H. Farooqi, L. Fagiano, P.Colaneri, D. Barlini, "Shrinking horizon parametrized predictive control with application to energy-efficient train operation", Automatica, vol. 112, pp. 108635

Shrinking horizon parametrized predictive control with application to energy-efficient train operation

Hafsa Farooqi [a], Lorenzo Fagiano [a], Patrizio Colaneri [a,b], Davide Barlini [c]

[a] *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy*

[b] *IEIIT - CNR, Politecnico di Milano, Milano, Italy*

[c] *Alstom SA, Séméac, Midi-Pyrénées, France*

Abstract

A nonlinear model predictive control approach is studied, for problems where a fixed terminal instant and corresponding terminal set to be reached are imposed. The new technique features a shrinking horizon, rather than the most common receding one, and an input parametrization strategy to reduce computational burden. The property of transferability of the parametrization strategy is introduced. Under this property, theoretical convergence guarantees in nominal conditions are obtained by construction. Two relaxed techniques are then proposed to retain recursive feasibility in presence of bounded additive input disturbance. A bound on the constraint violation achieved by these relaxed techniques as a function of the uncertainty bound is derived, too. The developed strategy is applied to the problem of energy-efficient operation of trains, in either a fully autonomous mode (with continuous input values) or a driver assistance mode (with discrete input values, resulting in a nonlinear integer program if no parametrization is used). Realistic simulation results in this context illustrate the effectiveness of the approach.

Key words: Model Predictive Control, Move blocking, Input parametrization, Nonlinear control systems, Constrained control, Train control

1 Introduction

Model Predictive Control (MPC) is an optimization-based technique with broad success in industry [14] thanks to its capability to deal with multivariable systems, state and input constraints, and both tracking and economic objectives. Research efforts of more than three decades result today in a mature theory for linear systems [10], [2], and several directions are open for further investigation [13]. In this paper, we provide new results pertaining to MPC for nonlinear systems and problems where a finite terminal instant is imposed, at which the state has to reach a given terminal set. The resulting MPC formulation features a shrinking horizon rather than the most common receding one. Moreover, we consider the use of an input parametrization strategy to reduce the computational burden of the underlying optimization program, making it possible to deal efficiently with a long prediction horizon and with discrete input constraints, which would otherwise result in a nonlinear integer program. We named the resulting control approach Shrinking horizon Parametrized Predictive Control (SPPC). To the best of our knowledge, the combined presence of nonlinear dynamics, shrinking horizon, and general input parametrization strategies is new in the literature. The existing works on input parametrization, such as move-blocking strategies, consider in fact either linear systems, see, e.g., [6–9,16,17], or nonlinear ones with a receding horizon formulation [3, 12, 18]. In our context, the relevant question is what property shall the employed parametrization enjoy for the control strategy to provide theoretical guarantees. In this respect, we introduce here the notion of transferability of the parametrization strategy, which is sufficient to obtain recursive feasibility and, in nominal conditions, satisfaction of the terminal constraints in closed loop. Then, we present two relaxed approaches, where state constraints are softened to retain recursive feasibility in presence of bounded additive input disturbance, and we derive a theoretical result linking the disturbance bound to the worst-case constraint violation.

Our approach and theoretical analysis are motivated by an industrial application pertaining to the energy efficient operation of railways (see [15] for a recent review), in collaboration with the company Alstom rail transport. MPC has been already applied in this context [1]. In addition to the above-mentioned theoretical contributions, we provide an application-oriented contribution by applying SPPC to this problem. In particular, for the first time we deal with both a fully autonomous scenario, where the input takes any value in a connected compact set, or a driver assistance scenario, where the predictive controller suggests to the driver one out of a finite discrete set of possible driv-

Email addresses: hafsa.farooqi@polimi.it (Hafsa Farooqi), lorenzo.fagiano@polimi.it (Lorenzo Fagiano), patrizio.colaneri@polimi.it (Patrizio Colaneri), davide.barlini@alstomgroup.com (Davide Barlini).

ing modes. In both scenarios, we consider the uncertainty derived from system-model mismatch and control input discretization/misapplication by the human driver. We present results obtained with Alstom in-house energy-based detailed simulation suite, CITHEL.
The paper is organized as follows. In Section 2, the railway application that motivates our research is described, and a general problem formulation is abstracted. In Section 3, we introduce the input parametrization strategies and the nominal SPPC formulation. Section 4 presents the two relaxed approaches. Section 5 provides realistic simulation results pertaining to our motivating application. We conclude and point out future steps in Section 6.

2 Motivating application and problem abstraction

Consider an electric train controlled by a digital control unit (see for example Fig. 1 depicting the one considered in the simulations of Section 5). In this application, it is convenient to take space (i.e., the train position along the track) as the independent variable, while time is one of the system's states. We discretize space with sampling distance D_s, and denote with $k \in \mathbb{Z}$ the discrete space variable. The actual position along the track at each sampling instant is thus equal to kD_s. For practical implementation, the controller has a high enough sampling rate, such that the discrete space instants can be met with sufficient accuracy. For example, with a train speed of $25\,\mathrm{ms}^{-1}$, a space discretization of 5 m, and a controller sampling rate of 50 Hz, the maximum actuation delay would be equal to 0.5 m, i.e., 10% of the sampling distance. We denote with $x(k) = [x_1(k), x_2(k)]^T$ the state of the train, where x_1 is its travel time and x_2 the speed ($\cdot^T$ denotes the matrix transpose operator), and with $u(k) \in [-1, 1]$ a normalized force (our control variable), where $u(k) = 1$ corresponds to the maximum applicable traction and $u(k) = -1$ to the maximum allowed braking within passenger comfort limits. Physically, $u(k)$ corresponds to the input handle usually employed by the driver to set the traction/braking force between zero and its maximum value, which is speed-dependent.

Fig. 1. One of the Amsterdam metro trains considered in the simulation example of this paper.

The train has to move from one station at time $x_1 = 0$ and reach the next one at time $x_1 = t_f$, covering the corresponding distance s_f. For a given pair of initial and final stations, the slope and curvature profiles are known in advance as a function of k. Thus, in nominal conditions (with rated values of the train parameters, like its mass and the specifications of the powertrain and braking systems), according to Newton's laws and using the forward Euler discretization method, the equations of motion of a reasonably accurate model of this system read:

$$\begin{aligned} x_1(k+1) &= x_1(k) + \frac{D_s}{|x_2(k)|} \\ x_2(k+1) &= x_2(k) + D_s \left(\frac{F_T(x(k),u(k)) - F_B(x(k),u(k)) - F_R(k,x(k))}{M|x_2(k)|} \right) \end{aligned} \tag{1}$$

where M is the total mass of the train, F_T the traction force, F_B the braking force, and F_R the resistive force. Functions $F_T(x,u)$, $F_B(x,u)$ are nonlinear and depend on the specific train. They include, for example, look-up tables that link the traction and braking forces to the train speed and to the control input value. These functions are derived experimentally and/or from detailed models of each train and its traction and braking systems. In our research, they are provided by the business unit at Alstom. More details on the actual functions are omitted for confidentiality reasons, however for completeness we employ here scaled versions, provided in Section 5. The resistive force $F_R(k,x)$ is also nonlinear, and it is the sum of a first term $R_v(x_2)$, accounting for resistance due to the velocity, and a second term $R_g(k)$, accounting for the effects of slopes and track curvature:

$$\begin{aligned} F_R(k,x) &= R_v(x_2) + R_g(k) \\ R_v(x_2) &= A + B|x_2| + Cx_2^2 \\ R_g(k) &= M_s \left(a_g \tan(\alpha(k)) + \frac{D}{r_c(k)} \right) \end{aligned} \tag{2}$$

where a_g is the gravity acceleration, the parameters A, B, C, D are specific to the considered train, M_s is the static mass of the train, i.e., the mass calculated without taking into account the effective inertia of the rotating components, and $r_c(k)$ and $\alpha(k)$ are, respectively, the track radius of curvature and its slope at position k. For example, an uphill track section corresponds to $\alpha(k) > 0$, i.e., a positive slope.
Besides the prescribed arrival time t_f and position s_f, state constraints include velocity limits, $\bar{x}_2(k)$, which depend on k, since a different maximum velocity is imposed for safety by the regulatory authority according to the track features at each position. Defining the terminal space step as $k_f \doteq \lfloor s_f/D_s \rfloor$ (where $\lfloor \cdot \rfloor$ denotes the flooring operation to the closest integer), the overall state constraints read:

$$\begin{aligned} x(0) &= [0, 0]^T \\ x(k_f) &= [t_f, 0]^T \\ x_2(k) &\leq \bar{x}_2(k),\ k = 0, \ldots, k_f \end{aligned}$$

The control objective is to minimize the traction energy consumption of the train while satisfying the constraints above. Applying a constant scaling factor D_s^{-1}, this corresponds to minimizing the following cost function:

$$J = \sum_{k=0}^{k_f} \left(F_T(x(k), u(k)) \right).$$

The braking energy is not included, since in our case we do not consider energy regeneration. This can be easily added

by changing the cost function accordingly.
Finally, the train can be operated in the following two modes:

(1) **Fully automatic mode**: the controller can issue any value of u in the continuous compact interval $[-1,1]$. This is typical for example in driverless urban metro systems.
(2) **Driver assistance mode**: the controller assists a human driver with a suggested driving mode, which has to be selected among the following four possible ones:
   - *Acceleration:* $u = 1$;
   - *Coasting:* $u = 0$;
   - *Cruising:* the controller engages a cruise control system that keeps a constant speed, i.e., u is computed by an inner control loop in such a way that $F_T = F_R$ for positive slopes and $F_B = F_R$ for negative slopes;
   - *Braking:* $u = -1$.

In our research we develop a MPC approach for the described problem. When resorting to optimization-based techniques like MPC, the main challenge in both operational modes is the computational burden, due to model nonlinearity and rather long prediction horizon. For example, in our simulation study the track is discretized in 220 position values. Considering the full horizon and all control moves as free variables, the resulting problem is a medium-size NLP. In the driver assistance mode, complexity is even larger, since the optimal control problem becomes a nonlinear integer program with 220 decision variables, each one with four possible values. To deal with this issue, we consider the use of input parametrization strategies, as explained in detail in Section 3.

2.1 Problem abstraction

The control problem described above can be cast in a rather standard form:

$$\min_{\boldsymbol{u}} \sum_{k=0}^{k_f} \ell(x(k),u(k)) \tag{3a}$$
subject to
$$x(k+1) = f(x(k),u(k)) \tag{3b}$$
$$u(k) \in U, k = 0,\dots,k_f-1 \tag{3c}$$
$$x(k) \in X(k), k = 1,\dots,k_f \tag{3d}$$
$$x(0) = x_0 \tag{3e}$$
$$x(k_f) \in X_f \tag{3f}$$

where $x \in \mathbb{X} \subset \mathbb{R}^n$ is the system state, x_0 is the initial condition, $u \in \mathbb{U} \subset \mathbb{R}^m$ is the input, $f(x,u) : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$ is a known nonlinear mapping representing the discrete-time system dynamics, and $\ell(x,u) : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ is a stage cost function defined by the designer according to the control objective. The symbol $\boldsymbol{u} = \{u(0),\dots,u(k_f-1)\} \in \mathbb{R}^{mk_f}$ represents the sequence of current and future control moves to be applied to the plant. The sets $X(k) \subset \mathbb{X}$ and $U \subset \mathbb{U}$ represent the state and input constraints, and the set $X_f \subset \mathbb{X}$ the terminal state constraints. Throughout the paper, we consider the following assumptions.

Assumption 1 *$X(k)$, U and X_f are compact.*

Recall that a continuous function $a : [0,\infty) \to [0,\infty)$ is a $\mathscr{K}$-function ($a \in \mathscr{K}$) if it is strictly increasing and $a(0) = 0$.

Assumption 2 *Functions f and ℓ enjoy the following continuity properties, where $a_{x_f}, a_{u_f}, a_{x_\ell}, a_{u_\ell} \in \mathscr{K}$:*

$$\|f(x^1,u) - f(x^2,u)\| \le a_{x_f}\left(\|x^1 - x^2\|\right), \forall x^1,x^2 \in \mathbb{X}, u \in \mathbb{U}$$
$$\|f(x,u^1) - f(x,u^2)\| \le a_{u_f}\left(\|u^1 - u^2\|\right), \forall u^1,u^2 \in \mathbb{U}, x \in \mathbb{X}$$
$$\|\ell(x^1,u) - \ell(x^2,u)\| \le a_{x_\ell}\left(\|x^1 - x^2\|\right), \forall x^1,x^2 \in \mathbb{X}, u \in \mathbb{U}$$
$$\|\ell(x,u^1) - \ell(x,u^2)\| \le a_{u_\ell}\left(\|u^1 - u^2\|\right), \forall u^1,u^2 \in \mathbb{U}, x \in \mathbb{X}.$$

In Assumption 2 and in the remainder of this paper, any vector norm $\|\cdot\|$ can be considered. Assumptions (1)-(2) are reasonable in most real-world applications, and they hold in the train control problem considered here.

3 Input parametrization strategies and nominal SPPC

To solve (3) and additionally include a feedback control strategy, we resort to Nonlinear Model Predictive Control (NMPC), however with two differences as compared to the standard formulation. First, since in our problem the terminal space k_f is fixed, the resulting strategy features a shrinking horizon rather than a receding one. Indeed, here the goal is to satisfy the terminal constraint X_f in the required finite time, and not asymptotically as usually guaranteed by a receding horizon approach. Second, we adopt an input parametrization strategy, motivated by applications like the one described in Section 2, featuring large values of k_f and/or possible integer optimization variables. The theoretical question we address next is about a sufficient condition on the input parametrization to guarantee recursive feasibility. To this end, we introduce the notion of transferability.

3.1 Input parametrization strategy and transferability

We consider a parametrization strategy that is generally dependent on k. For a given $k \in \mathbb{Z}_{[0,k_f-1]}$, let $N(k)$ be the number of parameters to be optimized, denoted with $\boldsymbol{\theta}_k = \{\theta(1),\dots,\theta(N(k))\} \in \mathbb{R}^{N(k)}$. Moreover, for a generic vector variable y, let us denote with $y(j|k)$ the future value at instant $j+k$, predicted at instant k. We call a *parametrization strategy* $\mathscr{P}$ a sequence of functions $g_k(\boldsymbol{\theta}_k, j)$ that link, at each step k and for each $j = 0,\dots,k_f-k-1$, the optimization variables $\boldsymbol{\theta}_k$ with the predicted input $u(j|k)$:

$$\begin{aligned} \mathscr{P} &= \{g_k\}, k = 0\dots,k_f-1 \\ g_k &: \mathbb{R}^{N(k)} \times \mathbb{N} \to \mathbb{U} \\ u(j|k) &= g_k(\boldsymbol{\theta}_k, j) \end{aligned}$$

Next, we define the transferability property. In practice, a transferable strategy is such that one can always recover, at instant $k+1$, the tail of the input sequence predicted at instant k, through suitable values of the optimization variables $\boldsymbol{\theta}_{k+1}$. This property is clearly sufficient for recursive

feasibility in nominal conditions, i.e., when the model and the system dynamics coincide. In [11], parametrizations of a multirate nonlinear predictive control strategy are considered, which enjoy a similar property.

Definition 1 *(Transferability) A parametrization strategy $\mathscr{P}$ is transferable if for any $k \in \{0,\dots,k_f-1\}$, any $\bar{\boldsymbol{\theta}}_k \in \mathbb{R}^{N(k)}$ and any $j \in \{0,\dots,k_f-k-1\}$, $\exists\, \bar{\boldsymbol{\theta}}_{k+1}$ such that $g_{k+1}(\bar{\boldsymbol{\theta}}_{k+1}, j) = g_k(\bar{\boldsymbol{\theta}}_k, j+1)$.*

Having defined the transferability property, in the remainder we consider the following assumption.

Assumption 3 *The chosen parametrization strategy $\mathscr{P}$ is transferable.*

We present next two parametrization strategies enjoying transferability.

3.1.1 Shrinking horizon move blocking parametrization

This is a modification of the well-known move-blocking strategy (see, e.g., [6]), adapted to the case of shrinking horizon. The prediction horizon is divided in intervals, each one featuring a fixed input value. Let us denote with L the maximum length of each interval. In particular, each interval contains exactly L blocked moves, except for the first one, which can contain a number between 1 and L of blocked input vectors as the index k increases. In this way, for a given value of $k \in \mathbb{Z}_{[0,k_f-1]}$, the number $N(k)$ of optimization variables is equal to $N(k) = m\left\lceil \frac{k_f-k}{L} \right\rceil$, where $\lceil . \rceil$ denotes the ceiling operation to the closest integer, see Fig. 2 for a graphical representation. The vector of optimization variables $\boldsymbol{\theta}_k$ is defined as $\boldsymbol{\theta}_k \doteq \left[\theta(1)^T, \dots \theta(N(k))^T\right]^T$, where $\theta(i) \in \mathbb{R}^m$ is the input value applied in the $i-$th interval. At each k, the values of $u(j|k)$ are then computed as $u(j|k) = g_k(\boldsymbol{\theta}_k, j) \doteq \theta\left(\left\lfloor \frac{j+k-\lfloor \frac{k}{L} \rfloor L}{L} \right\rfloor + 1\right)$. Transferability is easily checked by taking $\boldsymbol{\theta}_{k+1}$ equal to $\boldsymbol{\theta}_k$, if $N(k+1) = N(k)$, or $\boldsymbol{\theta}_{k+1}$ equal to $\boldsymbol{\theta}_k$ without the first m entries, if $N(k+1) = N(k)-1$, i.e., $\boldsymbol{\theta}_{k+1} = [\mathbf{0}_{N(k+1)\times m}\, \boldsymbol{I}_{N(k+1)\times N(k+1)}]\boldsymbol{\theta}_k$.

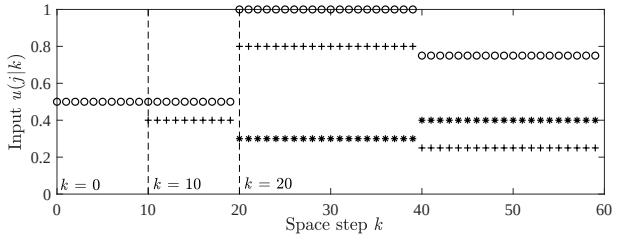

Fig. 2. Example of shrinking horizon move blocking parametrization with $L = 20$, $m = 1$ and $k_f = 60$: possible trajectories of predicted (scalar) input at instants $k = 0$ ('$\circ$'), $k = 10$ ('+'), and $k = 20$ ('$*$'). When $k = 20$, $N(k)$ reduces from 3 to 2.

Remark 1 *Another transferable move-blocking approach features initially a constant value of $N(k) = N(1)$, and the number of blocked input values in each interval is adjusted as k increases, up until the value $k = k_f - N(1)$ is reached, after which each predicted input vector is a free variable and $N(k)$ shrinks at each k. This strategy has the advantage to retain more degrees of freedom in the optimization as k approaches its final value.*

3.1.2 Shrinking horizon switching parametrization

This second strategy is effective in problems with discrete optimization variables, such as the driver assistance scenario described in Section 2. The idea is to first split the whole prediction horizon into a finite number of sectors. In each sector, a pre-defined switching sequence is imposed, and the corresponding switching instants are optimization variables. In this way, we retain a continuous nonlinear program, thus improving the computational efficiency, while still providing the controller with enough degrees-of-freedom to optimize the predicted system behavior. Let us indicate the total number of sectors with $s_n \in \mathbb{N}$. The $i-$th sector has length Γ_i, such that $\sum_{i=1}^{s_n} \Gamma_i = s_f$. The switching parametrization is best explained resorting to our train control application. The choice of sectors can be carried out by considering prior information such as the presence of constant velocity limits and of slopes and track curvature changes, see, e.g., Fig. 3. The chosen switching sequence, $\boldsymbol{u}_{d_i}$, features four phases:

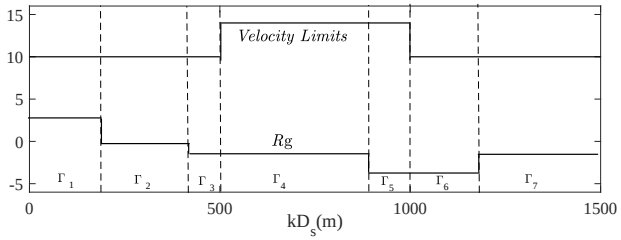

Fig. 3. Example of sector choice for a track with $s_f = 1500$ and s_n= 7. Possible sectors based on similar characteristics, such as velocity limits and R_g values, are depicted.

$$\begin{aligned} \boldsymbol{u}_{d_i} &= \{\ u_{(i,1)},\ u_{(i,2)},\ u_{(i,3)},\ u_{(i,4)}\ \} \\ &= \{\quad 1,\ \ u_{CR},\quad 0,\quad -1\ \},\ \forall i \in [1,\dots,s_n] \end{aligned} \tag{4}$$

where $u_{(i,\kappa)}$ is the input issued in the $\kappa-$th phase of the $i-$th sector ($\kappa \in \{1,2,3,4\}$) and u_{CR} is the input applied in the cruising mode (see Section 2). The optimization variables are the switching space instants, or more precisely the length $\delta s_{(i,\kappa)}$ of each phase within each switching sequence. These variables must be compliant with the following linear constraints:

$$\begin{aligned} 0 \le \delta s_{(i,\kappa)} &\le \Gamma_i \\ \sum_{\kappa=1}^{4} \delta s_{(i,\kappa)} &= \Gamma_i. \end{aligned} \tag{5}$$

As an example, $(\delta s_{(i,1)}, \delta s_{(i,2)}, \delta s_{(i,3)}, \delta s_{(i,4)}) = (0,0,\Gamma_i,0)$ corresponds to the train coasting throughout sector i, and so on. At each $k \in \mathbb{Z}_{[1,k_f]}$, the set of indexes identifying the

current and future sectors, from the current position kD_s until the end of the track s_f, is given by $\{i : \underline{i}(k)+1 \le i \le s_n\}$, where

$$\underline{i}(k) \doteq \begin{cases} \max_{\bar{i}\ge 1} \bar{i} \ \text{ s.t. } \sum_{i=1}^{\bar{i}} \Gamma_i < kD_s, & \text{if } \ \Gamma_1 < kD_s \\ 0, & \text{otherwise} \end{cases}$$

The number $N(k)$ of free variables to be computed corresponds to the number of remaining sectors, equal to $(s_n - \underline{i}(k))$, times the number of modes in each sector, i.e., four in our example. Therefore, we have $N(k) = 4(s_n - \underline{i}(k))$. We denote the vector of optimization variables with $\boldsymbol{\theta}_k \doteq \{\theta(1),\dots,\theta(N(k))\}^T = \{\delta s_{(\underline{i}(k)+1,1)},\dots,\delta s_{(\underline{i}(k)+1,4)},\dots,\delta s_{(s_n,4)}\}^T \in \mathbb{R}^{N(k)}$, and we define the switching parametrization strategy $g_k(\boldsymbol{\theta}_k, j)$ as:

$$u(j|k) = g_k(\boldsymbol{\theta}_k, j) \doteq u_{(\hat{i}(j,k),\hat{\kappa}(\boldsymbol{\theta}_k,j,k))},$$

where (compare with (4))

$$\hat{i}(j,k) \doteq \min_{\bar{i}=1,\dots,s_n} \bar{i} \ \text{ s.t. } \sum_{i=1}^{\bar{i}} \Gamma_i \ge (k+j)D_s$$

and

$$\hat{\kappa}(\boldsymbol{\theta}_k, j, k) \doteq \min_{\overline{\kappa}=1,\dots,4} \overline{\kappa}$$
$$\text{s.t. } \sum_{i=1}^{\hat{i}(j,k)} \Gamma_i + \sum_{\kappa=1}^{\overline{\kappa}} \delta s_{(\hat{i}(j,k),\kappa)} \ge (k+j)D_s.$$

Evaluating the switching parametrization is very efficient, since it just amounts to finding, for each predicted instant $k+j$, the index of the corresponding sector and phase and then to apply the pre-defined driving mode from (4). Transferability of this strategy is checked by taking, at time $k+1$, the same future switching intervals as those issued at k.

3.2 Nominal Shrinking horizon Parametrized Predictive Control

At each step $k \in \mathbb{Z}_{[0,k_f-1]}$, we formulate the following Finite Horizon Optimal Control Problem (FHOCP):

$$\min_{\boldsymbol{\theta}_k} \sum_{j=0}^{k_f-k} \ell(x(j|k), u(j|k)) \tag{6a}$$
subject to
$$u(j|k) = g_k(\boldsymbol{\theta}_k, j),\ j = 0,\dots,k_f-k-1 \tag{6b}$$
$$x(j+1|k) = f(x(j|k), u(j|k)),\ j = 0,\dots,k_f-k-1 \tag{6c}$$
$$u(j|k) \in U,\ j = 0,\dots,k_f-k-1 \tag{6d}$$
$$x(j|k) \in X(k),\ j = 1,\dots,k_f-k \tag{6e}$$
$$x(0|k) = x(k) \tag{6f}$$
$$\boldsymbol{\theta}_k \in \Theta_k \tag{6g}$$
$$x(k_f-k|k) \in X_f \tag{6h}$$

where (6g) takes into account possible constraints on the optimization variables, such as (5). We denote with $\boldsymbol{\theta}_k^*$ a locally optimal solution of (6). Moreover, we denote with $\boldsymbol{x}^*(k) = \{x^*(0|k),\dots,x^*(k_f-k|k)\}$ and $\boldsymbol{u}^*(k) = \{u^*(0|k),\dots,u^*(k_f-1-k|k)\}$ the corresponding predicted sequences of state and input vectors, where $x^*(0|k) = x(k)$, $x^*(j+1|k) = f(x^*(j|k), u^*(j|k))$, and $u^*(j|k) = g_k(\boldsymbol{\theta}_k^*, j)$.

Algorithm 1 Nominal SPPC strategy

(1) At instant k, get $x(k)$ and solve (6)*;*
(2) Apply to the plant the input $u(k) = u^*(0|k) = g_k(\boldsymbol{\theta}_k^*, 0)$;
(3) Repeat from 1) at the next sampling instant.

Algorithm 1 defines the feedback control law $u(k) = \mu(x(k)) = u^*(0|k)$, and the resulting model of the closed-loop system is $x(k+1) = f(x(k), \mu(x(k)))$. Under Assumption 3, when no disturbance or model mismatch are present (hence the term "nominal" SPPC), recursive feasibility and satisfaction of the terminal constraint are established by construction. In the next section, we deal with the uncertain case by introducing two possible variations of Algorithm 1.

4 Relaxed SPPC approaches: algorithms and properties

To model the uncertainty/disturbance, we consider an additive term $d(k)$ acting on the input vector (the so-called matched uncertainty), and denote with $\tilde{u}(k) \doteq u(k) + d(k)$ the disturbance-corrupted input applied to the plant. In our motivating application, $d(k)$ accounts for the effects of uncertainty in the train mass, drivetrain specs, track slope and curvature, as well as space and input discretization and/or misapplication by the human operator in a driver assistance scenario.

Assumption 4 $\forall k \in \mathbb{Z}_{[0,k_f]}$, $\|d(k)\| \le \overline{d} \in (0, +\infty)$.

This assumption holds in most practical cases and in the considered train application as well. We indicate the perturbed state trajectory due to the presence of d as $\tilde{x}(k+1) = f(\tilde{x}(k), \tilde{u}(k))$, $k = 0,\dots,k_f$, with $\tilde{x}(0) = x(0)$. Referring to Section 3.2, the recursive feasibility property can be easily lost when $d(k) \neq 0$, due to the deviation of the perturbed trajectory from the nominal one. As commonly done in standard NMPC, to retain recursive feasibility we therefore soften the state constraints. For the sake of simplicity, here we consider softening only the terminal state constraint in (3f), i.e., $x(k_f) \in X_f$. We do so without loss of generality, since the results and approaches below can be extended to any state constraint in the control problem. This is realistic in the train control application where, as a matter of fact, the maximum velocity limits never pose feasibility issues, since the braking force is always large enough to decrease the train speed within its bound notwithstanding possible extra accelerations, caused by, e.g., uncertain track slope or lower mass than the assumed one.
However, relaxation does not guarantee that, in closed-loop operation, the operational constraints are satisfied, or even that the constraint violation is uniformly decreasing as the worst-case disturbance bound $\overline{d}$ gets smaller.

Let us denote the distance between a point x and a set X as $\Delta(x,X) \doteq \min_{y\in X} \|x-y\|$. Then, we want to derive a modified SPPC strategy with softened terminal state constraint that guarantees a property of the following form in closed loop:

$$\exists \beta \in \mathscr{K}, \text{s.t.}, \Delta(\tilde{x}(k_f), X_f) \leq \beta(\overline{d}). \tag{7}$$

That is, the distance between the terminal state and the terminal constraint set is bounded by a value that decreases uniformly to zero as $\overline{d} \to 0$. We introduce next a relaxed SPPC approach with a two-step constraint softening procedure, which we prove to enjoy property (7).

4.1 Two-step relaxed SPPC

At step k, we solve two optimization problems in sequence:

a) compute the smallest achievable distance between the terminal state and the terminal set, starting from the current perturbed state $\tilde{x}(k)$:

$$\underline{\gamma}(k) \doteq \min_{\boldsymbol{\theta}_k, \gamma} \ \gamma \tag{8a}$$
subject to
$$u(j|k) = g_k(\boldsymbol{\theta}_k, j),\ j = 0,\dots,k_f-k-1 \tag{8b}$$
$$x(j+1|k) = f(x(j|k), u(j|k)),\ j = 0,\dots,k_f-k-1 \tag{8c}$$
$$u(j|k) \in U,\ j = 0,\dots,k_f-k-1 \tag{8d}$$
$$x(j|k) \in X(k),\ j = 1,\dots,k_f-k \tag{8e}$$
$$x(0|k) = \tilde{x}(k) \tag{8f}$$
$$\boldsymbol{\theta}_k \in \Theta_k \tag{8g}$$
$$\Delta(x(k_f-k|k), X_f) \leq \gamma \tag{8h}$$

b) optimize the input sequence using the original cost function, and softening the terminal constraint by $\underline{\gamma}(k)$:

$$\min_{\boldsymbol{\theta}_k} \ \sum_{j=0}^{k_f-k} \ell(x(j|k), u(j|k)) \tag{9a}$$
subject to
$$u(j|k) = g_k(\boldsymbol{\theta}_k, j),\ j = 0,\dots,k_f-k-1 \tag{9b}$$
$$x(j+1|k) = f(x(j|k), u(j|k)),\ j = 0,\dots,k_f-k-1 \tag{9c}$$
$$u(j|k) \in U,\ j = 0,\dots,k_f-k-1 \tag{9d}$$
$$x(j|k) \in X(k),\ j = 1,\dots,k_f-k \tag{9e}$$
$$x(0|k) = \tilde{x}(k) \tag{9f}$$
$$\boldsymbol{\theta}_k \in \Theta_k \tag{9g}$$
$$\Delta(x(k_f-k|k), X_f) \leq \underline{\gamma}(k) \tag{9h}$$

By construction, both problems are always feasible. Moreover the second can be warm-started at a feasible point with the solution of the first. We denote with $\boldsymbol{\theta}_k^r$, $\boldsymbol{x}^r(k)$ and $\boldsymbol{u}^r(k)$ the sequences of decision variables, state and inputs corresponding to a (locally) optimal solution of (9):

$$\boldsymbol{x}^r(k) = \{x^r(0|k), \dots, x^r(k_f-k|k)\} \tag{10a}$$
$$\boldsymbol{u}^r(k) = \{u^r(0|k), \dots, u^r(k_f-1-k|k)\} \tag{10b}$$
where
$$x^r(0|k) = \tilde{x}(k) \tag{10c}$$
$$x^r(j+1|k) = f(x^r(j|k), u^r(j|k)) \tag{10d}$$
$$u^r(j|k) = g_k(\boldsymbol{\theta}_k^r, j) \tag{10e}$$

Algorithm 2 Two-stage relaxed SPPC strategy

(1) At instant k, get $\tilde{x}(k)$ and solve in sequence (8)-(9)*. Let $\boldsymbol{\theta}_k^r$ be the computed solution;*
(2) Apply to the plant the input $u(k) = u^r(0|k) = g_k(\boldsymbol{\theta}_k^r, 0)$;
(3) Repeat from 1) at the next sampling instant.

Algorithm 2 defines the feedback control law $u(k) = \mu^r(\tilde{x}(k)) \doteq u^r(0|k)$, and the resulting closed-loop dynamics are given by:

$$\tilde{x}(k+1) = f(\tilde{x}(k), \mu^r(\tilde{x}(k)) + d(k)). \tag{11}$$

We show next that the closed-loop system (11) enjoys a uniformly bounded accuracy property of the form (7), provided that the nominal SPPC problem (6) is feasible at $k = 0$.

Theorem 1 *Let Assumptions 1- 4 hold and let* (6) *be feasible at space step $k = 0$. Then, the terminal state $\tilde{x}(k_f)$ of system* (11) *enjoys property* (7) *with*

$$\Delta(\tilde{x}(k_f), X_f) \leq \beta(\overline{d}) = \sum_{k=0}^{k_f-1} \beta_{k_f-k-1}(\overline{d}) \tag{12}$$

where $\beta_0(\overline{d}) = a_{u_f}(\overline{d})$ and $\beta_k(\overline{d}) = a_{u_f}(\overline{d}) + a_{x_f}(\beta_{k-1}(\overline{d}))$, $k = 1,\dots,k_f-1$.

Proof 1 *See the Appendix.*

Theorem 1 indicates that Algorithm 2 yields $\Delta(\tilde{x}(k_f), X_f) = 0$ for $\overline{d} = 0$, and a uniformly increasing bound otherwise. The result is conservative, since in practice such a bound is much larger than $\Delta(\tilde{x}(k_f), X_f)$, yet it still provides a theoretical support to the use of Algorithm 2, which, on the other hand, can be solved rather efficiently, as it involves nominal predictions only. The theoretically possible alternatives would be to consider closed-loop predictions with some preliminary controller (to reduce conservativeness of the bound), and/or to design a robust predictive strategy. The first approach could actually fit in the presented setup with minor modifications, since it would entail a change of the model equations to include the preliminary controller and the analysis of its effects on the exogenous input d, e.g., with a small-gain assumption. The second approach might be very challenging from the computational point of view, since we are dealing with general nonlinear systems, and eventually the obtained guarantees could still be conservative. For these reasons, here we decided to opt for Algorithm 2, analyze its

qualitative behavior with Theorem 1, and evaluate the actual performance, which are indeed very promising, in accurate simulations of our real-world application in Section 5.

4.2 Multi-objective relaxed SPPC strategy

Multi-objective minimization is an alternative to the two-step approach:

$$\min_{\boldsymbol{\theta}_k,\gamma} \sum_{j=0}^{k_f-k} \ell(x(j|k),u(j|k)) + \omega\gamma \tag{13a}$$

subject to

$$u(j|k) = g_k(\boldsymbol{\theta}_k, j),\ j=0,\dots,k_f-k-1 \tag{13b}$$
$$x(j+1|k) = f(x(j|k),u(j|k)),\ j=0,\dots,k_f-k-1 \tag{13c}$$
$$u(j|k) \in U,\ j=0,\dots,k_f-k-1 \tag{13d}$$
$$x(j|k) \in X(k),\ j=1,\dots,k_f-k \tag{13e}$$
$$x(0|k) = \tilde{x}(k) \tag{13f}$$
$$\boldsymbol{\theta}_k \in \Theta_k \tag{13g}$$
$$\Delta(x(k_f-k|k),X_f) \le \gamma \tag{13h}$$

where ω is a positive weight on the scalar γ. Problem (13) can be solved in Algorithm (2) in place of problems (8)-(9). In this case, a trade-off between constraint relaxation and economic performance can be set by tuning ω. Regarding the guaranteed bounds on constraint violation, with arguments similar to those of [4], under Assumptions 1-2 one can show that, at each $k \in \mathbb{Z}_{[0,k_f-1]}$, for any $\varepsilon > 0$ there exists a finite value of ω such that the distance between the terminal state and the terminal set is smaller than $\underline{\gamma}(k_f-k-1)(\overline{d}) + \varepsilon$. Thus, with large-enough ω, one can recover the behavior obtained with the two-step relaxed SPPC approach (see also [5, Thm. 14.3.1]). The theoretical derivation is omitted for the sake of brevity, as it is a rather minor extension of the results of [4].

5 Simulation results

We tested the proposed strategies using the Alstom simulator CITHEL, featuring coupled mechanical, electric and thermal calculations of the full train with its electro-mechanical back-end (incl. detailed models of the drives and electric motors), able to provide accurate time and energy consumption predictions. The tests pertain to the section of the Amsterdam metro rail between Rokin and Central Station. The train parameters are (see (1)-(2)) $M = 142403\,\text{kg}$, $M_s = 131403\,\text{kg}$, $A = 3975.9\,\text{N}$, $B = 24.36\,\text{Nsm}^{-1}$, $C = 4.38\,\text{Ns}^2\text{m}^{-2}$ and $D = 800\,\text{Nm}$. The maximum traction and braking forces as a function of train speed, $F_{Tmax}(x_2)$, $F_{Bmax}(x_2)$, are presented in Fig. 4. The actual forces are then computed as $F_T(x(k),u(k)) = F_{Tmax}(x_2(k))u(k)$, and $F_B(x(k),u(k)) = F_{Bmax}(x_2(k))u(k)$. The following equations provide approximate mathematical models of these forces:

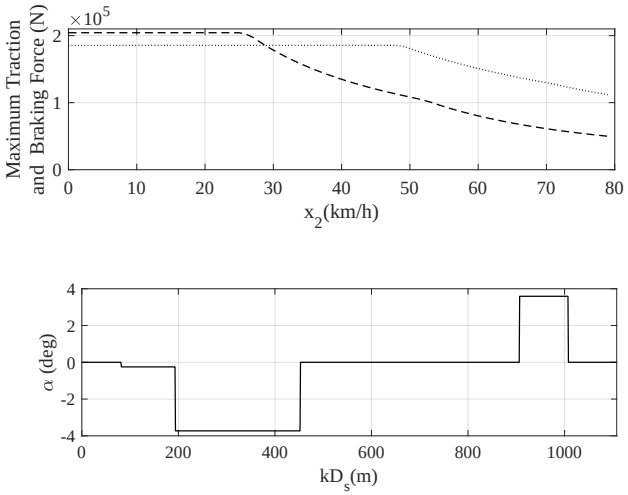

Fig. 4. Simulation parameters. Upper plot: maximum traction force F_{Tmax}(dashed) and braking force F_{Bmax} (dotted) for the considered metro train. Lower plot: slopes of the considered Amsterdam metro track segment.

$$F_T(x(k),u(k)) = \left\{ \begin{array}{ll} F_{Tmax}u(k), & \text{for } 0 \le 3.6x_2(k) \le 26 \\ \frac{F_{Tmax}u(k)}{(3.6x_2(k)-26)^{0.25}}, & \text{for } 27 \le 3.6x_2(k) \le 50 \\ \frac{F_{Tmax}u(k)}{2(3.6x_2(k)-50)^{0.05}}, & \text{for } 51 \le 3.6x_2(k) \le 80 \end{array} \right\}$$

$$F_B(x(k),u(k)) = \left\{ \begin{array}{ll} F_{Bmax}u(k), & \text{for } 0 \le 3.6x_2(k) \le 45 \\ \frac{F_{Bmax}u(k)}{(3.6x_2(k)-45)^{0.1}}, & \text{for } 46 \le 3.6x_2(k) \le 80 \end{array} \right\}$$

The considered track has zero curvature, slopes as plotted in Fig. 4, and velocity limits reported in Fig. 6. The track length is $s_f = 1106\,\text{m}$ and the nominal arrival time $t_f = 76\,\text{s}$. We take a sampling distance $D_s = 5\,\text{m}$, resulting in $k_f = 221$. Table 1 shows the computational times obtained with a non-parametrized NMPC approach and with the SPPC techniques. The results have been obtained using Matlab® `fmincon` and a laptop equipped with Intel® Core i7-6700Hq processor at 2.6 GHz and 8 GB of memory. All the functions were implemented in Matlab®. The non-parametrized NMPC was computationally feasible only in the fully automatic scenario, since in the driver assistance one at $k = 0$ an intractable optimization problem arises, with 221 discrete decision variables, each with four possible values. The relative comparison among the results in Table 1 indicates that the SPPC approaches achieve computational times that are two orders of magnitude smaller than the non-parametrized NMPC. We did not carry out any particular attempt to optimize the solver; the obtained results indicate that a real-world, fully optimized implementation at 100 Hz sampling rate is possible. The real-world implementation and testing of the approach are subject of ongoing development.
Fig. 5 presents the input courses obtained in both the fully automatic scenario (comparing NMPC without parametrization with move-blocking SPPC with $L = 20$ and $L = 45$) and in the driver assistance one, while the obtained velocity courses are shown in Fig. 6. The latter also presents

Table 1
Simulation results with CITHEL. Computational time per time step (s) of NMPC and SPPC approaches.

Non parametrized, continuous input	2.40
Non parametrized, discrete input	-
Move-blocking $L = 20$, two step	0.040
Move-blocking $L = 45$, two step	0.010
Move-blocking $L = 20$, multi-objective	0.025
Move-blocking $L = 45$, multi-objective	0.006
Switching sequence, two step	0.050
Switching sequence, multi-objective	0.030

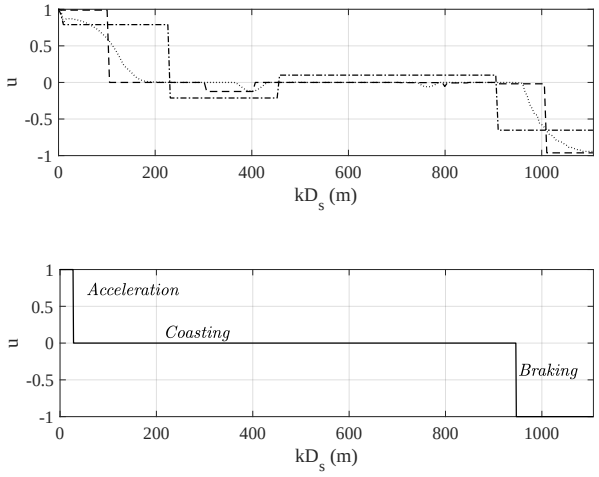

Fig. 5. CITHEL Simulation results. Upper plot: fully automated scenario, input handle vs. train position with no move-blocking (dotted line) and with move-blocking SPPC with $L = 20$ (dashed) and $L = 45$ (dash-dot). Lower plot: driver assistance scenario, input handle as a function of the train position obtained with the switching-sequence SPPC strategy, multi-objective approach.

the "all-out" solution (the one giving the shortest arrival time compatible with the system parameters and speed constraints) for comparison. The results show how the SPPC strategies favor coasting, thus exploiting the downhill slope of the track to save energy. The performance of SPPC strategies in terms of energy consumption and arrival time are summarized in Table 2, providing a comparison with the all-out solution and with Alstom current eco-drive strategy, based on genetic algorithm optimization. The SPPC strategies are able to reduce significantly the energy consumption, at the cost of a tolerable delay of 2 s with respect to the nominal arrival time t_f. Note that the use of CITHEL already introduces a mismatch with respect to the model (1) used in the predictive controllers. Moreover, we ran Monte Carlo simulations with parameter uncertainty of $\pm 10\%$ of each model parameter, always obtaining energy consumption and arrival times in the range of $\pm 0.5\%$ of those of Table 2. Finally, note that SPPC with move-blocking parametrization achieves slightly better results than with switching parametrization. This is reasonable, since the move-blocking one can still exploit the continuous input constraint set $[-1, 1]$, while the switching parametrization is bound to discrete driving modes.

Table 2
Comparison of performance obtained with Alstom simulation tool CITHEL, normalized with respect to Alstom "all-out" solution. The actual energy values are not provided for confidentiality reasons.

Strategy	Energy (%)	$x_1(k_f)$ (s)
All-out	100	73
Genetic algorithm Eco-drive	69	77
Non parametrized continuous NMPC	43	76
Move-blocking SPPC $L = 20$	45	78
Move-blocking SPPC $L = 45$	78	78
Switching-sequence SPPC	47	78

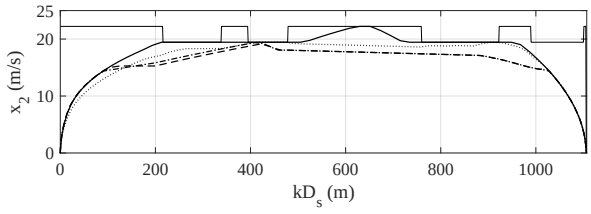

Fig. 6. CITHEL Simulation results. Velocity limits (thin solid line) and velocity profiles as a function of the train position for the "all-out" solution (thick solid line), Alstom genetic algorithm eco-drive (dotted line), move-blocking SPPC with $L = 20$ (dashed), and switching-sequence SPPC (dash-dot).

6 Conclusions and future work

We proposed a parametrized shrinking horizon predictive control approach to solve problems where a finite terminal instant and corresponding state constraint set are imposed. Under the assumption of transferable parametrization strategies, we derived a result supporting the use of relaxation to deal with matched uncertainty. We tested two relaxed approaches on a sophisticated train simulator by Alstom Rail Transport, proving improved energy consumption performance and feasible computation. The current research is aimed to implement the approach on a real train, and to extend it to the case of cooperative trains that are able to regenerate and exchange braking energy when connected to the same substation.

Acknowledgements

The authors thank Alstom rail transport for providing the model and track data employed in the simulation study. In particular, we would like to thank Davide Colombo and Piero Moia for their constant support while carrying out the CITHEL simulations.

Appendix - Proof of Theorem 1

Start at $k = 0$ and consider the relaxed optimized sequences $\boldsymbol{x}^r(k), \boldsymbol{u}^r(k)$ (10). For a sequence of disturbances $d(k), k =$

$0,\dots,k_f-1$, the corresponding *open-loop* input and state trajectories are:

$$\begin{aligned}&\tilde{u}(k) = u^r(0|k) + d(k), k = 0,\dots,k_f-1\\&\tilde{x}(0) = x^r(0|k)\\&\tilde{x}(k+1) = f(\tilde{x}(k),\tilde{u}(k)), k = 0,\dots,k_f-1\end{aligned}$$

From Assumption 2, we have:

$$\begin{aligned}&\|\tilde{x}(1) - x^r(1|0)\| = \|f(x(0),\tilde{u}(0)) - f(x(0),u^r(0|0))\| \leq\\&a_{u_f}(\|\tilde{u}(0) - u^r(0|0)\|) \leq a_{u_f}(\overline{d}) = \beta_0(\overline{d})\end{aligned}$$

And, for the perturbation 2-steps ahead:

$$\begin{aligned}&\|\tilde{x}(2) - x^r(2|0)\| = \|f(\tilde{x}(1),\tilde{u}(1)) - f(x^r(1|0),u^r(1|0))\| =\\&\|f(\tilde{x}(1),\tilde{u}(1)) - f(\tilde{x}(1),u^r(1|0)) +\\&f(\tilde{x}(1),u^r(1|0)) - f(x^r(1|0),u^r(1|0))\| \leq\\&a_{u_f}(\overline{d}) + a_{x_f}(\|\tilde{x}(1) - x^r(1|0)\|) \leq a_{u_f}(\overline{d}) + a_{x_f}(\beta_0(\overline{d})) = \beta_1(\overline{d}).\end{aligned}$$

By iterating until k_f-1 we obtain:

$$\|\tilde{x}(k_f) - x^r(k_f|0)\| \leq \beta_{k_f-1}(\overline{d}),$$

where $\beta_{k_f-1} \in \mathscr{K}$ since it is given by compositions and summations of class-$\mathscr{K}$ functions. Since the FHOCPs (6) is feasible, we have $\underline{\gamma}(0) = 0$, i.e., $\Delta(x^r(k_f|0,X_f) = 0$. Thus, by applying the triangular inequality to the distance operator:

$$\begin{aligned}\Delta(\tilde{x}(k_f),X_f) \;&\leq\; \|\tilde{x}(k_f) - x^r(k_f|0)\| + \Delta(x^r(k_f|0),X_f)\\&\leq\; \beta_{k_f-1}(\overline{d}).\end{aligned}$$

Now consider $k=1$ and the FHOCP (8). Under Assumption 3, the optimizer can be always initialized with a parameter vector $\tilde{\boldsymbol{\theta}}_1$ such that the tail of the previous optimal sequence $\boldsymbol{u}^r(0)$ is applied to the system, the corresponding minimum γ in (8) results to be upper bounded by $\beta_{k_f-1}(\overline{d})$ derived above.The optimal value $\underline{\gamma}(k)$ is therefore not larger than this bound as well:

$$\underline{\gamma}(1) \leq \beta_{k_f-1}(\overline{d}). \tag{14}$$

Now take the optimal sequences $\boldsymbol{x}^r(1)$ and $\boldsymbol{u}^r(1)$ computed by solving the FHOCP (9). By applying the same reasoning as we did for $k=0$, we have (compare with (14)):

$$\|\tilde{x}(k_f) - x^r(k_f|1)\| \leq \beta_{k_f-2}(\overline{d}).$$

Moreover, equation (14) implies:

$$\Delta(x^r(k_f|1),X_f) \leq \underline{\gamma}(1). \tag{15}$$

From (14)-(15) we have:

$$\begin{aligned}&\Delta(\tilde{x}(k_f),X_f) \leq \|\tilde{x}(k_f) - x^r(k_f|1)\| + \Delta(x^r(k_f|1),X_f) \leq\\&\beta_{k_f-2}(\overline{d}) + \beta_{k_f-1}(\overline{d}) = \sum_{k=0}^{1} \beta_{k_f-k-1}(\overline{d}).\end{aligned}$$

By applying recursively the up to $k = k_f-1$, we eventually obtain the bound (12). ■

References

[1] S. Aradi, T. Bécsi, and P. Gáspár. Design of predictive optimization method for energy-efficient operation of trains. In *European Control Conference (ECC)*, pages 2490–2495, 2014.

[2] F. Borrelli, A. Bemporad, and M. Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[3] R. Cagienard, P. Grieder, E.C. Kerrigan, and M. Morari. Move blocking strategies in receding horizon control. *Journal of Process Control*, 17(6):563–570, 2007.

[4] L. Fagiano and A.R.Teel. Generalized terminal state constraint for model predictive control. *Automatica*, 49(5):2622–2631, 2013.

[5] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, Inc., 1987.

[6] R. Gondhalekar and J. Imura. Recursive feasibility guarantees in move-blocking MPC. In *46th IEEE Conference on Decision and Control*, pages 1374–1379, 2007.

[7] R. Gondhalekar and J. Imura. Strong feasibility in input-move-blocking model predictive control. *IFAC Proceedings Volumes*, 40(12):816–821, 2007.

[8] R. Gondhalekar and J. Imura. Least-restrictive move-blocking model predictive control. *Automatica*, 46(7):1234–1240, 2010.

[9] R. Gondhalekar, J. Imura, and K. Kashima. Controlled invariant feasibility - a general approach to enforcing strong feasibility in MPC applied to move-blocking. *Automatica*, 45(12):2869–2875, 2009.

[10] G. Goodwin, M.M. Seron, and J.A. De Doná. *Constrained control and estimation: an optimisation approach*. Springer Science & Business Media, 2006.

[11] U. Halldorsson, M. Fikar, and H. Unbehauen. Nonlinear predictive control with multirate optimisation step lengths. *IEE Proceedings-Control Theory and Applications*, 152(3):273–284, 2005.

[12] S.E. Li, Z. Jia, K. Li, and B. Cheng. Fast online computation of a model predictive controller and its application to fuel economy–oriented adaptive cruise control. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1199–1209, 2015.

[13] D.Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50:2967–2986, 2014.

[14] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[15] G.M. Scheepmaker, R.M.P. Goverde, and L.G. Kroon. Review of energy-efficient train control and timetabling. *European Journal of Operational Research*, 257(2):355–376, 2017.

[16] R.C. Shekhar and J.M. Maciejowski. Robust variable horizon MPC with move blocking. *Systems & Control Letters*, 61(4):587–594, 2012.

[17] G. Valencia-Palomo, J.A. Rossiter, C.N. Jones, R. Gondhalekar, and B. Khan. Alternative parameterisations for predictive control: How and why? In *American Control Conference (ACC), 2011*, pages 5175–5180, 2011.

[18] M. Yu and L.T. Biegler. A stable and robust NMPC strategy with reduced models and nonuniform grids. *IFAC-PapersOnLine*, 49(7):31–36, 2016.