

## Safe Trajectory Tracking in Uncertain Environments

Questa è la versione sottoposta a revisione paritaria (postprint) della seguente opera:

*Original*

Safe Trajectory Tracking in Uncertain Environments / Batkovic, Ivo; Ali, Mohammad; Falcone, Paolo; Zanon, Mario. - In: IEEE TRANSACTIONS ON AUTOMATIC CONTROL. - ISSN 0018-9286. - (2023), pp. 4204-4217. [10.1109/TAC.2022.3207875]

*Availability:*

This version is available at: 20.500.11771/21798

*Publisher:*

*Published*

DOI:10.1109/TAC.2022.3207875

*Terms of use:*

This publication is made accessible in accordance with the terms for deposit in the institutional repository, as defined by the IMT School for Advanced Studies Lucca's Open Access Policy. ([https://library.imtlucca.it/sites/default/files/regolamento-policy-open-access-imtlib\\_0.pdf](https://library.imtlucca.it/sites/default/files/regolamento-policy-open-access-imtlib_0.pdf)).

Si prega di consultare le pagine informative dell'editore relative alle politiche di autoarchiviazione.

(Article begins on next page)

# Safe Trajectory Tracking in Uncertain Environments

Ivo Batkovic<sup>1,2</sup>, Mohammad Ali<sup>2</sup>, Paolo Falcone<sup>1,3</sup>, and Mario Zanon<sup>4</sup>

**Abstract**—In Model Predictive Control (MPC) formulations of trajectory tracking problems, infeasible reference trajectories and a-priori unknown constraints can lead to cumbersome designs, aggressive tracking, and loss of recursive feasibility. This is the case, for example, in trajectory tracking applications for mobile systems in the presence of constraints which are not fully known a-priori. In this paper, we propose a new framework called Model Predictive Flexible trajectory Tracking Control (MPFTC), which relaxes the trajectory tracking requirement. Additionally, we accommodate recursive feasibility in the presence of a-priori unknown constraints, which might render the reference trajectory infeasible. In the proposed framework, constraint satisfaction is guaranteed at all times while the reference trajectory is tracked as good as constraint satisfaction allows, thus simplifying the controller design and reducing possibly aggressive tracking behavior. The proposed framework is illustrated with three numerical examples.

**Index Terms**—flexible trajectory tracking, nonlinear MPC, safety, uncertain constraints, stability, recursive feasibility

## I. INTRODUCTION

Model Predictive Control (MPC) is an advanced control technique for linear, or nonlinear systems, that has been made successful by the possibility of introducing time-varying references with preview information as well as constraints. Standard MPC formulations penalize deviations from a setpoint or a (feasible) reference trajectory and stability and recursive feasibility guarantees have been derived for such settings [1], [2], [3]. However, in practice (a) not all constraints, that the real system could be subject to, are available at the design stage, and (b) a reference trajectory which satisfies all constraints might therefore not be available. This is the case, for example, in trajectory tracking applications for autonomous vehicles, see, e.g., [4], [5], [6], [7]. For such problems, it would be convenient to use a reference trajectory that is easy to generate, and that is not necessarily feasible for all constraints, while detecting and enforcing a-priori unknown constraints online at all times. Existing literature on motion planning in dynamic environments [8], [9] proposes ways of tackling (b), where graph-based approaches have been rather successful [10], [11], [12]. In particular, in [6] the authors

presented a unified framework, where the motion planning and control problems are solved in order to control and guide a quadcopter in various configurations with multiple moving obstacles. The authors show the benefit of being able to plan new trajectories in such environments, however, no formal safety (recursive feasibility) guarantees are provided.

In this paper, we take a different point of view and deal with the presence of a-priori unknown constraints by constructing a new MPC framework which ensures safety, i.e., recursive feasibility of the system, while only requiring mild assumptions on the reference trajectory, which needs not be feasible with respect to all constraints. Consequently, we do provide safety guarantees without the need of the additional layer of motion planning, e.g., we can use reference trajectories that are generated offline as we do not rely on the reference trajectory to be updated online. Nevertheless, our approach can still be combined with motion planning if this is deemed beneficial, e.g., to improve performance.

In standard MPC, the presence of constraints, can cause the system to slow or stop, while the reference trajectory does not: in this case, as the system gets far from the reference, an undesirably aggressive behavior is obtained. On the other hand, while safety and stability are the most important requirements, a smooth, or at least not excessively aggressive, behavior is also desirable. To alleviate aggressive behaviors introduced by infeasible reference trajectories, Model Predictive Path Following Control (MPFC) has been proposed in [13], [14], [15], [16], [17], [18] by penalizing deviations from a reference path instead of a trajectory, where additional variables are introduced in order to control the position along the reference path. The main difficulty in MPFC is the need to select an appropriate output function to define the path typically in a dimension lower than the system state space. Although the results in this paper can be combined with MPFC, we also propose an alternative strategy which we call Model Predictive Flexible trajectory Tracking Control (MPFTC). Based on ideas similar to MPFC, we introduce new variables to artificially modify the time derivative of the reference trajectory in a time warping fashion. Moreover, we also discuss safety in a general sense, and give safety guarantees which build on the assumption that a safe set exists, where all (including the unknown) constraints are satisfied at all time: this is an assumption often made in practice for stable systems at rest, as long as a safe configuration can be found [19], [20], [21].

In this paper, the open problem of proving safe control in the presence of unknown constraints is addressed by the use of an ad-hoc combination of existing approaches. Through the introduction of two fundamental assumptions (Assumption 4 and 5), we show how one can still use well-established techniques to ensure recursive feasibility of the control problem.

This work was partially supported by the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation, and by the COPPLAR project (VINNOVA. V.P. Grant No. 2015-04849).

<sup>1</sup> Ivo Batkovic and Paolo Falcone are with the Mechatronics group at the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden {ivo.batkovic, falcone}@chalmers.se

<sup>2</sup> Ivo Batkovic, and Mohammad Ali are with the research department at Zenseact AB {ivo.batkovic, mohammad.ali}@zenseact.com

<sup>3</sup> Paolo Falcone is with the Dipartimento di Ingegneria “Enzo Ferrari” Università di Modena e Reggio Emilia, Italy falcone@unimore.it

<sup>4</sup> Mario Zanon is with the IMT School for Advanced Studies Lucca mario.zanon@imtlucca.it

In order to illustrate in detail the theoretical developments, we consider two toy examples and compare our formulation with existing ones. We then design an MPFTC controller for a robotic arm, which has to follow a trajectory while avoiding an a-priori unknown obstacle. The main contributions of this paper can be summarized as follows: (a) the introduction of the MPFTC framework for flexible trajectory tracking with stability guarantees; and (b) the development of a safe framework for satisfaction of a-priori unknown constraints.

This paper is structured as follows. In Section II we outline the flexible trajectory tracking problem and in Section III we prove stability for MPFTC. In Section IV we introduce a framework with recursive feasibility guarantees for a-priori unknown constraints. We illustrate the theoretical developments in Section V with three numerical examples. Finally, we draw conclusions in Section VI.

### A. Notation

We denote a discrete-time nonlinear system by

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad (1)$$

where  $\mathbf{x}_k \in \mathbb{R}^{n_x}$  and  $\mathbf{u}_k \in \mathbb{R}^{n_u}$  are the state and input vectors at time  $k$ , respectively. The state and inputs are subject to two categories of constraints: a-priori known constraints  $h(\mathbf{x}, \mathbf{u}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_h}$ ; and a-priori unknown constraints  $g(\mathbf{x}, \mathbf{u}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_g}$ , i.e., the state and inputs must satisfy  $h(\mathbf{x}, \mathbf{u}) \leq 0$  and  $g(\mathbf{x}, \mathbf{u}) \leq 0$ , where the inequalities are defined element-wise. Note that constraint  $g$  is referred to as a-priori unknown due to the fact that the value  $g(\mathbf{x}, \mathbf{u})$  cannot be known beforehand. However, we must stress that both the functional forms of  $h$  and  $g$  are assumed to be known.

We use the notation  $g_{n|k}(\mathbf{x}, \mathbf{u})$  to denote  $g$  at time  $n$ , given the information available at time  $k$ . Moreover, we will denote by  $g_n(\mathbf{x}, \mathbf{u}) := g_{n|\infty}(\mathbf{x}, \mathbf{u}) = g_{n|k}(\mathbf{x}, \mathbf{u})$ ,  $\forall k \geq n$  the real constraint, while in general  $g_{n|k}(\mathbf{x}, \mathbf{u}) \neq g_n(\mathbf{x}, \mathbf{u})$ ,  $\forall k < n$ . Note that for a-priori known constraints  $h_{n|k}(\mathbf{x}, \mathbf{u}) := h_n(\mathbf{x}, \mathbf{u})$  holds by definition. We apply the same notation to state and inputs, e.g.,  $\mathbf{x}_{n|k}$  and  $\mathbf{u}_{n|k}$  denote the predicted state and input at time  $n$  given the current time  $k$ . In addition, to denote a set of integers, we use  $\mathbb{I}_a^b := \{a, a+1, \dots, b\}$ .

## II. PROBLEM DESCRIPTION

Our aim is to control system (1) such that both known constraints  $h(\mathbf{x}, \mathbf{u}) \leq 0$  and a-priori unknown constraints  $g(\mathbf{x}, \mathbf{u}) \leq 0$  are satisfied at all times. A situation in which some constraints cannot be known a priori occurs in the context of trajectory tracking for mobile robots, where  $h$  includes, e.g., actuator saturations and/or imposes the avoidance of collisions with fixed and known obstacles, while  $g$  includes, e.g., moving obstacles present in the environment, whose motion trajectories are not known a-priori, but can be estimated online based on measurements. In order to construct  $g$  one needs prediction models that capture the overall behavior of such moving obstacles in the environment and are able to provide rigorous though possibly conservative bounds. A deeper discussion regarding the construction of  $g$  is presented in Section IV-A.

Our first and essential objective is to guarantee safety of (1), which we define formally as

**Definition 1** (Safety). *A controller is said to be safe in a given set  $\mathcal{S} \subseteq \mathbb{R}^{n_x}$  if  $\forall \mathbf{x} \in \mathcal{S}$  it generates control inputs  $\mathbf{U} = \{\mathbf{u}_0, \dots, \mathbf{u}_\infty\}$  and corresponding state trajectories  $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_\infty\}$  such that  $h_k(\mathbf{x}_k, \mathbf{u}_k) \leq 0$  and  $g_k(\mathbf{x}_k, \mathbf{u}_k) \leq 0$ ,  $\forall k \geq 0$ .*

Our second objective is to control the system such that the state and input  $(\mathbf{x}_k, \mathbf{u}_k)$  track a parametrized reference trajectory  $\mathbf{r}(\tau) := (\mathbf{r}^x(\tau), \mathbf{r}^u(\tau))$  as closely as safety allows. If the reference parameter  $\tau$  is selected to be time, its natural dynamics are given by

$$\tau_{k+1} = \tau_k + t_s, \quad (2)$$

where  $t_s$  is the sampling time for sampled-data systems and  $t_s = 1$  in the discrete-time framework. Given the presence of nonlinear dynamics and constraints, we frame the problem in the context of MPC. Note that if  $\tau$  is forced to follow its natural dynamics (2), then the reference tracking problem in the absence of a-priori unknown constraints  $g$  is a standard MPC problem and, therefore, inherits all stability guarantees, but also a possibly aggressive behavior when the initial state is far from the reference.

Approaches developed for mechanical systems in presence of large tracking errors, especially caused by reference setpoint changes, have been proposed in, e.g., [22]. In the setting we consider, however, perfect tracking will not be impeded by sudden setpoint changes, but rather by the presence of constraints, e.g., a mobile system might have to temporarily stop in order to avoid collisions with other systems or obstacles. Therefore, we investigate complementary approaches to those proposed in [22].

One family of approaches for smooth reference tracking is the so-called MPFC [14], [17]. While MPFC is a valid technique for tackling our problem we propose a new, alternative, approach: MPFTC, which solves the problem of tracking an infeasible reference trajectory when the presence of constraints  $g(\mathbf{x}, \mathbf{u}) \leq 0$  might force the system to temporarily deviate from the reference. While the main difficulty in MPFC is to establish a suitable output  $y = \phi(\mathbf{x}, \mathbf{u})$  and the corresponding path, the main difficulty in MPFTC will be to pre-compute a parametrized feasible reference.

## III. MODEL PREDICTIVE FLEXIBLE TRACKING CONTROL

The main idea in MPFTC is to avoid aggressive tracking behaviors by adjusting  $\tau_k$ , which acts as a fictitious time of the reference trajectory, through relaxed dynamics given by

$$\tau_{k+1} = \tau_k + t_s + v_k, \quad (3)$$

where  $v$  is an additional auxiliary control input and  $\tau$  becomes an auxiliary state. By relaxing the natural dynamics of  $\tau_k$  we can control how the reference trajectory will be tracked, e.g.,  $v_k$  can be used to adjust the reference trajectory dynamics in the presence of obstacles. We illustrate the advantage of MPFTC over MPC in the double integrator example in Section V-C. Note that the system dynamics are unchanged

and the fictitious time  $\tau$  makes only the reference dynamics deviate from the natural ones.

We formulate the MPFTC problem as the following MPC problem

$$V(\mathbf{x}_k, \tau_k) := \min_{\substack{\mathbf{x}, \mathbf{u} \\ \tau, v}} \sum_{n=k}^{k+N-1} q_r(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_{n|k}) + wv_{n|k}^2 \quad (4a)$$

$$+ p_r(\mathbf{x}_{k+N|k}, \tau_{k+N|k})$$

s.t.  $\mathbf{x}_{k|k} = \mathbf{x}_k, \tau_{k|k} = \tau_k,$  (4b)

$$\mathbf{x}_{n+1|k} = f(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}), \quad n \in \mathbb{I}_k^{k+N-1}, \quad (4c)$$

$$\tau_{n+1|k} = \tau_{n|k} + t_s + v_{n|k}, \quad n \in \mathbb{I}_k^{k+N-1}, \quad (4d)$$

$$h_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, \quad n \in \mathbb{I}_k^{k+N-1}, \quad (4e)$$

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, \quad n \in \mathbb{I}_k^{k+N-1}, \quad (4f)$$

$$\mathbf{x}_{k+N|k} \in \mathcal{X}_r^f(\tau_{k+N|k}), \quad (4g)$$

where  $k$  is the current time,  $N$  is the prediction horizon, and  $w > 0$  is the weight defining the cost associated with the auxiliary input  $v_{n|k}$ , which can take any real value. In tracking MPC, typical choices for the stage and terminal costs are

$$\Delta \mathbf{x}_{n|k} := \mathbf{x}_{n|k} - \mathbf{r}^x(\tau_{n|k}), \quad \Delta \mathbf{u}_{n|k} := \mathbf{u}_{n|k} - \mathbf{r}^u(\tau_{n|k}),$$

$$q_r(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_{n|k}) = \begin{bmatrix} \Delta \mathbf{x}_{n|k} \\ \Delta \mathbf{u}_{n|k} \end{bmatrix}^\top W \begin{bmatrix} \Delta \mathbf{x}_{n|k} \\ \Delta \mathbf{u}_{n|k} \end{bmatrix}, \quad (5)$$

$$p_r(\mathbf{x}_{k+N|k}, \tau_{k+N|k}) = \Delta \mathbf{x}_{k+N|k}^\top P \Delta \mathbf{x}_{k+N|k}, \quad (6)$$

where  $\mathbf{r}(\tau_{n|k}) = (\mathbf{r}^x(\tau_{n|k}), \mathbf{r}^u(\tau_{n|k}))$  is a user-provided reference trajectory. Note that the cost functions  $q_r$  and  $p_r$  depend on  $\tau_{n|k}$  only through the reference trajectory. The matrices  $W \in \mathbb{R}^{(n_x+n_u) \times (n_x+n_u)}$  and  $P \in \mathbb{R}^{n_x \times n_x}$  are symmetric positive-definite matrices. Note that we use convex quadratic forms for the cost for simplicity, but the proposed framework can accommodate more general cost definitions. The predicted state and controls are defined as  $\mathbf{x}_{n|k}$ ,  $\tau_{n|k}$ , and  $\mathbf{u}_{n|k}$ ,  $v_{n|k}$  respectively. Constraint (4b) enforces that the prediction starts at the current states, and constraints (4c)-(4d) enforce that the predicted trajectories satisfy the system dynamics. Constraints (4e) denote known constraints such as, e.g., actuator saturations and reference trajectory bounds, while constraint (4f) enforces constraints which are not known a-priori like, e.g., constraints imposed to avoid the collision with obstacles detected by a perception layer. Finally, constraint (4g) is a terminal set, where, differently from standard formulations, the terminal constraint depends on the auxiliary state  $\tau_{k+N|k}$  relative to the reference parameter. Note that, while the introduction of one additional state and control results in an increased computational complexity, such increase is typically small, since these variables have decoupled linear dynamics.

**Remark 1.** If the constraint  $v_{n|k} = 0$  is added and constraints (4f) are not present, a standard MPC formulation is obtained. The terminal set  $\mathcal{X}_r^f$  can therefore be designed as in standard MPC, where one assumes that the reference trajectory evolves according to its natural dynamics (2). The challenges introduced by constraints (4f) will be further analyzed in the remainder of the paper.

In the following, we will first prove stability for MPFTC under the standard assumptions used to prove stability for

MPC, i.e., we will assume that (4f) is inactive at the reference and does not jeopardize recursive feasibility. Because, unlike in the standard setting, in which constraints (4f) are assumed to be known, in our setting constraints (4f) are not known a-priori, Assumptions 2b and 3b become unrealistic. We will first rely on these restrictive assumptions both for the sake of simplicity and to clearly indicate how our second contribution will extend the state of the art by replacing these assumptions with more realistic ones. While this issue is typically neglected in the MPC literature, situations in which a-priori unknown constraints are present are rather common in practice. Consequently, we believe that guaranteeing recursive feasibility and closed-loop system stability is still an open problem which can limit the practical applicability. In Section IV we will introduce a framework based on ideas typically used in robust MPC which does guarantee recursive feasibility by a slight modification of the standard MPC controller design.

In order to prove stability, we introduce the following assumptions which coincide with those commonly used in MPC, see, e.g., [2], [23].

**Assumption 1** (System and cost regularity). *The system model  $f$  is continuous, and the stage cost  $q_r : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ , and terminal cost  $p_r : \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ , are continuous at the origin and satisfy  $q_r(\mathbf{r}^x(\tau), \mathbf{r}^u(\tau), \tau) = 0$ , and  $p_r(\mathbf{r}^x(\tau), \tau) = 0$ . Additionally,  $w > 0$ , and  $q_r(\mathbf{x}_k, \mathbf{u}_k, \tau_k) \geq \alpha_1(\|\mathbf{x}_k - \mathbf{r}^x(\tau_k)\|)$  for all feasible  $\mathbf{x}_k, \mathbf{u}_k$ , and  $p_r(\mathbf{x}_N, \tau_N) \leq \alpha_2(\|\mathbf{x}_N - \mathbf{r}^x(\tau_N)\|)$ , where  $\alpha_1$  and  $\alpha_2$  are  $\mathcal{K}_\infty$ -functions.*

**Assumption 2** (Reference feasibility). *Under its natural time dynamics, the reference is feasible for the system dynamics, i.e.,  $\mathbf{r}^x(t + t_s) = f(\mathbf{r}^x(t), \mathbf{r}^u(t))$ , and:*

- a) *the reference satisfies the known constraints (4e), i.e.,  $h_n(\mathbf{r}^x(t_n), \mathbf{r}^u(t_n)) \leq 0$ , for all  $n \in \mathbb{I}_0^\infty$ ;*
- b) *the reference satisfies the unknown constraints (4f), i.e.,  $g_{n|k}(\mathbf{r}^x(t_n), \mathbf{r}^u(t_n)) \leq 0$ , for all  $n, k \in \mathbb{I}_0^\infty$ .*

Assumption 2b is a strong assumption since it assumes that the reference is feasible for constraints for all future times, i.e., at time  $k$  the constraint  $g_{n|k+1}$  is also assumed to be satisfied. Therefore, Assumption 2a serves as a relaxed version which is more realistic and will be used later on to replace Assumption 2b.

**Assumption 3** (Stabilizing Terminal Conditions). *There exists a parametric stabilizing terminal set  $\mathcal{X}_r^f(t)$  and a terminal control law  $\kappa_r^f(\mathbf{x}, t)$  yielding:*

$$\mathbf{x}_+^r = f(\mathbf{x}, \kappa_r^f(\mathbf{x}, t)), \quad t_+ = t + t_s,$$

such that  $p_r(\mathbf{x}_+^r, t_+) - p_r(\mathbf{x}, t) \leq -q_r(\mathbf{x}, \kappa_r^f(\mathbf{x}, t), t)$ , and

- a)  $\mathbf{x} \in \mathcal{X}_r^f(t) \Rightarrow \mathbf{x}_+^r \in \mathcal{X}_r^f(t_+)$ , and  $h_n(\mathbf{x}, \kappa_r^f(\mathbf{x}, t)) \leq 0$ , for all  $n, k \in \mathbb{I}_0^\infty$ ;
- b)  $\mathbf{x} \in \mathcal{X}_r^f(t) \Rightarrow g_{n|k}(\mathbf{x}, \kappa_r^f(\mathbf{x}, t)) \leq 0$ , for all  $n, k \in \mathbb{I}_0^\infty$ .

Similarly to Assumption 2b, Assumption 3b is also difficult to verify due to the unknown constraints. Hence, the relaxed version Assumption 3a, which is standard in MPC settings, will be used later on to replace Assumption 3b. Finally, we introduce the following assumption, imposing some structure on  $g$  that is needed in order to ensure that the feasibility of

a solution does not become jeopardized between consecutive time instances.

**Assumption 4** (Unknown constraint dynamics). *The a-priori unknown constraint functions satisfy  $g_{n|k+1}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k})$ , for all  $n \geq k$ .*

This assumption essentially requires the availability of a consistent characterization of the a priori unknown constraints, as we will further detail in Section IV-A. Note that in nominal MPC the constraints  $g$  are typically assumed to be known, in which case  $g_{n|k+1}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) = g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k})$ , for all  $n \geq k$ , which is a stronger requirement than Assumption 4. In case of robust MPC, instead, the uncertainty is handled in such a way that  $g_{n|k+1}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k})$ , holds for all  $n \geq k$ . The main difference between our setting and the standard robust MPC setting is that the uncertainty cannot be controlled.

We are now ready to prove asymptotic stability for the MPFTC framework.

**Theorem 1** (Nominal Asymptotic Stability). *Suppose that Assumptions 1, 2, 3, and 4 hold, and that the initial state  $(\mathbf{x}_k, \tau_k)$  at time  $k$  belongs to the feasible set of Problem (4). Then the system (1)-(3) in closed loop with the solution of (4) applied in receding horizon is an asymptotically stable system.*

*Proof.* The first part of the proof follows standard arguments used to prove stability for MPC. By assumption, there exists an optimal control input sequence  $\mathbf{U}_k^* = \{\mathbf{u}_{k|k}^*, \dots, \mathbf{u}_{k+N-1|k}^*\}$ ,  $\mathbf{V}_k^* = \{v_{k|k}^*, \dots, v_{k+N-1|k}^*\}$  and corresponding state trajectory  $\mathbf{X}_k^* = \{\mathbf{x}_{k|k}^*, \dots, \mathbf{x}_{k+N|k}^*\}$ ,  $\mathbf{T}_k = \{\tau_{k|k}^*, \dots, \tau_{k+N|k}^*\}$  at the initial time that gives the optimal value function

$$V(\mathbf{x}_k, \tau_k) = \sum_{n=k}^{k+N-1} q_{\mathbf{r}}(\mathbf{x}_{n|k}^*, \mathbf{u}_{n|k}^*, \tau_{n|k}^*) + wv_{n|k}^{*2} + p_{\mathbf{r}}(\mathbf{x}_{k+N|k}^*, \tau_{k+N|k}^*).$$

Applying the first control inputs  $\mathbf{u}_{k|k}^*$  and  $v_{k|k}^*$ , the system and auxiliary states evolve to  $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_{k|k}^*)$  and  $\tau_{k+1} = \tau_k + t_s + v_{k|k}^*$ , respectively. The suboptimal sequences  $\mathbf{U}_{k+1} = \{\mathbf{u}_{k+1|k}^*, \mathbf{u}_{k+2|k}^*, \dots, \kappa_{\mathbf{r}}^f(\mathbf{x}_{k+N|k}^*, \tau_{k+N|k}^*)\}$ , and  $\mathbf{V}_{k+1} = \{v_{k+1|k}^*, v_{k+2|k}^*, \dots, v_{k+N|k}^*\}$ , where  $\mathbf{x}_{k+N+1|k}^* = f(\mathbf{x}_{k+N|k}^*, \kappa_{\mathbf{r}}^f(\mathbf{x}_{k+N|k}^*, \tau_{k+N|k}^*))$ ,  $\tau_{k+N+1|k}^* = \tau_{k+N|k}^* + t_s + v_{k+N|k}^*$ , and  $v_{k+N|k}^* = 0$ , are still feasible under Assumptions 3 and 4, and yield the associated cost

$$\begin{aligned} \tilde{V}(\mathbf{x}_{k+1}, \tau_{k+1}) &= \sum_{n=k+1}^{k+N} q_{\mathbf{r}}(\mathbf{x}_{n|k}^*, \mathbf{u}_{n|k}^*, \tau_{n|k}^*) + wv_{n|k}^{*2} \\ &\quad + p_{\mathbf{r}}(\mathbf{x}_{k+N+1|k}^*, \tau_{k+N+1|k}^*) \\ &= V(\mathbf{x}_k, \tau_k) - q_{\mathbf{r}}(\mathbf{x}_{k|k}^*, \mathbf{u}_{k|k}^*, \tau_{k|k}^*) - wv_{k|k}^{*2} \\ &\quad + p_{\mathbf{r}}(\mathbf{x}_{k+N+1|k}^*, \tau_{k+N+1|k}^*) - p_{\mathbf{r}}(\mathbf{x}_{k+N|k}^*, \tau_{k+N|k}^*) \\ &\quad + q_{\mathbf{r}}(\mathbf{x}_{k+N|k}^*, \kappa_{\mathbf{r}}^f(\mathbf{x}_{k+N|k}^*, \tau_{k+N|k}^*), \tau_{k+N|k}^*). \end{aligned}$$

Using Assumption 3 and optimality, the optimal value function is shown to decrease between consecutive time instances

$$V(\mathbf{x}_{k+1}, \tau_{k+1}) \leq V(\mathbf{x}_k, \tau_k) - q_{\mathbf{r}}(\mathbf{x}_{k|k}^*, \mathbf{u}_{k|k}^*, \tau_{k|k}^*). \quad (7)$$

Assumption 1 entails the lower bound

$$V(\mathbf{x}_k, \tau_k) \geq q_{\mathbf{r}}(\mathbf{x}_{k|k}^*, \mathbf{u}_{k|k}^*, \tau_{k|k}^*) \geq \alpha_1(\|\mathbf{x}_{k|k}^* - \mathbf{r}^{\mathbf{x}}(\tau_{k|k}^*)\|), \quad (8)$$

and can be used to prove an upper bound [2, Proposition 2.17]

$$V(\mathbf{x}_k, \tau_k) \leq p_{\mathbf{r}}(\mathbf{x}_{k|k}^*, \tau_{k|k}^*) \leq \alpha_2(\|\mathbf{x}_{k|k}^* - \mathbf{r}^{\mathbf{x}}(\tau_{k|k}^*)\|). \quad (9)$$

Therefore, the value function is a Lyapunov function and closed-loop stability follows.

So far, we have proven that the states track a reference  $\mathbf{r}^{\mathbf{x}}(\tau_k)$  for some  $\tau_k$ . Hence, it has to be shown that  $v_{k|k} = 0$  asymptotically as well, which is the second and non-standard part of the proof. We observe that (a)  $\mathbf{r}^{\mathbf{x}}(\tau_{k+1}) = f(\mathbf{r}^{\mathbf{x}}(\tau_k), \mathbf{r}^{\mathbf{u}}(\tau_k))$  for  $\tau_{k+1} = \tau_k + t_s$  holds from Assumption 2, and (b)  $q_{\mathbf{r}}(\mathbf{x}_k, \mathbf{u}_k, \tau_k) = 0$  and  $p_{\mathbf{r}}(\mathbf{x}_{k+N}, \tau_{k+N}) = 0 \Rightarrow v_k = 0$  by optimality. Consequently,  $\lim_{k \rightarrow \infty} v_k = 0$  and  $\lim_{k \rightarrow \infty} \tau_{k+1} - \tau_k = t_s$ .

The implication (b) can be proven by noting that, in case  $\mathbf{x}_{n|k}^* = \mathbf{r}^{\mathbf{x}}(\tau_{n|k}^*)$ ,  $\mathbf{u}_{n|k}^* = \mathbf{r}^{\mathbf{u}}(\tau_{n|k}^*)$ , then  $V(\mathbf{x}_k, \tau_k) = \sum_{n=k}^{k+N-1} wv_{n|k}^{*2}$  and, consequently,

$$\begin{aligned} V(\mathbf{x}_{k+1}, \tau_{k+1}) &\leq \tilde{V}(\mathbf{x}_{k+1}, \tau_{k+1}) = \sum_{n=k+1}^{k+N} wv_{n|k}^{*2} \\ &\leq V(\mathbf{x}_k, \tau_k) - wv_{k|k}^{*2} \leq V(\mathbf{x}_k, \tau_k), \end{aligned}$$

where we used  $v_{k+N|k} = 0$ .  $\square$

**Remark 2.** *Assumption 2 can be relaxed to only require the reference to be feasible for an unspecified  $\tau_0 = t_0$ . In this case, the system will be stabilized to the reference with a time shift which is an integer multiple of the sampling time  $t_s$ . If, instead, feasibility holds for all initial times, then the time shift can be any real number. As opposed to MPFTC, in standard MPC the time shift is 0 by construction.*

**Remark 3.** *Note that the initial constraint  $\tau_{k|k} = \tau_k$  is not necessary, and Theorem 1 holds also in case the initial auxiliary state  $\tau_{k|k}$  is free to be selected by the optimizer.*

Theorem 1 proves that the proposed MPFTC formulation asymptotically stabilizes towards a reference  $\mathbf{r}(\tau)$ , while relying on assumptions which are standard in the MPC literature. However, Assumptions 2b, 3b, and 4 are difficult to enforce in practice, since they require feasibility with respect to constraints which are unknown. The next section therefore investigates how this difficulty can be tackled by replacing Assumptions 2b, and 3b with more realistic ones, and how Assumption 4 can be verified.

#### IV. SAFETY-ENFORCING MPC

The aim of this section is to tackle the issues posed by the presence of the a-priori unknown constraints (4f). While we cast the problem in the framework of MPFTC, we stress that the developments proposed to enforce safety are independent of the specific tracking scenario, i.e., flexible trajectory, path, setpoint, etc., and can also be deployed in the context of MPFC proposed in [14], [17].

We propose an approach that uses ideas typically found in robust MPC. In standard robust MPC settings the uncertainty

typically acts on the system that is being controlled and one can therefore use feedback to contain the uncertainty in a bounded set. However, in our setting the uncertainty is external to the system and not controllable. Since the uncertainty can grow unbounded, we will complement the usual worst-case approach with a suitably-defined safe set.

In Section IV-A we will discuss how the unknown constraints  $g$  can be constructed to ensure that Assumption 4 holds. Then, in Section IV-B we will introduce relaxed terminal conditions such that also Assumption 3b can be dropped. Finally, in Section IV-C, we conclude the main results of the paper with a theorem.

### A. Predictive Collision Avoidance

In this section we discuss how the a-priori unknown constraints  $g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k})$  can be constructed by relying on ideas typically used in robust MPC. Since full knowledge of these constraints is not available a-priori, one needs to assume that the uncertainty they model must be bounded, such that the problem of guaranteeing safety can be cast in a worst-case scenario planning. This is typically done in robust MPC where, however, the uncertainty is present in the system and not in the constraints. We will discuss this aspect further in the remainder of this section.

In order to model  $g_{n|k}$  as a constraint on the worst-case scenario, we introduce function  $\gamma(\mathbf{x}, \mathbf{u}, \mathbf{w}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \rightarrow \mathbb{R}^{n_g}$  and the uncertain variable  $\mathbf{w}_{n|k} \in \mathcal{W}_{n|k} \subseteq \mathbb{R}^{n_w}$  with bounded support  $\mathcal{W}_{n|k}$ , summarizing all uncertainty related to the a-priori unknown constraints. Then we define

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) := \max_{\mathbf{w}_{n|k} \in \mathcal{W}_{n|k}} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n|k}). \quad (10)$$

This formulation implies robust constraint satisfaction, i.e.,

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0 \Leftrightarrow \begin{cases} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n|k}) \leq 0, \\ \forall \mathbf{w}_{n|k} \in \mathcal{W}_{n|k}. \end{cases}$$

In a general setting,  $\mathbf{w}_{n|k}$  is the state of the dynamical system

$$\mathbf{w}_{n+1|k} = \omega(\mathbf{w}_{n|k}, \boldsymbol{\xi}_{n|k}, \mathbf{x}_{n|k}, \mathbf{u}_{n|k}), \quad (11)$$

with associated control variable  $\boldsymbol{\xi}_{n|k} \in \Xi \subseteq \mathbb{R}^{n_\xi}$ , acting as a source of (bounded) noise. The function  $\omega$  describes the dynamics, and the explicit dependence on  $\mathbf{x}_{n|k}, \mathbf{u}_{n|k}$  models possible interactions between the uncertainty and system (1). Note that when formulating  $g$  in accordance to (10) and (11), Problem (4) can be seen as a variant of robust MPC.

**Example 1.** In the simplest case,  $\mathbf{w}_{n|k}$  can model the sensor noise. Then, (11) reads as  $\mathbf{w}_{n+1|k} = \boldsymbol{\xi}_n$  such that  $\mathcal{W}_{n|k} \equiv \Xi$  and (10) reads as

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) = \max_{\mathbf{w}_{n|k} \in \mathcal{W}_{n|k}} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) + \mathbf{w}_{n|k},$$

i.e., the constraint has additive process noise and no dynamics are involved.

The case of process noise can be formulated as

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) = \max_{\mathbf{w}_{n|k} \in \mathcal{W}_{n|k}} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) + \mathbf{w}_{n|k}, \\ \mathbf{w}_{n+1|k} = \omega(\mathbf{w}_{n|k}, \boldsymbol{\xi}_{n|k}),$$

where there is no interaction between the uncertainty and the controlled system. This is the case in many robust MPC formulations, see, e.g., [24] and references therein.

**Remark 4.** The possibility of interaction between the system and the uncertainty dynamics is introduced in order to cover multi-agent settings in which the behavior of each agent can influence the behavior of other agents, e.g., a pedestrian changing his/her trajectory because of a vehicle not yielding.

Having introduced a model of the uncertainty dynamics, it becomes natural to rely on reachability analysis in order to predict the future evolution of the uncertainty sets, which are then defined as outer-approximations

$$\mathcal{W}_{n+1|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) := \supseteq \{ \omega(\mathbf{w}_{n|k}, \boldsymbol{\xi}_n, \mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \mid \mathbf{w}_{n|k} \in \mathcal{W}_{n|k}, \forall \boldsymbol{\xi}_n \in \Xi \}, \quad (12)$$

for some initial  $\mathcal{W}_{k|k} = \mathbf{w}_{k|k}$ .

In order to provide further explanation about the nature of the uncertainty sets (12), we provide the following example.

**Example 2.** Consider a non-cooperative, non-connected, multi-agent setting in which the behavior of the other agents is uncertain. In this case, we distinguish two types of agents: (a) the ones which are detectable by the sensors, and (b) those that are either beyond sensor range or hidden by other obstacles. For type (a), we require the model to be not underestimating the set of future states that can be reached by the other agents. For type (b), the uncertainty model must predict the possibility that an agent could appear at any moment either at the boundary of the sensor range or from behind an obstacle.

We can now state the following result.

**Lemma 1.** Suppose that  $g_{n|k}$  is defined according to (10) with  $\mathcal{W}_{n|k}$  satisfying (12). Then, Assumption 4 holds.

*Proof.* The properties of reachable sets imply  $\mathcal{W}_{n|k+1} \subseteq \mathcal{W}_{n|k}$ ,  $n \geq k+1$ . Therefore

$$\max_{\mathbf{w}_{n+1|k} \in \mathcal{W}_{n+1|k}} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n+1|k}) \leq \max_{\mathbf{w}_{n|k} \in \mathcal{W}_{n|k}} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n|k}),$$

since the two optimization problems have the same cost function, and the domain of the first one is not larger than the domain of the second one. Then,

$$g_{n|k+1}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}). \quad \square$$

Note that, in a robust MPC framework, this lemma amounts to assuming that the uncertainty cannot increase as additional information becomes available. Furthermore, a direct consequence of Assumption 4 is  $g_{n|k}(\mathbf{r}^x(t_n), \mathbf{r}^u(t_n)) \leq 0 \implies g_{n|k+1}(\mathbf{r}^x(t_n), \mathbf{r}^u(t_n)) \leq 0$ . However, this does not entail Assumption 2b, which requires feasibility of the reference for all  $n, k \in \mathbb{I}_0^\infty$ . In many cases of interest, the reference does become infeasible at certain future times since the uncertainty can grow indefinitely large over time. This makes Assumption 2b false for a predefined trajectory. Some approaches

tackle this issue by re-planning a feasible trajectory whenever some infeasibility is encountered [12]. However, only few such approaches can guarantee that a feasible trajectory always exists. Since we do not rely on any re-planning of the reference trajectory, our approach formalizes ideas similar to those of [19], [20], and always accounts for the worst-case scenario in order to guarantee that feasibility, hence safety, is never jeopardized even in case Assumption 2b does not hold. We ought to stress here that our approach can be deployed in combination with re-planning strategies in order to provide such guarantees.

**Remark 5.** *In the context of autonomous driving, the constraints  $g_{n|k}$  could enforce avoiding collisions with obstacles (e.g., other road users) detected by the sensors, whose behavior can just be predicted, to some extent. Hence, Assumption 4 amounts to assuming that the uncertainty on, e.g., position and velocity of the detected objects at a specific time instance cannot increase as additional information becomes available. We note however that limited sensor range makes it impossible to detect obstacles which are too far away. To ensure satisfaction of Assumption 4 one can adopt a worst-case approach which ensures that the predicted trajectory  $\mathbf{x}_{n|k}$  may never leave the sensor range, and by also assuming that new obstacles appear at the boundary of the sensor range at all times.*

Assumption 4 may be strong in some cases, however, this assumption is strictly necessary in order to be able to provide strict safety guarantees. In other words, it becomes impossible to ensure that constraint  $g$  will never be violated if Assumption 4 does not hold. This is a fundamental limitation, which is common to any robust approach and which states that safety only can be guaranteed for phenomena that are known sufficiently well.

## B. Terminal Conditions

Assumption 3b poses difficulties, since the terminal controller  $\kappa_r^f(\mathbf{x}, t)$  must satisfy  $g_{n|k}(\mathbf{x}_{n|k}, \kappa_r^f(\mathbf{x}_{n|k}, \tau_{n|k})) \leq 0$ , for all  $n, k \in \mathbb{I}_0^\infty$ , i.e., over an infinite horizon. However, this is in general impossible, unless additional assumptions are introduced, since the constraint uncertainty typically grows unbounded with time. Furthermore, the terminal controller  $\kappa_r^f(\mathbf{x}, t)$  is assumed to ensure stability with respect to the reference, i.e.,  $p_r(\mathbf{x}_+^c, t_+) - p_r(\mathbf{x}, t) \leq -q_r(\mathbf{x}, \kappa_r^f(\mathbf{x}, t), t)$ . This may not always be possible, since by dropping Assumption 2b, the reference might become infeasible with respect to some of the a-priori unknown constraints.

To cope with the recursive feasibility issue, we will assume the existence of a *safe set* in which a-priori unknown constraints are guaranteed to be satisfied in all circumstances. This will allow us to rely on standard approaches in MPC [3], [25], [26] which are based on the existence of a robust invariant set.

**Assumption 5.** *There exists a robust invariant set denoted  $\mathcal{X}_{\text{safe}}(\tau_{n|k}) \subseteq \mathbb{R}^{n_x}$  such that for all  $\mathbf{x}_{n|k} \in \mathcal{X}_{\text{safe}}(\tau_{n|k})$  there exists a safe control set  $\mathcal{U}_{\text{safe}}(\mathbf{x}_{n|k}, \tau_{n|k}) \subseteq \mathbb{R}^{n_u+1}$  entailing that  $f(\mathbf{x}_{n|k}, \mathbf{u}_{\text{safe}}) \in \mathcal{X}_{\text{safe}}(\tau_{n|k} + t_s + v_{\text{safe}})$ , and  $h_n(\mathbf{x}_{n|k}, \mathbf{u}_{\text{safe}}) \leq 0$ , for all  $(\mathbf{u}_{\text{safe}}, v_{\text{safe}}) \in \mathcal{U}_{\text{safe}}(\mathbf{x}_{n|k}, \tau_{n|k})$  and for all  $n \geq k$ . Moreover, for all  $\mathbf{x}_{n|k} \in \mathcal{X}_{\text{safe}}(\tau_{n|k})$  the*

*a-priori unknown constraints can never be positive, i.e., by construction  $g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{\text{safe}}) \leq 0$  for all  $\mathbf{x}_{n|k} \in \mathcal{X}_{\text{safe}}(\tau_{n|k})$  and  $(\mathbf{u}_{\text{safe}}, v_{\text{safe}}) \in \mathcal{U}_{\text{safe}}(\mathbf{x}_{n|k}, \tau_{n|k})$ .*

While this assumption might seem strong, it only postulates the existence of *known safe configurations* for system (1). However, if no such configurations exist, then the controller based on Problem (4) is intrinsically unsafe. On the other hand, if such configurations do exist for (1), then the safe set  $\mathcal{X}_{\text{safe}}$  is non-empty and invariant. Note that the safe set  $\mathcal{X}_{\text{safe}}$  depends on system (1), the problem setting, and must be known a-priori.

**Example 3.** *Many practical settings where safety is emphasized consider a system to be safe at steady-state, in which case the safety set  $\mathcal{X}_{\text{safe}}$  can be formulated as*

$$\mathcal{X}_{\text{safe}}(t_k) := \{ \mathbf{x} \mid \mathbf{x} = f(\mathbf{x}, \mathbf{u}), \quad h_k(\mathbf{x}, \mathbf{u}) \leq 0, m_k(\mathbf{x}, \mathbf{u}) \leq 0 \}, \quad (13)$$

where function  $m_k$  defines additional constraints which might be needed in the set definition. Notable examples include, e.g., the following: (a) a robotic manipulator operating in a mixed human-robot environment is considered safe if it does not move; (b) a vehicle parked in a safe configuration, e.g., a parking lot, emergency lane or any other safe environment that can be modeled by  $m_k$ , is not responsible for collisions with other road users; (c) an electric circuit which is switched off is generally safe; (d) a ship docked in a port can be considered safe. We stress that the examples together with set (13) presented above are simplified in order to convey the safety message, but in practice the safe set will need to be carefully designed for each specific use case.

In general, most processes controlled either by humans or by automatic controllers do have emergency procedures which are triggered whenever safety is jeopardized. Assumption 5 is meant to cover all these situations. Furthermore, we must stress that this assumption also entails that  $g_{n|k}$  directly depends on the safe set, since it is assumed that  $g_{n|k}$  may never be active when  $\mathbf{x}_{n|k} \in \mathcal{X}_{\text{safe}}(\tau_{n|k})$ . Hence, this assumption entails that when using the uncertainty model presented in Section IV-A, function  $\gamma$  used in (10) is not simply the output of the uncertainty model, but also includes the information that  $\mathbf{x}_{n|k} \in \mathcal{X}_{\text{safe}}(\tau_{n|k})$  implies  $\gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{\text{safe}}, \mathbf{w}_{n|k}) = 0$ . This condition is typically not physics-driven, but stems from Assumption 5: we provide a clarifying example next.

**Example 4.** *Consider Example 3, case (b), where functions  $g$  and  $\gamma$  model the set of positions that a pedestrian can reach. Whenever the vehicle is parked in a safe configuration, function  $\gamma$  is not simply yielding all positions that a pedestrian can reach, but it must include the information that the pedestrian cannot reach the position of the vehicle, even though it would be physically possible to do so.*

**Remark 6.** *Note that, while it is often reasonable to construct the set based on (13), the safe set does not necessarily need to be forcing a steady-state, other formulations, e.g., including safe periodic trajectories may also be considered. Therefore,*

we prefer formulating the assumption in a generic way in order to cover as many cases of interest as possible.

The introduction of Assumption 5, allows us to drop Assumption 3b. We build our approach based on standard strategies in MPC [3], [25], [26], i.e., we rely on stabilizing terminal control laws  $\kappa_{\mathbf{r}}^s(\mathbf{x}, t)$  and sets  $\mathcal{X}_{\mathbf{r}}^s(t)$  satisfying Assumption 3a only, such that the a-priori unknown constraints  $g$  can be neglected in this design phase. In order to obtain recursive feasibility also with respect to a-priori unknown constraints, we rely on the safe set  $\mathcal{X}_{\text{safe}}$  to introduce the following terminal set

$$\mathcal{X}_{\mathbf{r}}^f(\tau_{k+N|k}) := \{\mathbf{x}_{k+N|k} \mid \exists \mathbf{u}_{n|k}, v_{n|k}, \quad (14a)$$

$$\tau_{n+1|k} = \tau_{n|k} + t_s + v_{n|k}, \quad (14b)$$

$$\mathbf{x}_{n+1|k} = f(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}), \quad (14c)$$

$$h_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, \quad (14d)$$

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, \quad (14e)$$

$$\mathbf{x}_{n|k} \in \mathcal{X}_{\mathbf{r}}^s(\tau_{n|k}), \quad (14f)$$

$$\mathbf{x}_{k+M|k} \in \mathcal{X}_{\text{safe}}(\tau_{k+M|k}) \subseteq \mathcal{X}_{\mathbf{r}}^s(\tau_{k+M|k}), \quad (14g)$$

$$(14a) - (14f), \quad \forall n \in \mathbb{I}_{k+N}^{k+M-1}, \quad (14h)$$

where  $M \geq N$  is a degree of freedom. Note that the construction of (14) implies that  $\mathcal{X}_{\text{safe}}(\tau) \subseteq \mathcal{X}_{\mathbf{r}}^s(\tau)$ ,  $\forall \tau \geq 0$ . If (14) is a non-empty set we are guaranteed that for all  $\mathbf{x} \in \mathcal{X}_{\mathbf{r}}^f(\tau)$  a terminal control law exists, which steers the states to the safe set.

In order to provide a practical approach to design the terminal control law, we propose to first design a control law  $\kappa_{\mathbf{r}}^s$  as one would do in standard MPC formulations, i.e., by ignoring a-priori unknown constraints  $g$  and by forcing the time in the reference to evolve according to its true dynamics. We can then define the terminal control law ( $\kappa_{\mathbf{r}}^f(\mathbf{x}_{k+N|k}, \tau_{k+N|k}), \nu_{\mathbf{r}}^f(\mathbf{x}_{k+N|k}, \tau_{k+N|k})$ ) by using  $\kappa_{\mathbf{r}}^s$ , as the solution of

$$\min_{\mathbf{u}, \nu} \|\mathbf{u} - \kappa_{\mathbf{r}}^s(\mathbf{x}_{k+N|k}, \tau_{k+N|k})\|_2 + \nu^2 \quad (15a)$$

$$\text{s.t. } f(\mathbf{x}_{k+N|k}, \mathbf{u}) \in \mathcal{X}_{\mathbf{r}}^f(\tau_{k+N|k} + t_s + \nu), \quad (15b)$$

$$h_{k+N}(\mathbf{x}_{k+N|k}, \mathbf{u}) \leq 0, \quad (15c)$$

$$g_{k+N}(\mathbf{x}_{k+N|k}, \mathbf{u}) \leq 0. \quad (15d)$$

The idea behind the terminal set (14) is to ensure safety by forcing the system to be able to reach a safe set  $\mathcal{X}_{\text{safe}}$  in a finite amount of time  $M - N \geq 0$ , while always remaining inside a stabilizing set  $\mathcal{X}_{\mathbf{r}}^s(t)$  around the reference. An illustrative example is shown in Section V-C. Note that  $M$  is a parameter which can be used to tune the stabilizing terminal safe set and, consequently, the NMPC scheme (4). If  $M = N$ , then the terminal set coincides with the safe set, possibly limiting the capabilities of the terminal control law, i.e.,  $\kappa_{\mathbf{r}}^f(\mathbf{x}, \tau) \neq \kappa_{\mathbf{r}}^s(\mathbf{x}, \tau)$  and  $\nu_{\mathbf{r}}^f(\mathbf{x}, \tau) \neq 0$ . On the other hand if  $M \gg N$ , the computational complexity of  $\mathcal{X}_{\mathbf{r}}^f$  can become excessive.

We can now prove recursive feasibility for the terminal control law.

**Lemma 2.** *Suppose that Assumptions 1, 2a, 3a, 4, and 5 hold, and that the terminal set  $\mathcal{X}_{\mathbf{r}}^f$  given by (14) is nonempty.*

*Then, for an initial state  $\mathbf{x}_{k+N|k} \in \mathcal{X}_{\mathbf{r}}^f(\tau_{k+N|k})$ , the terminal controller (15) is recursively feasible.*

*Proof.* Consider the state  $(\mathbf{x}_{k+N|k}, \tau_{k+N|k})$  at time  $k$  such that  $\mathbf{x}_{k+N|k} \in \mathcal{X}_{\mathbf{r}}^f(\tau_{k+N|k})$ . By construction, there exist control sequences  $\mathbf{U}_k = \{\mathbf{u}_{k+N|k}, \dots, \mathbf{u}_{k+M-1|k}\}$  and  $\mathbf{V}_k = \{v_{k+M|k}, \dots, v_{k+M-1|k}\}$  that generate the corresponding state trajectories  $\mathbf{X}_k = \{\mathbf{x}_{k+N|k}, \dots, \mathbf{x}_{k+M|k}\}$  and  $\mathbf{T}_k = \{\tau_{k+N|k}, \dots, \tau_{k+M|k}\}$  satisfying (14). Hence, we have that  $h_{k+N}(\mathbf{x}_{k+N|k}, \mathbf{u}_{k+N|k}) \leq 0$  and  $g_{k+N|k}(\mathbf{x}_{k+N|k}, \mathbf{u}_{k+N|k}) \leq 0$ . By applying the first control inputs  $\mathbf{u}_{k+N|k}$  and  $v_{k+N|k}$ , we know that the following trajectories

$$\mathbf{X}_{k+1} = \{\mathbf{x}_{k+N+1|k}, \dots, \mathbf{x}_{k+M|k}, f(\mathbf{x}_{k+M|k}, \mathbf{u}_{\text{safe}})\}, \quad (16)$$

$$\mathbf{T}_{k+1} = \{\tau_{k+N+1|k}, \dots, \tau_{k+M|k}, \tau_{k+M|k} + t_s + v_{\text{safe}}\}, \quad (17)$$

$$\mathbf{U}_{k+1} = \{\mathbf{u}_{k+N+1|k}, \dots, \mathbf{u}_{k+M-1|k}, \mathbf{u}_{\text{safe}}\}, \quad (18)$$

$$\mathbf{V}_{k+1} = \{v_{k+N+1|k}, \dots, v_{k+M-1|k}, v_{\text{safe}}\}, \quad (19)$$

satisfy  $h_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0$  for  $n \in \mathbb{I}_{k+N+1}^{k+M-1}$  at time  $k+1$  by definition, while Assumption 4 ensures that

$$g_{n|k+1}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, \quad \forall n \in \mathbb{I}_{k+N}^{k+M-1}.$$

Furthermore, since  $\mathbf{x}_{k+M|k} \in \mathcal{X}_{\text{safe}}(\tau_{k+M|k})$ , Assumption 5 ensures  $h_{k+M}(\mathbf{x}_{k+M|k}, \mathbf{u}_{\text{safe}}) \leq 0$ ,  $g_{k+M|k+1}(\mathbf{x}_{k+M|k}, \mathbf{u}_{\text{safe}}) \leq 0$ , and that  $f(\mathbf{x}_{k+M|k}, \mathbf{u}_{\text{safe}}) \in \mathcal{X}_{\text{safe}}$ . Hence, we have shown that there exists control inputs  $\mathbf{u} := \mathbf{u}_{k+N|k}$  and  $v := v_{k+N|k}$  ensuring  $f(\mathbf{x}_{k+N|k}, \mathbf{u}) \in \mathcal{X}_{\mathbf{r}}^f(\tau_{k+N|k} + t_s + v)$  and that Problem (15) is feasible at time  $k$ . To prove feasibility for times  $k' \geq k+1$ , we can use the invariance condition of Assumption 5 and construct similar guesses.  $\square$

### C. MPC Recursive Feasibility in Uncertain Environments

Building on the results presented in Sections IV-A and IV-B, we can prove recursive feasibility for the proposed MPC scheme.

**Theorem 2** (Recursive Feasibility). *Suppose that Assumptions 1, 2a, 3a, 4, and 5 hold, and that Problem (4) is feasible for the initial state  $(\mathbf{x}_k, \tau_k)$ , with terminal set and terminal controllers given by (14) and (15), respectively. Then, system (1)-(3) in closed loop with the solution of (4) applied in receding horizon is safe (recursively feasible) at all times.*

*Proof.* Consider the following control trajectories  $\mathbf{U}_k^* = \{\mathbf{u}_{k|k}^*, \dots, \mathbf{u}_{k+N-1|k}^*\}$  and  $\mathbf{V}_k^* = \{v_{k|k}^*, \dots, v_{k+N-1|k}^*\}$ , with corresponding state trajectories  $\mathbf{X}_k^* = \{\mathbf{x}_{k|k}^*, \dots, \mathbf{x}_{k+N|k}^*\}$  and  $\mathbf{T}_k^* = \{\tau_{k|k}^*, \dots, \tau_{k+N|k}^*\}$ , to be the solution of Problem (4). Recursive feasibility follows from Assumption 4 and Lemma 2, which ensure that the control and state trajectories  $\mathbf{U}_k^*$ ,  $\mathbf{V}_k^*$ ,  $\mathbf{X}_k^*$ , and  $\mathbf{T}_k^*$ , and their prolongation to infinite time using  $\kappa_{\mathbf{r}}^f(\mathbf{x}_{k'|k'}, \tau_{k'|k'})$  and  $\nu_{\mathbf{r}}^f(\mathbf{x}_{k'|k'}, \tau_{k'|k'})$ , satisfy constraints  $g_{k'|k''}$  for all  $k' \geq k''$  and  $k'' \geq k$ . This, in turn, implies that the controller is safe in the sense of Definition 1.  $\square$

While we have proven recursive feasibility, the presence of obstacles makes it harder to discuss closed-loop stability. In principle, it seems possible to prove Input-To-State Stability (ISS), and we will investigate this possibility in future research. Here, we limit ourselves to the observation

that, if the a-priori unknown constraints become inactive, then the proposed formulation yields nominal asymptotic stability, which can be proven by standard arguments such as those used in Theorem 1.

## V. SIMULATIONS

In this section we propose three examples to illustrate the developed theory. The first example aims at discussing how the novel MPFTC framework works, and how it compares to the similar and competing MPFC framework using the example in [14]. The second example focuses the recursive feasibility aspect. Here a very simple double integrator is used in order to illustrate all important aspects, and to be able to visualize all features. Finally, the third example, which was also used in [17], shows how our safety enforcing formulation performs for a more involved setting.

In all the examples we will use stage and terminal cost in (5)-(6), i.e.,

$$\begin{aligned} \Delta \mathbf{x}_{n|k} &:= \mathbf{x}_{n|k} - \mathbf{r}^x(\tau_{n|k}), & \Delta \mathbf{u}_{n|k} &:= \mathbf{u}_{n|k} - \mathbf{r}^u(\tau_{n|k}), \\ q_{\mathbf{r}}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_{n|k}) &= \begin{bmatrix} \Delta \mathbf{x}_{n|k} \\ \Delta \mathbf{u}_{n|k} \end{bmatrix}^\top W \begin{bmatrix} \Delta \mathbf{x}_{n|k} \\ \Delta \mathbf{u}_{n|k} \end{bmatrix}, \\ p_{\mathbf{r}}(\mathbf{x}_{k+N|k}, \tau_{k+N|k}) &= \Delta \mathbf{x}_{k+N|k}^\top P \Delta \mathbf{x}_{k+N|k}. \end{aligned}$$

Furthermore, all simulations ran on a laptop computer (i7 2.8GHz, 16GB RAM) and were implemented in Matlab using the CasADi [27] software together with the IPOPT [28] solver. The simulation scripts have been made available on Github<sup>1</sup>.

### A. Practical Safe MPFTC Formulation

Since computing terminal set (14) explicitly is impractical for non-trivial cases, we extend the MPC prediction horizon from  $N$  to  $M$  and include the conditions defining (14) as constraints in the MPC problem. More formally, for practical applications, we propose to use the following problem formulation, which is equivalent to Problem (4)

$$\min_{\substack{\mathbf{x} \\ \mathbf{u} \\ \tau, v}} \sum_{n=k}^{k+N-1} q_{\mathbf{r}}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_{n|k}) + wv_{n|k}^2 \quad (20a)$$

$$\text{s.t. } \mathbf{x}_{k|k} = \mathbf{x}_k, \quad \tau_{k|k} = \tau_k, \quad (20b)$$

$$\mathbf{x}_{n+1|k} = f(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}), \quad n \in \mathbb{I}_k^{k+M-1}, \quad (20c)$$

$$\tau_{n+1|k} = \tau_{n|k} + t_s + v_{n|k}, \quad n \in \mathbb{I}_k^{k+M-1}, \quad (20d)$$

$$h_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, \quad n \in \mathbb{I}_k^{k+M-1}, \quad (20e)$$

$$g_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, \quad n \in \mathbb{I}_k^{k+M-1}, \quad (20f)$$

$$\mathbf{x}_{k+n|k} \in \mathcal{X}_{\mathbf{r}}^s(\tau_{k+n|k}), \quad n \in \mathbb{I}_{k+N}^{k+M-1}, \quad (20g)$$

$$\mathbf{x}_{k+M|k} \in \mathcal{X}_{\text{safe}}(\tau_{k+M|k}) \subseteq \mathcal{X}_{\mathbf{r}}^s(\tau_{k+M|k}). \quad (20h)$$

This scheme can be seen as a 2-stage problem, where the first stage ( $n \in \mathbb{I}_k^{k+N}$ ) defines the MPC problem and the second stage ( $n \in \mathbb{I}_{k+N+1}^{k+M}$ ) defines the terminal set implicitly. While this formulation solves the issue of precomputing  $\mathcal{X}_{\mathbf{r}}^f$  explicitly, it can still suffer from numerical difficulties since the

trajectory in the second stage ( $n \in \mathbb{I}_{k+N+1}^{k+M}$ ) is not penalized by any cost. While a thorough discussion on possible remedies is out of the scope of this paper, we observe the following: (a) for interior-point methods, the primal interpretation results in a cost penalizing deviations from the center of the feasible domain, such that the solution is unique and well-defined; (b) for any iterative solver, introducing a penalization on deviations from the previous iterate has a regularizing effect which alleviates the numerical difficulties; (c) tracking the reference also beyond  $n = N$  with a small penalty function introduces a perturbation on the optimal solution which can be analyzed in the context of ISS and is expected to result in a small tracking inaccuracy which could be acceptable in many practical situations.

### B. Autonomous Vehicle Reference Tracking

We evaluate MPFTC and we compare the results with the MPFC formulation of [14] using the kinematic model of a car

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} u_1 \cos \psi \\ u_1 \sin \psi \\ u_1 \tan u_2 \end{bmatrix}, \quad (21)$$

where  $x$  and  $y$  describe the position,  $\psi$  is the yaw angle,  $u_1$  is the speed, and  $u_2$  the steering angle. The system is subject to the constraints  $0 \leq u_1 \leq 6$ ,  $\|u_2\|_2 \leq 0.63$ . As reference, we use the path

$$p(\theta) = \left[ \rho(\theta)^\top, \arctan \left( \frac{\partial \rho_2}{\partial \theta} \right) \right]^\top, \quad (22)$$

with  $\rho(\theta) = [\theta, -6 \log(20/(5 + |\theta|)) \sin(0.35\theta)]^\top$  and parameter  $\theta \in [-30, 0]$ . Since MPFTC requires a trajectory to track, we design the a-priori path evolution of  $\theta$  to be

$$\dot{\theta}(t) = \frac{v_{\text{ref}}(t)}{\|\nabla_{\theta} \rho(\theta(t))\|_2}, \quad v_{\text{ref}}(t) = \begin{cases} 5 & t \leq 7 \\ \max(5 - at, 0) & t > 7 \end{cases}$$

with  $a = -5.38$ . This predefined path evolution implies that the reference velocity along the trajectory will be 5 m/s for  $t \leq 7$ , and monotonically decreasing for  $t > 7$  until reaching zero velocity. For MPFTC (4), with auxiliary state  $\tau$  and control  $v$  we formulate a feasible reference as

$$\mathbf{r}^x(t) = \left[ \rho(\theta(t))^\top, \arctan \left( \frac{\partial \rho_2}{\partial \theta} \right) \right]^\top, \quad (23)$$

$$\mathbf{r}^u(t) = \begin{bmatrix} \dot{\theta}(t) \sqrt{1 + \left( \frac{\partial \rho_1(\theta(t))}{\partial \theta} \right)^2} \\ \arctan \left( \frac{\partial^2 \rho_2}{\partial \theta^2} \left( 1 + \left( \frac{\partial \rho_2}{\partial \theta} \right)^2 \right)^{-\frac{3}{2}} \right) \end{bmatrix}, \quad (24)$$

where the control reference can directly be derived from (21). For the cost we use  $W = \text{blockdiag}(Q, R)$  with

$$Q = \text{diag}(1, 1, 1), \quad R = \text{diag}(1, 1), \quad w = 10.$$

Using the results from ([14], Corollary 1) it can be shown that the terminal cost given by  $P = Q$ , together with the terminal set  $\mathcal{X}_{\mathbf{r}}^f(t) = \{\mathbf{x} \mid \mathbf{x} = \mathbf{r}^x(t)\}$ , is a suitable, although conservative, choice to stabilize the system.

For MPFC, we use the tuning parameters  $Q_{\text{MPFC}} = 8 \cdot \text{diag}(10^4, 10^5, 10^5, 1/16)$ ,  $R_{\text{MPFC}} = \text{diag}(10, 10, 1)$ , terminal

<sup>1</sup>github.com/ivobatkovic/safe-trajectory-tracking

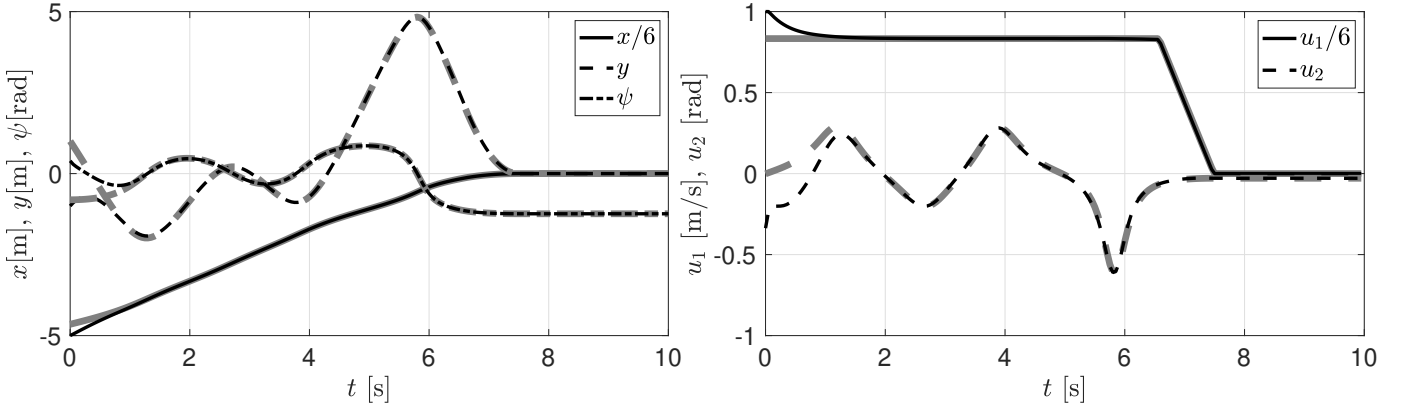


Fig. 1. Closed loop trajectories for the autonomous vehicle example with MPFTC with  $w = 10$ . Gray lines with matching line style denote the reference trajectory.

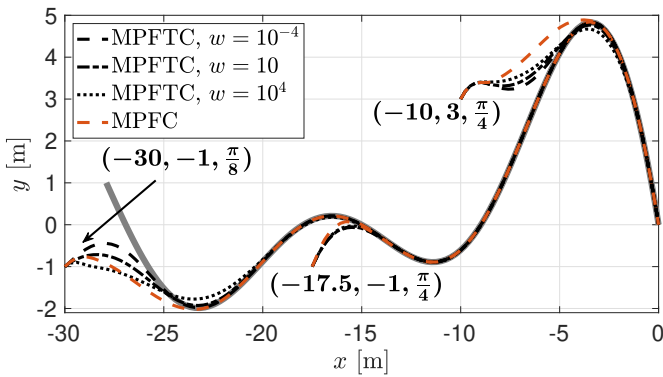


Fig. 2. Closed loop trajectories of the  $x$ - $y$  states for the autonomous vehicle example.

weight  $Q_{\text{MPFC},N} = 1/2 \cdot \text{diag}(0, 0, 0, 1740)$ , and same setup as in [14].

For the simulation, we use a prediction horizon of 1 s, and solve the OCP repeatedly with 20 direct multiple shooting intervals. The closed loop sampling time is  $t_s = 0.05$  s. The initial value for  $\tau_0$  is selected by projecting the initial position  $(x_0, y_0)$  on the trajectory, i.e.,

$$\tau_0 = \arg \min \|(x_0, y_0) - \rho(\theta(\tau))\|_2.$$

Figure 1 shows the closed-loop trajectories for the initial value  $\mathbf{x}_0 = [-30, -1, \pi/8]^\top$ ,  $\tau_0 = 0.573$  s for the MPFTC controller with  $w = 10$ . Starting with an offset from the reference trajectory, for all initial states the system is stabilized to the reference and converges to the point  $p(0)$ . Figure 2 shows closed-loop trajectories for different initial values for both MPFTC and MPFC. For MPFTC, we show how the closed-loop behavior depends on the auxiliary input cost  $w$ : higher penalties result in a more aggressive tracking than lower ones. Note that while MPFTC and MPFC result in different closed-loop trajectories, their behavior is similar. For the simulations with initial condition  $\mathbf{x}_0 = [-30, -1, \pi/8]^\top$ , we obtained the average runtime of  $\hat{t}_{\text{MPFTC}} = 0.049$  s and  $\hat{t}_{\text{MPFC}} = 0.044$  s for MPFTC and MPFC, respectively. We stress our implementation was far from being implemented in a computationally optimal way, such that a real-time implementation is expected to provide much faster run times.

### C. Double Integrator with A-Priori Unknown Obstacles

We consider double integrator dynamics to illustrate in the simplest fashion the effect of the safe set formulation in the presence of a-priori unknown obstacles. The state and control are given by  $\mathbf{x} = [p, \dot{p}]^\top$ ,  $\dot{p} \geq 0$  and  $\mathbf{u} = a \in [-1, 5]$  respectively, with reference  $\mathbf{r}^x(t) = [tv^r, v^r]^\top$ ,  $\mathbf{r}^u(t) = 0$ , and  $v^r = 4$  m/s. To model constraint  $g$  we introduce a static obstacle at position  $p_{n|k}^{\text{obs}}$ , such that

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) = p_{n|k} - p_{n|k}^{\text{obs}} \leq 0, \quad (25)$$

where

$$p_{n|k}^{\text{obs}} = \begin{cases} 20 & \text{for } k \in [0, 75], \\ \infty & \text{otherwise.} \end{cases} \quad (26)$$

For the cost we use  $W = \text{blockdiag}(Q, R)$  with

$$Q = \text{diag}(10, 10), \quad R = 1, \quad w = 1,$$

and the terminal cost matrix  $P$  is obtained from the LQR cost corresponding to  $Q_{\text{LQR}} = \text{diag}(1, 1)$ ,  $R_{\text{LQR}} = 10$ . The corresponding stabilizing set  $\mathcal{X}_r^s$  for the LQR controller is computed using the Multi-Parametric Toolbox (MPT) [29]. We consider (13) and (14) to construct the safe set and assume that a zero velocity around the reference trajectory is considered to be safe. We therefore define the safe set for this example as  $\mathcal{X}_{\text{safe}}(t) = \{\mathbf{x} | \dot{p} = 0\} \cap \mathcal{X}_r^s(t)$ . With  $\mathcal{X}_r^s$  and  $\mathcal{X}_{\text{safe}}$ , we can now express the terminal set  $\mathcal{X}_r^f(\tau_{k+N|k})$  given by (14).

For the following scenario we compare MPFTC with standard MPC, i.e., without reference adaptation. To highlight the benefit of a safety set, we set the sampling time to  $t_s = 0.2$  s and use control intervals  $N = 5$ ,  $M = 10$ . For MPFTC and MPC without the stabilizing safe set, we use the same sampling time but set  $N = 10$ . For the problems not including the safety formulation, we relax the constraint  $g_{n|k}$  with an exact penalty [30].

Figure 3 shows that MPC and MPFTC without the proposed safety formulation are not able to satisfy the constraint. Additionally, MPC is very aggressive, especially after the obstacle is removed: while the system was stopped by the obstacle the gap in the position reference kept increasing, resulting in a wind-up effect. On the contrary, safe MPFTC satisfies the constraint and does not have an aggressive behavior. Moreover,

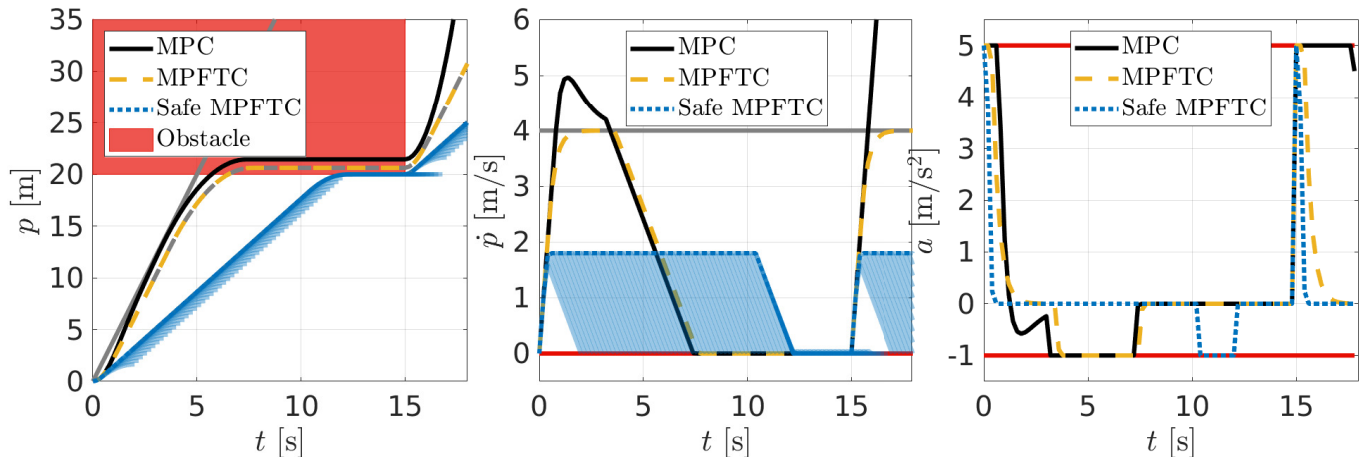


Fig. 3. Closed-loop trajectories for the double integrator example. The red box and lines denote constraints, while the gray lines denote the reference trajectory, with the line style matching the corresponding state trajectory. The opaque blue lines in the two left-most figures show the safe MPFTC open-loop trajectories at all times for the position and velocity respectively.

it never attains the reference velocity of 4 m/s since it would not be safe to do so. With a longer prediction horizon a velocity of 4 m/s would be safe and in that case also safe MPFTC would reach the reference. Note that between times  $t \in [12, 15]$ , the state  $\mathbf{x}_k = [p_k, \dot{p}_k]^\top$  is in fact in the safe set, since the velocity is zero, i.e.,  $\dot{p}_k = 0$ . Then, after 15 s, the constraint is lifted and the controller continues to track the reference same as before. Figure 4 shows the closed loop trajectories of  $\tau_k$  and  $v_k$ . We note that MPFTC and safe MPFTC halt the reference time evolution so that the reference trajectory does not move when the system is stopped by the obstacle. MPC on the other hand does not adjust the reference time  $\tau_k$ , which causes a more aggressive behavior compared to MPFTC and safe MPFTC. Finally, the blue opaque open loop state trajectories show how the safe set forces the velocity to be zero at the end of the horizon. For each simulation, we obtained the following average runtimes:  $\hat{t}_{\text{MPC}} = 0.028$  s,  $\hat{t}_{\text{MPFTC}} = 0.016$  s, and  $\hat{t}_{\text{MPFTC}}^{\text{Safe}} = 0.016$  s, for MPC, MPFTC and safe MPFTC, respectively. Also in this case we stress that these run times only provide an indication and a real-time implementation is expected to be much faster.

Since in general it is impractical or even impossible to compute the terminal set (14) explicitly, we formulated safe MPFTC implicitly using the formulation proposed in Problem (20). However, in this simple example, such a set can be computed explicitly using MPT [29].

Figure 5 shows the terminal set  $\mathcal{X}_r^f$  expressed both in the global frame  $(p, \dot{p})$  and in the reference error frame  $(p - v^r t, \dot{p} - v^r)$  for different values of  $\tau_{k+N|k}$ . When the reference is far from the obstacle, the set is defined by  $\mathcal{X}_s^r$  and the requirement to reach  $\mathcal{X}_{\text{safe}}$  in a finite number of time steps. However, as  $\tau_{k+N|k}$  approaches 5 s, the reference approaches the obstacle  $p^{\text{obs}}$  and the set contracts in order to satisfy constraint (25).

It is well known that in optimal control, constraints appearing in the far future have a negligible impact on the initial control. This fact is tacitly exploited together with the exact penalty constraint relaxation in order to avoid feasibility

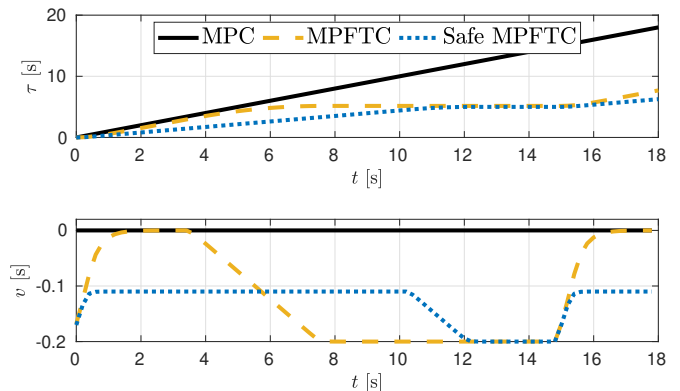


Fig. 4. Closed loop trajectories of the auxiliary state and input  $\tau$  and  $v$  for the double integrator example.

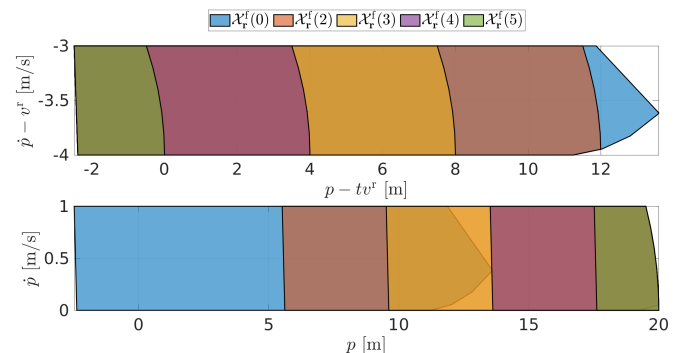


Fig. 5. Comparison of the terminal set  $\mathcal{X}_r^f(t)$  across different times. The top figure shows the terminal set expressed in the relative reference frame while the bottom figure shows the terminal set expressed in the global frame  $(p, \dot{p})$ .

issues. However, our formulation provides a rigorous approach and is particularly useful in cases in which the obstacles cannot be detected well in advance.

Finally, in this example the constraints are not time-varying for the sake of simplicity. We provide next a more involved example.

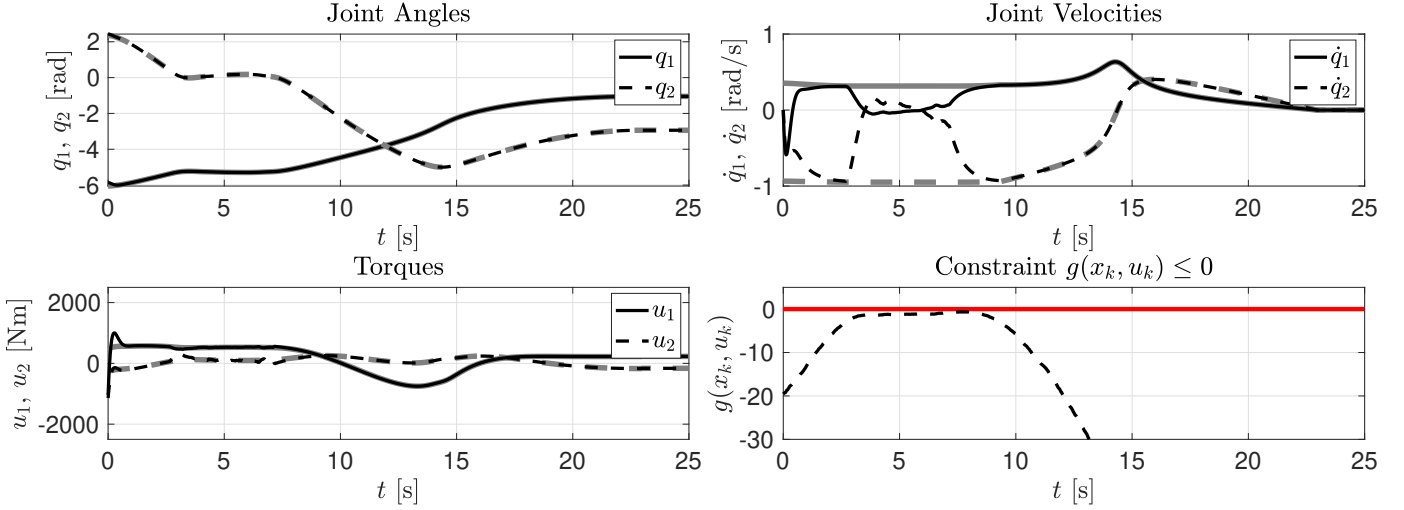


Fig. 6. Closed-loop trajectories for the robotic joint example. The gray lines denote the reference trajectory for each state and control, while the red lines denote constraint limits.

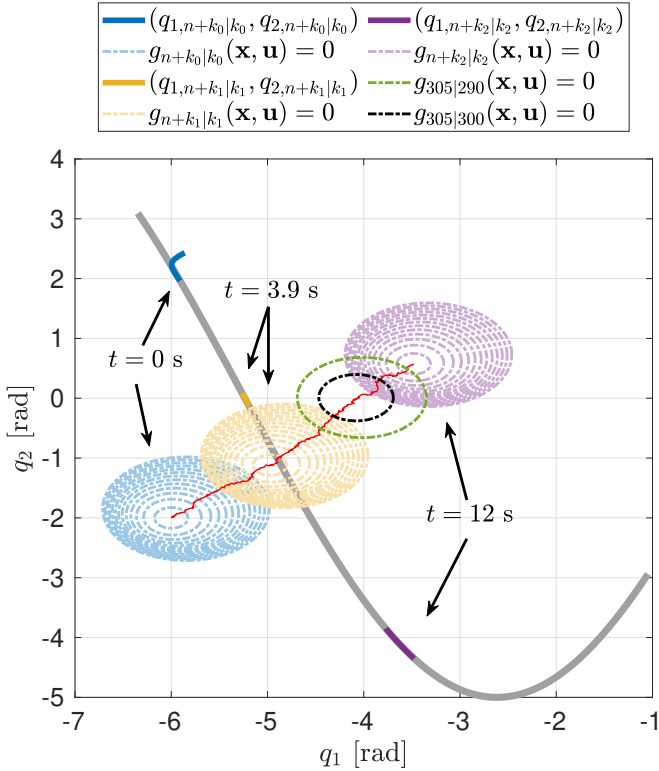


Fig. 7. The gray line denotes the reference trajectory, while the red line denotes the closed-loop evolution of the obstacle. The colored lines denote the open-loop state trajectories  $(q_{1,n|k}, q_{2,n|k})$  and the dashed colored lines denote the constraints  $g_{n+k|k}$  for different time instances and for  $n \in \mathbb{I}_0^N$ . The blue lines denote the predictions made at time  $t = 0$  s ( $k_0 = 0$ ), the yellow lines denote the predictions made at time  $t = 3.9$  s ( $k_1 = 130$ ), and the purple lines denote the predictions made at time  $t = 12$  s ( $k_2 = 400$ ). The green and black dashed lines show the constraint  $g_{n|k}$  at time  $t = 9.15$  s ( $n = 305$ ) as predicted at times  $t = 8.7$  s ( $k = 290$ ) and  $t = 9$  s ( $k = 300$ ), respectively.

#### D. Robotic Arm: Flexible Tracking with Obstacles

We consider a fully actuated planar robot with two degrees of freedom, no friction, nor external forces, and dynamics

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ B^{-1}(x_1)(u - C(x_1, x_2)x_2 - g(x_1)) \end{bmatrix}, \quad (27)$$

where  $x_1 = (q_1, q_2)$  are the joint angles, and  $x_2 = (\dot{q}_1, \dot{q}_2)$  the joint velocities. The full model description and the parameter values are given in Appendix A. We consider the following box constraints on the state and control

$$\|u\|_\infty \leq \bar{u}, \quad \|x_2\|_\infty \leq \bar{q}, \quad (28)$$

with  $\bar{u} = 4000$  Nm and  $\bar{q} = (3/2)\pi$  rad/s. We consider

$$p(\theta) = \left( \theta - \frac{\pi}{3}, 5 \sin \left( 0.6 \left( \theta - \frac{\pi}{3} \right) \right) \right), \quad (29)$$

with  $\theta \in [-5.3, 0]$ , as the desired path to be tracked and define the timing law

$$\dot{\theta}(t) = \frac{v_{\text{ref}}(t)}{\|\nabla_{\theta} p(\theta(t))\|_2}, \quad v_{\text{ref}}(t) = \begin{cases} 1 & t \leq 5 \\ \max(1 - at, 0) & t > 5 \end{cases},$$

with  $a = -0.0734$ . This predefined path evolution implies that the norm of the reference trajectory for the joint velocities will be 1 rad/s for  $t \leq 5$ , and monotonically decreasing for  $t > 5$  until reaching zero velocity. The state and input reference trajectories are given by

$$\begin{aligned} \mathbf{r}^x(t) &= \begin{bmatrix} p(\theta(t)) & \frac{\partial p}{\partial \theta} \dot{\theta}(t) \end{bmatrix}^\top, \\ \mathbf{r}^u(t) &= \begin{bmatrix} B(x_1) \ddot{p}(\theta(t)) + C(x_1, x_2)x_2 + g(x_1) \end{bmatrix}^\top, \end{aligned}$$

where the control reference follows from (27).

In order to construct  $g$ , we introduce the following model of a moving obstacle

$$\mathbf{w}_{k+1} = \omega(\mathbf{w}_k, \boldsymbol{\xi}_k) = \mathbf{w}_k + \begin{bmatrix} 0.3t_s \cos(\pi/4) + \xi_{1,k} \\ 0.3t_s \sin(\pi/4) + \xi_{2,k} \end{bmatrix}, \quad (30)$$

where  $\mathbf{w} \in \mathbb{R}^2$  is the obstacle state,  $\|(\xi_1, \xi_2)\|_2 \leq 0.03$  are bounded noise inputs, and  $t_s = 0.03$  s is the sampling time. To

construct the robust constraint under Assumption 4 we must first compute the set of reachable states

$$\mathcal{W}_{n+1|k} = \{\omega(\mathbf{w}_{n|k}, \boldsymbol{\xi}_n) \mid \mathbf{w}_{n|k} \in \mathcal{W}_{n|k}, \|\boldsymbol{\xi}_n\|_2 \leq 0.03\}$$

for some initial  $\mathcal{W}_{k|k} = \mathbf{w}_{k|k}$ . An over-approximation of such a set can be computed by first propagating a disturbance-free motion of the obstacle

$$\mathbf{w}_{n+1|k} = \mathbf{w}_{n|k} + \begin{bmatrix} 0.3t_s \cos(\pi/4) \\ 0.3t_s \sin(\pi/4) \end{bmatrix}, \quad (31)$$

with  $\mathbf{w}_{k|k} = \mathbf{w}_k$ . Then, one can encapsulate the uncertainty from the noise with a circle of increasing radius

$$\hat{r}_{n+1|k} = \hat{r}_{n|k} + \delta\hat{r}, \quad (32)$$

where  $\delta\hat{r} = 0.03$  and  $\hat{r}_{k|k} = \delta\hat{r}$ . The constraint  $g$  can then be expressed as

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) = \hat{r}_{n|k}^2 - (\mathbf{x}_{n|k}^{1,2} - \mathbf{w}_{n|k})^\top (\mathbf{x}_{n|k}^{1,2} - \mathbf{w}_{n|k}),$$

where  $\mathbf{x}_{n|k}^{1,2} = (q_{1,n|k}, q_{2,n|k})$ . Generally, one could include a more complex environment uncertainty and prediction model, however, here we choose to use simple models to highlight the performance of our framework in a clear manner.

For the cost we use  $W = \text{blockdiag}(Q, R)$  with

$$Q = \text{diag}(10^5, 10^5, 10, 10), \quad R = \text{diag}(10^{-3}, 10^{-3}), \quad w = 10.$$

The terminal cost matrix is given by  $P = P_\eta$ , and corresponding stabilizing set

$$\mathcal{X}_r^s(\tau_{k+N|k}) = \{\mathbf{x}_{k+N|k} \mid \Delta \mathbf{x}_{k+N|k}^\top P_\eta \Delta \mathbf{x}_{k+N|k} \leq \gamma_*\}, \quad (33)$$

where the values and derivation of  $P_\eta$  and  $\gamma_*$  are given in Appendix B. The safety set is constructed as

$$\mathcal{X}_{\text{safe}}(t) = \{(\dot{q}_1, \dot{q}_2) \mid \dot{q}_1 = 0, \dot{q}_2 = 0\} \cap \mathcal{X}_r^s(t), \quad (34)$$

and we use the terminal set  $\mathcal{X}_r^f(\tau_{k+N|k})$  given by (14) using (33) and (34). Since the safe set ensures a steady state, a minimal feasibility condition is that  $\mathbf{x}_{k+N|k} = (p(\theta(\tau_{k+N|k})), 0, 0) \in \mathcal{X}_r^f(\tau_{k+N|k})$ .

For the simulation, we set the sampling time to  $t_s = 0.03$  s and use control intervals  $N = 25$ ,  $M = 50$ , and  $\mathbf{w}_0 = (-6, -2)$  as initial value for the uncertainty. The initial value for  $\tau_0$  is selected by projecting  $(q_1, q_2)$  on the trajectory, i.e.,

$$\tau_0 = \arg \min \| (q_1, q_2) - p(\theta(\tau)) \|_2.$$

Figure 6 shows the closed-loop trajectories for the initial condition  $(x_1, x_2) = (-5.86, 2.43, 0, 0)$ ,  $\tau = 0.79$ . The system is quickly stabilized to the reference; after 3 s the system deviates from the desired velocity reference due to the presence of an uncertainty. Tracking is temporarily lost for the velocities  $(\dot{q}_1, \dot{q}_2)$  but to a lesser extent for the positions  $(q_1, q_2)$ . When the obstacle leaves, the system is again stabilized towards the reference.

Figure 7 shows three open-loop trajectories of the joint positions together with the corresponding predictions of the moving obstacle. The gray line shows the reference trajectory and the red line illustrates the closed-loop trajectory of the obstacle  $\mathbf{w}_k$  for  $t = [0, 12]$  s. The blue lines show the predictions of  $(q_{1,n+k|k}, q_{2,n+k|k})$  and  $g_{n+k|k}(\mathbf{x}, \mathbf{u}) = 0$  for

$n \in \mathbb{I}_0^N$  at time  $t = 0$  s ( $k = 0$ ), while the yellow lines show the predictions at time  $t = 3.9$  s ( $k = 130$ ), and the purple lines show the predictions at time  $t = 12$  s ( $k = 400$ ). The green and black dashed lines show that  $g_{305|300} \leq g_{305|290}$ , i.e., that Assumption 4 is satisfied for  $g_{n|k}$ . At times  $t = 0$  s and  $t = 12$  s the robotic arm can track the reference trajectory freely, while for  $t = 3.9$  s the robotic arm is forced to come close to a stop, and even reverse, in order to avoid the obstacle before being allowed to continue along the reference trajectory. We observe that the simulation results remain indistinguishable if the initial auxiliary state  $\tau_{k|k}$  is free to be selected by the optimizer, as noted in Remark 3. Finally, we recorded an average run time of  $\hat{t}_{\text{MPFTC}} = 0.235$  s. While the run time might seem large, we ought to stress that our implementation was not based on efficient real-time algorithms which are expected to be much faster.

## VI. CONCLUSIONS

In this paper, we have introduced a new predictive control framework solving the problem of tracking reference trajectories, which might become infeasible in the presence of a-priori unknown constraints, while avoiding undesirably aggressive behaviors. We have discussed safety in a general sense, and provided new terminal conditions, which hold for reference trajectories with mild assumptions, and ensure safety and recursive feasibility with respect to a-priori unknown, time-varying constraints. Our proposed approach can be combined with existing motion planning techniques in order to deliver better performance. Future work will focus on practical real-time implementations of the framework in the context of urban autonomous driving, where moving obstacles are detected by the onboard sensors and are a-priori unknown.

## REFERENCES

- [1] D. Mayne, J. Rawlings, C. Rao, and P. Sokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [2] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*. Nob Hill Pub. Madison, Wisconsin, 2009.
- [3] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
- [4] O. Ljungqvist, D. Axehill, and A. Helmersson, "Path following control for a reversing general 2-trailer system," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 2455–2461.
- [5] O. Ljungqvist, D. Axehill, and J. Löfberg, "On stability for state-lattice trajectory tracking control," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 5868–5875.
- [6] O. Andersson, O. Ljungqvist, M. Tiger, D. Axehill, and F. Heintz, "Receding-horizon lattice-based motion planning with dynamic obstacle avoidance," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4467–4474.
- [7] O. Ljungqvist, N. Evestedt, D. Axehill, M. Cirillo, and H. Pettersson, "A path planning and path-following control framework for a general 2-trailer with a car-like tractor," *Journal of Field Robotics*, vol. 36, no. 8, pp. 1345–1377, 2019.
- [8] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [9] M. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, 2018.
- [10] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

- [11] J. P. Van Den Berg and M. H. Overmars, "Roadmap-based motion planning in dynamic environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 885–897, 2005.
- [12] C. Fulgenzi, C. Tay, A. Spalanzani, and C. Laugier, "Probabilistic navigation in dynamic environment using rapidly-exploring random trees and gaussian processes," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1056–1062.
- [13] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotovic, "Path-following for nonminimum phase systems removes performance limitations," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 234–239, 2005.
- [14] T. Faulwasser, B. Kern, and R. Findeisen, "Model predictive path-following for constrained nonlinear systems," in *Proceedings of the 48th IEEE Conference on Decision and Control. CDC, 2009*. IEEE, 2009, pp. 8642–8647.
- [15] K. Kanjanawanishkul and A. Zell, "Path following for an omnidirectional mobile robot based on model predictive control," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 3341–3346.
- [16] A. Alessandretti, A. P. Aguiar, and C. N. Jones, "Trajectory-tracking and path-following controllers for constrained underactuated vehicles using model predictive control," in *2013 european control conference (ecc)*. IEEE, 2013, pp. 1371–1376.
- [17] T. Faulwasser and R. Findeisen, "Nonlinear model predictive control for constrained output path following," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 1026–1039, April 2016.
- [18] T. Faulwasser, T. Weber, P. Zometa, and R. Findeisen, "Implementation of nonlinear model predictive path-following control for an industrial robot," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1505–1511, July 2017.
- [19] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 2210–2215.
- [20] S. B. Liu, H. Roehm, C. Heinzemann, I. Lütkebohle, J. Oehlerking, and M. Althoff, "Provably safe motion of mobile robots in human environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1351–1357.
- [21] D. Beckert, A. Pereira, and M. Althoff, "Online verification of multiple safety criteria for a robot trajectory," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 6454–6461.
- [22] S. Gros and M. Zanon, "Penalty functions for handling large deviation of quadrature states in nmpc," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3848–3860, Aug 2017.
- [23] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. London: Springer, 2011.
- [24] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967 – 2986, 2014.
- [25] E. C. Kerrigan, "Robust constraint satisfaction: Invariant sets and predictive control," Ph.D. dissertation, University of Cambridge, 2001.
- [26] S. Yu, C. Maier, H. Chen, and F. Allgöwer, "Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems," *Systems & Control Letters*, vol. 62, no. 2, pp. 194–200, 2013.
- [27] 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, no. 1, pp. 1–36, 2019.
- [28] 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, no. 1, pp. 25–57, 2006.
- [29] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference, Zürich, Switzerland, July 17–19 2013*, pp. 502–510, <http://control.ee.ethz.ch/~mpt>.
- [30] P. Scokaert and J. Rawlings, "Feasibility Issues in Linear Model Predictive Control," *AICHE Journal*, vol. 45, no. 8, pp. 1649–1659, 1999.
- [31] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge, UK: Cambridge university press, 2004.
- [32] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*. IEEE, 2004, pp. 284–289.



**Ivo Batkovic** received his M.Sc degree in Engineering Physics in 2016, and his Ph.D. degree in Electrical Engineering in 2022, both from Chalmers University of Technology. He is currently a research engineer at Zenseact in Gothenburg, Sweden, working with decision-making and control for autonomous driving. His research interests include model predictive control for constrained autonomous systems in combination with prediction models for safe decision-making in autonomous driving applications.



**Mohammad Ali** received his M.Sc degree in Electrical Engineering and Ph.D. degree in Mechatronics from Chalmers University of Technology in 2005 and 2012, respectively. He is currently responsible for Autonomous driving development at Zenseact AB. His research interests include, methods for safe decision making, trajectory planning with safety guarantees and verification of safety critical perception and control systems.



**Paolo Falcone** received his M.Sc. ("Laurea degree") in 2003 from the University of Naples "Federico II" and his Ph.D. degree in Information Technology in 2007 from the University of Sannio, in Benevento, Italy. He is Professor at the Department of Electrical Engineering of the Chalmers University of Technology, Sweden. His research focuses on constrained optimal control applied to autonomous and semi-autonomous mobile systems, cooperative driving and intelligent vehicles, in cooperation with the Swedish automotive industry, with a focus on autonomous driving, cooperative driving and vehicle dynamics control.



**Mario Zanon** received the Master's degree in Mechatronics from the University of Trento, and the Diplôme d'Ingénieur from the Ecole Centrale Paris, in 2010. After research stays at the KU Leuven, University of Bayreuth, Chalmers University, and the University of Freiburg he received the Ph.D. degree in Electrical Engineering from the KU Leuven in November 2015. He held a Post-Doc researcher position at Chalmers University until the end of 2017 and is now Assistant Professor at the IMT School for Advanced Studies Lucca. His research interests include numerical methods for optimization, economic MPC, reinforcement learning, and the optimal control and estimation of nonlinear dynamic systems, in particular for aerospace and automotive applications.

## APPENDIX

### A. Model details

The functions used in (27) are defined as

$$B(q) := \begin{bmatrix} b_1 + b_2 \cos(q_2) & b_3 + b_4 \cos(q_2) \\ b_3 + b_4 \cos(q_2) & b_5 \end{bmatrix}, \quad (35a)$$

$$C(q, \dot{q}) := -c_1 \sin(q_2) \begin{bmatrix} \dot{q}_1 & \dot{q}_1 + \dot{q}_2 \\ -\dot{q}_1 & 0 \end{bmatrix} \quad (35b)$$

$$g(q) := \begin{bmatrix} g_1 \cos(q_1) + g_2 \cos(q_1 + q_2) \\ g_2 \cos(q_1 + q_2) \end{bmatrix}, \quad (35c)$$

with all parameters given in Table I.

TABLE I  
MODEL PARAMETERS FOR ROBOTIC JOINT

$b_1$	200.0	[kg m <sup>2</sup> / rad]	$b_2$	50.0	[kg m <sup>2</sup> / rad]
$b_3$	23.5	[kg m <sup>2</sup> / rad]	$b_4$	25.0	[kg m <sup>2</sup> / rad]
$b_5$	122.5	[kg m <sup>2</sup> / rad]	$c_1$	-25.0	[Nms <sup>-2</sup> ]
$g_1$	784.8	[Nm]	$g_2$	245.3	[Nm]

### B. Terminal set computation

In order to construct the terminal region, similarly to [17], we apply the following variable transformation

$$\eta_1 = x_1 - p(\theta(\tau)), \quad \eta_2 = x_2 - \frac{\partial p}{\partial \theta} \dot{\theta}(\tau), \quad (36)$$

and rewrite (27) as a deviation from the trajectory

$$\begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} \eta_2 \\ \alpha(\eta, \tau, u) \end{bmatrix}, \quad (37)$$

$$\begin{aligned} \alpha(\eta, \tau, u) = & B^{-1}(\eta, \tau) \left( u - C(\eta, \tau) (\eta_2 + \frac{\partial p}{\partial \theta} \dot{\theta}(\tau)) \right. \\ & \left. - g(\eta_1, \tau) \right) - \frac{\partial^2 p}{\partial \theta^2} \dot{\theta}^2(\tau) - \frac{\partial p}{\partial \theta} \ddot{\theta}(\tau). \end{aligned}$$

Exploiting this form, we define the terminal feedback law

$$\begin{aligned} u_{\mathcal{X}}(\eta, \tau) = & C(\eta, \tau) \left( \eta_2 + \frac{\partial p}{\partial \theta} \dot{\theta}(\tau) \right) + g(\eta_1, \tau) \\ & + B(\eta_1, \tau) (-K_{\eta} \eta + \ddot{p}(\theta(\tau))), \end{aligned} \quad (38)$$

yielding closed-loop dynamics  $\dot{\eta} = (A_{\eta} - B_{\eta} K_{\eta}) \eta$ , with corresponding Lyapunov function

$$V(\eta) = \eta^{\top} P_{\eta} \eta, \quad P > 0. \quad (39)$$

We define the terminal region  $\mathcal{X}_{\eta} \in \mathbb{R}^4$  as level set of (39), and use the following bounds

$$\forall x \in \mathcal{X} : \|B(x_1)\|_2 \leq \bar{B}, \|C(x_1, x_2)\|_2 \leq \bar{C}, \|g(x_1)\|_2 \leq \bar{g}.$$

The upper bounds on  $\|\dot{p}(\theta(t))\|_2 \leq \bar{p}$  and  $\|\ddot{p}(\theta(t))\|_2 \leq \bar{\ddot{p}}$  are given directly through the design of the timing law. Furthermore, we tighten the state and input constraints

$$\|u\|_2 \leq \bar{u}, \quad \|x_2\|_2 \leq \bar{q}.$$

In order to obtain a tightened bound on the terminal control input we impose

$$\|u_{\mathcal{X}}(\eta, \tau)\|_2 \leq \bar{C} \bar{q} + \bar{g} + \bar{B} (\bar{p} + \|K_{\eta} \eta\|_2) \leq \bar{u}, \quad (40)$$

which in turn yields

$$\|\eta\|_2 \leq \frac{\bar{u} - \bar{C} \bar{q} - \bar{g} - \bar{B} \bar{p}}{\bar{B} \|K_{\eta}\|_2}. \quad (41)$$

Finally, the terminal set is given as the level set of the Lyapunov function  $V(\eta)$

$$\mathcal{E}_{\eta} = \{\eta \in \mathbb{R}^4 \mid \eta^{\top} P_{\eta} \eta \leq \gamma_*\}, \quad (42)$$

where parameter  $\gamma_*$  maximizes the volume of the ellipsoid. Given  $P_{\eta}$  and  $K_{\eta}$ , we use the S-procedure [31] to formulate the maximization as convex optimization problem

$$\gamma_* := \max_{\gamma, \lambda} \gamma \quad (43a)$$

$$\text{s.t.} \quad \begin{bmatrix} P_{\eta} & \\ & -\gamma \end{bmatrix} - \lambda_1 \begin{bmatrix} I & \\ & -d_1 \end{bmatrix} \succeq 0, \quad (43b)$$

$$\begin{bmatrix} P_{\eta} & \\ & -\gamma \end{bmatrix} - \lambda_2 \begin{bmatrix} \tilde{I}^0 & \\ & -d_2 \end{bmatrix} \succeq 0, \quad (43c)$$

$$\lambda \geq 0, \quad (43d)$$

where  $\tilde{I}^0 = \text{blockdiag}(0, 0, I) \in \mathbb{R}^4$  and

$$d_1 = \frac{\bar{u} - \bar{C} \bar{q} - \bar{g} - \bar{B} \bar{p}}{\bar{B} \|K_{\eta}\|_2}, \quad d_2 = \bar{q} - \bar{p}.$$

Constraint (43b) ensures that (41) holds, i.e., the terminal control  $u_{\mathcal{X}}(\xi, \eta)$  satisfies (40); while constraint (43c) ensures that  $\|x_2\|_2 \leq \bar{q}$ . To solve (43), we use the model data from Table I to get the bounds:  $\bar{B} = 266.4$ ,  $\bar{C} = 269.6$  and  $\bar{g} = 1058.9$ . From the timing law we know that  $\bar{p} = 1$  and  $\bar{\ddot{p}} = 0.823$ . We compute the feedback matrix  $K_{\eta}$  via an LQR controller with tuning  $Q_{\eta} = I$ , and  $R_{\eta} = 10I$  to get

$$K_{\eta} = \begin{bmatrix} K_1 & K_2 \end{bmatrix}, \quad (44)$$

where  $K_1 = 0.31 \cdot \text{diag}(1, 1)$  and  $K_2 = 0.85 \cdot \text{diag}(1, 1)$ . Finally, the terminal cost is obtained by solving the Lyapunov equation

$$P_{\eta} = (A_{\eta} - B_{\eta} K_{\eta})^{\top} P_{\eta} (A_{\eta} - B_{\eta} K_{\eta}) + (Q + K_{\eta}^{\top} R K_{\eta}),$$

which gives us

$$P_{\eta} = 10^6 \cdot \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix}, \quad (45)$$

with  $P_1 = 6.51 \cdot \text{diag}(1, 1)$ ,  $P_2 = 5.27 \cdot \text{diag}(1, 1)$ ,  $P_3 = 6.16 \cdot \text{diag}(1, 1)$ . Then, solving (43) using [32], we obtain

$$\mathcal{E}_{\eta} = \{\eta \in \mathbb{R}^4 \mid \eta^{\top} P_{\eta} \eta \leq 2.29 \cdot 10^7\}. \quad (46)$$