2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Leveraging the Template and Anchor Framework for Safe, Online Robotic Gait Design

Jinsun Liu*, Pengcheng Zhao*, Zhenyu Gan, Matthew Johnson-Roberson, and Ram Vasudevan

Abstract— **Online control design using a high-fidelity, full-order model for a bipedal robot can be challenging due to the size of the state space of the model. A commonly adopted solution to overcome this challenge is to approximate the full-order model (anchor) with a simplified, reduced-order model (template), while performing control synthesis. Unfortunately it is challenging to make formal guarantees about the safety of an anchor model using a controller designed in an online fashion using a template model. To address this problem, this paper proposes a method to generate safety-preserving controllers for anchor models by performing reachability analysis on template models by relying on functions that bound the difference between the two models. This paper describes how this reachable set can be incorporated into a Model Predictive Control framework to select controllers that result in safe walking on the anchor model in an online fashion. The method is illustrated on a 5-link RABBIT model, and is shown to allow the robot to walk safely while utilizing controllers designed in an online fashion.**

Bipeds, underactuated system, safety guarantee.

I. Introduction

Legged robots are ideal systems to perform locomotion on unstructured terrains. Unfortunately designing controllers for legged systems to operate safely in such situations has proven challenging. To robustly traverse such environments, an ideal control synthesis technique for legged robotic systems should satisfy several requirements. First, since uncertainties and disturbances may appear during operation, any algorithm for control synthesis should run in real-time. Second, since modeling contact can be challenging, any control synthesis technique should be able to accommodate model uncertainty. Third, since the most appropriate controller may be a function of the environment and given task, a control synthesis algorithm should optimize over as rich a family of control inputs at run-time as possible. Finally, since falling can be costly both in time and expense, a control synthesis technique should be able to guarantee the satisfactory behavior of any constructed controller. As illustrated in Fig. 1, this paper presents an optimization-based algorithm to design gaits for legged robotic systems while satisfying each of these requirements.

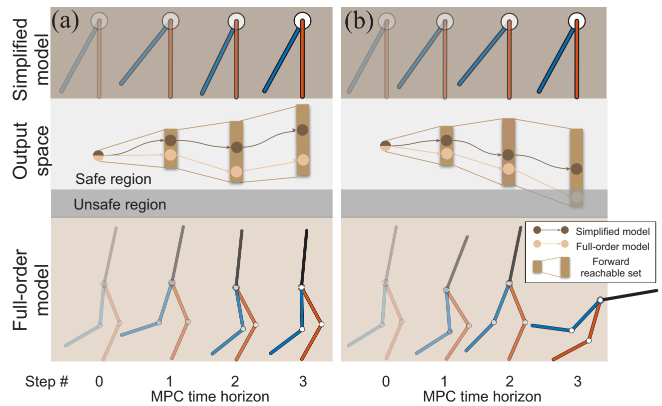

Fig. 1: This paper proposes a method to design gaits that are certified to be tracked by a full-order robot model (bottom row sub-figures) for N-steps without falling over. To construct this method, this paper defines a set of outputs that are functions of the state of the robot and a chosen gait (middle row sub-figures). If the outputs associated with a particular gait remain within the safe region, which is described by a set of inequality constraints, then the gait is proven to be safely tracked by the legged system without falling. Due to the high-dimensionality of the robot's dynamics, forward propagating these outputs via the robot's dynamics for N-steps to design a gait that is certified to be tracked safely is intractable. To address this challenge, this paper constructs a template model (top row sub-figures) whose outputs are sufficient to predict the behavior of the anchor's outputs. In particular, if all of the points in a bounded neighborhood of the forward reachable set of the outputs of the template model remain within the safe region, then the anchor is certified to behave safely.

We begin by summarizing related work with an emphasis on techniques that are able to make guarantees on the safety of the designed controller. For instance, the Zero-Moment Point approach [1] characterizes the stability of a legged robot with planar feet by defining the notion of the Zero-Moment Point and requiring that it remains within the robot's base of support. Though this requirement can be used to design a controller that can avoid falling at run-time, the gaits designed by the ZMP approach are energetically expensive [2], [3].

In contrast, the Hybrid Zero Dynamics approach, which relies upon feedback linearization to drive the actuated degrees of freedom of a robot towards a lower dimensional manifold, is able to synthesize a controller which generates gaits that are more dynamic. Though this approach can generate safety preserving controllers for legged systems in real-time in the presence of model uncertainty [4]–[8], it is only able to prove that the gait associated with a synthesized control is locally stable. As a result, it is non-trivial to switch between

Jinsun Liu is with the Robotics Institute, University of Michigan, Ann Arbor, MI 48109 jinsunl@umich.edu

Pengcheng Zhao, Zhenyu Gan and Ram Vasudevan are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 {pczhao,ganzheny,ramv}@umich.edu

Matthew Johnson-Roberson is with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109 mattjr@umich.edu

This work is supported by the Ford Motor Company via the Ford-UM Alliance under award N022977 and by the Office of Naval Research under Award Number N00014-18-1-2575.

*These two authors contributed equally to this work.

multiple constructed controllers while preserving any safety guarantee. Recent work has extended the ability of the hybrid zero dynamic approach beyond a single neighborhood of any synthesized gait [9]–[12]. These extensions either assume full-actuation [11] or ignore the behavior of the legged system off the lower dimensional manifold [9], [10], [12].

Rather than designing controllers for legged systems, other techniques have focused on characterizing the limits of safe performance by using Sums-of-Squares (SOS) optimization [13]. These approaches use semi-definite programming to identify the limits of safety in the state space of a system as well as associated controllers for hybrid systems [14], [15]. These safe sets can take the form of *reachable sets* [15], [16] or *invariant sets* in state space [14], [17], [18]. However, the representation of each of these sets in state space restricts the size of the problem that can be tackled by these approaches and as a result, these SOS-based approaches have been primarily applied to reduced models of walking robots: ranging from spring mass models [19], to inverted pendulum models [16], [20] and to inverted pendulum models with an offset torso mass [18]. Unfortunately the differences between these simple models and real robots makes it challenging to extend the safety guarantees to more realistic real-world models.

This paper addresses the shortcomings of prior work by making the following four contributions. First, in Section III-A, we describe a set of outputs that are functions of the state of the robot, which can be used to determine whether a particular gait can be safely tracked by a legged system without falling. To design gaits over N-steps that do not fall over, one could begin by forward propagating these outputs via the robot's dynamics for N-steps. Unfortunately performing this computation can be intractable due to the high-dimensionality of the robot's dynamics. To address this challenge, our second contribution, in Section III-B, leverages the anchor and template framework to construct a simple model (template) whose outputs are sufficient to predict the behavior of the full model's (anchor's) outputs [21] under the assumption that the modeling error between the anchor and template can be bounded. Third, in Section IV-A, we develop an offline method to compute a gait parameterized forward reachable set that describes the evolution of the outputs of the simple model.

Similar to recently developed work on motion planning for ground and aerial vehicles [22]–[26], one can then require that all possible outputs in the forward reachable set satisfy a family of conditions to ensure that the robot does not fall over during the N-steps. Finally, in Section IV-B , we describe how to incorporate these conditions in a Model Predictive Control (MPC) framework that are sufficient to ensure N-step walking safely. Note, to simplify exposition, this paper focuses on an example implementation on a 14-dimensional model of the robot RABBIT that is described in Section II. Section V demonstrates the performance of the proposed approach on a walking example.

II. Preliminaries

The following notation is adopted in this manuscript. All sets are denoted using calligraphic capital letters. Let $\mathbb{R}$ denote the set of real numbers, and let $\mathbb{N}_+$ denote the collection of all non-negative integers. Give a set $\mathcal{A} \subset \mathbb{R}^n$ for some $n \in \mathbb{N}_+$, let $C^1(\mathcal{A})$ denote the set of all differentiable continuous functions from $\mathcal{A}$ to $\mathbb{R}$ whose derivative is continuous and let $\lambda_{\mathcal{A}}$ denote the Lebesgue measure which is supported on $\mathcal{A}$.

A. RABBIT Model (Anchor)

Considers the walking motion of a planar 5-link model of RABBIT [27], which consists of alternating phases of *single stance* (one leg in contact with the ground) and *double stance* (both legs in contact with the ground). While in single stance, the leg in contact with the ground is called the *stance leg*, and the non-stance leg is called the *swing leg*. The double stance phase is instantaneous. The configuration of the robot at time t is $q(t) := [q_h(t), q_v(t), q_1(t), q_2(t), q_3(t), q_4(t), q_5(t)]^\top \in Q \subset \mathbb{R}^7$, where $(q_h(t), q_v(t))$ are Cartesian position of the robot hip; $q_1(t)$ is the torso angle relative to the upright direction; $q_2(t)$ and $q_4(t)$ are the hip angles relative to stance and swing leg, respectively; and $q_3(t)$ and $q_5(t)$ are the knee angles. The joints (q_2, q_3, q_4, q_5) are actuated, and q_1 is an underactuated degree of freedom. Let $\theta(q) := q_1 + q_2 + q_3/2$ denote the *stance leg angle*, and let $\phi(q) := q_1 + q_4 + q_5/2$ denote the *swing leg angle*. We refer to the configuration when the robot hip is right above the stance foot, i.e. $\theta = \pi$, as *mid-stance*. We refer to the motion between the i-th and $(i+1)$-st swing leg foot touch down with the ground as the *i-th step*.

Using the method of Lagrange, we can obtain a continuous dynamic model of the robot during swing phase:

$$\dot{a}(t) = f(a(t), u(t)) \tag{1}$$

where $a(t) = [q^\top(t), \dot{q}^\top(t)]^\top \in TQ \subset \mathbb{R}^{14}$ denotes the tangent bundle of Q, $u(t) \in U$. U describes the permitted inputs to the system, and t denotes time. We model the RABBIT as a hybrid system and describe the instantaneous change using the notation of a *guard* and a *reset map*. That is, suppose $(\theta(q(t)), c_{\text{foot}}(q(t)))$ denotes the stance leg angle and the vertical position of the swing foot relative to the stance foot, respectively, given a configuration $q(t)$ at time t. The guard $\mathcal{G}$ is $\{(b,b') \in TQ \mid \pi/2 < \theta(b) < 3\pi/2, c_{\text{foot}}(b) = 0 \text{ and } \dot{c}_{\text{foot}}(b,b') < 0\}$. Notice the force of the ground contact imposes a holonomic constraint on stance foot position, which enables one to obtain a reset map [3, Section 3.4.2] $\dot{q}^+(t) = \Delta(\dot{q}^-(t))$, where Δ describes the relationship between the pre-impact and post impact velocities. More details about the definition and derivation of this hybrid model can be found in [3, Section 3.4].

To simplify exposition, this paper at run-time optimizes over a family of reference gaits that are characterized by their average velocity and step length. These reference gaits are described by a vector of *control parameters* $P(i) = (p_1(i), p_2(i)) \in \mathcal{P}$ for all $i \in \mathbb{N}$, where $p_1(i)$ denotes the average horizontal velocity and $p_2(i)$ denotes the step length between the i-th and $(i+1)$-st mid-stance position. Note $\mathcal{P}$ is compact.

These reference gaits are generated by solving a finite family of nonlinear optimization problems in which we incorporate $p_1(i)$, $p_2(i)$, and periodicity as constraints, and minimize the average torque squared over the gait period [28]. Each of these problems yields a reference trajectory parameterized by $P(i)$ and interpolation is applied over these generated gaits to generate a continuum of gaits. Given a control parameter, a control input into the RABBIT model is generated by tracking the corresponding reference trajectories using a PD controller.

Next, we define a solution to the hybrid model as a pair $(\mathcal{I}, a)$, where $\mathcal{I} = \{I_i\}_{i=0}^N$ is a *hybrid time set* with I_i being intervals in $\mathbb{R}$, and $a = \{a_i(\cdot)\}_{i=0}^N$ is a finite sequence of functions with each element $a_i(\cdot) : I_i \to TQ$ satisfying the dynamics (1) over I_i where $N \in \mathbb{N}$ [29, Definitions 3.3, 3.4, 3.5]. Denote each $I_i := [\tau_i^+, \tau_{i+1}^-]$ for all $i < N$. τ_i corresponds to the time of the transition between $(i-1)$-th to i-th step. We let τ_i^- correspond to the time just before the transition and and τ_i^+ correspond to the time just after the transition. Since transitions are assumed to be instantaneous, $\tau_i = \tau_i^- = \tau_i^+$ if all values exist. When a transition never happens during the i-th step, we denote $\tau_{i-1}^- = +\infty$. Note when $\tau_{i+1} < \infty$, $a_i(\tau_{i+1}^-) \in \mathcal{G}$ and $a_{i+1}(\tau_{i+1}^+) \in \Delta(a_i(\tau_{i+1}^-))$.

B. Simplified Biped Model (Template)

Performing online optimization with the *Simplified Biped Model* (SBM) adopted from [30] in contrast with the full RABBIT model is tractable. This model consists of a point-mass M and two mass-less legs each with a constant length l. The configuration of the SBM at time t is described by the stance leg angle, $\hat{\theta}$, and the swing leg angle, $\hat{\phi}$. The guard is the set of configurations when $\hat{\theta} + \hat{\phi} = 2\pi$. The swing leg swings immediately to a specified step length. During the swing phase, one can use the method of Lagrange to describe the evolution of the configuration as a function of the current configuration and the input. Subsequent to the instantaneous double stance phase, an impact with the ground happens with a coefficient of restitution of 0. We denote a hybrid execution of the SBM as a pair $(\hat{\mathcal{I}}, \hat{a})$ where $\hat{\mathcal{I}} = \{\hat{I}_i\}_{i=0}^N$ is a hybrid time set with $\hat{I}_i := [\hat{\tau}_i^+, \hat{\tau}_{i+1}^-]$ and $\hat{a} = \{\hat{a}_i(\cdot)\}_{i=0}^N$ is a finite sequence of solutions to the SBM's equations of motion.

III. Outputs to Describe Successful Walking

During online optimization, we optimize over the space of parameterized inputs while introducing a constraint to guarantee that the robot does not fall over. This section formalizes what it means for the RABBIT model to walk successfully without falling over. Section III-A defines a set of outputs that are functions of the state of RABBIT and proves that the value of these outputs can determine whether RABBIT is able to walk successfully. Subsequently in Section III-B, we define a corresponding set of outputs that are functions of the state of the SBM and illustrate how their values can be used to determine whether RABBIT is able to walk successfully.

To define successful walking on RABBIT, we begin by defining the time during step i at which mid-stance occurs

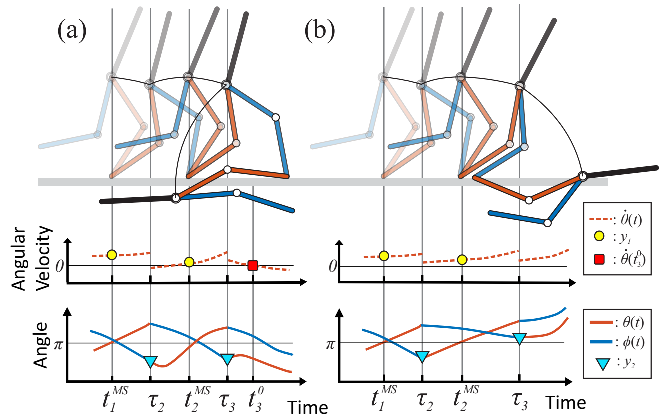

Fig. 2: An illustration of how the values of the outputs can be used to determine whether the robot walks safely. To ensure that the robot does not fall backwards, one can require that $y_1(i) \geq 0$ (left column). To ensure that the robot does not fall forward, one can require that $y_2(i) \leq \pi$ (right column).

(i.e., the largest time t at which $\theta(q(t)) = \pi$ during I_i) as

$$t_i^{MS} := \begin{cases} +\infty, & \text{if } \theta(q(t)) < \pi \quad \forall t \in I_i, \\ -\infty, & \text{if } \theta(q(t)) > \pi \quad \forall t \in I_i, \\ \max\{t \in I_i \mid \theta(q(t)) = \pi\}, & \text{otherwise.} \end{cases} \quad (2)$$

Note if mid-stance is never reached during step i, then the mid-stance time is defined as plus or minus infinity depending upon if the hip-angle remains less than π or greater than π during step i, respectively. Using this definition, we formally define successful walking for the RABBIT model as:

Definition 1. *The RABBIT model* walks successfully *in step $i \in \mathbb{N}$ if $t_i^{MS} \neq \pm\infty$, $\pi/2 < \theta(q(t)) < 3\pi/2$ for all $t \in I_i$, and $\tau_{i+1}^- < +\infty$.*

The first requirement ensures that mid-stance is reached, the second requirement ensures that the hip remains above the ground, and the final requirement ensures that the swing leg actually makes contact with the ground.

A. Outputs to Describe Successful RABBIT Walking

We begin by defining another time variable t_i^0:

$$t_i^0 := \begin{cases} \tau_i^+, & \text{if } \dot{\theta}(q(t), \dot{q}(t)) < 0 \quad \forall t \in I_i, \\ \tau_{i+1}^-, & \text{if } \dot{\theta}(q(t), \dot{q}(t)) > 0 \quad \forall t \in I_i, \\ \max\{t \in I_i \mid \dot{\theta}(q(t), \dot{q}(t)) = 0\}, & \text{otherwise.} \end{cases} \quad (3)$$

Note t_i^0 is defined to be the last time in I_i when a sign change of $\dot{\theta}$ occurs; when a sign change does not occur, t_i^0 is defined as an endpoint of I_i associated with the sign of $\dot{\theta}$.

We first define an output, $y_1 : \mathbb{N} \to \mathbb{R}$ that can be used to ensure that $t_i^{MS} \neq +\infty$:

$$y_1(i) := \begin{cases} \dot{\theta}(q(t_i^{MS}), \dot{q}(t_i^{MS})), & \text{if } t_i^{MS} \neq \pm\infty, \\ -\sqrt{2g(l_{\text{st}}(t_i^0) - q_v(t_i^0))}/l_{\text{st}}(t_i^0), & \text{if } t_i^{MS} = +\infty, \\ 1 & \text{if } t_i^{MS} = -\infty, \end{cases} \quad (4)$$

where g is gravity and $l_{\text{st}}(t_i^0)$ is the stance leg length at time t_i^0. Note that $y_1(i)$ is the hip angular velocity when the mid-stance position is reached during the i-th step. When the mid-stance position is not reached, $-y_1(i)$ represents the additional hip angular velocity needed to reach the mid-stance position. In particular, notice $t_i^{MS} \neq +\infty$ whenever $y_1(i) \geq 0$.

Next, we define an output $y_2 : \mathbb{N} \to \mathbb{R}$ that can be used to ensure that $t_i^{MS} \neq -\infty$:

$$y_2(i) := \begin{cases} \phi(q(\tau_{i+1}^-)), & \text{if } \tau_{i+1}^- < +\infty, \\ 2\pi, & \text{otherwise.} \end{cases} \quad (5)$$

Note, $y_2(i)$ is the swing leg angle at touch-down at the end of the i-th step; if touch-down does not occur, $y_2(i)$ is defined as 2π. Recall $\phi(q(\tau_{i+1}^-)) = \theta(q(\tau_{i+1}^+))$, so if $y_2(i) \leq \pi$, it then follows from (2) and (5) that $t_{i+1}^{MS} \neq -\infty$ and $\tau_{i+1}^- < +\infty$. Fig. 2 illustrates the behavior of y_1 and y_2.

We now define our last two outputs $y_3, y_4 : \mathbb{N} \to \mathbb{R} \cup \{-\infty, +\infty\}$ that can be used to ensure that the hip stays above the ground:

$$y_3(i) := \begin{cases} \inf\{\theta(q(t)) \mid t \in [t_i^{MS}, t_{i+1}^{MS}]\}, & \text{if } t_{i+1}^{MS}, t_i^{MS} \in \mathbb{R}, \\ -\infty, & \text{otherwise.} \end{cases} \quad (6)$$

$$y_4(i) := \begin{cases} \sup\{\theta(q(t)) \mid t \in [t_i^{MS}, t_{i+1}^{MS}]\}, & \text{if } t_{i+1}^{MS}, t_i^{MS} \in \mathbb{R}, \\ +\infty, & \text{otherwise.} \end{cases} \quad (7)$$

Finally, we let $\mathcal{Y} := \mathbb{R} \times \mathbb{R} \times (\mathbb{R} \cup \{-\infty, +\infty\}) \times \mathbb{R}$.

The outputs are defined based on the observation that the hip usually has forward speed (e.g. moving forward, rather than falling backwards) at mid-stance and appears between the two legs at touch-down when the RABBIT model walks safely. Specifically, y_1 represents the hip angular velocity at mid-stances and y_2 represents the swing leg angle at touch-downs. y_3 and y_4 are defined as the maximum and minimum stance leg angles between adjacent mid-stances, which are used to indicate whether the hip hits the ground.

Using these definitions, we can prove the following theorem that constructs a sufficient condition to ensure successful walking by RABBIT.

Theorem 2. *Suppose that the* 0*-th step can be successfully completed (i.e.* τ_0^+ *and* t_0^{MS} *are finite,* $\inf\{\theta(q(t)) \mid t \in [\tau_0^+, t_0^{MS}]\} > \pi/2$, *and* $\sup\{\theta(q(t)) \mid t \in [\tau_0^+, t_0^{MS}]\} < 3\pi/2$*)). Suppose* $y_1(i) \geq 0$, $y_2(i) \leq \pi$, $y_3(i) > \pi/2$ *and* $y_4(i) < 3\pi/2$ *for each* $i \in \{0, \cdots, N\}$*, then the robot walks successfully at the i-th step for each* $i \in \{0, \cdots, N\}$.

Proof. Notice $y_1(i) \geq 0 \Rightarrow t_i^{MS} \neq +\infty$ and $y_2(i) \leq \pi \Rightarrow t_{i+1}^{MS} \neq -\infty$ for each $i \in \{1, \cdots, N\}$. By induction we have t_i^{MS} is finite $\forall i \in \{1, \cdots, N\}$. $y_2(i) \leq \pi < 2\pi$ implies that $\tau_{i+1}^- < +\infty$. By using the definitions of y_3 and y_4, one has that the robot walks successfully in the i-th step based on Definition 1. □

B. Approximating Outputs Using the SBM

Finding an analytical expression describing the evolution of each of the outputs can be challenging. Instead we define corresponding outputs $\hat{y}(i) := (\hat{y}_1(i), \hat{y}_2(i), \hat{y}_3(i), \hat{y}_4(i)) \in \mathcal{Y}$ for SBM. Such outputs $\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4$ are directly comparable to those of the RABBIT model[1], and a bound between them can be established as described next. The discrete-time dynamics of each of these outputs of SBM can be described by the following difference equations:

$$\begin{aligned} \hat{y}_1(i+1) &= f_{\hat{y}_1}(\hat{y}_1(i), P(i)) \\ \hat{y}_2(i) &= f_{\hat{y}_2}(P(i)) \\ \hat{y}_3(i) &= f_{\hat{y}_3}(\hat{y}_1(i), P(i)) \\ \hat{y}_4(i) &= f_{\hat{y}_4}(\hat{y}_1(i), P(i)) \end{aligned} \quad (8)$$

for each $i \in \mathbb{N}$, $\hat{y}(i) \in \mathcal{Y}$, and $P(i) \in \mathcal{P}$. Such functions $f_{\hat{y}_1}$, $f_{\hat{y}_2}$, $f_{\hat{y}_3}$ and $f_{\hat{y}_4}$ can be generated using elementary mechanics.

To describe the gap between the discrete signals y and $\hat{y}$ we make the following assumption:

Assumption 3. *For any sequence of control parameters,* $\{P(i)\}_{i \in N}$*, and corresponding sequences of outputs,* $\{y_1(i), y_2(i), y_3(i), y_4(i)\}_{i \in \mathbb{N}}$ *and* $\{\hat{y}_1(i), \hat{y}_2(i), \hat{y}_3(i), \hat{y}_4(i)\}_{i \in \mathbb{N}}$*, generated by the RABBIT dynamics and* (8)*, respectively, there exists* bounding functions $\underline{B}_1$, $\overline{B}_1 : \mathbb{R} \times \mathcal{P} \to \mathbb{R}$, $\overline{B}_2 : \mathcal{P} \times \mathbb{R} \times \mathcal{P} \to \mathbb{R}$*, and* $\underline{B}_3$, $\overline{B}_4 : \mathbb{R} \times \mathcal{P} \to \mathbb{R}$ *satisfying*

$$\underline{B}_1(y_1(i), P(i)) \leq y_1(i+1) - \hat{y}_1(i+1) \leq \overline{B}_1(y_1(i), P(i)) \quad (9)$$

$$y_2(i) - \hat{y}_2(i) \leq \overline{B}_2(P(i-1), y_1(i), P(i)) \quad (10)$$

$$y_3(i) - \hat{y}_3(i) \geq \underline{B}_3(y_1(i), P(i)) \quad (11)$$

$$y_4(i) - \hat{y}_4(i) \leq \overline{B}_4(y_1(i), P(i)). \quad (12)$$

In other words, if $y_1(i) = \hat{y}_1(i)$, then $\overline{B}_1$, $\underline{B}_1$, $\underline{B}_2$, $\underline{B}_3$, and $\overline{B}_4$ bound the maximum possible difference between $(y_1(i+1), y_2(i), y_3(i), y_4(i))$ and $(\hat{y}_1(i+1), \hat{y}_2(i), \hat{y}_3(i), \hat{y}_4(i))$. One could apply SOS optimization to generate them [31] or bound the dynamics of the system [26]. Note, constructing such a bound precisely using SOS optimization can be challenging due to the high-dimensionality of the full-order model. However, several papers have proposed techniques that have begun to address these scaling challenges while applying SOS optimization [32]–[35]. To simplify further exposition, we define the following:

$$\begin{aligned} \mathcal{B}(y_1(i), P(i)) := [&f_{\hat{y}_1}(y_1(i), P(i)) + \underline{B}_1(y_1(i), P(i)), \\ &f_{\hat{y}_1}(y_1(i), P(i)) + \overline{B}_1(y_1(i), P(i))] \end{aligned} \quad (13)$$

for all $(y_1(i), P(i)) \in \mathbb{R} \times \mathcal{P}$. In particular, it follows from (9) that for any sequence of control parameters, $\{P(i)\}_{i \in N}$, and corresponding sequences of outputs, $\{y_1(i)\}_{i \in \mathbb{N}}$ generated by the RABBIT dynamics, $y_1(i+1) \in \mathcal{B}(y_1(i), P(i))$ for all $i \in \mathbb{N}$.

IV. Enforcing N-Step Safe Walking

This section proposes an online MPC framework to design a controller for the RABBIT model that can ensure successful walking for N-step. In fact, when $N = 1$ one can directly apply Theorem 2 and Assumption 3 to generate the following inequality constraints over $y_1(i)$, $P(i-1)$ and $P(i)$ to guarantee walking successfully from the i-th to the $(i+1)$-th mid-stance:

$$f_{\hat{y}_1}(y_1(i), P(i)) + \underline{B}_1(y_1(i), P(i)) \geq 0, \quad (14)$$

$$f_{\hat{y}_2}(P(i)) + \overline{B}_2(P(i-1), y_1(i), P(i)) \leq \pi, \quad (15)$$

$$f_{\hat{y}_3}(y_1(i), P(i)) + \underline{B}_3(y_1(i), P(i)) > \pi/2, \quad (16)$$

$$f_{\hat{y}_4}(y_1(i), P(i)) + \overline{B}_4(y_1(i), P(i)) < 3\pi/2. \quad (17)$$

[1]Detailed definition of $\hat{y}(i)$ can be found in `https://github.com/pczhao/TA_GaitDesign/blob/master/SBM_dynamics.pdf`

Unfortunately, to construct a similar set of constraints when $N > 1$, one has to either compute $(y_1(i+M), y_2(i+M), y_3(i+M), y_4(i+M))$ for each $1 \leq M \leq N$, which can be computationally taxing, or one can apply (9) recursively to generate an outer approximation to $y_1(i+M)$ for each $1 \leq M \leq N$ and then apply the remainder of Assumption 3 to generate an outer approximation to $y_2(i+M), y_3(i+M)$, and $y_4(i+M)$ for each $1 \leq M \leq N$. In the latter instance, one would need the entire set of possible values for the outputs to satisfy conditions in Theorem 2 from the i-th step to the $(i+N)$-th step to ensure N-step safe walking. This requires introducing set inclusion constraints that can be cumbersome to enforce at run-time. To address these challenges, Section IV-A describes how to compute in an offline fashion, an N-step *Forward Reachable Set* (FRS) that captures all possible outcomes for the output y_1 from a given initial state and set of control parameters for up to N steps. Subsequently, Section IV-B illustrates how to write down N-step successful walking conditions on outputs (y_2, y_3, y_4) and set up an MPC framework to update gait parameters for RABBIT using nonlinear programming with set inclusion constraints.

A. Forward Reachable Set

Letting $\mathcal{Y}_1 \subset \mathbb{R}$ be compact, we define the *N-step FRS of the output* y_1:

Definition 4. *The N-step FRS of the output beginning from* $(y_1(i), P(i)) \in \mathcal{Y}_1 \times \mathcal{P}$ *for* $i \in \mathbb{N}$ *and for* $N \in \mathbb{N}$ *is defined as*

$$\mathcal{W}_N(y_1(i), P(i)) := \bigcup_{n=i+1}^{i+N} \Big\{ y_1(n) \in \mathcal{Y}_1 \mid \exists P(i+1), \ldots, P(n-1) \in \mathcal{P} \text{ such that } \forall j \in \{i, \ldots, i+n-1\}, \; y_1(j+1) \text{ is generated by the RABBIT dynamics from } y_1(j) \text{ under } P(j) \Big\} \tag{18}$$

In other words, given a fixed output $y_1(i)$ and the current control parameter $P(i)$, the FRS $\mathcal{W}_N$ captures all the outputs $y_1(j)$ that can be reached within N steps, provided that all subsequent control parameters are contained in a set $\mathcal{P}$. Since $\mathcal{W}_N$ is the union of all possible y_1 within the next N steps, it follows that:

$$\mathcal{W}_M(y_1(i), P(i)) \subseteq \mathcal{W}_N(y_1(i), P(i)) \quad \forall 1 \leq M \leq N. \tag{19}$$

As a result of (19), to predict the behavior of RABBIT system over N steps, it is unnecessary to compute distinct FRS-es for each of the next N steps. Instead one only needs to compute a single FRS.

To compute an outer approximation of the FRS, inspired by [36], one can solve the following infinite-dimensional linear problem over the space of functions:

$$\begin{aligned}
\inf_{w_N, v_1, \cdots, v_N} \; & \int_{\mathcal{Y}_1 \times \mathcal{P} \times \mathcal{Y}_1} w_N(x_1, x_2, x_3) \, d\lambda_{\mathcal{Y}_1 \times \mathcal{P} \times \mathcal{Y}_1} \qquad \text{(FRSopt)} \\
\text{s.t.} \quad & v_1(x_1, x_2, x_3) \geq 0, \\
& \qquad \forall x_3 \in \mathcal{B}(x_1, x_2) \\
& \qquad \forall (x_1, x_2) \in \mathcal{Y}_1 \times \mathcal{P} \\
& v_{\zeta+1}(x_1, x_2, x_4) \geq v_\zeta(x_1, x_2, x_3), \\
& \qquad \forall \zeta \in \{1, 2, \cdots, N-1\} \\
& \qquad \forall x_4 \in \mathcal{B}(x_3, x_5) \\
& \qquad \forall (x_1, x_2, x_5) \in \mathcal{Y}_1 \times \mathcal{P} \times \mathcal{P} \\
& w_N(x_1, x_2, x_3) \geq 0, \\
& \qquad \forall (x_1, x_2, x_3) \in \mathcal{Y}_1 \times \mathcal{P} \times \mathcal{Y}_1 \\
& w_N(x_1, x_2, x_3) \geq v_\zeta(x_1, x_2, x_3) + 1, \\
& \qquad \forall \zeta = 1, 2, \cdots, N \\
& \qquad \forall (x_1, x_2, x_3) \in \mathcal{Y}_1 \times \mathcal{P} \times \mathcal{Y}_1
\end{aligned}$$

where the sets $\mathcal{Y}_1$ and $\mathcal{P}$ are given, and the infimum is taken over an $(N+1)$-tuple of continuous functions $(w_N, v_1, \cdots, v_N) \in \left(C^1(\mathcal{Y}_1 \times \mathcal{P} \times \mathcal{Y}_1; \mathbb{R})\right)^{N+1}$. Note that only the SBM's dynamics appear in this program via $\mathcal{B}(\cdot,\cdot)$. Next, one can prove that the FRS is contained in the 1-superlevel set of all feasible w's in (FRSopt) using an argument similar to [15, Theorem 4]

Lemma 5. *Let* $(w_N, v_1, \cdots, v_N)$ *be feasible functions to (FRSopt), then for all* $(y_1(i), P(i)) \in \mathcal{Y}_1 \times \mathcal{P}$

$$\mathcal{W}_N(y_1(i), P(i)) \subseteq \{x_3 \in \mathcal{Y}_1 \mid w_N(y_1(i), P(i), x_3) \geq 1\}. \tag{20}$$

A feasible polynomial solution to (FRSopt) can be computed offline by applying SOS programming [37], [38].

B. N-step Successful Walking and MPC

To ensure safe walking through N-steps beginning at step i, we require several set inclusions to be satisfied during online optimization based on Theorem 2. First, we require that $\mathcal{W}_N(y_1(i), P(i)) \subseteq [0, \infty)$, which sufficiently guarantee $y_1(i) \geq 0$ for each $i \leq N$. Since we cannot compute $\mathcal{W}_N(y_1(i), P(i))$ exactly, from Lemma 5 we instead can require that the 1-superlevel set of w_N is a subset of $[0, \infty)$.

With the help of the FRS, N-step successful walking conditions on (y_2, y_3, y_4) can be guaranteed in a fashion similar to (15), (16), (17) if

$$f_{\hat{y}_2}(P(i+M)) + \overline{B}_2(P(i+M-1), y_1(i+M), P(i+M)) \leq \pi, \tag{21}$$

$$f_{\hat{y}_3}(y_1(i+M), P(i+M)) + \underline{B}_3(y_1(i+M), P(i+M)) > \pi/2, \tag{22}$$

$$f_{\hat{y}_4}(y_1(i+M), P(i+M)) + \overline{B}_4(y_1(i+M), P(i+M)) < 3\pi/2. \tag{23}$$

hold for all $y_1(i+M) \in \mathcal{W}_M(y_1(i), P(i))$, $1 \leq M \leq N$. Applying (19), one can instead enforce (21), (22), (23) for all $y_1(i+M) \in \mathcal{W}_N(y_1(i), P(i))$ to avoid computing $\mathcal{W}_M(y_1(i), P(i))$ for each $1 \le M < N$. We then use a MPC framework to select gait parameter for RABBIT by solving the nonlinear program:

$$\min_{\substack{P(i)\\ \vdots \\ P(i+N-1)}} \quad r(y(i), P(i), P(i+1), \cdots, P(i+N-1)) \tag{OL}$$
$$\begin{aligned}
\text{s.t.} \quad & P(i), P(i+1), \cdots, P(i+N-1) \in \mathcal{P} \\
& f_{\hat{y}2}(P(i)) + \overline{B}_2(P(i-1), y_1(i), P(i)) \le \pi, \\
& f_{\hat{y}3}(y_1(i), P(i)) + \underline{B}_3(y_1(i), P(i)) > \pi/2, \\
& f_{\hat{y}4}(y_1(i), P(i)) + \overline{B}_4(y_1(i), P(i)) < 3\pi/2, \\
& \mathcal{W}_N(y_1(i), P(i)) \subseteq [0, \infty) \\
& f_{\hat{y}_2}(P(i+M)) + \overline{B}_2(P(i+M-1), y_1(i+M), P(i+M)) \le \pi, \\
& \qquad \text{if } y_1(i+M) \in \mathcal{W}_N(y_1(i), P(i)), \ 1 \le M \le N-1 \\
& f_{\hat{y}_3}(y_1(i+M), P(i+M)) + \underline{B}_3(y_1(i+M), P(i+M)) > \pi/2, \\
& \qquad \text{if } y_1(i+M) \in \mathcal{W}_N(y_1(i), P(i)), \ 1 \le M \le N-1 \\
& f_{\hat{y}_4}(y_1(i+M), P(i+M)) + \overline{B}_4(y_1(i+M), P(i+M)) < 3\pi/2, \\
& \qquad \text{if } y_1(i+M) \in \mathcal{W}_N(y_1(i), P(i)), \ 1 \le M \le N-1
\end{aligned}$$

where $r \in C^1(\mathcal{Y} \times \mathcal{P}^N; \mathbb{R})$ is any user specified cost function. Note that the last four constraints in (OL) are set inclusion constraints. These can be difficult to implement these directly. However, one can conservatively represent these set inclusion constraints by the 0-super level set of a set of polynomials by using the generalized S-procedure [39, Section 2.6.3].

Notice that (OL) is solved at the i-th mid-stance and only the optimal $P(i)$ is applied to the RABBIT and the problem is then solved again for the $(i+1)$-st step. The constraints of (OL) lead to the following theorem:

Theorem 6. *Suppose that RABBIT is at the i-th mid-stance, then tracking the gait parameters associated with any feasible solution to* (*OL*) *ensures that RABBIT can walk successfully for the next N-steps.*

V. Results

To illustrate that the proposed method is able to guarantee safe walking performance online, we evaluate the performance of our method when it is tasked with tracking a randomly generated speed sequence. In each trial, the RABBIT model is required to track a randomly generated speed sequence that holds still for the first 4 steps and changes to a different value starting from the 5th step, i.e. a step function. The speed sequence is restricted to be in a range $[0.2, 2]$. We repeat this experiment on 300 randomly generated speed sequences. The space of control parameter is restricted to be $\mathcal{P} = [0.25, 2] \times [0.15, 0.7]$ on which the gait library is generated. The RABBIT model is initialized with the gait whose speed is closest to the initial value of the desired speed sequence in each trial. The control parameter can only be updated at the mid-stance of each step. Our MATLAB implementation of the experiments can be found at `https://github.com/pczhao/TA_GaitDesign.git`.

In the proposed method, the cost function of (FRSopt) is set to be the weighted Euclidean norm of the difference

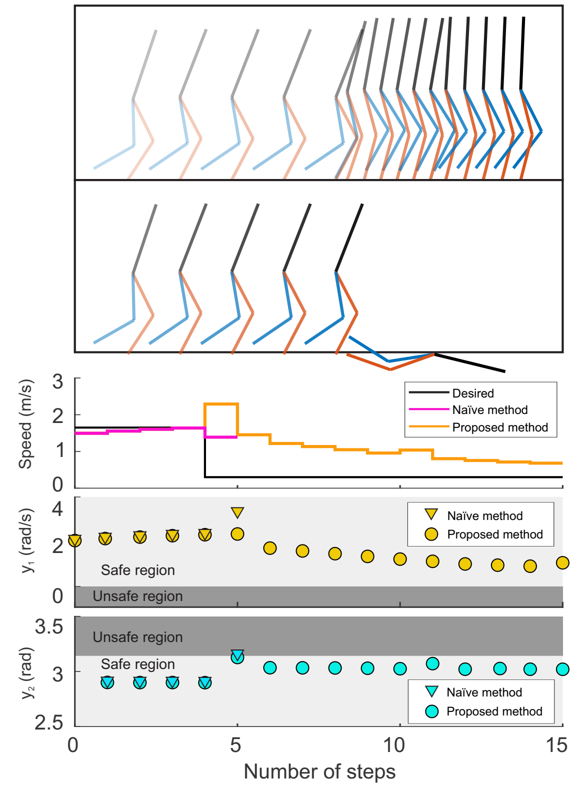

Fig. 3: An illustration of the performance of the method proposed in this paper (top) and a naïve method (second from top). Note that the rapid change in the desired speed (third from top) results in a gait that cannot be tracked by just considering a SBM model without successful walking constraints. By ensuring that the outputs satisfy the inequality constraints proposed in Theorem 2 (bottom two sub-figures), the proposed method is able to safely track the synthesized gaits. Note the naïve method violates the y_2 constraint proposed in Theorem 2 on Step 5.

between the predicted speeds and the desired speeds within the next 3 steps. Note $N = 3$. We compute an outer approximation to the (FRSopt) using the commercial solver MOSEK. We compare our method with a naïve method and a direct method using the same speed tracking sequences. The naïve method uses the SBM model to update gaits in an MPC framework without enforcing walking successful conditions. The direct method uses the full-order dynamics of the RABBIT model to design a controller by solving an optimal control problem [28].

Fig. 3 illustrates the performance of the naïve method and the method proposed in this paper on one of the 300 trials. Note that the gait generated by the naïve method is unable to be followed by the full-order RABBIT model. On the other hand, the method proposed in this paper is able to generate a gait that can satisfy the safety requirements described in Theorem 2. This results in a controller that can track the synthesized gait without falling over. Across all 300 trials the computation time of the naïve method is 0.01 seconds, the direct method is 93.12 seconds, and the proposed method is 0.11 seconds. Moreover, the RABBIT model falls 2% of the time with the naïve method, but never falls with the proposed or the direct methods.

References

[1] M. Vukobratović and J Stepanenko, "On the stability of anthropomorphic systems," *Mathematical biosciences*, vol. 15, no. 1-2, pp. 1–37, 1972.

[2] A. D. Kuo, "Choosing your steps carefully," *IEEE Robotics & Automation Magazine*, vol. 14, no. 2, pp. 18–29, 2007.

[3] E. R. Westervelt, C. Chevallereau, J. H. Choi, B. Morris, and J. W. Grizzle, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007.

[4] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[5] S. C. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with application to bipedal robotic walking," in *2015 American Control Conference (ACC)*, 2015, pp. 4542–4548.

[6] Q. Nguyen and K. Sreenath, "Optimal robust control for bipedal robots through control lyapunov function based quadratic programs.," in *Robotics: Science and Systems*, 2015.

[7] ——, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference (ACC), 2016*, IEEE, 2016, pp. 322–328.

[8] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, "3d dynamic walking on stepping stones with control barrier functions," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 827–834.

[9] M. S. Motahar, S. Veer, and I. Poulakakis, "Composing limit cycles for motion planning of 3d bipedal walkers," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 6368–6374.

[10] S. Veer and I. Poulakakis, "Safe adaptive switching among dynamical movement primitives: Application to 3d limit-cycle walkers," *arXiv preprint arXiv:1810.00527*, 2018.

[11] A. D. Ames, P. Tabuada, A. Jones, W.-L. Ma, M. Rungger, B. Schürmann, S. Kolathaya, and J. W. Grizzle, "First steps toward formal controller synthesis for bipedal robots with experimental implementation," *Nonlinear Analysis: Hybrid Systems*, vol. 25, pp. 155–173, 2017.

[12] N. Smit-Anseeuw, C. D. Remy, and R. Vasudevan, "Walking with confidence: Safety regulation for full order biped models," *arXiv preprint arXiv:1903.08327*, 2019.

[13] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," PhD thesis, California Institute of Technology, 2000.

[14] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *International Workshop on Hybrid Systems: Computation and Control*, Springer, 2004, pp. 477–492.

[15] V. Shia, R. Vasudevan, R. Bajcsy, and R. Tedrake, "Convex computation of the reachable set for controlled polynomial hybrid systems," in *53rd IEEE Conference on Decision and Control*, 2014, pp. 1499–1506.

[16] T. Koolen, M. Posa, and R. Tedrake, "Balance control using center of mass height variation: Limitations imposed by unilateral contact," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, IEEE, 2016, pp. 8–15.

[17] P.-B. Wieber, "On the stability of walking systems," in *Proceedings of the international workshop on humanoid and human friendly robotics*, 2002.

[18] M. Posa, T. Koolen, and R. Tedrake, "Balancing and step recovery capturability via sums-of-squares optimization," in *2017 Robotics: Science and Systems Conference*, 2017.

[19] P. Zhao, S. Mohan, and R. Vasudevan, "Optimal control for nonlinear hybrid systems via convex relaxations," *arXiv preprint arXiv:1702.04310*, 2017.

[20] J. Z. Tang, A. M. Boudali, and I. R. Manchester, "Invariant funnels for underactuated dynamic walking robots: New phase variable and experimental validation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3497–3504.

[21] R. J. Full and D. E. Koditschek, "Templates and anchors: Neuromechanical hypotheses of legged locomotion on land," *Journal of experimental biology*, vol. 202, no. 23, pp. 3325–3332, 1999.

[22] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[23] 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.

[24] S. Kousik, S. Vaskov, M. Johnson-Roberson, and R. Vasudevan, "Safe trajectory synthesis for autonomous driving in unforeseen environments," in *ASME 2017 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, 2017, V001T44A005–V001T44A005.

[25] S. Kousik, S. Vaskov, F. Bu, M. Johnson-Roberson, and R. Vasudevan, "Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots," *arXiv preprint arXiv:1809.06746*, 2018.

[26] S. Kousik, P. Holmes, and R. Vasudevan, "Safe, aggressive quadrotor flight via reachability-based trajectory design," *arXiv preprint arXiv:1904.05728*, 2019.

[27] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. C. de Wit, and J. Grizzle, "Rabbit: A testbed for advanced control theory," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, 2003.

[28] A. Hereid and A. D. Ames, "Frost: Fast robot optimization and simulation toolkit," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE/RSJ, Vancouver, BC, Canada, Sep. 2017.

[29] J. Lygeros, S. Sastry, and C. Tomlin, "Hybrid systems: Foundations, advanced topics and applications," *under copyright to be published by Springer Verlag*, 2012.

[30] D. L. Wight, E. G. Kubica, and D. W. Wang, "Introduction of the foot placement estimator: A dynamic measure of balance for bipedal robotics," *Journal of computational and nonlinear dynamics*, vol. 3, no. 1, p. 011 009, 2008.

[31] S. Smith, H. Yin, and M. Arcak, "Continuous abstraction of nonlinear systems using sum-of-squares programming," in *58th Conference on Decision and Control*, 2019.

[32] A. A. Ahmadi and A. Majumdar, "Dsos and sdsos optimization: Lp and socp-based alternatives to sum of squares optimization," in *2014 48th annual conference on information sciences and systems (CISS)*, IEEE, 2014, pp. 1–5.

[33] X. Kuang, B. Ghaddar, J. Naoum-Sawaya, and L. F. Zuluaga, "Alternative sdp and socp approximations for polynomial optimization," *EURO Journal on Computational Optimization*, vol. 7, no. 2, pp. 153–175, 2019.

[34] D. Papp and S. Yildiz, "Sum-of-squares optimization without semidefinite programming," *SIAM Journal on Optimization*, vol. 29, no. 1, pp. 822–851, 2019.

[35] M. Tacchi, C. Cardozo, D. Henrion, and J. Lasserre, "Approximating regions of attraction of a sparse polynomial differential system," *arXiv preprint arXiv:1911.09500*, 2019.

[36] D. Henrion and M. Korda, "Convex computation of the region of attraction of polynomial control systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 297–312, 2013.

[37] P. Zhao, S. Mohan, and R. Vasudevan, "Control synthesis for nonlinear optimal control via convex relaxations," in *2017 American Control Conference (ACC)*, IEEE, 2017, pp. 2654–2661.

[38] S. Mohan, J. Liu, and R. Vasudevan, "Synthesizing the optimal luenberger-type observer for nonlinear systems," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, 2017, pp. 3658–3663.

[39] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Siam, 1994, vol. 15.