Abstract—In this article, we propose a novel model predictive control (MPC) framework for output tracking that deals with partially unknown constraints. The MPC scheme optimizes over a learning and a backup trajectory. The learning trajectory aims to explore unknown and potentially unsafe areas, if and only if this might lead to a potential performance improvement. On the contrary, the backup trajectory lies in the known space, and is intended to ensure safety and convergence. The cost function for the learning trajectory is divided into a tracking and an offset cost, while the cost function for the backup trajectory is only marginally considered and only penalizes the offset cost. We show that the proposed MPC scheme is not only able to safely explore the unknown constraints but also escape from local minima that may arise from the presence of obstacles. Moreover, we provide formal guarantees for convergence and recursive feasibility of the MPC scheme, as well as closed-loop constraint satisfaction. Finally, the proposed MPC scheme is demonstrated in simulations using an example of autonomous vehicle driving in a partially unknown environment where unknown obstacles are present.

Index Terms—Model predictive control, optimal control, learning systems.

I. INTRODUCTION

MOTIVATION model predictive control (MPC) [1], [2], [3] is an optimization-based control technique that is widely applied for constrained nonlinear systems. The performance of MPC largely depends on the accuracy of the prediction model, as well as on the accuracy of the constraint set, which are often inaccurate or partially unknown. In order to improve the closed-loop control performance, learning-based MPC approaches aim by actively refining the model (e.g., [4], [5], [6], [7], [8]) and the constraint set (e.g., [9], [10]) online. In several applications, such as autonomous driving, the system evolves in a partially unknown environment wherein the control objective is to get as close to a desired, but potentially nonreachable destination as possible. Due to the potentially changing environment wherein the system evolves, active exploration of unknown area may be necessary to improve the control performance. However, even though such an exploration process might be beneficial, it can also lead to violation of system constraints, loss of recursive feasibility of the MPC scheme, and instability of the system itself. In addition, it is important to note that the presence of obstacles, combined with a (potentially nonconvex) constraint set, might lead the system to converge into local minima, as shown in Fig. 1.

Related work: Optimizing for performance while ensuring safety is the goal of many applications. In theory, the ideal closed-loop performance under model or constraint inaccuracy can be realized using dual control [11]. However, except for a few specific problem setups (e.g., [12], [13]), a computationally tractable reformulation of the dual control problem is, in general, not attainable. For this reason, several approaches modify the control problem by artificially introducing a heuristic cost function that explicitly excites the system, as well as novel learning formulations that have the goal to reduce uncertainties affecting the system, e.g., [14], [15], [16]; see [17] for a recent survey on dual control. In [9] and [18], a control approach is proposed for known systems that evolve in a partially unknown environment.
and are equipped with a sensor that detects its surrounding. The approach relies on the notion of constructing two different trajectories, one for exploration and one for exploitation. The work additionally proposes a compelling computationally tractable solution for constructing convex safe sets, as demonstrated on several practical scenarios. On the other hand, the approach only focuses on navigation for UAVs, and does not provide theoretical guarantees, which is one of the objectives of this work. In [19], an MPC scheme for path following that deals with obstacle avoidance is presented. Collision avoidance is ensured by appropriately modifying the value function, rather than introducing constraints in the state space. A similar problem setup is also considered in [20], where, in addition to collision avoidance, the goal is to ensure recursive feasibility of the MPC scheme while tracking a reference trajectory, even when this becomes infeasible due to a priori unknown obstacles. In [21], the authors consider the case where the sensing quality of the unknown constraints is of statistical nature and can depend upon system operation. As such, a perception-aware chance-constrained MPC scheme is proposed that seeks to tradeoff between the impact of control on sensing and sensing on control. The stability properties of the approach proposed in [21] are subsequently analyzed in [22]. The idea of constructing multiple trajectories is also exploited in [23] with the goal of ensuring closed-loop constraint satisfaction in a robust MPC setting. In [10], a multitrajectory MPC approach is used for autonomous UAV navigation, wherein the problem setup only considers linear systems and does not provide convergence guarantees. Dealing with partially unknown models or constraints is also of high interest in the field of machine learning (ML). In [24], a framework that works as an add-on to existing interactive machine learning (IML) algorithms with the aim to render them safe is proposed. In this case, given a possibly unsafe suggestion by the IML algorithm, the algorithm safely learns about the safety of this decision by exploiting continuity properties of the constraints in terms of a GP prior. In [25], it is shown an approach to tackle chance-constrained trajectory planning problems with nonconvex constraints, while in [26] a way to combine ideas from robust control and GP-based RL to design an MPC scheme that recursively guarantees the existence of a safety trajectory that satisfies the constraints of the system is analyzed. Finally, in [27], an MPC scheme is employed as a new type of function approximator in reinforcement learning (RL) in order to ensure that the MPC can deliver the optimal policy of the real system even when the nominal model is wrong.

In order to increase the region of attraction of an MPC scheme, the approach in [28] introduces the idea of online computation of a so-called artificial setpoint. In particular, the cost function is divided into a tracking cost, which penalizes the distance between the system and the artificial setpoint, and an offset cost that steers the artificial setpoint toward the desired one. The main advantages of the approach in [28] are the increased region of attraction of the MPC scheme, the possibility to consider potentially unreachable setpoints, and ensuring recursive feasibility even when the desired state changes over time. This scheme has been extended for output tracking of nonlinear systems in [29] and for deterministic and stochastic uncertainties in [30], [31], [32], and [33]. To simplify its implementation, in [29] and [34], the case of no terminal ingredients is considered. In order to ensure convergence on a given setpoint, approaches for self-tuning terminal cost in economic MPC are analyzed in [35] and [36]. Both the papers propose ideas to escape local minima by appropriately tuning the terminal cost online. However, in both the approaches, the knowledge of the best achievable setpoint is required, and the overall problem setup differs from the one considered in this work since it does not consider learning or exploration.

Contributions: For nonlinear systems with partially unknown constraints, this article presents a novel MPC scheme that enables the controlled system to safely explore unknown areas within the state-space while ensuring convergence to the closest reachable setpoint. In particular, the MPC scheme considers two trajectories: 1) a learning trajectory that is intended to improve the performance by exploring unknown areas of the state-space; and 2) a backup trajectory to ensure safety as well as to find the best possible setpoint in the reachable space. The novelty of the proposed MPC scheme is summarized as follows.

1) Exploration-Oriented Cost: The proposed MPC scheme mainly optimizes over the learning trajectory, and only marginally considers the backup trajectory, which is multiplied with a small weight. This way the learning trajectory is left free to improve the performance by exploring unknown areas and, thus, does not involve a tradeoff between exploration and exploitation.

2) Escaping Local Minima: Thanks to the proposed cost function and constraints in the MPC scheme, we show that the system is able to escape, in closed-loop, a subset of local minima that can possibly result from the presence of obstacles, or the given constraint set, as shown in Fig. 1. Such a subset of local minima is formally defined in Definition 2.

3) Output Tracking: Motivated by practical applications and theoretical works, the proposed approach minimizes the distance between the output of the system and a desired output. The desired output can be nonreachable, and may change over time without influencing the theoretical properties of the proposed approach.

4) Theoretical Guarantees: In addition recursive feasibility of the proposed MPC scheme (which implies closed-loop constraint satisfaction), we propose an additional constraint in the MPC scheme to ensure convergence to the closest reachable setpoint. Using a simple motivating example, we demonstrate how the lack of such a constraint cannot ensure convergence.

As an additional contribution, we also extend the work presented in [28] and [29], to the case of nonconnected steady-state manifolds. Such an extension is needed in order to have a more fair comparison between the theoretical results obtained in the proposed approach and the additional ones obtainable by extending [29].

Outline: Section II presents the problem setup and shows the significance of exploring unknown areas within the state-space using a motivating example. In Section III, the preliminaries of the MPC scheme are presented, while discussing the
shortcomings of an MPC scheme that employs only one trajectory for the problem setup at hand (i.e., partially unknown constraints). The proposed MPC scheme is presented in Section IV, along with its theoretical analysis in Section V. Section VI demonstrates the effectiveness of the MPC scheme using a numerical example. Finally, Section VII concludes this article.

Notation: The quadratic norm with respect to a positive definite matrix $Q = Q^\top$ is denoted by $\|x\|_Q = x^\top Q x$. The minimal and maximal eigenvalue of $Q$ are denoted by $\lambda_{\text{min}}(Q)$ and $\lambda_{\text{max}}(Q)$, respectively. $\mathbb{R}_{\geq 0} = \{ r \in \mathbb{R} | r \geq 0 \}$ denotes positive real numbers. By $K_{x_0}$, we denote the class of functions $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, which are continuous, strictly increasing, unbounded, and satisfy $\alpha(0) = 0$. A variable $x$ predicted at time $t$ for $k$ steps ahead will be denoted by $x_{k|t}$, where $0 \leq k \leq N$, with $N \in \mathbb{N}$ being the prediction horizon. The notation $x_{|t}$ will be used when the entire predicted trajectory is considered. The variable $B_a$ represents a hyper-ball defined with respect to the Euclidean norm in a given dimension with radius $a \in \mathbb{R}_{\geq 0}$, centered at the origin.

II. PROBLEM SETUP

In this article, we consider the following nonlinear, time-invariant, and discrete-time system:

\begin{equation}
    x_{t+1} = f(x_t, u_t) \\
    y_t = h(x_t, u_t)
\end{equation}

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the system dynamics, $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ is the output function, $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, $y \in \mathbb{R}^p$ is the output, and $t \in \mathbb{N}$ is the time. At each time instant $t \in \mathbb{I}_{\geq 0}$, the system is subject to the following state and input constraints:

\begin{equation}
    (x_t, u_t) \in Z \subseteq \mathbb{R}^{n+m}
\end{equation}

where $Z$ is a compact and connected set. Note that we do not require convexity of the set $Z$, since this might directly include potential obstacles that the system must avoid.

Assumption 1: The constraint set $Z$ is only partially known, i.e., at each time instant $t \in \mathbb{I}_{\geq 0}$, the set $\mathbb{R}^{n+m}$ is subdivided into a known compact safe set $E_t$ that satisfies

\begin{equation}
    E_t \subseteq Z
\end{equation}

and a remaining unknown set $Z \setminus E_t$.

Assumption 2: There exists a neighborhood $H : \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$ of the state and input pair $z_t = (x_t, u_t)$ that satisfies the following:

\begin{equation}
    [z_t + B_h] \cap Z \subseteq H(z_t) \subseteq E_t \forall t \in \mathbb{I}_{\geq 0}
\end{equation}

where $h \in \mathbb{R}_{>0}$ is a known parameter that satisfies $h \ll 1$. Moreover, at each time instant $t \in \mathbb{I}_{\geq 0}$, an estimated constraint set $Z_t$ is available, such that the following condition holds for all $t \in \mathbb{I}_{\geq 0}$:

\begin{equation}
    E_t \subseteq Z_t.
\end{equation}

The set $B_h$ in (3) is only necessary to ensure that $z_t \in H(z_t) \Rightarrow z_t \in \text{int}(Z)$, which is then required in the theoretical analysis, see Section V-B2. Such a requirement can also be removed at the price of a more involved theoretical analysis.

As an example, in several autonomous systems, the set $H(z_t)$ might correspond to the set detected from a LIDAR sensor, see Fig. 2. The estimated constraint set $Z_t$ contains, in addition to the safe set $E_t$, also the set that we wish to explore, without, however, having guarantees on its safety. In order to improve the performance of the proposed approach (discussed in Section IV), the estimated constraint set $Z_t$ should be constructed as similar as possible to the actual constraint set $Z$. However, if no information about the constraint set $Z$ is available, then Assumption 2 is satisfied by choosing $Z_t = \mathbb{R}^{n+m}$.

Steady-State Manifold: Given a set $A \subseteq Z$, the steady-state and input manifold $S(A)$ is defined as follows:

\begin{equation}
    S(A) := \{(x, u) \in \mathbb{R}^{n+m} | (x, u) + B_h \subseteq A, x = f(x, u)\}
\end{equation}

where $\lambda \ll h$. Note that the set $B_h$ is only necessary in order to ensure that $S(A) \subseteq \text{int}(A)$, which is required for the local controllability argument discussed in Assumption 5, and later used in the theoretical analysis in Section V.

Remark 1: Even if the set $A$ is convex, the manifold $S(A)$ can nevertheless be nonconnected. In this article, we explicitly deal with such a problem setup, and show the theoretical results that can be obtained with existing approaches and compare them with the proposed MPC framework.

Control goal: The goal of the proposed approach is to steer the output $y_t$ of system (1) as close as possible to a user-defined, potentially nonreachable, output $y^d \in \mathbb{R}^p$, while satisfying the constraint set $Z$ in (2). Simultaneously, exploration of the unknown and potentially unsafe space $Z \setminus E_t$ must be incentivized only if this might lead to a performance improvement.

A. Motivating Example

Consider the case of an autonomous vehicle driving in a partially unknown environment, with the goal to get as close as possible to a desired and nonreachable output $y^d$ while satisfying a constraint set $Z$, as shown in the left in Fig. 3. Note that since the output $y^d$ is not reachable, the only achievable goal is to reach the closest reachable output denoted with $y^d_r$. Such an output $y^d_r$ should, ideally, be directly computed by the MPC scheme, and therefore might not be known a priori. Note that for the sake of simplicity, in the following example we focus on conveying
the high-level motivations of the proposed approach, without employing rigorous mathematical tools.

First, the car is subject to internal constraints that limit its velocity, acceleration, and steering angle, and are encoded in the set $\mathcal{Z}_{sc}$. At time $t=0$, the safe set $\mathcal{E}_0$ is formed by the internal constraints $\mathcal{Z}_{sc}$, and the safe path “A,” i.e.,
\[
\mathcal{E}_0 := \mathcal{Z}_{sc} \cap A.
\]
On the contrary, the path “B” is unknown, and therefore cannot be considered as safe. In order to enable the system to explore the path “B,” we define the set $\mathcal{Z}_0$ as follows:
\[
\mathcal{Z}_0 := \mathcal{Z}_{sc} \cap (A \cup B).
\]
In scenario i and scenario ii, we show two of the potential outcomes resulting from exploring the path “B.” In the following, we give a more detailed explanation about the tradeoff between exploration and exploitation.

**Exploration:** As classical in the tracking MPC literature, [1], [2], [3], [28], the MPC scheme exploits the safe constraint set $\mathcal{E}_0$ in order to find the optimal open-loop trajectory that minimizes a given cost function while ensuring constraint satisfaction. In the considered example, this means that the vehicle neglects the path “B,” and directly follows the path “A,” which is, however, potentially slower.

**Exploitation:** Exploration can be achieved by actively incentivizing the system to explore the potentially unsafe set $\mathcal{Z}\setminus \mathcal{E}_0$. In the considered example, this means that the vehicle is directed toward the path “B,” even though this is not known at time $t=0$. As the system explores new safe areas, the safe set $\mathcal{E}_e$ expands, and the resulting performance might significantly improve. However, as shown in scenario i in Fig. 3, exploration of unknown areas might lead to constraint violation if not appropriately done, loss of recursive feasibility, as well as the possibility to get stuck in a new local minimum, as shown in Fig. 1. On the contrary, in scenario ii, path “B” is clear and the vehicle is able to make use of the less stringent speed limit, and therefore arrives at the closest reachable output in a shorter amount of time.

In conclusion, the desired approach should not only incentivize exploration, but must simultaneously ensure constraint satisfaction, convergence, and should also escape from potential local minima. Note that in Section IV-A, we consider the example above at a more mathematical level, where we show additional problems that existing approaches suffer from.

### III. Preliminaries for MPC Scheme

In this section, we discuss the preliminaries needed for the proposed MPC scheme, and we extend to the case of nonconnected manifolds $\mathcal{S}(\mathcal{Z})$ the work done in [29], [37], and [34]. In particular, in Section III-A, we first introduce an MPC scheme that employs the concept of artificial setpoints. Such a scheme will serve as a basis for the proposed MPC framework that will be later introduced and discussed in Section IV. In Section III-B, we detail the tracking cost function as well as the terminal ingredients, while in Section III-C we discuss the so-called offset cost. In Section III-E, we present the theoretical analysis of such an MPC scheme extended to the case of nonconnected manifolds, and we discuss its shortcomings.

#### A. MPC Scheme With Artificial Setpoints and Nonconnected Manifold

In the following, we show an MPC scheme that is based on the concept of artificial setpoints and that deals with a nonconnected manifold $\mathcal{S}(\mathcal{Z})$. In a classical MPC scheme, see, e.g., [1], the optimization problem considers a user-defined tracking cost function which minimizes the distance between the predicted trajectory and a desired, reachable, setpoint $r^{d} = (x^{d}, u^{d}) \in \mathcal{S}(\mathcal{Z})$. On the contrary, the following MPC scheme simultaneously minimizes the tracking cost between the predicted trajectory and an online optimized artificial setpoint $r^{a}$, as well as an offset cost that penalizes the distance of the output of such an artificial setpoint $y^{a} = h(r^{a})$, and the desired output $y^{d}$. This idea was initially introduced in [28] for the case of linear systems, and is extended in [29], [37], and [34] to the case of nonlinear systems, output tracking, and output tracking for MPC without terminal ingredients, respectively. The main benefits of introducing an artificial setpoint are the following.

1) The desired state and input $(x^{d}, u^{d})$ that satisfy $y^{d} = h(x^{d}, u^{d})$ do not need to be known, nor to exist. Indeed, feasibility of the MPC scheme is not influenced by the desired output $y^{d}$, which can therefore be even unreachable, i.e., $(x^{d}, u^{d}) \notin \mathcal{Z}$, or it might not correspond to a steady-output of (1), i.e., $(x^{d}, u^{d}) \notin \mathcal{S}(\mathcal{Z})$.

2) The region of attraction of the MPC scheme increases compared to the one of a classical MPC scheme, i.e., where only the tracking cost function is considered. In particular, while in a classical MPC scheme the initial state $x_{0}$ needs to be in a neighborhood of the desired...
state $x^d$, in the considered scheme the state $x_0$ only needs to be in a neighborhood of the feasible manifold $S(Z)$.

3) Initial feasibility of the MPC scheme ensures recursive feasibility of the scheme even when the desired output $y^d$ is subject to online changes. Such a property is not guaranteed in a classical MPC scheme and is crucial in several applications where stabilizing the origin is not the ultimate goal.

At each time $t \geq 0$, given the state $x_t$ and a compact set $\mathbb{E}_t \subseteq \mathbb{Z}$, the following optimization problem is solved:

$$
\min_{x_{t:t+1}^r, r_t^r} eV_N(x_{t:t}, u_{t:t}, r_t^r) + T(r_t^r) \tag{6a}
$$

subject to:

$$
x_{k+1|t} = f(x_{k|t}, u_{k|t}) \tag{6b}
$$

$$
(x_{k|t}, u_{k|t}) \in \mathbb{E}_t \tag{6c}
$$

$$
x_{N|t} \in X_t(r_t^r, \mathbb{E}_t) \tag{6d}
$$

$$
r_t^r \in \mathbb{S}(\mathbb{E}_t) \tag{6e}
$$

$$
x_{0|t} = x_t \tag{6f}
$$

where $\epsilon \in \mathbb{R}_{>0}$ is a user-defined parameter, $V_N$ is the tracking cost function that penalizes the distance between the open-loop trajectory $(x_{t|t}, u_{t|t})$ and an online optimized artificial setpoint $r_t^r$, $X_t$ is the terminal region, while $T$ is the offset cost function that penalizes the distance between the output of such an artificial setpoint $y_t^r = h(r_t^r)$ and the user-defined desired output $y^d$. For additional details on $V_N$ and $T$ compare Section III-B and III-C, respectively.

Remark 2: Note that the constant $\epsilon$ is usually implicitly included in the function $V_N$, and therefore its ideal value is $\epsilon = 1$. However, in this article, we show the effects that $\epsilon$ plays on the MPC scheme (6), and for this reason we explicitly include it.

The optimal open loop trajectory resulting from (6) is denoted with $(x_{t|t}, u_{t|t})$, and the optimal setpoint is referred as $r_{t|t}^n := (x_{t|t}^*, u_{t|t}^*)$. The resulting closed-loop is given as follows:

$$
x_{t+1} = f(x_t, u_t), \quad u_t := u_{0|t}. \tag{7}
$$

Moreover, we define the function $V_N^*$ as follows:

$$
V_N^*(x, r_t^r, \mathbb{E}_t) := V_N(x_{t|t}, u_{t|t}, r_t^r). \tag{6}
$$

Remark 3: The optimal trajectory $x_{t|t}, u_{t|t}, r_{t|t}^n$ resulting from (6) is analogous to the optimal trajectory resulting from a classical MPC scheme where only the tracking cost $V_N$ is minimized with respect to the optimal setpoint $r_{t|t}^n \in \mathbb{S}(Z)$ (and not to the desired output $y^d$), i.e., where the offset cost $T$ is not considered and the setpoint $r_{t|t}^n$ is given and not optimized. For this reason, the optimal cost function $V_N^*$ only depends on the current state $x_t$, on the given setpoint $r_{t|t}^n$, and on the given set $\mathbb{E}_t$.

In Section V, we will make use of this property to analyze the theoretical properties of the proposed framework.

Assumption 3: The safe set $\mathbb{E}_t$ is updated so that the following holds:

$$
(x_{k|t-1}^*, u_{k|t-1}^*), k \in \{0, N-1\} \subseteq \mathbb{E}_t \tag{7}
$$

$$
X_t(r_{t|t-1}^n, \mathbb{E}_t) \subseteq \mathbb{E}_t \quad \forall t \in \{2, \ldots, N\}. \tag{7}
$$

Remark 4: Note that if the computational power is especially limited, then the set $\mathbb{E}_t$ can be recomputed every several time instants since condition (7) trivially holds for $\mathbb{E}_t = \mathbb{E}_{t-1}$. In addition, one can make use of the approach in [9] where a computationally tractable way for online computation of convex sets in a nonconvex environment is proposed.

B. Tracking Cost Function $V_N$

Given a user-defined prediction horizon $N \in \mathbb{N}$, and an admissible setpoint $r^* = (x^*, u^*) \in \mathbb{S}(Z)$, the tracking cost function $V_N$ is defined as follows:

$$
V_N(x_{t:t}, u_{t:t}, r^*) := \sum_{k=0}^{N-1} \ell(x_{k|t}, u_{k|t}, r^*) + V_I(x_{N|t}, r^*) \tag{8}
$$

where $(x_{t|t}, u_{t|t}) \in \mathbb{N}^{N+1}$ is a state and input sequence, $\ell : \mathbb{Z} \times \mathbb{S}(Z) \rightarrow \mathbb{R}_{\geq 0}$ is a user-defined stage cost, and $V_I : \mathbb{R}^n \times \mathbb{S}(Z) \rightarrow \mathbb{R}_{\geq 0}$ is the so-called terminal cost, detailed as follows. The stage cost is designed so that the following assumptions hold.

Assumption 4: Given a compact set $\mathbb{A} \subseteq \mathbb{Z}$, there exist constants $a_1, a_2 \in \mathbb{R}_{>0}$ such that for all $(x, u) \in \mathbb{A}$ and $r^* = (x^*, u^*) \in \mathbb{S}(\mathbb{A})$, the following holds:

$$
a_1\|x - x^*\|^2 \leq \ell^*(x, r^*, \mathbb{A}) \leq a_2\|x - x^*\|^2 \tag{9}
$$

where $\ell^*$ is defined as follows:

$$
\ell^*(x, r^*, \mathbb{A}) := \min_{u'} \ell(x, u, r^*) \quad \text{s.t.} \quad (x, u) \in \mathbb{A}. \tag{10}
$$

Moreover, there exist constants $k_1^b, k_2^b \in \mathbb{R}_{>0}$ such that given two admissible setpoints $r_1^b, r_2^b \in \mathbb{S}(Z)$, and any pair $(x, u) \in \mathbb{Z}$, the following bound holds:

$$
\ell(x, u, r_1^b) \leq k_1^b\ell(x, u, r_2^b) + k_2^b\|r_1^b - r_2^b\|^2. \tag{11}
$$

In [34], it is shown how Assumption 4 is, for example, satisfied when the following standard quadratic stage cost is employed:

$$
\ell(x, u, r^*) = \|x - x^*\|^2_Q + \|u - u^*\|^2_P \tag{12}
$$

where $Q, R > 0$ are matrices of appropriate dimension.

Assumption 5: There exists a constant $\chi \in \mathbb{R}_{>0}$ such that for all $\mathbb{A} \subseteq \mathbb{Z}$, and for all $r^* = (x^*, u^*) \in \mathbb{S}(\mathbb{A})$ then it holds that:

$$
\|x - x^*\|^2 \leq \chi \Rightarrow V_N^*(x, r^*, \mathbb{A}) \leq \gamma \ell^*(x, r^*, \mathbb{A}) \tag{13}
$$

where $\ell^*$ is defined in (10).

Assumption 5 corresponds to a local exponential cost controllability condition (cf. [38] [Ass. 1]), and is trivially satisfied whenever (quadratically bounded) terminal costs can be computed offline (this can be shown since the terminal cost $V_I$ is an upper-bound for the optimal cost function $V_N^*$, and then knowing that $V_I$ can be upper-bounded by the stage cost $\ell^*$ because they are both quadratically bounded). Moreover, this assumption holds if the linearization of the considered system around every admissible setpoint $r^* \in \mathbb{S}(\mathbb{A})$ is stabilizable, using the fact that $r^* \in \text{int}(\mathbb{A})$ [see (5)] and $\ell$ quadratically bounded (Ass. 4).

Terminal Ingredients: Given a set $\mathbb{A} \subseteq \mathbb{Z}$ and a setpoint $r^* \in \mathbb{S}(\mathbb{A})$, the terminal cost $V_I : \mathbb{R}^n \times \mathbb{S}(Z) \rightarrow \mathbb{R}_{\geq 0}$, combined with a so-called terminal controller $\kappa_I : \mathbb{R}^n \times \mathbb{S}(Z) \rightarrow \mathbb{R}^m$, and
a terminal region $X_t(r, A) \subseteq A$ are chosen so that the following condition holds for any $x^* = (x^*, u^*) \in S(A)$, and any $x \in X_t(r^*, A)$:

$$V_t(x^+, r^*) \leq V_t(x, r^*) - \ell(x, \kappa_t(x, r^*), r^*)$$

$$(x, \kappa_t(x, r)) \in A$$

$$x^+ \in X_t(r^*, A)$$

with $x^+ = f(x, u)$.

Note that especially for the case of nonlinear systems, designing a terminal region $X_t$ that depends on a potentially time-varying set $A$ can be especially challenging. For this reason, a more conservative, but easier solution is to employ a terminal equality constraint, i.e.,

$$X_t(r^*, A) := x^*, \quad \kappa_t(x, r^*) := u^* \quad \forall r^* \in S(A).$$

However, depending on the considered problem setup, such a solution might reduce the region of attraction of the MPC scheme (6). In order to overcome this problem, in [34] it is shown how an MPC scheme that employs artificial setpoints can be also used without terminal ingredients. Even though in this article we do not explicitly consider the case without terminal ingredients, we conjecture that the results from [34] can be nevertheless applied in the proposed approach.

C. Offset Cost $T$

In the following, we discuss the technical conditions that the offset cost $T : S(Z) \to \mathbb{R}_{\geq 0}$ must satisfy.

**Assumption 6:** There exist constants $k_0^0, k_1^T \in \mathbb{R}_{> 0}$, and a function $\bar{r} : S(Z) \to \mathbb{R}_{> 0}$, such that for any setpoint $r = (x, u) \in \text{int}(S(Z))$ and any $\tau \in [0, \bar{r}(r)]$, there exists a setpoint $\hat{r} = (\hat{x}, \hat{u}) \in S(Z)$ satisfying

$$||\hat{r} - r|| \leq k_0^0 \tau ||r - r^d(Z)||$$

$$T(\hat{r}) - T(r) \leq -k_1^T \tau ||r - r^d(Z)||$$

where $r^d(Z) := (x^d(Z), u^d(Z))$ is the closest setpoint in the set $Z$, defined as follows:

$$x^d(Z), u^d(Z) := \arg \min_{x, u} T(r)$$

s.t. $(x, u) \in S(Z)$. (15)

Assumption 6 ensures that for any setpoint $r \in \text{int}(S(Z))$, we can find a new setpoint $\hat{r} \in S(Z)$ which is close to $r$ [cf. (14a)] and has a smaller offset cost [cf. (14b)]. It is crucial to note that Assumption 6 only considers setpoints that are strictly contained in the manifold $S(Z)$, and therefore such a property does not need to hold for the setpoints that belong to the boundaries of $S(Z)$. This implies that Assumption 6 can also be satisfied if the manifold $S(Z)$ is nonconnected.

In [34] it is shown how the following offset cost:

$$T(r) = ||y - y^d||_P^p, \quad y = h(r),$$

where $P$ is a positive definite matrix, satisfies Assumption 6 for the case of linear systems. However, satisfaction of Assumption 6 for the general case of nonlinear systems is subject to current research.

Now that the offset cost $T$ is formally defined, the desired control goal with respect to tracking described in Section II can be mathematically reformulated as follows:

$$\lim_{t \to \infty} y_t = h(r^d (Z)).$$

D. Theoretical Analysis of the MPC Scheme (6)

In the following, we show the theoretical analysis of the MPC scheme (6), where we especially focus on the effect of $\epsilon$ on the closed-loop, and on the case of nonconnected manifold $S(Z)$. Even though this represents already an extension compared to existing works ([28], [29]), it is only shown here with the purpose to have a fair comparison between the proposed approach in Section IV, and the one in (6).

**Definition 1:** Given a set $A \subseteq Z$, a prediction horizon $N$, and a scalar $\epsilon \in \mathbb{R}_{> 0}$, we say that a setpoint $r^*_1 \in S(A)$ is a transitory setpoint for the MPC scheme (6) if there exists a scalar $\delta \in \mathbb{R}_{> 0}$ such that for all $(x, u) \in A$ with $V_N(x, r^*_1, A) \leq \delta$ there exists an artificial setpoint $r^*_2 \in S(A)$ that satisfies the following:

$$\epsilon V_N(x, r^*_2, A) + T(r^*_2) < V_N(x, r^*_1, A) + T(r^*_1).$$

(16)

We define with $G(\epsilon, N, A)$ the set formed by all the transitory setpoints of the set $A$, related to the scalar $\epsilon$ and to the MPC horizon $N$.

The role that $\epsilon$ plays in Definition 1 is discussed in Section III-E, where we discuss the shortcomings of the scheme (6).

A graphical explanation of Definition 1 is shown in Fig. 4, where, in this case, the setpoint $r^*_1$ is a transitory setpoint, while the obstacle $O_3$ is too large and therefore the setpoint $r^*_3$ is not a transitory setpoint according to Definition 1. It is important to note that Definition 1 only states the sufficient conditions needed for overcoming an obstacle. This implies that it might be possible that the system will nevertheless overcome obstacle $O_3$ and pass from $r^*_3$ to $r^*_2$ even if $r^*_3$ is not a transitory setpoint in the sense of Definition 1.

Based on the definition of transitory setpoint, in the following we propose a Theorem that is based on the result shown in [34], but that additionally includes the case of nonconnected manifold as well as the set update $Z_t$.

**Theorem 1:** Let Assumptions 1–6 hold, and suppose that at time $t = 0$ (6) is feasible. Then, the MPC scheme is recursively feasible, and the system converges to a nontransitory setpoint.
\( r_1^{s,*} \) that satisfies 
\[
\begin{align*}
T(r_1^{s,*}) & \leq \lim_{t \to \infty} \inf_{i \in [0, \ell]} e V_N(x_i, r_1^{s,*}, \mathbb{H}(r_1^{s,*})) + T(r_i^{s,*}) \\
& \leq e V_N(x_0, r_1^{s,*}, E_0) + T(r_0^{s,*}).
\end{align*}
\]

Theorem 1 states that the system converges to a setpoint that belongs to the boundaries of the manifold \( S(Z) \), and that is not transitory for the set \( G(e, N, \mathbb{H}(r_1^{s,*})) \). This implies that if the \( Z \) is convex and \( S(Z) \) is connected, then the system converges to the closest reachable setpoint \( r_1^c(Z) \), as defined in (15), which is analogous to the result shown in [34] and [37]. On the contrary, if the manifold \( S(Z) \) is nonconnected, then the system is nevertheless able to incrementally slide on it, but it might converge to a nontransitory setpoint due to the potential presence of obstacles, or to the shape of the manifold itself. The proof of Theorem 1 is omitted since it can be obtained based on the proof of the proposed Theorem 2, shown in Section V.

### E. Shortcomings of the MPC Scheme (6)

Even though the MPC scheme in (6) enjoys some interesting properties, such as convergence and constraint satisfaction, it is also affected by strong drawbacks that limit its applicability to a large class of problem setups. In the following, we list some of the major shortcomings that affect (6).

**(

**Role of \( e \) in the MPC scheme (6)**) Based on (16), it is easy to see that by decreasing the value of \( e \), the MPC scheme (6) manages to overcome larger obstacles. In particular, given a state \( x \) and two feasible setpoints \( r_1^c, r_2^c \in S(A) \), then the minimum value of \( e \) required to go from \( r_1^c \) to \( r_2^c \) decreases as the cost \( V_N(x, r_2^c, A) \) increases, according to the following condition:

\[
V_N(x, r_2^c, A) < \frac{1}{e} T(r_2^c) - T(r_1^c) + V_N(x, r_1^c, A)
\]

where \( T(r_1^c) > T(r_2^c) \), and \( V_N(x, r_2^c, A) > V_N(x, r_1^c, A) \). Note that for limit case of \( e \ll 1 \), we have that it is possible to overcome obstacles as long as \( T(r_2^c) < T(r_1^c) \). However, it is nevertheless important to remark this is only possible if \( V_N(x, r_2^c, A) \) exists, i.e., there must exist a feasible trajectory that starts in the \( \delta \)-neighborhood of the setpoint \( r_1^c \), and stabilizes the setpoint \( r_2^c \). Moreover, note that based on the definition above, it is possible to show that the following holds:

\[
G(e_1, N, A) \subseteq G(e_2, N, A)
\]

for all \( 0 < e_2 < e_1 \). This implies that by reducing \( e \) it is possible to escape more local minima, compare Theorem 1.

Even though choosing a very small value of \( e \) is beneficial to overcome larger obstacles, it is easy to see that the cost function considered in the MPC scheme (6) behaves as follows:

\[
e \ll 1 \Rightarrow e V_N(x, u, r^s) + T(r^s) \approx T(r^s).
\]

This means that, for \( e \ll 1 \), the MPC scheme (6) only tries to bring the output \( y_t \) of the system as close as possible to the desired output \( y^d \), while marginally (or not at all for \( e \to 0 \)) considering the tracking cost \( V_N \). Hence, since the transitory behavior of the system will not be optimized, the MPC scheme might make full use of the entire constraint set. Even though this might sound beneficial, note that this is undesired in several practical applications. For example, consider an autonomous vehicle controlled with the MPC scheme (6) with an \( e \ll 1 \). Then, the vehicle might travel at its maximum speed and acceleration, without considering fuel consumption, passenger comforts, nor the safety perceived by the surrounding systems, with the sole goal to get to the output \( y^d \) as quick as possible. For this reason, in order to ensure that the system behaves according to a user-defined tracking cost function \( V_N \), the value of \( e \) cannot be chosen arbitrarily small, but must be appropriately tuned in order to find a tradeoff between the maximum obstacle that can be overcome according to (17), and the desired behavior obtained with \( e = 1 \), as discussed above.

**Lack of Active Exploration:** The MPC scheme does not actively incentivize exploration of the unknown and potentially unsafe space \( Z \setminus E_t \) since the open loop trajectory must entirely lie in the safe set \( E_t \), as enforced in condition (6c). This implies that, as also motivated in the motivating example in Section II-A, and shown in Fig. 3, there might exist safe trajectories with higher performance that will never be explored. The lack of active exploration is critical in several applications where the surrounding is not only unknown, but potentially also changing over time.

### IV. PROPOSED MPC FRAMEWORK

In order to overcome the shortcomings resulting from the MPC scheme (6) and discussed in Section III-E, in the following, we propose a novel MPC framework which differs from the one in (6) as follows.

1. **Overall Idea:** Instead of optimizing only over one trajectory, the proposed MPC scheme optimizes over a so-called **learning trajectory** \((x_{i+1}^{L}, u_{i+1}^{L}, r_{i+1}^{L})\), and a so-called **backup trajectory** \((x_{i+1}^{B}, u_{i+1}^{B}, r_{i+1}^{B})\), see Fig. 5. At each time instant \( t \in \mathbb{N}_{>0} \), both the trajectories share the current state \( x_t \), and the first predicted input \( u_{t+1}^{L} = u_{t+1}^{B} \) (therefore, the predicted state \( x_{t+1}^{L} = x_{t+1}^{B} \) as well). The learning trajectory, denoted with the apex \( L \), only satisfies the estimated constraint set \( Z_{E_t} \), and can therefore be unsafe since it might violate the actual constraint set \( Z \) (cf. Assumption 2). On the contrary, the backup trajectory, denoted with the apex \( B \), lies in the safe constraint set \( E_t \subseteq Z \). In this way, we have that the learning trajectory is left free to explore the potentially unsafe set \( Z_{E_t} \), with the goal to find a trajectory with an improved open-loop performance, while the backup trajectory has the role to ensure safety as well as recursive feasibility of the MPC scheme.
2) Exploration-Based Cost Function: Given the considered learning and backup trajectories, by choosing the following cost function:

\[ V_N(x_t^L, u_t^B, r_t^L) + T(r_t^L) + V_N(x_{t+1}^L, u_{t+1}^B, r_{t+1}^L) + T(r_{t+1}^L). \] (18)

the first predicted state \( x_{t+1}^L = x_{t+1}^B \) would be computed based on a tradeoff between the learning and backup cost functions. However, in order to make sure that, in closed-loop, the system safely explores unknown areas with the aim to get closer to the desired output \( y^d \), the ideal cost function should only minimize the learning trajectory, and neglect the backup one, i.e.,

\[ V_N(x_t^L, u_t^L, r_t^L) + T(r_t^L). \] (19a)

In this way, the first predicted state \( x_{t+1}^L = x_{t+1}^B \) is not computed based on a tradeoff between the two trajectories, but it is only influenced by the learning cost. In this case, the backup trajectory would not be optimized and simply results as an arbitrary feasible trajectory that lies in the set \( E_t \). Therefore, in order to ensure that also the backup trajectory points toward the desired output \( y^d \), we propose the following modification of the cost function in (18):

\[ V_N(x_t^L, u_t^L, r_t^L) + T(r_t^L) + \epsilonT(r_t^B). \] (19b)

where \( \epsilon \in \mathbb{R}_{>0} \), with \( \epsilon \ll 1 \)

Proposed MPC scheme: At each time \( t \in \mathbb{N}_0 \), given the state \( x_t \), the safe set \( E_t \), the estimated constraint set \( Z_t \), the user-defined values of \( \epsilon \ll 1 \), and \( S_0 \geq 0 \), the following optimization problem is solved:

\[
\begin{align*}
\min_{x_t^L, u_t^L, x_{t+1}^L, u_{t+1}^L, r_t^L, r_{t+1}^L} & \quad V_N(x_t^L, u_t^L, r_t^L) + T(r_t^L) + \epsilon T(r_t^B) \\
\text{s.t.} & \quad x_{t+1}^B = f(x_{t+1}^B, u_{t+1}^B) \\
& \quad \text{Backup traj.:} \quad (x_{t+1}^B, u_{t+1}^B) \in E_t \\
& \quad x_{t+1}^B \in X_t(x_t^B, \mathbb{E}_t), \quad r_t^B \in S(\mathbb{E}_t) \\
& \quad \text{Learning traj.:} \quad (x_t^L, u_t^L) \in \mathbb{Z}_t \\
& \quad x_t^L \in X_t(x_t^L, \mathbb{Z}_t), \quad r_t^L \in S(\mathbb{Z}_t) \\
& \quad \text{Initial cond.:} \quad x_0^L = x_0^B = x_t \\
& \quad \text{Safety:} \quad u_0^B = u_0^L \\
& \quad \text{Convergence:} \quad V_N(x_t^B, u_t^B, r_t^B) + T(r_t^B) \leq S_t + \hat{F}_t \\
& \quad k = 0, \ldots, N - 1
\end{align*}
\] (19c)

where condition (19j) together with the variables \( \hat{F}_t \) and \( S_t \) are explained in the following.

The optimal open-loop learning and backup trajectories obtained from (19) are denoted with

\[
\begin{align*}
& x_t^L, u_t^L, r_t^L \quad \text{and} \\
& x_t^B, u_t^B, r_t^B \quad \text{where the resulting closed-loop is given as follows:}
\end{align*}
\]

\[ x_{t+1} = f(x_t, u_t), \quad u_t := u_{t+1}^B = u_{t+1}^L. \]

Note that conditions (19b)–(19h) are analogous to the constraints used in the MPC scheme (6) but applied for the backup and learning trajectories. Condition (19i) is included to ensure safety by forcing the system to remain in the safe set \( E_t \subset \mathbb{Z}_t \).

Closed-loop exploration is then obtained thanks to the considered cost function, which will lead the system toward the unknown set \( \mathbb{Z}_t \setminus E_t \) if this improves the performance.

Condition (19j): In order to ensure convergence, we enforce, at each time step \( t \in \mathbb{N}_0 \), a decrease of the backup cost function. As shown in Fig. 6, we have that when (19j) is not active, the system in closed-loop follows the learning trajectory (which penalizes the tracking and offset costs), leading toward \( r_t^B \).

The purpose of (19j) is therefore to ensure that when the system converges to a local minimum \( r_t^L \), then the system is eventually forced to instead follow the backup trajectory (which only penalizes the offset cost), leading it toward the closer setpoint \( r_t^B \).

For the sake of notational simplicity, we introduce the variable \( \hat{F}_t^* \) as follows:

\[ \hat{F}_t^* := V_N(x_t^L, u_t^L, r_t^B) + T(r_t^B). \]

(20)

Similarly, we define with \( \hat{F}_{t+1}^* \) an upper-bound for the optimal value function \( \hat{F}_{t+1}^* \) computed at time \( t \), and based on the optimal value function \( F_t^* \), i.e.,

\[ \hat{F}_{t+1}^* := F_t^* - \ell(x_t^B, u_t^B, r_t^B). \]

(21)

The value of \( S_0 \in \mathbb{R}_{>0} \) is a user-defined parameter that is updated as follows for all \( t \in \mathbb{N}_0 \):

\[ S_{t+1} = S_t - F_t^* + \hat{F}_t. \]

(22)

Note that \( S_t \) can be interpreted as a storage function. In particular, its value can be understood as a profit that is obtained when the backup cost \( F_t^* \) decreases more than expected, i.e., if the exploration of new areas was successful and led to a better...
cost $F^*_t$ compared to the expected one $\hat{F}_t$. On the contrary, if the backup cost $F^*_t$ is higher than expected ($\hat{F}_t$), then the value of $S_{t+1}$ decreases. Eventually, especially if the system gets stuck in a local minimum, the value of $S_t$ keeps decreasing until (19j) is active, forcing the system to follow the backup trajectory, related to $\hat{F}_t$. We want to clarify that, while the learning trajectory weights the tracking cost $V_N$ and the offset cost $T$, the backup trajectory only weighs the offset cost $T$, and therefore it is able to overcome (some) local minima, see Fig. 6. A condition similar to (19j) is also employed in [8] and [39], even though the two approaches consider a different problem setup, and different goals.

Initial Feasibility of (19): Given the fact that the value $\hat{F}_0$ cannot be defined at time $t = 0$, then, to ensure initial feasibility of the MPC scheme (19), the value of $\hat{F}_0$ can be temporarily chosen large enough so that (19j) holds. Then, at time $t = 1$, the value of $\hat{F}_0$ is reset with $\hat{F}_0 = F^*_0$, which also leads to $S_1 = S_0$. Moreover, we want to highlight that the proposed MPC scheme (19) has the same region of attraction of the MPC scheme (6) since the open-loop learning trajectory can always be chosen as the backup one, ensuring satisfaction of the added conditions. Recursive feasibility is given in Theorem 2 and then shown in the theoretical analysis in Section V.

A. Motivating Example for Condition (19j)

In the following, we show through a motivating example, depicted in Fig. 7, why condition (19j) is needed to ensure convergence. For the sake of simplicity, we consider a simplified version of the example discussed in Section II-A. By considering such a simple example, it is possible to clearly illustrate both the problems arising by the lack of condition (19j), as well as the effect that its presence has. The considered constraint set $\mathcal{Z}$ is a nonconvex subset of the set of natural numbers $\mathbb{N}^{n+m}$, which, therefore, contains only a finite number of points and hence the optimal trajectories can be easily computed.

Consider the following 1-D system

$$x_{t+1} = u_t$$

with an MPC horizon $N = 3$, initial state $x_0 = 2$, and desired setpoint $d^d = (x^d, u^d) = (0, 0)$. The sets $\mathcal{E}_t$ and $\mathcal{Z}_t$ are defined as follows:

$$\mathcal{E}_t := \{(2, 2), (2, 0), (0, 0)\}$$

$$\mathcal{Z}_t := \{(2, 2), (2, 1), (2, 0), (1, 0), (0, 0)\}$$

which means that the state $x = 1$ is potentially unsafe, and therefore it belongs to the set $\mathcal{Z}_t$ and does not belong to the set $\mathcal{E}_t$. For the sake of simplicity, we enforce terminal equality constraint w.r.t. the terminal region $\mathcal{Y}_t := \{0\}$, and we do not consider the offset cost $T$. The stage cost $\ell(x, u)$ and the terminal cost $V_f$ are as follows:

$$\ell(2, 2) = 2, \quad \ell(2, 1) = 1, \quad \ell(2, 0) = 10$$

$$\ell(1, 0) = 1, \quad \ell(0, 0) = 0, \quad V_f(0, 0) = 0.$$

In the following, we compare the case where condition (19j) is not included, and the case where it is included in the MPC scheme (19). We initialize $S_0 = 5$ and, as discussed before, we temporarily set $\hat{F}_0$ large enough to ensure that condition (19j) is not active at time $t = 0$. We recall that $\hat{F}_0$ is then reset as $\hat{F}_0 = F^*_0$ at time $t = 1$. The optimal trajectories resulting from the MPC scheme (19) with or without condition (19j) are equivalent

$$x_0 = x_{L_0} = x_{B_0} = 2$$

$$x_{10} = x_{10} = 2,$$

$$x_{20} = 0, \quad x_{20} = 1,$$

$$x_{30} = 0, \quad x_{30} = 0$$

which leads to $V_N(x_{L_0}, u_{L_0}, r^d) = 4$, and $V_N(x_{B_0}, u_{B_0}, r^d) = 12$.

1) Without Condition (19j): At time $t = 1$, we have that $x_1 = x_{B_10} = x_0$. This implies that, since the trajectory (23) remains feasible at time $t + 1$ as well, the system evolves in closed loop as follows:

$$x_t = 2, \quad \forall t \in \mathbb{Z}_{\geq 0},$$

Therefore, even in such a simple scenario, and even in the presence of a safe trajectory, the system does not converge to the desired state $x^d = 0$. In addition, we want to remark that even by arbitrarily increasing the prediction horizon $N \geq 3$, then such a problem is not solved and the closed loop trajectory does not change.

2) With (19j): At time $t = 0$, starting from the user-defined $S_0 = 5$, and since the value of $\hat{F}_0$ is reset to $F^*_0$, we have that $S_1 = S_0 = 5$. At time $t = 1$, based on the optimal trajectory shown in (23), we have that, according to (21), $\hat{F}_1$ is defined as follows:

$$\hat{F}_1 = V_N \left( x_{B_10}, u_{B_10} \right) - \ell \left( x_{B_10}, u_{B_10} \right) = 11$$

which implies that

$$S_2 \stackrel{(22)}{=} 5 - 12 + 11 = 4.$$

By iterating the reasoning above, we have that as long as the system remains at $x = 2$, the value of $S_t \geq 1$ is updated as $S_t = S_{t-1} - (t - 1), \ t \in \mathbb{Z}_{\geq 1}$. Therefore, regardless of the initial value of $S_0 \in \mathbb{Z}_{\geq 0}$, there will be a time instant $t^* \in \mathbb{Z}_{\geq 0}$ where the value of $S_t$ is too low that the optimal solution shown in (23) is not feasible anymore. This means that at time $t^*$ the only feasible trajectory is the following safe trajectory:

$$x_{t^*} = x_{L_{t^*}} = x_{B_{t^*}} = 2$$

$$x_{1|t^*} = x_{1|t^*} = 0.$$
Transitory setpoint \((r_{1}^{T})\) for the MPC scheme (19), compare Definition 2. Obstacles are referred with \(\mathcal{O}_{i}, i = 1, 2, 3\). In contrast to Fig. 4, the two trajectories share the current state \(x\) and the first predicted state \(x^{+}\).

\[
x_{2|i}^{B} = 0, \quad x_{2|i}^{L} = 0
\]

which also implies \(x_{t} = 0\) for all \(t \geq t^{*} + 1\).

In conclusion, with such an example, we showed that even for simple problem setups, and for arbitrarily large prediction horizons \(N \geq 3\), an MPC scheme that employs two trajectories without additionally including constraint \((19j)\) does not ensure convergence to the desired setpoint.

Remark 5: Note that the bound \(\bar{e} > 0\) is needed to ensure that the backup trajectory points toward the desired output \(y^{d}\), as shown in Fig. 6. Indeed, for \(\bar{e} = 0\), the backup trajectory resulting from the proposed MPC scheme \((19)\) would simply be a trajectory that does not minimize any cost, and could potentially point toward the opposite direction of the desired goal \(y^{d}\).

Assumption 7: The safe set \(\mathcal{E}_{t}\) is updated at each time instant \(t \in \mathbb{I}_{\geq 0}\) so that the following holds:

\[
\left(x_{i|t}^{B}, u_{i|t}^{B}\right) \in \mathcal{E}_{t+1}, \quad \forall k \in [1, \ldots, N - 1]
\]

\[
\left(x_{N|t}^{B}, v_{t}^{B}, x_{t+1|t}^{B}\right) \in \mathcal{E}_{t+1}
\]

\[
x_{t}\left(x_{t}^{B}, \mathcal{E}_{t}\right) \subseteq \mathcal{E}_{t+1}.
\]

Condition (25) is trivially satisfied if the set \(\mathcal{E}_{t+1}\) is chosen so that \(\mathcal{E}_{t} \subseteq \mathcal{E}_{t-1}\). Note that the estimated constraint set \(\mathcal{Z}_{t}\) is updated so that condition (4) (i.e., \(\mathcal{E}_{t} \subseteq \mathcal{Z}_{t}\)) holds also with the new safe set \(\mathcal{E}_{t}\). In the following, we modify Definition 1 in order to take into account not only the different cost function considered in the proposed MPC scheme \((19)\) compared to the scheme \((6)\), but also to take into account the fact that, due to constraint \((19)\), the two trajectories share the current state \(x\) and the first predicted state \(x^{+}\), see Fig. 8.

Definition 2: Consider a set \(\mathcal{A} \subseteq \mathbb{Z}\), we say that an artificial setpoint \(r_{1}^{T} \in \mathcal{S}(\mathcal{A})\) is a transitory setpoint for the MPC scheme \((19)\), if there exists a scalar \(\delta \in \mathbb{R}_{\geq 0}\) such that for all \((x, u) \in \mathcal{A} \cap \mathcal{W}_{N}(x, r_{1}^{T}, \mathcal{A})\) with \(V_{N}(x, r_{1}^{T}, \mathcal{A}) \leq \delta\) there exists an artificial setpoint \(r_{2}^{T} \in \mathcal{S}(\mathcal{A})\) with \(T(r_{2}^{T}) < T(r_{1}^{T})\) that satisfies the following:

\[
\epsilon \left[\ell(x, u, r_{2}^{*}) + V_{N-1}(x^{+}, r_{2}^{*}, \mathcal{A})\right] + T(r_{2}^{*})
\]

where \(x^{+} = f(x, u)\). We define with \(\mathcal{M}(\epsilon, N, \mathcal{A})\) the set constructed by all the transitory setpoints related to the set \(\mathcal{A}\), the scalar \(\epsilon \in \mathbb{R}_{>0}\), and the MPC horizon \(N\).

In Fig. 8, we give a graphical representation of Definition 2.

Lemma 1: Based on Definition 2, it holds that

\[
\mathcal{M}(\epsilon, 1, \mathcal{A}) \subseteq \mathcal{M}(\epsilon, 2, \mathcal{A})
\]

for all \(0 < \epsilon_{2} < \epsilon_{1}\).

Proof: Assuming that (26) holds for \(\epsilon_{1}\), then we have

\[
T(r_{1}^{T}) - T(r_{2}^{T}) > \epsilon_{1} \left[\ell(x, u, r_{2}^{*}) + V_{N-1}(x^{+}, r_{2}^{*}, \mathcal{A}) - V_{N}(x, r_{1}^{T}, \mathcal{A})\right]
\]

\[
\epsilon_{1}^{\geq} > \epsilon_{2} \left[\ell(x, u, r_{2}^{*}) + V_{N-1}(x^{+}, r_{2}^{*}, \mathcal{A}) - V_{N}(x, r_{1}^{T}, \mathcal{A})\right].
\]

Proposition 1: Consider a set \(\mathcal{A} \subseteq \mathbb{Z}\), let Assumptions 4–6 hold. If \(r \in \text{int}(\mathcal{S}(\mathcal{A}))\), then \(r\) is a transitory setpoint for the MPC scheme \((19)\), i.e., \(r \in \mathcal{M}(\epsilon, N, \mathcal{A})\), for any \(\epsilon \in \mathbb{R}_{>0}\).

The proposition above directly implies that a nontransitory setpoint \(r\) must belong to the boundaries of the manifold \(\mathcal{S}(\mathcal{A})\). The proof of Proposition 1 is shown in the Section V-A.

Role of \(\bar{e}\) in the MPC scheme \((19)\): Differently from the MPC scheme \((6)\), in the proposed MPC framework, the value of \(\bar{e}\) is chosen arbitrarily small without yielding undesired effects. In particular, for \(\bar{e} \rightarrow 0\), then the cost function optimized in the MPC scheme \((19)\) approaches the ideal cost function discussed in \((18)\), i.e.,

\[
\bar{e} \ll 1 \Rightarrow V_{N}\left(x_{i|t}^{B}, u_{i|t}^{L}, r_{t}^{sL}\right) + T\left(r_{t}^{sL}\right) + \bar{e}T\left(r_{t}^{sB}\right)
\]

This implies that the closed-loop does not result from a tradeoff between the learning and backup trajectory, but it focuses on the learning trajectory, leading to a safe exploration.

Remark 6: Note that by increasing the prediction horizon \(N\), the region of attraction of the MPC scheme \((19)\) as well as the set \(\mathcal{M}(\epsilon, N, H(r))\) tend to expand. However, it is important to note that the value of \(N\) cannot be arbitrarily increased due to the limited computational power of the chosen hardware.

Theorem 2: Let Assumptions 1–7 hold, and suppose that at time \(t = 0\) \((19)\) is feasible. Then, the MPC scheme is recursively feasible for all \(t \in \mathbb{I}_{\geq 0}\), and the following holds:

\[
\lim_{t \to \infty} z_{t} = r_{\infty}^{sB}
\]

where \(r_{\infty}^{sB}\) satisfies the following:

\[
r_{\infty}^{sB} \in \text{bound}(\mathcal{S}(\mathbb{Z})) \setminus \mathcal{M}(\epsilon, N, \mathbb{H}(r_{\infty}^{sB}))
\]

and

\[
T(r_{\infty}^{sB}) \leq S_{0} + \hat{F}_{0} - \lim_{t \to \infty} \sum_{i=0}^{t-1} \ell\left(x_{i}^{B}, u_{i}^{B}, r_{i}^{sB}\right).
\]

Escape Transitory Setpoints (e.g., Local Minima): The formal benefit resulting form Theorem 2 is that since the backup cost function in \((19a)\) only penalizes the offset cost, then the system can escape all the transitory setpoints (and local minima) that
belong to the set $M(\varepsilon, N, \mathbb{H}(r_{x^*}^n))$, which is not possible by using the MPC scheme (6). In addition, we want to remark that the proposed MPC scheme (19) enables the system to explore unknown areas, which represents one of the main contributions of the algorithm.

The proof of Theorem 2 is shown in Section V.

V. THEORETICAL ANALYSIS

In this section, we show the proof of Proposition 1 in Section V-A, and the proof of Theorem 2 in Section V-B1 and V-B2, where we first discuss recursive feasibility and then convergence, respectively.

A. Proof of Proposition 1

In order to prove Proposition 1, in the following, we show that given an artificial setpoint $r^* = (x^*, u^*) \in \text{int}(\mathbb{S}(\hat{A}))$, there exists a setpoint $\hat{r}^* = (\hat{x}^*, \hat{u}^*) \in S(\hat{A})$ that satisfies
\[
\begin{align*}
\mathcal{E} \left[ \ell(x, u, \hat{r}^*) + V_{N-1}(x^+, \hat{r}^*, \hat{A}) \right] + T(\hat{r}^*) < eV_{N}^*(x, r^*, \hat{A}) + T(r^*)
\end{align*}
\]
which will imply that the setpoint $r^*$ is a transitory setpoint according to Definition 2, and will conclude the proof.

Given $r^* = (x^*, u^*) \in \text{int}(\mathbb{S}(\hat{A}))$, we know that based on Assumption 6 there exists a different artificial setpoint $\hat{r}^* = (\hat{x}^*, \hat{u}^*) \in S(\hat{A})$ that satisfies:
\[
\begin{align*}
&\|\hat{r}^* - r^*\| \leq k_0^T \tau\|r^* - r^d(\hat{A})\| \\
&\mathcal{T}(\hat{r}) - \mathcal{T}(r) \leq -k_1^T \tau\|r^* - r^d(\hat{A})\|
\end{align*}
\]
with a later specified value of $\tau \in (0, \bar{\tau}(r^*))$.

In the following, we show that for a small enough $\tau$, there exists a feasible trajectory that stabilizes the artificial setpoint $\hat{r}^*$ and that has a smaller cost compared to the trajectory that stabilizes the artificial setpoint $r^*$. Therefore, this results in a contradiction, and implies that the setpoint $r^*$ is a transitory setpoint in the sense of Definition 2.

Existence of a Feasible Solution for $\hat{r}^*$: In order to ensure that the MPC problem is feasible, we focus on the following neighborhood of the artificial setpoint $r^* = (x^*, u^*)$:
\[
\|x - x^*\|^2 \leq \chi
\]
which allows us to make use of Assumption 5 to ensure the existence of a solution that stabilizes $r^*$.

Since we assume that $r^* \neq r^d(\hat{A})$, then based on the continuity argument in Assumption 5, it is always possible to upper bound the optimal cost function as follows:
\[
V_{N}^*(x, r^*, \hat{A}) \leq \bar{a}\|r^* - r^d(\hat{A})\|^2
\]
with a later specified value of $\bar{a} \in \mathbb{R}_{> 0}$. Based on the properties of the stage cost, we can construct the following upper bound on the optimal cost function:
\[
\begin{align*}
&\|x^+ - \hat{x}^*\|^2 = \|x^+ - x^* + x^* - \hat{x}^*\|^2 \\
&\leq \|x^+ - x^*\|^2 + \|x^* - \hat{x}^*\|^2 + 2\|x^+ - x^*\|\|x^* - \hat{x}^*\| \\
&\leq \frac{1}{a_1} \ell^*(x^+, r^*, \hat{A}) + \|x^* - \hat{x}^*\|^2
\end{align*}
\]
where we used the fact that $\|x^* - \hat{x}^*\| \leq \|r^* - \hat{r}^*\|$, and that
\[
\ell^*(x^+, r^*, \hat{A}) \leq \frac{1}{a_1} \ell(x, r^*, \hat{A}) + k^T_0 \tau^2\|r^* - r^d(\hat{A})\|^2
\]
and that
\[
\ell^*(x^+, r^*, \hat{A}) \leq \frac{1}{a_1} \ell(x, r^*, \hat{A}) + k^T_0 \tau^2\|r^* - r^d(\hat{A})\|^2
\]

First, we analyze each unknown term of (34) individually, and then we combine the obtained bounds to show satisfaction of (34).

1) Stage Cost $\ell(x, u, \hat{r}^*)$:
\[
\ell(x, u, \hat{r}^*) \leq \frac{1}{a_1} \ell(x, r^*, \hat{A}) + k^T_0 \tau^2\|r^* - \hat{r}^*\|^2
\]
with a later specified value of $\bar{a} \in \mathbb{R}_{> 0}$. Based on the properties of the stage cost, we can construct the following upper bound:
\[
\begin{align*}
&\|x^+ - \hat{x}^*\|^2 = \|x^+ - x^* + x^* - \hat{x}^*\|^2 \\
&\leq \|x^+ - x^*\|^2 + \|x^* - \hat{x}^*\|^2 + 2\|x^+ - x^*\|\|x^* - \hat{x}^*\| \\
&\leq \frac{1}{a_1} \ell^*(x^+, r^*, \hat{A}) + \|x^* - \hat{x}^*\|^2
\end{align*}
\]

2) Optimal Cost Function $V_{N-1}(x^+, \hat{r}^*, \hat{A})$:
\[
V_{N-1}(x^+, \hat{r}^*, \hat{A}) \leq \gamma \ell^*(x^+, \hat{r}^*, \hat{A})
\]
where we used the fact that $\ell(x, u, r^*) \leq V_{N}^*(x, r^*, \hat{A})$, which holds because $\ell(x, u, r^*)$ is related to the optimal trajectory related to the cost $V_{N}^*(x, r^*, \hat{A})$.

3) Offset Costs $T(\hat{r}^*) - T(r^*)$: As shown in (30b), it holds that
\[
T(\hat{r}) - T(r) \leq -k_1^T \tau\|r^* - r^d(\hat{A})\|
\]
Finally, we can choose \( \bar{a} \) and \( \tau \) small enough, and combine (30b), (35), and (36), to obtain the following bound:

\[
\epsilon \left[ \ell(x, u, \hat{r}^s) + V^s_{N-1}(x^+, \hat{r}^s, \bar{A}) \right] - \epsilon V^s_N(x, r^s, \bar{A}) + T(\hat{r}^s) - T(r^s) \\
\leq 0
\]

where \( c \in \mathbb{R} \) is defined as follows:

\[
c := c_{\gamma}a_2 + c_k^2 \bar{a} + \left( \frac{2c_{\gamma}a_2k_0^T}{a_1} - k_1^2 \right) \tau
\]

Thanks to the linear term \(-k_1^2 \tau\), it is possible to choose \( \bar{a} \) small enough to ensure the existence of a \( \tau \in (0, \bar{\tau}(r^s))] \) that renders \( c < 0 \), while satisfying (30a) and (30b). Finally, based on the fact that \( c < 0 \), we have that the artificial setpoint \( r^s \) is a transitory setpoint according to Definition 2 since there exists another setpoint \( \hat{r}^s \) that satisfies (30a) and (30b) as well as the following:

\[
\epsilon \left[ \ell(x, u, \hat{r}^s) + V^s_{N-1}(x^+, \hat{r}^s, \bar{A}) \right] + T(\hat{r}^s) < \epsilon V^s_N(x, r^s, \bar{A}) + T(r^s).
\]

\( \square \)

B. Proof of Theorem 2

1) Recursive Feasibility of the MPC Problem (19):

Given the optimal backup trajectory \((\hat{x}^B_{|t+1}, \hat{u}^B_{|t+1}, \hat{r}^s_{|t+1})\), the candidate learning trajectory \((\hat{x}^L_{|t+1}, \hat{u}^L_{|t+1}, \hat{r}^s_{|t+1})\), and the candidate learning trajectory \((\hat{x}^B_{|t+1}, \hat{u}^B_{|t+1}, \hat{r}^s_{|t+1})\), we define as follows:

\[
\hat{x}^s_{|t+1} = \hat{x}^s_{|t} := \hat{r}^s_{|t+1}
\]

\[
\hat{x}^s_{|0} = \hat{x}^s_{|0} := x_{|t+1}
\]

\[
\hat{x}^B_{|t+1} = \hat{x}^B_{|t} = k, \quad k = 1, \ldots, N
\]

\[
\hat{u}^B_{|t+1} = \hat{u}^B_{|t} = k, \quad k = 1, \ldots, N - 1
\]

\[
\hat{u}^B_{|0} = \hat{u}^B_{|0} := \kappa \left( \hat{x}^B_{|t+1}, \hat{r}^s_{|t+1} \right)
\]

\[
\hat{x}^B_{|N+1} = \hat{x}^B_{|t+1} := f \left( \hat{x}^B_{|N+1}, \hat{u}^B_{|N+1} \right)
\]

Note that due to the potentially changing estimated constraint set \( Z_i \), it is not possible to ensure that the optimal learning trajectory \((\hat{x}^L_{|t}, \hat{u}^L_{|t}, \hat{r}^s_{|t})\), computed at time \( t \) can be employed for constructing the candidate learning trajectory at time \( t + 1 \). For this reason, both the learning and backup trajectories are defined based on the optimal backup trajectory.

In the following, we analyze how the candidate trajectories in (38) satisfy the constraints (19c), (19d), (19f), (19g), and (19j). Note that the remaining constraints in (19) are trivially satisfied, and therefore will not be discussed.

\( \text{Satisfaction of (19c) and (19j):} \) First, based on the candidate trajectory defined above, and thanks to the terminal controller \( \kappa_L \), introduced in (13), we have that

\[
\left( \hat{x}^B_{|N+1}, \hat{u}^B_{|N+1} \right) = \left( \hat{x}^B_{|N+1}, \hat{u}^B_{|N+1} \right) \in E_{t+1}.
\]

\( \text{Satisfaction of (19d) and (19g):} \) Satisfaction of these constraints is ensured based on the properties of the terminal region \( X_t \) shown in (13), on the candidate trajectories defined in (38), and on how the safe set \( E_t \) is updated, cf. (25). More details on how the conditions above are satisfied can be found in [1].

\( \text{Satisfaction of (19j):} \) First, note that based on satisfaction of (19) up to time \( t \), and since we start with \( S_0 \geq 0 \), as discussed above (22), we have that the following holds:

\[
S_t - F^s_t + \hat{F}_t \geq 0 \quad \forall t \in I_{\geq 0}.
\]

Considering the candidate backup trajectory defined in (38), we have that

\[
V_N \left( \hat{x}^B_{|t+1}, \hat{u}^B_{|t+1}, \hat{r}^s_{|t+1} \right) + T(\hat{r}^s_{|t+1}) \leq V_N \left( x^B_{|t}, u^B_{|t}, r^s_{|t} \right) + T(\hat{r}^s_{|t}) - \ell \left( x^B_{|t}, u^B_{|t}, r^s_{|t} \right) \leq F^s_t - \ell \left( x^B_{|t}, u^B_{|t}, r^s_{|t} \right) \leq F^s_{t+1}
\]

which ensures satisfaction of (19j) at time \( t + 1 \) as well.

2) Convergence [Satisfaction of (27)–(29)]: For the sake of simplicity, instead of defining \( S_t \) recursively as done in (22), we reformulate \( S_t \) as follows:

\[
S_{t+1} = S_t + \hat{F}_t - F^s_t = S_0 + \sum_{i=0}^t \hat{F}_i - F^s_i
\]

\[
\Rightarrow S_{t+1} = S_0 + \hat{F}_0 - \hat{F}_t - \sum_{i=0}^t \hat{F}_i - F^s_i \leq S_0 + \hat{F}_0.
\]

Therefore, based on (19j) and on (41), the following holds:

\[
F^s_{t+1} \leq S_{t+1} + \hat{F}_t - F^s_{t+1}
\]

\[
F^s_{t+1} = S_0 + \hat{F}_0 - \sum_{i=0}^t \ell \left( x^B_{i}, u^B_{i}, r^s_{i} \right).
\]
By rearranging (42), and taking the infinite average sum of both sides, we have

\[
\lim_{T \to \infty} \frac{F_{T+1}^{*} - \tilde{F}_0 + \sum_{t=0}^{T} \ell \left( x_{t}^{B,s}, u_{t}^{B,s}, r_{t}^{B,s} \right)}{T} \leq \lim_{T \to \infty} \sup_{t} S_{0,10} = 0 \tag{42}
\]

which shows satisfaction of (27). The result above holds since \( F_{t}^{*} \) is lower and upper bounded for all \( t \in \mathbb{I}_{\geq 0} \). In particular, the lower bound for \( F_{t}^{*} \) (i.e., \( F_{t}^{*} \geq 0 \)) can be trivially ensured by construction of the cost function in (8), and thanks to the lower and upper bounds of the stage costs discussed in Assumption 4, while the upper-bound is shown in (42), based on the fact that both \( S_{0} \) and \( \tilde{F}_0 \) are bounded. Based on both (19j) and on (43), then satisfaction of (29) can be shown as follows:

\[
\lim_{t \to \infty} V_{N} \left( x_{0}, u_{0}^{*}, r_{0}^{B,s} \right) + T \left( r_{0}^{B,s} \right) \leq \lim_{t \to \infty} \sum_{i=0}^{t-1} \ell \left( x_{i}^{B,s}, u_{i}^{B,s}, r_{i}^{B,s} \right) - \lim_{t \to \infty} S_{0,10} = 0 \tag{43}
\]

Note that in (43) we only showed that the system converges to a feasible artificial setpoint \( r_{0}^{B,s} \in \mathbb{S}(\mathbb{Z}) \), i.e.,

\[
\lim_{t \to \infty} z_{t} = r_{0}^{B,s} \in \mathbb{S}(\mathbb{Z}), \quad z_{t} = (x_{t}, u_{t}) \tag{44}
\]

However, in order to show satisfaction of (28), we additionally need to show that \( r_{0}^{B,s} \notin \mathcal{M}(\epsilon, N, \mathbb{H}(r_{0}^{B,s})) \), and \( r_{0}^{B,s} \in \mathbb{S}(\mathbb{Z}) \).

The following part will be done by contradiction and therefore we assume that the system converges to a transitory setpoint \( r_{0}^{B,s} \in \mathcal{M}(\epsilon, N, \mathbb{H}(r_{0}^{B,s})) \). In particular, in the following proof, we do not modify the learning trajectory, while we only show that there exists a different backup trajectory that has the same first predicted state \( x_{1t}^{B,s} = x_{1t} \) [to ensure (19i)], and is feasible for the MPC scheme (19), and leads to a decrease in the overall cost function. This means that the original MPC solution is not the optimal one, which results in a contradiction.

First, based on (43), we know that there exists a time \( t_{\delta} \in \mathbb{I}_{\geq 0} \) such that

\[
V_{N} \left( x_{0}, u_{0}^{*}, r_{0}^{B,s} \right) - \delta \leq \ell \left( x_{t}^{B,s}, u_{t}^{B,s}, r_{t}^{B,s} \right) \forall t \geq t_{\delta} \tag{45}
\]

for \( 0 < \delta \ll 1 \). According to Definition 2, we know that since \( r_{0}^{B,s} \) is a transitory artificial setpoint, then there exists a different artificial setpoint \( \bar{r}^{B} \) with \( T(\bar{r}^{B}) < T(r_{0}^{B,s}) \) such that the following holds:

\[
\epsilon \left[ \ell \left( x_{0}, u_{0}^{*}, r_{0}^{B,s} \right) + V_{N-1} \left( x_{0}, u_{0}^{B,s}, \mathbb{H}(r_{0}^{B,s}) \right) \right] + T \left( r_{0}^{B,s} \right) < V_{N} \left( x_{t}, r_{t}^{B,s}, \mathbb{H}(r_{t}^{B,s}) \right) + T \left( r_{t}^{B,s} \right) \tag{46}
\]

Note that the alternative backup trajectory is then constructed by taking the first state-input pair \( (x_{0}, u_{0}^{*}) \), so that (19i) holds, and then attaching the trajectory resulting from \( V_{N-1} \in \mathcal{M}(\epsilon, N, \mathbb{H}(r_{0}^{B,s})) \). Condition (19j) holds based on (46), which also implies that the chosen alternative trajectory is feasible for the MPC scheme (19). In addition, since \( T(\bar{r}^{B}) < T(r_{0}^{B,s}) \), we have that the following holds:

\[
V_{N} \left( x_{0}, u_{0}^{*}, r_{0}^{B,s} \right) + T \left( r_{0}^{B,s} \right) + \epsilon T \left( \bar{r}^{B} \right) < V_{N} \left( x_{t}, r_{t}^{B,s}, \mathbb{H}(r_{t}^{B,s}) \right) + T \left( r_{t}^{B,s} \right) \tag{47}
\]

Hence, we showed not only that there exists an alternative feasible candidate trajectory for the MPC scheme (19) but also that this trajectory has a smaller overall cost function, which, therefore, results in a contradiction.

\( r_{0}^{B,s} \in \mathbb{S}(\mathbb{Z}) \): This part is again done by contradiction, and by making use of Proposition 1, as well as the definition of the set \( \mathbb{H} \) in Assumption 2. In the following, we assume that \( r_{0}^{B,s} \in \mathbb{S}(\mathbb{Z}) \). Based on (44), we know that there exists a time \( t^{\ast} \in \mathbb{I}_{\geq 0} \) such that \( z_{t^{\ast}} = (x_{t^{\ast}}, u_{t^{\ast}}) \) satisfies

\[
\left\| z_{t^{\ast}} - r_{0}^{B,s} \right\|^{2} < h - \lambda \quad \forall t \geq t^{\ast}
\]

which, based on the construction of the set \( \mathbb{H} \) in Assumption 2 and on the definition of the manifold in (5), it implies that \( r_{0}^{B,s} \in \mathcal{M}(\epsilon, \mathbb{H}(r_{0}^{B,s})) \). In particular, in the following proof, we do not modify the learning trajectory, while we only show that there exists a different backup trajectory that has the same first predicted state \( x_{1t}^{B,s} = x_{1t} \) [to ensure (19i)], and is feasible for the MPC scheme (19), and leads to a decrease in the overall cost function. This means that the original MPC solution is not the optimal one, which results in a contradiction.

First, based on (43), we know that there exists a time \( t_{\delta} \in \mathbb{I}_{\geq 0} \) such that

\[
V_{N} \left( x_{B,s}, u_{B,s}^{*}, r_{B,s}^{*} \right) \leq \delta \quad \forall t \geq t_{\delta} \tag{45}
\]

for \( 0 < \delta \ll 1 \). According to Definition 2, we know that since \( r_{0}^{B,s} \) is a transitory artificial setpoint, then there exists a different artificial setpoint \( \bar{r}^{B} \) with \( T(\bar{r}^{B}) < T(r_{0}^{B,s}) \) such that the following holds:
Evolution of system (48) with and without constraint (19). (a) Ideal configuration where the entire set $\mathcal{Z}$ (including obstacles) is known, i.e., $\mathcal{E}_0 = \mathcal{Z}$. (b) Actual configuration at $t = 0$ where only a neighborhood of $x_0$ is known. (c) Closed-loop of system (48) controlled with (19) without (19). (d) Closed-loop of system (48) controlled with (19) with (19).

**Problem Setup:** We consider a nonlinear system representing an autonomous car, defined as follows:

$$
\begin{align*}
\dot{x} &= \begin{bmatrix} x_1 \\ x_2 \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \psi / \sin(\beta) \\ \sin(\beta) \\ \beta \\ 0 \end{bmatrix}, \\
y &= \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
\end{align*}
$$

where $x_1 \in [-\infty, 12]$ and $x_2 \in [-\infty, 2]$ represent the coordinates of the center of mass of the car, $\psi \in [-\infty, \infty]$ indicates its rotation, $v \in [-1 \text{ m/s}, 36 \text{ m/s}]$ is the longitudinal velocity, and $\beta \in [-37^\circ, 37^\circ]$ is the angle of the current velocity of the center of mass with respect to the longitudinal axis of the car, compare [40]. The inputs $u_1 \in [-10 \text{ m/s}^2, 1 \text{ m/s}^2]$, and $u_2 \in [-10^\circ / \text{s}, 10^\circ / \text{s}]$ control the acceleration of the car and the velocity of the steering angle, respectively. We define the set $\mathcal{Z}_{sc}$ as the hyper-box that encodes the state and input constraints introduced above. The parameter $l_r = 1.7 \text{ m}$ represents the distance from the center of the mass of the car to the rear axle. For the implementation we use a Euler discretization with a sampling time of $h = 200 \text{ ms}$, as done in [41], while the optimization problems are solved with CasADi, [42]. Note that, for the sake of simplicity, we consider a point-mass object. However, the actual size of the vehicle can additionally be included into the MPC scheme by adding additional constraints, as shown in [40].

As discussed above (22), to ensure initial feasibility of the MPC scheme, we initialize $\hat{F}_0$ with an arbitrarily high number, to ensure that constraint (19)) is not active at time $t = 0$, i.e., $\hat{F}_0 = 10^5$. On the contrary, the user-defined parameter $S_0$ is set to $S_0 = 0$, and $\epsilon = 0.0001$. The control goal is to get as close as possible to the desired unreachable output $y_d = [12, 1]$.

**Obstacles Description:** We define the obstacles as ellipsoidal sets in the output space $y = (x_1, x_2)$, compare Fig. 9. Each obstacle is formally defined as $\mathcal{O}_i := \{x \in \mathbb{R}^{n+m} | [x_1, x_2]^\top - o_i \|l_2 \leq r_i\}$, where $o_i \in \mathbb{R}^2$ is the center of the obstacle, $r_i \in \mathbb{R}^2$ its radius, and, $I_2$ the identity matrix, with $i = 1, \ldots, 4$. The values of $o_i$ and $r_i$ are chosen as follows:

$$
\begin{align*}
o_1 &= \begin{bmatrix} 4.0 \\ 0.3 \end{bmatrix}, \\
o_2 &= \begin{bmatrix} 4.0 \\ -0.3 \end{bmatrix}, \\
o_3 &= \begin{bmatrix} 10.0 \\ 1.0 \end{bmatrix}, \\
o_4 &= \begin{bmatrix} 7.0 \\ -1.0 \end{bmatrix}
\end{align*}
$$

$$
\begin{align*}
r_1 &= 0.51, \\
r_2 &= 0.51, \\
r_3 &= 1.50, \\
r_4 &= 1.00.
\end{align*}
$$

Note that each obstacle is visible to the vehicle if and only if it is in the neighborhood $\mathbb{H}(z_i)$ of the current output $y_i$ of the system, where $\mathbb{H}$ is introduced in the following.

The constraint set $\mathcal{Z}$ is defined based on the set $\mathcal{Z}_{sc}$, and the obstacles $\mathcal{O}_i$ described above, i.e.,

$$
\mathcal{Z} := \mathcal{Z}_{sc} \setminus \bigcup_{i=1}^4 \mathcal{O}_i.
$$

**Set $\mathbb{H}(z_i)$ and Update of $\mathcal{E}_t$ and $\mathcal{Z}_t$:** In this example, we consider the case where the vehicle is equipped with a sensor (e.g., a LIDAR sensor) that is able to perceive the surrounding $\mathbb{H}(z_i)$ of the vehicle, up to a radius of $2.5$ m in the output space $\mathbb{H}(z_i) := \{x \in \mathbb{R}^{n+m} | [x_1, x_2]^\top - y_i \|l_2 \leq 2.5\} \cap \mathcal{Z}$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Based on the set $\mathcal{H}(z_t)$, we define, at each time $t \geq 0$, the set $E_t$ as follows:

$$
E_t := Z_{sc} \cap \bigcup_{i=0}^{t} \mathcal{H}(z_i) = E_{t-1} \cup \mathcal{H}(z_t).
$$

On the contrary, the set $Z_t$ is initialized as $Z_0 := Z_{sc}$, and additionally includes the obstacles $O_t$ only if they are in a neighborhood $\mathcal{H}(z_t)$ of the system itself. This implies that the learning trajectory can freely explore the unknown space $Z_t \setminus E_t$, while the backup trajectory is forced to stay inside the explored set $E_t$.

**MPC Scheme:** We use a standard quadratic stage cost with weighting matrices $Q := \text{diag}[1, 1, 10^{-5}, 10^{-5}, 10^{-5}]$ and $R := \text{diag}[1, 1]$. In addition, to simplify the implementation, we consider a terminal equality constraints w.r.t. the artificial setpoints $r^a$. This means that the terminal region $X_t$ and the manifold set $\mathcal{S}(E_t)$ are implemented as follows:

$$
X_t(r, E_t) := \mathcal{S}(E_t) := \{(x, u) \in \mathbb{R}^{n+m} | (x, u) \in E_t, x = f(x, u)\}.
$$

Note that for the more general case where a nonsingleton terminal region is constructed, then both MPC schemes become mixed-integer nonlinear programs. The offset cost is defined as $T(y) = \|y - y^d\|^2$, and we set an MPC horizon of $N = 50$, which corresponds to 100 s in the given setup. Finally, the initial state is set as

$$
x_0 := [1.5, 0, 0, 5, 0]^\top
$$

as shown in Fig. 9.

**Discussion:** In Fig. 9(a), we show the ideal configuration where the entire set $\mathcal{Z}$ is known, i.e., all the obstacles are visible at time $t = 0$. On the contrary, in Fig. 9(b), we show the actual configuration that we employ in the considered example at $t = 0$. In particular, we see that only the neighborhood of the initial set $x_0$ is visible, which includes also the obstacles $O_1$ and $O_2$, together with the constraint $x_2 \leq 2$. In Fig. 9(c), we show how the MPC scheme (19) without the constraint (19j) behaves. Specifically, we see that the system starts in $x_0$ and converges to the artificial steady-state $[x_1, x_2]^\top \approx [3.5, 0]^\top$, which corresponds to a nontransitory setpoint due to the presence of obstacles. This happens because the vehicle is not able to get closer to the desired output $y_d$ since this would require a phase where it first moves farther away from the desired output $y^d$, and only after a proper maneuver it is able to overcome the obstacles and it can then get closer to $y^d$. Note that Fig. 9(c) also represents the behavior of system (48) under the MPC scheme (6), which is analogous to the approach proposed in [29]. Finally, in Fig. 9(d), we show the closed-loop system of system (48) under the control law resulting from (19) when (19j) is included. In this case, we see that even though the system reaches several nontransitory setpoints, it is nevertheless able to move backward (i.e., temporarily increasing the overall cost) and then move toward a different artificial setpoint that is closer to the desired output $y^d$. At time $t = 60$ s, the system converges to the output $y = [12, 1]^\top$, which corresponds to the closest reachable output $y^d_t(Z)$, as defined in (15). We want to additionally clarify that the controlled system temporarily converges to local minima because this is optimal according to the cost function (19a), and based on the safe set $E_t$ known at the related time. This behavior could be potentially avoided by enlarging the neighborhood set $\mathcal{H}(z_t)$, which is, however, limited by the employed hardware.

In conclusion, this example illustrates the capabilities of the proposed MPC scheme to escape from local minima, while ensuring safety (i.e., closed-loop constraint satisfaction), convergence, and the possibility to consider a nonreachable output $y^d$.

**VII. CONCLUSION**

In this article, we presented a novel MPC framework for output tracking that deals with partially unknown constraints and ensures theoretical guarantees. We showed how the proposed MPC scheme optimizes over a learning and a backup trajectory, with the final goal to explore unknown areas while ensuring closed-loop safety. The cost function for the learning trajectory is divided into a tracking cost, which minimizes the difference w.r.t. an online optimized setpoint, and an offset cost that minimizes the output of such a setpoint and the desired one. On the contrary, the cost function for the backup trajectory is only marginally considered and only minimizes the offset cost w.r.t. a potentially different setpoint. We provided formal guarantees for convergence, recursive feasibility, and closed-loop constraint satisfaction. Finally, we showed the applicability of the proposed approach on a numerical example that considers an autonomous vehicle that drives in a partially unknown environment.

**REFERENCES**

[1] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI, USA: Nob Hill Pub., 2009.

[2] B. Kouvaritakis and M. Cannon, *Model Predictive Control: Classical, Robust and Stochastic*. Berlin, Germany: Springer, 2016.

[3] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. Berlin, Germany: Springer, 2017.

[4] M. Zanon and S. Gros, “Safe reinforcement learning using robust MPC,” *IEEE Trans. Autom. Control*, vol. 66, no. 8, pp. 3638–3652, Aug. 2021.

[5] M. Lorenzen, M. Cannon, and F. Allgöwer, “Robust MPC with recursive model update,” *Automatica*, vol. 103, pp. 461–471, 2019.

[6] R. Soloperto, J. Köhler, and F. Allgöwer, “Augmenting MPC schemes with active learning: Intuitive tuning and guaranteed performance,” *IEEE Control Syst. Lett.*, vol. 4, no. 3, pp. 713–718, Jul. 2020.

[7] J. Köhler, P. Kotting, R. Soloperto, F. Allgöwer, and M. A. Müller, “A robust adaptive model predictive control framework for nonlinear uncertain systems,” *Int. J. Robust Nonlinear Control*, vol. 31, no. 18, pp. 8725–8749, 2021.

[8] R. Soloperto, M. A. Müller, and F. Allgöwer, “Guaranteed closed-loop learning in model predictive control,” *IEEE Trans. Autom. Control*, vol. 68, no. 2, pp. 991–1006, Feb. 2023.

[9] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, “FASTER: Fast and safe trajectory planner for navigation in unknown environments,” *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 922–938, Apr. 2022.

[10] D. Saccani and L. Fagiano, “Autonomous UAV navigation in an unknown environment via multi-trajectory model predictive control,” in *Proc. Eur. Control Conf., 2021*, pp. 1577–1582.

[11] A. A. Feldbaum, “Dual control theory,” *Automat. Remote Control*, vol. 21, no. 9, pp. 1240–1249, 1960.

[12] T. A. N. Heirung, B. E. Ydstie, and B. Foss, “Dual adaptive model predictive control,” *Automatica*, vol. 80, pp. 340–348, 2017.

[13] R. Soloperto, J. Köhler, M. A. Müller, and F. Allgöwer, “Dual adaptive MPC for output tracking of linear systems,” in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 1377–1382.

[14] A. Weiss and S. Di Cairano, “Robust dual control MPC with guaranteed constraint satisfaction,” in *Proc. IEEE 53rd Conf. Decis. Control*, 2014, pp. 6713–6718.
SOLOPERTO et al.: SAFE EXPLORATION AND ESCAPE LOCAL MINIMA WITH MPC UNDER PARTIALLY UNKNOWN CONSTRAINTS

[15] G. Marafiotti, R. R. Bitmead, and M. Hovd, “Persistently exciting model predictive control,” Int. J. Adaptive Control Signal Process., vol. 28, no. 6, pp. 536–552, 2014.

[16] A. Alessandretti, A. P. Aguiar, and C. N. Jones, “Trajectory-tracking and path-following controllers for constrained underwater vehicles using model predictive control,” in Proc. Eur. Control Conf., 2013, pp. 1371–1376.

[17] A. Mesbah, “Stochastic model predictive control with active uncertainty learning: A survey on dual control,” Annu. Rev. Control, vol. 45, pp. 107–117, 2018.

[18] J. Tordjman, D. Limon, and J. P. How, “Faster: Fast and safe trajectory planning for flights in unknown environments,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2019, pp. 1934–1940.

[19] I. Sánchez, A. D’Jorge, G. V. Raffo, A. H. González, and A. Ferramosca, “Nonlinear model predictive path following controller with obstacle avoidance,” J. Intel. Robot. Syst., vol. 102, no. 1, 2021, Art. no. 16.

[20] I. Batkovic, M. Ali, P. Falcone, and M. Zanon, “Safe trajectory tracking in uncertain environments,” IEEE Trans. Autom. Control, to be published, doi: 10.1109/TAC.2022.3207875.

[21] A. D. Bonzanini, A. Mesbah, and S. Di Cairano, “Perception-aware chance-constrained model predictive control for uncertain environments,” in Proc. Amer. Control Conf., 2021, pp. 2082–2087.

[22] A. D. Bonzanini, A. Mesbah, and S. Di Cairano, “On the stability properties of perception-aware chance-constrained MPC in uncertain environments,” in Proc. IEEE 60th Conf. Decis. Control, 2021, pp. 2177–2182.

[23] S. Lucia, R. Paulen, and S. Engell, “Multi-stage nonlinear model predictive control with verified robust constraint satisfaction,” in Proc. IEEE 53rd Conf. Decis. Control, 2014, pp. 2816–2821.

[24] M. Turchetta, F. Berkenkamp, and A. Krause, “Safe exploration for interactive machine learning,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 1–11.

[25] T. Lew, R. Bonalli, and M. Pavone, “Chance-constrained sequential convex programming for robust trajectory optimization,” in Proc. Eur. Control Conf., 2020, pp. 1871–1878.

[26] T. Köfler, F. Berkenkamp, M. Turchetta, and A. Krause, “Learning-based model predictive control for safe exploration,” in Proc. IEEE Conf. Decis. Control, 2018, pp. 6059–6066.

[27] S. Gros and M. Zanon, “Data-driven economic NMPC using reinforcement learning,” IEEE Trans. Autom. Control, vol. 65, no. 2, pp. 636–648, Feb. 2020.

[28] D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho, “MPC for tracking piecewise constant references for constrained linear systems,” Automatica, vol. 44, pp. 2382–2387, 2008.

[29] D. Limon, A. Ferramosca, I. Alvarado, and T. Alamo, “Nonlinear MPC for tracking piece-wise constant reference signals,” IEEE Trans. Autom. Control, vol. 63, no. 11, pp. 3735–3750, Nov. 2018.

[30] J. A. Paulson, T. L. Santos, and A. Mesbah, “Mixed stochastic-deterministic tube MPC for off-road trucking in the presence of plant-model mismatch,” J. Proc. Control, vol. 83, pp. 102–120, 2019.

[31] D. Limon, I. Alvarado, T. Alamo, and E. Camacho, “Robust tube-based MPC for tracking of constrained linear systems with additive disturbances,” J. Proc. Control, vol. 20, pp. 248–260, 2010.

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

[33] A. D’Jorge, B. F. Santoro, A. Anderson, A. H. González, and A. Ferramosca, “Stochastic model predictive control for tracking linear systems,” Optimal Control Appl. Methods, vol. 41, no. 1, pp. 65–83, 2020.

[34] R. Soloperto, J. Köhler, and F. Allgöwer, “A nonlinear MPC scheme for output tracking without terminal ingredients,” IEEE Trans. Autom. Control, vol. 68, no. 4, pp. 2368–2375, Apr. 2023.

[35] M. A. Müller, D. Angelii, and F. Allgöwer, “Economic model predictive control with self-tuning terminal cost,” Eur. J. Control, vol. 19, no. 5, pp. 408–416, 2013.

[36] L. Fagiano and A. R. Teel, “Generalized terminal state constraint for model predictive control,” Automatica, vol. 49, pp. 2622–2631, 2013.

[37] I. Köfler, M. A. Müller, and F. Allgöwer, “A nonlinear tracking model predictive control scheme for dynamic target signals,” Automatica, vol. 118, 2020, Art. no. 109030.

[38] A. Boccia, L. Grüne, and K. Worthmann, “Stability and feasibility of state constrained MPC without stabilizing terminal constraints,” Syst. Control Lett., vol. 72, pp. 14–21, 2014.

[39] M. Ellis and P. D. Christofides, “Economic model predictive control with time-varying objective function for nonlinear process systems,” AIChE J., vol. 60, no. 2, pp. 507–519, 2014.

[40] R. Soloperto, J. Köhler, M. A. Müller, and F. Allgöwer, “Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control,” in Proc. 18th Eur. Control Conf., 2019, pp. 811–817.

[41] J. Kong, M. Pfiffer, G. Schildbach, and F. Borrelli, “Kinematic and dynamic vehicle models for autonomous driving control design,” in Proc. Intell. Veh. Symp., 2015, pp. 1094–1099.

[42] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “Casadi: A software framework for nonlinear optimization and optimal control,” Math. Program. Comput., vol. 11, no. 1, pp. 1–36, 2019.

Raffaele Soloperto (Member, IEEE) received both the bachelor’s and master’s degrees in automation engineering from the University of Bologna, Bologna, Italy, in 2014 and 2016, respectively, and the Ph.D. degree from the Institute for Systems Theory and Automatic Control, University of Stuttgart, Stuttgart, Germany, in 2022, and from the International Max-Planck Research School (IMPRS), under the supervision of Prof. Frank Allgöwer.

Since 2022, he is a Postdoctoral Researcher with the Automatic Control Laboratory, ETH Zurich, Zurich, Switzerland. During his studies, he visited the TU-Münich, Munich, Germany, ETH-Zürich, EPF-Lousanne, Lausanne, Switzerland, and UC-Berkeley, Berkeley, CA, USA. His research interests include robust, adaptive, and learning-based model predictive control.

Ali Mesbah (Senior Member, IEEE) received the B.S. degree in chemical engineering from the University of Tehran, in 2004, M.S. degree in chemical engineering from the Delft University of Technology, in 2006, and Ph.D. degree in systems and control from the Delft University of Technology, in 2010. He is Associate Professor of Chemical and Biomolecular Engineering with the University of California at Berkeley, Berkeley, CA, USA. His research interests include learning and predictive control of uncertain systems.

He was Postdoctoral Fellow in Delft University of Technology, from 2011 to 2012 and Senior Postdoctoral Associate in Massachusetts Institute of Technology from 2012 to 2014. Dr. Ali is a Senior Member of AIChE. He was the recipient of the Best Application Paper Award of the IFAC World Congress in 2020, the AIChE’s 35 Under 35 Award in 2017, the IEEE Control Systems Magazine Outstanding Paper Award in 2017, and the AIChE CAST W. David Smith, Jr. Graduation Publication Award in 2015. He is on the IEEE Control Systems Society conference editorial board as well as the editorial board of IEEE TRANSACTIONS ON RADIATION AND PLASMA MEDICAL SCIENCES and Optimal Control Applications and Methods.

Frank Allgöwer received the degree in engineering cybernetics and applied mathematics from the University of Stuttgart, Stuttgart, Germany, and the University of California (UCLA), Los Angeles, CA, USA, respectively, and the Ph.D. degree from the University of Stuttgart, Stuttgart, Germany.

Since 1999, he is the Director of the Institute for Systems Theory and Automatic Control, and a Professor with the University of Stuttgart. His research interests include networked control, cooperative control, predictive control, and nonlinear control with application to a wide range of fields including systems biology.

Dr. Allgöwer was the President of the International Federation of Automatic Control, from 2017 to 2020, and since 2012, as Vice President of the German Research Foundation DFG.