Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Repetitive model based predictive controller to reject periodic disturbances. ⋆

Daniel M. Cruz * **Julio E. Normey-Rico** *
Ramon Costa-Castelló **

* *Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, 88040-900, Florianópolis, SC, Brazil (e-mail: danielcruz@das.ufsc.br, julio@das.ufsc.br)*
** *Departament d'Enginyeria de Sistemes, Automàtica i Informàtica Industrial, Universitat Politècnica de Catalunya, 08028, Barcelona, Spain (e-mail: ramon.costa@upc.edu)*

Abstract: This paper presents a repetitive model predictive controller to reject periodic disturbances in industrial processes. The proposed technique uses a state-space model with embedded repetitive action to integrate most important characteristics from both repetitive and predictive controllers. Thus, the obtained control strategy combines best characteristics from both repetitive and model predictive control. A simulation case study is presented to discuss several aspects related to disturbance rejection performance, parameters tuning and constraints effects, and to compare this new control strategy with traditional repetitive and model predictive controllers.

Keywords: Periodic Signals, Repetitive Control, Model Predictive Control, Constraints.

1. INTRODUCTION

Many processes are subject to periodic disturbances or references, some examples are: tubular heat exchangers [Álvarez et al., 2007], reverse osmosis desalination [Emad et al., 2012], casting processes [Manayathara et al., 1996] or olive oil mills [Bordons and Cueli, 2004] among others. Repetitive Control (RC) [Li Cuiyan, 2004] has been extensively used to deal with periodic references/disturbances. Furthermore, RC is a technique which is closely related to other strategies like Iterative Learning Control (ILC) and Run-to-Run (R2R) [Wang et al., 2009], which have been successfully used in different types of process control [Xu et al., 2009].

As an Internal Model Principle (IMP) - based strategy (see Francis and Wonham [1976]), RC uses an Internal Model (IM) that, guarantees tracking/rejection capabilities of periodic references/disturbances. This IM provides infinite or very high gain at a given frequency and its harmonics. In order to guarantee closed-loop stability usually a plug-in architecture is used combined with a phase-cancellation technique. Although this formulation is simple to implement (reduced computation burden and number of tuning parameters). In practice it is very restrictive in the closed-loop behavior than can be achieved.

Additionally, it is well known that, in systems with actuator saturation, a controller with these characteristics may produce a wind-up effect in which the states of the controller can grow unbounded [Hippe, 2010]. Even if the gain is not infinite but high, the states can overgrow significantly making harder to recover the system to the linear behavior. In order to overcome this problem several anti-windup schemes have been proposed.

In order to preserve steady-state RC performance, improve time response and handle actuator limitation properly in this work combining repetitive control and Model Predictive Control (MPC) is proposed.

MPC is a control technique which uses internally the plant model to compute the system output predictions and determines the optimal input according to a certain cost function [Camacho and Bordons, 2004]. One of the nicest properties of MPC is that it can be tuned by using very simple and intuitive parameters. Another very important property of MPC is its capability to handle constrains both in the output and control action. Among others these constraint handling capabilities implies that no anti-windup scheme is required.

In order to guarantee null steady-state error in front of constant (o piecewise constant) signals (references or disturbances) most MPC schemes introduce an integrator to augment the plant model (incremental control), this can be obtained if closed-loop stability is ensured. This integrator can be thought as the step IM, similarly if the control system is dealing with periodical signals an IM for periodical signals should be used. Combining RC and MPC has been previously introduced in the field of surgery [Natarajan and Lee, 2000, Ginhoux et al., 2005]. A more formal introduction can be found in the state-space domain in Lee et al. [2001], Gupta and Lee [2006]. Recently these concepts have been extended to multivariable systems [Wang et al., 2012a] and systems with constrains [Wang et al., 2012b, 2013]. Different from these works, in this paper we propose a new and simple

⋆ This work was supported in part by CNPq - Brasil.

Copyright © 2014 IFAC

methodology to introduce an IM in MPC to reject/track periodical signals.

To present this new controller this paper is organized as follows, section 2 reviews most relevant concepts on MPC and RC, in sections 3 and 4 the proposed architecture is explained in detail, section 5 analysis proposed controller through a case study and finally section 6 contains the conclusions.

2. MPC AND RC REVIEW

This section presents the principals ideas behind MPC and RC and the mathematical background needed to develop the proposed controller.

2.1 MPC Review

MPC algorithms use a model of the process to obtain the control signal by minimizing an objective function. The principal ideas of MPC are: (i) explicit use of a model to predict the process output in a certain time horizon; (ii) calculation of a set of future control actions minimizing an objective function and (iii) receding strategy, where only the first computed control action is applied and the horizon is displaced towards the future [Camacho and Bordons, 2004].

Objective Function The following quadratic cost function is proposed:

$$J(k) = \sum_{j=1}^{N_y} [\hat{y}(k+j|k) - w(k+j)]^2 + \sum_{j=1}^{N_u} \lambda[\Delta u(k+j-1)]^2 \quad (1)$$

where $\hat{y}(k+j|k)$ stands for the output predictions computed at $k+j$ with information available up to time k and $w(k+j)$ is the future reference, $\Delta u(k+j-1)$ is a future control increment sequence obtained from cost function minimization, N_y define the prediction horizon, N_u is the control horizon, λ is the control weighting.
In compact matrix form:

$$J = (\hat{Y} - W)^T(\hat{Y} - W) + \Delta U^T R \Delta U \quad (2)$$

where

$$\begin{aligned} \Delta U &= [\Delta u(k)\ \Delta u(k+1)\ \Delta u(k+2)\ \dots\ \Delta u(k+N_u-1)]^T \\ \hat{Y} &= [\hat{y}(k+1|k)\ \hat{y}(k+2|k)\ \hat{y}(k+3|k)\ \dots\ \hat{y}(k+N_y|k)]^T \end{aligned}$$

and $R = \lambda I_{Nu}$ is the weighting matrix with $\lambda > 0$ [Camacho and Bordons, 2004].
The parameters N_y, N_u, λ are the tunning parameters.

Plant Model and Predictions To relate the predictions to the future control actions a linear state-space model is used to compute the predictions:

$$\begin{aligned} \tilde{\mathbf{x}}(k+1) &= \tilde{A}\tilde{\mathbf{x}}(k) + \tilde{B}u(k) \\ y(k) &= \tilde{C}\tilde{\mathbf{x}}(k) \end{aligned} \quad (3)$$

where $\tilde{\mathbf{x}} \in \mathbb{R}^n$ is the state, $y \in \mathbb{R}$ is the output, $u \in \mathbb{R}$ is the control input, and $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ are matrices of appropriate dimensions. In order to obtain an off-set-free controller an incremental model can be obtained using:

$$\begin{aligned} \overbrace{\begin{bmatrix} \Delta\tilde{\mathbf{x}}(k+1) \\ y(k+1) \end{bmatrix}}^{\mathbf{x}(k+1)} &= \overbrace{\begin{bmatrix} \tilde{A} & 0 \\ \tilde{C}\tilde{A} & I \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} \Delta\tilde{\mathbf{x}}(k) \\ y(k) \end{bmatrix}}^{\mathbf{x}(k)} + \overbrace{\begin{bmatrix} \tilde{B} \\ \tilde{C}\tilde{B} \end{bmatrix}}^{B} \Delta u(k) \\ y(k) &= \overbrace{\begin{bmatrix} 0' & I \end{bmatrix}}^{C} \mathbf{x}(k) \end{aligned} \quad (4)$$

with $\mathbf{x}(k)$ being the augmented state and $\Delta u(k) = u(k) - u(k-1)$.
Using the solution of the state-space system, $\hat{y}(k+j|k)$ can be written as [Camacho and Bordons, 2004]:

$$\hat{y}(k+j|k) = CA^j\mathbf{x}(k) + C\sum_{l=1}^{j} A^{j-l}B\Delta u(k-1+l) \quad (5)$$

that is:

$$\begin{aligned} \hat{y}(k+1|k) &= CAx(k) + CB\Delta u(k|k) \\ \hat{y}(k+2|k) &= CAx(k+1|k) + CB\Delta u(k+1|k) \\ &= CA^2x(k) + CAB\Delta u(k|k) + CB\Delta u(k+1|k) \\ &\vdots \\ \hat{y}(k+N_y|k) &= CA^{N_y}x(k) + CA^{N_y-1}B\Delta u(k|k) + \dots \\ &+ CA^{N_y-N_u}B\Delta u(k+N_y-1|k). \end{aligned} \quad (6)$$

in matrix form

$$\hat{Y} = G\Delta U + fx(k) \quad (7)$$

where

$$G = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_y-1}B & CA^{N_y-2}B & CA^{N_y-3}B & \dots & CA^{N_y-N_u}B \end{bmatrix} \quad f = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_y} \end{bmatrix}$$

Obtaining the Control Law In order to obtain future control sequence the prediction model (7) is substituted into the objective function (2). Giving:

$$J = \Delta U^T H \Delta U + 2F^T\Delta U + J_0 \quad (8)$$

where

$$\begin{aligned} H &= G^TG + R, \\ F &= G^T(fx(k) - W), \\ J_0 &= (fx(k) - W)^T(fx(k) - W). \end{aligned}$$

The unconstrained minimization solution is obtained making $\frac{\partial J}{\partial \Delta U} = 0$ giving:

$$\Delta U^* = -H^{-1}F = (G^TG + R)^{-1}G^T(W - fx(k)) \quad (9)$$

thus, because of the receding horizon strategy only the first element of ΔU^* is applied as control action.
Although in presence of constraints the minimization procedure needs to be done using a iterative method that computes, on each sample time, the minimal control action for the system.
In real applications constraints have to be considered in the optimization, typically in the control action amplitude, control action slew rate (variation) and process output:

$$\begin{aligned} u_{\min} \leq &\quad u(k) &\leq u_{\max}; \quad \forall k \geq 0, \\ I_{\min} \leq &\ u(k) - u(k-1) &\leq I_{\max}; \quad \forall k \geq 0, \\ Y_{\min} \leq &\quad y(k) &\leq Y_{\max}; \quad \forall k \geq 0. \end{aligned}$$

That can be re-written in a compact form as a linear inequality in the vector of future increments of the control action [Camacho and Bordons, 2004]. With this analysis the MPC solution is obtained by means of a quadratic program optimization which is efficiently solved.

2.2 RC

In a plug-in scheme, the repetitive controller is attached to a control loop defined by an existing nominal controller $G_c(z)$ and the plant, $G_p(z)$. Figure 1 shows the complete architecture.
The repetitive controller is composed by two elements, the

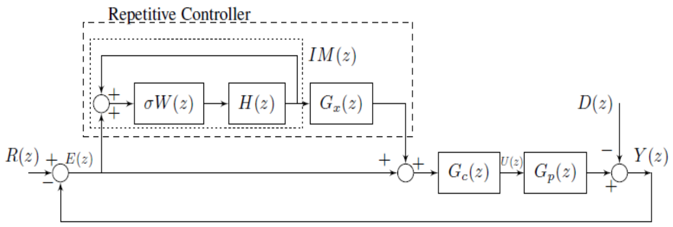

Fig. 1. Block diagram of plug-in scheme.

IM, $IM(z)$, and the stabilizing controller, $G_x(z)$. The IM is composed by a delay function $W(z)$, a real number $\sigma = \{-1, 1\}$ and a low-pass null-phase Finite-Impulse Response (FIR) filter, $H(z)$, in positive feedback connection.

In the plug-in architecture each element has a concrete role:

- The nominal controller, $G_c(z)$, is assumed to robustly stabilize the closed-loop system composed by the conventional controller and the plant : $T_o(z) = \frac{G_p(z)G_c(z)}{1+G_p(z)G_c(z)}$. It is also assumed that $G_c(z)$ allows to reject disturbances, $D(z)$, in a certain frequency spectrum.
- The delay function, $W(z)$, and the number σ are used to construct the IM. Depending on the values of $W(z)$ and σ different internal models can be constructed depending on the specific signal to deal with.
  In this work the generic N-periodic signal IM is used : $\frac{1}{z^N-1}$. It is obtained by placing $W(z) = z^{-N}$ and $\sigma = 1$. Including this element inside the system null steady-state error in front to N-periodic references or disturbances is guaranteed if the system is closed-loop stable.
- The low-pass filter $H(z)$ is used to reduce the gain introduced by the IM at high frequencies. The design of $H(z)$ is a trade-off between robustness and performance.
  Note that $H(z)$ is implemented in series connection with a delay function, as a consequence $H(z)$ can be selected to be non causal. Usually a low-pass null-phase FIR filter is used.
- The stabilizing controller role is to guarantee closed-loop stability.

The transfer function from $R(z)$ to $E(z)$ in the system from Figure 1 can be written as :

$$S_M(z) = S_o(z)S_M(z) \tag{10}$$

where $S_o(z) = \frac{1}{1+G_p(z)G_c(z)}$ corresponds to the sensitivity function of the system without the repetitive controller and $S_M(z) = \frac{1-\sigma W(z)H(z)}{1-\sigma W(z)H(x)(1-G_x(z)T_o(z))}$ is usually called the modifying sensitivity function.

The transfer function (10) is stable if the following two conditions are fulfilled:

- The closed-loop system, without repetitive controller must be stable. This is achieved with the nominal controller $G_c(z)$ design.
- The following inequality is fulfilled:
  $$\|\sigma W(z)H(x)\,(1 - G_x(z)T_o(z))\,\| < 1.$$
  Assuming $\|\sigma W(z)\| \leq 1$, this inequality can be forced by selecting $\|H(z)\| < 1$ an choosing and appropriate $G_x(z)$.
  A common way to design $G_x(z)$, guaranteeing the stability conditions, is selecting $G_x(z) = k_r T_o(z)^{-1}$. This approach is not appropriated for non-minimum-phase plants. In the generic case phase-cancellation techniques must be used [Tomizuka, 1987].

3. THE PROPOSED CONTROLLER

In this section a state-space model predictive controller with embedded repetitive control action is proposed. This scheme is based on a state-space model with embedded repetitive action to compute the system predictions and the control action.

3.1 State-Space Model With Embedded Repetitive Action

Taking the process model in state-space equations as in equation (3), it is necessary to make a few changes in this representation in order to add the repetitive action. These changes are based on the same ideas of the augmented model construction, equation (4), similarly to the development in Wang [2009].
Many MPC algorithms are based on the control deviation parameter ($\Delta u(k) = u(k) - u(k-1)$) to compute the control action. Thus the control system has a embedded integral action ($1/1-z^{-1}$). Considering the IMP and the

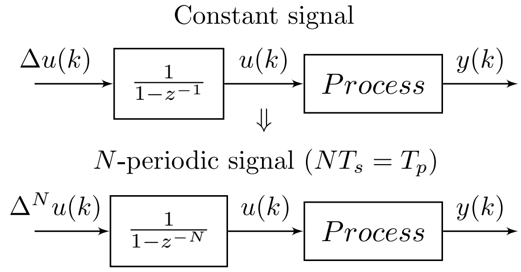

Fig. 2. Step and N-periodic internal models.

existence of periodic disturbances/references with period T_p, a repetitive loop needs to be inserted into the model as shown in figure 2. This IM augments the system with all poles necessary to reproduce desirable periodic signal and needs to be chose considering $T_p = NT_s$.
The following variables need to be defined for the state space model development

$$\Delta^N \tilde{\mathbf{x}}(k) = \tilde{\mathbf{x}}(k) - \tilde{\mathbf{x}}(k-N)$$
$$\Delta^N u(k) = u(k) - u(k-N)$$

where $\Delta^N \tilde{\mathbf{x}}(k)$ and $\Delta^N u(k)$ represent respectively state $\tilde{\mathbf{x}}(k)$ and control $u(k)$ increment from each period. Applying then on process model (3), we have the following equation

$$\Delta^N \tilde{\mathbf{x}}(k+1) = \tilde{A}\Delta^N \tilde{\mathbf{x}}(k) + \tilde{B}\Delta^N u(k) \tag{11}$$

where the input is the N-periodic variation of the control action. Therefore, the model needs to be connected to the process output making possible to express output predictions as function of the input $\Delta^N u(k)$. Using the expression:

$$\begin{aligned} &y(k+1) - y(k-N+1) = \tilde{C}(\tilde{\mathbf{x}}(k+1) - \tilde{\mathbf{x}}(k-N+1)) \\ &y(k+1) = \tilde{C}\Delta^N\tilde{\mathbf{x}}(k+1) + y(k-N+1) \\ &y(k+1) = \tilde{C}\tilde{A}\Delta^N\tilde{\mathbf{x}}(k) + \tilde{C}\tilde{B}\Delta^N u(k) + y(k-N+1) \end{aligned} \tag{12}$$

one can observe that the output $y(k+1)$ from the current period is related to the output $y(k-N+1)$ from the past period. Therefore, the vector

$$x(k) = [\Delta^N\tilde{\mathbf{x}}(k)^T \; y(k) \; y(k-1) \; y(k-2) \; \dots \; y(k-N+1) \; y(k-N)]^T$$

represents the state vector from the state-space model with embedded repetitive action, which can be described by the following set of equations:

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta^N u(k) \\ y(k) &= Cx(k) \end{aligned} \tag{13}$$

where:

$$A = \begin{bmatrix} \tilde{A} & 0 & 0 & \dots & 0 & 0 \\ \tilde{C}\tilde{A} & 0 & 0 & \dots & 1 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ & & & \ddots & & \\ 0 & 0 & 1 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}; B = \begin{bmatrix} \tilde{B} \\ \tilde{C}\tilde{B} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}; C = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \end{bmatrix}$$

With this model it is not only possible to make model predictions for controller design, but also do it with the advantage of a embedded repetitive action that reproduces N period signals.

3.2 Unconstrained Closed-loop Control System

The unconstrained control law, considering receding horizon strategy, is the first element of $\Delta^N U$ as the N incremental control, with (considering the reference equal within all prediction horizon)

$$\Delta^N u(k) = [1, 0, \dots, 0](G^T G + R)^{-1} G^T(\mathbf{1}r(k) - fx(k)) \tag{14}$$

$$= K_r r(k) - K_{rmpc} x(k) \tag{15}$$

where $K_r r(k)$ refers to the set-point change and $-K_{rmpc}x(k)$ to the state feedback gain.
Using the state-space model with the IM the closed-loop system is obtained as

$$\begin{aligned} x(k+1) &= Ax(k) - BK_{rmpc}(k) + BK_r r(k) \\ &= (A - BK_{rmpc})x(k) + BK_r r(k) \end{aligned} \tag{16}$$

The closed-loop system can be represented as shown in fig-

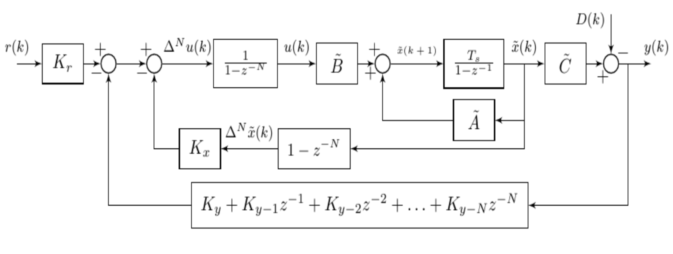

Fig. 3. Unconstrained closed-loop block diagram

ure 3 where the state-feedback gain K_{rmpc}, has been separated in two different gains: $[K_x]$ that refers to the process states period variation $\Delta^N\tilde{x}$, and $[K_y \; K_{y-1} \; K_{y-2} \dots K_{y-N}]$ that refers to the process outputs within past period.

4. CONSTRAINTS TREATMENT

In order to use QP optimization in the proposed controller MPC constrains must be written as a function of $\Delta^N u(k) = u(k) - u(k-N)$ and described as:

$$A_{ineq}\Delta^N u \leqslant b_{ineq} \tag{17}$$

Hereafter typical constraints are written like that.

4.1 Input Amplitude Constraint

These constraints are specified as

$$u_{min} \leqslant u(k) \leqslant u_{max}$$

The relation between $u(k)$ and $\Delta^N u(k)$ is

$$\begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N_u-1) \end{bmatrix} = \begin{bmatrix} u(k-N) \\ u(k-N+1) \\ u(k-N+2) \\ \vdots \\ u(k-N+N_u-1) \end{bmatrix} + \begin{bmatrix} \Delta^N u(k) \\ \Delta^N u(k+1) \\ \Delta^N u(k+2) \\ \vdots \\ \Delta^N u(k+N_u-1) \end{bmatrix}$$

in matrix form

$$U = U(k-N) + \Delta^N U \tag{18}$$

$$\begin{bmatrix} I \\ -I \end{bmatrix} \Delta^N U \leqslant \begin{bmatrix} U_{max} - U(k-N) \\ -U_{min} + U(k-N) \end{bmatrix} \tag{19}$$

4.2 Slew-rate constraint

This constraint restricts the control action variation in one sample time, represented by:

$$I_{min} \leqslant \Delta u(k) \leqslant I_{max}$$

In this case it is necessary to describe $\Delta u(k)$ using $\Delta^N u(k)$, which is the main variable for the minimization problem. With the equality:

$$\Delta u(k) = \Delta^N u(k) - \Delta^N u(k-1) + \Delta u(k-N) \tag{20}$$

the variables are related as

$$\Delta U = S_0 \Delta^N U - c_o \Delta^N u(k-1) + \Delta U(k-N) \tag{21}$$

where

$$S_0 = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ -I & I & 0 & \dots & 0 \\ 0 & -I & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & I \end{bmatrix}, c_0 = \begin{bmatrix} I \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

and

$$\begin{bmatrix} S_0 \\ -S_0 \end{bmatrix} \Delta^N U \leqslant \begin{bmatrix} I_{max} + c_o\Delta^N u(k-1) - \Delta U(k-N) \\ -I_{min} - c_o\Delta^N u(k-1) + \Delta U(k-N) \end{bmatrix} \tag{22}$$

4.3 Output Constraint

This constraint imposes a min-max boundary for the process output using the prediction model. Considering that

$$y_{min} \leqslant y(k) \leqslant y_{max}$$

and the prediction model (7) follows

$$Y_{min} \leqslant G\Delta^N U + fx(k) \leqslant Y_{max} \tag{23}$$

and, in matrix form

$$\begin{bmatrix} G \\ -G \end{bmatrix} \Delta^N U \leqslant \begin{bmatrix} Y_{max} - fx(k) \\ -Y_{min} + fx(k) \end{bmatrix} \tag{24}$$

4.4 Adding Constraints to the Optimization Problem

It is possible to combine all constraints (22, 19, 24) in one linear inequality (17)

$$\underbrace{\begin{bmatrix} S_0 \\ -S_0 \\ I \\ -I \\ G \\ -G \end{bmatrix}}_{A_{ineq}} \Delta^N U \leqslant \underbrace{\begin{bmatrix} I_{max} + c_o \Delta^N u(k-1) - \Delta U(k-N) \\ -I_{min} - c_o \Delta^N u(k-1) + \Delta U(k-N) \\ U_{max} - U(k-N) \\ -U_{min} + U(k-N) \\ Y_{max} - fx(k) \\ -Y_{min} + fx(k) \end{bmatrix}}_{b_{ineq}} \tag{25}$$

Having a quadratic objective function, the minimization problem becomes a classic quadratic programming problem

$$\min_{\Delta^N U} \ (\Delta^N U^T H \Delta^N U + 2F^T \Delta^N U + J_0) \\ \text{Subject to: } \ A_{ineq} \Delta^N U \leqslant b_{ineq} \tag{26}$$

This computation must be done on every sample time, as well as the update of the variables F and b_{ineq} using previous control action.

5. SIMULATION RESULTS

Considering the following scenario

- Process Model:
$$G(s) = \frac{16.152}{0.457s + 1} \ ; \ G(z) = \frac{0.0353}{z - 0.9978} \tag{27}$$
(discretization with zero order hold - zoh)
- Change in the set-point from 0 to 4 at $t = 1s$.
- Disturbance Signal:
$d(t) = 2sin(\frac{2\pi}{T_p}t) + 0.5sin(\frac{4\pi}{T_p}t) + 0.1sin(\frac{8\pi}{T_p}t)$,
applied on the output with $T_p = 0.25$ from $t = 0s$ to the end.
- Amount of samples $N = 250$, and the sample time is $T_s = T_p/N = 0.25/250 = 10^{-3}s$
- Controllers Parameters:
  - **RMPC**: $N_y = N = 250, \ N_u = N/25 = 10, \ R = 0.005$
  - **MPC**: $N_y = 100, \ N_u = 5, \ R = 5$
  - **RC Plug-in**: $N = 250, \ k_r = 0.7, \ G_c = \frac{1.8z-1.796}{z-1}$

The simulations are separated in three blocks. The first one is for the comparison between the proposed RMPC and a classical MPC. Second one, compares RMPC with the RC plug-in. And the last one, shows how the proposed controller can treat constraints and the effects of them in the system response.

5.1 Comparison between RMPC and MPC

Figure 4 shows the results of the comparison of RMPC and MPC tuned for the same transient response. As expected, only RMPC rejects the periodic disturbance in steady-state, while the settling time and overshoot are similar in the two responses.

5.2 Comparison between RMPC and RC plug-in

The comparative analysis between RMPC and RC plug-in focus in the transient response, as in this case both controllers reject the periodic disturbance in steady-state. Note that RC plug-in has bigger settling time and overshoot, figure 5, because the only tunning parameter k_r does not give enough degrees of freedom to tune the transient response.

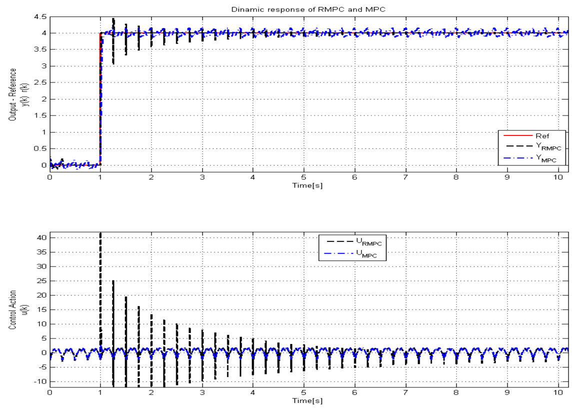

Fig. 4. System output (top) and control action (bottom) from both RMPC and MPC controllers

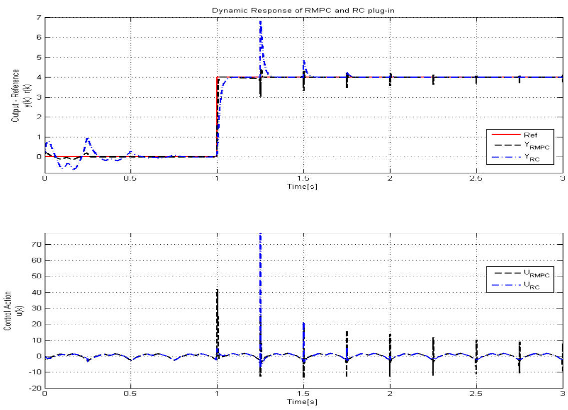

Fig. 5. System output (top) and control action (bottom) from both RMPC and RC plug-in controllers

5.3 Constrained RMPC

Previous sections shows how RMPC overperforms MPC and RC plug-in basic strategies as it has enough degrees of freedom to achieve both, a desired steady state and transient performance.
Hereafter the effect of constraints in RMPC performance is evaluated. Two cases are analyzed. In the first one, shown in figure 6, constraints in U are imposed. The limits for U are $U_{min} = -1$ and $U_{max} = 1$, which do not allow the control action to attempt the necessary steady-state amplitude, therefore the disturbance can not be completely rejected.

In the second case, illustrated in figure 7, only constraints in the slew rate are considered: $\Delta U_{min} = -0.15$ and $\Delta U_{max} = 0.15$. These constraints limit only the control action during the transient making the system to have a slow response but rejecting the disturbance in steady state. U and ΔU constraints actuate directly on the control action (amplitude and deviation) they impose limits to these variables that, looking with a repetitive point of view, could compromise the control system capability of tracking/rejecting some periodic signals.

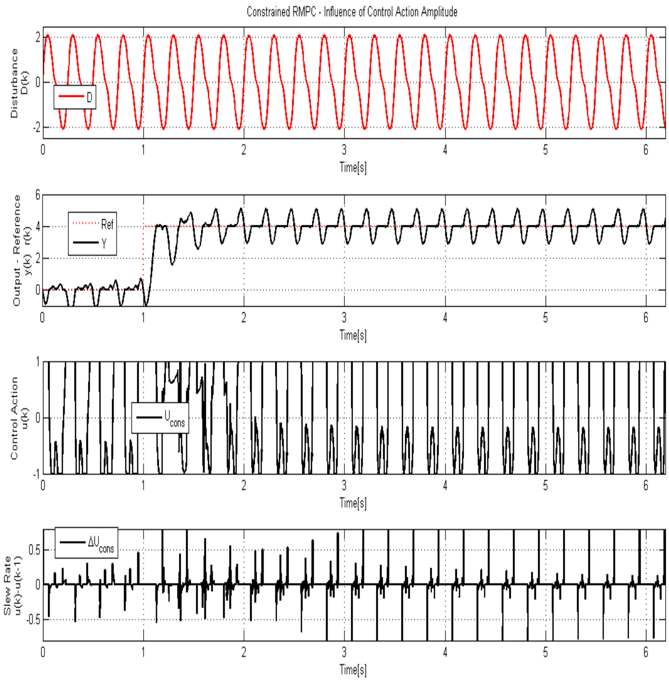

Fig. 6. Constrained RMPC dynamic response with U constraint active.

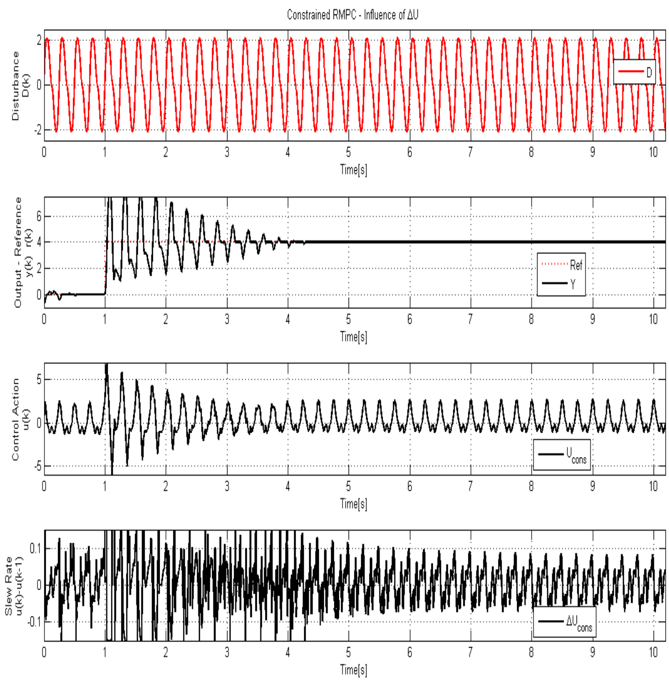

Fig. 7. Constrained RMPC dynamic response with ΔU constraint active.

6. CONCLUSIONS

The proposed RMPC controller combines the advantages of the MPC strategy such as constraint handling and flexible tunning with the repetitive control properties. This new control strategy has enough degrees of freedom to obtain desirable closed-loop disturbance rejection during the transient and in steady-state for periodic disturbances with constraints satisfaction. Simulations results have shown the advantages of RMPC over MPC and RC plug-in, therefore the proposed strategy seems to be an important tool to control process affected by periodic disturbances and constraints.

REFERENCES

J.D. Álvarez, L.J. Yebra, and M. Berenguel. Repetitive control of tubular heat exchangers. Journal of Process Control, 17:689 – 701, 2007.

C. Bordons and J.R. Cueli. Predictive controller with estimation of measurable disturbances. application to an olive oil mill. Journal of Process Control, 14:305 – 315, 2004.

E.F. Camacho and C. Bordons. Model Predictive Control. Springer-Verlag GmbH, 2004.

Ali Emad, A. Ajbar, and I. Almutaz. Periodic control of a reverse osmosis desalination process. Journal of Process Control, 22:218 – 227, 2012.

B.A. Francis and W.M. Wonham. Internal model principle in control theory. Automatica, 12:457–465, 1976.

R. Ginhoux, J. Gangloff, M. De Mathelin, L. Soler, M.M.A. Sanchez, and J. Marescaux. Active filtering of physiological motion in robotized surgery using predictive control. Robotics, IEEE Transactions on, 21(1):67–79, 2005.

Manish Gupta and Jay H. Lee. Period-robust repetitive model predictive control. Journal of Process Control, 16(6):545 – 555, 2006.

P. Hippe. Windup in Control: Its Effects and Their Prevention. Advances in Industrial Control. Springer, 2010.

Jay H. Lee, Seshatre Natarajan, and Kwang S. Lee. A model-based predictive control approach to repetitive control of continuous processes with periodic operations. Journal of Process Control, 11(2):195 – 207, 2001.

Zhuang Xianyi Li Cuiyan, Zhang Dongchun. A survey of repetitive control. In Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, volume 2, pages 1160 – 1166 vol.2, 2004.

T.J. Manayathara, Tsu-Chin Tsao, and J. Bentsman. Rejection of unknown periodic load disturbances in continuous steel casting process using learning repetitive control approach. IEEE Transactions on Control Systems Technology, 4:259–265, 1996.

Seshatre Natarajan and Jay H. Lee. Repetitive model predictive control applied to a simulated moving bed chromatography system. Computers & Chemical Engineering, 24(2–7):1127 – 1133, 2000.

Masayoshi Tomizuka. Zero phase error tracking algorithm for digital control. Journal of Dynamic Systems, Measurements and Control, 109:65–68, March 1987.

L. Wang. Model Predictive Control System Design and Implementation Using MATLAB (Advances in Industrial Control). Springer, 2009.

Liuping Wang, Shan Chai, E. Rogers, and C.T. Freeman. Multivariable repetitive-predictive controllers using frequency decomposition. Control Systems Technology, IEEE Transactions on, 20(6): 1597–1604, 2012a.

Liuping Wang, C.T. Freeman, Shan Chai, and E. Rogers. Experimentally validated repetitive-predictive control of a robot arm with constraints. In American Control Conference (ACC), 2012, pages 5495–5500, 2012b.

Liuping Wang, Chris T. Freeman, Shan Chai, and Eric Rogers. Predictive-repetitive control with constraints: From design to implementation. Journal of Process Control, 23:956 – 967, 2013.

Youqing Wang, Furong Gao, and Francis J. Doyle III. Survey on iterative learning control, repetitive control, and run-to-run control. Journal of Process Control, 19:1589 – 1600, 2009.

Jian-Xin Xu, Sanjib K. Panda, and Tong Heng Lee. Ilc for process temperature control: Application to a water-heating plant. In Real-time Iterative Learning Control, Advances in Industrial Control, pages 65–83. Springer London, 2009.