expressed in the following form:

$$\begin{aligned}\lambda(k+i)&=A_{\mathcal{X}}S_{\mathcal{X}}\mathcal{X}(k+i+1)+Q\mathcal{X}(k+i),\\ &\qquad\qquad i=0,1,\dots,U-1,\\ &=(A_{\mathcal{X}}S_{\mathcal{X}}A_{\mathcal{X}}+Q)\mathcal{X}(k+i)\\ &\quad+A_{\mathcal{X}}S_{\mathcal{X}}B_{\mathcal{U}}\mathcal{U}(k+i).\end{aligned}$$

Since the predicted states are given by

$$\mathcal{X}(k+i)=A_{\mathcal{X}}^{i}\mathcal{X}(k)+\sum_{j=0}^{i-1}A_{\mathcal{X}}^{j}B_{\mathcal{U}}\mathcal{U}(k+i-j),$$

the costates can be written in the following form:

$$\begin{aligned}\lambda(k+i)={}&(A_{\mathcal{X}}S_{\mathcal{X}}A_{\mathcal{X}}+Q)\Bigg[A_{\mathcal{X}}^{i}\mathcal{X}(k)\\ &+\sum_{j=0}^{i-1}A_{\mathcal{X}}^{j}B_{\mathcal{U}}\mathcal{U}(k+i-j)\Bigg]\\ &+A_{\mathcal{X}}S_{\mathcal{X}}B_{\mathcal{U}}\mathcal{U}(k+i).\end{aligned} \tag{21}$$

Replacing these expression in Eqs. (12c) and (13) we obtain the unconstrained control actions vector,

$$\begin{aligned}U(k)={}&I_{U\times U}P_{U}\mathcal{X}(k)+SR_{\mathcal{U}}^{-1}B_{\mathcal{U}}^{T}[L_1L_2\mathcal{X}(k)\\ &+(L_3+L_4)B_{\mathcal{U}}\Delta U(k)],\end{aligned} \tag{22}$$

where

$$L_1=\begin{bmatrix}(A_{\mathcal{X}}S_{\mathcal{X}}A_{\mathcal{X}}+Q) & & 0\\ \vdots & \ddots & \vdots\\ 0 & & (A_{\mathcal{X}}S_{\mathcal{X}}A_{\mathcal{X}}+Q)\end{bmatrix},$$

$$L_2=\begin{bmatrix}A_{\mathcal{X}}^{1}\\ \vdots\\ A_{\mathcal{X}}^{U-1}\end{bmatrix}, \tag{23a}$$

$$L_3=\begin{bmatrix}I & 0 & \cdots & 0\\ A_{\mathcal{X}} & I & & 0\\ \vdots & & \ddots & \vdots\\ A_{\mathcal{X}}^{U-1} & A_{\mathcal{X}}^{U-2} & \cdots & I\end{bmatrix},$$

$$L_4=\begin{bmatrix}A_{\mathcal{X}}S_{\mathcal{X}} & \cdots & 0\\ \cdots & \ddots & \vdots\\ 0 & \cdots & A_{\mathcal{X}}S_{\mathcal{X}}\end{bmatrix}. \tag{23b}$$

Note that $P_U\mathcal{X}(k)$ is just $u(k-1)$, which can be also obtained from $U(k)$ by multiplying by $\mathcal{P}$ and applying a time delay,

$$P_U\mathcal{X}(k)=\mathcal{P}U(k)z^{-1}. \tag{24}$$

Taking into account this idea and working with Eq. (22) we can write it as

$$\begin{aligned}U(k)={}&I_{U\times U}\mathcal{P}U(k)z^{-1}+SR_{\mathcal{U}}^{-1}B_{\mathcal{U}}^{T}L_1L_2\mathcal{X}(k)\\ &+SR_{\mathcal{U}}^{-1}B_{\mathcal{U}}^{T}(L_1L_3+L_4)B_{\mathcal{U}}S^{-1}\\ &\times(I-I_{U\times U}\mathcal{P}z^{-1})U(k).\end{aligned}$$

Observe that any control action $u(k+i)$ is given in terms of the actual state, the past control action, and the system and weighting matrices. Therefore the gains of the controller does not depend on the state trajectory and they can be computed before the control is ever applied to the plant.

Because MPC controllers employ a receding control philosophy, a sequence of control actions $[U(k)]$ is computed at each sample time, but only the first is applied and the procedure is repeated again in the next sample, the actual control action is given by Eq. (14). Therefore we can define the gains K_e and K_u as

$$K_e=\mathcal{P}SR_{\mathcal{U}}^{-1}B_{\mathcal{U}}^{T}L_1L_2, \tag{25}$$

$$\begin{aligned}K_u={}&\mathcal{P}SR_{\mathcal{U}}^{-1}B_{\mathcal{U}}^{T}(L_1L_3+L_4)B_{\mathcal{U}}S^{-1}\\ &\times(I-I_{U\times U}\mathcal{P}z^{-1}).\end{aligned} \tag{26}$$

If there are amplitude and rate constraints we have to apply the condition (17). In this case the structure of the controller is shown in Fig. 3. The main characteristic of this structure is the presence of a feedback path around the constraints. As seen from comparison of Figs. 3 and 4, the resulting controller structure of MPC shows strong resemblance to the classical anti-wind-up feedback structure [4].

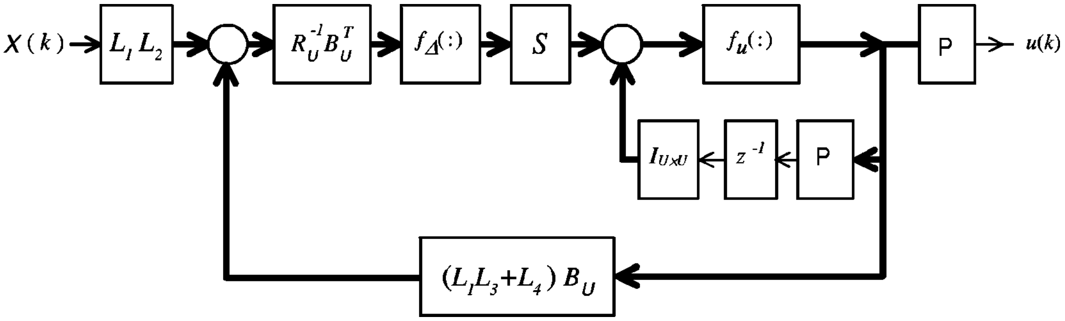

Fig. 3. Structure of the standard MPC controller.

4. Simulations and results

We present two examples that illustrate the effectiveness of the proposed anti-wind-up approach. Furthermore, we will illustrate the capabilities in comparison to the solution that have appeared in the literature.

Example 1. *Consider the following example studied by Doyle* et al. [16], *which is somewhat artificial and it was constructed to expose the limitations of existing anti-wind-up schemes. The plant model comprises two second-order Butterwort filter in series*

$$Gp(s) = 0.2 \frac{s^2 + 2\varsigma_1 \omega_1 s + \omega_1^2}{s^2 + 2\varsigma_1 \omega_2 s + \omega_2^2} \frac{s^2 + 2\varsigma_2 \omega_1 s + \omega_1^2}{s^2 + 2\varsigma_2 \omega_2 s + \omega_2^2}, \tag{27}$$

where $\omega_1 = 0.2115$, $\omega_2 = 0.0473$, $\varsigma_1 = 0.3827$, and $\varsigma_2 = 0.9239$. In addition, this problem still requires the following definitions: (i) The adopted sampling time is $T_S = 1.0$ sec, and assuming a zero-order holder for converting the transfer function (27) to $\mathcal{Z}$ domain, (ii) the convolution length was fixed $N = 150$ (this quantity guarantees a 98% of the total response), (iii) the prediction horizon V was set equal to the convolution length (V=N). Finally, the predictive horizon (U) was chosen using the principal component analysis [17] such that the condition number of matrix gain has a condition number $c = 1500$ with $w_u = 0$, obtaining $V = 10$. The closed-loop response and the associated (unconstrained) control action are given in Figs. 5 and 6, respectively. In Doyle et al. [16], *the effect the input signal is bounded within the interval ±1 with an amplitude saturation*

$$f(u(k)) = \begin{cases} 1, & u(k) \geqslant 1 \\ u(k), & -1 < u(k) < 1 \\ -1, & u(k) \leqslant -1. \end{cases}$$

The resulting closed-loop response is shown in Fig. 5. With no anti-wind-up compensation, the closed-loop response degenerates to a limit cycle of large amplitude. Using a conventional anti-wind-up approach [16], the response exhibits a large overshoot and large settling time. That is,

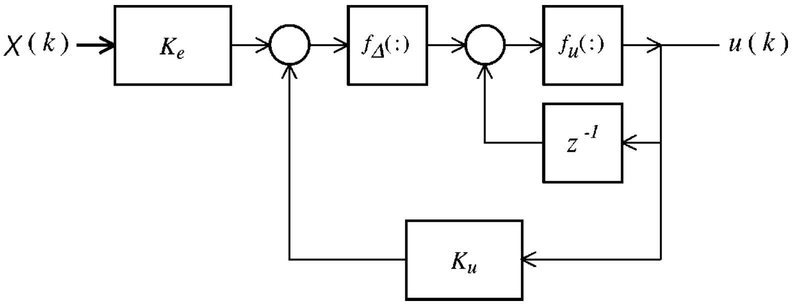

Fig. 4. Classical feedback structure with anti-wind-up.

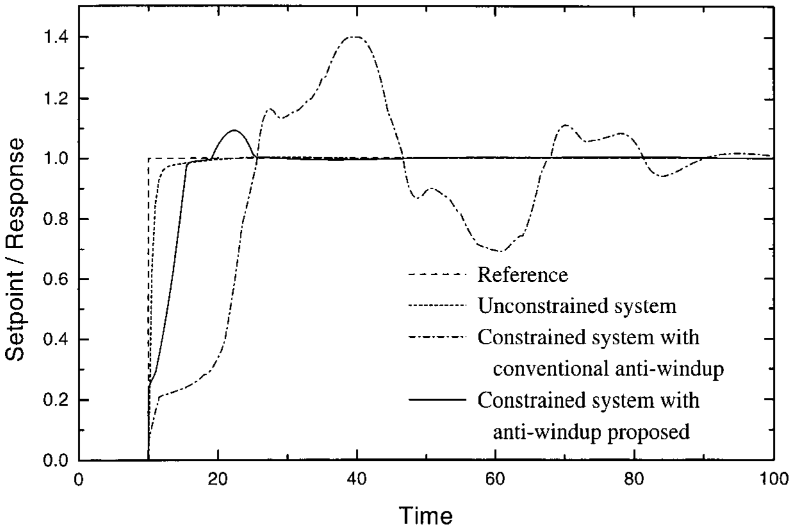

Fig. 5. Closed-loop responses of the linear system (27) for a setpoint change showing the effects of amplitude saturation and anti-wind-up schemes.

the closed-loop performance exhibits a significant deterioration, in comparison with the unconstrained response, due to the poor compensation. The closed-loop response obtained using the structure proposed earlier has a better performance than the others. It only exhibits a small overshoot due to the fact that the closed-loop system becomes temporarily uncontrollable.

Example 2. *Now, let us consider a heat exchanger, whose hot outlet temperature is controlled by manipulating the cold stream flow rate, modeled by [10]*

$$Gp(s) = -\frac{35.41}{(4.5s+1)^5}. \tag{28}$$

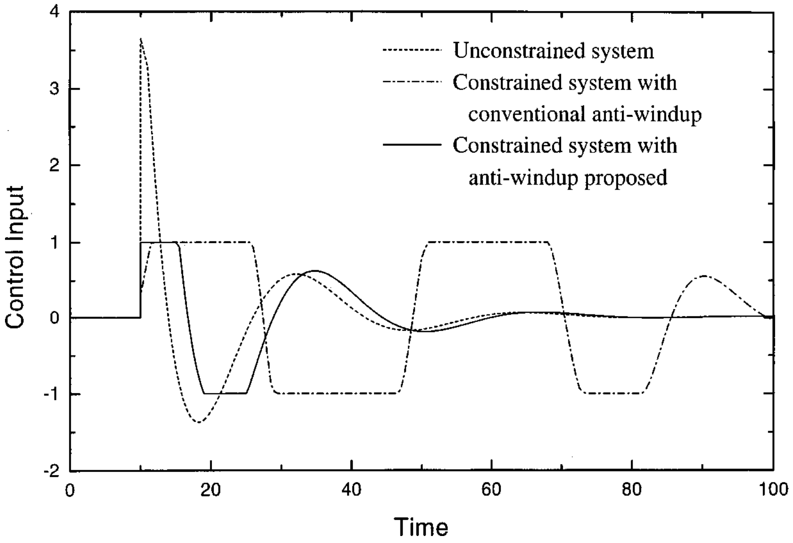

Fig. 6. Control variables corresponding to closed-loop responses shown in Fig. 5.

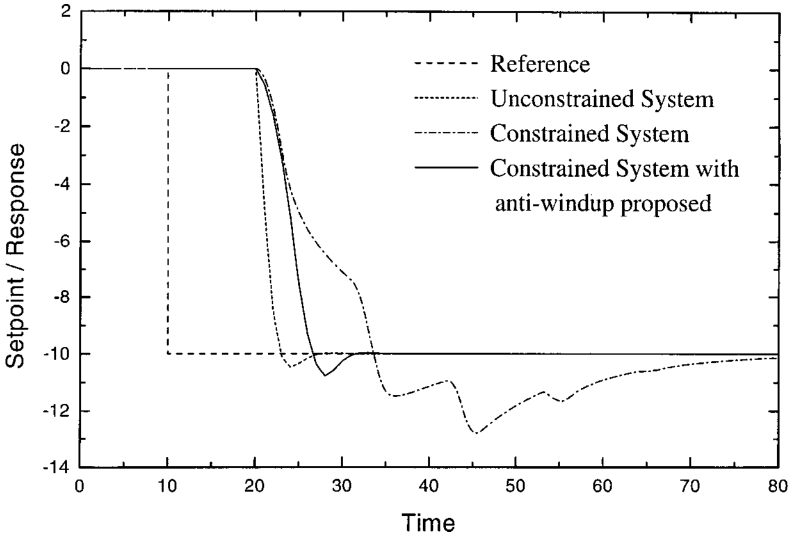

Fig. 7. Closed-loop responses of the linear system heat exchanger for a setpoint change, showing the effect of the amplitude saturation and the anti-wind-up scheme.

The system includes an amplitude

$$f_u(u(k))=\begin{cases}1, & u(k)\geqslant 1\\ u(k), & 0<u(k)<1\\ 0, & u(k)\leqslant 0,\end{cases} \tag{29}$$

and a rate saturation

$$f_\Delta(\Delta u(k))=\begin{cases}+0.2, & \Delta u(k)\geqslant +0.2\\ \Delta u(k), & -0.2<\Delta u(k)<+0.2\\ -0.2, & \Delta u(k)\leqslant -0.2,\end{cases} \tag{30}$$

due to the bounded amplitude and rate actuation of the valve. The predictive controller was designed following the tuning procedure developed by Rahul and Cooper [12]. The transfer function (28) was approximated through a first order plus a time delay model,

$$\tilde{G}p(s)=-35.41\frac{e^{-10\ s}}{12.04\ s+1}, \tag{31}$$

which was discretized with a sampling time $T_S=1$ sec, and the convolution length was fixed in 65

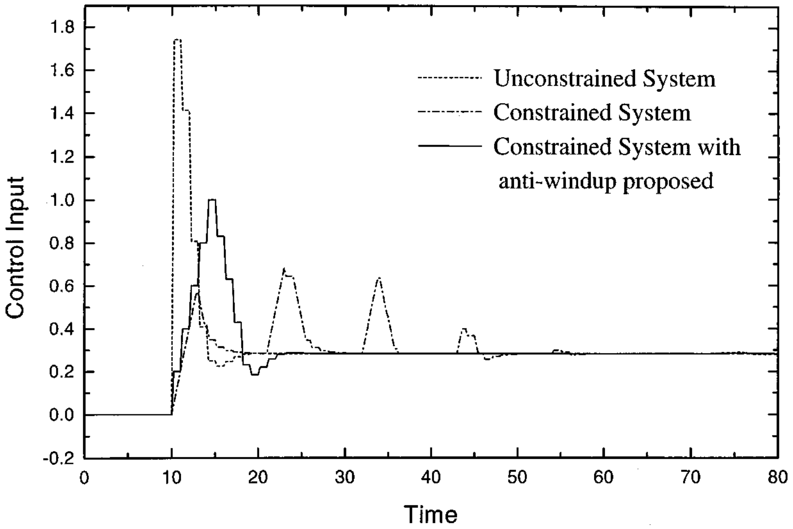

Fig. 8. Control variable corresponding to the closed-loop responses shown in Fig. 7.

Table 1
Computational load vs performance costs.

Controller	Performance	MFLOPS	Simulation times
DMC	175.32	1.92	5 sec
QDMC	154.87	574.7	300 sec
Proposed approach	155.14	1.92	5 sec

terms. The prediction horizon V has the length as the convolution model (V=N) and the control horizon was fixed in five samples. Finally, the control weights w_u was computed using the following formula [12], assuming a condition number c=500:

$$w_u=\frac{M}{c}\left(3.5\frac{\tau}{T_S}+2-\frac{M-1}{2}\right)K_p=14.92. \tag{32}$$

The unconstrained closed-loop response is shown in Fig. 7. We can see the response of the closed-loop system. However, when the manipulated variable has a rate and amplitude saturation (due to physical actuator's limitations), the closed-loop response shows a significant overshoot and an increase in its settling time. When the anti-wind-up compensation is added, the closed-loop performance is improved and it recovers its original characteristic. The difference in the closed-loop performance between both system (with and without anti-wind-up scheme) is due to the use of the amplitude and rate constraints for the correct updating of the past control actions. Fig. 8 shows the control actions computed by all the systems.

Table 1 compares the computational effort and control performance for different MPC controllers considered in the specialized literature. In this work we consider the QDMC [1], which performs an on-line optimization. While we have included representative simulations time for relative comparison, the experiments were performed with several programs developed in Matlab. The codes were not optimized for efficient implementation. Clearly, a substantial speed up in time can be achieved by the proposed approach with a small deterioration of the closed-loop performance.

In a future work we will address the robustness problem and the effect of uncertainties on the performance. However, in this example we include two simulations where a gain mismatch is considered. We assume that the process gain varies 20% ($-28.3 \leqslant Kp \leqslant -42.5$). In Fig. 9 we show the response obtained for the constrained system with this variation in the gain. We can clearly see that the proposed approach provided a good performance in spite of the important deviation in the gain process. Fig. 10 summarizes the results obtained during the simulations of system (28) with and without uncertainty by showing the ISE index obtained in each case.

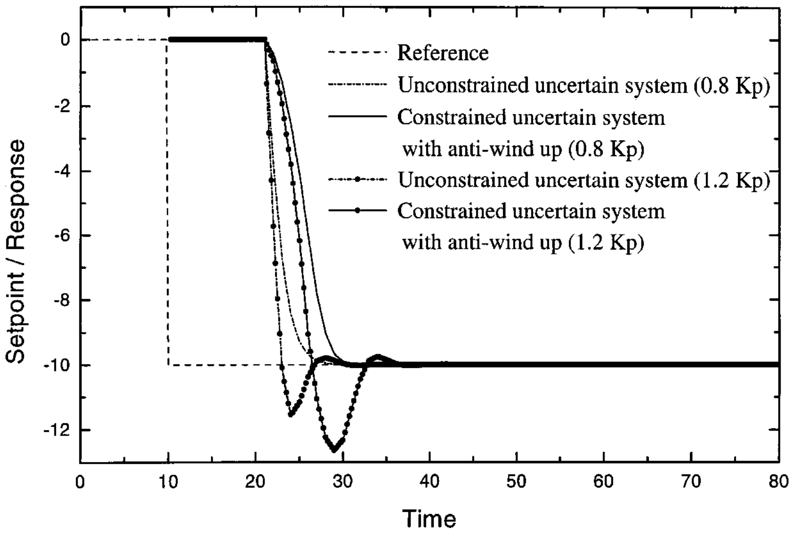

Fig. 9. ISE indices obtained in the simulations of system (28) with and without uncertainty, including the proposed anti-wind-up scheme.

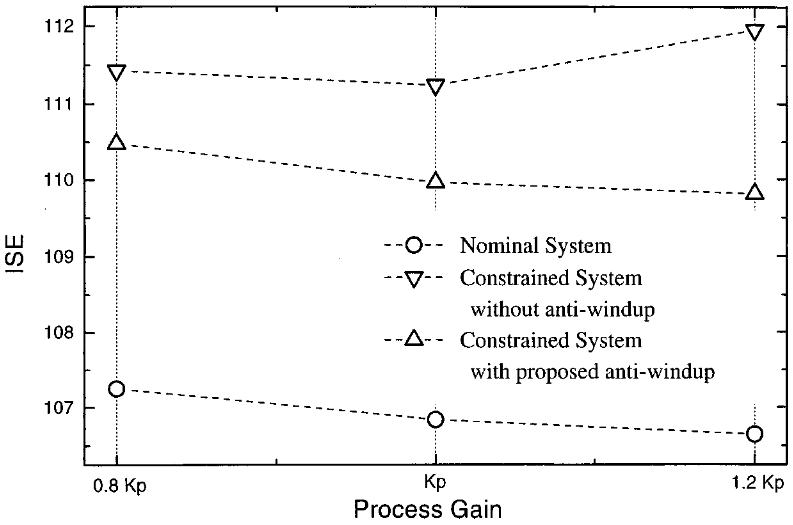

Fig. 10. Closed-loop responses of the linear system (28) with uncertain in the process gain.

5. Conclusions

In this work we solve the simultaneous actuator amplitude and rate saturation MPC problem by formulating it as an equivalent optimal control problem. Furthermore, we established a direct relationship between AWBT schemes and MPC with quadratic objective, input constrains and plant model structure that are affine in the input variables. The key to solving the problem and to establishing that relationship was the application of optimality conditions and Pontryagin's minimum principle to the optimization problem solved by MPC. The proposed framework employs saturation models as part of the controller architecture to ensure that no rate and amplitude commands are sent to the actuators that exceed their specific limits. The effectiveness of the proposed approach was illustrated by two numerical examples.

In spite of the results obtained in this work, several questions about stability and robustness issues still remain open as future research topics. A future work must include a stability analysis of the constrained closed-loop system, which leads to stability criterion to tune the controller parameters and a sensitivity analysis of the control structure presented in this work.

Appendix A: Extended system model

Given a linear discrete system,

$$x(k+1)=Ax(k)+Bu(k), \quad x(0)=x^0,$$

$$y(k)=Cx(k), \tag{33}$$

and the setpoint is described by the following linear dynamic system:

$$x_r(k+1)=A_r x_r(k)+B_r u_r(k), \quad x_r(0)=x_r^0,$$

$$r(k)=C_r x_r(k), \tag{34}$$

the system error $e(k)$ is given by

$$x(k+1)=Ax(k)+Bu(k), \quad x(0)=x^0,$$

$$x_r(k+1)=A_r x_r(k)+B_r u_r(k), \quad x_r(0)=x_r^0, \tag{35}$$

$$e(k)=r(k)-y(k).$$

Expressing the manipulated variable in incremental way and defining the extended states,

$$\mathcal{X}(k)=[x(k) \quad u(k-1) \quad x_r(k)]^T,$$

the system's manipulated variables,

$$\mathcal{U}(k)=\Delta u(k), \quad \mathcal{R}(k)=u_r(k),$$

and the system's matrices,

$$A_{\mathcal{X}}=\begin{bmatrix} A & B & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & 0 & A_r \end{bmatrix}, \quad B_{\mathcal{U}}=\begin{bmatrix} B \\ 1 \\ \mathbf{0} \end{bmatrix},$$

$$B_{\mathcal{R}}=\begin{bmatrix} 0 \\ 0 \\ B_r \end{bmatrix}, \quad C_{\mathcal{X}}=[-C \quad 0 \quad C_r], \qquad (36)$$

the extended system (35) can be written as the following:

$$\mathcal{X}(k+1)=A_{\mathcal{X}}\mathcal{X}(k)+B_{\mathcal{U}}\mathcal{U}(k)+B_{\mathcal{R}}\mathcal{R}(k),$$

$$e(k)=C_{\mathcal{X}}\mathcal{X}(k). \qquad (37)$$

For system (37), the following well-known results hold.

Lemma 1. *If the pair (A,B) and (A_r,B_r) are reachable, then the pair ($A_{\mathcal{X}}$,$B_{\mathcal{U}}$) is stabilizable with nonreachable eigenvalues only at the origin*.

Lemma 2. *If the pair (A,C) and (A_r,C_r) are observable and system (33) does not possess transmission zeros at one, then the pair ($A_{\mathcal{X}}$,$C_{\mathcal{X}}$) is observable*.

Appendix B: Weighting matrix

The first term of objective function (4),

$$\sum_{i=U}^{V} \mathcal{X}^T(k+i)Q\mathcal{X}(k+i), \qquad (38)$$

measures the system behavior outside the control horizon. It acts like a Lyapunov function, such that if it guarantees a reduction in the changes of system states $[\Delta\mathcal{X}(k+i)<0,\ U\leqslant i\leqslant V]$, the closed loop will be stable. Since there are no control actions for $i\geqslant U$, the behavior of the system states outside the control horizon can be described as a function of $\mathcal{X}(k+U)$, and $\mathcal{U}(k+U)$,

$$\mathcal{X}(k+i)=A_{\mathcal{X}}^{i-U}\mathcal{X}(k+U) + \sum_{l=0}^{i-U-1} A_{\mathcal{X}}^{l}B_{\mathcal{U}}\mathcal{U}(k+i-l),$$

$$i=U+1,U+2,\ldots,V. \qquad (39)$$

Replacing these states in Eq. (38) we have

$$\sum_{i=U}^{V} \mathcal{X}^T(k+i)Q\mathcal{X}(k+i)$$

$$=\mathcal{X}^T(k+U)\sum_{i=U}^{V} A_{\mathcal{X}}^{i-U^T}QA_{\mathcal{X}}^{i-U}\mathcal{X}(k+U)$$

$$+\mathcal{U}^T(k+U)\sum_{i=U}^{V} B_{\mathcal{U}}^T\left[\sum_{l=0}^{i-U\text{-}1} A_{\mathcal{X}}^{l\ T}\right]$$

$$\times Q\left[\sum_{l=0}^{i-U\text{-}1} A_{\mathcal{X}}^{l}\right]B_{\mathcal{U}}\mathcal{U}(k+U). \qquad (40)$$

Defining the following matrices:

$$S_{\mathcal{X}}=\left[\sum_{i=0}^{V-U} A_{\mathcal{X}}^{i\ T}\right]Q\left[\sum_{i=0}^{V-U} A_{\mathcal{X}}^{i}\right], \qquad (41a)$$

$$S_{\mathcal{U}}=\sum_{i=U}^{V} B_{\mathcal{U}}^T\left[\sum_{l=0}^{i-U-1} A_{\mathcal{X}}^{l\ T}\right]Q\left[\sum_{l=0}^{i-U-1} A_{\mathcal{X}}^{l}\right]B_{\mathcal{U}}, \qquad (41b)$$

we can write Eq. (40) as the following:

$$\sum_{i=U}^{V} \mathcal{X}^T(k+i)Q\mathcal{X}(k+i)$$

$$=\mathcal{X}^T(k+U)S_{\mathcal{X}}\mathcal{X}(k+U)$$

$$+\mathcal{U}^T(k+U)S_{\mathcal{U}}\mathcal{U}(k+U). \qquad (42)$$

Because of the control action $\mathcal{U}(k+U)$ is given,

$$\mathcal{U}(k+U)=\sum_{i=0}^{U-1} \mathcal{U}(k+i),$$

the last expression becomes

$$\sum_{i=U}^{V} \mathcal{X}^T(k+i)Q\mathcal{X}(k+i)$$

$$=\mathcal{X}^T(k+U)S_{\mathcal{X}}\mathcal{X}(k+U)$$

$$+\sum_{i=0}^{U-1} \mathcal{U}^T(k+i)S_{\mathcal{U}}\mathcal{U}(k+i). \qquad (43)$$

Observe that Eq. (43) not only affects the behavior of the states, but also influences the control actions employed to control the system. So, modifying the prediction horizon V, or the control horizon U, we modify $S_{\mathcal{X}}$ [penalizing, or not, $\Delta\mathcal{X}(k+U)$] and $S_{\mathcal{U}}$ (penalizing, or not, the energy employed to control the system). Furthermore, we must note that the matrices $S_{\mathcal{X}}$ has the same form as the solution of the algebraic Ricatti equation [15]. The matrix $S_{\mathcal{U}}$ is related with $S_{\mathcal{X}}$ through the following expression:

$$S_{\mathcal{U}}=B_{\mathcal{U}}^{T}\sum_{j=U}^{V} S_{\mathcal{X}}(j)B_{\mathcal{U}},$$

where $S_{\mathcal{X}}(j)$ is given by Eq. (41a) with the upper limit of sumatory given by j instead of $V-U$.

References

[1] Garcı́a, C. E., Prett, D. M., and Morari, M., Model predictive control: Theory and practice—a survey. Automatica **25**, 335–348 (1989).

[2] Edwards, C. and Postlethwaite, I., Anti-wind up and bumpless transfer schemes. Automatica **34**, 199–210 (1998).

[3] Giovanini, L. and Marchetti, J., Flexible Structure Control—A Practical Solution to Manipulated Constrains. International Symposium on Advanced Control in Chemical Process (AdChem 2000), Vol. 2, Pisa, Italy, pp. 743–748.

[4] Kothare, M., Campo, P., Morari, M., and Nett, C., A unified framework for the study of anti-wind up designs. Automatica **30**, 1869–1883 (1994).

[5] Kothare, M., Balakrishanan, V., and Morari, M., Robust constrained model predictive control using linear matrix inequalities. Automatica **32**, 1–34 (1996).

[6] Badgwell, T., Robust Stability Conditions for a SISO Model Predictive Control Algorithm. Automatica **33**, 355–380 (1998).

[7] Kouvaritakis, B., Rossiter, J., and Cannon, M., Linear quadratic feasible predictive control. Automatica **34**, 1583–1592 (1998).

[8] Rawlings, J. and Muske, K., The stability of constrained receding horizon control. IEEE Trans. Autom. Control **38**, 1512–1516 (1993).

[9] Rossiter, J., Kouvaritakis, B., and Rice, M., A numerically robust state-space approach to stable-predictive control strategies. Automatica **34**, 65–73 (1998).

[10] Giovanini, L., Robust Predictive Control with Constrains. Ph.D. dissertation, Santa Fe, Argentina, 2000.

[11] Maurath, P., Seborg, D., and Mellichamp, D., Predictive controller design for single input/single output systems. Ind. Eng. Chem. Res. **27**, 956–970 (1989).

[12] Rahul, S. and Cooper, D., A tuning strategy for unconstrained SISO model predictive control. Ind. Eng. Chem. Res. **36**, 729–746 (1997).

[13] Geromel, P., Peres, J., and Bernussou, J., On a convex parameter space method for linear control design of uncertain systems. SIAM J. Control Optim. **29**, 381–402 (1991).

[14] Bryson, A. and Ho, Y., Applied Optimal Control. Hemisphere Publishing, New York, 1975.

[15] Lewis, F., Optimal Control. John Wiley and Sons, New York, 1986.

[16] Doyle, J., Smith, R., and Enns, D., Control of Plants with Input Saturation Nonlinearities. In 32th IEEE Conference on Decision and Control, Minneapolis, MN, 1987, pp. 1034–1039.

[17] Maurath, P., Laub, A., Seborg, D., and Mellichamp, D., Predictive controller design by principal component analysis. Ind. Eng. Chem. Res. **27**, 1204–1212 (1988).

Leonardo L. Giovanini was born in Argentina in 1969. He received the B.S. degree in electronic engineering from the Universidad Tecnologica Nacional (UTN), Regional School at Villa Maria in 1995, and the Ph.D. degree in Engineering Science from the Universidad Nacional del Litoral, School of Water Resource in 2000. Since then he has been Lecturer in the Department of Electronic Engineering at UTN, Regional School at Villa Maria. Actually, he is working as research fellow in the Industrial Control Center, University of Strathclyde. His research interests include predictive control for nonlinear system, fault tolerant control, and distributed control systems.