Multi-Rate Planning and Control of Uncertain Nonlinear Systems: Model Predictive Control and Control Lyapunov Functions

Noel Csomay-Shanklin*, Andrew J. Taylor*, Ugo Rosolia, Aaron D. Ames

Abstract—**Modern control systems must operate in increasingly complex environments subject to safety constraints and input limits, and are often implemented in a hierarchical fashion with different controllers running at multiple time scales. Yet traditional constructive methods for nonlinear controller synthesis typically "flatten" this hierarchy, focusing on a single time scale, and thereby limited the ability to make rigorous guarantees on constraint satisfaction that hold for the entire system. In this work we seek to address the stabilization of constrained nonlinear systems through a *multi-rate* control architecture. This is accomplished by iteratively planning continuous reference trajectories for a nonlinear system using a linearized model and Model Predictive Control (MPC), and tracking said trajectories using the full-order nonlinear model and Control Lyapunov Functions (CLFs). Connecting these two levels of control design in a way that ensures constraint satisfaction is achieved through the use of *Bézier curves*, which enable planning continuous trajectories respecting constraints by planning a sequence of discrete points. Our framework is encoded via convex optimization problems which may be efficiently solved, as demonstrated in simulation.**

I. Introduction

The study and design of nonlinear control systems has long been framed through the lens of stabilization, often in an optimal sense. This is coupled with the fact that one typically considers a single model, implicitly representing a single time scale. However in most modern engineering settings, especially in the context of autonomous and robotic systems, the task of stabilization is complicated by the need to meet safety-critical constraints on the system's state while respecting input limitations. To address this need, implementations often utilize a hierarchical approach that spans multiple time-scales, from the planning layer—which typically leverages discrete-time models—to the real-time controller layer which often considers continuous-time representations. Thus, it is necessary to develop efficient control synthesis techniques that provide rigorous guarantees of stability, even in the presence of such constraints, and across multiple time scales.

At the level of real-time control design, a rich catalog of methods have been developed for stabilizing nonlinear systems in the presence of unknown disturbances by utilizing underlying structural properties of the system [1]–[4]. In particular, the tools of Control Lyapunov Functions (CLFs) [5], [6] and Input-to-State Stability [7] have enabled the joint synthesis of stabilizing controllers and Lyapunov certificates of stability in the presence of disturbances, including through convex optimization [8]–[10]. These methods for stabilization yield highly structured controllers, and modifying these designs to accommodate state and input constraints may destroy the stability properties guaranteed by the controller. This issue is often circumnavigated theoretically by limiting the domain on which stability is guaranteed, effectively ignoring constraints.

This work is supported by the National Science Foundation (CPS Award #1932091, NRI Award #1924526, CMMI Award #1923239)), and the AFOSR Test and Evaluation Program (FA9550-19-1-0302).

* Authors contributed equally. N. Csomay-Shanklin, A. J. Taylor, U. Rosolia, and A. D. Ames are with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA, {noelcs, ajtaylor, urosolia, ames}@caltech.edu.

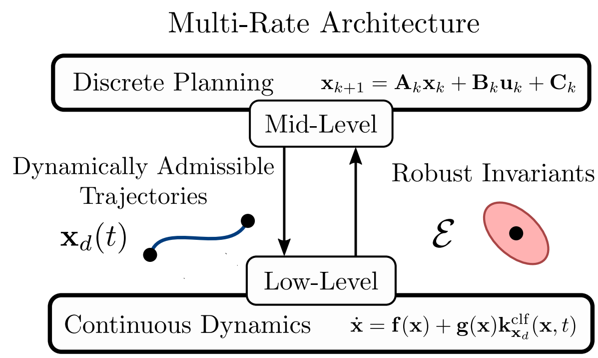

Fig. 1. Overview of Multi-Rate Architecture, with discrete planning producing reference trajectories at a mid-level and continuous controllers producing invariant sets at a low-level.

In contrast, Model Predictive Control (MPC) provides an effective method for addressing constraints [11]–[13]. This is achieved by directly incorporating constraints a into controller that iteratively plans a finite sequence of states and inputs that are related through a discrete model of the system dynamics and satisfy required constraints. Although MPC has been successfully demonstrated in several challenging control settings [14]–[24], it is rarely implemented in real-time using the full-order continuous time nonlinear dynamics while accounting for unknown disturbances acting on the system. Thus, MPC implementations for nonlinear systems usually lack strong theoretical guarantees on constraint satisfaction in the presence of disturbances. This is because (i) it is typically difficult to find a closed-form expression for the exact temporal discretization of continuous time nonlinear dynamics [25], (ii) approximating the exact discretization through numerical integration typically yields a non-convex relationship between planned states and inputs, and (iii) exactly propagating disturbances through high-dimensional nonlinear dynamics is often computationally intractable [26]. These challenges often preclude the computational efficiency needed for real-time implementation.

The difficulty in realizing MPC based controllers at a fast enough rate to allow for real-time implementation is often resolved by using an approximate model of the system dynamics that is amenable to efficient planning, typically through reduced-order models or via linearization and temporal discretization of the continuous time nonlinear system dynamics [13], [22]–[24], [27]. The use of such approximations creates a gap between the system which is being planned for and the actual evolution of the nonlinear system, requiring an additional measure of robustness to ensure constraint satisfaction. This robustness is often achieved by tightening the constraint sets by the maximum deviation between the approximate model and the continuous time nonlinear system dynamics [28]–[35]. Approximating worse-case deviations is typically done using properties of the dynamics which may be difficult to compute, such as Lipschitz constants for which over-approximations yield conservativeness, or by solving computationally intensive optimization programs. More recently, hierarchical control frameworks have been proposed that plan with an approximate model and address nonlinear dynamics with a low-level controller [36], [37]. However, this work does not address if the low-level controller respects state and input constraints as it follows the planned trajectory under disturbances.

In this work we propose a novel multi-rate control architecture that unifies the planning capabilities of Model Predictive Control with the ability to directly address nonlinear dynamics provided by Control Lyapunov Functions. The fundamental tool that allows our framework to explicitly address the relationship between a planner and controller operating at different time scales are *Bézier curves* [38], [39]. By directly planning over the *control points* that parameterize Bézier curves, we capitalize on a critical convex hull property to ensure that state and input constraints are met by the nonlinear system evolving under an optimization-based CLF controller. While Bézier curves have been used in motion planning, or to verify constraint satisfaction after solving an MPC problem [40], this is to the best of our knowledge the first result directly planning over Bézier control points in an MPC formulation, and using the resulting continuous trajectories to ensure constraint satisfaction for a nonlinear system with disturbances.

We begin in Section II by reviewing nonlinear dynamics, and how structural properties can be used to synthesize CLFs and optimization-based controllers for stabilizing a class of *dynamically admissible* reference trajectories. These controllers yield a description of how accurately a reference trajectory is tracked in the presence of disturbances that is amenable to being incorporated into planning. Next, in Section III we provide a review of Bézier curves, and show how they may be used to synthesize reference trajectories for the disturbed nonlinear system such that state constraints are satisfied. Section IV uses the properties of Bézier curves in conjunction with the structure of the low-level controller to formulate constraints on Bézier control points that ensure the low-level controller satisfies input constraints. In Section V we integrate the preceding constructions into an MPC formulation that plans over Bézier control points and synthesizes continuous reference trajectories using a locally linearized and discretized model while ensuring recursive feasibility. We conclude in Section VI with simulation results. Space constraints prevent the inclusion of proofs; these can be found in the extended version [41].

II. Low-Level Controller Design

In this section we review nonlinear dynamical systems and discuss the design of nonlinear feedback controllers that provide a measure of disturbance rejection. Importantly, these controllers will yield a description of reference trajectory tracking that is amenable to being directly incorporated into the synthesis of the reference trajectory itself.

Consider the nonlinear control-affine system:

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ 0 & \mathbf{0}^\top \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ f(\mathbf{x}) \end{bmatrix}}_{\mathbf{f}(\mathbf{x})} + \underbrace{\begin{bmatrix} \mathbf{0} \\ g(\mathbf{x}) \end{bmatrix}}_{\mathbf{g}(\mathbf{x})} u + \mathbf{w}(t), \tag{1}$$

with state $\mathbf{x} \in \mathbb{R}^n$, input $u \in \mathbb{R}$, piecewise continuous[1] disturbance signal $\mathbf{w} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, and functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$, assumed to be continuously differentiable on $\mathbb{R}^n$. Furthermore, we make the following assumption:

Assumption 1. The function f satisfies $f(\mathbf{0}) = 0$ and the function g satisfies $g(\mathbf{x}) \neq 0$ for all $\mathbf{x} \in \mathbb{R}^n$.

The first assumption takes the origin to be an unforced equilibrium point of the undisturbed system. The second assumption amounts to the system (1) possessing a relative degree [1]. We note that while we consider a single-input, single-output system, this is purely to simplify the presentation of our contributions, with our results being easily extended to the multiple-input, multiple-output setting under an equivalent assumption of a vector relative degree.

Let $\underline{t}, \overline{t} \in \mathbb{R}_{\geq 0}$ with $\underline{t} < \overline{t}$, and let $k : \mathbb{R}^n \times [\underline{t}, \overline{t}] \to \mathbb{R}$ be a feedback controller that is locally Lipschitz continuous with respect to its first argument[2] and piecewise continuous with respect to its second argument on $\mathbb{R}^n \times [\underline{t}, \overline{t}]$. This controller yields the closed-loop system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}) k(\mathbf{x}, t) + \mathbf{w}(t). \tag{2}$$

As the functions $\mathbf{f}$, $\mathbf{g}$, and k are locally Lipschitz continuous with respect to $\mathbf{x}$ and k is piecewise continuous with respect to t, for any initial condition $\mathbf{x}_0 \in \mathbb{R}^n$ and any piecewise continuous disturbance $\mathbf{w} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$, there exists an interval $I(\underline{t}, \mathbf{x}_0, \mathbf{w}) \triangleq [\underline{t}, \underline{t} + \delta(\mathbf{x}_0, \mathbf{w}))$ with $\delta(\mathbf{x}_0, \mathbf{w}) \in \mathbb{R}_{>0}$ such that the system (2) has a unique piecewise continuously differentiable[3] solution $\boldsymbol{\varphi} : I(\underline{t}, \mathbf{x}_0, \mathbf{w}) \to \mathbb{R}^n$ satisfying:

$$\dot{\boldsymbol{\varphi}}(t) = \mathbf{f}(\boldsymbol{\varphi}(t)) + \mathbf{g}(\boldsymbol{\varphi}(t)) k(\boldsymbol{\varphi}(t), t) + \mathbf{w}(t), \tag{3}$$

$$\boldsymbol{\varphi}(\underline{t}) = \mathbf{x}_0, \tag{4}$$

[1] This definition is taken as in [3], with piecewise continuity requiring the existence of one-sided limits at points of discontinuity.

[2] This definition is taken as in [3], with local Lipschitz continuity holding with a Lipschitz constant that is uniform in the function's second argument.

[3] Piecewise continuous differentiability is taken to mean a continuous function with a derivative defined on the open intervals of a finite partition with one-sided limits.

for almost all $t \in I(\underline{t}, \mathbf{x}_0, \mathbf{w})$ [3].

With a view towards controller design, the system (1) may also be used to define a class of reference trajectories:

Definition 1 (*Dynamically Admissible Trajectory*)**.** A piecewise continuously differentiable function $\mathbf{x}_d : [\underline{t}, \overline{t}] \to \mathbb{R}^n$ is a *dynamically admissible trajectory* for the system (1) if there is a piecewise continuous $u_d : [\underline{t}, \overline{t}] \to \mathbb{R}$ such that:

$$\dot{\mathbf{x}}_d(t) = \mathbf{f}(\mathbf{x}_d(t)) + \mathbf{g}(\mathbf{x}_d(t))u_d(t), \tag{5}$$

for almost all $t \in [\underline{t}, \overline{t}]$.

Given a dynamically admissible trajectory $\mathbf{x}_d : [\underline{t}, \overline{t}] \to \mathbb{R}^n$ for (1), let us denote: $\dot{\mathbf{x}}_d(t) = \begin{bmatrix} \dot{x}_d^1(t) & \cdots & \dot{x}_d^n(t) \end{bmatrix}^\top$, and define a error function $\mathbf{e}_{\mathbf{x}_d} : \mathbb{R}^n \times [\underline{t}, \overline{t}] \to \mathbb{R}^n$:

$$\mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t) = \mathbf{x} - \mathbf{x}_d(t), \tag{6}$$

and its derivative $\dot{\mathbf{e}}_{\mathbf{x}_d} : \mathbb{R}^n \times [\underline{t}, \overline{t}] \times \mathbb{R} \to \mathbb{R}^n$ as:

$$\dot{\mathbf{e}}_{\mathbf{x}_d}(\mathbf{x}, t, u) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u + \mathbf{w}(t) - \dot{\mathbf{x}}_d(t). \tag{7}$$

Denoting:

$$\mathcal{F}_{\mathbf{x}_d}(\mathbf{x}, t) = f(\mathbf{x}) - \dot{x}_d^n(t), \tag{8}$$

the structure of the system (1) implies that:

$$\dot{\mathbf{e}}_{\mathbf{x}_d}(\mathbf{x}, t, u) = \overbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ 0 & \mathbf{0}^\top \end{bmatrix} \mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t) + \begin{bmatrix} \mathbf{0} \\ \mathcal{F}_{\mathbf{x}_d}(\mathbf{x}, t) \end{bmatrix}}^{\mathbf{f}_{\mathbf{x}_d}(\mathbf{x},t)} + \mathbf{g}(\mathbf{x})u + \mathbf{w}(t). \tag{9}$$

This structure with the assumption that $g(\mathbf{x}) \neq 0$ for any $\mathbf{x} \in \mathbb{R}^n$ enables a controller $k_{\mathbf{x}_d}^{\text{fbl}} : \mathbb{R}^n \times [\underline{t}, \overline{t}] \to \mathbb{R}$:

$$k_{\mathbf{x}_d}^{\text{fbl}}(\mathbf{x}, t) = g(\mathbf{x})^{-1} \left(-\mathcal{F}_{\mathbf{x}_d}(\mathbf{x}, t) - \mathbf{K}^\top \mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t) \right), \tag{10}$$

where $\mathbf{K} \in \mathbb{R}^n$ is selected to yield the relationship:

$$\dot{\mathbf{e}}_{\mathbf{x}_d}(\mathbf{x}, t, k_{\text{fbl}}(\mathbf{x}, t)) = \mathbf{F}\mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t) + \mathbf{w}(t), \tag{11}$$

with $\mathbf{F} \in \mathbb{R}^{n \times n}$ a Hurwitz matrix. For any $\mathbf{Q} \in \mathbb{S}^n_{\succ 0}$ (symmetric positive definite matrices) there exists a unique $\mathbf{P} \in \mathbb{S}^n_{\succ 0}$ solving the Continuous Time Lyapunov Equation:

$$\mathbf{F}^\top \mathbf{P} + \mathbf{P}\mathbf{F} = -\mathbf{Q}. \tag{12}$$

For a particular $\mathbf{Q}$, the corresponding solution $\mathbf{P}$ may be used to define a function $V_{\mathbf{x}_d} : \mathbb{R}^n \times [\underline{t}, \overline{t}] \to \mathbb{R}_{\geq 0}$:

$$V_{\mathbf{x}_d}(\mathbf{x}, t) = \mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t)^\top \mathbf{P} \mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t). \tag{13}$$

Denoting $\nabla V_{\mathbf{x}_d}(\mathbf{x}, t) = 2\mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t)^\top \mathbf{P}$, we have that:

$$\lambda_{\min}(\mathbf{P})\|\mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t)\|_2^2 \leq V_{\mathbf{x}_d}(\mathbf{x}, t) \leq \lambda_{\max}(\mathbf{P})\|\mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t)\|_2^2, \tag{14}$$

$$\nabla V_{\mathbf{x}_d}(\mathbf{x}, t)(\mathbf{f}_{\mathbf{x}_d}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x})k_{\mathbf{x}_d}^{\text{fbl}}(\mathbf{x}, t)) \leq -\lambda_{\min}(\mathbf{Q})\|\mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t)\|_2^2, \tag{15}$$

for all $\mathbf{x} \in \mathbb{R}^n$ and $t \in [\underline{t}, \overline{t}]$. Let $\gamma = 4\lambda_{\max}(\mathbf{P})^3/\lambda_{\min}(\mathbf{Q})^2$ and for a given disturbance signal $\mathbf{w} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ define $\|\mathbf{w}\|_\infty = \sup_{t \geq 0} \|\mathbf{w}(t)\|_2$. The preceding construction yields the following result:

Lemma 1. *Let $\overline{w} \in \mathbb{R}_{\geq 0}$, and for $t \in [\underline{t}, \overline{t}]$ define the set:*

$$\Omega_{\mathbf{x}_d}(t, \overline{w}) = \{\mathbf{x} \in \mathbb{R}^n \mid V_{\mathbf{x}_d}(\mathbf{x}, t) \leq \gamma \overline{w}^2\}. \tag{16}$$

Let the controller $k : \mathbb{R}^n \times [\underline{t}, \overline{t}] \to \mathbb{R}$ satisfy:

$$\nabla V_{\mathbf{x}_d}(\mathbf{x}, t)(\mathbf{f}_{\mathbf{x}_d}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x})k(\mathbf{x}, t)) \leq -\lambda_{\min}(\mathbf{Q})\|\mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t)\|_2^2, \tag{17}$$

for almost all $t \in [\underline{t}, \overline{t}]$ and all $\mathbf{x} \in \Omega_{\mathbf{x}_d}(t, \overline{w})$. Then for initial time $\underline{t}$, any $\mathbf{x}_0 \in \Omega_{\mathbf{x}_d}(\underline{t}, \overline{w})$, and any $\mathbf{w}$ satisfying $\|\mathbf{w}\|_\infty \leq \overline{w}$, we have that $I(\underline{t}, \mathbf{x}_0, \mathbf{w}) = [\underline{t}, \overline{t})$, and $\boldsymbol{\varphi}(t) \in \Omega_{\mathbf{x}_d}(t, \overline{w})$ for all $t \in [\underline{t}, \overline{t})$, and $\lim_{t \to \overline{t}} \boldsymbol{\varphi}(t)$ exists and satisfies $\lim_{t \to \overline{t}} \boldsymbol{\varphi}(t) \in \Omega_{\mathbf{x}_d}(\overline{t}, \overline{w})$.

The preceding result follows by a standard input-to-state stability argument [7]. For any $t \in [\underline{t}, \overline{t}]$, the set $\Omega_{\mathbf{x}_d}(t, \overline{w})$ captures how accurately the nonlinear closed-loop system (2) tracks $\mathbf{x}_d$ under disturbances. Importantly, for a given $t \in [\underline{t}, \overline{t}]$ the set $\Omega_{\mathbf{x}_d}(t, \overline{w})$ is convex – as we will see later, this will allow us to efficiently synthesize a dynamically admissible trajectory $\mathbf{x}_d$ knowing how accurately it will be tracked and ensuring state and input constraint satisfaction.

In contrast to cancelling the nonlinear dynamics to achieve linear dynamics as in (11), which may be unnecessary and inefficient [8], Control Lyapunov Functions (CLFs) provide a method for synthesizing stabilizing controllers via convex optimization. In particular, (15) implies that:

$$\inf_{u \in \mathbb{R}} \nabla V_{\mathbf{x}_d}(\mathbf{x}, t)(\mathbf{f}_{\mathbf{x}_d}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x})u) \leq -\lambda_{\min}(\mathbf{Q})\|\mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t)\|_2^2. \tag{18}$$

for all $\mathbf{x} \in \mathbb{R}^n$ and $t \in [\underline{t}, \overline{t}]$. Define a feed-forward controller $k_{\mathbf{x}_d}^{\text{ff}} : \mathbb{R}^n \times [\underline{t}, \overline{t}] \to \mathbb{R}$ as:

$$k_{\mathbf{x}_d}^{\text{ff}}(\mathbf{x}, t) = -g(\mathbf{x})^{-1}\mathcal{F}_{\mathbf{x}_d}(\mathbf{x}, t). \tag{19}$$

This feed-forward controller is incorporated into a controller specified via a convex quadratic program (QP):

$$k_{\mathbf{x}_d}^{\text{clf}}(\mathbf{x}, t) = \underset{u \in \mathbb{R}}{\operatorname{argmin}} \frac{1}{2}\|u - k_{\mathbf{x}_d}^{\text{ff}}(\mathbf{x}, t)\|_2^2 \tag{CLF-QP}$$
$$\text{s.t. } \nabla V_{\mathbf{x}_d}(\mathbf{x}, t)(\mathbf{f}_{\mathbf{x}_d}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x})u) \leq -\lambda_{\min}(\mathbf{Q})\|\mathbf{e}_{\mathbf{x}_d}(\mathbf{x}, t)\|_2^2.$$

Note that the constraint in this controller ensures that $k_{\mathbf{x}_d}^{\text{clf}}$ satisfies the condition in (17).

III. Bézier Curves & State Constraints

In this section we present the first main contribution of this work by addressing how the properties of the low-level tracking controller can be used to place requirements on a dynamically admissible trajectory $\mathbf{x}_d$. These requirements ensure state constraint satisfaction by the closed-loop nonlinear system (2) using the controllers $k_{\mathbf{x}_d}^{\text{fbl}}$ and $k_{\mathbf{x}_d}^{\text{clf}}$.

We first make the following assumption regarding the state constraints for the system:

Assumption 2. The state constraint set $\mathcal{X} \subset \mathbb{R}^n$ is a compact, convex polytope, with the existence of $\mathbf{L}_j \in \mathbb{R}^n$

and $\ell_j \in \mathbb{R}$ for $j = 1, \ldots, q$ such that $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \forall j,\, \mathbf{L}_j^\top \mathbf{x} \leq \ell_j\}$. Furthermore, we have that $\mathbf{0} \in \text{Int}(\mathcal{X})$.

Given the above state constraints, it is not the case – even for a dynamically admissible trajectory satisfying $\mathbf{x}_d(t) \in \mathcal{X}$ for all $t \in [\underline{t}, \overline{t}]$ – that the state will remain inside the set $\mathcal{X}$, as we have that $\Omega_{\mathbf{x}_d}(t, \overline{w}) \nsubseteq \mathcal{X}$ for some $t \in [\underline{t}, \overline{t}]$. To ensure that these constraints are met by the closed-loop system without directly modifying the low-level control design, we will incorporate information about the low-level controller when constructing $\mathbf{x}_d$. The tool that will enable incorporating this information is *Bézier curves* [38].

Let $T \in \mathbb{R}_{>0}$. A Bézier curve $r : [0, T] \to \mathbb{R}$ of order p is defined as:

$$r(\tau) = \boldsymbol{\xi}_0^\top \mathbf{z}(\tau), \tag{20}$$

where $\boldsymbol{\xi}_0 = \begin{bmatrix} \xi_{0,0} & \ldots & \xi_{0,p} \end{bmatrix}^\top \in \mathbb{R}^{p+1}$ is a vector with elements consisting of the $p+1$ *control points*, $\xi_{0,i} \in \mathbb{R}$, of the curve and $\mathbf{z} : [0, T] \to \mathbb{R}^{p+1}$ is a Bernstein polynomial defined elementwise as:

$$z_i(\tau) = \binom{p}{i} \left(\frac{\tau}{T}\right)^i \left(1 - \frac{\tau}{T}\right)^{p-i}, \quad i = 0, \ldots, p. \tag{21}$$

The curve r is smooth, and there exists a matrix[4] $\mathbf{H} \in \mathbb{R}^{p+1 \times p+1}$ such that the j^{th} derivative of r is given by:

$$r^{(j)}(\tau) = \frac{1}{T^j} \boldsymbol{\xi}_0^\top \mathbf{H}^j \mathbf{z}(\tau) \triangleq \boldsymbol{\xi}_j^\top \mathbf{z}(\tau). \tag{22}$$

Consequently, $r^{(j)} : [0, T] \to \mathbb{R}$ is a Bézier curve of order p with the elements of $\boldsymbol{\xi}_j$ (which are uniquely and linearly defined by $\boldsymbol{\xi}_0$) as control points. Define the function $\mathbf{r} : [0, T] \to \mathbb{R}^n$:

$$\mathbf{r}(\tau) = \begin{bmatrix} r(\tau) & r^{(1)}(\tau) & \cdots & r^{(n-1)}(\tau) \end{bmatrix}^\top. \tag{23}$$

There exists a matrix[4] $\mathbf{D} \in \mathbb{R}^{2n \times 2n}$ such that for any two vectors $\mathbf{x}_0, \mathbf{x}_1 \in \mathbb{R}^n$, the unique Bézier curve r of order $2n-1$ satisfying $\mathbf{r}(0) = \mathbf{x}_0$ and $\mathbf{r}(T) = \mathbf{x}_1$ with a vector of control points $\boldsymbol{\xi}_0 \in \mathbb{R}^{2n}$ is given by:

$$\boldsymbol{\xi}_0^\top = \begin{bmatrix} \mathbf{x}_0^\top & \mathbf{x}_1^\top \end{bmatrix} \mathbf{D}^{-1}. \tag{24}$$

The following result (proven in [41]) shows how a sequence of points may be used to construct a set of Bézier curves that yield a dynamically admissible trajectory for (1):

Lemma 2. *Let $N \in \mathbb{N}$, $\underline{t} \in \mathbb{R}_{\geq 0}$, and define $\overline{t} = \underline{t} + NT$. For $k = 0, \ldots, N$, consider a collection of points $\{\mathbf{x}_k\}$ with $\mathbf{x}_k \in \mathbb{R}^n$ and define $t_k \in \mathbb{R}_{\geq 0}$ as $t_k = \underline{t} + kT$. For $k = 0, \ldots, N-1$, let $r_k : [0, T] \to \mathbb{R}$ be a Bézier curve of order $2n-1$ with control points $(\boldsymbol{\xi}_k)_0 = \begin{bmatrix} (\xi_k)_{0,0} & \ldots & (\xi_k)_{0,2n-1} \end{bmatrix}^\top \in \mathbb{R}^{2n}$ given by:*

$$(\boldsymbol{\xi}_k)_0^\top = \begin{bmatrix} \mathbf{x}_k^\top & \mathbf{x}_{k+1}^\top \end{bmatrix} \mathbf{D}^{-1}. \tag{25}$$

Defining the functions $\mathbf{r}_k : [0, T] \to \mathbb{R}^n$ as in (23)*, we have that the function $\mathbf{x}_d : [\underline{t}, \overline{t}] \to \mathbb{R}^n$ defined as:*

$$\begin{aligned} \mathbf{x}_d(t) &= \mathbf{r}_k(t - t_k),\ t \in [t_k, t_{k+1}), \\ \mathbf{x}_d(\overline{t}) &= \mathbf{x}_N, \end{aligned} \tag{26}$$

is a dynamically admissible trajectory for the system (1).

[4]The matrices $\mathbf{H}$ and $\mathbf{D}$ are uniquely defined by the order of the Bézier curve p and can be constructed as shown in the extended version [41].

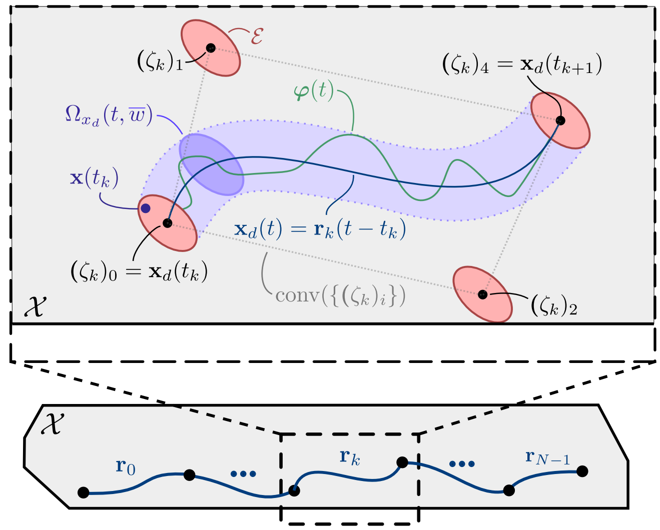

Fig. 2. A depiction of the proposed method, where the control points of the Bézier curve are constraint tightened by the size of the robust invariant tube coming from the low-level controller.

We note that the preceding result reduces planning of an (infinite dimensional) continuous time trajectory to planning a finite sequence of points. This aligns with planning dynamically admissible trajectories online in a multi-rate approach. While other classes of functions (such as general polynomials) may similarly be used to construct dynamically admissible trajectories for (1), the motivation for using Bézier curves lies in the convex hull relationship between the curve $\mathbf{r}_k$ and the control points $(\boldsymbol{\xi}_k)_0, \ldots, (\boldsymbol{\xi}_k)_{n-1}$. More precisely, for $i = 0, \ldots, 2n-1$ denote:

$$(\boldsymbol{\zeta}_k)_i \triangleq \begin{bmatrix} (\xi_k)_{0,i} & \ldots & (\xi_k)_{n-1,i} \end{bmatrix}^\top \in \mathbb{R}^n. \tag{27}$$

The points $(\boldsymbol{\xi}_k)_j$ can be viewed as the control points in time for the curve $r_k^{(j)}$, while $(\boldsymbol{\zeta}_k)_i$ reflects the control points for the curve $\mathbf{r}_k$ in the state space. This enables the following:

Fact 1 ([38] §4)**.** We have that $\mathbf{r}_k(\tau) \in \text{conv}(\{(\boldsymbol{\zeta}_k)_i\})$ for all $\tau \in [0, T]$.

We may immediately use this property to establish the following result (proven in [41]) regarding state constraints:

Lemma 3. *Define the convex, compact set $\mathcal{E} \subseteq \mathbb{R}^n$ as:*

$$\mathcal{E} = \{\mathbf{v} \in \mathbb{R}^n \mid \mathbf{v}^T \mathbf{P} \mathbf{v} \leq \gamma \overline{w}^2\}. \tag{28}$$

If $(\boldsymbol{\zeta}_k)_i \in \mathcal{X} \ominus \mathcal{E}$ for $i = 0, \ldots, 2n-1$ and $k = 0, \ldots, N-1$, then we have that $\Omega_{\mathbf{x}_d}(t, \overline{w}) \subseteq \mathcal{X}$ for all $t \in [\underline{t}, \overline{t}]$.

This result states that by constraining the Bézier curve control points, we can ensure the evolution of the system under the low-level controller satisfies state constraints. The requirement that $(\boldsymbol{\zeta}_k)_i \in \mathcal{X} \ominus \mathcal{E}$ can be expressed as an affine inequality constraint as in the following (proven in [41]):

Lemma 4. *We have that for $j = 1, \ldots, q$:*

$$(\boldsymbol{\zeta}_k)_i \in \mathcal{X} \ominus \mathcal{E} \Leftrightarrow \mathbf{L}_j^\top (\boldsymbol{\zeta}_k)_i \leq \ell_j - \sqrt{\gamma \overline{w}^2 \mathbf{L}_j^\top \mathbf{P}^{-1} \mathbf{L}_j}. \quad (29)$$

IV. Input Constraints

In this section, we present the second main contribution of this work. We show how the structure of a low-level tracking controller can be used to place requirements on a dynamically admissible trajectory $\mathbf{x}_d$ to ensure input constraint satisfaction. We will make the following assumption regarding input constraints for the system:

Assumption 3. The input constraint set $\mathcal{U} \subset \mathbb{R}$ is given by $\mathcal{U} = [-u_{\max}, u_{\max}]$ for some $u_{\max} \in \mathbb{R}_{>0}$.

Neither of the controllers $k_{\mathbf{x}_d}^{\mathrm{fbl}}$ or $k_{\mathbf{x}_d}^{\mathrm{clf}}$ are necessarily required to take values in the set $\mathcal{U}$. Thus, satisfying input constraints may require violating the inequality constraint in (17), potentially invalidating the claim that $\boldsymbol{\varphi}(t) \in \Omega_{\mathbf{x}_d}(t, \overline{w})$ for all $t \in [\underline{t}, \overline{t}]$. To address this limitation, knowledge of how much control action is required by the controller to track the reference trajectory under disturbances should be incorporated when synthesizing $\mathbf{x}_d$.

To this end, we state the following definitions. For $\alpha, \beta \in \mathbb{R}_{\geq 0}$, define the matrix $\mathbf{M}_{\alpha,\beta} \in \mathbb{S}^2_{\succeq 0}$ and the functions $\mathbf{N}_{\alpha,\beta} : \mathcal{X} \to \mathbb{R}^2_{\geq 0}$ and $\Gamma_{\alpha,\beta} : \mathcal{X} \to \mathbb{R}_{\geq 0}$ as:

$$\mathbf{M}_{\alpha,\beta} = \boldsymbol{\pi}_{\mathrm{PSD}} \left(\begin{bmatrix} 2\alpha\beta & \beta \\ \beta & 0 \end{bmatrix} \right), \quad (30)$$

$$\mathbf{N}_{\alpha,\beta}(\overline{\mathbf{x}}) = \begin{bmatrix} 2\alpha\beta\overline{e} + \alpha|g(\overline{\mathbf{x}})^{-1}| + \beta\|\mathbf{K}\|_2 \overline{e} \\ |g(\overline{\mathbf{x}})^{-1}| + \beta\overline{e} \end{bmatrix}, \quad (31)$$

$$\Gamma_{\alpha,\beta}(\overline{\mathbf{x}}) = \overline{e}(\beta\overline{e} + |g(\overline{\mathbf{x}})^{-1}|)(\alpha + \|\mathbf{K}\|_2), \quad (32)$$

where $\boldsymbol{\pi}_{\mathrm{PSD}} : \mathbb{S}^2 \to \mathbb{S}^2_{\succeq 0}$ denotes the projection from symmetric matrices to symmetric positive semidefinite matrices, $\overline{e} \triangleq \sqrt{\gamma \overline{w}^2 / \lambda_{\min}(\mathbf{P})}$, and $\mathbf{K}$ is defined in (10). Given these definitions, we state one of our main results (proven in [41]):

Theorem 1. *There exists constants $\underline{\alpha}, \underline{\beta} \in \mathbb{R}_{\geq 0}$ such that if $\alpha \geq \underline{\alpha}$, $\beta \geq \underline{\beta}$, and $\Omega_{\mathbf{x}_d}(t, \overline{w}) \subseteq \mathcal{X}$ for all $t \in [\underline{t}, \overline{t}]$, then for any collection of points $\{\overline{\mathbf{x}}_k\}$ with $\overline{\mathbf{x}}_k \in \mathcal{X}$ for $k = 0, \ldots, N-1$, we have that for all $t \in [t_k, t_{k+1})$:*

$$\begin{aligned} \|k_{\mathbf{x}_d}^{\mathrm{fbl}}(\mathbf{x}, t)\|_2 \leq \frac{1}{2} &\boldsymbol{\sigma}_{\mathbf{x}_d}(t)^\top \mathbf{M}_{\alpha,\beta} \boldsymbol{\sigma}_{\mathbf{x}_d}(t) \quad (33) \\ &+ \mathbf{N}_{\alpha,\beta}(\overline{\mathbf{x}}_k)^\top \boldsymbol{\sigma}_{\mathbf{x}_d}(t) + \Gamma_{\alpha,\beta}(\overline{\mathbf{x}}_k), \end{aligned}$$

for all $\mathbf{x} \in \Omega_{\mathbf{x}_d}(t, \overline{w})$, where $\boldsymbol{\sigma}_{\mathbf{x}_d} : [\underline{t}, \overline{t}] \to \mathbb{R}^2_{\geq 0}$ is defined as:

$$\boldsymbol{\sigma}_{\mathbf{x}_d}(t) = \begin{bmatrix} \|\mathbf{x}_d(t) - \overline{\mathbf{x}}_k\|_2 \\ \|\dot{x}_d^n(t) - f(\overline{\mathbf{x}}_k)\|_2 \end{bmatrix}, \quad t \in [t_k, t_{k+1}). \quad (34)$$

This result is motivated by the key observation that the upper bound achieved in (33) is convex in the quantity $\boldsymbol{\sigma}_{\mathbf{x}_d}(t)$ for each $t \in [\underline{t}, \overline{t}]$, such that the constraint:

$$\frac{1}{2}\boldsymbol{\sigma}_{\mathbf{x}_d}(t)^\top \mathbf{M}_{\alpha,\beta} \boldsymbol{\sigma}_{\mathbf{x}_d}(t) + \mathbf{N}_{\alpha,\beta}(\overline{\mathbf{x}}_k)^\top \boldsymbol{\sigma}_{\mathbf{x}_d}(t) + \Gamma_{\alpha,\beta}(\overline{\mathbf{x}}_k) \leq u_{\max}. \quad (35)$$

is a convex quadratic inequality constraint in the quantity $\boldsymbol{\sigma}_{\mathbf{x}_d}(t)$. As the function $\boldsymbol{\sigma}_{\mathbf{x}_d}$ is defined by Bézier control points, we seek to express this constraint with control points.

Remark 1. We note that the proof of Theorem 1 in [41] establishes the existence of values of $\underline{\alpha}$ and $\underline{\beta}$ through Lipschitz properties of the dynamics. In practice, it may be difficult to compute these values, and they may not necessarily be the minimum values for which this result holds. Moreover, choosing very large values of α and β may lead to conservative behavior, as the constraint in (35) will constrain the dynamically admissible trajectory $\mathbf{x}_d$ to a small neighborhood of $\overline{\mathbf{x}}_k$. With this result we seek to highlight an important monotonic structural property of the system that permits a well-posed and practical approach for achieving input constraint satisfaction. In particular, one may begin with small values of α and β and increase them until the closed-loop nonlinear system meets input constraints. We will demonstrate this type of procedure in Section VI.

Before relating Theorem 1 to the Bézier control points defining $\mathbf{x}_d$, we state the following lemma (proven in [41]):

Lemma 5. *For any $\mathbf{x} \in \mathbb{R}^n$, we have that:*

$$\|\mathbf{r}_k(\tau) - \mathbf{x}\|_2 \leq \sup_i \|(\boldsymbol{\zeta}_k)_i - \mathbf{x}\|_2, \quad (36)$$

$$\|r_k^{(n)}(\tau) - f(\mathbf{x})\|_2 \leq \sup_i \|(\xi_k)_{n,i} - f(\mathbf{x})\|_2, \quad (37)$$

for all $\tau \in [0, T]$.

With this result, we now state one of our main results (proven in [41]) for tractably enforcing input bounds:

Lemma 6. *If given a collection of points $\{\overline{\mathbf{x}}_k\}$ with $\overline{\mathbf{x}}_k \in \mathcal{X}$ for $k = 0, \ldots, N-1$, there exists $\mathbf{s}_k \in \mathbb{R}^2_{\geq 0}$ such that:*

$$\begin{bmatrix} \|(\boldsymbol{\zeta}_k)_i - \overline{\mathbf{x}}_k\|_2 \\ \|(\xi_k)_{n,i} - f(\overline{\mathbf{x}}_k)\|_2 \end{bmatrix} \leq \mathbf{s}_k, \quad (38)$$

$$\frac{1}{2}\mathbf{s}_k^\top \mathbf{M}_{\alpha,\beta} \mathbf{s}_k + \mathbf{N}_{\alpha,\beta}(\overline{\mathbf{x}}_k)^\top \mathbf{s}_k + \Gamma_{\alpha,\beta}(\overline{\mathbf{x}}_k) \leq u_{\max}, \quad (39)$$

for $i = 0, \ldots, 2n-1$ and $k = 0, \ldots, N-1$, then we have that the inequality (35) is satisfied with $\boldsymbol{\sigma}_{\mathbf{x}_d}(t)$ defined as in (34) for all $t \in [\underline{t}, \overline{t}]$.

A consequence of this result is that for sufficiently high values of α and β, meeting the conditions of Lemma 6 implies $\|k_{\mathbf{x}_d}^{\mathrm{fbl}}(\mathbf{x}, t)\|_2 \leq u_{\max}$ for all $t \in [\underline{t}, \overline{t}]$ and $\mathbf{x} \in \Omega_{\mathbf{x}_d}(t, \overline{w})$. Moreover, the constraint (38) is a second-order cone constraint and the constraint (39) is a convex quadratic constraint (which may be written as a second-order cone constraint [41]), and thus they may be used in a convex program for determining Bézier control points. Lastly, the following corollary relates bounds on $k_{\mathbf{x}_d}^{\mathrm{fbl}}$ and $k_{\mathbf{x}_d}^{\mathrm{clf}}$:

Corollary 1. *If the function $k_{\mathbf{x}_d}^{\mathrm{fbl}}$ is bounded as in (33) for all $t \in [t_k, t_{k+1})$ and $\mathbf{x} \in \Omega_{\mathbf{x}_d}(t, \overline{w})$, then we have that:*

$$\begin{aligned} \|k_{\mathbf{x}_d}^{\mathrm{clf}}(\mathbf{x}, t)\|_2 \leq \frac{1}{2} &\boldsymbol{\sigma}_{\mathbf{x}_d}(t)^\top \mathbf{M}_{\alpha,\beta} \boldsymbol{\sigma}_{\mathbf{x}_d}(t) \quad (40) \\ &+ \mathbf{N}_{\alpha,\beta}(\overline{\mathbf{x}}_k)^\top \boldsymbol{\sigma}_{\mathbf{x}_d}(t) + \Gamma_{\alpha,\beta}(\overline{\mathbf{x}}_k), \end{aligned}$$

for all $t \in [t_k, t_{k+1})$ and all $\mathbf{x} \in \Omega_{\mathbf{x}_d}(t, \overline{w})$.

V. Multi-Rate Control Architecture

Utilizing the developments presented in the previous sections, we now construct a multi-rate control architecture which iteratively produces dynamically admissible trajectories for the system (1) and tracks them with the low-level controller designed in Section II. By achieving robustness to disturbances with the low-level controller, the trajectory planning can be done with a disturbance-free system.

A. Model Predictive Control

In this section we establish how to compute the collection of points $\{\mathbf{x}_k\}$ used to define $\mathbf{x}_d$ in Lemma 2 while meeting the desired constraints on the Bézier control points. Consider a collection of points $\{\overline{\mathbf{x}}_k\}$ with $\overline{\mathbf{x}}_k \in \mathcal{X}$ and $\{\overline{u}_k\}$ with $\overline{u}_k \in \mathbb{R}$ for $k = 0, \dots N-1$. To incorporate information about the system dynamics when synthesizing $\mathbf{x}_d$ as in Lemma 2, we will use linearizations of the system dynamics (1) around these collections of points. This approximation of the dynamics will provide constraints on sequential state points (and the corresponding Bézier control points as defined by (25)) defining $\mathbf{x}_d$. We neglect the disturbances $\mathbf{w}$ in this approximation as the low-level controller rejects these disturbances and provides a robust invariant set around $\mathbf{x}_d$. Consider a linear, temporal discretization of (1):

$$\mathbf{x}_{k+1} = \mathbf{A}(\overline{\mathbf{x}}_k, \overline{u}_k)\mathbf{x}_k + \mathbf{B}(\overline{\mathbf{x}}_k)u_k + \mathbf{C}(\overline{\mathbf{x}}_k, \overline{u}_k), \tag{41}$$

where $\mathbf{A} : \mathcal{X} \times \mathbb{R} \to \mathbb{R}^{n \times n}, \mathbf{B} : \mathcal{X} \to \mathbb{R}^n$, and $\mathbf{C} : \mathcal{X} \times \mathbb{R} \to \mathbb{R}^n$ come from linearizing and taking the exact temporal discretization[5] (with sample period T) of the dynamics in (1). For notational simplicity let us define:

$$\mathbf{A}_k \triangleq \mathbf{A}(\overline{\mathbf{x}}_k, \overline{u}_k), \quad \mathbf{B}_k \triangleq \mathbf{B}(\overline{\mathbf{x}}_k), \quad \mathbf{C}_k \triangleq \mathbf{C}(\overline{\mathbf{x}}_k, \overline{u}_k). \tag{42}$$

Given these, let us denote the state at a time $t \in \mathbb{R}_{\geq 0}$ by $\mathbf{x}(t)$. Building upon the previous two sections, we propose a Finite Time Optimal Control Problem (FTOCP):

$$\begin{aligned}
\min_{\substack{u_k, \mathbf{x}_k \\ \mathbf{s}_k, \boldsymbol{\xi}_k}} \quad & \sum_{k=0}^{N-1} h(\mathbf{x}_k, u_k) + J(\mathbf{x}_N) && \text{(FTOCP)} \\
\text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k u_k + \mathbf{C}_k, && \text{(43a)} \\
& \mathbf{x}_0 \in \mathbf{x}(t) \oplus \mathcal{E}, && \text{(43b)} \\
& \mathbf{x}_N = \mathbf{0}, && \text{(43c)} \\
& (\boldsymbol{\xi}_k) = \begin{bmatrix} \mathbf{x}_k^\top & \mathbf{x}_{k+1}^\top \end{bmatrix} \mathbf{D}^{-1}, && \text{(43d)} \\
& (\boldsymbol{\zeta}_k)_i \in \mathcal{X} \ominus \mathcal{E}, \qquad \forall i \in \mathcal{I} && \text{(43e)} \\
& \begin{bmatrix} \|(\boldsymbol{\zeta}_k)_i - \overline{\mathbf{x}}_k\|_2 \\ \|(\xi_k)_{n,i} - f(\overline{\mathbf{x}}_k)\|_2 \end{bmatrix} \leq \mathbf{s}_k, \qquad \forall i \in \mathcal{I} && \text{(43f)} \\
& \frac{1}{2}\mathbf{s}_k^\top \mathbf{M}_{\alpha,\beta}\mathbf{s}_k + \mathbf{N}_{\alpha,\beta}(\overline{\mathbf{x}}_k)^\top \mathbf{s}_k \\
& \qquad + \Gamma_{\alpha,\beta}(\overline{\mathbf{x}}_k) \leq u_{\max}, && \text{(43g)}
\end{aligned}$$

where $h : \mathcal{X} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ is a convex stage cost, $J : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is a convex terminal cost, and $\mathcal{I} = \{0, \dots, 2n-1\}$. The constraint in (43a) requires that the sequence of discrete points defining $\mathbf{x}_d$ satisfy a linear, discrete time approximation of the system dynamics. The constraint in (43b) requires that the beginning of $\mathbf{x}_d$ is close to the current state $\mathbf{x}(t)$, such that $\mathbf{x}(t) \in \Omega(t, \overline{w})$ as required by Lemma 1. The constraint in (43c) requires the end of $\mathbf{x}_d$ to be placed at origin. The constraints in (43d)-(43g) relate the discrete points $\mathbf{x}_k$ to Bézier control points, and consequently the continuous trajectory $\mathbf{x}_d$ tracked by the low-level controller. Note that as in Fact 1, the coefficients $(\boldsymbol{\xi})_k$ and $(\boldsymbol{\zeta}_k)_i$ are linearly related for $i = 0, \dots, 2n-1$, a constraint implicitly assumed in (FTOCP). If h and J are positive definite quadratic functions, (FTOCP) is a second-order cone program (SOCP), which can be efficiently solved [42].

Remark 2. Note that we do not explicitly enforce input constraints on the decision variables u_k. Instead, constraints are induced on these decision variables through the linear dynamics constraint (43a) and the constraints on the Bézier coefficients in (43d) and (43f)-(43g). Moreover, these constraints ensure that the low-level controller will satisfy input constraints as desired.

B. The Multi-Rate Architecture

We now present the multi-rate architecture that integrates the low-level controller design posed in Section II with the preceding trajectory planner encoded in (FTOCP).

We first recall the role T plays in dynamically admissible trajectories synthesized through Bézier curves as in Lemma 2, as well as its role as a sampling period for the temporal discretization established in (41). Let us denote $\mathcal{T} = \cup_{i=0}^{\infty}\{iT\}$. This set serves to index the discrete points in time (separated by T) at which a dynamically admissible trajectory for the system will be replanned by solving the (FTOCP). The multi-rate architecture is initialized at time $t = 0$ with collections of points $\{\overline{\mathbf{x}}_{k|0}\}$ and $\{\overline{u}_{k|0}\}$ with $\overline{\mathbf{x}}_{k|0} \in \mathcal{X}$ and $\overline{u}_{k|0} \in \mathbb{R}$ for $k = 0, \dots, N-1$. Let us denote the linearized and discretized dynamics computed around these collections by $\{\mathbf{Lin}_{k|0}\} = \{(\mathbf{A}_{k|0}, \mathbf{B}_{k|0}, \mathbf{C}_{k|0})\}$.

Assumption 4. Given an initial condition $\mathbf{x}(0) \in \mathcal{X}$, (FTOCP) is feasible using $\{\mathbf{Lin}_{k|0}\}$.

Algorithm 1 u = C-MPC($\mathbf{x}, t$)

```
if t ∈ 𝒯 = ∪_{i=0}^∞ {iT} then
    Compute {Lin_{k|i}} in (41) about {x̄_{k|i}} and {ū_{k|i}};
    Solve (FTOCP) with {Lin_{k|i}};
    if (FTOCP) is infeasible then
        {Lin_{k|i}} ← {Lin_{1|i−1}, ..., Lin_{N−1|i−1}, Lin_O};
        Solve (FTOCP) with {Lin_{k|i}};
    end if
    {x̄_{k|i+1}} ← {x*_{1|i}, ..., x*_{N−1|i}, x*_{N|i}};
    {ū_{k|i+1}} ← {u*_{1|i}, ..., u*_{N−1|i}, 0};
end if
Calculate x_d|i from {x*_{k|i}}, as in (25)–(26);
return u = k^clf_{x_d|i}(x, t);
```

[5]See [41] for a formula for these linearizations and discretizations.

We now describe our multi-rate framework as summarized in Algorithm 1. As in Line 1, let $t \in \mathcal{T}$ such that $t = iT$ for some $i \in \mathbb{Z}$. In Line 2, the linearized and discretized dynamics are computed around the collections $\{\overline{\mathbf{x}}_{k|i}\}$ and $\{\overline{u}_{k|i}\}$, and are denoted by $\{\mathbf{Lin}_{k|i}\} = \{(\mathbf{A}_{k|i}, \mathbf{B}_{k|i}, \mathbf{C}_{k|i})\}$. In Line 3 these dynamics are used to solve the (FTOCP) using the state at the current time, $\mathbf{x}(t)$, in (43b). If the (FTOCP) is feasible, it returns collections of points $\{\mathbf{x}^*_{k|i}\}$ with $\mathbf{x}^*_{k|i} \in \mathcal{X}$ for $k = 0, \ldots, N$ and $\{u^*_{k|i}\}$ with $u^*_{k|i} \in \mathbb{R}$ for $k = 0, \ldots, N-1$. If (FTOCP) is infeasible, in Line 5 we set the linearized and discretized dynamics $\{\mathbf{Lin}_{k|i}\}$ to the previous linearization shifted by one and appending the linearization and discretization around the origin, denoted $\mathbf{Lin}_O = (\mathbf{A}(\mathbf{0},0), \mathbf{B}(\mathbf{0}), \mathbf{C}(\mathbf{0},0))$. In Line 6 we solve (FTOCP) and similarly return collections of points $\{\mathbf{x}^*_{k|i}\}$ and $\{u^*_{k|i}\}$. As we will show in Theorem 2, our assumption about feasibility at time $t = 0$ will ensure that switching to this set of linearizations will always ensure (FTOCP) is feasible. In Line 8–9 the collection $\{\mathbf{x}^*_{k|i}\}$ is shifted and the collection $\{u^*_{k|i}\}$ is shifted and appended with 0 to define collections $\{\overline{\mathbf{x}}_{k|i+1}\}$ and $\{\overline{u}_{k|i+1}\}$ used for linearization and discretization in the next iteration. In Line 11 the collection $\{\mathbf{x}^*_{k|i}\}$ is then used to define a dynamically admissible trajectory $\mathbf{x}_d|i$ as in Lemma 2, which yields a corresponding low-level controller $k^{\mathrm{clf}}_{\mathbf{x}_d|i}$ that defines the output of our algorithm. We may view our algorithm as a time-varying controller that yields a closed-loop system (2). Importantly, our algorithm ensures state and input constraints are satisfied as the continuous time system evolves under this controller, as stated in the following theorem (proven in [41]):

Theorem 2. *Suppose that $\alpha \geq \underline{\alpha}$ and $\beta \geq \underline{\beta}$ are such that $\Gamma_{\alpha,\beta}(\mathbf{0}) \leq u_{\max}$. Let* (FTOCP) *be defined with α and β, and consider the closed-loop system* (2) *with a feedback controller given by C-MPC in Algorithm 1 and a disturbance signal satisfying $\|\mathbf{w}\|_\infty \leq \overline{w}$. If $\mathbf{0} \in \mathcal{X} \ominus \mathcal{E}$ and* (FTOCP) *is feasible at $t_0 = 0$ with initial condition $\mathbf{x}(0) \in \mathcal{X}$, then C-MPC is well-defined for all time, and the closed-loop system* (2) *satisfies state and input constraints.*

VI. Simulation

We consider the following nonlinear system in simulation:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \sin(x_1) + x_2^3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}.$$

The goal is to drive the system to the origin while satisfying state and input constraints for all time. Fig. 3 demonstrates that at different time scales, both with and without added disturbances, using only either a low-level or mid-level controller results in state and/or input violation, whereas the proposed combined approach is able to satisfy both for all time. Fig. 4 shows the behavior of the system for increasing values of α and β. As the parameter values increase, the planned MPC points become closer to reduce deviation from the linearization points, and the deviation of the low-level controller from the planned input u_k decreases as the system evolves from $\mathbf{x}^*_k$ to $\mathbf{x}^*_{k+1}$. Simulation code is given at [43].

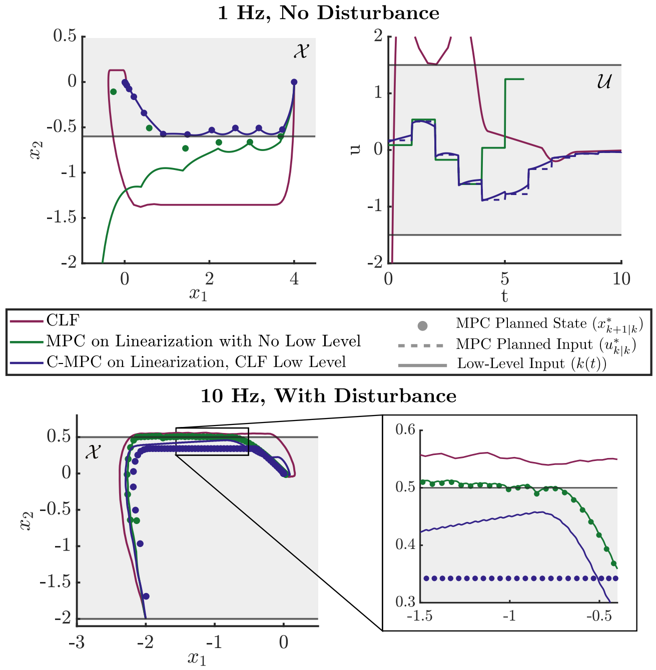

Fig. 3. Comparison of three control methods: only using a low level controller (CLF), applying MPC with no low-level controller, and applying the proposed C-MPC with a CLF at the low-level. In both scenarios, just using the low-level or mid-level controller separately yields both state and input violation.

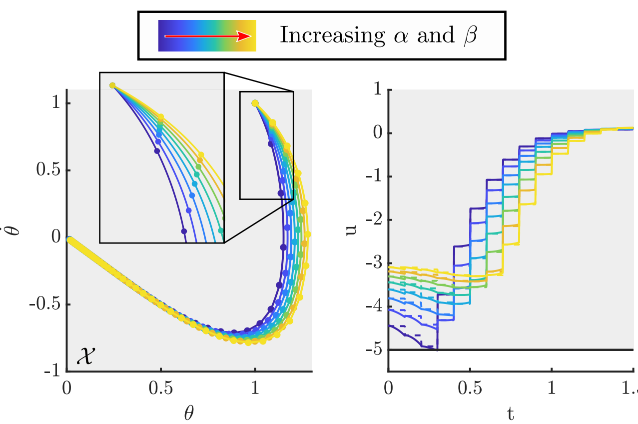

Fig. 4. The proposed C-MPC for increasing user parameter values α and β. Notice that as the parameters increase, the planned MPC points become spatially closer to reduce the linearization error, resulting in the deviation of the low-level controller from the planned control input decreasing.

VII. Conclusion and Future Work

In conclusion, we have presented a multi-rate control architecture for nonlinear systems that utilizes MPC in conjunction with Bézier curves to iteratively plan continuous time trajectories that are tracked using Control Lyapunov Function based controllers. Our approach allows us to ensure that the low-level controller satisfies state and input constraints as it tracks the desired trajectory. We believe there are a number of meaningful directions for future work. First, in the pursuit of a truly multi-*rate* scheme, the low-level CLF control design could be adapted to the sampled-data setting [44]. Next, our work uses the origin as the terminal set, but constructive approaches to synthesize terminal sets using the ideas in [45] could greatly improve the feasible domain of our method. Lastly, we believe that underactuation and unstable *zero-dynamics* may be best approached through joint planning and low-level control, and that our work is a first step in this direction [46].

REFERENCES

[1] A. Isidori, *Nonlinear control systems*. Springer-Verlag London, 1995, vol. 3.

[2] P. Kokotović and M. Arcak, "Constructive nonlinear control: a historical perspective," *Automatica*, vol. 37, no. 5, pp. 637–662, 2001.

[3] H. K. Khalil and J. W. Grizzle, *Nonlinear Systems*. Upper Saddle River, NJ: Prentice Hall, 2002, vol. 3.

[4] R. Sepulchre, M. Jankovic, and P. V. Kokotovic, *Constructive nonlinear control*. Springer Science & Business Media, 2012.

[5] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Analysis: Theory, Methods & App.*, vol. 7, no. 11, pp. 1163–1173, 1983.

[6] E. D. Sontag, "A 'universal' construction of artstein's theorem on nonlinear stabilization," *Sys. & Control Let.*, vol. 13, no. 2, pp. 117–123, 1989.

[7] ——, "Smooth stabilization implies coprime factorization," *Trans. on Automatic Control*, vol. 34, no. 4, pp. 435–443, 1989.

[8] R. Freeman and P. V. Kokotovic, *Robust nonlinear control design: state-space and Lyapunov techniques*. Birkhauser Basel, 1996.

[9] A. D. Ames and M. Powell, "Towards the unification of locomotion and manipulation through control lyapunov functions and quadratic programs," in *Control of Cyber-Physical Sys.* Springer, 2013, pp. 219–240.

[10] S. Kolathaya, J. Reher, A. Hereid, and A. D. Ames, "Input to state stabilizing control lyapunov functions for robust bipedal robotic locomotion," in *American Control Conf. (ACC)*. IEEE, 2018, pp. 2224–2230.

[11] F. Allgower, R. Findeisen, Z. K. Nagy, *et al.*, "Nonlinear model predictive control: From theory to application," *J.-Chinese Institute Of Chemical Engineers*, vol. 35, no. 3, pp. 299–316, 2004.

[12] F. Allgöwer and A. Zheng, *Nonlinear model predictive control*. Birkhäuser, 2012, vol. 26.

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

[14] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *Int. Conf. on Intelligent Robots and Sys. (IROS)*. IEEE/RSJ, 2018, pp. 1–9.

[15] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *Robotics and Automation Let.*, vol. 6, no. 3, pp. 4688–4695, 2021.

[16] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *Trans. on Control Sys. Tech.*, vol. 15, no. 3, pp. 566–580, 2007.

[17] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *Int. Conf. on Control App.* IEEE, 2012, pp. 295–302.

[18] P. F. Lima, G. C. Pereira, J. Mårtensson, and B. Wahlberg, "Experimental validation of model predictive control stability for autonomous driving," *Control Eng. Practice*, vol. 81, pp. 244–255, 2018.

[19] S. Bengea, A. Kelman, F. Borrelli, R. Taylor, and S. Narayanan, "Model predictive control for mid-size commercial building hvac: Implementation, results and energy savings," in *Int. Conf. on Building Energy and Environment*, 2012, pp. 979–986.

[20] G. Serale, M. Fiorentini, A. Capozzoli, D. Bernardini, and A. Bemporad, "Model predictive control (mpc) for enhancing building and hvac system energy efficiency: Problem formulation, applications and opportunities," *Energies*, vol. 11, no. 3, p. 631, 2018.

[21] E. T. Maddalena, Y. Lian, and C. N. Jones, "Data-driven methods for building control—a review and promising future directions," *Control Eng. Practice*, vol. 95, p. 104211, 2020.

[22] U. Rosolia and F. Borrelli, "Learning how to autonomously race a car: a predictive control approach," *Trans. on Control Sys. Tech.*, vol. 28, no. 6, pp. 2713–2719, 2019.

[23] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1: 43 scale rc cars," *Opt. Control App. and Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[24] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *Trans. on Control Sys. Tech.*, vol. 28, no. 6, pp. 2736–2743, 2019.

[25] D. Nešić, A. R. Teel, and P. V. Kokotović, "Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations," *Sys. & Control Let.*, vol. 38, no. 4-5, pp. 259–270, 1999.

[26] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *Conf. on Decision and Control (CDC)*. IEEE, 2017, pp. 2242–2253.

[27] A. Carvalho, Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "Predictive control of an autonomous ground vehicle using an iterative linearization approach," in *Int. Conf. on Intelligent Transportation Sys. (ITSC)*. IEEE, 2013, pp. 2335–2340.

[28] Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles," *Vehicle Sys. Dynamics*, vol. 52, no. 6, pp. 802–823, 2014.

[29] M. Kögel and R. Findeisen, "Discrete-time robust model predictive control for continuous-time nonlinear systems," in *American Control Conf. (ACC)*. IEEE, 2015, pp. 924–930.

[30] S. Yu, C. Maier, H. Chen, and F. Allgöwer, "Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems," *Sys. & Control Let.*, vol. 62, no. 2, pp. 194–200, 2013.

[31] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust online motion planning via contraction theory and convex optimization," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5883–5890.

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

[33] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Fastrack: A modular framework for fast and guaranteed safe motion planning," in *Conf. on Decision and Control (CDC)*. IEEE, 2017, pp. 1517–1522.

[34] S. Singh, M. Chen, S. L. Herbert, C. J. Tomlin, and M. Pavone, "Robust tracking with model mismatch for fast and safe planning: an sos optimization approach," in *Int. Work. on the Algorithmic Foundations of Robotics (WAFR)*. Springer, 2018, pp. 545–564.

[35] H. Yin, M. Bujarbaruah, M. Arcak, and A. Packard, "Optimization based planner–tracker design for safety guarantees," in *American Control Conf. (ACC)*. IEEE, 2020, pp. 5194–5200.

[36] U. Rosolia, A. Singletary, and A. D. Ames, "Unified multi-rate control: From low-level actuation to high-level planning," *Trans. on Automatic Control*, 2022.

[37] U. Rosolia and A. D. Ames, "Multi-rate control design leveraging control barrier functions and model predictive control policies," *Control Sys. Let.*, vol. 5, no. 3, pp. 1007–1012, 2021.

[38] M. Kamermans, "A primer on bézier curves," *(online book)*, 2020.

[39] G. E. Farin, *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002.

[40] M. Muehlebach, C. Sferrazza, and R. D'Andrea, "Implementation of a parametrized infinite-horizon model predictive control scheme with stability guarantees," in *Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2723–2730.

[41] N. Csomay-Shanklin, A. J. Taylor, U. Rosolia, and A. D. Ames, "Multi-rate planning and control of uncertain nonlinear systems: Model predictive control and control lyapunov functions," *arXiv:2204.00152*, 2022.

[42] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 9.3.*, 2022.

[43] "C-mpc matlab code," *https://github.com/noelc-s/C-MPC*, 2022.

[44] A. J. Taylor, V. D. Dorobantu, Y. Yue, P. Tabuada, and A. D. Ames, "Sampled-data stabilization with control lyapunov functions via quadratically constrained quadratic programs," *Control Sys. Let.*, vol. 6, pp. 680–685, 2022.

[45] T. Marcucci, J. Umenberger, P. A. Parrilo, and R. Tedrake, "Shortest paths in graphs of convex sets," *arXiv:2101.11565*, 2021.

[46] J. Koehler, M. A. Muller, and F. Allgower, "Constrained nonlinear output regulation using model predictive control," *Trans. on Automatic Control*, 2021.