Efficient implementation of MPC for tracking using ADMM by decoupling its semi-banded structure

Victor Gracia†, Pablo Krupa⋆, Daniel Limon†, Teodoro Alamo†

Abstract— **Model Predictive Control (MPC) for tracking formulation presents numerous advantages compared to standard MPC, such as a larger domain of attraction and recursive feasibility even when abrupt changes in the reference are produced. As a drawback, it includes some extra decision variables in its related optimization problem, leading to a semi-banded structure that differs from the banded structure encountered in standard MPC. This semi-banded structure prevents the direct use of the efficient algorithms available for banded problems. To address this issue, we present an algorithm based on the alternating direction method of multipliers that explicitly takes advantage of the underlying semi-banded structure of the MPC for tracking.**

Index Terms— **Model predictive control, embedded optimization, embedded systems, ADMM, MPC for tracking.**

I. INTRODUCTION

Model Predictive Control (MPC) is an advanced control policy whose control action is obtained from a constrained Optimization Problem (OP) posed at every sample time [1], [2]. MPC has become widely popular due to its ability to optimize the plant operation performance while dealing with constraints. However, it presents disadvantages, such as the computational cost required to solve its associated OP at each sample time, or the unavailability of a suitable control action when the OP is infeasible or if it cannot be solved in a short amount of time compared to the sample time of the system.

Recently, there has been a significant amount of academic literature providing results which mitigate these issues, such as the proposal of efficient solvers suitable for the implementation of MPC [3], [4], [5], [6], [7], or results which palliate the problem of MPC infeasibility [8], [9].

In particular, in this article we focus on the *MPC for tracking* (MPCT) formulation, originally proposed in [8], which introduces an artificial reference as an additional decision variable of the OP. The main benefit of introducing this artificial reference is that MPCT attains a notably larger domain of attraction and feasibility region when compared with standard MPC formulations. Additionally, MPCT guarantees recursive feasibility, even when sudden changes in the reference occur. Furthermore, it also guarantees asymptotic stability to an admissible steady state of the system, even

if the reference is infeasible. These benefits make MPCT a strong candidate for its use in a practical setting, since it significantly mitigates some of the main issues of standard MPC. However, these benefits come at the cost of a more complex OP due to the addition of the artificial reference. In particular, the inclusion of the additional decision variables leads to a semi-banded structure in the MPCT OP, whereas standard MPC formulations present a banded structure than can be exploited by the optimization solver [3], [10].

In [11], the authors propose an efficient solver for MPCT where the banded structure of standard MPC formulations is recovered by using the Extended Alternating Direction Method of Multipliers (EADMM) [12]. This provides a solver whose computational cost per iteration is nearly identical to the one for standard MPC formulations using first-order optimization methods such as (non-extended) ADMM [13]. However, the disadvantage is that EADMM presents several drawbacks when compared with ADMM, both theoretical and in terms of its practical performance.

In this article we present an alternative way of solving MPCT using the ADMM algorithm by decomposing the semi-banded structure of the MPCT OP. This decomposition recovers the same banded structure associated with the standard MPC formulation, which can thus be exploited in the numerical solver. The computational cost per iteration of the resulting solver is over two times larger than the one for the EADMM solver proposed in [11]. However, in spite of this, the use of the ADMM algorithm instead of EADMM provides better theoretical guarantees and practical performance, as illustrated by the numerical results.

This article is structured as follows. Section II introduces the MPCT formulation. Section III presents the proposed ADMM algorithm for MPCT, where we show how we decompose its semi-banded structure to attain an efficient solver. Section IV shows numerical results demonstrating the practical benefits of the proposed solver. Finally, Section V summarizes the main results of the article.

Notation: Given a square matrix A, $\det(A)$ is its determinant. Given two integers a, b, $\mathbb{I}_a^b = \{a, a+1, \ldots, b-1, b\}$. $\mathcal{S}_\succ^n$ denotes the set of symmetric positive definite matrices of size $n \times n$. Given a vector x, we denote its j-th component as $x_{(j)}$. Given $Q \in \mathcal{S}_\succ^n$, $\|x\|_Q \doteq \sqrt{x^\top Q x}$ and $\|x\|_\infty \doteq \max_{j=1\ldots n} |x_{(j)}|$. The identity matrix of dimension n is denoted by I_n and the vector of ones by $\mathbb{1}_n \in \mathbb{R}^n$ (we may drop the sub-index n if the dimension is clear from the context). Given $x, y \in \mathbb{R}^n$, $x \leq (\geq) y$ denotes component-wise inequalities. For vectors x_1 to x_N of any dimension, $(x_1, \ldots, x_N)$ denotes the column vector formed by their

concatenation. We denote by $\mathrm{diag}(A_1, \ldots, A_N)$ the block diagonal matrix formed by the concatenation of scalars and/or matrices A_1 to A_N (possibly of different dimensions). Function $\max(\cdot) \colon \mathbb{R} \times \mathbb{R} \times \cdots \times \mathbb{R} \to \mathbb{R}$ returns the maximum element of its scalar inputs. The Kronecker product between matrices A and B is denoted by $A \otimes B$.

II. MPC FOR TRACKING FORMULATION

Consider a controllable discrete-time system described by

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

where $x(t) \in \mathbb{R}^{n_x}$ and $u(t) \in \mathbb{R}^{n_u}$ are the state and input at sample time t, respectively, subject to box constraints

$$\underline{x} \le x(t) \le \overline{x}, \tag{2a}$$
$$\underline{u} \le u(t) \le \overline{u}, \tag{2b}$$

where $\underline{x}, \overline{x} \in \mathbb{R}^{n_x}$ and $\underline{u}, \overline{u} \in \mathbb{R}^{n_u}$ satisfy $\underline{x} < \overline{x}$ and $\underline{u} < \overline{u}$. The control objective is to steer the system to the steady-state reference (x_r, u_r) while satisfying the system constraints. If (x_r, u_r) is admissible, then the closed-loop system should converge to it. Otherwise, we wish to converge to some admissible steady-state that is close to (x_r, u_r).

This control objective can be achieved by using the *MPC for tracking* (MPCT) proposed in [8]. In particular, we address the MPCT formulation with terminal equality constraint from [11], whose OP is given by

$$\min_{\substack{\mathbf{x}, \mathbf{u}, \\ x_s, u_s}} \sum_{i=0}^{N-1} (\|x_i - x_s\|_Q^2 + \|u_i - u_s\|_R^2) + \|x_s - x_r\|_T^2 + \|u_s - u_r\|_S^2 \tag{3a}$$

$$\text{s.t. } x_0 = x(t), \tag{3b}$$
$$x_{i+1} = Ax_i + Bu_i, \ i \in \mathbb{I}_0^{N-2}, \tag{3c}$$
$$x_s = Ax_{N-1} + Bu_{N-1}, \tag{3d}$$
$$x_s = Ax_s + Bu_s, \tag{3e}$$
$$\underline{x} \le x_i \le \overline{x}, \ i \in \mathbb{I}_1^{N-1}, \tag{3f}$$
$$\underline{u} \le u_i \le \overline{u}, \ i \in \mathbb{I}_0^{N-1}, \tag{3g}$$
$$\underline{x}_\varepsilon \le x_s \le \overline{x}_\varepsilon, \tag{3h}$$
$$\underline{u}_\varepsilon \le u_s \le \overline{u}_\varepsilon, \tag{3i}$$

where the decision variables are the artificial reference (x_s, u_s) and the predicted states $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1})$ and inputs $\mathbf{u} = (u_0, u_1, \ldots, u_{N-1})$ along the prediction horizon N; $x(t)$ is the current state of the system at sample time t; the matrices $Q \in \mathcal{S}_{\succ}^{n_x}$, $R \in \mathcal{S}_{\succ}^{n_u}$, $T \in \mathcal{S}_{\succ}^{n_x}$ and $S \in \mathcal{S}_{\succ}^{n_u}$ are the cost function matrices; and given the arbitrarily small scalar $\varepsilon > 0$, $\underline{x}_\varepsilon = \underline{x} + \varepsilon \mathbb{1}_{n_x}$, $\overline{x}_\varepsilon = \overline{x} - \varepsilon \mathbb{1}_{n_x}$, $\underline{u}_\varepsilon = \underline{u} + \varepsilon \mathbb{1}_{n_u}$ and $\overline{u}_\varepsilon = \overline{u} - \varepsilon \mathbb{1}_{n_u}$. The ε-tightened constraints (3h) and (3i) are considered to avoid a possible controllability loss if any constraint is active at the equilibrium point [8].

The MPCT formulation (3) has several advantages with respect to standard MPC [1], such as guaranteed recursive feasibility under nominal conditions, i.e., when controlling the model used as prediction model with no disturbances, or asymptotic stability to the admissible steady state $(\hat{x}, \hat{u})$ that minimizes $\|\hat{x} - x_r\|_T^2 + \|\hat{u} - u_r\|_S^2$ [8]. However, the inclusion

---

Algorithm 1: ADMM

Require : v^0, λ^0, $\rho > 0$, $\epsilon_p > 0$, $\epsilon_d > 0$

1   $k \leftarrow 0$
2   **repeat**
3      $z^{k+1} \leftarrow \arg\min_z \mathcal{L}_\rho(z, v^k, \lambda^k)$
4      $v^{k+1} \leftarrow \arg\min_v \mathcal{L}_\rho(z^{k+1}, v, \lambda^k)$
5      $\lambda^{k+1} \leftarrow \lambda^k + \rho(Cz^{k+1} + Dv^{k+1})$
6      $k \leftarrow k + 1$
7   **until** $\|Cz^{k+1} + Dv^{k+1}\|_\infty \le \epsilon_p$ *and* $\|v^{k+1} - v^k\|_\infty \le \epsilon_d$

Output: $\tilde{z}^* \leftarrow z^k$, $\tilde{v}^* \leftarrow v^k$, $\tilde{\lambda}^* \leftarrow \lambda^k$

---

of (x_s, u_s) leads to a more complex OP than the one of standard MPC, as the banded structure that arises when solving the OP of MPC is lost in (3). We note that the banded structure of standard MPC is crucial for the implementation of efficient solvers [3], [10]. Thus, even though MPCT only adds $n_x + n_u$ extra decision variables with respect to standard MPC, the time required to solve (3) can be notably higher if a naive approach is used to solve the OP, e.g., if non-sparse matrices are used when solving the OP. In the following section we present an efficient ADMM-based solver for (3).

III. EFFICIENTLY SOLVING MPCT USING ADMM

We now show how to efficiently solve (3) using the ADMM algorithm [13] by decomposing its most computationally expensive step into several simple-to-solve steps. We start by describing the version of the ADMM we consider.

A. Alternating Direction Method of Multipliers

Let $f \colon \mathbb{R}^{n_z} \to (-\infty, \infty]$ and $g \colon \mathbb{R}^{n_z} \to (-\infty, \infty]$ be proper, closed and convex functions, $z, v \in \mathbb{R}^{n_z}$, and C, $D \in \mathbb{R}^{n_z \times n_z}$. Consider the OP

$$\min_{z,v} \quad f(z) + g(v) \tag{4a}$$
$$\text{s.t.} \quad Cz + Dv = 0, \tag{4b}$$

with augmented Lagrangian $\mathcal{L}_\rho \colon \mathbb{R}^{n_z} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_z} \to \mathbb{R}$,

$$\mathcal{L}_\rho(z, v, \lambda) = f(z) + g(v) + \lambda^\top (Cz + Dv) + \frac{\rho}{2}\|Cz + Dv\|_2^2,$$

where $\lambda \in \mathbb{R}^{n_z}$ is the vector of dual variables and the scalar $\rho > 0$ is the penalty parameter. We denote a solution of (4) by (z^*, v^*, λ^*), provided that one exists.

Starting from an initial point (v^0, λ^0), ADMM, shown in Algorithm 1, returns a suboptimal solution $(\tilde{z}^*, \tilde{v}^*, \tilde{\lambda}^*)$ of (4), where suboptimality is determined by the choice of the primal and dual exit tolerances $\epsilon_p > 0$ and $\epsilon_d > 0$ [13, §3.3].

B. Applying ADMM to MPCT

For $y, \underline{y}, \overline{y} \in \mathbb{R}^{n_y}$, $\hat{G} \in \mathbb{R}^{m_y \times n_y}$, $\hat{b} \in \mathbb{R}^{m_y}$, let us define

$$\mathcal{I}_{[\underline{y}, \overline{y}]}(y) = \begin{cases} 0, & \text{if } \underline{y} \le y \le \overline{y}, \\ +\infty, & \text{otherwise}, \end{cases}$$

$$\mathcal{I}_{(\hat{G}y = \hat{b})}(y) = \begin{cases} 0, & \text{if } \hat{G}y = \hat{b}, \\ +\infty, & \text{otherwise}. \end{cases}$$

Problem (3) can be posed as (4) by taking $C = \mathrm{I}$, $D = -\mathrm{I}$,

$$z = (x_0, u_0, x_1, u_1, \ldots, x_{N-1}, u_{N-1}, x_s, u_s),$$

and

$$v = (\tilde{x}_0, \tilde{u}_0, \tilde{x}_1, \tilde{u}_1, \ldots, \tilde{x}_{N-1}, \tilde{u}_{N-1}, \tilde{x}_s, \tilde{u}_s)$$

as a copy of the decision variables of (3), leading to

$$f(z) = \frac{1}{2} z^\top H z + q^\top z + \mathcal{I}_{(Gz=b)}(z), \tag{5a}$$

$$g(v) = \mathcal{I}_{[\underline{v}, \overline{v}]}(v) = \mathcal{I}_{[\underline{x}_\varepsilon, \overline{x}_\varepsilon]}(\tilde{x}_s) + \mathcal{I}_{[\underline{u}_\varepsilon, \overline{u}_\varepsilon]}(\tilde{u}_s) \tag{5b}$$

$$+ \sum_{i=1}^{N-1} \mathcal{I}_{[\underline{x}, \overline{x}]}(\tilde{x}_i) + \sum_{i=0}^{N-1} \mathcal{I}_{[\underline{u}, \overline{u}]}(\tilde{u}_i),$$

where $q = -(0, 0, \ldots, 0, T x_r, S u_r)$, $b = (x(t), \ldots, 0)$,

$$H = \begin{bmatrix} Q & 0 & \cdots & -Q & 0 \\ 0 & R & \cdots & 0 & -R \\ 0 & 0 & \ddots & \vdots & \vdots \\ -Q & 0 & \cdots & NQ+T & 0 \\ 0 & -R & \cdots & 0 & NR+S \end{bmatrix}, \tag{6a}$$

$$G = \begin{bmatrix} \mathrm{I} & 0 & 0 & 0 & \cdots & 0 \\ A & B & -\mathrm{I} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & A & B & -\mathrm{I} & 0 \\ 0 & 0 & 0 & 0 & (A-\mathrm{I}) & B \end{bmatrix}, \tag{6b}$$

$$\underline{v} \doteq (\underline{x}, \underline{u}, \ldots, \underline{x}, \underline{u}, \underline{x}_\varepsilon, \underline{u}_\varepsilon), \tag{6c}$$

$$\overline{v} \doteq (\overline{x}, \overline{u}, \ldots, \overline{x}, \overline{u}, \overline{x}_\varepsilon, \overline{u}_\varepsilon). \tag{6d}$$

With these elements, Step 3 of Algorithm 1 consists of solving a quadratic program subject to equality constraints, constituting the main computational load when solving MPCT with ADMM. On the other hand, Step 4 of the algorithm requires solving a simple separable convex problem (i.e., solving n_z simple scalar OPs). The next subsections are devoted to explaining how these steps are computed efficiently.

C. Efficient computation of z^{k+1}

Variable z^{k+1} updated in Step 3 of Algorithm 1 applied to problem (5a) is obtained from the optimal solution of

$$\min_z \frac{1}{2} z^\top P z + p^\top z \tag{7a}$$

$$\text{s.t. } Gz = b, \tag{7b}$$

where $P = H + \rho \mathrm{I}$ and $p = q + \lambda^k - \rho v^k$.

As shown in the following proposition, problem (7) can be solved by posing a linear system of equations describing its Karush-Kuhn-Tucker optimality conditions.

Proposition 1 ([14, §5.5.3]): Consider the OP (7), where $P \in \mathbb{R}^{n_z \times n_z}$ is positive semi-definite, $p \in \mathbb{R}^{n_z}$, $G \in \mathbb{R}^{m_z \times n_z}$ and $b \in \mathbb{R}^{m_z}$. A vector $z^* \in \mathbb{R}^{n_z}$ is an optimal solution of this problem if and only if there exists a vector $\mu \in \mathbb{R}^{m_z}$ such that

$$Gz^* = b, \tag{8a}$$

$$Pz^* + G^\top \mu + p = 0. \tag{8b}$$

As shown in [15], simple algebraic manipulations of (8) along with the definition of matrix $W \doteq GP^{-1}G^\top$ lead to the alternative form

$$P\xi = p, \tag{9a}$$

$$W\mu = -(G\xi + b), \tag{9b}$$

$$Pz^* = -(G^\top \mu + p), \tag{9c}$$

from where the optimal solution z^* of (7), and thus the update z^{k+1} of Step 3 of Algorithm 1, can be obtained. Solving (9) is the main computational burden of Algorithm 1. Thus, we wish to solve the three linear systems efficiently. However, matrices P and W are semi-banded due to the semi-banded structure of H shown in (6a). The following definition formalizes the notion of a *semi-banded* matrix.

Definition 1: Given the non-singular matrix $M \in \mathbb{R}^{n \times n}$ and vector $d \in \mathbb{R}^n$, we say that the linear system

$$Mz = d, \tag{10}$$

is *semi-banded* if there exists a non-singular banded matrix $\Gamma \in \mathbb{R}^{n \times n}$, and $U \in \mathbb{R}^{n \times m}$ and $V \in \mathbb{R}^{m \times n}$ satisfying

$$M = \Gamma + UV, \tag{11}$$

where the dimension m is assumed to be significantly smaller than the dimension of M, i.e., $m \ll n$.

A naive approach to solving the three linear systems (9) will generally be computationally expensive. However, we now show how the decomposition (11) can be used to solve (9) efficiently. We start by showing that, indeed, matrices P and W in (9) are semi-banded, providing explicit values for their decomposition (11) in the following proposition, which makes use of the well-known Woodbury matrix identity [16].

Lemma 1 (Woodbury matrix identity): Let $\Gamma \in \mathbb{R}^{n \times n}$ be non-singular. Then, if $\mathrm{I} + V\Gamma^{-1}U$ is non-singular, $\Gamma + UV$ is also non-singular and its inverse is given by

$$(\Gamma + UV)^{-1} = \Gamma^{-1} - \Gamma^{-1}U(\mathrm{I} + V\Gamma^{-1}U)^{-1}V\Gamma^{-1}. \tag{12}$$

Proposition 2: Matrices P and W of (9) are semi-banded and can be decoupled as $P = \widehat{\Gamma} + \widehat{U}\widehat{V}$ and $W = \tilde{\Gamma} + \tilde{U}\tilde{V}$, where, denoting $Y \doteq -\mathbb{1}_N^\top \otimes \mathtt{diag}(Q, R)$,

$$\widehat{\Gamma} = \mathtt{diag}(Q, R, Q, R \ldots, NQ+T, NR+S) + \rho \mathrm{I},$$

$$\widehat{U} = \begin{bmatrix} Y^\top & 0 \\ 0 & \mathrm{I}_{(n_x+n_u)} \end{bmatrix}, \quad \widehat{V} = \begin{bmatrix} 0 & \mathrm{I}_{(n_x+n_u)} \\ Y & 0 \end{bmatrix},$$

$$\tilde{\Gamma} = G\widehat{\Gamma}^{-1}G^\top,$$

$$\tilde{U} = -G\widehat{\Gamma}^{-1}\widehat{U}(\mathrm{I}+\widehat{V}\widehat{\Gamma}^{-1}\widehat{U})^{-1}, \tilde{V} = \widehat{V}\widehat{\Gamma}^{-1}G^\top.$$

Moreover, provided that G is full column rank, matrices P, W, $\widehat{\Gamma}$ and $\tilde{\Gamma}$ are positive definite.

Proof: The decomposition $P = \widehat{\Gamma} + \widehat{U}\widehat{V}$ immediately follows from the definition of $P \doteq H + \rho \mathrm{I}$ and the semi-banded structure of H shown in (6a). Next, by applying (12) to P we have that

$$P^{-1} = \widehat{\Gamma}^{-1} - \widehat{\Gamma}^{-1}\widehat{U}(\mathrm{I} + \widehat{V}\widehat{\Gamma}^{-1}\widehat{U})^{-1}\widehat{V}\widehat{\Gamma}^{-1}.$$

Thus, from the definition $W \doteq GP^{-1}G^\top$, we have

$$W = G(\widehat{\Gamma}^{-1} - \widehat{\Gamma}^{-1}\widehat{U}(I + \widehat{V}\widehat{\Gamma}^{-1}\widehat{U})^{-1}\widehat{V}\widehat{\Gamma}^{-1})G^\top$$
$$= G\widehat{\Gamma}^{-1}G^\top - G\widehat{\Gamma}^{-1}\widehat{U}(I + \widehat{V}\widehat{\Gamma}^{-1}\widehat{U})^{-1}\widehat{V}\widehat{\Gamma}^{-1}G^\top,$$

from where the claim $W = \tilde{\Gamma} + \tilde{U}\tilde{V}$ then follows from the definitions of $\tilde{\Gamma}$, $\tilde{U}$ and $\tilde{V}$. Finally, the fact that $\tilde{\Gamma}$ is banded-diagonal follows from the banded-diagonal structures of $\widehat{\Gamma}$ and G (6b), as shown in [3, Eq. (33)].

We notice that H is the matrix that corresponds to the quadratic terms of MPC formulation (3). From the convexity of the quadratic cost codified by H, we infer that $H \succeq 0$. Therefore, $P = H + \rho I \succeq \rho I \succ 0$. Also, from the positive definite nature of matrices Q, R, S and T, we have that the block-diagonal matrix $\widehat{\Gamma}$ satisfies $\widehat{\Gamma} \succeq \rho I \succ 0$. Since P is positive definite, $W = GP^{-1}G^\top$ is also positive definite provided that G is full column rank. The same argument applies to $\tilde{\Gamma} = G\widehat{\Gamma}^{-1}G^\top$. ∎

Next, we show in the following proposition that (10), and thus (9) by virtue of Proposition 2, can be solved by means of Algorithm 2. This result also follows from Lemma 1.

Proposition 3: Consider the semi-banded system $Mz = d$ of Definition 1 and its decomposition $M = \Gamma + UV$, where both M and Γ are non-singular matrices. Algorithm 2 returns a solution $\tilde{z}$ satisfying $M\tilde{z} = d$.

Proof: From $\det(M) \neq 0$ and $\det(\Gamma) \neq 0$ we obtain

$$0 \neq \det(\Gamma + UV) = \det(\Gamma)\det(I_n + \Gamma^{-1}UV)$$
$$= \det(\Gamma)\det(I_m + V\Gamma^{-1}U), \qquad (13)$$

where the last equality is due to the well-known identity $\det(I_n + AB) = \det(I_m + BA)$, $\forall A \in \mathbb{R}^{n \times m}, \forall B \in \mathbb{R}^{m \times n}$. Thus, we infer from (13) that $I_m + V\Gamma^{-1}U$ is non-singular. From this, and (12), we have that $M^{-1} = (\Gamma + UV)^{-1}$ can be written as $M^{-1} = \Gamma^{-1} - \Gamma^{-1}U(I_m + V\Gamma^{-1}U)^{-1}V\Gamma^{-1}$. Therefore,

$$\tilde{z} = M^{-1}d = \Gamma^{-1}d - \Gamma^{-1}U(I_m + V\Gamma^{-1}U)^{-1}V\Gamma^{-1}d. \quad (14)$$

Defining $z_1 = \Gamma^{-1}d$, $z_2 = (I_m + V\Gamma^{-1}U)^{-1}V\Gamma^{-1}d$ and $z_3 = \Gamma^{-1}U(I_m + V\Gamma^{-1}U)^{-1}V\Gamma^{-1}d$, we obtain Step 1 of Algorithm 2 by the definition of z_1, Step 2 by including the definition of z_1 into z_2, and Step 3 by substituting the definition of z_2 into z_3. Finally, $\tilde{z} = z_1 - z_3$ by substituting the definitions of z_1 and z_3 into (14). ∎

We notice that the computation of z_1 and z_3 in Algorithm 2 can be done efficiently by exploiting the banded structure of Γ, e.g., using a banded Cholesky decomposition if Γ is positive definite [3]. Moreover, since $I_m + V\Gamma^{-1}U \in \mathbb{R}^{m \times m}$, and we assume that $m \ll n$, z_2 is the solution of a small-dimensional linear system, which is thus computationally cheap to solve in comparison to Steps 1 and 3 of Algorithm 2.

Corollary 1: By means of Proposition 3, the optimal solution z^* of problem (7) can be obtained by using Algorithm 2 to solve the three linear systems (9) using the decomposition of matrices P and W provided in Proposition 2.

Remark 1: Matrix $\widehat{\Gamma} \succ 0$ in Proposition 2 is block-diagonal. Therefore, Steps 1 and 3 of Algorithm 2 applied to solve (9a) and (9c) are very simple. On the other hand,

---

Algorithm 2: Solve semi-banded system $(\Gamma + UV)\tilde{z} = d$

Require: Γ, U, V, d
1 Compute z_1 solving $\Gamma z_1 = d$
2 Compute z_2 solving $(I + V\Gamma^{-1}U)z_2 = Vz_1$
3 Compute z_3 solving $\Gamma z_3 = Uz_2$
Output: $\tilde{z} \leftarrow z_1 - z_3$

Algorithm 3: Efficient ADMM applied to MPCT (3)

Require: $\rho > 0$, $\epsilon_p > 0$, $\epsilon_d > 0$, Q, R, S, T, N
Input: $x(t)$, x_r, u_r, v^0, λ^0,
1 $k \leftarrow 0$
2 Compute q and b in (5a) using $x(t)$, x_r and u_r.
3 **repeat**
4     $p \leftarrow q + \lambda^k - \rho v^k$
5     $\xi \leftarrow$ solution of $P\xi = p$ using Alg. 2
6     $\mu \leftarrow$ solution of $W\mu = -(G\xi + b)$ using Alg. 2
7     $z^{k+1} \leftarrow$ solution of $Pz = -(G^\top\mu + p)$ using Alg. 2
8   Update v^{k+1} using (16)
9     $\lambda^{k+1} \leftarrow \lambda^k + \rho(z^{k+1} - v^{k+1})$
10     $k \leftarrow k + 1$
11 **until** $\|z^{k+1} - v^{k+1}\|_\infty \leq \epsilon_p$ *and* $\|v^{k+1} - v^k\|_\infty \leq \epsilon_d$
Output: $u(t) \leftarrow$ elements of v^k corresponding to $\tilde{u}_0$

---

matrix $\tilde{\Gamma} \succ 0$ is banded-diagonal, but not block-diagonal. However, $\tilde{\Gamma}$ has the same banded-diagonal structure that is exploited by the solvers proposed in [3], [11]. Therefore, Steps 1 and 3 of Algorithm 2 applied to (9b) can be solved by computing the banded Cholesky decomposition of $\tilde{\Gamma}$ and using [15, Alg. 11].

Remark 2: Note that the operations $G\xi$ and $G^\top\mu$ in (9b) and (9c) can be performed sparsely, since G (6b) is sparse.

D. Computation of v^{k+1}

Variable v^{k+1} updated in Step 4 of Algorithm 1, when $g(v)$ is given by (5b), is taken from the optimal solution of

$$\min_{v \in \mathbb{R}^{n_z}} \frac{\rho}{2}\sum_{j=1}^{n_z}(v_{(j)}^2 - 2z_{(j)}^{k+1}v_{(j)}) - \sum_{j=1}^{n_z}\lambda_{(j)}^k v_{(j)}$$
$$\text{s.t. } \underline{x} \leq x_i \leq \overline{x}, \ i \in \mathbb{I}_1^{N-1},$$
$$\underline{u} \leq u_i \leq \overline{u}, \ i \in \mathbb{I}_0^{N-1}, \qquad (15)$$
$$\underline{x}_\varepsilon \leq x_s \leq \overline{x}_\varepsilon,$$
$$\underline{u}_\varepsilon \leq u_s \leq \overline{u}_\varepsilon,$$

which is separable for each decision variable $v_{(j)}$. Indeed, each element $v_{(j)}^{k+1}$, $j \in \mathbb{I}_1^{n_z}$, of v^{k+1} is given by

$$v_{(j)}^{k+1} = \min\left(\max\left(z_{(j)}^{k+1} + \frac{1}{\rho}\lambda_{(j)}^k, \underline{v}_{(j)}\right), \overline{v}_{(j)}\right). \quad (16)$$

E. Comparison with the EADMM-based solver

Algorithm 3 shows the particularization of Algorithm 1 applied to the MPCT problem (3) using the results presented

in the previous subsections. Steps 5, 6 and 7 of Algorithm 3 are its main computational burden. They make use of Algorithm 2 to solve the three semi-banded linear systems in (9) to update the decision variables z^{k+1}. As discussed in Remark 1, Step 6 requires solving two linear systems whose matrix has the structure obtained when solving standard MPC formulations. The EADMM solver (3) proposed in [11] also recovers this very same banded structure. However, it only needs to solve it once, instead of twice. Additionally, the EADMM solver needs a problem computationally equivalent to Step 8 of Algorithm 3 and a $(n_x + n_u)$ dense linear system, whereas Algorithm 3 needs to solve three $2(n_x + n_u)$-dimensional linear system (one for each call to Algorithm 2) and four block-diagonal systems $\hat{\Gamma} z = d$ (two in Step 5 and two in Step 7).

We conclude that the computational cost per iteration of Algorithm 3 is more than double the one of the EADMM solver proposed in [11]. Therefore, ADMM should require, on average, less than half the number of iterations than the EADMM solver from [11] to be computationally better. However, the are several advantages to using ADMM instead of EADMM. First, the convergence of EADMM is only guaranteed if its step-size ρ belongs to a certain range that depends on the properties of the OP [12, Theorem 3.1]. However, in practice the EADMM algorithm typically performs very poorly when using values of ρ satisfying this theoretical condition. On the other hand, the value of ρ for Algorithm 3 can be freely chosen, thus improving the practical performance of the algorithm. Second, the theoretical convergence results for ADMM are better than the current ones available for EADMM, leading also to a better worst-case iteration complexity. Finally, even when choosing values of ρ for EADMM according to [12, Theorem 3.1], the number of iterations required by the algorithm in practice is typically much larger than the ones required by ADMM when applied to the same OP.

IV. NUMERICAL RESULTS

We provide a computational comparison between the proposed Algorithm 3 and the EADMM solver for MPCT presented in [11]. We consider the ball and plate system presented in [17, §V.A], which consists of a ball whose position on a (nominally) horizontal plate is controlled by motors on each of its two main axes. Consequently, the system has two inputs, angular accelerations [rad/s^2] of the motors, and eight states, position [m] and velocity [m/s] of the ball with respect to each axis, as well as angular position [rad] and velocity [rad/s] of the plate in each axis. The physical parameters of the system are the same as in [17, §V.A], as well as the sample time of 0.2 seconds. We take the constraints (2) as $\overline{u} = (0.2, 0.2)$, $\underline{u} = -\overline{u}$, $\overline{x} = (2, 1, 0.785, \infty, 2, 1, 0.785, \infty)$, $\underline{x} = -(0, 1, 0.785, \infty, 0, 1, 0.785, \infty)$, and, for (3), $N=30$, $\varepsilon=10^{-6}$, $R = \texttt{diag}(0.5, 0.5)$, $S = \texttt{diag}(0.3, 0.3)$,

$$Q = \texttt{diag}(10, 0.05, 0.05, 0.05, 10, 0.05, 0.05, 0.05),$$
$$T = \texttt{diag}(200, 50, 50, 50, 200, 50, 50, 50),$$

where the order of the state and input elements are taken from [17]. Additionally, we take the exit tolerances of Algorithm 3 and EADMM as 10^{-4}, i.e., $\epsilon_p = \epsilon_d = 10^{-4}$. Finally, we note that we use the scaling of the system model used in [11] to improve the numerical conditioning of (3).

Using version v0.3.11 of the Spcies Toolbox [18] in an I5-1135G7 laptop, Table I shows results on the number of iterations and computation time of Algorithm 3 and the EADMM solver from [11] (for different values of ρ) when applied to the above system for 500 random initial states, where the position and velocity of the ball are taken from a uniform distribution in the intervals $[0.3, 1.8]$ and $[-0.2, 0.2]$, respectively. Additionally, we consider two references: the *Reachable* reference $x_r = (1, 0, 0, 0, 0.8, 0, 0, 0)$, and the *Unreachable* reference $x_r = (2.15, 0, 0, 0, 2.2, 0, 0, 0)$, which violates the constraints on the position of the ball. In both cases $u_r = (0, 0)$.

The results indicate that Algorithm 3 is noticeably faster than the EADMM solver when the reference is *Reachable*. However, when it is *Unreachable*, the number of iterations and computation times grow considerably for Algorithm 3. Even though ADMM seems less efficient in that specific case, asymptotic convergence of the algorithm is guaranteed for any positive value of ρ [13], while EADMM applied to our case study requires $\rho \in (0, 0.0176)$ (see [12, Theorem 3.1]). If we take ρ for EADMM in the range such that its asymptotic convergence is guaranteed, Algorithm 3 outperforms EADMM in both the *Reachable* and *Unreachable* cases, as it leads to a poor performance of EADMM.

Figure 1 illustrates the closed-loop evolution of the system controlled by applying (3) to one of the 500 tests with *Reachable* reference from Table I. Figure 2 shows the closed-loop evolution considering instead the *Unreachable* reference. As mentioned in Section II, Figure 2 shows that when the reference is infeasible, the MPCT formulation (3) steers the system to its closest admissible steady-state.

V. CONCLUSION

This article has presented an efficient method for solving the MPCT formulation using ADMM. We have shown how the semi-banded structure of the most computationally-expensive step of the ADMM algorithm can be solved efficiently by decomposing it into three simpler-to-solve steps. Indeed, the decomposition recovers the simple matrix structure exploited by several first-order solvers for standard MPC from the literature. We have presented numerical results comparing the proposed solver with a recent EADMM-based MPCT solver from the literature, showing that the proposed approach may outperform the EADMM algorithm from a computational point of view. This, along with the solid theoretical convergence guarantees of ADMM, results in a rather sensible and efficient solution for the practical implementation of MPCT.
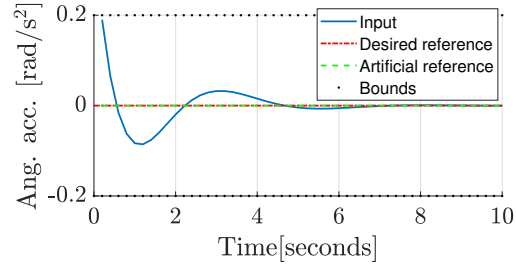
REFERENCES

[1] E. F. Camacho and C. B. Alba, *Model Predictive Control.* Springer Science & Business Media, 2013.

Algorithm	ρ	Reference	Iterations				Computation time [ms]			
			Average	Median	Max.	Min.	Avg.	Median	Max.	Min.
ADMM	0.1	Reachable	14.2	7.0	73.0	5.0	0.52	0.24	4.14	0.18
ADMM	0.6	Reachable	16.5	17.0	18.0	11.0	0.60	0.60	0.12	0.39
ADMM	2	Reachable	37.8	39.0	43.0	22.0	1.36	1.37	2.55	0.78
EADMM	4	Reachable	222.3	225.0	298.0	119.0	1.65	1.68	2.94	0.87
EADMM	6	Reachable	178.5	180.0	206.0	131.0	1.34	1.34	2.24	0.97
EADMM	10	Reachable	228.7	231.0	280.0	173.0	1.66	1.66	2.57	1.24
ADMM	4	Unreachable	90.5	80.0	721.0	68.0	3.24	2.87	25.90	2.41
ADMM	6	Unreachable	112.5	110.0	483.0	91.0	4.03	3.91	17.45	3.23
ADMM	10	Unreachable	159.3	160.0	319.0	126.0	5.61	5.64	12.56	4.43
EADMM	2	Unreachable	717.6	662.0	1418.0	526.0	5.31	4.90	10.43	3.84
EADMM	6	Unreachable	235.3	221.0	459.0	193.0	1.85	1.74	3.69	1.44
EADMM	10	Unreachable	289.0	295.0	343.0	227.0	2.08	2.12	3.13	1.65

TABLE I: Performance of Algorithm 3 (ADMM) and EADMM [11] applied to the ball and plate system for random current states.
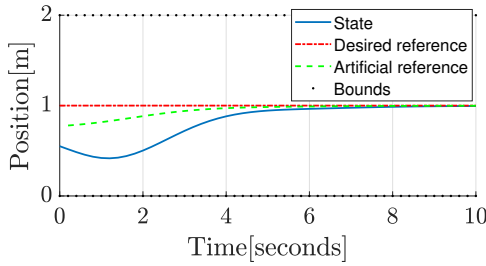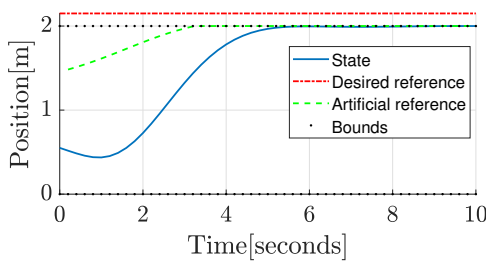

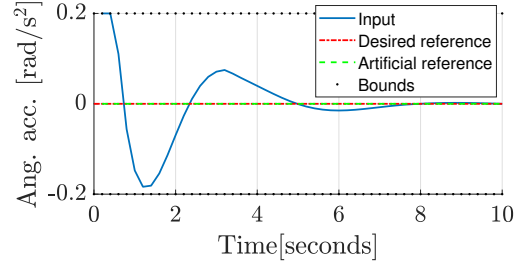(a) Position in the horizontal axis.

(b) Angular acceleration of the horizontal axis.

Fig. 1: Closed-loop trajectory of the ball and plate system with (3) for an admissible reference.

(a) Position in the horizontal axis.

(b) Angular acceleration of the horizontal axis.

Fig. 2: Closed-loop trajectory of the ball and plate system with (3) for a non-admissible reference.

[2] J. Rawlings, D. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017.

[3] 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, p. 1117–1130, 2021.

[4] 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.

[5] B. O'Donoghue, "Operator splitting for a homogeneous embedding of the linear complementarity problem," *SIAM Journal on Optimization*, vol. 31, pp. 1999–2023, August 2021.

[6] S. Richter, C. Jones, and M. Morari, "Computational complexity certification for real-time MPC with input constraints based on the fast gradient method," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1391–1403, 2012.

[7] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 18–33, 2014.

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

[9] J. Köhler and M. N. Zeilinger, "Globally stable and locally optimal model predictive control using a softened initial state constraint–extended version," *arXiv preprint arXiv:2207.10216*, 2022.

[10] 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.

[11] 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," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 4, pp. 1798–1805, 2021.

[12] X. Cai, D. Han, and X. Yuan, "On the convergence of the direct extension of ADMM for three-block separable convex minimization models with one strongly convex function," *Computational Optimization and Applications*, vol. 66, pp. 39–73, 2017.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[15] P. Krupa, "Implementation of MPC in embedded systems using first order methods," Ph.D. dissertation, University of Seville, 2021, available at arXiv:2109.02140.

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

[17] 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.

[18] 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.