

# Fast MPC with Staircase Parametrization of the Inputs: Continuous Input Blocking

Marco Faroni

Dipartimento di Ingegneria  
Meccanica e Industriale  
University of Brescia  
Brescia, Italy

Email: m.faroni003@unibs.it

Manuel Beschi

Istituto di Tecnologie  
Industriali e Automazione  
National Research Council

Milano, Italy

Manuel Berenguel

Departamento de  
Informática  
University of Almería

Almería, Spain

Antonio Visioli

Dipartimento di Ingegneria  
Meccanica e Industriale  
University of Brescia

Brescia, Italy

Email: antonio.visioli@unibs.it

**Abstract**—In this paper we present a new method to reduce the computational complexity of model predictive control algorithms with online optimization. The formulation of the predictive equations is performed in the continuous-time domain, while the control inputs are parametrized as piecewise constant functions, with less steps than the control horizon. The continuous-time formulation permits the arbitrary choice of the prediction and control time-instants, disregarding the sampling period of the system, and this improves the goodness of the approximation. Moreover, since the inputs are forced to be piecewise constant, the resulting controller can be directly implemented in discrete-time. A tuning method for the choice of the prediction and control instants is proposed, minimizing the deviation with respect to the non-approximated controller. Numerical results show the effectiveness of this strategy against other methods with same reduction of computational complexity.

## I. INTRODUCTION

Although microprocessors are every day more and more powerful, the reduction of computational complexity of Model Predictive Control (MPC) algorithms is still an open topic for both the research and the industrial community. Quoting Y. Wang and S. Boyd, “a widely recognized shortcoming of model predictive control is that it can be used in applications with slow dynamics, where the sample time is measured in seconds or minutes” [1]. This is due to the high computational burden required to calculate the optimal control action. As a matter of fact, the optimal control inputs are computed by solving a Quadratic Program (QP). Then, in the spirit of receding horizon control, the first element of the input sequence is applied to the system, and the procedure is iterated at each time-step [2], [3]. The computational complexity of such algorithm may be prohibitive for small sampling periods and this limits the use of MPC in the control of fast-dynamic systems.

Several methods have been proposed to speed up the resolution of the online optimization [4]. Many of them focus on improving the efficiency of the QP solver, by exploiting the peculiar structure of the problem [5], [6]. Others approximate the constrained problem in order to quickly find a suboptimal solution [7], [8]. In [1] it is shown that a (very) approximated solution of the QP still leads to satisfactory

control performance, alleviating the required computational time. An alternative approach for complexity reduction is the well-known Explicit MPC [9], in which the state space is explored offline, in order to obtain an explicit solution for the control action. The online computation of the input, then, is reduced to the evaluation of a piecewise linear function. Another interesting technique is represented by the *Event-based MPC*, which permits to reduce the occurrences of the computation of the control action by including the predictive algorithm in an event-based framework [10].

However, the most intuitive and widespread approach for reducing the computational complexity consists in the reduction of the number of degrees of freedom of the controller, that is, the number of optimization variables of the QP problem.

In the classic formulation of MPC, the degrees of freedom of the optimization are equal to the number of inputs multiplied by the lengths of the control horizon (in terms of number of sampling periods). A common practice to reduce the number of variables consists in choosing a control horizon shorter than the prediction one [11]. Another widespread technique for reducing the number of variables involved in the optimization is the parametrization of the input functions [12]–[14]. This is the case, for instance, of the so-called *Predictive Functional Control* [15]. The parametrization consists in approximating the control sequence by means of a function with a minor number of degrees of freedom. This is often implemented by approximating every input with a piecewise constant function (with less steps than the control horizon). In other words, the input sequence is forced to be constant along several time-steps in the resolution of the QP problem. This technique is usually referred to as Input Blocking (IB).

However, such parametrization is strongly limited by the sampling period of the system. As a matter of fact, the steps of the staircase parametrization must occur at a multiple of the sampling period. Considering that common prediction horizons are typically in the order of ten sampling periods, such constraint limits the goodness of the approximation. This could be overcome by formulating the predictive equations in the continuous-time domain.

Continuous-time MPC was extensively studied in several works [16], [17]. Generally, in its formulation, the inputs

are parametrized by means of a sum of exponential function, allowing simpler QPs compared to the discrete case. Of course, the resolution of the online optimization problem needs a finite time to be performed. During the time required by the optimization, the open-loop trajectory is implemented and an intermittent feedback approach can be used, giving rise to the so-called *Intermittent Predictive Control*, which somehow lies between the continuous-time and the discrete-time approaches [18], [19].

In this paper, we exploit the advantages of the continuous-time approach to formulate a predictive model in which the control inputs are parametrized as staircase functions (similarly to IB). This permits to set the control and prediction time-instants arbitrarily, disregarding the sampling period of the system. In this way, the steps of the staircase parametrization do not need to coincide with a multiple of the sampling period, as happens with classic IB techniques. The same reasoning can be applied to the prediction time-instants: in classic MPC the output prediction is evaluated in the multiples of the sampling period, while our formulation allows the arbitrary choice of the prediction instants. With respect to IB, this allows a more flexible choice in the parameters, ensuring better flexibility in the controller. As a matter of fact, the piecewise constant function better approximates the input sequence, while keeping the same structure of the IB parametrization (staircase function and same number of variables). The basic principle of such method was successfully implemented in [20], in order to cope with the fast sampling period of robotic systems. The method is also formulated in terms of relative increment of the control action (i.e.  $\Delta u$ ), by imposing that such increment must be piecewise constant along the control horizon.

Note that even though the predictive equations are derived in the continuous-time domain, the resulting controller can be directly implemented in discrete-time, since it is based on the assumption that the control input is a piecewise constant function.

We also provide a tuning method in which we choose the optimal distribution of the prediction and control instants along the horizon, by minimizing the deviation, in terms of integral of the absolute error with respect to the full-degree-of-freedom (FDOF) controller, that is, the standard MPC without any parametrization. By applying the same optimal tuning rule to the IB technique, we show that our method guarantees a better approximation of the FDOF controller. In other words, given the same number of variables for both the staircase parametrizations and aiming at minimizing the deviation with respect to the FDOF controller, our method ensures better results than the ones of classic IB.

The paper is organized as follows. Section II recalls the basic notions of standard MPC and Input Blocking. In Section III the proposed method is presented; then Section IV discusses the tuning and the choice of the prediction and control time-instants. In Section V a second-order fast-dynamics system is analyzed as a case study, in order to demonstrate the effectiveness of the proposed strategy. Finally, conclusions are exposed in Section VI.

## II. PRELIMINARIES

### A. Generalities

Consider a discrete-time linear dynamic model, with  $n$  states,  $m$  inputs and  $q$  outputs, such that:

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) \end{aligned} \quad (1)$$

where  $A_d \in \mathbb{R}^{n \times n}$ ,  $B_d \in \mathbb{R}^{n \times m}$ , and  $C_d \in \mathbb{R}^{q \times n}$ .

Now, consider the system at sampling time  $k > 0$  and the corresponding state  $x(k)$ . The prediction of the output can be calculated by forward solving the difference equation (1) from the current state  $x(k)$ . In particular, consider:

- a prediction horizon  $N_p$ , which represents the number of future sampling times in which the output prediction has to be calculated;
- a control horizon  $N_c$ , which represents the number of future sampling times for which the control actions are taken into account ( $N_c \leq N_p$ ).

The output prediction in the future  $N_p$  sampling times, calculated at time  $i$ , is given by:

$$\begin{aligned} y(k+1|k) &= C_d A_d x(k) + C_d B_d u(k) \\ y(k+2|k) &= C_d A_d^2 x(k) + C_d A_d B_d u(k) + C_d B_d u(k+1) \\ &\vdots \\ y(k+N_p|k) &= C_d A_d^{N_p} x(k) + \cdots + C_d A_d^{N_p-N_c} B_d u(k+N_c-1) \end{aligned} \quad (2)$$

or, in matrix form:

$$Y = Fx(k) + GU \quad (3)$$

where  $F \in \mathbb{R}^{qN_p \times n}$  and  $G \in \mathbb{R}^{qN_p \times mN_c}$  are defined as

$$F := \begin{bmatrix} C_d A_d \\ C_d A_d^2 \\ \vdots \\ C_d A_d^{N_p} \end{bmatrix}$$

$$G := \begin{bmatrix} C_d B_d & 0 & 0 & \cdots & 0 \\ C_d A_d B_d & C_d B_d & 0 & \cdots & 0 \\ \vdots & & & & \\ C_d A_d^{N_p-1} B_d & \cdots & \cdots & \cdots & C_d A_d^{N_p-N_c} B_d \end{bmatrix}$$

Then,  $Y \in \mathbb{R}^{qN_p}$  is the predicted output, and  $U := [u(k), \dots, u(k+N_c-1)]$  is the future output along the control horizon.

Now, given a reference signal  $W = [w(k+1), \dots, w(k+N_p)]$ , it is possible to write an objective function  $J$  that accounts for the predicted control error and control effort, that is:

$$J = (W - Y)^T Q (W - Y) + U^T R U \quad (4)$$

where  $Q \in \mathbb{R}^{qN_p \times qN_p}$  and  $R \in \mathbb{R}^{mN_c \times mN_c}$  are suitable weighting matrices. Using the predictive equation (3), it is possible to rewrite (4) as a function of the sole control inputs  $U$ :

$$J = (W - Fx(k) - GU)^T Q (W - Fx(k) - GU) + U^T R U \quad (5)$$

Therefore, at each sampling time, the following QP problem has to be solved:

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} U^T H U + f^T U \\ & \text{subject to} \quad E U \leq d \end{aligned} \quad (6)$$

where  $H = 2G^T QG + R$ ,  $f^T = 2(F - W)^T QG$ , and  $E U \leq d$  is a generic set of linear inequalities, which typically represent the constraints on the amplitude and slew rate of inputs and outputs. The first element of the optimal control sequence  $U$  is applied to the system, and the procedure is repeated, in the spirit of receding horizon control.

Often, the MPC problem is formulated in terms of relative increment of the control input  $\Delta u$ , instead of  $u$  [21]. Given the model (1), this can be obtained by means of a state augmentation, which adds an integrator to the original model. Thus, defining the new state as

$$z := \begin{bmatrix} \Delta x \\ x \end{bmatrix} \quad (7)$$

the new model results to be:

$$z(i+1) = \tilde{A}_d z(i) + \tilde{B}_d \Delta u(i) \quad (8)$$

$$y(i) = \tilde{C}_d z(i) \quad (9)$$

where:

$$\tilde{A}_d = \begin{bmatrix} A_d & o_{n \times n} \\ I_n & o_{n \times n} \end{bmatrix}, \quad \tilde{B}_d = \begin{bmatrix} B_d \\ o_{n \times m} \end{bmatrix}, \quad \tilde{C}_d = [o_{q \times n} \quad C_d]. \quad (10)$$

Using the system  $(\tilde{A}_d, \tilde{B}_d, \tilde{C}_d)$  instead of  $(A_d, B_d, C_d)$  the derivation of the equations remains exactly the same. In this way, the resulting controller calculates the optimized control actions in terms of  $\Delta U$ . Then, the control action  $u(i)$  is updated by adding the first element of the sequence  $\Delta U$  and the resulting value is applied to the process.

### B. Input Blocking

A method to reduce the computational complexity of the QP problem (6) consists in reducing the number of degrees of freedom of the optimization. This can be done by choosing a control horizon shorter than the prediction one. Another widespread strategy is the so-called Input Blocking [12], which consists in forcing the control inputs to be constant along several sampling times. The method is equivalent to parametrizing the input by means of a staircase function, in which the steps occur at multiples of the sampling period. To this end, let  $\hat{U} := [\hat{u}_1, \dots, \hat{u}_M]$ , with  $M < N_c$ , be the new optimization variable, such that:

$$U = T \hat{U} \quad (11)$$

where  $T \in \mathbb{R}^{N_c \times M}$  is the so-called *blocking matrix*. The matrix  $T$  is a Boolean matrix, which represents the mapping between the new and the old optimization variables. For instance, considering a SISO system and aiming at implementing

an IB strategy in which  $N_c = 3$ ,  $M = 2$ , and  $u_2 = u_3$ , the blocking matrix results to be:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Therefore, the QP problem (6) can be reformulated with respect to the new optimization variable  $\hat{U}$ , becoming:

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \hat{U}^T \hat{H} \hat{U} + \hat{f}^T \hat{U} \\ & \text{subject to} \quad \hat{E} \hat{U} \leq d \end{aligned} \quad (12)$$

where  $\hat{H} = T^T H T$ ,  $\hat{f}^T = f^T T$ ,  $\hat{E} = E T$ . Then, the original optimization variable can be obtained as  $U = T \hat{U}$  and the first element of the sequence is applied to the system.

This strategy permits to reduce the computational complexity of the optimization problem, since the number of variables is lowered from  $mN_c$  to  $mM$ . However, the solution to the problem (12) is suboptimal for the problem (6) and the goodness of such approximation depends on the choice of the matrix  $T$ . Moreover, note that this parametrization forces the steps of the staircase function to happen at a multiple of the sampling period and this represents a limit in the approximation, considering that common prediction horizons are typically in the order of  $N_p = 10$ . For this reason we propose to formulate the predictive model in the continuous-time domain, as it will be explained in the next section.

### III. PROPOSED METHOD

As mentioned above, the IB technique is basically a step-wise parametrization of the input, in which the steps of the staircase function necessarily happen at a multiple of the sampling period. This is due to the formulation of the predictive equations in the discrete-time domain. In order to overcome this limitation, we formulate the predictive equations by using a continuous-time model, but imposing the input to be piecewise constant, as in the discrete case. In this way the staircase parametrization will result somehow more flexible, since the choice of the time-instants in which each step occurs will not be constrained by the sampling period. Note that the resulting controller will be directly implemented in discrete-time, since it is based on the assumption that the input is piecewise constant.

#### A. Formulation of the predictive model

Consider the continuous-time state-space formulation of a given process with  $m$  inputs,  $q$  outputs and an  $n$ -dimensional state vector:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (13)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ , and  $C \in \mathbb{R}^{q \times n}$ .

Now, consider a time instant  $t_0$  and the corresponding state  $x_0 = x(t_0)$ . The predicted output at a generic time  $t = t_0 + \tau$ ,  $\tau > 0$ , calculated at time  $t_0$  is given by:

$$y(t_0 + \tau | t_0) = Ce^{A\tau} x_0 + Ce^{A\tau} \int_0^\tau e^{-A\gamma} Bu(\gamma) d\gamma \quad (14)$$

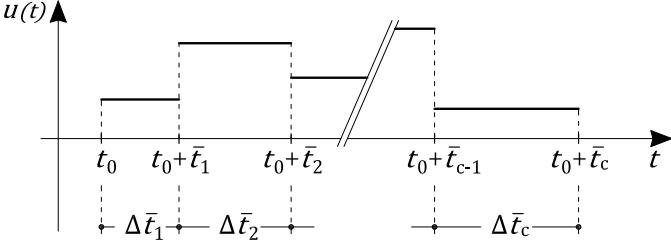


Fig. 1. Sketch of the parametrization of the input as a staircase function.

Assume that  $B$  is constant and  $u$  is a staircase function with a number of steps equal to  $c$ . Let  $\bar{t}_i$  be the time instants, with respect to  $t_0$ , in which the steps of the function are applied and let  $u(t_0 + \bar{t}_i)$  be the input value in the time interval between  $\bar{t}_{i-1}$  and  $\bar{t}_i$ . These intervals are defined, then, as  $\Delta \bar{t} := [\bar{t}_1 - t_0, \bar{t}_2 - \bar{t}_1, \dots, \bar{t}_c - \bar{t}_{c-1}]$ . A sketch of the input function  $u(t)$  is drawn in Figure 1. Note that the lengths of the intervals do not need to be the same. Thus, (14) becomes:

$$y(t_0 + \tau | t_0) = Ce^{A\tau}x_0 + C \sum_{i=1}^c \Gamma_i u(t_0 + \bar{t}_i) \quad (15)$$

where, assuming  $\bar{t}_0 = 0$  for notation simplicity,

$$\Gamma_i := \begin{cases} e^{A(\tau - \bar{t}_i)} \int_0^{\Delta \bar{t}_i} e^{A\gamma} d\gamma B, & \text{if } \tau \geq \bar{t}_i \\ \int_0^{\tau - \bar{t}_{i-1}} e^{A\gamma} d\gamma B, & \text{if } \bar{t}_{i-1} < \tau < \bar{t}_i \\ 0, & \text{if } \tau < \bar{t}_i \wedge \tau < \bar{t}_{i-1} \end{cases} \quad (16)$$

Having defined the control time instants  $\bar{t}_i$ , a constant matrix  $\Phi_\tau$  can be defined for a given prediction time  $\tau$ , such that:

$$\Phi_\tau := [C\Gamma_1, C\Gamma_2, \dots, C\Gamma_c] \quad (17)$$

Substituting (17) in (15) gives:

$$y(t_0 + \tau | t_0) = Ce^{A\tau}x_0 + \Phi_\tau U \quad (18)$$

where  $U := [u(t_0 + \bar{t}_1), \dots, u(t_0 + \bar{t}_c)]^T$ .

Extending this reasoning to a generic number of prediction times  $(\tau_1, \dots, \tau_p)$ ,  $p > 0$ , we obtain the following system of equations:

$$\begin{cases} y(t_0 + \tau_1 | t_0) = Ce^{A\tau_1}x_0 + \Phi_{\tau_1} U \\ \vdots \\ y(t_0 + \tau_p | t_0) = Ce^{A\tau_p}x_0 + \Phi_{\tau_p} U \end{cases} \quad (19)$$

As for the control time instants, the prediction instants  $(\tau_1, \dots, \tau_p)$  are not necessarily equally spaced. System (19) can be rewritten in matrix form as:

$$Y = Fx_0 + GU \quad (20)$$

where  $Y \in \mathbb{R}^{qp \times 1}$  is the predicted output vector at times  $(t_0 + \tau_1, \dots, t_0 + \tau_p)$ ,  $U \in \mathbb{R}^{nc \times 1}$  is the vector of the control inputs, and:

$$F = \begin{bmatrix} Ce^{A\tau_1} \\ \vdots \\ Ce^{A\tau_p} \end{bmatrix} \in \mathbb{R}^{qp \times np}, \quad G = \begin{bmatrix} \Phi_{\tau_1} \\ \vdots \\ \Phi_{\tau_p} \end{bmatrix} \in \mathbb{R}^{qp \times mc}.$$

In classic discrete MPC, prediction and control instants are equally distributed along the respective horizons and correspond to the sampling times. However, although the derivation of the predictive equations in continuous time might seem less intuitive, its great advantage consists in the possibility of using a free distribution of prediction and control instants along the horizons, disregarding the real sampling period of the system. As a matter of fact, it is not necessary for the time instants  $\tau_i$  and  $\bar{t}_i$  to be neither equally distributed nor coincident with a sampling time-instant. Anyhow, note that the controller can be implemented using a receding horizon approach with a given sampling period  $T$ , under the only condition  $\bar{t}_1 \geq T$  (as better explained in Section IV).

### B. Handling of constraints

Constraints expressed as linear combinations of the control actions can be included in the MPC framework, just like in the classic MPC formulation, resulting in a constrained LQ problem.

### C. Optimal control action

As mentioned in Section II, in classic linear MPC, the control sequence is obtained by minimizing the cost function (4). In the continuous-time domain this can be rewritten as:

$$J = \int_{t_0}^{t_0 + \tau_p} (y(t) - w(t))^2 dt + \lambda \int_{t_0}^{\bar{t}_c} u^2(t) dt \quad (21)$$

This cost function can be approximated as:

$$J \approx (Y - W)^T \bar{\Delta\tau} (Y - W) + \lambda U^T \bar{\Delta t} U \quad (22)$$

where  $W \in \mathbb{R}^{qp \times 1}$  is the reference signal at times  $(t_0 + \tau_1, \dots, t_0 + \tau_p)$ ,  $\bar{\Delta\tau} = \text{diag}(\tau_1, \tau_2 - \tau_1, \dots, \tau_p - \tau_{p-1})$ ,  $\bar{\Delta t} = \text{diag}(\Delta t_1, \dots, \Delta t_c)$ , and  $\lambda > 0$ .

Note that (22) is derived from (21) by substituting the integrals with a discrete approximation, with step size equal to the intervals  $(\tau_1, \tau_2 - \tau_1, \dots, \tau_p - \tau_{p-1})$ . Therefore, using (20), the following QP results (in terms of the sole control action  $U \in \mathbb{R}^{mc \times 1}$ ):

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} U^T H U + f^T U \\ & \text{subject to} \quad E U \leq d \end{aligned} \quad (23)$$

where  $H = 2G^T \bar{\Delta\tau} G + \lambda \bar{\Delta t}$ ,  $f^T = 2(F - W)^T \bar{\Delta\tau} G$ , and  $E U \leq d$  is a generic set of linear inequalities, as in (6).

### D. Formulation in terms of $\Delta U$

As already mentioned, classic MPC controllers are usually formulated in terms of increments of the control action (i.e.,  $\Delta u$  instead of  $u$ ). The first optimal  $\Delta u$  is then added to the previous control action and this value is applied to the system. Given a discrete-time state-space system, this formulation can be achieved by means of a state augmentation, as mentioned in Section II. In our case, an analogous state augmentation of the continuous-time model (13) would lead to a formulation of the predictive equations in which the derivative of the inputs is a staircase function. This means that the control input would

result to be a piecewise linear function. However, we want to formulate the predictive equations by imposing the increment of the control action ( $\Delta u$ ) to be piecewise constant along the correspondent time-intervals  $\Delta \bar{t}_i$  (as in discrete MPC). To this aim, the formulation retraces the derivation of the predictive equations from (14). By assuming  $u = u_0 + \Delta u(t)$ , where  $u_0 = u(t_0)$  and  $\Delta u$  is a staircase function with  $c$  steps, it results:

$$y(t_0 + \tau | t_0) = Ce^{A\tau}x_0 + \Omega_\tau u_0 + C \sum_{i=1}^c \tilde{\Gamma}_i \Delta u(t_0 + \bar{t}_i) \quad (24)$$

where  $\Delta u(t_0 + \bar{t}_i) = u(t_0 + \bar{t}_i) - u(t_0 + \bar{t}_{i-1})$  and

$$\begin{aligned} \Omega_\tau &:= C \int_0^{\tau-t_0} e^{A\gamma} d\gamma B \\ \tilde{\Gamma}_i &:= \begin{cases} \int_0^{\tau-\bar{t}_{i-1}} e^{A\gamma} d\gamma B, & \text{if } \tau > \bar{t}_{i-1} \\ 0, & \text{if } \tau \leq \bar{t}_{i-1} \end{cases} \end{aligned}$$

Therefore:

$$\tilde{\Phi}_\tau := \left[ C\tilde{\Gamma}_1, C\tilde{\Gamma}_2, \dots, C\tilde{\Gamma}_c \right] \quad (25)$$

and finally:

$$Y = \tilde{F} \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} + \tilde{G} \Delta U \quad (26)$$

where  $\Delta U = [\Delta u(t_0 + \bar{t}_1), \dots, \Delta u(t_0 + \bar{t}_c)]^T$  and

$$\tilde{F} = \begin{bmatrix} Ce^{A\tau_1}, & \Omega_{\tau_1} \\ \vdots & \vdots \\ Ce^{A\tau_p}, & \Omega_{\tau_p} \end{bmatrix} \in \mathbb{R}^{qp \times np}, \quad \tilde{G} = \begin{bmatrix} \tilde{\Phi}_{\tau_1} \\ \vdots \\ \tilde{\Phi}_{\tau_p} \end{bmatrix} \in \mathbb{R}^{qp \times mc}.$$

Once the new predictive equations have been derived, the optimal control problem can be set up as in Section III-C, but in terms of  $\Delta U$ . Then, at each time-step, the current value of the control action  $u_0$  will be updated by summing the first element of the optimal sequence  $\Delta U$  and will be applied to the system.

#### IV. TUNING OF THE PREDICTION AND CONTROL HORIZONS

The length of the prediction horizon and the weight on the control action  $\lambda$  can be chosen by following standard MPC tuning rules (e.g. [11]). Therefore, the main issue in the practical application of the proposed method is the choice of an optimal distribution of the prediction and control time-instants along the respective horizons. As a matter of fact, in discrete MPC, such instants are fixed and correspond to the future sampling times. However, the proposed method requires to explicit the time-instants  $\tau_i$  and  $\bar{t}_j$ , necessary for the approximation of the integral cost function (21) and the parametrization of the control inputs, respectively. This choice must be made during the design of the controller, then the distribution of time-instants will always be the same.

The tuning strategy is based on the assumption that the proposed approximated control law should mimic at best the FDOF controller (i.e. the discrete-time MPC without any approximation). Thus, given a prediction horizon  $T_p$  in terms

of seconds, the number of prediction time-instants  $p$  along the predictive horizon, and the number of steps  $c$  of the approximated input function; the distribution of  $\tau_i$  and  $\bar{t}_j$  should minimize the integral of the value of the difference between the control laws given by the FDOF controller and the proposed one, that is:

$$\begin{aligned} \text{minimize} \quad & \int_0^{2T_p} |(u_{FDOF}(t) - u(t))| dt \\ \text{subject to} \quad & \sum_{i=1}^p \tau_i \leq T_p ; \quad \tau_i \geq \tau_{i-1} \\ & \sum_{j=1}^c \bar{t}_j \leq T_p ; \quad \bar{t}_j \geq \bar{t}_{j-1} \\ & \bar{t}_1 \geq T_s \end{aligned} \quad (27)$$

where  $\tau_i$  and  $\bar{t}_j$  are the optimization variables, while  $u_{FDOF}$  and  $u$  are the control inputs given by the FDOF and the proposed controller, respectively.  $T_s$  represents the sampling period at which the receding horizon is implemented. The upper integration extreme was chosen equal to  $2T_p$ , in order to account for all the transient in the minimization (since  $T_p$  is generally shorter than the rise time of the system). Moreover, note that the last condition  $\bar{t}_1 \geq T_s$  is necessary for the proposed controller to be feasible for the use of receding horizon with a given sampling period  $T_s$ . As a matter of fact, in order for this to happen, the successive sampling time must occur within the first step of the input parametrization.

Of course the goodness of the approximation depends on  $c$  and  $p$ : the closer they are to the FDOF's values, the smaller is the deviation between the two inputs. Figure 2 shows the Pareto curve of the objective function of Problem (27) over the number of control steps  $c$  for the case study (28), given a fixed number of prediction instants  $p = 10$ . As expected, the cost value decreases for  $c$  getting closer to the FDOF's value.

It is important to point out that a limitation of this tuning approach is that the choice of the prediction and control time-instants is optimal only with respect to the reference signal used to obtain the control input  $u_{FDOF}$ . In this work only step responses were considered. The same limitation applies to the presence of disturbances or unmodeled dynamics, since they are not considered in the generation of  $u_{FDOF}$ .

Note that the optimization problem (27) can be addressed by means of gradient-based algorithms (e.g. *fmincon* in Matlab). As a matter of fact, the optimization variables can span continuously the prediction horizon, without any constraint given by the sampling period. On the contrary, the analogous discrete optimization problem would constrain the time-instants to be a multiple of the sampling period. This happens, for instance, when the same tuning rule is applied to IB: the input is forced to be constant along a whole number of sampling periods, leading to a discrete optimization problem. The possibility of using gradient-based algorithms is an advantage in case a high number of optimization variables is involved, since the discrete problem should be solved by means of combinatorial or stochastic methods (e.g. genetic algorithms) and this could be much more time-consuming.

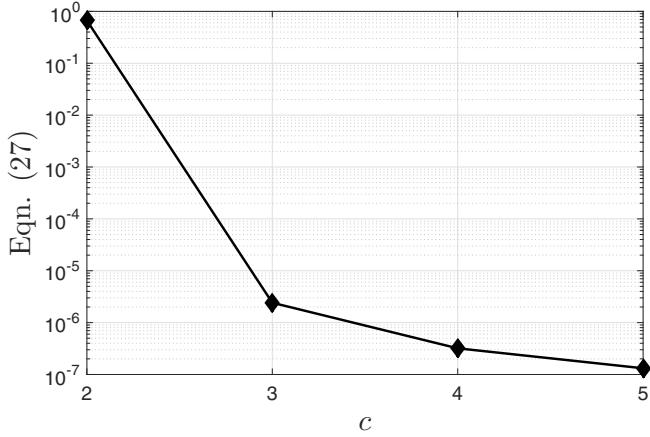


Fig. 2. Pareto curve of the cost function (27) over  $c$ , with  $p = 10$ , for the system (28). The y-axis is in logarithmic scale.

## V. NUMERICAL RESULTS

As an illustrative example, consider the fast-dynamics second-order system

$$G(s) = \frac{2500}{(s + 50)^2} \quad (28)$$

and a sampling time  $T_s = 0.01$  s. In order to control the system by means of a model predictive algorithm, consider a prediction horizon  $N_p = 10$ , accordingly with common tuning rules used in MPC practice [11]. In terms of seconds, this corresponds to a prediction horizon equal to  $T_p = 0.1$  s. To prove the effectiveness of the proposed method, four MPC strategies are analyzed:

- FDOF controller, that is, classic MPC with  $N_c = N_p$  (this controller represents the reference case for the following approximated strategies);
- classic MPC with  $N_c < N_p$ ;
- Input Blocking (IB);
- the proposed method (in the remainder of the section it will also be referred to as Continuous Input Blocking (CIB)).

In particular, the aim of this section is the comparison of the last three methods ( $N_c < N_p$ , IB, CIB), in order to demonstrate which one ensures the best approximation of the FDOF case, being equal the reduction of computational complexity (i.e., with the same number of optimization variables). Therefore, given the same number of control instants along the horizon, the three approximated controllers are set as follows:

- CIB: the time-instants  $\tau_i$  and  $\bar{\tau}_j$  are chosen by following the tuning rule (27).
- IB: similarly to the previous case, the Blocking Matrix  $T$  is selected by minimizing the same objective function as in (27).
- $N_c < N_p$ : since it is the classic MPC with  $N_c < N_p$ , the control instants correspond to the first  $N_c$  sampling times.

Figure 3 shows the different choices in the distribution of the prediction and control time-instants between the IB and the

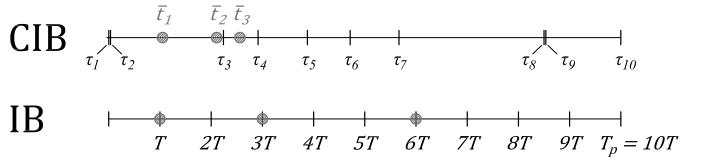


Fig. 3. Different choices in the distribution of the prediction and control time-instants between CIB and IB ( $\lambda = 5$ ).

CIB strategies following the tuning rule (27) for a number of control moves equal to 3 (i.e.  $c = M = 3$ ) and with  $\lambda = 5$ . Of course the prediction instants in the IB technique are the multiples of the sampling period, while for the CIB method they can be chosen arbitrarily. Similarly, the control instants in the CIB method can be chosen freely, whereas in IB the control inputs must be kept constant along a whole number of sampling periods (in this case it results  $u_2 = u_3$  and  $u_4 = u_5 = u_6$ ).

All the controllers are implemented in terms of relative increments of the control input (i.e. in terms of  $\Delta u$ ) and implementing the receding horizon approach. The simulations are performed for a number of control moves ( $c$ ,  $M$ ,  $N_c$  in the corresponding cases) equal to 2 and 3. For each controller, the goodness of the approximation of the FDOF case is evaluated by means of the following indices:

$$\eta_1 := \frac{\text{IAE}(y - y_{\text{FDOF}})}{\text{IAE}(w - y_{\text{FDOF}})} \cdot 100 \quad (29)$$

$$\eta_2 := \sqrt{\frac{\text{ISE}(y - y_{\text{FDOF}})}{\text{ISE}(w - y_{\text{FDOF}})}} \cdot 100 \quad (30)$$

$$\eta_3 := \frac{\text{IAE}(u - u_{\text{FDOF}})}{\text{IAE}(u_\infty - u_{\text{FDOF}})} \cdot 100 \quad (31)$$

$$\eta_4 := \sqrt{\frac{\text{ISE}(u - u_{\text{FDOF}})}{\text{ISE}(u_\infty - u_{\text{FDOF}})}} \cdot 100 \quad (32)$$

where  $y$  is the step-response of the evaluated method,  $y_{\text{FDOF}}$  is the step-response in the FDOF case,  $w$  is the unitary step,  $u$  is the control input of the evaluated method,  $u_{\text{FDOF}}$  is the control input given by the FDOF controller,  $u_\infty$  is the steady-state value of the control input, and IAE( $\cdot$ ) and ISE( $\cdot$ ) calculate the integral of the absolute value and the integral of the square value of a signal, that is:

$$\text{IAE}(z) := \int_0^\infty |z(t)| dt$$

$$\text{ISE}(z) := \int_0^\infty (z(t))^2 dt$$

The indices  $\eta_1$  and  $\eta_2$  measure how well the step-response of the method under evaluation approximates the FDOF step-response. Similarly,  $\eta_3$  and  $\eta_4$  evaluate how well the control action mimics the FDOF control law. Such indices can be considered as a percentage decay-measure with respect to the ideal case: the smaller is the index value, the better is the approximation.

The results of the simulations are reported in Table I and II. The weight on the control action was chosen as  $\lambda = 5$  and

TABLE I  
NUMERICAL COMPARISON OF THE APPROXIMATED METHODS (CIB = PROPOSED METHOD, IB = INPUT BLOCKING,  $N_c < N_p$  = CLASSIC MPC WITH SHORTER CONTROL HORIZON) WITH  $\lambda = 5$ .

N. control instants	Method	$\eta_1$	$\eta_2$	$\eta_3$	$\eta_4$
2	CIB	0.91	0.70	2.89	1.74
	IB	1.52	1.30	5.17	3.71
	$N_c < N_p$	6.73	5.99	23.4	17.6
3	CIB	$5 \cdot 10^{-5}$	$4 \cdot 10^{-5}$	$1.5 \cdot 10^{-4}$	$8 \cdot 10^{-5}$
	IB	0.55	0.47	1.74	1.26
	$N_c < N_p$	2.57	2.18	9.14	6.51

TABLE II  
NUMERICAL COMPARISON OF THE APPROXIMATED METHODS (CIB = PROPOSED METHOD, IB = INPUT BLOCKING,  $N_c < N_p$  = CLASSIC MPC WITH SHORTER CONTROL HORIZON) WITH  $\lambda = 0.2$ .

N. control instants	Method	$\eta_1$	$\eta_2$	$\eta_3$	$\eta_4$
2	CIB	$8 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$2.4 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$
	IB	3.66	2.95	13.5	12.8
	$N_c < N_p$	6.24	5.10	14.6	13.0
3	CIB	$6 \cdot 10^{-5}$	$4 \cdot 10^{-5}$	$2.0 \cdot 10^{-4}$	$1.3 \cdot 10^{-4}$
	IB	1.19	0.87	4.54	4.24
	$N_c < N_p$	1.36	1.06	4.83	4.54

$\lambda = 0.2$  respectively. As expected, the classic strategy with  $N_c < N_p$  generally leads to worse results with respect to the other two methods (being equal the number of optimization variables). Most important, compared to the other strategies, the proposed method gives better values for all the indices and for all the considered cases. This is also visible in Figures 4 and 6, which show the step response of the four controllers, with  $\lambda = 5$  and 0.2 respectively. As expected, the proposed controller (solid black line) follows the FDOF one (dashed grey line) more accurately. Similarly, Figures 5 and 7 show the control inputs given by the four controllers, with  $\lambda = 5$  and 0.2, respectively. As in the previous case, the proposed method (solid black line) is evidently closer to the reference control signal (dashed grey line).

The use of a number of control instants greater than 3 would lead to similar considerations. However, the use of  $c > 3$  for this case study would not give significant improvements, whereas it would unnecessarily increase the computational burden of the algorithm. As a matter of fact, the Pareto curve depicted in Figure 2 shows that a greater number of control instants does not lead to a remarkable reduction in the minimum of the objective function in (27).

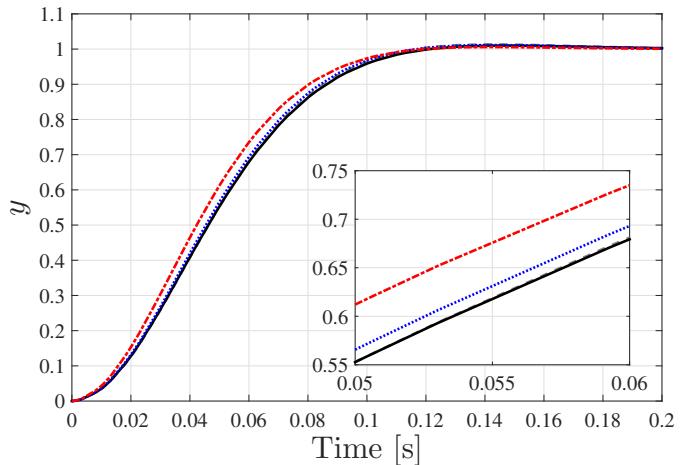


Fig. 4. Step-response comparison between the methods with two control instants ( $\lambda = 5$ ). FDOF (dashed grey), CIB (solid black), IB (dotted blue), and  $N_c < N_p$  (dashed-dotted red).

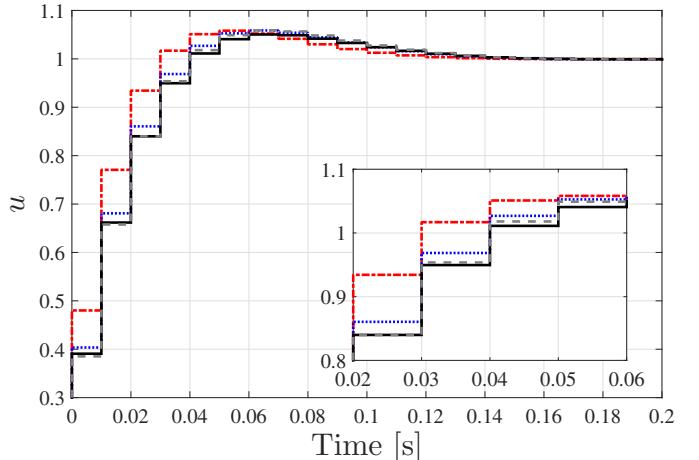


Fig. 5. Control-input comparison between the methods with two control instants ( $\lambda = 5$ ). FDOF (dashed grey), CIB (solid black), IB (dotted blue), and  $N_c < N_p$  (dashed-dotted red).

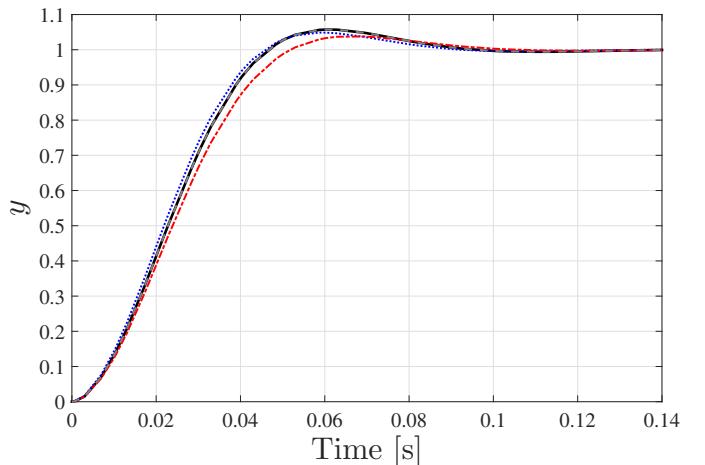


Fig. 6. Step-response comparison between the methods with two control instants ( $\lambda = 0.2$ ). FDOF (dashed grey), CIB (solid black), IB (dotted blue), and  $N_c < N_p$  (dashed-dotted red).

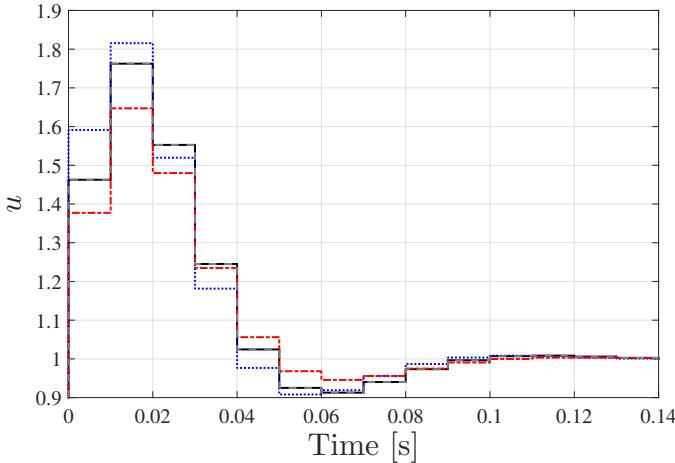


Fig. 7. Control-input comparison between the methods with two control instants ( $\lambda = 0.2$ ). FDOF (dashed grey), CIB (solid black), IB (dotted blue), and  $N_c < N_p$  (dashed-dotted red).

## VI. CONCLUSIONS

This paper presented a new method to reduce the computational complexity of MPC with online optimization. The core of the proposed method is the formulation of the predictive equations and the parametrization of the inputs in the continuous-time domain. The control sequence is forced to be piecewise constant along the control horizon, but with less steps than in the FDOF case. The spirit of such parametrization is similar to the widespread IB technique. However, the continuous-time formulation permits to set the prediction and control time-instants disregarding the sampling period of the system. This ensures a better flexibility of the parametrization with respect to IB. Note that the staircase parametrization of the input permits to consider the resulting controller directly in the discrete-time domain.

The optimal tuning of the proposed method has been addressed by minimizing the deviation between the approximated controller and the ideal one (i.e. discrete MPC without any parametrization). It should be pointed out that such minimization can be performed by means of gradient-based algorithms, since the optimization variables can continuously span the time-interval. This is an advantage in case a high number of variables is involved in the optimization. As a matter of fact, the discrete counterpart of such minimization should be carried out by means of combinatorial or stochastic algorithms, which could be much more time-consuming and difficult to tune.

The analysis of a case study has demonstrated the effectiveness of the proposed method against two common strategies for complexity reduction (IB and control horizon shorter than the predictive one). The numerical results show that the proposed method guarantees a better approximation of the ideal controller, in terms of both output response and control law.

Future work will include the analysis of the effects of modeling uncertainties and disturbances on the optimal tuning methodology.

## ACKNOWLEDGMENT

This work has been partially funded by the National R+D+i Plan Project DPI2014-56364-C2-1-R of the Spanish Ministry of Economy and Competitiveness and ERDF funds.

## REFERENCES

- [1] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, pp. 267–278, 2010.
- [2] E. F. Camacho and C. Bordons, *Model Predictive Control*. Springer-Verlag, 2007.
- [3] J. M. Maciejowski, *Predictive control with constraints*. Pearson education, 2002.
- [4] J. A. Rossiter, L. Wang, and G. Valencia-Palomo, "Efficient algorithms for trading off feasibility and performance in predictive control," *International Journal of Control*, vol. 83, pp. 789–797, 2010.
- [5] H. Ferreau, H. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *International Journal of Robust and Nonlinear Control*, vol. 18, pp. 816–830, 2008.
- [6] R. A. Bartlett, L. T. Biegler, J. Backstrom, and V. Gopal, "Quadratic programming algorithms for large-scale model predictive control," *Journal of Process Control*, vol. 12, pp. 775–795, 2002.
- [7] B. Kouvaritakis, M. Cannon, and J. A. Rossiter, "Who needs QP for linear MPC anyway?" *Automatica*, vol. 27, pp. 879–884, 2002.
- [8] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans, "Efficient robust predictive control," *IEEE Transactions on Automatic Control*, vol. 45, pp. 1545–1549, 2002.
- [9] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming – the explicit solution," *IEEE Transactions on Automatic Control*, vol. 47, pp. 1974–1985, 2003.
- [10] A. Pawłowski, J. L. Guzmán, M. Berenguel, J. E. Normey-Rico, and S. Dormido, "Event-based gpc for multivariable processes: A practical approach with sensor deadband," *IEEE Transactions on Control Systems Technology*, 2016.
- [11] J. L. Garriga and M. Soroush, "Model predictive control tuning methods: A review," *Industrial and Engineering Chemistry Research*, vol. 49, pp. 3505–3515, 2010.
- [12] R. Cagienard, P. Grieder, E. C. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *Journal of Process Control*, vol. 17, pp. 563–570, 2007.
- [13] P. Tøndel and T. A. Johansen, "Complexity reduction in explicit model predictive control," in *IFAC World Congress*, 2002, pp. 232–256.
- [14] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733–764, 2003.
- [15] J. Richalet and D. O'Donovan, *Predictive functional control: principles and industrial applications*. Springer Science & Business Media, 2009.
- [16] H. Demircioğlu and P. J. Gawthrop, "Continuous-time generalized predictive control (CGPC)," *Automatica*, vol. 38, pp. 55–74, 1991.
- [17] L. Wang, "Continuous time model predictive control design using orthonormal functions," *International Journal of Control*, vol. 74, pp. 1588–1600, 2001.
- [18] P. J. Gawthrop and L. Wang, "Intermittent predictive control of an inverted pendulum," *Control Engineering Practice*, vol. 14, pp. 1347–1356, 2006.
- [19] ———, "Constrained intermittent model predictive control," *International Journal of Control*, vol. 82, pp. 1138–1147, 2009.
- [20] M. Faroni, M. Beschi, A. Visioli, and L. Molinari Tosatti, "A predictive approach to redundancy resolution for robot manipulators," in *Proceedings IFAC World Congress*, Toulouse (France), 2017.
- [21] L. Wang, *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, 2009.