2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Model predictive control for tracking using artificial references: Fundamentals, recent results and practical implementation

P. Krupa[a], J. Köhler[b], A. Ferramosca[c], I. Alvarado[d], M.N. Zeilinger[b], T. Alamo[d], D. Limon[d]

***Abstract*— This paper provides a comprehensive tutorial on a family of Model Predictive Control (MPC) formulations, known as *MPC for tracking*, which are characterized by including an artificial reference as part of the decision variables in the optimization problem. These formulations have several benefits with respect to the classical MPC formulations, including guaranteed recursive feasibility under online reference changes, as well as asymptotic stability and an increased domain of attraction. This tutorial paper introduces the concept of using an artificial reference in MPC, presenting the benefits and theoretical guarantees obtained by its use. We then provide a survey of the main advances and extensions of the original linear *MPC for tracking*, including its non-linear extension. Additionally, we discuss its application to learning-based MPC, and discuss optimization aspects related to its implementation.**

***Index Terms*— Tutorial, Model predictive control, reference tracking, constrained control, artificial reference.**

I. Introduction

The main benefit of Model Predictive Control (MPC) [1], [2] is its ability to steer the system to a given reference without violating the constraints while minimizing some objective. A suitably designed MPC controller guarantees asymptotic stability of the closed-loop system to the desired reference as long as its optimization problem is feasible for the initial state of the system. This is typically achieved by means of a terminal cost function and a terminal invariant set, which must satisfy certain stabilizing conditions in a neighborhood of the reference.

However, it is not unusual for the desired reference to be changed during the online operation of the system, in order to steer it to a more convenient goal. One of the limitations of classical MPC is that changing the reference online may lead to an unfeasible MPC problem or to the loss of the stabilizing design guarantees. Furthermore, due to a lack of deep knowledge of the system and the uncertain nature of the change of reference, it is possible for the user to provide a desired reference that is unfeasible or non-attainable for the MPC controller, leading to the same problem.

This tutorial brings together recently-proposed MPC formulations that have been designed to address these issues. Recursive feasibility, even in the event of a sudden reference change, is guaranteed thanks to the idea of introducing an artificial reference as a decision variable of the MPC's optimization problem. This property is guaranteed regardless of the reference provided by the user, even if it violates the system constraints or if it is not a steady state of the system. Additionally, the resulting predictive controller provides asymptotic stability and (in general) a larger domain of attraction when compared to classical MPC.

The adoption of an artificial reference to deal with constraint satisfaction, even in the case of set-point changes, was introduced in the reference governor techniques, where a optimization-based reference filter is used in cascade with a stabilizing controller, see [3] and the references therein. An artificial reference was first integrated into a generalized predictive controller in [4], but without stability guarantees.

In 2008, the original stabilizing linear *MPC for tracking* formulation for tracking piece-wise affine references was introduced [5], [6], and in 2018 it was extended to non-linear MPC [7]. Based on these seminal papers, the idea of using an artificial reference has been extended to other classical MPC paradigms, such as tracking periodic references [8]–[14], economic MPC [8], [9], [15]–[18], robust control [19]–[23], stochastic MPC [24], [25], zone-control [26]–[28], output-tracking [29], path following and obstacle avoidance [30], [31], or distributed/coordinated control [14], [32]–[35].

Other articles have provided additional theoretical guarantees on the performance of *MPC for tracking* [36], extended the non-linear case to the use of a semidefinite cost functions [37], [38], presented a novel way of parameterizing the artificial reference as a harmonic signal [39]–[41], focused on a data-driven/learning approach [35], [37], [42], [43], or worked on the use of soft-constraints [44], [45].

One of the possible drawbacks of these formulations is the additional complexity of their optimization problems due to the inclusion of the artificial reference as decision variables. Thus, the role of optimization is fundamental to their success, since their theoretical and practical benefits are only useful in practice if they can compete with the classical MPC formulations. Over the years, several publications have tackled the problem of solving these formulations [12], [45]–[49], presenting solvers and toolboxes [50] tailored to them.

The *MPC for tracking* formulation and its extensions have

[a] Computer Science Group, Gran Sasso Science Institute (GSSI), L'Aquila, Italy (e-mail: pablo.krupa@gssi.it).
[b] Institute for Dynamic Systems and Control, ETH Zürich, Zürich CH-8092, Switzerland (e-mails: mzeilinger@ethz.ch, jkoehle@ethz.ch).
[c] Department of Management, Information and Production Engineering, University of Bergamo, Via Marconi 5, 24044 Dalmine (BG), Italy (e-mail: antonio.ferramosca@unibg.it).
[d] Department of Systems Engineering and Automation, Universidad de Sevilla, Seville, Spain (e-mails:ialvarado@us.es, talamo@us.es, dlm@us.es).

Daniel Limon, Ignacio Alvarado and Teodoro Alamo aknowledge the support of grant PID2022-141159OB-I00 funded by MICIU/AEI/ 10.13039/501100011033 and by ERDF/EU. Johannes Köhler was supported by the Swiss National Science Foundation under NCCR Automation (grant agreement 51NF40 180545). Pablo Krupa acknowledges the support of the MUR-PRO3 project on Software Quality and the MUR-PRIN project DREAM (20228FT78M).

979-8-3503-1632-2/24/$31.00 ©2024 IEEE

also been used in many (academic) case studies and applications, including automatic insulin injection for diabetes [28], aerospace rendezvous [51], [52], robotics [21], [35], [53], or economic building heat and ventilation [54], among others.

This article presents a comprehensive tutorial on the use of artificial references in MPC, starting from the presentation of the original linear *MPC for tracking* formulation [5], [6] in Section II, then presenting its main linear extensions and variations in Section III, and its non-linear extensions in Section IV. Section V presents the application of *MPC for tracking* to learning-based MPC, and Section VI discusses optimization aspects related to the implementation of these formulations, including guidelines on how to implement them using existing tools from academia. Final conclusions and remarks are presented in Section VII.

Notation: The set of natural numbers is denoted by $\mathbb{N}$, and $\mathbb{N}_{[a,b]}$ denotes the set of integers from the integers a to b, both included. I_n denotes the identity matrix of dimension n. We denote $\|x\|_Q \doteq \sqrt{x^\top Q x}$. Given two vectors x and y, $x \leq (\geq)\ y$ denotes componentwise inequalities. For vectors x_1 to x_N, $(x_1, x_2, \dots, x_N)$ denotes the column vector formed by their concatenation. By $\mathbb{S}^n_{\succ}$ and $\mathbb{S}^n_{\succeq}$ we denote the set of positive definite and positive semi-definite matrices in $\mathbb{R}^{n\times n}$, respectively. For a set $\mathbb{X} \subseteq \mathbb{R}^n$, $\text{int}(\mathbb{X})$ denotes the interior and for a given $\sigma \in \mathbb{R}$, $\sigma\mathbb{X} = \{\sigma x \in \mathbb{R}^n : x \in \mathbb{X}\}$. Given sets $\mathbb{X}, \mathbb{Y} \subseteq \mathbb{R}^n$, their Minkowski sum if defined by $\mathbb{X} \oplus \mathbb{Y} \doteq \{x + y : x \in \mathbb{X}, y \in \mathbb{Y}\}$ and their Pontryagin set difference by $\mathbb{X} \ominus \mathbb{Y} \doteq \{x : x \oplus \mathbb{Y} \subseteq \mathbb{X}\}$. A function $f\colon \mathbb{R} \to \mathbb{R}$ is of class $\mathcal{K}_\infty$ if it is continuous, strictly increasing, $f(0) = 0$ and $f(x) \to +\infty$ as $x \to +\infty$. We use $v_{(i)}$ to denote the i-th element of vector v.

II. Fundamentals of MPC for tracking using an artificial reference: the original formulation

As an introduction to the idea of using an artificial reference in MPC, let us start by considering the original linear *MPC for tracking* piece-wise affine references [5], [6]. As in classical linear MPC [1], [2], we consider a controllable system whose dynamics are described by a linear, discrete-time, time-invariant, state-space model

$$x(t+1) = Ax(t) + Bu(t), \tag{1a}$$
$$y(t) = Cx(t) + Du(t), \tag{1b}$$

where $x(t) \in \mathbb{R}^{n_x}$, $u(t) \in \mathbb{R}^{n_u}$, $y(t) \in \mathbb{R}^{n_y}$ are the state, input and output at sample time t, respectively, $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $C \in \mathbb{R}^{n_y \times n_x}$, $D \in \mathbb{R}^{n_y \times n_u}$. The control objective of the *MPC for tracking* formulation [5] is to steer (1) to a desired reference setpoint $y_r \in \mathbb{R}^{n_y}$ while satisfying the system constraints

$$(x(t), u(t)) \in \mathcal{Z}, \forall t, \tag{2}$$

where $\mathcal{Z} \subseteq \mathbb{R}^{n_x+n_u}$ is a closed convex polyhedron that contains the origin in its interior. The asymptotic convergence properties of the MPC formulations presented in this article require the following notion of *(strictly) rechable setpoints*.

Definition 1 (Reachable setpoints)**.** *The sets of (strictly) reachable steady states and setpoint of system* (1) *constrained by* (2)*, for a given $\sigma \in [0, 1)$, are given by*

$$\mathcal{Z}_s \doteq \{(x, u) \in \sigma\mathcal{Z} : x = Ax + Bu\} \subseteq \text{int}(\mathcal{Z}), \tag{3a}$$
$$\mathcal{Y}_s \doteq \{Cx + Du : (x, u) \in \mathcal{Z}_s\}\,. \tag{3b}$$

In classical MPC, the above control objective is achieved by first computing the optimal steady-state $(x_r, u_r) \in \mathcal{Z}_s$ satisfying $y_r = Cx_r + Du_r$ and then posing the finite-horizon optimal control problem

$$\min_{\mathbf{x},\mathbf{u}} \sum_{k=0}^{N-1} \ell(x_k, u_k, x_r, u_r) + V_f(x_N, x_r) \tag{4a}$$
$$\text{s.t. } x_0 = x(t), \tag{4b}$$
$$x_{k+1} = Ax_k + Bu_k,\ k \in \mathbb{N}_{[0,N-1]}, \tag{4c}$$
$$(x_k, u_k) \in \mathcal{Z},\ k \in \mathbb{N}_{[0,N-1]}, \tag{4d}$$
$$x_N \in \mathcal{X}_f, \tag{4e}$$

where $\mathbf{x} \doteq (x_0, x_1, \dots, x_N)$, $\mathbf{u} \doteq (u_0, u_1, \dots, u_{N-1})$ are the predicted states and control inputs, respectively, along the prediction horizon N; the stage cost function is given by

$$\ell(x_k, u_k, x_r, u_r) \doteq \|x_k - x_r\|_Q^2 + \|u_k - u_r\|_R^2, \tag{5}$$

with $Q \in \mathbb{S}^{n_x}_{\succeq}$, $R \in \mathbb{S}^{n_u}_{\succ}$, the terminal cost is given by $V_f(x_N, x_r) \doteq \|x_N - x_r\|_P^2$, with $P \in \mathbb{S}^{n_x}_{\succ}$; and the terminal set $\mathcal{X}_f$ is assumed to contain the desired reference x_r. Let $\mathbf{x}^* = (x_0^*, \dots, x_N^*)$, $\mathbf{u}^* = (u_0^*, \dots, u_{N-1}^*)$ be the optimal solution of (4). The control law of (4) is $u(t) = u_0^*$.

It is well known that, under a suitable design of the terminal ingredients $V_f(\cdot)$ and $\mathcal{X}_f$, the MPC formulation (4) steers the system to the desired setpoint if $y_r \in \mathcal{Y}_s$ and the initial state $x(0)$ belongs to the feasibility region of (4) [1]. The above statement highlights two possible drawbacks of (4):

i) If y_r is changed online, then the terminal set $\mathcal{X}_f$ must be recomputed as a suitable admissible invariant set containing the new steady-state x_r, which could be computationally expensive. Additionally, this could lead to the loss of feasibility of the MPC controller.

ii) The reference y_r must belong to the set of reachable setpoints $\mathcal{Y}_s$ (Definition 1). In a practical setting, a lack of deep knowledge of the system could result in the selection of a reference $y_r \notin \mathcal{Y}_s$.

To solve these issues, the *MPC for tracking* formulation [5] introduces new decision variables $(x_a, u_a) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$ that act as an *artificial reference*, leading to

$$\min_{\substack{\mathbf{x},\mathbf{u},\\ x_a, u_a}} \sum_{k=0}^{N-1} \ell(x_k, u_k, x_a, u_a) + V_f(x_N, x_a) + V_o(y_a - y_r) \tag{6a}$$
$$\text{s.t. } x_0 = x(t), \tag{6b}$$
$$x_{k+1} = Ax_k + Bu_k,\ k \in \mathbb{N}_{[0,N-1]}, \tag{6c}$$
$$(x_k, u_k) \in \mathcal{Z},\ k \in \mathbb{N}_{[0,N-1]}, \tag{6d}$$
$$y_a = Cx_a + Du_a, \tag{6e}$$
$$(x_a, u_a) \in \mathcal{Z}_s, \tag{6f}$$
$$(x_N, x_a, u_a) \in \mathcal{X}_a, \tag{6g}$$

where $V_o(\cdot) : \mathbb{R}^{n_y} \to \mathbb{R}$ is the *offset cost function*, $\mathcal{X}_a \subseteq \mathbb{R}^{2n_x+n_u}$ is the terminal *invariant set for tracking* [5, §2.2], and (6f) requires the artificial reference to be a strictly admissible steady state of the system. The intuition behind this formulation is to include an artificial reference that acts as a *proxy* of the desired reference. The formulation steers the system state towards the artificial reference while also penalizing the distance between this artificial reference and the desired one. The idea is for both the system state and artificial reference to converge towards the desired reference as t increases.

We note that (6f) considers *strictly* admissible steady states to avoid a possible loss of controllability that could result from the presence of active constraints on the artificial reference [5], [55]. However, in practice σ can be chosen arbitrarily close to 1, so the artificial reference can be arbitrarily close to the system constraints.

Under a suitable design of its ingredients, as detailed in the following assumption, the *MPC for tracking* formulation (6) provides recursive feasibility and asymptotic stability to the following *optimal reachable reference*.

Definition 2 (Optimal reachable reference)**.** *The* optimal reachable reference *of the MPC for tracking controller* (6) *for a given $y_r \in \mathbb{R}^{n_y}$ and $\sigma \in [0, 1)$ is given by*

$$y_a^\circ = \arg\min_{y \in \mathcal{Y}_s} V_o(y - y_r).$$

Assumption 1. *Let $Q \in \mathbb{S}^{n_x}_{\succeq}$ be such that $(Q^{1/2}, A)$ is observable; N be greater or equal to the controllability index of* (1)*; $K \in \mathbb{R}^{n_x \times n_u}$ be a state-feedback control gain such that $A + BK$ is Schur; and $P \in \mathbb{S}^{n_x}_{\succ}$ satisfy*

$$(A + BK)^\top P(A + BK) - P \preceq -(Q + K^\top RK).$$

Let $V_o(\cdot)$ be a convex, positive-definite, subdifferentiable function with $V_o(0) = 0$, such that the optimal solution of $\min_{y\in\mathcal{Y}_s} V_o(y - y_r)$ is unique. Let $\mathcal{X}_a \subseteq \mathbb{R}^{2n_x+n_u}$ be such that, for all $(x, x_a, u_a) \in \mathcal{X}_a$:

$$(x, K(x - x_a) + u_a) \in \mathcal{Z},$$
$$(Ax + B(K(x - x_a) + u_a), x_a, u_a) \in \mathcal{X}_a.$$

We note that the invariant set for tracking $\mathcal{X}_a$ can be computed as the (maximal) positive invariant set of a system extended by the artificial reference, as detailed in [5, §2.2], that is, following standard procedures for the computation of terminal invariant sets for MPC controllers. Additionally, a common choice is to take V_o as a quadratic function $\|y_a - y_r\|_S^2$, with $S \in \mathbb{S}^{n_y}_{\succ}$, since in this case problem (6) is a quadratic programming (QP) problem. The following theorems formalize the theoretical properties of (6). We refer the reader to [5], [6] for their proofs.

Theorem 1 (Recursive feasibility)**.** *Let $\hat{\mathbf{x}} \doteq (\hat{x}_0, \ldots, \hat{x}_N)$, $\hat{\mathbf{u}} \doteq (\hat{u}_0, \ldots, \hat{u}_{N-1})$ be any feasible solution of* (6) *for the current state $x(t)$. Then, problem* (6) *is feasible for the successor state $Ax(t) + B\hat{u}_0$ for any value of $y_r \in \mathbb{R}^{n_y}$.*

The first major benefit obtained thanks to the use of an artificial reference is that recursive feasibility is guaranteed in the event of a reference change, even if the new reference does not belong to the set of reachable references $\mathcal{Y}_s$. Notice that the set of constraints in (6) does not depend on the reference y_r. Additionally, there is no need to recompute any of the ingredients of (6) when the reference is changed.

Theorem 2 (Asymptotic stability)**.** *Consider* (1) *subject to* (2) *controlled by* (6)*, and let Assumption 1 be satisfied. Assume $x(0)$ belongs to the feasibility region of* (6)*. Then, the closed-loop system is stable, fulfills the system constraints, and asymptotically converges to the optimal reachable reference y_a° given by Definition 2.*

The second major benefit is that asymptotic stability to an admissible steady state is guaranteed even if the reference $y_r \notin \mathcal{Y}_s$, i.e., if the reference is unreachable. In particular, the consequence of Theorem 2 is that the closed-loop system will asymptotically converge to y_r if $y_r \in \mathcal{Y}_s$. Otherwise, it will converge to the *optimal reachable reference* y_a° provided by Definition 1, i.e., to the strictly reachable setpoint that minimizes its distance to y_r as measured by the offset cost function $V_o(\cdot)$. We note that the MPC formulation (6) cannot converge to references with active constraints due to the use of $\sigma < 1$ in $\mathcal{Y}_s$, see (6f). However, once again, σ can be taken arbitrarily close to 1, so in practice the closed-loop system can converge to steady states that are arbitrarily close to the system constraints.

Finally, another benefit of (6) is its increased domain of attraction when compared to (4). This is best seen by considering the following commonly-used particularization of (6), which considers a terminal equality constraint:

$$\min_{\substack{\mathbf{x},\mathbf{u},\\ x_a,u_a}} \sum_{k=0}^{N-1} \ell(x_k, u_k, x_a, u_a) + \|x_a - x_r\|_T^2 + \|u_a - u_r\|_S^2 \tag{7a}$$

$$\text{s.t. } x_0 = x(t), \tag{7b}$$
$$x_{k+1} = Ax_k + Bu_k,\ k \in \mathbb{N}_{[0,N-1]}, \tag{7c}$$
$$(x_k, u_k) \in \mathcal{Z},\ k \in \mathbb{N}_{[0,N-1]}, \tag{7d}$$
$$(x_a, u_a) \in \mathcal{Z}_s, \tag{7e}$$
$$x_N = x_a, \tag{7f}$$

where $T \in \mathbb{S}^{n_x}_{\succ}$ and $S \in \mathbb{S}^{n_u}_{\succ}$ model a quadratic offset cost $V_o(\cdot)$. This formulation avoids the need to compute a terminal invariant set for tracking $\mathcal{X}_a$ and does not require the use of a terminal cost $V_f(\cdot)$. Additionally, it uses a quadratic offset cost function, leading to a QP problem with a simple structure that can be exploited by first-order optimization algorithms (see Section VI).

An intuitive way of seeing the increase in the domain of attraction that is typically obtained when using an artificial reference is to compare (7) with its classical MPC counterpart, i.e., with (4) taking (4e) as $x_N = x_r$. In (7) all states $x(0)$ that can reach *any* setpoint $y \in \mathcal{Y}_s$ in N steps belong to its feasibility region, and therefore to its domain of attraction (as per Theorems 1 and 2). On the other hand, only states $x(0)$ that can reach the setpoint y_r in N steps may belong to the domain of attraction of the classical MPC counterpart

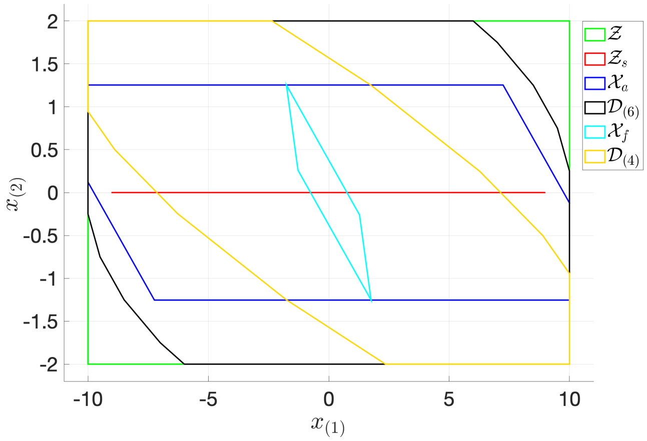

Fig. 1: Domain of attraction of standard MPC (4) and MPC for tracking (6). Legend: $\mathcal{Z}$ are the state constraints; $\mathcal{Z}_s$ the states that belong to the manifold of steady states; $\mathcal{X}_a$ the invariant set for tracking of (6); $\mathcal{X}_f$ the terminal invariant set of (4); $\mathcal{D}_{(4)}$ and $\mathcal{D}_{(6)}$ the domains of attraction of (4) and (6), respectively.

of (7). Similar reasoning can be used when comparing the domains of attraction of (4) and (6), as illustrated by the following example.

Example 1 (Domain of attraction)**.** *We consider* (1) *with*

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix},\ C = \begin{bmatrix} 1 & 0 \end{bmatrix},\ D = \begin{bmatrix} 0 \end{bmatrix},$$

and constraints $|x_{(1)}| \leq 10$*,* $|x_{(2)}| \leq 2$*,* $|u| \leq 0.5$*. We also consider the classical MPC formulation* (4) *and the* MPC for tracking *formulation* (6)*, taking* $Q = 100 I_2$*,* $R = 1$*, and* $N = 5$*. We take* P *as the solution of the discrete algebraic Riccati equation and compute the terminal invariant sets of the MPC controllers for the control gain* K *of the LQR controller. The terminal set of* (4) *considers the origin as the reference. Fig. 1 shows the terminal invariant sets and domains of attraction of both controllers. The results illustrate how the domain of attraction of the* MPC for tracking *formulation can be significantly larger. This is mostly a result of the difference between both terminal sets. Indeed, note that the terminal set of* (4) *is "centered" around the reference, whereas the terminal invariant set for tracking of* (6) *is "centered" around the set of feasible steady states* $\mathcal{Z}_s$*, leading to a much larger set. We note that as the prediction horizon increases, the difference between their domains of attraction becomes smaller.*

As a final note, it can be shown that every admissible steady state $(x_s, u_s) \in \mathcal{Z}_s$ is in the interior of the domain of attraction of (6). This means that *MPC for tracking* can stabilize the system for any initial state that is a strictly admissible steady state, which is very interesting from a practical point of view.

Local optimality: A drawback of the inclusion of the artificial reference is that the *MPC for tracking* formulation (6) looses the local optimality property of classical MPC with respect to the unconstrained LQR controller [56]. That is, the optimal solution of (6) will generally differ from the optimal solution of (6) with the additional constraint $y_a = y_r$ (i.e., its classical MPC counterpart) [57]. However, this local optimality property can be recovered by a proper selection of the offset cost function $V_o(\cdot)$. In particular, local optimality is recovered if V_o satisfies $\alpha_1 \|y\| \leq V_o(y) \leq \alpha_2 \|y\|$, $\forall y \in \mathcal{Y}_s$, for a sufficiently large $\alpha_1 > 0$. We refer the reader to [6, §4] for additional details.

III. Extensions of MPC for tracking

We now present extensions of the original linear *MPC for tracking* formulation [5], [6], to some of the main MPC paradigms: robust MPC, tracking periodic references, and economic MPC. We focus on presenting formulations that provide a light-weight introduction into the use of artificial references within these MPC paradigms. Therefore, we focus on linear MPC formulations that resemble the original *MPC for tracking* presented in Section II. Section IV will present non-linear extensions of *MPC for tracking*.

A. Robust MPC

The use of *MPC for tracking* to robustly control systems with parametric and/or additive uncertainties has been explored using different approaches [19]–[23]. We recall here the approach presented in [19], since it more closely resembles the problem setup presented in Section II. This robust *MPC for tracking* formulation is based on the classical tube-based robust MPC for regulation presented in [58].

Let us now consider system (1) constrained by (2) with an additive state disturbance, i.e., we take (1a) as

$$x(t+1) = Ax(t) + Bu(t) + w(t), \tag{8}$$

where $w(t) \in \mathbb{R}^{n_x}$ is the unknown state disturbance at time t, which is assumed to be contained within a known compact convex polyhedron $\mathcal{W}$. The control objective is to steer the system to a desired reference setpoint $y_r \in \mathbb{R}^{n_y}$ while satisfying the system constraints (2) despite the presence of the unknown state disturbance.

The tube-based robust *MPC for tracking* formulation from [19] is given by

$$\min_{\substack{\mathbf{x},\mathbf{u},\\ x_a, u_a}} \sum_{k=0}^{N-1} \ell(x_k, u_k, x_a, u_a) + V_f(x_N, x_a) + V_o(y_a - y_r) \tag{9a}$$

$$\text{s.t. } x(t) \in x_0 \oplus \phi_K, \tag{9b}$$

$$x_{k+1} = Ax_k + Bu_k,\ k \in \mathbb{N}_{[0,N-1]}, \tag{9c}$$

$$(x_k, u_k) \in \mathcal{Z}_{\phi_K},\ k \in \mathbb{N}_{[0,N-1]}, \tag{9d}$$

$$y_a = Cx_a + Du_a, \tag{9e}$$

$$(x_a, u_a) \in \mathcal{Z}_w, \tag{9f}$$

$$(x_N, x_a, u_a) \in \mathcal{X}_w, \tag{9g}$$

where its ingredients satisfy the following assumption and its control law is given by $u(t) = K(x(t) - x_0^*) + u_0^*$.

Assumption 2. *The ingredients of* (9) *satisfy:*

(i) $Q \in \mathbb{S}_{\succeq}^{n_x}$*,* $R \in \mathbb{S}_{\succ}^{n_u}$*, and* $(Q^{1/2}, A)$ *is observable.*

(ii) $V_o(y_a - y_r) = \|y_a - y_r\|_S^2$*, with* $S \in \mathbb{S}_{\succ}^{n_y}$*.*

(iii) *$K, \bar{K} \in \mathbb{R}^{n_x \times n_u}$ are such that $A_K \doteq A + BK$ and $A_{\bar{K}} \doteq A + B\bar{K}$ are Schur stable matrices.*

(iv) *ϕ_K is a robust positive invariant set for the uncertain system $x(t+1) = A_K x(t) + w(t)$, i.e., a set satisfying $A_K \phi_k \oplus \mathcal{W} \subseteq \phi_k$.*

(v) *$\mathcal{Z}_{\phi_K} \doteq \mathcal{Z} \ominus (\phi_K \times K\phi_K)$.*

(vi) *$\mathcal{Z}_w \doteq \{(x,u) \in \sigma \mathcal{Z}_{\phi_K} : x = Ax + Bu\}$, for some given $\sigma \in [0,1)$ that is taken arbitrarily close to 1.*

(vii) *$V_f(x_N, x_a) = \|x_N - x_a\|_P^2$, with $P \in \mathbb{S}_{\succ}^{n_x}$ satisfying*

$$P - (A + B\bar{K})^\top P (A + B\bar{K}) = Q + \bar{K}^\top R \bar{K}.$$

(viii) *Set $\mathcal{X}_w \subseteq \mathbb{R}^{2n_x + n_u}$ is an invariant set for tracking for system (1) constrained by $(x(t), u(t)) \in \mathcal{Z}_{\phi_K}$ for the control gain $\bar{K}$, i.e., for all $(x, x_a, u_a) \in \mathcal{X}_w$:*

$$(x, \bar{K}(x - x_a) + u_a) \in \mathcal{Z}_{\phi_K},$$
$$(Ax + B(\bar{K}(x - x_a) + u_a), x_a, u_a) \in \mathcal{X}_w.$$

Notice that formulation (9) is similar to (6). The main difference if that (9d) considers constraints that are tightened by means of the robust positive invariant set ϕ_K, whose computation, as well as the computation of a suitable K, can be done following standard results [19, §7], [58]. Additionally, the initial predicted state x_0 is constrained to a region around $x(t)$, and the terminal invariant set for tracking $\mathcal{X}_w$ considers the tightened constraints $\mathcal{Z}_{\phi_K}$ and a terminal control gain $\bar{K}$ which can be different to the one used for the computation of ϕ_K. Indeed, the two gains K and $\bar{K}$ can be designed independently, so as to improve the performance of the controller [19, §7]. We refer the reader to [59] for guidelines on the design of the ingredients of (9).

Under the satisfaction of Assumption 2, the robust *MPC for tracking* formulation (9) provides the following recursive feasibility and asymptotic stability guarantees.

Theorem 3 ([19], Theorem 1)**.** *Let Assumption 2 hold and suppose that (9) is feasible for $x(0)$. Then, the closed-loop system formed by (8)-(9) satisfies:*

i) *Recursive feasibility: Problem (9) remains feasible for all t, regardless of the realization of $w(t) \in \mathcal{W}$ and even if y_r is changed online.*

ii) *Asymptotic stability: If $y_r \in \mathcal{Y}_w$, where*

$$\mathcal{Y}_w \doteq \{Cx + Du : (x,u) \in \mathcal{Z}_w\}$$

is the set of reachable setpoint considering the tightened constraints $\mathcal{Z}_w$ (cf. Definition 1), then the output $y(t)$ converges asymptotically to the set $y_r \oplus (C + DK)\phi_K$. Otherwise, output $y(t)$ asymptotically converges to the set $y_a^\circ \oplus (C+DK)\phi_K$, where y_a° is the optimal reachable setpoint considering the tightened constraints $\mathcal{Z}_w$, i.e.,

$$y_a^\circ = \arg \min_{y \in \mathcal{Y}_w} \|y - y_r\|_S^2.$$

B. Tracking periodic references

Periodic systems and references naturally occur in many applications, such as water distribution networks [60], HVAC system [54], robotics [61], [62], or aerospace [51], [63]. This has motivated the application of MPC to control periodic systems and/or to track periodic references [62]–[66]. In this subsection, we present the extension of the *MPC for tracking* piece-wise affine reference setpoints presented in Section II to this control paradigm. The extension naturally follows by considering a periodic artificial reference.

We focus on the periodic *MPC for tracking* formulation presented in [10], which can be viewed as an extension of the linear *MPC for tracking* with terminal equality constraint (7) to the problem of tracking a periodic reference. That is, the control objective is to steer the output $y(t)$ of system (1) to a given periodic reference trajectory $y_r(\cdot) \in \mathbb{R}^{n_y}$ with known period $\tau \in \mathbb{N}$, i.e., a trajectory satisfying $y_r(t) = y_r(t + \tau)$, $\forall t$, while satisfying the system constraints (2). The notion of reachable setpoints presented in Definition 1 readily extends to the notion of a reachable periodic reference.

Definition 3 (Reachable periodic trajectory)**.** *For a given $\sigma \in [0,1)$ and period $\tau \in \mathbb{N}$, the set of (strictly) reachable periodic trajectories of (1) constrained by (2) is*

$$\mathcal{Y}_s^\tau \doteq \left\{ y(\cdot) \in \mathbb{R}^{n_y} : \begin{array}{l} y(t) = Cx(t) + Du(t) \\ x(t+1) = Ax(t) + Bu(t) \\ x(t) = x(t{+}\tau),\ u(t) = u(t{+}\tau) \\ (x(t), u(t)) \in \sigma\mathcal{Z} \end{array}, \forall t \right\}.$$

The periodic *MPC for tracking* formulation makes use of a periodic artificial reference $(\mathbf{x_a}, \mathbf{u_a})$, with period τ, where $\mathbf{x_a} \doteq (x_{a,0}, \dots, x_{a,\tau})$, $\mathbf{u_a} \doteq (u_{a,0}, \dots, u_{a,\tau-1})$, leading to

$$\min_{\substack{\mathbf{x}, \mathbf{u}, \\ \mathbf{x_a}, \mathbf{u_a}}} \sum_{k=0}^{N-1} \ell(x_k, u_k, x_{a,k}, u_{a,k}) + V_p(\mathbf{y_a}, \mathbf{y_r}) \tag{10a}$$
$$\text{s.t. } x_0 = x(t), \tag{10b}$$
$$x_{k+1} = Ax_k + Bu_k,\ k \in \mathbb{N}_{[0,N-1]}, \tag{10c}$$
$$(x_k, u_k) \in \mathcal{Z},\ k \in \mathbb{N}_{[0,N-1]}, \tag{10d}$$
$$x_{a,k+1} = Ax_{a,k} + Bu_{a,k},\ k \in \mathbb{N}_{[0,\tau-1]}, \tag{10e}$$
$$(x_{a,k}, u_{a,k}) \in \sigma\mathcal{Z},\ k \in \mathbb{N}_{[0,\tau-1]}, \tag{10f}$$
$$y_{a,k} = Cx_{a,k} + Du_{a,k},\ k \in \mathbb{N}_{[0,\tau-1]}, \tag{10g}$$
$$x_{a,0} = x_{a,\tau}, \tag{10h}$$
$$x_N = x_{a,N}, \tag{10i}$$

where $\mathbf{y_a} \doteq (y_{a,0}, \dots, y_{a,\tau-1})$,

$$V_p(\mathbf{y_a}, \mathbf{y_r}) = \sum_{k=0}^{\tau-1} \|y_{a,k} - y_r(t+k)\|_S^2,$$

for $S \in \mathbb{S}_{\succ}^{n_y}$, is the offset cost function. Constraints (10e)-(10h) force the artificial reference $(\mathbf{x_a}, \mathbf{u_a})$ to describe a reachable periodic trajectory (Definition 3) for the given $\sigma \in [0,1)$, which, once again, is taken arbitrarily close to 1. Finally, (10i) is a terminal equality constraint that forces the predicted terminal state x_N to reach the artificial periodic reference.

The MPC controller (10) shares analogous recursive feasibility and asymptotic stability guarantees to the ones presented in Theorems 1 and 2, as shown in [10]. That is, if (10) is feasible for $x(0)$, then the closed-loop system remains feasible at future sample times, even if the reference trajectory $y_r(\cdot)$ is changed to a different periodic trajectory with the same period. Furthermore, if initially feasible, the closed-loop system converges to the optimal reachable periodic trajectory given by the optimal solution of

$$\min_{\mathbf{y_a}} \{V_p(\mathbf{y_a}, \mathbf{y_r}), \text{ s.t. } \mathbf{y_a} \in \mathcal{Y}_s^\tau\}.$$

That is, if $y_r(\cdot)$ describes a reachable periodic trajectory, for the σ used in (10), then the closed-loop system asymptotically converges to it, i.e., $\|y(t) - y_r(t)\| \to 0$ as $t \to +\infty$. Otherwise, the closed-loop system asymptotically converges to the closest reachable periodic trajectory, as measured by the offset cost function $V_p(\cdot)$.

Periodic system: The periodic *MPC for tracking* formulation (10) can be easily adapted to consider a time-varying periodic system, with period $\tau \in \mathbb{N}$,

$$x(t+1) = A(t)x(t) + B(t)u(t), \tag{11}$$

where $A(t) = A(t+\tau)$, $B(t) = B(t+\tau)$, $\forall t$. In particular, we refer the reader to [9] for an *MPC for tracking* formulation that considers this paradigm, which naturally arises in several practical applications and has been widely researched within the MPC community [63], [66].

C. Harmonic MPC for tracking

One of the main disadvantages of (10) is that its number of constraints and decision variables grows with the period $\tau \in \mathbb{N}$ of the reference, since the system dynamics and constraints must be imposed along the complete period of the artificial reference, as shown in (10e)-(10f). This can result in long solve times if the period is too long. This issue inspired the development of the *MPC for tracking* formulation presented in [39], [40], named Harmonic MPC (HMPC). This formulation was originally developed for tracking piece-wise affine setpoints, although it has recently been extended to the problem of tracking harmonic (aka sinusoidal) reference trajectories [41]; a particular class of periodic reference that arises in many practical applications. We now present its original setpoint tracking formulation, since it provides a less notation-heavy introduction of the main idea behind it.

HMPC is an *MPC for tracking* formulation that considers system (1) taking the constraints (2) as

$$\mathcal{Z} = \left\{(x,u) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} : \underline{y} \leq Cx + Du \leq \overline{y}\right\}$$

for some $\underline{y}, \overline{y} \in \mathbb{R}^{n_y}$ satisfying $\underline{y} < 0 < \overline{y}$. As in the *MPC for tracking* formulation (7), HMPC considers the problem of tracking a given setpoint (x_r, u_r) and makes use of a terminal equality constraint. However, its artificial reference is given by a harmonic signal, whose value at each prediction step k is given by

$$x_{h,k} = x_e + x_s \sin(\omega k) + x_c \cos(\omega k), \tag{12a}$$
$$u_{h,k} = u_e + u_s \sin(\omega k) + u_c \cos(\omega k), \tag{12b}$$

where $\omega > 0$ is its frequency and its parameters $x_e, x_s, x_c \in \mathbb{R}^{n_x}$, $u_e, u_s, u_c \in \mathbb{R}^{n_u}$ are included as decision variables in the optimization problem.

Using the notation $\mathbf{x_h} = (x_e, x_c, x_c)$, $\mathbf{u_h} = (u_e, u_s, u_c)$, $y_e = Cx_e + Du_e$, $y_s = Cx_s + Du_s$, $y_c = Cx_c + Du_c$, the optimization problem of HMPC is given by

$$\min_{\substack{\mathbf{x},\mathbf{u},\\ \mathbf{x_h},\mathbf{u_h}}} \sum_{k=0}^{N-1} \ell(x_k, u_k, x_{h,k}, u_{h,k}) + V_h(\mathbf{x_h}, \mathbf{u_h}, x_r, u_r) \tag{13a}$$

$$\begin{aligned}
\text{s.t. } & x_0 = x(t), && (13b)\\
& x_{k+1} = Ax_k + Bu_k,\ k \in \mathbb{N}_{[0,N-1]}, && (13c)\\
& \underline{y} \leq Cx_k + Du_k \leq \overline{y},\ k \in \mathbb{N}_{[0,N-1]}, && (13d)\\
& x_N = x_e + x_s \sin(\omega N) + x_c \cos(\omega N), && (13e)\\
& x_e = Ax_e + Bu_e, && (13f)\\
& x_s \cos(\omega) - x_c \sin(\omega) = Ax_s + Bu_s, && (13g)\\
& x_s \sin(\omega) + x_c \cos(\omega) = Ax_c + Bu_c, && (13h)\\
& \sqrt{y_{s(i)}^2 + y_{c(i)}^2} \leq \sigma \overline{y}_{(i)} - y_{e(i)},\ i \in \mathbb{N}_{[1,n_y]} && (13i)\\
& \sqrt{y_{s(i)}^2 + y_{c(i)}^2} \leq y_{e(i)} - \sigma \underline{y}_{(i)},\ i \in \mathbb{N}_{[1,n_y]}, && (13j)
\end{aligned}$$

where $\sigma \in [0, 1)$ is taken arbitrarily close to 1, the terminal equality constraint (13e) enforces the terminal predicted state x_N to reach $x_{h,N}$, i.e., the value of the artificial harmonic reference at prediction time $k = N$, and the offset cost function is given by

$$\begin{aligned} V_h(\cdot) = &\|x_e - x_r\|_T^2 + \|u_e - u_r\|_S^2 \\ &+ \|x_s\|_{T_h}^2 + \|x_c\|_{T_h}^2 + \|u_s\|_{S_h}^2 + \|u_c\|_{S_h}^2, \end{aligned}$$

with $T \in \mathbb{S}_{\succ}^{n_x}$, $T_h \in \mathbb{S}_{\succ}^{n_x}$, $S \in \mathbb{S}_{\succ}^{n_u}$, $S_h \in \mathbb{S}_{\succ}^{n_u}$, and T_h, S_h diagonal. In [40] it is shown that the equality constraints (13f)-(13h) impose the satisfaction of the system dynamics on (12), whereas the second order cone constraints (13i)-(13j) impose the satisfaction of the system constraints. Notice that, even though the HMPC formulation makes use of a periodic artificial reference, the number of constraints of (13) does not depend on its period, determined by the choice of ω.

The terms of V_h corresponding to T and S are analogous to the terms from the offset cost function in (7), in that they penalize a measure of the distance with the desired reference (x_r, u_r). The other terms of V_h penalize the magnitude of the $\sin$ and $\cos$ terms of the artificial harmonic reference. The idea is that the "center" (x_e, u_e) will converge towards the reference (x_r, u_r), whereas the other parameters of the artificial harmonic reference will converge to 0. Indeed, as shown in [40], the HMPC formulation (13) shares the same recursive feasibility guarantees and asymptotic stability properties of (7). In particular, it is shown that the HMPC formulation asymptotically converges to the optimal reachable reference of (7), i.e., to the admissible steady state $(x^\circ, u^\circ) \in \sigma\mathcal{Z}$ that minimizes $\|x^\circ - x_r\|_T^2 + \|u^\circ - u_r\|_S^2$.

The main benefit of the HMPC formulation (13), when compared to (7), is its increased domain of attraction and performance when working with small prediction horizons, as reported in [39]. This benefit is particularly apparent when working with systems that have integrator states and/or slew-rate constraints on the control inputs.

Tracking harmonic references: HMPC (13) has been recently extended in [41] to the problem of tracking a harmonic reference trajectory of frequency $\omega > 0$, i.e., a reference trajectory whose value at time t is given by

$$x_r(t) = x_{re} + x_{rs}\sin(\omega t) + x_{rc}\cos(\omega t),$$
$$u_r(t) = u_{re} + u_{rs}\sin(\omega t) + u_{rc}\cos(\omega t),$$

for some parameters $x_{re}, x_{rs}, x_{rc} \in \mathbb{R}^{n_x}$, $u_{re}, u_{rs}, u_{rc} \in \mathbb{R}^{n_u}$. The formulation is very similar to (13), as well as its properties, which guarantee asymptotic stability to the admissible harmonic reference that minimizes its offset cost function. Since the complexity of its optimization problem does not depend on the period of the reference, this formulation could be rather useful when dealing with harmonic/sinusoidal references with large periods.

D. Economic MPC for tracking

Optimal operation and control of processes is a widely studied field in the overall control community, as well as in the MPC community, where it receives the name of *economic* MPC [67]. There are several formulations of economic MPC that consider artificial references and/or deal with the problem of changing economic criteria [9], [15]–[18], as well as several applications, e.g., [8], [54]. In this subsection, we recall the economic *MPC for tracking* presented in [17], since it considers the problem setup from Section II applied to the problem of minimizing an economic performance metric, given by the function $\ell_{\rm e}(x,u,\theta) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$, where $\theta \in \mathbb{R}^{n_\theta}$ are the parameters that characterize the economic criterion. This economic cost leads to the following notion of the optimal economic setpoint.

Definition 4 (Optimal economic setpoint)**.** *The optimal economic setpoint (x_*, u_*) of system (1) subject to (2) for a given economic cost $\ell_e(\cdot)$ parameterized by θ is given by*

$$(x_*, u_*) = \arg \min_{(x,u)\in\mathcal{Z}_s} \ell_e(x,u,\theta). \tag{14}$$

The economic cost is assumed to satisfy the following standard assumption [17], [68].

Assumption 3. *The economic cost $\ell_e(\cdot)$ is locally Lipschitz continuous at (x_*, u_*), i.e., there exist scalars $\epsilon, L_e > 0$ such that $\forall\theta$, $\forall(x,u)\in\mathcal{Z}$, $\|(x,u)-(x_*,u_*)\| \leq \epsilon$ implies*

$$\|\ell_e(x,u,\theta) - \ell_e(x_*,u_*,\theta)\| \leq L_e \|(x,u)-(x_*,u_*)\|.$$

Additionally, for each $\theta \in \mathbb{R}^{n_\theta}$ there exists a multiplier λ such that (x_*, u_*) is the unique solution of

$$\min_{(x,u)\in\mathcal{Z}} \ell_r(x,u,\theta).$$

where, additionally, the rotated *stage cost*

$$\ell_r(x,u,\theta) \doteq \ell_e(x,u,\theta) + \lambda^\top(x-(Ax+Bu)) - \ell_e(x_*,u_*,\theta)$$

can be lower bounded by two $\mathcal{K}_\infty$ functions α_x, α_u as

$$\ell_r(x,u,\theta) \geq \alpha_x(\|x-x_*\|) + \alpha_u(\|u-u_*\|),\ \forall(x,u)\in\mathcal{Z}.$$

The economic *MPC for tracking* from [17] adds an artificial reference (x_a, u_a) to the classical economic MPC formulation from [68], leading to

$$\min_{\substack{\mathbf{x},\mathbf{u},\\ x_a,u_a}} \sum_{k=0}^{N-1} \ell_{\rm e}(x_k - x_a + x_*, u_k - u_a + u_*, \theta) + V_o(x_a,u_a,x_*,u_*) \tag{15a}$$

$$\text{s.t. } x_0 = x(t), \tag{15b}$$
$$x_{k+1} = Ax_k + Bu_k,\ k\in\mathbb{N}_{[0,N-1]}, \tag{15c}$$
$$(x_k, u_k) \in \mathcal{Z},\ k\in\mathbb{N}_{[0,N-1]}, \tag{15d}$$
$$(x_a, u_a) \in \mathcal{Z}_s, \tag{15e}$$
$$x_N = x_a, \tag{15f}$$

where the offset cost $V_o(\cdot)$ satisfies the following assumption. Note that (15) requires (x_*, u_*). Therefore, problem (14) must be solved online every time θ is changed.

Assumption 4. *$V_o(\cdot)$ is a positive definite function such that (x_*, u_*) is the unique minimizer of*

$$\min_{(x,u)\in\mathcal{Z}_s} V_o(x,u),$$

and there exists a scalar $\gamma > 0$ such that, $\forall(x,u)\in\mathcal{Z}_s$,

$$V_o(x,u,x_*,u_*) - V_o(x_*,u_*,x_*,u_*) \geq \gamma\|x - x_*\|.$$

Formulation (15) provides recursive feasibility guarantees following the same arguments as the original *MPC for tracking* formulation. Indeed, notice that the constraints of (15) coincide with the constraints of the *MPC for tracking* with terminal equality constraint (7). Additionally, the formulation has the following asymptotic stability guarantee.

Theorem 4 ([17, Theorem 1])**.** *Let Assumptions 3-4 be satisfied. Assume that N is greater than the controllability index of system (1), that there exists a control gain K_e such that $A + BK_e$ has null eigenvalues, and that $(x_*, u_*) \in \mathcal{Z}_s$. Then, for a sufficiently large γ, the closed-loop system formed by (1) and (15) asymptotically converges to x_*.*

We refer the reader to [17, Eq. (21)] for the lower bound that γ must satisfy for the claim in the previous theorem to hold, which depends on the values of L_e, N and K_e. We note that, in the case of the economic *MPC for tracking* formulation (15), asymptotic stability to a reachable steady state is only guaranteed if $(x_*, u_*) \in \mathcal{Z}_s$. Therefore, asymptotic stability is not always guaranteed under online changes of θ, although recursive feasibility is always guaranteed.

Periodic economic control: Economic MPC for tracking has also been extended to periodic control, either due to the use of a periodic economic cost [8], [69] or to its application to a periodic system (11), as in [9].

E. Other extensions of MPC for tracking

As mentioned in the introduction, linear *MPC for tracking* has also been extended to many other control paradigms. Some notable examples include:

- *Stochastic MPC* [24], [25], which consider an uncertain system and allow a certain level of constraint violation with the use of *chance constraints* [70], typically leading to less conservative controllers when compared to robust MPC.

- *Zone-control* [26], where the control objective is to steer the system output to some given set (aka zone), instead of to a specific setpoint. Within the reference set, there is no preference between one point or another. The use of an artificial reference provides recursive feasibility if the reference set is changed online. The extension of zone control *MPC for tracking* to the robust paradigm was presented in [27]. A practical example is the problem of blood-glucose control presented in [28].
- *Collision avoidance* [31], where obstacles are considered, leading to non-convex constraints.
- *Distributed/coordinated control* [14], [32]–[34], where various (coupled) agents that share a common objective are controlled by a distributed optimization algorithm.

IV. MPC FOR TRACKING WITH NON-LINEAR SYSTEMS

In this section, we discuss how the *MPC for tracking* formulations naturally generalize to non-linear system dynamics. We first explain the setpoint tracking formulation for non-linear systems and discuss the design of terminal ingredients, convexity, and theoretical properties (Sec. IV-A). Then, we discuss extensions to periodic reference tracking problems (Sec. IV-B) and robust designs (Sec. IV-C).

A. Setpoint tracking

First, we explain the basic non-linear setpoint tracking formulation based on [7] and then address convexity and the design of terminal ingredients following [12], [29]. We consider non-linear discrete-time constrained systems

$$x(t+1) = f(x(t), u(t)), \tag{16a}$$
$$y(t) = h(x(t), u(t)), \tag{16b}$$
$$(x(t), u(t)) \in \mathcal{Z}, \forall t \in \mathbb{N}, \tag{16c}$$

with $\mathcal{Z}$ compact and continuous functions f, h. We define the set of (strictly) feasible steady-state and setpoints

$$\mathcal{Z}_s \doteq \left\{(x_a, u_a) \in \bar{\mathcal{Z}} : x_a = f(x_a, u_a)\right\} \tag{17a}$$
$$\mathcal{Y}_s \doteq \left\{h(x_a, u_a) : (x_a, u_a) \in \mathcal{Z}_s\right\}, \tag{17b}$$

with some user chosen set $\bar{\mathcal{Z}} \subseteq \text{int}(\mathcal{Z})$. Given a reference y_r, an optimal setpoint can be computed using

$$\min_{y_a \in \mathcal{Y}_s} V_o(y_a, y_r), \tag{18}$$

and a minimizer is denoted by $(x_a^\circ, u_a^\circ, y_a^\circ)$. Analogous to Problem (6), the MPC formulation is given by

$$\min_{\substack{\mathbf{x},\mathbf{u},\\ x_a,u_a}} \sum_{k=0}^{N-1} \ell(x_k, u_k, x_a, u_a) + V_f(x_N, x_a, u_a) + V_o(y_a, y_r) \tag{19a}$$
$$\text{s.t. } x_0 = x(t), \tag{19b}$$
$$x_{k+1} = f(x_k, u_k),\ k \in \mathbb{N}_{[0,N-1]}, \tag{19c}$$
$$(x_k, u_k) \in \mathcal{Z},\ k \in \mathbb{N}_{[0,N-1]}, \tag{19d}$$
$$y_a = h(x_a, u_a), \tag{19e}$$
$$(x_a, u_a) \in \mathcal{Z}_s, \tag{19f}$$
$$(x_N, x_a, u_a) \in \mathcal{X}_f. \tag{19g}$$

We solve Problem (19) at each time t and apply the input $u(t) = u_0^\star$ to system (16). For simplicity of exposition, we consider a quadratic cost ℓ (5) and $V_o(y_a, y_r) = \|y_a - y_r\|_S^2$, with Q, R, S positive definite, see also [7] for a more general setting. Similar to Assumption 1, we consider the following conditions on the terminal ingredients $V_f, \mathcal{X}_\text{f}$.

Assumption 5. *There exists a feedback $\kappa_f : \mathcal{X}_f \to \mathbb{R}^{n_u}$, such that for any steady-state $(x_a, u_a) \in \mathcal{Z}_s$ and any $x \in \mathbb{R}^{n_x}$: $(x, x_a, u_a) \in \mathcal{X}_f$, $u = \kappa_f(x, x_a, u_a)$ satisfies*

- *Constraint satisfaction: $(x, u) \in \mathcal{Z}$;*
- *Positive invariance: $(f(x, u), x_a, u_a) \in \mathcal{X}_f$;*
- *Control Lyapunov function:* $V_f(f(x, u), x_a, u_a) - V_f(x, x_a, u_a) \leq -\ell(x, u, x_a, u_a)$.

Theorem 5 (Recursive feasibility [7, Thm 1])**.** *Let Assumption 5 hold and suppose that Problem (19) is feasible at $t = 0$. Then, Problem (19) is feasible and the closed-loop system satisfies the constraints (16c) for all $t \in \mathbb{N}$, independent of y_r.*

To ensure stability and convergence to the optimal steady-state, we require the following additional conditions.

Assumption 6.

a) Convexity: The set $\mathcal{Y}_s$ is convex.

b) Uniqueness: There exists a unique Lipschitz continuous function $g : \mathcal{Y}_s \to \mathcal{Z}_s$, such that for any $y_a \in \mathcal{Y}_s$, $(x_a, u_a) = g(y_a) \in \mathcal{Z}_s$ satisfy $y_a = h(x_a, u_a)$.

c) Local upper bound on value function: There exists a constant $\epsilon > 0$, such that for any $(x_a, u_a) \in \mathcal{Z}_s$ and any $x \in \mathbb{R}^{n_x}$: $\|x - x_a\| \leq \epsilon$, Problem (19) is feasible and the optimal cost is quadratically bounded [12, Asm. 2].

Theorem 6 (Exponential stability [12, Thm. 8]))**.** *Let Assumptions 5–6 hold and suppose Problem (19) is feasible at $t = 0$. Then, the optimal steady-state x_a° is exponentially stable for the resulting closed-loop system.*

Next, we discuss the role and constructive satisfaction or relaxation of the conditions in Assumptions 5–6. Note that in the absence of Assumption 6, the closed-loop system may in general converge to some steady-state x_a, which is not the global minimizer of (18).

Design of terminal ingredients: The simplest design of terminal ingredients uses a terminal equality constraint, i.e., $x_N = x_a$. This trivially satisfies Assumption 5, while Assumption 6c) holds if the linearized dynamics around any steady-state are controllable and the prediction horizon is larger than the controllability index, see [7, Sec. III.A], [12, Prop. 4]. In [7], [12], a less restrictive terminal cost and set of the form

$$\mathcal{X}_f = \{(x, x_a, u_a) : V_f(x, x_a, u_a) \leq \alpha(x_a, u_a)\} \tag{20a}$$
$$V_f(x, x_a, u_a) = \|x - x_a\|_{P(x_a,u_a)}^2 \tag{20b}$$
$$\kappa_f(x, x_a, u_a) = u_a + K(x_a, u_a)(x - x_a) \tag{20c}$$

have been proposed with P, K, α parametrized by the artificial reference. The standard design of terminal ingredients for non-linear systems involves the linearization at a fixed steady-state, computing P, K based on the LQR, and then choosing a constant α [1]. To extend this design to MPC for tracking formulations, [7, App. B] suggest to partition $\mathcal{Z}_s$ and apply a similar design of each partition, resulting in piecewise constant parametrizations of P, K, α. This design was improved in [12] by deriving a continuous parametrization of P, K using gain-scheduling techniques and determining α implicitly using linear inequality constraints, see also (24) below.

While good terminal ingredients can significantly improve performance, it can also increases the overall complexity of the offline design. One attractive alternative is to remove or relax the requirements on $\mathcal{X}_f, V_f$. In particular, the terminal set constraint $\mathcal{X}_f$ can be removed if the terminal cost V_f or the prediction horizon N are chosen sufficiently large [7, Sec. III.B]. Furthermore, under a suitable cost-controllability condition, the terminal cost V_f can be removed if the prediction horizon N is chosen large enough [29]. This is particularly relevant for applications with non-holonomic vehicles (cf., e.g., [35]), as the linearized dynamics are not stabilizable but an exponential cost controllability condition can still be satisfied by choosing a non-quadratic stage cost, see [71] for details.

Convexity and obstacle avoidance: The (strongly convex) quadratic offset cost V_o in combination with $\mathcal{Y}_s$ convex (cf. Asm. 6a)) is crucial to ensure that the closed-loop system converges to the globally optimal setpoint y_a°. Notably, convexity is only required in the output space $\mathcal{Y}_s$, which is crucial as the steady-state manifold $\mathcal{Z}_s$ is often non-convex. Nonetheless, especially for motion-planning applications with collision avoidance constraints, $\mathcal{Y}_s$ is typically non-convex and hence the system may get stuck at a local minima. Different solutions have been proposed to address this problem, such as using a homeomorphism to a convex space to define a non-quadratic offset cost V_o [72] or choosing a large weighting V_o [73]. In [31], it was suggested to replace the non-convex constraints with corresponding penalties, which allows for simple implementation with (input-to-state) stability guarantees, see also [30, Sec. 5] for a related approach. Lastly, in [29, Prop. 2], a method is presented that ensures convergence to the globally optimal setpoint y_a° for any connected non-convex set $\mathcal{Y}_s$. This is achieved by defining the offset cost V_o implicitly as the shortest (feasible) distance using a continuous path γ:

$$\begin{aligned} &V_o(y_a, y_r) := \min_{\gamma}\ \text{length}(\gamma(\cdot)) \qquad (21)\\ &\text{s.t. } \gamma(0) = y_a,\ \gamma(1) = y_r,\ \gamma(s) \in \mathcal{Y}_s, \forall s \in [0,1]. \end{aligned}$$

By using a finite parametrization, this cost can be embedded in Problem (19) with additional decision variables and thus ensure convergence for arbitrary (connected) non-convex constraints $\mathcal{Y}_s$, see also [29, Rk. 3].

Uniqueness: The uniqueness condition (Asm. 6b)) ensures that not only the optimal output y_a° is unique, but also the corresponding state and input (x_a°, u_a°). This condition can be reduced to a so called trackability condition for the linearized system (cf. [7, Rk. 1]). However, even if this condition is not satisfied, the system will instead converge to the set of optimal steady-states.

B. Tracking periodic references

In the following, we discuss how the methodology for tracking periodic references (Sec. III-B) can be extended to non-linear systems based on [12]. On a conceptual level, this extensions is a relatively straightforward combination of the results for linear periodic tracking (Sec. III-B) and non-linear setpoint tracking (Sec. IV-A), see [12] for corresponding details. However, this indirectly comes with additional challenges regarding the design of terminal ingredients and in general an increased online computational demand.

Terminal ingredients: A method to design terminal ingredients for generic reference trajectories has been proposed in [12], [74]. This approach mirrors the design of continuously parametrized terminal ingredients in (20), which can be related to a gain-scheduling design for quasi-linear parameter varying systems. Simply choosing a terminal equality constraint is also quite attractive in the non-linear periodic case, as the offline design becomes more challenging.

Computational complexity and partially decoupled tracking-planning: Another issue inherent to the usage of periodic artificial references is the increased computational demand. While this issue is also present for the linear formulation (Sec. III-B), in the non-linear case this issue is exacerbated as the non-linear optimization problems are computationally more demanding. Furthermore, the Harmonic MPC parametrization (Sec. III-C) is difficult to generalize to non-linear system. Thus, we present a different methodology based on [12, Sec. 3.4] and [49, Sec. 4.4]. The main idea is to split the periodic tracking MPC (cf. Problem (10) for the linear case) into two problems:

- a periodic reference planner, which optimizes the artificial references $\mathbf{x_a}, \mathbf{u_a}$ to track the target $\mathbf{y_r}$;
- a trajectory tracking MPC that optimizes $\mathbf{u}$ to track the artificial reference $\mathbf{x_a}$.

The main idea is that the computationally intensive long horizon planner does not need to be re-optimized at every time t, while the trajectory tracking MPC requires a fast sampling rate to react to disturbances. The challenge is that running the planner and tracker independently could lead to references that cause feasibility issues in the terminal set constraint of the tracking MPC. This issue is addressed by adding additional constraints to the planning problem that consider the current tracking error on the terminal set, such that the usual candidate solution remains feasible in the tracker, see [12, Sec. 3.4] for details on a tractable formulation. As a result, a standard trajectory tracking MPC can be implemented at each time t, while a partially coupled planner can solve a larger optimization problem over a user chosen sampling period. The partial coupling ensures that both optimization problems remain recursively feasible. Furthermore, if a suitable terminal set is designed, then the reference $\mathbf{x_a}$ converges to the optimal target in finite time and the state x converges exponentially to the corresponding optimal trajectory $\mathbf{x}_{\mathbf{a}}^{\circ}$ [12, Prop. 13]. This strategy has been further refined in [75] to address (non-periodic) motion-planning problems in embedded autonomous mobile robots.

C. Robust tracking MPC

In the following, we focus on designing a robust formulation that ensures safe operation in the presence of bounded model-mismatch. This design leverages non-linear robust MPC formulations [76]–[78] which are integrated into the *MPC for tracking* formulation following [21]. We consider the perturbed dynamics

$$x(t+1) = f_w(x(t), u(t), w(t)),\ w(t) \in \mathcal{W} \subseteq \mathbb{R}^{n_w}, \quad (22)$$

with disturbances w and some known compact set $\mathcal{W}$ containing the origin. The dynamics f_w are assumed to be continuous and we denote the nominal dynamics by $f(x,u) \doteq f_w(x,u,0)$. Furthermore, we assume that the constraint set is characterized as

$$\mathcal{Z} = \left\{(x,u) : g_j(x,u) \leq 0,\ j \in \mathbb{N}_{[1,n_c]}\right\}$$

with Lipschitz continuous scalar functions g_j. In contrast to linear systems (Sec. III-A), the error propagation for non-linear systems also depends on the nominal dynamics, which makes exact propagation challenging. We focus on computationally efficient approaches that provide robust guarantees based on a simple offline computed constraint tightening [76]–[78]. To this end, we consider the following conditions on an offline computed stabilizing feedback $\kappa(x,z,v)$ and corresponding (incremental) Lyapunov function $V_\delta(x,z)$.

Assumption 7. *There exist a feedback $\kappa : \mathbb{R}^{n_x} \times \mathcal{Z} \to \mathbb{R}^{n_u}$, an incremental Lyapunov function $V_\delta : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}$ and constants $\rho \in [0,1)$, $\bar{w}, c_{\mathrm{l}}, c_{\mathrm{u}} > 0$, $c_j \geq 0$, $j \in \mathbb{N}_{[0,n_c]}$, such that for all $(z,v) \in \mathcal{Z}$, all $w \in \mathcal{W}$ and all $x \in \mathbb{R}^{n_x}$, $u = \kappa(x,z,v)$ satisfies:*

$$\begin{aligned} c_{\mathrm{l}}\|x-z\| \leq V_\delta(x,z) &\leq c_{\mathrm{u}}\|x-z\| && (23a)\\ V_\delta(f_w(x,u,w), f(z,v)) &\leq \rho V_\delta(x,z) + \bar{w}, && (23b)\\ g_j(x,u) - g_j(z,v) &\leq c_j V_\delta(x,z). && (23c)\end{aligned}$$

Furthermore, the following triangular inequality holds for any $x, z, \xi \in \mathbb{R}^{n_x}$:

$$V_\delta(x,\xi) + V_\delta(\xi,z) \geq V_\delta(x,z). \quad (23d)$$

Conditions (23a)–(23b) ensures that the feedback κ drives the state x exponentially to a neighborhood around any nominal trajectory z, which is called incremental stability. Inequalities (23c) follows naturally if g_j, κ are Lipschitz continuous, see [77, Prop. 5] for analytical formulas. General incremental Lyapunov functions V_δ and feedbacks κ can be constructed offline using (control) contraction metrics [77], [78].[1] In case of only mildly non-linear dynamics, a simple weighted norm $V_\delta(x,z) = \|x-z\|_M$ and linear feedback $\kappa(x,z,v) = v + K(x-z)$ can be constructed, in which case $\bar{w}, \rho$ can be computed as (weighted) Lipschitz constants. Condition (23d) follows naturally if V_δ is defined as a weighted norm or with a contraction metric.

Given this offline design, we formulate the following robust *MPC for tracking* formulation from [21]:

$$\min_{\substack{\mathbf{x},\mathbf{u},\\ x_a,u_a,\alpha}} \sum_{k=0}^{N-1} \ell(x_k,u_k,x_a,u_a) + V_f(x_N,x_a,u_a) + V_o(y_a,y_r) \quad (24a)$$

$$\begin{aligned} \text{s.t. } & x_0 = x(t), && (24b)\\ & x_{k+1} = f(x_k,u_k),\ k \in \mathbb{N}_{[0,N-1]}, && (24c)\\ & g_j(x_k,u_k) + c_j\bar{w}\frac{1-\rho^k}{1-\rho} \leq 0, && (24d)\\ & j \in \mathbb{N}_{[1,n_c]},\ \ k \in \mathbb{N}_{[0,N-1]}, && \\ & y_a = h(x_a,u_a), && (24e)\\ & x_a = f(x_a,u_a), && (24f)\\ & g_j(x_a,u_a) + c_j\alpha \leq 0,\ j \in \mathbb{N}_{[1,n_c]}, && (24g)\\ & \alpha \geq \frac{\bar{w}}{1-\rho}, && (24h)\\ & V_\delta(x_N,x_a) + \frac{1-\rho^N}{1-\rho}\bar{w} \leq \alpha. && (24i)\end{aligned}$$

This formulation initializes the predictions at the measured state $x(t)$ and the applied control input is $u(t) = u_0^\star$. Contrary to the nominal formulation (19), the constraints (24d) are tightened proportional to the disturbance bound $\bar{w}$. The terminal set constraint is characterized by (24g)–(24i), which incorporates an online optimization of the scaling α following [12]. In particular, Condition (24i) defines the terminal set as a sublevel set of the Lyapunov function around the artificial steady-state x_a with an online optimized scaling $\alpha > 0$, while Condition (24g) ensures that this set lies in the (tightened) constraints. Furthermore, Condition (24h) provides a lower bound on the size of the terminal set to ensure robust positive invariance and recursive feasibility. Notably, these formulas are particularly simple as the feedback κ and Lyapunov function V_δ from the robust formulation are also used to define the terminal ingredients.

Theorem 7 ([21, Thm. 1])**.** *Let Assumption 7 hold and suppose that Problem (24) is feasible at $t = 0$. Then, Problem (24) is feasible and the closed-loop system satisfies the constraints (16c) for all $t \in \mathbb{N}$, independent of y_r and for any disturbance realization $w(t) \in \mathcal{W}$.*

Given the additional conditions regarding the terminal cost V_f, convexity, and uniqueness (Asm. 5–6), and uniform continuity of all the involved functions, this approach also ensures input-to-state stability of the (robustly) optimal steady-state, see [21, Thm. 1] for details.

D. Other extensions and experimental results

Other notable extensions of the MPC for tracking framework include non-linear MPC formulations that leverage the

[1] Following [77, Prop. 2,5], condition (23b) does in fact not hold for all $x \in \mathbb{R}^{n_x}$ but for all states visited during closed-loop operation due to the tightened constraints.

flexibility of artificial references but aim to directly minimize some economic cost ℓ_{e}. Corresponding results for non-linear systems have been developed in [15], [16], [18], see also [49, Chap. 5] for an in depth overview. Experimental hardware results for non-linear MPC for tracking formulations include setpoint tracking with a four-tank [7], robotic manipulation with collision avoidance [21], and distributed coverage and collision avoidance with a fleet of miniature race cars [35].

V. Application to Learning-based MPC

In this section, we highlight how *MPC for tracking* formulations can be a key enabler to address problems in learning-based MPC. Learning-based techniques are typically leveraged if the system model or the external environment (objective, constraints) are uncertain, see [79] for an overview. We first provide a general discussion, highlighting where *MPC for tracking* formulations are a crucial component in learning-based approaches. Then, we explore two applications in more detail:

- safely exploring uncertain constraints [43];
- distributed coverage with unknown objectives [35].

Learning system dynamics: In many applications, the system model (16) is subject to significant uncertainty and hence online generated data can be used to improve the model knowledge and thus boost performance. However, the fact that our model changes during runtime also poses challenges in determining an optimal steady-state that should be regulated. This issue can be naturally addressed with *MPC for tracking* formulations as the desired steady-state is jointly optimized in the MPC. Corresponding results for different data-driven and learning-based models can be found in [22], [42], [77], [80]–[82].

Exploring unknown environment: Especially in robotics applications, we often do not have an accurate ‘map’ regarding the location of possible obstacles, which are needed to ensure safe exploration with the MPC. This problem is typically solved by having a cautious approximation of the obstacle free space which expands by exploring the unknown environment during online operation. For such problems, we naturally do not know beforehand where exactly we want to go and in general the targets we may wish to reach can be very far away. These problems can be naturally addressed by using *MPC for tracking* formulations, see [43], [73], [83].

Guaranteed safe exploration: In [43], an MPC-based method to systematically guarantee safe exploration of unknown environments is developed. In addition to the user-specified system constraints $(x,u)\in\mathcal{Z}$, safe operation requires $q(y)\leq 0$ where $q:\mathbb{R}^{n_y}\rightarrow\mathbb{R}$ is an unknown/uncertain function. By collecting n (noisy) measurements of this function (e.g., through LIDAR) and given suitable regularity conditions on q, one can learn a function $\underline{q}^n(y)\leq q(y)$, which can be used to pessimistically ensure safe operation. In particular, this is achieved by leveraging Gaussian processes [84], a versatile machine learning method with guaranteed error bounds. During runtime, an MPC determines a location that can be safely reached and where collecting measurements will maximize the information about the uncertain environment. This method ensures that (with high probability) the system is safely operated for all times and the unknown domain is explored up to a user chosen tolerance in finite time. While the theoretical guarantees largely rely on methods from machine learning and optimal control, the practicality hinges on the utilization of concepts from *MPC for tracking*. In particular, persistent safety is ensured by planning trajectories that end in some positively invariant safe set. While in principle such a set could be designed offline, this would require a computationally prohibitive prediction horizon to ensure exploration of a large domain. These issues are circumvented by relying on techniques from *MPC for tracking* to implicitly define the safe set with an artificial reference. In particular, the online learned (pessimistic) constraint $\underline{q}^n(y)$ is used to define a (time-varying) terminal set constraint given by

$$\mathcal{X}_f^n=\{x:\ \exists u:(x,u)\in\mathcal{Z}_s,\ y=h(x,u),\ \underline{q}^n(y)\leq 0\}.$$

By leveraging techniques similar to Section IV, [43, Lemma 3] shows that for path-connected (non-convex) constraints $\mathcal{Y}_s$ and locally controllable dynamics, any point in this set can be reached with finite time, which is crucial to ensure exploration of the complete domain in finite time.

Distributed coordination and learning: In [35], a learning-based tracking MPC framework is presented to address the distributed coverage problem. The coverage problem requires distributed agents to optimally cover an area, which can be posed as minimizing the coverage cost with partitions $\mathcal{Y}_i\subseteq\mathbb{R}^{n_y}$ and density $\phi:\mathbb{R}^{n_y}\rightarrow\mathbb{R}$:

$$\min_{y_i,\mathcal{Y}_i}\sum_i\int_{p\in\mathcal{Y}_i}\|p-y_i\|^2\phi(p)dp. \tag{25}$$

This formulation can, e.g., be used to model optimal repositioning of a fleet of taxis or aerial robots observing a wild fire. The key challenge lies in the fact that methods need to be scalable to a large number of agents, ensure collision avoidance, and deal with uncertain density functions ϕ. The framework in [35] addresses all these problems simultaneously by leveraging non-linear *MPC for tracking* formulations on multiple levels. By adapting the *MPC for tracking* formulation without terminal ingredients from [29], a controller is developed to steer non-holonomic vehicles to arbitrary position references. Given these strong properties of the local controllers, distributed coordination is enabled independent of the dynamics of the system by simply coordinating positions and Voronoi partitions based on the Lloyd algorithm. Furthermore, a framework is developed to enable efficient exploration and learning. This formulation achieves active exploration by using an offset cost V_o that also depends on the variance of the density function $\phi(y)$ at the artificial steady-state.

VI. Optimization and aspects and practical implementation of MPC for tracking

A. *Implementing linear MPC for tracking*

The linear *MPC for tracking* formulations presented in Section II can be posed as QP problems if the offset cost V_o is taken as a quadratic or linear function and the invariant set for tracking $\mathcal{X}_a$ is computed as a polyhedron. From a computational standpoint this is rather advantageous, since QP problems are a particularly simple class of optimization problem for which there are many state-of-the-art efficient solvers, such as [85]–[87], to name a few. Furthermore, the resulting QP problem is sparse; a feature that is exploited by most modern solvers [85], [88]. In fact, the resulting sparsity pattern is rather simple, and can thus be exploited by tailored solvers [46], [48], [50], see Section VI-B.

The robust and periodic formulations presented in Sections III-A and III-B, respectively, are also QP problems under the previous two conditions. The HMPC formulation presented in Section III-C is not a QP problem due to the inclusion of second-order cone constraints (13i)-(13j), leading instead to a second-order cone program (SOCP), which can also be rewritten as a quadratically constrained quadratic program (QCQP); both of which can be solved using several state-of-the-art solvers, e.g., [88], [89]. In particular, [47] presents a solver tailored to the HMPC formulation (available in [50]) and shows that it can be solved in computation times comparable to state-of-the-art QP solvers.

A tool that is rather useful for prototyping and testing linear *MPC for tracking* formulations is the MATLAB toolbox YALMIP [90], which provides a simple syntax to build the optimization problem and to then solve it by selecting from a large pool of state-of-the-art solvers, including most of the ones referenced above.

Computing the invariant set for tracking: As mentioned in Section II, and as discussed in detail in [5, §2.2], $\mathcal{X}_a$ can be obtained as the maximal invariant set of a system obtained by extending (1) with the artificial reference. This results in a polyhedral set than can be computed using standard procedures [1], [91]. The MATLAB MPT3 toolbox [92] can be rather useful to this end.

Computation of the maximal invariant set can be very expensive. Several alternatives are typically considered: use a terminal equality constraint, as in (7); dynamically scale the terminal set [93]; avoid the use of a terminal set by choosing a sufficiently large prediction horizon [7]; or compute an elliptic terminal invariant set [94, §4.1] by solving an optimization problem with linear matrix inequality constraints [95]. The downside of using an ellipsoid is that the resulting optimization problem is no longer a QP, but instead a SOCP/QCQP. However, as previously discussed, there are several state-of-the-art solvers well suited for these problems [88], [89], [96].

Soft constraints: In spite of the larger domain of attraction typically obtained with the use of artificial references, the presence of constraints can still lead to feasibility issues when dealing with real systems or in the presence of unknown uncertainties. To avoid feasibility issues, a common practical solution is to make use of soft constraints [44], [45].

B. *Exploiting the semi-banded structure of linear MPC for tracking*

The stage cost $\ell(x_k, u_k, x_a, u_a) \doteq \|x_k - x_a\|_Q^2 + \|u_k - u_a\|_R^2$, commonly used in many MPC for tracking formulations, includes cross products between (x_k, u_k) and (x_a, u_a) that break down the strictly banded structure of the optimization problem related to standard MPC [46], [97]. Consider, for example, the cost of the linear *MPC for tracking* formulation with equality constraints (7):

$$V(z) = \sum_{k=0}^{N-1} \ell(x_k, u_k, x_a, u_a) + \|x_a - x_r\|_T^2 + \|u_a - u_r\|_S^2,$$

where z denotes the vector of decision variables, i.e.,

$$z = \left[x_0^\top,\, u_0^\top,\, x_1^\top,\, u_1^\top,\, \dots,\, x_{N-1}^\top,\, u_{N-1}^\top,\, x_a^\top,\, u_a^\top\right]^\top.$$

In this case, the cost $V(z)$ is a quadratic function on z that can be rewritten as $V(z) = z^\top (H_B + H_{LR}) z + q^\top z + c$, where

$$H_B = \begin{bmatrix} Q & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & R & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & Q & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & R & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & NQ & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & NR \end{bmatrix},$$

$$H_{LR} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & -Q & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -R \\ 0 & 0 & 0 & 0 & \cdots & -Q & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -R \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -Q & 0 & -Q & 0 & \cdots & T & 0 \\ 0 & -R & 0 & -R & \cdots & 0 & S \end{bmatrix}.$$

We notice that H_B is a banded matrix and H_{LR} is a low-rank matrix. Thus, the matrix defining the quadratic terms of $V(z)$ has a semi-banded structure that prevents the direct use of the efficient optimization techniques developed for the fully banded case of standard MPC [97], [98]. Two different approaches have been presented in the literature to address this issue. The first approach involves using an extended ADMM formulation that recovers the banded structure by considering three primal decision variables in the ADMM formulation [46]. The second approach relies on the use of the Woodbury matrix identity [99], which allows for solving, in an efficient way, the semi-banded systems of equations required to implement the standard ADMM algorithm (instead of the extended one) [45], [48]. The obtained computational times are similar to the ones corresponding to standard MPC.

C. Solving non-linear MPC for tracking

Assuming f, h are smooth, non-linear *MPC for tracking* problems can be efficiently coded with CasADi [100] in MATLAB or Python. This provided automatic differentiation and interfaces general purpose sparse non-linear optimization solvers like IPOPT [101]. For embedded application and long prediction horizons, Acados [102] can offer efficient implementation by also leveraging QP solvers that exploit the stage wise structure of MPC [87]. Problem (19) can be written in this form by augmenting the state space with the artificial reference, as, e.g., done in experiments in [35].

VII. Conclusion

We have provided a tutorial exposition on *MPC for tracking* formulations, which incorporate an artificial reference as an additional decision variable of the MPC's optimization problem. The addition of the artificial reference in linear and non-linear MPC provides several important benefits: an increase of the domain of attraction, the ability to deal with non-reachable references, recursive feasibility under online operational changes, and asymptotic stability to the "best" reachable objective. This paradigm is very versatile, as exemplified by the many extensions developed in the literature, including robust designs, economic costs, periodic references, or zone control. Furthermore, tailored solvers have been developed that provide solve-times in the order of milliseconds for some of the linear formulations. Finally, *MPC for tracking* can be a key enabler to address problems beyond tracking in learning-based MPC.

References

[1] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Madison, WI, USA: Nob Hill Publishing, 2017.

[2] E. F. Camacho and C. B. Alba, *Model Predictive Control*, 2nd ed. London, UK: Springer-Verlag, 2007.

[3] E. Garone, S. Di Cairano, and I. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, 2017.

[4] C. Prada, S. Cristea, and A. Kuznetsov, "Stability guarantee in predictive control delta domain," in *14th IFAC World Congress 1999, Beijing, China*. IFAC, 1999, pp. 4624–4629.

[5] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, "MPC for tracking piecewise constant references for constrained linear systems," *Automatica*, vol. 44, no. 9, pp. 2382–2387, 2008.

[6] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E. Camacho, "MPC for tracking with optimal closed-loop performance," *Automatica*, vol. 45, no. 8, pp. 1975–1978, 2009.

[7] D. Limon, A. Ferramosca, I. Alvarado, and T. Alamo, "Nonlinear MPC for tracking piece-wise constant reference signals," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3735–3750, 2018.

[8] M. Pereira, D. Limon, D. M. de la Peña, L. Valverde, and T. Alamo, "Periodic economic control of a nonisolated microgrid," *IEEE Transactions on industrial electronics*, vol. 62, no. 8, pp. 5247–5255, 2015.

[9] D. Limon, M. Pereira, D. M. de la Peña, T. Alamo, and J. Grosso, "Single-layer economic model predictive control for periodic operation," *Journal of Process Control*, vol. 24, no. 8, pp. 1207–1224, 2014.

[10] D. Limon, M. Pereira, D. M. de la Peña, T. Alamo, C. N. Jones, and M. N. Zeilinger, "MPC for tracking periodic references," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 1123–1128, 2016.

[11] J. Köhler, M. A. Müller, and F. Allgöwer, "MPC for nonlinear periodic tracking using reference generic offline computations," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 556–561, 2018.

[12] ——, "A nonlinear tracking model predictive control scheme for dynamic target signals," *Automatica*, vol. 118, p. 109030, 2020.

[13] H. Yang, H. Zhao, Y. Xia, and J. Zhang, "Nonlinear MPC with time-varying terminal cost for tracking unreachable periodic references," *Automatica*, vol. 123, p. 109337, 2021.

[14] M. Köhler, M. A. Müller, and F. Allgöwer, "Distributed model predictive control for periodic cooperation of multi-agent systems," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 3158–3163, 2023.

[15] L. Fagiano and A. R. Teel, "Generalized terminal state constraint for model predictive control," *Automatica*, vol. 49, no. 9, pp. 2622–2631, 2013.

[16] M. A. Müller, D. Angeli, and F. Allgöwer, "Economic model predictive control with self-tuning terminal cost," *European Journal of Control*, vol. 19, no. 5, pp. 408–416, 2013.

[17] A. Ferramosca, D. Limon, and E. F. Camacho, "Economic MPC for a changing economic criterion for linear systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2657–2667, 2014.

[18] J. Köhler, M. A. Müller, and F. Allgöwer, "Periodic optimal control of nonlinear constrained systems using economic model predictive control," *Journal of Process Control*, vol. 92, pp. 185–201, 2020.

[19] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, "Robust tube-based MPC for tracking of constrained linear systems with additive disturbances," *Journal of Process Control*, vol. 20, no. 3, pp. 248–260, 2010.

[20] M. N. Zeilinger, D. M. Raimondo, A. Domahidi, M. Morari, and C. N. Jones, "On real-time robust model predictive control," *Automatica*, vol. 50, no. 3, pp. 683–694, 2014.

[21] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, "Safe and fast tracking on a robot manipulator: Robust MPC and neural network control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3050–3057, 2020.

[22] T. Peschke and D. Görges, "Robust adaptive tube tracking model predictive control for piece-wise constant reference signals," *International Journal of Robust and Nonlinear Control*, vol. 33, no. 14, pp. 8158–8182, 2023.

[23] M. M. Morato, V. M. Cunha, T. L. Santos, J. E. Normey-Rico, and O. Sename, "A robust nonlinear tracking MPC using qLPV embedding and zonotopic uncertainty propagation," *Journal of the Franklin Institute*, p. 106713, 2024.

[24] J. A. Paulson, T. L. Santos, and A. Mesbah, "Mixed stochastic-deterministic tube MPC for offset-free tracking in the presence of plant-model mismatch," *Journal of process control*, vol. 83, pp. 102–120, 2019.

[25] A. D'Jorge, B. F. Santoro, A. Anderson, A. H. González, and A. Ferramosca, "Stochastic model predictive control for tracking linear systems," *Optimal Control Applications and Methods*, vol. 41, no. 1, pp. 65–83, 2020.

[26] A. Ferramosca, D. Limon, A. H. González, D. Odloak, and E. F. Camacho, "MPC for tracking zone regions," *Journal of Process Control*, vol. 20, no. 4, pp. 506–516, 2010.

[27] A. Ferramosca, D. Limon, A. H. González, I. Alvarado, and E. F. Camacho, "Robust MPC for tracking zone regions based on nominal predictions," *Journal of Process Control*, vol. 22, no. 10, pp. 1966–1974, 2012.

[28] P. Abuin, A. Ferramosca, C. Toffanin, L. Magni, and A. H. González, "Pulsatile zone MPC with asymmetric stationary cost for artificial pancreas based on a non-standard IOB constraint," *Journal of Process Control*, vol. 136, p. 103191, 2024.

[29] R. Soloperto, J. Köhler, and F. Allgöwer, "A nonlinear MPC scheme for output tracking without terminal ingredients," *IEEE Transactions on Automatic Control*, vol. 68, no. 4, pp. 2368–2375, 2022.

[30] I. Sánchez, A. D'Jorge, G. V. Raffo, A. H. González, and A. Ferramosca, "Nonlinear model predictive path following controller with obstacle avoidance," *Journal of Intelligent & Robotic Systems*, vol. 102, pp. 1–18, 2021.

[31] M. A. dos Santos, A. Ferramosca, and G. V. Raffo, "Set-point tracking MPC with avoidance features," *Automatica*, vol. 159, p. 111390, 2024.

[32] A. Ferramosca, D. Limon, I. Alvarado, and E. F. Camacho, "Cooperative distributed MPC for tracking," *Automatica*, vol. 49, no. 4, pp. 906–914, 2013.

[33] P. Chanfreut, J. M. Maestre, A. Ferramosca, F. J. Muros, and E. F. Camacho, "Distributed model predictive control for tracking: A coalitional clustering approach," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6873–6880, 2021.

[34] A. Aboudonia, A. Eichler, F. Cordiano, G. Banjac, and J. Lygeros, "Distributed model predictive control with reconfigurable terminal ingredients for reference tracking," *IEEE transactions on automatic control*, vol. 67, no. 11, pp. 6263–6270, 2021.

[35] R. Rickenbach, J. Köhler, A. Scampicchio, M. N. Zeilinger, and A. Carron, "Active learning-based model predictive coverage control," *IEEE Transactions on Automatic Control*, 2024.

[36] M. Köhler, L. Krügel, L. Grüne, M. A. Müller, and F. Allgöwer, "Transient performance of MPC for tracking," *IEEE Control Systems Letters*, vol. 7, pp. 2545–2550, 2023.

[37] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven tracking MPC for changing setpoints," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6923–6930, 2020.

[38] G. Galuppini, L. Magni, and A. Ferramosca, "Nonlinear MPC for tracking piecewise-constant reference signals: the positive semidefinite stage cost case," *IFAC-PapersOnLine*, vol. 56, no. 1, pp. 210–215, 2023.

[39] P. Krupa, M. Pereira, D. Limon, and T. Alamo, "Single harmonic based model predictive control for tracking," in *58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 151–156.

[40] P. Krupa, D. Limon, and T. Alamo, "Harmonic based model predictive control for set-point tracking," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 48–62, 2022.

[41] P. Krupa, D. Limon, A. Bemporad, and T. Alamo, "Harmonic model predictive control for tracking periodic references," *arXiv preprint arXiv:2310.16723*, 2023.

[42] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Linear tracking MPC for nonlinear systems—Part ii: The data-driven case," *IEEE Transactions on Automatic Control*, vol. 67, no. 9, pp. 4406–4421, 2022.

[43] M. Prajapat, J. Köhler, M. Turchetta, A. Krause, and M. N. Zeilinger, "Safe guaranteed exploration for non-linear systems," *arXiv preprint arXiv:2402.06562*, 2024.

[44] M. N. Zeilinger, M. Morari, and C. N. Jones, "Soft constrained model predictive control with robust stability guarantees," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1190–1202, 2014.

[45] V. Gracia, P. Krupa, D. Limon, and T. Alamo, "Implementation of soft-constrained MPC for tracking using its semi-banded problem structure," *IEEE Control Systems Letters*, 2024.

[46] P. Krupa, I. Alvarado, D. Limon, and T. Alamo, "Implementation of model predictive control for tracking in embedded systems using a sparse extended ADMM algorithm," *Transactions on Control Systems Technology*, vol. 30, no. 4, pp. 1798–1805, 2022.

[47] P. Krupa, D. Limon, A. Bemporad, and T. Alamo, "Efficiently solving the harmonic model predictive control formulation," *IEEE Transactions on Automatic Control*, vol. 68, no. 9, pp. 5568–5575, 2023.

[48] V. Gracia, P. Krupa, D. Limon, and T. Alamo, "Efficient implementation of MPC for tracking using ADMM by decoupling its semi-banded structure," in *Accepted in the IEEE European Control Conference*, 2024, preprint available at arXiv:2402.09912.

[49] J. Köhler, M. A. Müller, and F. Allgöwer, "Analysis and design of model predictive control frameworks for dynamic operation — An overview," *Annual Reviews in Control*, vol. 57, p. 100929, 2024.

[50] P. Krupa, V. Gracia, D. Limon, and T. Alamo, "SPCIES: Suite of Predictive Controllers for Industrial Embedded Systems," https://github.com/GepocUS/Spcies, Dec 2020.

[51] K. Dong, J. Luo, and D. Limon, "A novel stable and safe model predictive control framework for autonomous rendezvous and docking with a tumbling target," *Acta Astronautica*, vol. 200, pp. 176–187, 2022.

[52] J. A. Rebollo, R. Vazquez, I. Alvarado, and D. Limon, "MPC for tracking applied to rendezvous with non-cooperative tumbling targets ensuring stability and feasibility," *arXiv preprint arXiv:2403.10986*, 2024.

[53] P. Krupa, J. Camara, I. Alvarado, D. Limon, and T. Alamo, "Real-time implementation of MPC for tracking in embedded systems: Application to a two-wheeled inverted pendulum," in *European Control Conference (ECC)*, 2021, pp. 669–674.

[54] J. A. Borja-Conde, J. Nadales, J. Ordonez, F. Fele, and D. Limon, "Efficient management of HVAC systems through coordinated operation of parallel chiller units: An economic predictive control approach," *Energy and Buildings*, vol. 304, p. 113879, 2024.

[55] C. V. Rao and J. B. Rawlings, "Steady states and constraints in model predictive control," *AIChE Journal*, vol. 45, no. 6, pp. 1266–1278, 1999.

[56] B. Hu and A. Linnemann, "Toward infinite-horizon optimality in nonlinear model predictive control," *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 679–682, 2002.

[57] I. Alvarado, "Model predictive control for tracking constrained linear systems," Ph.D. dissertation, Department of Systems Engineering and Automation, Universidad de Sevilla, Seville, Spain, 2007.

[58] D. Q. Mayne, M. M. Seron, and S. V. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[59] D. Limon, I. Alvarado, T. Alamo, and E. Camacho, "On the design of robust tube-based MPC for tracking," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 15 333–15 338, 2008.

[60] M. Pereira, D. M. de la Peña, D. Limon, I. Alvarado, and T. Alamo, "Application to a drinking water network of robust periodic MPC," *Control Engineering Practice*, vol. 57, pp. 50–60, 2016.

[61] C. Cosner, G. Anwar, and M. Tomizuka, "Plug in repetitive control for industrial robotic manipulators," in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE, 1990, pp. 1970–1975.

[62] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model predictive contouring control for time-optimal quadrotor flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3340–3356, 2022.

[63] M. Leomanni, G. Bianchini, A. Garulli, and R. Quartullo, "Sum-of-norms MPC for linear periodic systems with application to spacecraft rendezvous," in *Proc. 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 4665–4670.

[64] M. Gupta and J. H. Lee, "Period-robust repetitive model predictive control," *Journal of Process Control*, vol. 16, no. 6, pp. 545–555, 2006.

[65] R. Cao and K.-S. Low, "A repetitive model predictive control approach for precision tracking of a linear motion system," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1955–1962, 2008.

[66] R. Gondhalekar and C. N. Jones, "MPC of constrained discrete-time linear periodic systems — A framework for asynchronous control: Strong feasibility, stability and optimality via periodic invariance," *Automatica*, vol. 47, no. 2, pp. 326–333, 2011.

[67] M. Ellis, H. Durand, and P. D. Christofides, "A tutorial review of economic model predictive control methods," *Journal of Process Control*, vol. 24, no. 8, pp. 1156–1178, 2014.

[68] M. Diehl, R. Amrit, and J. B. Rawlings, "A Lyapunov function for economic optimizing model predictive control," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 703–707, 2010.

[69] T. J. Broomhead, C. Manzie, R. C. Shekhar, and P. Hield, "Robust periodic economic MPC for linear systems," *Automatica*, vol. 60, pp. 30–37, 2015.

[70] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer, "Constraint-tightening and stability in stochastic model predictive control," *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3165–3177, 2016.

[71] J.-M. Coron, L. Grüne, and K. Worthmann, "Model predictive control, cost controllability, and homogeneity," *SIAM Journal on Control and Optimization*, vol. 58, no. 5, pp. 2979–2996, 2020.

[72] A. Cotorruelo, D. R. Ramirez, D. Limon, and E. Garone, "Nonlinear MPC for tracking for a class of nonconvex admissible output sets," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3726–3732, 2020.

[73] R. Soloperto, A. Mesbah, and F. Allgöwer, "Safe exploration and escape local minima with model predictive control under partially unknown constraints," *IEEE Transactions on Automatic Control*, 2023.

[74] J. Köhler, M. A. Müller, and F. Allgöwer, "A nonlinear model predictive control framework using reference generic terminal ingredients," *IEEE Transactions on Automatic Control*, vol. 65, no. 8, pp. 3576–3583, 2020.

[75] D. Benders, J. Köhler, T. Niesten, R. Babuška, J. Alonso-Mora, and L. Ferranti, "Embedded hierarchical MPC for autonomous navigation," *arXiv preprint arXiv:2406.11506*, 2024.

[76] J. Köhler, R. Soloperto, M. A. Müller, and F. Allgöwer, "A computationally efficient robust model predictive control framework for uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 794–801, 2020.

[77] A. Sasfi, M. N. Zeilinger, and J. Köhler, "Robust adaptive MPC using control contraction metrics," *Automatica*, vol. 155, p. 111169, 2023.

[78] P. Zhao, A. Lakshmanan, K. Ackerman, A. Gahlawat, M. Pavone, and N. Hovakimyan, "Tube-certified trajectory tracking for nonlinear systems with robust control contraction metrics," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5528–5535, 2022.

[79] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

[80] T. Strelnikova, J. Köhler, and J. Berberich, "Adaptive tracking MPC for nonlinear systems via online linear system identification," in *Proc. Symposium on Systems Theory in Data and Optimization (SysDO)*, 2024.

[81] J. Manzano, J. Calliess, D. M. de la Peña, and D. Limon, "Online learning robust MPC: an exploration-exploitation approach," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5292–5297, 2020.

[82] Y. Wang, Y. Yang, Y. Pu, and C. Manzie, "Data-driven predictive tracking control based on Koopman operators," *arXiv preprint arXiv:2208.12000*, 2022.

[83] D. Saccani, L. Cecchin, and L. Fagiano, "Multitrajectory model predictive control for safe uav navigation in an unknown environment," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 5, pp. 1982–1997, 2023.

[84] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[85] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[86] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014.

[87] G. Frison and M. Diehl, "HPIPM: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

[88] B. O'Donoghue, "Operator splitting for a homogeneous embedding of the linear complementarity problem," *SIAM Journal on Optimization*, vol. 31, no. 3, pp. 1999–2023, 2021.

[89] G. Frison, J. Frey, F. Messerer, A. Zanelli, and M. Diehl, "Introducing the quadratically-constrained quadratic programming framework in HPIPM," in *2022 European Control Conference (ECC)*. IEEE, 2022, pp. 447–453.

[90] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, 2004.

[91] M. Fiacchini, T. Alamo, and E. F. Camacho, "On the computation of convex robust control invariant sets for nonlinear systems," *Automatica*, vol. 46, no. 8, pp. 1334–1338, 2010.

[92] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *Proc. European control conference (ECC)*. IEEE, 2013, pp. 502–510.

[93] D. Simon, J. Löfberg, and T. Glad, "Reference tracking MPC using dynamic terminal set transformation," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2790–2795, 2014.

[94] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[95] Z. Wan and M. V. Kothare, "An efficient off-line formulation of robust model predictive control using linear matrix inequalities," *Automatica*, vol. 39, no. 5, pp. 837–846, 2003.

[96] P. Krupa, R. Jaouani, D. Limon, and T. Alamo, "A sparse ADMM-based solver for linear MPC subject to terminal quadratic constraint," *IEEE Transactions on Control Systems Technology*, 2024.

[97] P. Krupa, D. Limon, and T. Alamo, "Implementation of model predictive control in programmable logic controllers," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1117–1130, 2021.

[98] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on control systems technology*, vol. 18, no. 2, pp. 267–278, 2009.

[99] D. J. Tylavsky and G. R. Sohie, "Generalization of the matrix inversion lemma," *Proceedings of the IEEE*, vol. 74, no. 7, pp. 1050–1052, 1986.

[100] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[101] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

[102] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, 2022.