Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 54-6 (2021) 365–370

Output-Lifted Learning Model Predictive Control

Siddharth H. Nair * Ugo Rosolia ** Francesco Borrelli *

* *Department of Mechanical Engineering, University of California Berkeley. Email IDs: {siddharth_nair, fborrelli}@berkeley.edu*
** *Department of Mechanical and Civil Engineering, California Institute of Technology. Email ID: urosolia@caltech.edu*

Abstract: We propose a computationally efficient Learning Model Predictive Control (LMPC) scheme for constrained optimal control of a class of nonlinear systems where the state and input can be reconstructed using lifted outputs. For the considered class of systems, we show how to use historical trajectory data collected during iterative tasks to construct a convex value function approximation along with a convex safe set in a lifted space of virtual outputs. These constructions are iteratively updated with historical data and used to synthesize predictive control policies. We show that the proposed strategy guarantees recursive constraint satisfaction, asymptotic stability, and non-decreasing closed-loop performance at each policy update. Finally, simulation results demonstrate the effectiveness of the proposed strategy on the kinematic unicycle.

Keywords: Learning and Predictive Control, Stability and Recursive Feasibility

1. INTRODUCTION

Infinite-horizon optimal control has a long and celebrated history, with the cornerstones laid in the 1950s by Pontryagin (2018) and Bellman (1966). The problem involves seeking a control signal that minimizes the cost incurred by a trajectory of a dynamical system starting from an initial condition over an infinite time horizon. While certain problem settings admit analytical solutions (like unconstrained LQR (Kwakernaak and Sivan (1972))), the infinite-horizon optimal control problem for general nonlinear dynamical systems subject to constraints, is challenging to solve. This is because these problems require the numerical solution of an infinite-dimensional optimization problem, which is intractable even in the discrete-time setting (where the solution is an infinite sequence of control inputs instead of a control input signal).

Model Predictive Control (MPC) is an attractive methodology for tractable synthesis of feedback control of constrained nonlinear discrete-time systems. The control action at every instant requires the solution of a finite-horizon optimal control problem with a suitable constraint and cost on the terminal state of the system to approximate the infinite-horizon problem. These terminal components are designed so that the closed-loop system is stable and satisfies constraints. This is achieved by constraining the terminal state to lie in a control invariant set with an associated Control Lyapunov function (CLF). The computation of these sets with an accompanying CLF for nonlinear systems is challenging, in general, and a proper review goes outside the scope of this conference paper.

For iterative tasks where the system starts from the same position for every iteration of the optimal control problem, data from previous iterations may be used to update the MPC design using ideas from Iterative Learning Control (ILC) (Bristow et al. (2006); Lee and Lee (1997); Cueli and Bordons (2008)). In these strategies the goal of the controller is to track a given reference trajectory, and the tracking error from the previous execution is used to update the controller. For control problems where a reference trajectory may be hard to compute, Rosolia and Borrelli (2017) proposed a reference-free iterative policy synthesis strategy, called Learning Model Predictive Control (LMPC) which iteratively constructs a control invariant terminal set and an accompanying terminal cost function using historical data. These quantities are discrete, therefore the LMPC relies on the solution of a Mixed-Integer Nonlinear Program (MINLP) at each instant for guaranteed stability and constraint satisfaction. We build on the work of Rosolia and Borrelli (2017) and propose a strategy to reduce the computational burden for a class of nonlinear systems by replacing these discrete sets and functions with convex ones while still maintaining safety and performance guarantees.

In this work, we present a LMPC framework for a class of discrete-time nonlinear systems for which the state and input can be reconstructed using *lifted* outputs. These *lifted* outputs are constructed using *flat* outputs (Guillot and Millerioux (2019)) which have also been used in Aranda-Bricaire et al. (1996) to construct dynamic feedback linearizing inputs for discrete-time systems. Existing works on constrained control for such systems require a carefully designed reference trajectory which is then tracked using MPC with a linear model obtained either by a first order approximation (De Doná et al. (2009)) or by feedback linearization (Wang et al. (2019); Greeff and Schoellig (2018); Kandler et al. (2012)). In both cases, there are

* This work has been sponsored by the Office of Naval Research

10.1016/j.ifacol.2021.08.571

no formal guarantees of closed-loop system stability and constraint satisfaction. The contribution of this article is twofold. First, we show how to construct convex terminal set and terminal cost using historical *lifted* output data for the MPC optimization problem. Second, we show that with some mild assumptions, a convex synthesis of the terminal cost is permissible on the space of *lifted* outputs. As opposed to the discrete formulation of the terminal set and cost in Rosolia and Borrelli (2017) (thus requiring solutions to MINLPs), our formulation enables us to solve continuous Nonlinear Programs (NLPs). This can significantly decrease the computational overheads associated with computing the control action at each instant.

The paper is organized as follows. We begin by formally describing the problem we want to solve in Section 2 and briefly discuss prerequisites in Section 3. Section 4 shows how to construct the terminal set and terminal cost in the lifted output space and it introduces the control design. Finally, Section 5 presents numerical results that illustrate our proposed approach on the kinematic unicycle.

2. PROBLEM FORMULATION

Consider a nonlinear discrete-time system given by the dynamics

$$x_{t+1} = f(x_t, u_t), \tag{1}$$

where $x_t \in \mathbb{X} \subseteq \mathbb{R}^n$ and $u_t \in \mathbb{U} \subseteq \mathbb{R}^m$ are the system state and input respectively at time t. Let x_F be an unforced equilibrium of (1), $x_F = f(x_F, 0)$ with $f(\cdot, 0)$ being continuous at x_F. The *lifted* output for the nonlinear system (1) is defined below.

Definition 1. Let $y_t = h(x_t)$ with $h : \mathbb{X} \to \mathbb{R}^m$ be the output of system (1). If $\exists R \in \mathbb{N}$ and a function $\mathcal{F} : \mathbb{R}^{m\times R+1} \to \mathbb{X} \times \mathbb{U}$, such that the state/input pair (x_t, u_t) can be uniquely reconstructed from a sequence of outputs $y_t, \dots, y_{t+R}$

$$(x_t, u_t) = \mathcal{F}([y_t, y_{t+1}, \dots, y_{t+R}]), \tag{2}$$

then the lifted output is the matrix

$$\mathbf{Y}_t = [y_t, \dots, y_{t+R}] \in \mathbb{R}^{m\times R+1}. \tag{3}$$

Remark 1. The output $y_t = h(x_t)$ corresponding to the lifted output in definition 1 is also called a flat output in Guillot and Millerioux (2019). For linear discrete-time systems, the existence of the lifted output is equivalent to the system being controllable and *strongly* observable with the output $y_t = Cx_t$ (Yong et al. (2015)).

Assumption 1. The system (1) has a lifted output $\mathbf{Y}_t$ with $y_t = h(x_t)$.

Consider the following infinite-horizon constrained optimal control problem for system (1) with initial state $x_0 = x_S$:

$$\begin{aligned} J^*_{0\to\infty}(x_S) = \min_{u_0,u_1,\dots} \quad & \sum_{k\geq 0} c(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k),\ \forall k \geq 0 \\ & x_k \in \mathcal{X}, u_k \in \mathcal{U},\ \forall k \geq 0 \\ & x_0 = x_S. \end{aligned} \tag{4}$$

The state constraints $\mathcal{X}$ and input constraints $\mathcal{U}$ are described by convex sets, and $c(\cdot,\cdot)$ is a continuous, convex, and positive definite function that equals zero only at the equilibrium, i.e., $c(x_F, 0) = 0$. Observe that due to continuity and positive definiteness of stage cost $c(\cdot)$, a trajectory corresponding to the optimizer of (4) has bounded cost and so must necessarily have its state converge to x_F.

We aim to synthesize a state-feedback policy that approximates the solution to the infinite-horizon (and infinite-dimensional) problem (4) such that it captures its most desirable properties: (i) constraint satisfaction (feasibility) and (ii) asymptotic convergence to x_F. To tackle the infinite-dimensional nature of the problem, we use MPC which solves finite-horizon versions of (4) at each time step. To ensure that the MPC has the desired properties, we build on the Learning Model Predictive Control (LMPC) framework which solves problem (4) iteratively using historical data. In the next section, we proceed to briefly describe these two techniques.

Remark 2. Notice that to streamline the presentation we considered iterative tasks. However, the proposed strategy can be used also when the initial condition changes at each iteration. As we will show in Section V, to guarantee safety and closed-loop stability it is only required that the initial condition belongs to the region of attraction of the LMPC policy.

Remark 3. We considered deterministic systems, but the proposed strategy may be extended to handle additive uncertainty using robust tube MPC strategies (Mayne et al. (2011)). The key idea is to leverage the proposed approach to control the nominal dynamics and use a precomputed feedback gain to handle the uncertainty, as discussed in (Rosolia and Borrelli, 2020, Section 7).

3. PRELIMINARIES

3.1 Model Predictive Control

Consider the following finite-horizon problem at each time t from state x_t.

$$\begin{aligned} J_{t\to t+N}(x_t) = \min_{\mathbf{u_t}} \quad & Q(x_{N|t}) + \sum_{k=0}^{N-1} c(x_{k|t}, u_{k|t}) \\ \text{s.t.} \quad & x_{k+1|t} = f(x_{k|t}, u_{k|t}),\ \forall k \in \{0, \dots, N-1\} \\ & x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U},\ \forall k \in \{0, \dots, N-1\} \\ & x_{N|t} \in \mathcal{X}_f, \\ & x_{0|t} = x_t \end{aligned} \tag{5}$$

where $\mathbf{u_t} = [u_{0|t}, \dots, u_{N-1|t}]$, the initial condition $x_0 = x_S$, $\mathcal{X}_f \subseteq \mathcal{X}$ is a control invariant set ((Borrelli et al., 2017, Definition 10.9)) for the system (1) with associated Control Lyapunov Function (CLF) ((Borrelli et al., 2017, Remark 12.3)) $Q(\cdot)$ for the equilibrium x_F chosen as the terminal cost function. If $\mathbf{u}^*_t = [u^*_{0|t}, \dots, u^*_{N-1|t}]$ is the minimizer of (5), then the MPC controller is given by

$$u_t = \pi_{MPC}(x_t) = u^*_{0|t} \tag{6}$$

The control invariant set $\mathcal{X}_f$ and the CLF $Q(\cdot)$ are coupled with each other and are critical to ensuring that the MPC policy (6) in closed-loop yields a feasible and stabilizing solution to the infinite-horizon problem (4). Observe that if the optimal cost $J^*_{t\to\infty}(x_t)$ was known $\forall t \geq 0$, setting $Q(x_{N|t}) = J^*_{t+N\to\infty}(x_{N|t})$ solves (4) without requiring a terminal constraint $\mathcal{X}_f$ in (5). The other extreme case is setting $\mathcal{X}_f = \{x_F\}$ in (5) which would yield a stable and feasible solution without requiring a terminal cost $Q(\cdot)$. That being said, computing $J^*_{t\to\infty}(\cdot)$ exactly is possible only in trivial cases and setting $\mathcal{X}_f = \{x_F\}$ may lead to an infeasible optimization problem if x_F is not reachable from x_S in N steps. The goal is to design $\mathcal{X}_f$ and $Q(\cdot)$ so that (5) is feasible for all $t \geq 0$ while capturing the convergence properties of the infinite-horizon optimal control problem.

3.2 Learning Model Predictive Control

LMPC iteratively approximates the solution of (4) using the MPC problem (5). At iteration j, it uses historical data in the form of state-input trajectories from completed iterations $i \in \{0, 1, \dots j-1\}$ to construct the terminal set $\mathcal{X}_f$ and terminal cost $Q(\cdot)$. Let x^j_t, u^j_t and y^j_t be the state, input, and output of the system respectively at time t, corresponding to the jth iteration. At iteration j, the terminal set $\mathcal{X}^j_f$ and terminal cost $Q^{j-1}(\cdot)$ are defined as follows:

$$\mathcal{X}^j_f = \mathcal{SS}^{j-1} = \bigcup_{i=0}^{j-1} \bigcup_{t\geq 0} \{x^i_t\} \tag{7}$$

$$Q^{j-1}(x) = \begin{cases} \min\limits_{(i,t)\in\mathcal{I}_{j-1}(x)} \sum\limits_{k\geq t} c(x^i_k, u^i_k) & x \in \mathcal{SS}^{j-1} \\ \infty & x \notin \mathcal{SS}^{j-1} \end{cases} \tag{8}$$

where $\mathcal{I}_{j-1}(x) = \{(i,t) | x = x^i_t \in \mathcal{SS}^{j-1}\}$. Simply stated, the terminal set is chosen as the collection of states from previous iterations (the safe set $\mathcal{SS}^{j-1}$) and the terminal cost ($Q^{j-1}(\cdot)$) at these states is the cost of the trajectory obtained starting from that state. The terminal set is discrete and the terminal cost function is only defined on these discrete states which makes (5) a mixed-integer nonlinear program (MINLP). The computational overhead for computing such MINLP solutions is prohibitive for online, repeated solutions of (5).

We would like to investigate if the lifted output from definition 1 helps alleviate the combinatorial nature of the optimization problem for more tractable synthesis of feedback control to solve problem (4).

4. OUTPUT-LIFTED LMPC

In this section we present our LMPC design using flat outputs. First, we highlight some technical assumptions that we impose on the lifted output map $\mathcal{F}(\cdot)$ from equation (2). We then show how to use the stored lifted outputs from previous iterations to construct a *convex* safe set in the lifted output space. The constructed set is shown to be control invariant in this space and therefore it can be used to guarantee safety in a receding horizon scheme. Afterwards, we construct a *convex* terminal cost in the lifted output space and prove that it is a CLF on the constructed set. Finally, we combine these components and present our control design.

4.1 Lifted Output Map Properties

In addition to the existence of the lifted outputs in definition 1, we require that the map $\mathcal{F}(\cdot)$ has the properties described in the following assumption.

Assumption 2. The lifted output $\mathbf{Y}_t$ corresponding to $y_t = h(x_t)$ and the map $\mathcal{F}(\cdot)$ satisfy the following properties:

(A) The map $\mathcal{F}(\cdot)$ in (2) requires R and $R+1$ outputs for identifying the state and the input, respectively. More formally, we have that

$$x_t = \mathcal{F}_x([y_t, y_{t+1}, \dots, y_{t+R-1}]) \tag{9}$$
$$u_t = \mathcal{F}_u([y_t, y_{t+1}, \dots, y_{t+R}]) \tag{10}$$

(B) The map $\mathcal{F} = (\mathcal{F}_x, \mathcal{F}_u) : \mathbb{R}^{m\times R+1} \to \mathbb{X} \times \mathbb{U}$ is continuous at $\mathbf{Y}_F = [y_F, \dots, y_F] \in \mathbb{R}^{m\times R+1}$ where $y_F = h(x_F)$.

(C) Let $\mathcal{F}^i : \mathbb{R}^{m\times R+1} \to \mathbb{R}$ be the ith component of the map $\mathcal{F} : \mathbb{R}^{m\times R+1} \to \mathbb{X} \times \mathbb{U} \subset \mathbb{R}^{n+m}$ where $i = 1, \dots, n+m$. Then $\forall i \in \{1, \dots, n+m\}$, the maps $\mathcal{F}^i$ are monotonic on any line restriction, i.e., $\mathcal{F}^i(t\mathbf{y}_1 + (1-t)\mathbf{y}_2)$ is monotonic $\forall \mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^{m\times R+1}$, $t \in [0, 1]$.

The intuition for consideration of outputs in Assumptions 2(A) arises from observing the kinematics of simple mechanical systems, where the kinematics are not affected explicitly by control. Assumption 2(B) is technical, and it is required for showing that an optimizer of the infinite-horizon optimal control problem stabilizes the system to x_F. The following proposition clarifies the need of Assumption 2(C) for constraint satisfaction in the LMPC.

Proposition 1. Suppose that $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^m$ are given by box constraints, $\|D_x x - d_x\|_\infty \leq 1$ and $\|D_u u - d_u\|_\infty \leq 1$ respectively for some real, constant diagonal matrices D_x, D_u and vectors d_x, d_u. Let $\{\mathbf{Y}^1, \dots, \mathbf{Y}^p\}$ be any set of lifted outputs such that $\mathcal{F}(\mathbf{Y}^j) \in \mathcal{X} \times \mathcal{U}$ for each $j = 1, \dots, p$. Then if assumption 2(C) holds, we have $\mathcal{F}(\mathbf{Y}) \in \mathcal{X} \times \mathcal{U}$ for any $\mathbf{Y} \in \text{conv}(\{\mathbf{Y}^1, \dots, \mathbf{Y}^p\})$.

Proof. Please see appendix of Nair et al. (2020). ∎

Assumption 2(C) is equivalent to requiring that each $\mathcal{F}^i(\cdot)$ is quasiconvex and quasiconcave, which can be checked using first-order and second-order sufficiency conditions detailed in (Boyd and Vandenberghe, 2004, Section 3.4). In Sections 5.1, we see that either may be relaxed depending on the domain of dynamics $f(\cdot, \cdot)$, which could trivially lower or upper bound the state space or input space. In view of proposition 1, we make the following assumption.

Assumption 3. The state constraints $\mathcal{X} \subset \mathbb{R}^n$ and input constraints $\mathcal{U} \subset \mathbb{R}^m$ are box constraints,

$$\mathcal{X} = \{\|D_x x - d_x\|_\infty \leq 1\} \quad \mathcal{U} = \{\|D_u u - d_u\|_\infty \leq 1\}$$

for some real, constant diagonal matrices D_x, D_u and vectors d_x, d_u.

Remark 4. If the map $\mathcal{F}(\cdot)$ is linear-fractional (projective)

$$\mathcal{F}(\mathbf{Y}) = \frac{A\mathbf{Y} + b}{c^\top \mathbf{Y} + d}, \quad c^\top \mathbf{Y} + d > 0,$$

then Assumption 3 can be relaxed to requiring $\mathcal{X}$ and $\mathcal{U}$ to be any convex set. This follows because linear-fractional functions preserve convexity of sets by mapping the line between any two points in its domain space to a line between the images of the two points ((Boyd and Vandenberghe, 2004, Section 3.4)).

4.2 Convex Safe Set

Let $\mathbf{Y}^j_t = [y^j_t, \dots, y^j_{t+R}]$ be the lifted output at time t and iteration j. Similarly, define the matrix $\mathbf{y}^j_t$ using R outputs as

$$\mathbf{y}_t^j = [y_t^j, \ldots, y_{t+R-1}^j] \in \mathbb{R}^{m \times R} \tag{11}$$

Note that each $\mathbf{y}_t^j$ uniquely identifies a state x_t^j via the map (9), $\mathcal{F}_x(\mathbf{y}_t^j) = x_t^j$. We define a successful iteration as one whose corresponding state trajectory converges to x_F while simultaneously meeting state constraints $\mathcal{X}$ and input constraints $\mathcal{U}$. This implies that a successful iteration corresponds to a feasible trajectory of (4).

For iteration j, define the *Output Safe Set* as the set of $\mathbf{y}_t^j$s corresponding to trajectories from preceding successful iterations (denoted by $\mathcal{I}_j \subseteq \{0, 1, \ldots, j-1\}$), i.e.,

$$\mathcal{SS}_{\mathbf{y}}^{j-1} = \bigcup_{i \in \mathcal{I}_j} \bigcup_{k=0}^{\infty} \{\mathbf{y}_k^i\}. \tag{12}$$

Taking the convex hull of this set, we now define the *Convex Output Safe Set*

$$\mathcal{CS}_{\mathbf{y}}^{j-1} = \mathrm{conv}(\mathcal{SS}_{\mathbf{y}}^{j-1}). \tag{13}$$

Define the forward-time shift $\delta(\cdot,\cdot)$ dynamics on $\mathcal{CS}_{\mathbf{y}}^{j-1}$ given by

$$\begin{aligned}\mathbf{y}_{t+1} &= [y_{t+1}, y_{t+2}, \ldots, y_{t+R}] \\ &= \delta([y_t, y_{t+1}, \ldots, y_{t+R-1}], y_{t+R}) \\ &= \delta(\mathbf{y}_t, y_{t+R})\end{aligned} \tag{14}$$

We now show that the Convex Safe Set $\mathcal{CS}_{\mathbf{y}}^{j-1}$ is in fact, control invariant for (14) in the following proposition and correspond to states and inputs within constraints. (as depicted in Figure 1).

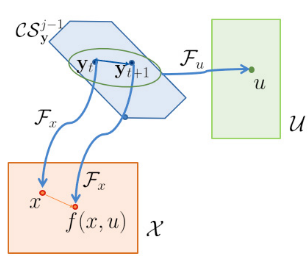

Fig. 1. Illustration of the claim in Proposition 2

Proposition 2. Under Assumptions 2 and 3, the set $\mathcal{CS}_{\mathbf{y}}^{j-1}$ defined in (13) is control invariant for the forward-time shift dynamics (14), i.e.,

$$\forall \mathbf{y}_t \in \mathcal{CS}_{\mathbf{y}}^{j-1}, \exists y_{t+R} \;:\; \mathbf{y}_{t+1} = \delta(\mathbf{y}_t, y_{t+R}) \in \mathcal{CS}_{\mathbf{y}}^{j-1}, \tag{15}$$

with

$$\begin{gathered} x = \mathcal{F}_x(\mathbf{y}_t) \in \mathcal{X},\; u = \mathcal{F}_u(\mathbf{y}_t, y_{t+R}) \in \mathcal{U}, \\ \mathcal{F}_x(\mathbf{y}_{t+1}) = f(x,u) \in \mathcal{X}\end{gathered}$$

Proof. By definition of $\mathcal{CS}_{\mathbf{y}}^{j-1}$ we have for $\mathbf{y}_t \in \mathcal{CS}_{\mathbf{y}}^{j-1}$ there exists a set of multipliers λ_k^i such that

$$\begin{gathered}\mathbf{y}_t = \sum_{i \in \mathcal{I}_{j-1}} \sum_{k \geq 0} \lambda_k^i \mathbf{y}_k^i,\; \mathbf{y}_k^i \in \mathcal{SS}_{\mathbf{y}}^{j-1} \subset \mathbb{R}^{m \times R}, \\ \sum_{i \in \mathcal{I}_{j-1}} \sum_{k \geq 0} \lambda_k^i = 1,\; \lambda_k^i \geq 0.\end{gathered} \tag{16}$$

By the definition of $\mathcal{SS}_{\mathbf{y}}^{j-1}$, each of the $\mathbf{y}_k^i$s in (16) corresponds to a feasible state, meaning $\mathcal{F}_x(\mathbf{y}_k^i) \in \mathcal{X}$. Invoking Proposition 1 then gives us,

$$\mathcal{F}_x(\mathbf{y}_t) = x \in \mathcal{X} \tag{17}$$

Again, the definition of $\mathcal{SS}_{\mathbf{y}}^{j-1}$ gives $\mathbf{y}_k^i \in \mathcal{SS}_{\mathbf{y}}^{j-1} \Rightarrow \mathbf{y}_{k+1}^i \in \mathcal{SS}_{\mathbf{y}}^{j-1}$. We use the lifted output $\mathbf{Y}_k^i$ and map $\mathcal{F}_u(\cdot)$ to reconstruct the input applied in the ith iteration at time k as $u_k^i = \mathcal{F}_u([y_k^i, y_{k+1}^i, \ldots, y_{k+R}^i]) = \mathcal{F}_u([\mathbf{y}_k^i, y_{k+R}^i])$ and note that $u_k^i \in \mathcal{U}$ for all $i \in \mathcal{I}_{j-1}$ by the definition of the set $\mathcal{SS}_{\mathbf{y}}^{j-1}$. Consider the following control input

$$\begin{aligned} u &= \mathcal{F}_u\Big(\sum_{i \in \mathcal{I}_{j-1}} \sum_{k \geq 0} \lambda_k^i [\mathbf{y}_k^i, y_{k+R}^i]\Big) \\ &= \mathcal{F}_u([\mathbf{y}_t, y_{t+R}])\end{aligned} \tag{18}$$

where $y_{t+R} = \sum_{i \in \mathcal{I}_{j-1}} \sum_{k \geq 0} \lambda_k^i y_{k+R}^i$. Invoking proposition 1 again proves $u \in \mathcal{U}$. Also see that

$$\begin{aligned}\mathbf{y}_{t+1} &= \delta([\mathbf{y}_t, y_{t+R}]) \\ &= \delta\Big(\sum_{i \in \mathcal{I}_{j-1}} \sum_{k \geq 0} \lambda_k^i [\mathbf{y}_k^i, y_{k+R}^i]\Big) \\ &= \sum_{i \in \mathcal{I}_{j-1}} \sum_{k \geq 0} \lambda_k^i \mathbf{y}_{k+1}^i \Rightarrow \mathbf{y}_{t+1} \in \mathcal{CS}_{\mathbf{y}}^{j-1}.\end{aligned} \tag{19}$$

Let $u_2, \ldots, u_{R-1} \in \mathbb{R}^m$ be the remaining inputs that generate $[y_t, \mathbf{y}_{t+1}] \in \mathbb{R}^{m \times R+1}$, i.e.,

$$\begin{aligned}[y_t, \mathbf{y}_{t+1}] =& [h(x), h(f(x,u)), h(f^{(2)}(x,u,u_2)), \ldots, \\ & h(f^{(R-1)}(x,u,\ldots,u_{R-1}))] \in \mathbb{R}^{m \times R+1}\end{aligned} \tag{20}$$

where $f^{(k)}(x,u,\ldots,u_k) = \underbrace{f(\ldots(f}_{k \text{ times}}(x,u),\ldots u_k)$.Using the map (9) to construct the state, we can write

$$\begin{aligned}\mathcal{F}_x(\mathbf{y}_{t+1}) =& \mathcal{F}_x([h(f(x,u)), h(f^{(2)}(x,u,u_2)), \ldots, \\ & h(f^{(R-1)}(x,u,\ldots,u_{R-1}))]) \\ =& f(x,u)\end{aligned}$$

where the last equality is true because of the unique correspondence from

$$[y_t, \ldots, y_{t+R-1}] = [h(x_t), \ldots, h(f^{(R-1)}(x_t, u_t, \ldots, u_{t+R-1}))]$$

to x_t (Definition 1). Finally, invoking proposition 1 using sequences $\mathbf{y}_{k+1}^i, \forall i \in \mathcal{I}_{j-1}$ gives us $f(x,u) \in \mathcal{X}$. ∎

The result of proposition 2 is powerful; this allows us to consider the continuous set (13) instead of the discrete set (12) while still retaining the property of control invariance in the space of output sequences $\mathbf{y_t}$ with each pair $(\mathbf{y}_t, \mathbf{y}_{t+1})$ (equivalently, $\mathbf{Y}_t$) corresponding to state-input pairs within constraints. We use this continuous set for our MPC problem in Section 4.4 to get a NLP instead of a MINLP.

4.3 Convex Terminal Cost

Now we proceed to construct a terminal cost function which approximates the optimal cost-to-go from a state using lifted outputs from previous iterations. For some iteration i and some time t, we define the cost-to-go for points in $\mathcal{SS}_{\mathbf{y}}^{j-1}$ as

$$\mathcal{C}_t^i = \sum_{k \geq t} C(\mathbf{y}_k^i) \tag{21}$$

where the function $C(\cdot)$ is convex, continuous and satisfies

$$C(\mathbf{y}_F) = 0,\; C(\mathbf{y}) \succ 0 \;\forall \mathbf{y} \in \mathbb{R}^{m \times R} \backslash \{\mathbf{y}_F\}. \tag{22}$$

Observe that since each $\mathbf{y} \in \mathcal{SS}_{\mathbf{y}}^{j-1}$ corresponds to a unique x via (9), $C(\cdot)$ is an implicit function of state. If $c(\cdot)$

from (4) is only a function of state and $c \circ \mathcal{F}_x(\cdot)$ is continuous, convex and positive definite, we can define $C(\cdot)$ by composing $c(\cdot)$ with $\mathcal{F}_x(\cdot)$, i.e. $C(\cdot) = c \circ \mathcal{F}_x(\cdot)$. We address the case of input costs in Section 5 of Nair et al. (2020). For iteration j, we use (21) to construct the terminal cost on the convex safe set $\mathcal{CS}_{\mathbf{y}}^{j-1}$ using Barycentric interpolation with tuples $(\mathbf{y}_t^i, \mathcal{C}_t^i), \forall \mathbf{y}_t^i \in \mathcal{SS}_{\mathbf{y}}^{j-1}$.

$$
\begin{aligned}
Q^{j-1}(\mathbf{y}) = \min_{\substack{\lambda_k^i \in [0,1] \\ \forall i \in \mathcal{I}_{j-1}}} \quad & \sum_{i \in \mathcal{I}_{j-1}} \sum_{k \geq 0} \lambda_k^i \mathcal{C}_k^i \\
\text{s.t.} \quad & \sum_{i \in \mathcal{I}_{j-1}} \sum_{k \geq 0} \lambda_k^i \mathbf{y}_k^i = \mathbf{y}, \ \sum_{i \in \mathcal{I}_{j-1}} \sum_{k \geq 0} \lambda_k^i = 1
\end{aligned} \tag{23}
$$

For $\mathbf{y}_F = [y_F, \ldots, y_F]$, we set $Q(\mathbf{y}_F) = 0$ and for any $\mathbf{y} \notin \mathcal{CS}_{\mathbf{y}}^{j-1} \cup \{\mathbf{y}_F\}$, we set $Q^{j-1}(\mathbf{y}) = +\infty$. The following proposition identifies CLF-like characteristics of the function given by (23) on the set $\mathcal{CS}_{\mathbf{y}}^{j-1}$.

Proposition 3. The cost function $Q^{j-1}(\cdot)$ satisfies the following properties:

(1) $Q^{j-1}(\mathbf{y}_F) = 0, \ Q^{j-1}(\mathbf{y}) \succ 0 \ \forall \mathbf{y} \in \mathcal{CS}_{\mathbf{y}}^{j-1} \backslash \{\mathbf{y}_F\}$

(2) $Q^{j-1}(\mathbf{y}_{t+1}) - Q^{j-1}(\mathbf{y}_t) \leq -C(\mathbf{y}_t), \ \forall \mathbf{y}_t \in \mathcal{CS}_{\mathbf{y}}^{j-1}$ where $\mathbf{y}_{t+1} = \delta(\mathbf{y}_t, y_{t+R})$ as in (14).

Proof. Please refer Nair et al. (2020). ■

The above proposition shows that $Q^{j-1}(\cdot)$ is in fact a CLF for the dynamics $\mathbf{y}_{t+1} = \delta(\mathbf{y}_t, y_{t+R})$ with input y_{t+R} on the convex output safe set $\mathcal{CS}_{\mathbf{y}}^{j-1}$.

4.4 LMPC Feedback Policy

In this section, we show how to use constructions (13) and (23) to design our LMPC policy. Before doing so, as in Rosolia and Borrelli (2017) we make the following assumption to initialise our recursive construction (7) of $\mathcal{SS}_{\mathbf{y}}^{j-1}$.

Assumption 4. We are provided with an output trajectory $\{y_t^0\}_{t \geq 0}$ corresponding to a state-input trajectory of system (1) satisfying constraints and with bounded cost-to-go, i.e.,

$$
\bigcup_{t \geq 0} \{(x_t^0, u_t^0)\} \subset \mathcal{X} \times \mathcal{U}, \ y_t^0 = h(x_t^0), \ \forall t \geq 0,
$$
$$
\mathcal{C}_0^0 = \sum_{t \geq 0} C(\mathbf{y}_t^0) < \infty.
$$

Using Assumption 4, the Output Safe Set (12) is initialised for $j = 1$ as $\mathcal{SS}_{\mathbf{y}}^0 = \bigcup_{t=0}^{\infty} \{\mathbf{y}_t^0\}$ where $\mathbf{y}_t^0 = [y_t^0, \ldots, y_{t+R-1}^0]$.

At iteration $j \geq 1$, we define the terminal cost on the space $\mathcal{CS}_{\mathbf{y}}^{j-1}$ as $Q^{j-1}(\cdot)$ and constrain the terminal state as $x_{N|t} = \mathcal{F}_x(\mathbf{y})$ for $\mathbf{y} \in \mathcal{CS}_{\mathbf{y}}^{j-1}$. The stage cost is set as $C(\cdot)$ which implicitly penalises only state(see Nair et al. (2020) for input costs). Like the forward-shift operator (14), we define the backward-time shift operator as

$$
\begin{aligned}
\mathbf{y}_t &= [y_t, \ldots, y_{t+R-1}] \\
&= \delta^-([y_{t+1}, y_{t+1}, \ldots, y_{t+R}], y_t) \\
&= \delta^-(\mathbf{y}_{t+1}, y_t)
\end{aligned} \tag{24}
$$

Employing these definitions, the LMPC optimization problem is given by

$$
\begin{aligned}
J_{t \to t+N}^j(x_t^j) = \min_{\boldsymbol{u}_t^j} \quad & Q^{j-1}(\mathbf{y}_{N|t}) + \sum_{k=0}^{N-1} C(\mathbf{y}_{k|t}) \\
\text{s.t.} \quad & x_{k+1|t} = f(x_{k|t}, u_{k|t}), \\
& \mathbf{y}_{k|t} = \delta^-(\mathbf{y}_{k+1|t}, h(x_{k|t})) \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \\
& x_{N|t} = \mathcal{F}_x(\mathbf{y}_{N|t}), \ \mathbf{y}_{N|t} \in \mathcal{CS}_{\mathbf{y}}^{j-1}, \\
& x_{0|t} = x_t^j, \\
& \forall k \in \{0, \ldots, N-1\}
\end{aligned} \tag{25}
$$

where the vector $\boldsymbol{u}_t^j = [u_{0|t}, \ldots, u_{N-1|t}]$ are the decision variables whose optimal solution defines the LMPC control as

$$
u_t^j = \pi_{LMPC}(x_t^j) = u_0^*. \tag{26}
$$

Notice that the above control policy is well-defined for all state x_0 for which problem (25) is feasible. Thus, we define the region of attraction

$$
\mathcal{R}^j = \{x \in \mathcal{X} | J_{t \to t+N}^j(x) < \infty\}, \tag{27}
$$

which collects the states from which problem (25) is feasible. By assumption 4, we readily have $\cup_{t=0}^{\infty} \{\mathcal{F}_x(\mathbf{y}_t^0)\} \subseteq \mathcal{R}^1$. We can show that for all states $x_0^j \in \mathcal{R}^j$ the closed-loop system is stable and it satisfies the state and input constraints. In the interest of space, we request the reader to refer to Nair et al. (2020) for an analysis of the properties of system (1) in closed-loop with (26).

5. NUMERICAL EXAMPLE

In this section, we demonstrate our approach on the kinematic unicycle. We urge the reader to refer to Nair et al. (2020) for our experiments on a PWA system and a Bilinear DC motor.

5.1 Kinematic Unicycle

Consider the following kinematic unicycle model with state $x_k = [X_k \ Y_k \ \theta_k]^\top$, controls $u_k = [v_k \ w_k]^\top$ and discretization step $dt = 0.1$ s

$$
\begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \\ \theta_k \end{bmatrix} + dt \begin{bmatrix} v_k \cos(\theta_k) \\ v_k \sin(\theta_k) \\ w_k \end{bmatrix} \tag{28}
$$

The lifted output and associated maps are given by

$$
y_k = \begin{bmatrix} X_k \\ Y_k \end{bmatrix}, \ \mathbf{Y}_k = [y_k, y_{k+1}, y_{k+2}] \tag{29}
$$

$$
\mathcal{F}_x(y_k, y_{k+1}) = \begin{bmatrix} y_k \\ \tan^{-1}\left(\dfrac{\begin{bmatrix} 0 & 1 \end{bmatrix} (y_{k+1} - y_k)}{\begin{bmatrix} 1 & 0 \end{bmatrix} (y_{k+1} - y_k)} \right) \end{bmatrix} \tag{30}
$$

$$
\mathcal{F}_u(y_k, y_{k+1}, y_{k+2}) = \begin{bmatrix} v_k \\ w_k \end{bmatrix}
$$

$$
v_k = \frac{1}{dt} \|y_{k+1} - y_k\|_2 \tag{31}
$$

$$
w_k = \begin{bmatrix} 0 & 0 & \frac{1}{dt} \end{bmatrix} \left(\mathcal{F}_x(y_{k+1}, y_{k+2}) - \mathcal{F}_x(y_k, y_{k+1}) \right) \tag{32}
$$

From (30), we see that $\mathcal{F}_x(\cdot)$ is linear in its first two components and monotonic in the third component (composition of monotonic and quasilinear map Boyd and Vandenberghe

(2004)). For speed input v_k, (31) is quasiconvex (because it is in fact, convex) and doesn't require quasiconcavity because speed is always positive. The state and input constraints are given by $\mathcal{X} = \{(X, Y) \in \mathbb{R}^2 | (X \geq 0) \wedge (Y \leq 10) \wedge (X - Y \leq 2)\} \times \{\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]\}$, $\mathcal{U} = [0, 5] \times \mathbb{R}$. To address the discontinuity in (30), the constraint on θ is tightened to $[-\frac{\pi}{2} + \epsilon, \frac{\pi}{2} - \epsilon]$. To steer the unicycle to the position $(5, 10)$, we minimize the sum of convex stage costs $c(x_k, u_k) = 20(X_k - 5)^2 + 20(Y_k - 10)^2 + v_k^2$. This is transformed for our MPC stage cost over a prediction horizon $N = 5$ in (25) as $C(\mathbf{y}_k) = 20\|y_k - [5\ 10]^\top\|_2^2 + \frac{1}{dt^2}\|y_{k+1} - y_k\|_2^2$. This cost is convex in $\mathbf{y}$ because of convexity of (31). The optimization problem (25) with the terminal set and terminal cost constructed as in (13) and (23) respectively is an NLP, solved using fmincon in MATLAB.

We see that the proposed controller successfully steers the unicycle (28) to the position $(5, 10)$ (Figure 2), while meeting state constraints and input constraints (Figure 3). The trajectory costs $J^j_{0\to\infty} = \sum_{k\geq 0} \tilde{C}(\tilde{\mathbf{y}}^j_k)$ are non-increasing with iteration j as is evident in Table 1.

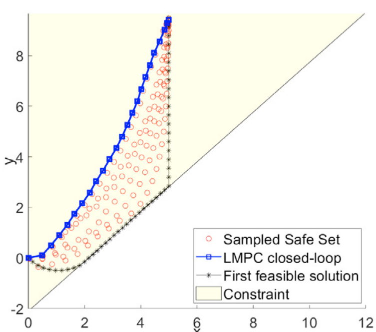

Fig. 2. Unicycle (28) trajectories in closed-loop with LMPC. The final trajectory is indicated in blue.

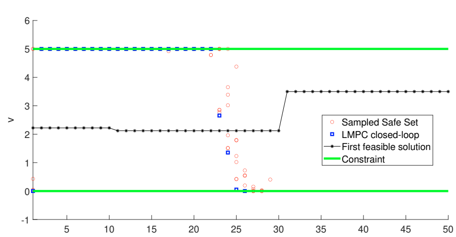

Fig. 3. Speed profile over time across iterations. The first trajectory takes 50 steps while the final trajectory takes 26 steps.

Iteration	0	1	2	3	4
Cost$\times 10^{-4}$	6.4931	2.8536	2.8088	2.7301	2.6944
Iteration	5	6	7	8	9
Cost$\times 10^{-4}$	2.6624	2.5695	2.4971	2.4515	2.4343

Table 1. Iteration Costs of closed-loop system trajectories

REFERENCES

Aranda-Bricaire, E., Kotta, Ü., and Moog, C. (1996). Linearization of discrete-time systems. *SIAM Journal on Control and Optimization*, 34(6), 1999–2023.

Bellman, R. (1966). Dynamic programming. *Science*, 153(3731), 34–37.

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

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Bristow, D.A., Tharayil, M., and Alleyne, A.G. (2006). A survey of iterative learning control. *IEEE Control Systems*, 26(3), 96–114.

Cueli, J.R. and Bordons, C. (2008). Iterative nonlinear model predictive control. stability, robustness and applications. *Control Engineering Practice*, 16(9), 1023–1034.

De Doná, J., Suryawan, F., Seron, M., and Lévine, J. (2009). A flatness-based iterative method for reference trajectory generation in constrained nmpc. In *Nonlinear Model Predictive Control*, 325–333. Springer.

Greeff, M. and Schoellig, A.P. (2018). Flatness-based model predictive control for quadrotor trajectory tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6740–6745. IEEE.

Guillot, P. and Millerioux, G. (2019). Flatness and submersivity of discrete-time dynamical systems. *IEEE Control Systems Letters*, 4(2), 337–342.

Kandler, C., Ding, S.X., Koenings, T., Weinhold, N., and Schultalbers, M. (2012). A differential flatness based model predictive control approach. In *2012 IEEE International Conference on Control Applications*, 1411–1416. IEEE.

Kwakernaak, H. and Sivan, R. (1972). *Linear optimal control systems*, volume 1. Wiley-interscience New York.

Lee, K.S. and Lee, J.H. (1997). Model predictive control for nonlinear batch processes with asymptotically perfect tracking. *Computers & Chemical Engineering*, 21, S873–S879.

Mayne, D.Q., Kerrigan, E.C., Van Wyk, E., and Falugi, P. (2011). Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11), 1341–1353.

Nair, S.H., Rosolia, U., and Borrelli, F. (2020). Output-lifted learning model predictive control. *arXiv preprint arXiv:2004.05173*.

Pontryagin, L.S. (2018). *Mathematical theory of optimal processes*. Routledge.

Rosolia, U. and Borrelli, F. (2017). Learning model predictive control for iterative tasks. a data-driven control framework. *IEEE Transactions on Automatic Control*, 63(7), 1883–1896.

Rosolia, U. and Borrelli, F. (2020). Minimum time learning model predictive control. *International Journal of Robust and Nonlinear Control*.

Wang, Z., Zha, J., and Wang, J. (2019). Flatness-based model predictive control for autonomous vehicle trajectory tracking. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 4146–4151. IEEE.

Yong, S.Z., Paden, B., and Frazzoli, E. (2015). Computational methods for mimo flat linear systems: Flat output characterization, test and tracking control. In *2015 American Control Conference (ACC)*, 3898–3904.