Multi-Rate Control Design Leveraging Control Barrier Functions and Model Predictive Control Policies

Ugo Rosolia and Aaron D. Ames

Abstract— In this paper we present a multi-rate control architecture for safety critical systems. We consider a high level planner and a low level controller which operate at different frequencies. This multi-rate behavior is described by a piecewise nonlinear model which evolves on a continuous and a discrete level. First, we present sufficient conditions which guarantee recursive constraint satisfaction for the closed-loop system. Afterwards, we propose a control design methodology which leverages Control Barrier Functions (CBFs) for low level control and Model Predictive Control (MPC) policies for high level planning. The control barrier function is designed using the full nonlinear dynamical model and the MPC is based on a simplified planning model. When the nonlinear system is control affine and the high level planning model is linear, the control actions are computed by solving convex optimization problems at each level of the hierarchy. Finally, we show the effectiveness of the proposed strategy on a simulation example, where the low level control action is updated at a higher frequency than the high level command.

I. INTRODUCTION

Autonomous systems are designed to take control actions upon sensing the environment around them. The decision making process is usually divided into different layers. For instance, in autonomous driving the top layer determines a goal or intention, such as lane keeping, merging or overtaking. Then, a high level planner computes a desired collision-free trajectory, which is then fed to a low level controller that computes the control action. Each layer operates at different frequency and it is designed using model of increasing accuracy and complexity.

Combining high level planners with low level controllers has been extensively studied in literature [1]–[11]. Safety can be guaranteed using low level filters which, given a desired high level command, compute the closest safe control action using control barrier functions [1]–[3] or feasibility of an MPC problem [4]. The high level planner may be designed using a simplified model and the planned trajectory can be tracked using low level controllers. The tracking error and the associated tracking policy can be computed using Hamilton-Jacobi (HJ) reachability analysis [5] or sum-of-squares programming [6], [7]. Finally, high level planning and low level control can be implemented using nonlinear tube MPC strategies [8]–[12], where the difference between the planned trajectory and the actual one is over approximated using Lyapunov based analysis or Lipschitz properties of the nonlinear dynamics.

In the aforementioned papers, the low level and high level control actions are updated at the same frequency. In this

Ugo Rosolia and Aaron D. Ames are with the AMBER lab at Caltech, Pasadena, USA. E-mails: {urosolia, ames}@caltech.edu.
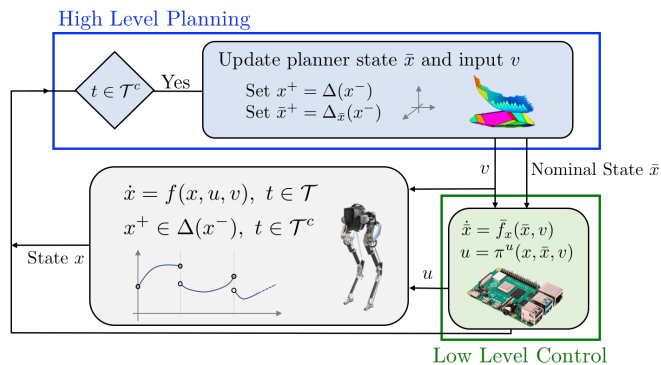

Fig. 1. Representation of the multi-rate control architecture. The high level planner computes the desired state $\bar{x}$ and the high level command v. The low level controller computes at higher frequency the action u.

paper, we consider a high level planner which operates at a lower frequency than the low level controller. Multi-rate strategies are used in several applications, for instance in bipedal locomotion [13], [14], autonomous driving [15], [16] and power grids [17], [18]. In this work, we introduce sufficient conditions to analyze the closed-loop safety properties of such control architectures. Our contribution is threefold. First, we introduce sufficient conditions which guarantee recursive constraint satisfaction for a multi-frequency high level planning and low level control architecture, where the high level planner can reset its internal state as a function of the current state of the system. Second, we present a control design which leverages CBFs for low level control and MPC for high level planning. We show that when the true system is nonlinear control affine and the planning model is linear, then the proposed strategy is implemented solving convex optimization problems. Third, we benchmark the proposed strategy against linear and nonlinear MPC policies. Simulation results demonstrate the benefit of the proposed multi-rate architecture, where the low level control action is updated at a higher frequency than the high level command.

This paper is organized as follows. In Section II we introduce the problem formulation. Section III describes the control architecture and the sufficient conditions which guarantee safety. The synthesis process is described in Section IV and it is demonstrated on a numerical example in Section V. *Notation:* The Minkowski sum of two sets $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^n$ is denoted as $\mathcal{X} \oplus \mathcal{Y}$, and the Pontryagin difference as $\mathcal{X} \ominus \mathcal{Y}$. The set $\mathcal{K}^e$ is the set of extended class-$\mathcal{K}^e$ functions β which are strictly increasing and $\beta(0) = 0$. Finally, given a function $f : \mathbb{R}^n \to \mathbb{R}^m$ and a set $\mathcal{X} \subset \mathbb{R}^n$ we denote the set $f(\mathcal{X}) = \{y \in \mathbb{R}^m : \exists\, x \in \mathcal{X} \text{ such that } y = f(x)\}$.

II. Problem Formulation

This section introduces the system model and the synthesis objectives. Consider a *piecewise nonlinear model*:

$$\Sigma : \begin{cases} \dot{x}(t) = f(x(t), u(t), v(t)), & t \in \mathcal{T} = \cup_{k=0}^{\infty}(t_k, t_{k+1}) \\ x^+(t) = \Delta(x^-(t)), & t \in \mathcal{T}^c = \cup_{k=0}^{\infty}\{t_k\} \end{cases},$$
(1)

where the state $x \in \mathbb{R}^n$, the set $\mathcal{T}$ collects the open intervals from time t_k to time t_{k+1} and its complement $\mathcal{T}^c$ collects the time instances t_k. As a result, the above system (1) evolves accordingly to the differential equation $\dot{x}(t) = f(x(t), u(t), v(t))$ between time t_k and time t_{k+1}. On the other hand, at time t_k the system evolution is defined by the *reset map* $\Delta(\cdot)$, where $x^-(t) = \lim_{\tau \nearrow t} x(\tau)$ and $x^+(t) = \lim_{\tau \searrow t} x(\tau)$ are the right and left limits of a trajectory $x(t)$ which is assumed right continuous. Furthermore, we assume that the input $u \in \mathbb{R}^d$ is a continuous function of the state x and the input $v \in \mathbb{R}^d$ is a piecewise-constant function which is updated when $t \in \mathcal{T}^c$, i.e.,

$$\Pi : \begin{cases} u(t) = \pi^u(x(t), v(t)), \ \dot{v}(t) = 0, & t \in \mathcal{T} \\ u^+(t) = u^-(t), \ v^+(t) = \pi^v(x^+(t)), & t \in \mathcal{T}^c \end{cases}.$$
(2)

The above control policies (2) in closed-loop with system (1) results in a piecewise nonlinear autonomous system, which evolves on a discrete and a continuous level.

Objective: Our goal is to steer the system from a starting state x_s to a goal state x_g while satisfying the following state and input constraints:

$$\begin{aligned} & x(t) \in \mathcal{X}_c \subset \mathbb{R}^n, \ \forall t \in \mathcal{T}, \\ & x^+(t) \in \mathcal{X}_d \subseteq \mathcal{X}_c \subset \mathbb{R}^n, \ \forall t \in \mathcal{T}^c, \\ & u(t) \in \mathcal{U} \subset \mathbb{R}, v(t) \in \mathcal{V} \subset \mathbb{R}^d, \forall t \geq 0. \end{aligned}$$
(3)

III. Framework Architecture And Properties

In this section we present the multi-rate control architecture. First, we introduce an augmented model, which is composed by the piecewise nonlinear system (1) and a high level planning model. The latter is affected by the piecewise constant input v and it is used to compute the planner state $\bar{x} \in \mathbb{R}^n$. Afterwards, the planned trajectory together with the input v are fed to the low level controller which computes the control action u, as shown in Figure 1.

A. Augmented System

The augmented system is defined as

$$\Sigma_{\bar{x}} : \begin{cases} \dot{x}(t) = f(x(t), u(t), v(t)) \\ \dot{\bar{x}}(t) = f_{\bar{x}}(\bar{x}(t), v(t)) \end{cases}, \quad t \in \mathcal{T} = \cup_{k=0}^{\infty}(t_k, t_{k+1}) \\ \begin{cases} x^+(t) = \Delta(x^-(t)) \\ \bar{x}^+(t) = \Delta_{\bar{x}}(x^-(t)) \end{cases}, \quad t \in \mathcal{T}^c = \cup_{k=0}^{\infty}\{t_k\}$$
(4)

and the control actions are given by the policies

$$\Pi_{\bar{x}} : \begin{cases} u(t) = \pi^u(x(t), \bar{x}(t), v(t)), \ \dot{v}(t) = 0, & t \in \mathcal{T} \\ u^+(t) = u^-(t), \ v^+(t) = \pi^v(x^+(t)), & t \in \mathcal{T}^c \end{cases},$$
(5)

where $\mathcal{T}$ and $\mathcal{T}^c$ are defined as in (4) and $\bar{x} \in \mathbb{R}^n$ represents the planned state which is affected by the piecewise input v.

B. High Level and Low Level Properties

In this section, we define four properties associated with the high level planner and low level controller. As we will discuss later on, when these properties hold the closed-loop system is guaranteed to recursively satisfy state and input constraints (3).

Consider the closed-loop system (4)-(5) and let t_k be the time at which the kth discontinuous transition occurs, i.e., $x^+(t_k) = \Delta(x^-(t_k)) \ \forall k \in \{1, 2, \ldots\}$. We define the error $e = x - \bar{x}$ and we introduce the following error dynamics:

$$\Sigma_e : \begin{cases} \dot{e}(t) = f_e(x(t), u(t), v(t), \bar{x}(t)), & t \in \mathcal{T} \\ e^+(t) = \Delta_e(x^-(t)), & t \in \mathcal{T}^c \end{cases},$$
(6)

where $\mathcal{T}$ and $\mathcal{T}^c$ are defined as in (4), and the reset map $\Delta_e(\cdot)$ is designed so that the following properties hold.

Property 1 (low level safety). *The control policy $\pi^u(\cdot)$ from (5) guarantees low level safety for the closed-loop system (4)-(5) and the set $\mathcal{S}_x \subseteq \mathcal{X}_c \subset \mathbb{R}^n$, if $\forall x^+(t_k) \in \mathcal{S}_x \cap \mathcal{X}_d$ and $\forall v^+(t_k) \in \mathcal{V}$ we have that*

$$x(t) \in \mathcal{S}_x \subseteq \mathcal{X}_c \text{ and } u(t) \in \mathcal{U}, \forall t \in (t_k, t_{k+1}).$$
(7)

Basically, the above property guarantees that state and input constraints are satisfied when the system evolves smoothly between time t_k and time t_{k+1}. In particular, if at time t_k the state $x^+(t_k)$ belongs to the set $\mathcal{S}_x \cap \mathcal{X}_d$, then the low level controller $\pi^u(\cdot)$ guarantees state and input constraint satisfaction until the next discontinuous transition at time t_{k+1}.

Property 2 (low level tracking). *The control policy $\pi^u(\cdot)$ from (5) guarantees low level tracking for the closed-loop system (4)-(5), the set $\mathcal{S}_e \subset \mathbb{R}^n$ and the set $\mathcal{S}_x \subseteq \mathcal{X}_c \subset \mathbb{R}^n$, if $\forall e^+(t_k) = x^+(t_k) - \bar{x}^+(t_k) \in \mathcal{S}_e$, $\forall x^+(t_k) \in \mathcal{S}_x \cap \mathcal{X}_d$ and $\forall v^+(t_k) \in \mathcal{V}$ we have that*

$$e(t) = x(t) - \bar{x}(t) \in \mathcal{S}_e, \forall t \in (t_k, t_{k+1}).$$
(8)

The low level tracking property ensures that the difference between the planned trajectory and the true state is contained into the set $\mathcal{S}_e$ for all time $t \in (t_k, t_{k+1})$. The above Properties 1-2 guarantee that the planned trajectory can be safely executed by the true system.

Property 3 (high level safety). *The control policy $\pi^v(\cdot)$ from (5) guarantees high level safety for the closed-loop system (4)-(5), the set $\mathcal{S}_e \subset \mathbb{R}^n$ and the set $\mathcal{S}_x \subseteq \mathcal{X}_c \subset \mathbb{R}^n$, if for the initial conditions $x(0) = \bar{x}(0) + e(0) \in \mathcal{S}_x \cap \mathcal{X}_d$ and $e(0) \in \mathcal{S}_e$ we have that $\pi^v(x^+(0)) \in \mathcal{V}$ and*

$$\begin{aligned} & z \in \mathcal{S}_x \cap \mathcal{X}_d, \\ & \pi^v(z) \in \mathcal{V}, \ \forall z \in \Delta(\{\bar{x}^-(t_k)\} \oplus \mathcal{S}_e), \forall k \in \{1, 2, \ldots\}. \end{aligned}$$
(9)

Property 4 (high level tracking). *The reset map $\Delta_e(\cdot)$ from (6) guarantees high level tracking for the closed-loop system (4)-(5), the set $\mathcal{S}_e \subset \mathbb{R}^n$ and the set $\mathcal{S}_x \subseteq \mathcal{X}_c \subset \mathbb{R}^n$, if for the initial conditions $x(0) = \bar{x}(0) + e(0) \in \mathcal{S}_x \cap \mathcal{X}_d$ and $e(0) \in \mathcal{S}_e$ we have that*

$$\begin{aligned} & \Delta(z) = \Delta_{\bar{x}}(z) + \Delta_e(z), \\ & \Delta_e(z) \in \mathcal{S}_e, \forall z \in \ \bar{x}^-(t_k)\} \oplus \mathcal{S}_e, \forall k \in \{0, 1, \ldots\}. \end{aligned}$$
(10)

It is important to underline that the above Properties 3-4 are defined for the planning model at the discontinuous transitions, i.e., at time t_k for $k \in \{0, 1, \ldots\}$. This fact allows us to design a high level planner based on a discrete time model, which describes the evolution of the planned trajectory from $\bar{x}^+(t_k)$ to $\bar{x}^+(t_{k+1})$.

C. Safety Guarantees

In this section, we show that when the control policies from (5) satisfy Properties 1-4, the closed-loop system (4)-(5) does not violate state and input constraints (3).

Theorem 1. *Assume that Properties 1-4 are satisfied for the closed-loop system* (4)-(5)*, the set* $\mathcal{S}_e \subset \mathbb{R}^n$ *and the set* $\mathcal{S}_x \subseteq \mathcal{X}_c \subset \mathbb{R}^n$*. Let* $x(0) = \bar{x}(0) + e(0) \in \mathcal{S}_x \cap \mathcal{X}_d$ *and* $e(0) \in \mathcal{S}_e$*. Then, the closed-loop system* (4)-(5) *satisfies state and input constraints* (3) *for all time* $t \geq 0$*.*

Proof: The proof proceeds by induction. Assume that after the kth discontinuous transition $x^+(t_k) \in \mathcal{S}_x \cap \mathcal{X}_d$, $e^+(t_k) = x^+(t_k) - \bar{x}^+(t_k) \in \mathcal{S}_e$ and $v^+(t_k) \in \mathcal{V}$, then by Property 1

$$x(t) \in \mathcal{S}_x \subseteq \mathcal{X}_c, u(t) \in \mathcal{U}, \ \forall t \in (t_k, t_{k+1}). \quad (11)$$

Furthermore, by Property 2 we have that at time t_{k+1}

$$x^-(t_{k+1}) \in \{\bar{x}^-(t_{k+1})\} \oplus \mathcal{S}_e.$$

The above equation together with Property 3 implies that

$$x^+(t_{k+1}) = \Delta(x^-(t_{k+1})) \in \mathcal{S}_x \cap \mathcal{X}_d, v(t_{k+1}) \in \mathcal{V}. \quad (12)$$

Finally, from Property 4 we have

$$e^+(t_{k+1}) = x^+(t_{k+1}) - \bar{x}^+(t_{k+1}) \in \mathcal{S}_e. \quad (13)$$

The above equations (11)-(13) imply that, if $x^+(t_k) \in \mathcal{S}_x \cap \mathcal{X}_d$, $e^+(t_k) = x^+(t_k) - \bar{x}^+(t_k) \in \mathcal{S}_e$ and $v^+(t_k) \in \mathcal{V}$, then state and input constraints (3) are satisfied for all $t \in (t_k, t_{k+1})$. Furthermore, we have that the state $x^+(t_{k+1}) \in \mathcal{S}_x \cap \mathcal{X}_d$, the error $e^+(t_{k+1}) = x^+(t_{k+1}) - \bar{x}^+(t_{k+1}) \in \mathcal{S}_e$ and the input $v^+(t_{k+1}) \in \mathcal{V}$.
Finally, by assumption $x(0) = \bar{x}(0) + e(0) \in \mathcal{S}_x \cap \mathcal{X}_d$ and $e(0) \in \mathcal{S}_e$, which imply from Property 3 that $v(0) \in \mathcal{V}$. Therefore, from equations (11)-(13), we conclude by induction that the closed-loop system (4)-(5) recursively satisfies state and input constraints for all $t \geq 0$. ∎

Remark 1. We underline that guarantees from Theorem 1 hold when the control action $u(t)$ is updated continuously. However, in practice the control action is updated at a high frequency, for instance at $1kHz$ in our simulations.

IV. SYNTHESIS: LEVERAGING CBFs AND MPC

In this section, we discuss how the properties from Section III-B may be used to synthesize a safe controller. First, we show that Control Barrier Functions (CBFs) may be used to enforce low level safety and low level tracking. Afterwards, we design a Model Predictive Controller (MPC) to enforce high level safety and high level tracking.

We consider a control affine system where the input is given by the summation of the continuous control action u and the piecewise constant action v, i.e.,

$$\Sigma^a : \begin{cases} \dot{x}(t) = f(x(t)) + g(x(t))(u(t) + v(t)), & t \in \mathcal{T} \\ x^+(t) = \Delta(x^-(t)), & t \in \mathcal{T}^c \end{cases} \quad (14)$$

where $\mathcal{T} = \cup_{k=0}^{\infty}(kT, (k+1)T)$, $\mathcal{T}^c = \cup_{k=0}^{\infty}\{kT\}$ and $1/T$ is the frequency at which the high level command is updated. Furthermore, we assume the f and g are locally Lipschitz continuous with respect to their arguments and that the map $\Delta(\cdot)$ is affine, as stated in Assumption 1. Finally, the augmented model is given by

$$\Sigma_{\boldsymbol{z}}^a : \begin{cases} \dot{\boldsymbol{z}} = \begin{bmatrix} \dot{x} \\ \dot{\bar{x}} \end{bmatrix} = f^a(\boldsymbol{z}) + g^a(\boldsymbol{z})(u+v), & t \in \mathcal{T} \\ \boldsymbol{z}^+ = \begin{bmatrix} x^+ \\ \bar{x}^+ \end{bmatrix} = \begin{bmatrix} \Delta(x^-) \\ \Delta_{\bar{x}}(\bar{x}^-) \end{bmatrix}, & t \in \mathcal{T}^c \end{cases}. \quad (15)$$

where we dropped the dependence on time t to simplify the notation and the continuous evolution of the planning state $\bar{x}$ is described by linear dynamics, i.e., $\dot{\bar{x}} = A\bar{x} + Bv$.

Assumption 1. The functions f and g are locally Lipschitz continuous and the reset map $\Delta(\cdot)$ from (14) is affine. Consequently, the reset map Δ can be written as $\Delta(x) = Tx + p$, for some matrix $T \in \mathbb{R}^{n \times n}$ and some vector $p \in \mathbb{R}^n$.

A. Control Barrier Functions

In this section, we show that CBFs [2] can be used to enforce low level safety and low level tracking. Furthermore, we introduce a Control Lyapunov Function (CLF) which is used to reduce the tracking error. Finally, we combine CFBs and CLF into a QP, which defines the low level control policy from Figure 1.

First we define the following sets:

$$\begin{aligned} \mathcal{S}_x &= \{x \in \mathbb{R}^n : h_x(x) \geq 0\} \subseteq \mathcal{X}_c \subset \mathbb{R}^n, \\ \mathcal{S}_e &= \{e \in \mathbb{R}^n : h_e(e) \geq 0\} \subset \mathbb{R}^n, \end{aligned} \quad (16)$$

which will be used to check if Properties 1-4 hold. The above functions h_x and h_e are designed by the user based on the application, as shown in the result section. Furthermore, we define $||x||_Q = x^T Q x$ and we introduce the candidate Lyapunov function

$$V(\boldsymbol{z}) = ||x - \bar{x}||_{Q_v}, \quad (17)$$

which penalizes the deviation of the true system from the planned trajectory. Finally, the CBFs associated with the sets in (16), and the CLF (17) are used to define the following CLF-CBF Quadratic Program (QP):

$$\begin{aligned} \min_{u \in \mathcal{U}, \gamma} \quad & ||u||_2 + c_1 \gamma^2 \\ \text{s.t.} \quad & \frac{\partial V(\boldsymbol{z})}{\partial \boldsymbol{z}}(f^a(\boldsymbol{z}) + g^a(\boldsymbol{z})(v+u)) \leq -c_2 V + \gamma \\ & \frac{\partial h_x(x)}{\partial x}(f(x) + g(x)(v+u)) \geq -\alpha_1(h_x) \\ & \frac{\partial h_e(e)}{\partial \boldsymbol{z}}(f^a(\boldsymbol{z}) + g^a(\boldsymbol{z})(v+u)) \geq -\alpha_2(h_e). \end{aligned} \quad (18)$$

where $e = x - \bar{x} \in \mathbb{R}^n$ and $\boldsymbol{z} = [x^T, \bar{x}^T]^T \in \mathbb{R}^{2n}$. Furthermore, in the above QP $c_1 > 0$, $c_2 > 0$, $\alpha_1 \in \mathcal{K}^e$ and $\alpha_2 \in \mathcal{K}^e$. Let $u^*(x, \bar{x}, v)$ and $\gamma^*(x, \bar{x}, v)$ be the optimal solution to (18), the low level policy is defined as

$$\pi^u(x, \bar{x}, v) = u^*(x, \bar{x}, v). \tag{19}$$

Assumption 2. The Quadratic Program (QP) (18) is feasible for all $\boldsymbol{z} \in \mathcal{I} = \{\boldsymbol{z} = [x^T, \bar{x}^T]^T \in \mathbb{R}^{2n} : h_e(x - \bar{x}) \geq 0, h_x(x) \geq 0\}$ and for all $v \in \mathcal{V}$.

Proposition 1. *Consider the system* (15) *and* (19) *with* $v(t) \in \mathcal{V}, \forall t \geq 0$. *If Assumptions 1-2 hold, then the control policy* (19) *guarantees that Properties 1 and 2 are satisfied for the sets* $\mathcal{S}_x$ *and* $\mathcal{S}_e$ *from* (16).

Proof: The proof follows from [2]. ∎

Remark 2. We underline that Assumption 2 is satisfied for some $\alpha_1 \in \mathcal{K}^e$ and $\alpha_2 \in \mathcal{K}^e$ when the set $\mathcal{I}$ is robust control invariant for system (4) and mild assumptions on the Lie derivative of (4) hold (see [2] for further details). The set $\mathcal{I}$ may be hard to compute and standard techniques are based on HJB reachability analysis [5], SOS programming [7], Lyapunov-based methods [8] and Lipschitz properties of the system dynamics [11], [19].

B. Discrete Uncertain Model

The CLF-CBF QP (18) computes a control action which constraints the difference between the planned trajectory and the true system into $\mathcal{S}_e$. In this section, we leverage this property to construct a discrete time linear uncertain model, which over-approximates the evolution of the true system from $x^+(t_k)$ to $x^+(t_{k+1})$.

First, we define the following reset maps for the error and planning dynamics from (14):

$$\begin{aligned} \bar{x}^+ &= \Delta_{\bar{x}}(x^-) = \Delta(x^-) = x^+ \\ e^+ &= \Delta_e(e^-) = 0. \end{aligned} \tag{20}$$

Basically, the above reset maps set the planning state $\bar{x}$ equal to the true state x, and consequently the error state $e = 0$ after each kth discontinuous transition. We underline that setting $\bar{x}^+ = x^+$ is a design choice. It would be possible to design $\Delta_{\bar{x}}$, Δ and Δ_e such that $x^+ - \bar{x}^+ = e^+$ and let the high-level planner to select $\bar{x}^+$.

As the planning model is linear for all $t \in (t_k, t_{k+1})$, we have that

$$\bar{x}^-(t_{k+1}) = \bar{A}\bar{x}^+(t_k) + \bar{B}v^+(t_k), \tag{21}$$

where the transition matrices are $\bar{A} = e^{AT}$ and $\bar{B} = \int_0^T e^{A(T-\eta)} B d\eta$. We notice that, when Assumptions 1-2 hold, from Proposition 1 we have that $x^-(t_{k+1}) \in \{\bar{x}^-(t_{k+1})\} \oplus \mathcal{S}_e$. Furthermore, from equations (20)-(21) we have that

$$\begin{aligned} x^+(t_{k+1}) &= \Delta(x^-(t_{k+1})) \in \Delta(\{\bar{x}^-(t_{k+1})\} \oplus \bar{\mathcal{S}}_e) \\ &= \{T\bar{A}\bar{x}^+(t_k) + T\bar{B}v^+(t_k) + p\} \oplus T\bar{\mathcal{S}}_e \\ &= \{T\bar{A}x^+(t_k)) + T\bar{B}v^+(t_k)\} \oplus \Delta(\bar{\mathcal{S}}_e), \end{aligned} \tag{22}$$

where the polytope $\bar{\mathcal{S}}_e$ contains the set $\mathcal{S}_e$, i.e., $\mathcal{S}_e \subseteq \bar{\mathcal{S}}_e$. Equation (22) defines a discrete time uncertain linear system which can be used to check if Property 3 is satisfied, as stated by the following proposition. Notice that in (22) we used the definition of $\Delta(\cdot)$ from Assumption 1.

Proposition 2. *Let Assumptions 1-2 hold. Consider the autonomous discrete time uncertain system*

$$\begin{aligned} x_d^+(k+1) &= T\bar{A}x_d^+(k) + T\bar{B}\pi^v(x_d^+(k)) + \bar{w}_d^+(k) \\ x_d^+(0) &= x(0) = \bar{x}(0) = x_0, \end{aligned} \tag{23}$$

where the control policy $\pi^v : \mathbb{R}^n \to \mathcal{V}$ *and the disturbance* $\bar{w}_d^+(k) \in \Delta(\bar{\mathcal{S}}_e)$. *If the state of the above system* $x_d^+(k) \in \mathcal{X}_d \cap \mathcal{S}_x$, $\forall \bar{w}_d^+(k) \in \Delta(\bar{\mathcal{S}}_e)$ *and* $\forall k \in \{0, 1, \ldots\}$. *Then Property 3 is satisfied for the sets* $\mathcal{S}_x$ *and* $\mathcal{S}_e$ *from* (16) *and system* (15) *in closed-loop with*

$$\Pi_{\bar{x}}^v : \begin{cases} u(t) = \pi^u(x(t), \bar{x}(t), v(t)), & \dot{v}(t) = 0, & t \in \mathcal{T} \\ u^+(t) = u^-(t), & v^+(t) = \pi^v(x^+(t)), & t \in \mathcal{T}^v \end{cases}, \tag{24}$$

where π^u *is defined in* (19) *and* π^v *is the control policy from* (23).

Proof: First, we recursively define the k-steps robust reachable sets for the discrete time autonomous uncertain system (23) and for $k \in \{0, 1, \ldots\}$

$$\begin{aligned} \mathcal{R}_{k+1} = \{x^+ \in \mathbb{R}^n : &\exists x \in \mathcal{R}_k, \exists w \in \Delta(\bar{\mathcal{S}}_e), \\ &x^+ = T\bar{A}x + T\bar{B}\pi^v(x) + w\}, \end{aligned}$$

where $\mathcal{R}_0 = \{x_0\}$. Notice that by assumption $\mathcal{R}_k \subset \mathcal{X}_d \cap \mathcal{S}_x$, $\forall k \in \{0, 1, \ldots\}$, $\pi^v : \mathbb{R}^n \to \mathcal{V}$, $x(0) = \bar{x}(0) = x_0$ and $e(0) = 0$. Finally, Assumptions 1-2 and equations (22)-(23) imply that the state $x(t)$ of the closed-loop system (15) and (24) satisfies $x^+(t_k) \in \mathcal{R}_k \subset \mathcal{X}_d \cap \mathcal{S}_x$, $\forall k \in \{0, 1, \ldots\}$. Therefore, Property 3 is satisfied for the sets $\mathcal{S}_x$ and $\mathcal{S}_e$ from (16) and the closed-loop system (15) and (24). ∎

C. Model Predictive Control

In this section, we design a Model Predictive Controller that allows us to guarantee high level safety and high level tracking from Properties 3-4. In particular, we leverage the result from Proposition 2 and we design a robust tube MPC with time-varying cross section as in [20].

At time t_k given the state of the system $x(t_k)$ we solve the following finite time optimal control problem:

$$\begin{aligned} \min_{\boldsymbol{v}_t} \quad & \sum_{k=t}^{t+N} ||x_{k|t} - x_g||_Q + ||v_{k|t}||_R + ||x_{t+N|t} - x_g||_{Q_f} \\ \text{s.t.} \quad & x_{k+1|t} = T(\bar{A} + \bar{B}K)x_{k|t} + T\bar{B}v_{k|t} \\ & x_{t|t} = x^+(t_k) \\ & x_{k|t} \in \mathcal{X}_d \cap \mathcal{S}_x \ominus \mathcal{E}_k, \ v_{k|t} \in \mathcal{V} \ominus K\mathcal{E}_k \\ & x_{t+N|t} \in \mathcal{X}_F \ominus \mathcal{E}_{t+N}, \forall k = \{t, \ldots, t+N\} \end{aligned} \tag{25}$$

where K is a stabilizing feedback gain, $||x||_Q = x^T Q x$, $\mathcal{E}_{k+1} = T(\bar{A} + \bar{B}K)\mathcal{E}_k \oplus \Delta(\bar{\mathcal{S}}_e)$ and $\mathcal{E}_0 = \{0\}$. The above control problem computes a sequence of open loop actions $\boldsymbol{v}_t = [v_{t|t}, \ldots, v_{t+N|t}]$ which robustly steer system (23)

from the current state $x(t_k)$ to the terminal set $\mathcal{X}_F$, while minimizing the nominal cost and robustly satisfying state and input constraints [20]. Let $\boldsymbol{v}_t^* = [v_{t|t}^*, \ldots, v_{t+N|t}^*]$ be the optimal solution and $[x_{t|t}^*, \ldots, x_{t+N|t}^*]$ the associated optimal trajectory, then the MPC policy is

$$\pi^v\big(x^+(t_k)\big) = v_{t|t}^* + Kx_{t|t}^*. \tag{26}$$

Assumption 3. The terminal constraint set $\mathcal{X}_F \subset \mathcal{X}_d$ in (25) is a robust positive invariant set for the discrete time uncertain autonomous system $x(k+1) = T(\bar{A}+\bar{B}K)x(k)+w(k)$ with $w(k) \in \Delta(\bar{\mathcal{S}}_e)$ for all $k \in \{0, 1, \ldots\}$.

D. Closed-loop Properties

In this section, we show that Properties 1-4 hold for the closed-loop system (15), (19) and (26) and the sets in (16). Therefore, the closed-loop system satisfies state and input constraints (3).

Theorem 2. *Consider system* (15) *in closed-loop with* (19) *and* (26)*. Let Assumptions 1-3 hold. Assume that problem* (25) *is feasible at time* $t = 0$*, then the closed-loop system* (15)*,* (19) *and* (26) *satisfies state and input constraints* (3) *for all time* $t \geq 0$*.*

Proof: Notice that from Proposition 1 and equation (20), we have that the closed-loop system satisfies Properties 1, 2 and 4. Moreover from standard MPC arguments [20], [21], we have that the closed-loop system (23), where the control policy $\pi^v(\cdot)$ is the MPC policy (26), evolves inside $\mathcal{X}_d \cap \mathcal{S}_x$ and $\pi^v(x(t_k)) \in \mathcal{V}$ for all $k \in \{0, 1, \ldots\}$ (thus Property 3 hold). Concluding, Properties 1-4 are guaranteed for the closed-loop system (15), (19) and (26) and state and input constraints (3) are satisfied for all time $t \geq 0$. ∎

V. Simulation Results

We use the proposed strategy to steer a segway to a goal state[1], as shown in Figure 2. The state of the system are the position p_x, the velocity v_x, the rod angle θ and the angular velocity ω. The control action is the voltage commanded to the motor and the equations of motion used to simulate the system can be found in [1, Section IV.B]. The nominal model is obtained using a small angle approximation and the MPC is implemented for $Q = \text{diag}(0, 10^{-3}, 10^{-3}, 10^{-2})$, $R = 1$, $Q_F = \text{diag}(100, 100, 100, 200)$, $\mathcal{V} = \{v \in \mathbb{R} : ||v||_\infty \leq 20\}$ and $K = [0, -7.3989, -10.435, -3.7039]$. Finally, we implemented the CLF-CBF (18) for $\mathcal{S}_e = \{e \in \mathbb{R}^n : e^T Q_e e \leq 1\}$ with $Q_e = \text{diag}(1/0.2^2, 1/0.1^2, 1/0.05^2, 1/0.01^2)$, $\mathcal{X}_c = \mathbb{R}^n$ and $\mathcal{U} = \mathbb{R}$.

A. Unconstrained Example with Low Frequency Update

In this example, we run the high level MPC planner at 2Hz, we set $\mathcal{X}_d = \mathbb{R}^n$ and the MPC horizon $N = 10$. Figure 3 shows the closed-loop trajectories for the proposed strategy, a linear MPC and nonlinear MPC policies, which are implemented at 100Hz and 20Hz for prediction horizons $N^{100\text{Hz}} = 500$ and $N^{20\text{Hz}} = 100$, respectively. All strategies

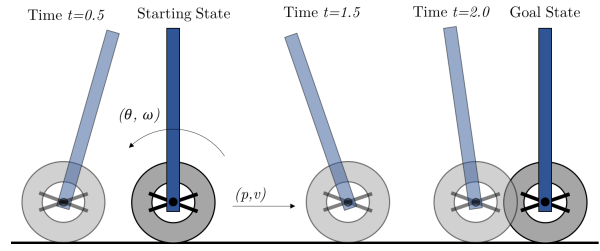[1]Code available at https://github.com/urosolia/MultiRate

Fig. 2. The goal of the controller is to steer the segway to a goal state while keeping the rod upright. Furthermore, we reported snippets of the state trajectory from Section V-A at 0.5s,1.5s and 2s.

plan the desired trajectory over a receding time window of 5 seconds. We notice that the linear MPC overshoots the goal state and the nonlinear MPC discretized at 20Hz oscillates before reaching the target state. On the other hand, the proposed strategy performs similarly to the high frequency nonlinear MPC (discretized at 100Hz with prediction horizon $N^{100\text{Hz}} = 500$), while being implemented with a 2Hz model update rate, a prediction horizon $N = 10$ and solving convex optimization problems. This example shows the advantage of the proposed multi-rate architecture, where the high level control action is updated at a lower frequency than the low level input command, as shown in Figure 4.
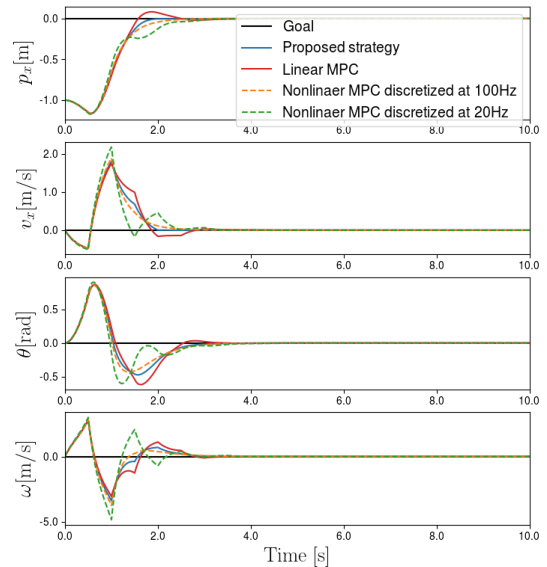

Fig. 3. Closed-loop trajectories for the proposed strategy, linear and nonlinear MPC policies. The evolution of the position x and velocity v underline that the proposed methodology steers the system to the goal state without overshooting.

B. Constrained Example with High Frequency Update

In this example, we run the high level MPC planner at 10Hz, we set $\mathcal{X}_d = \{x = [p_x, v_x, \theta, \omega]^T \in \mathbb{R}^n : |\theta| \leq 0.78\}$ and the MPC horizon $N = 10$. We compare the proposed strategy with linear MPCs and nonlinear MPCs discretized at 10Hz, 20Hz and 100Hz. Also for linear and nonlinear MPCs, we use the constraint tightening from (25). Figure 5 shows that when the high frequency input from the low level controller is not used, the closed-loop system
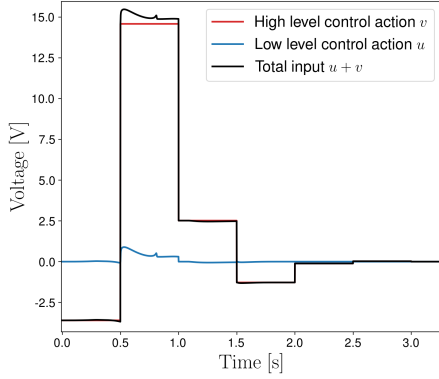
Fig. 4. The total input applied to the system is given by the summation of the high level control action from the MPC (in red) and the low level CLF-CBF QP (in blue). We notice that the high level input is updated at 2Hz, whereas the low level input is updated at 1000Hz.

violates the state constraints. We underline that constraint satisfaction for nonlinear MPC policies can be guaranteed using the approaches from [8]–[12]. However, this example shows the advantage of using the high frequency low level controller to reduce the tracking error. Indeed, when the low level controller is not used, the constraint tightening from (25) is not sufficient to guarantee constraint satisfaction, both when linear and nonlinear models are leveraged for planning. Finally, we underline that the computational cost associated with the proposed strategy is ~ 0.1s. Whereas, the computational cost associated with linaer MPCs discretized at 10Hz, 20Hz and 100Hz is ~ 0.1s, ~ 0.25s and ~ 1s, respectively.
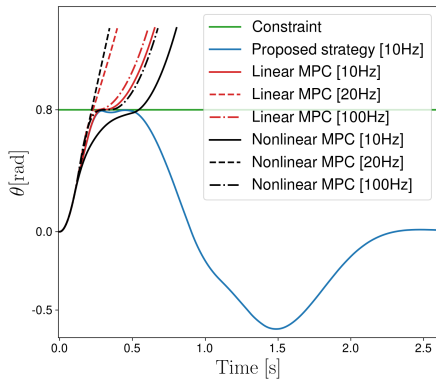

Fig. 5. Evolution of the rod angle θ for the proposed strategy, the linear and nonlinear model predictive controllers. When the low level controller is not used the closed-loop system violates the state constraints.

VI. CONCLUSIONS

In this paper, we presented a multi-rate control architecture, where the high level planner and the low level controller run at different frequencies. First, we introduced sufficient conditions which guarantee recursive constraint satisfaction

for the closed-loop system. Afterwards, we presented a controller design which leverages control barrier functions and MPC policies.

REFERENCES

[1] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018, pp. 98–106.

[2] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, Aug 2017.

[3] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[4] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 7130–7135.

[5] 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 *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 1517–1522.

[6] H. Yin, M. Bujarbaruah, M. Arcak, and A. Packard, "Optimization based planner tracker design for safety guarantees," *arXiv preprint arXiv:1910.00782*, 2019.

[7] 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," *arXiv preprint arXiv:1808.00649*, 2018.

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

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

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

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

[12] J. Köhler, R. Soloperto, M. A. Muller, and F. Allgower, "A computationally efficient robust model predictive control framework for uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, 2020.

[13] J. Luo, Y. Su, L. Ruan, Y. Zhao, D. Kim, L. Sentis, and C. Fu, "Robust bipedal locomotion based on a hierarchical control structure," *Robotica*, vol. 37, no. 10, pp. 1750–1767, 2019.

[14] J. Reher, W.-L. Ma, and A. D. Ames, "Dynamic walking with compliance on a cassie bipedal robot," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2589–2595.

[15] U. Rosolia, S. De Bruyne, and A. G. Alleyne, "Autonomous vehicle control: A nonconvex approach for obstacle avoidance," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 469–484, 2016.

[16] N. R. Kapania and J. C. Gerdes, "Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling," *Vehicle System Dynamics*, vol. 53, no. 12, pp. 1687–1704, 2015.

[17] Y. Chen, J. Anderson, K. Kalsi, A. D. Ames, and S. H. Low, "Safety-critical control synthesis for network systems with control barrier functions and assume-guarantee contracts," *arXiv preprint arXiv:1911.03452*, 2019.

[18] M. Farina, X. Zhang, and R. Scattolini, "A hierarchical mpc scheme for coordination of independent systems with shared resources and plug-and-play capabilities," *IEEE Transactions on Control Systems Technology*, 2018.

[19] Y. Chen, H. Peng, J. Grizzle, and N. Ozay, "Data-driven computation of minimal robust control invariant set," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4052–4058.

[20] L. Chisci, J. A. Rossiter, and G. Zappa, "Systems with persistent disturbances: predictive control with restricted constraints," *Automatica*, vol. 37, no. 7, pp. 1019–1028, 2001.

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