


---

# Combining Trajectory Optimization, Supervised Machine Learning, and Model Structure for Mitigating the Curse of Dimensionality in the Control of Bipedal Robots

Journal Title  
XX(X):1–36  
©The Author(s) 2017  
Reprints and permission:  
sagepub.co.uk/journalsPermissions.nav  
DOI: 10.1177/ToBeAssigned  
www.sagepub.com/  


Xingye Da<sup>1</sup> and Jessy Grizzle<sup>2</sup>

## Abstract

To overcome the obstructions imposed by high-dimensional bipedal models, we embed a stable walking motion in an attractive low-dimensional surface of the system's state space. The process begins with trajectory optimization to design an open-loop periodic walking motion of the high-dimensional model and then adding to this solution, a carefully selected set of additional open-loop trajectories of the model that steer toward the nominal motion. A drawback of trajectories is that they provide little information on how to respond to a disturbance. To address this shortcoming, Supervised Machine Learning is used to extract a low-dimensional state-variable realization of the open-loop trajectories. The periodic orbit is now an attractor of the low-dimensional state-variable model but is not attractive in the full-order system. We then use the special structure of mechanical models associated with bipedal robots to embed the low-dimensional model in the original model in such a manner that the desired walking motions are locally exponentially stable. The design procedure is first developed for ordinary differential equations and illustrated on a simple model. The methods are subsequently extended to a class of hybrid models and then realized experimentally on an Atrias-series 3D bipedal robot.

## Keywords

Bipedal robots, machine learning, trajectory optimization, zero dynamics.

## 1 Introduction and Problem Statement

We seek to design controllers for high degree-of-freedom (DoF) bipedal robots with several degrees of underactuation (DoU), or, if the robot is “fully actuated”, we wish to take into account the limited ability of ankle torques to affect the overall evolution of the robot. The paper will focus on the tasks of walking stably forward, backward, or in place, and transitioning among such motions. We want the gaits to be dynamic in the sense that they can use the full capability of the robot regarding speed, terrain type, and other forms of agility. Moreover, of course, we need to embed the controller on the robot for real-time implementation. Our unique approach to this well-studied problem is illustrated in Figure 1.

### 1.1 Proposed Approach to Controller Design

We begin with a full-order dynamical model of the robot, expressing its many degrees of freedom and actuation

capability, and a simplified representation of its contact with the environment so that the overall model can be expressed as a system with impulse effects, a special class of hybrid models. To overcome the obstructions imposed by high-dimensional models, we first seek to embed a stable walking motion in a low-dimensional surface of the system's state space. Subsequently, we seek to stabilize the gait in the full-order model by rendering the surface attractive.

The most common approach in the literature to getting around the high dimension of the model is to represent the

---

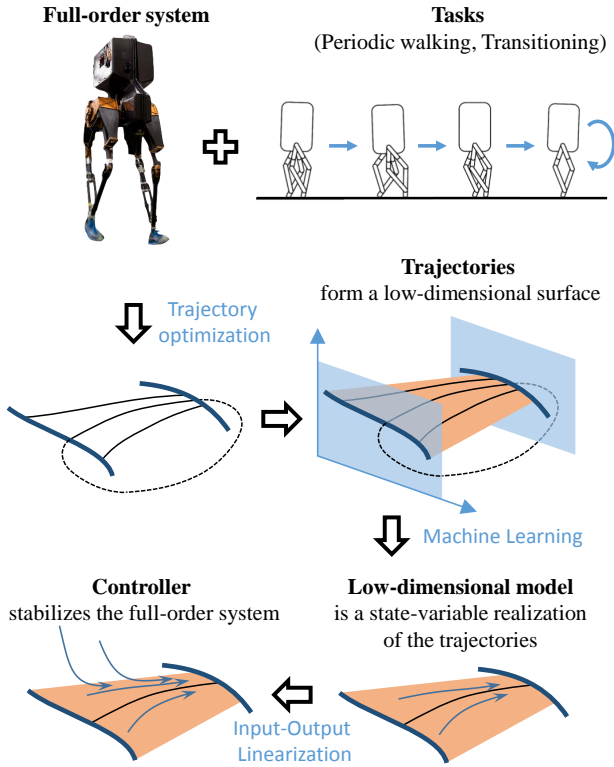
<sup>1</sup> Mechanical Engineering Department, University of Michigan, Ann Arbor, MI, US

<sup>2</sup> Electrical and Computer Engineering and Robotics Institute, University of Michigan, Ann Arbor, MI, US

#### Corresponding author:

Xingye Da, Mechanical Engineering Department, University of Michigan, Ann Arbor, MI 48109, US

Email: xda@umich.edu



**Figure 1.** Our overall approach. The full-order model, the desired objectives, and physical constraints are combined into a trajectory optimization problem for designing a periodic gait. Using model structure or physical insight, a low-dimensional surface of initial conditions is selected for trajectory building, with the trajectories designed to approach the periodic orbit. If the trajectories form a low-dimensional surface, Supervised Machine Learning is used to extract a vector field from the data that realizes the trajectories. System structure is then used again to render the low-dimensional model (surface and vector field) invariant and attractive.

walking task through the dynamics of a low-dimensional inverted pendulum (e.g., LIP, SLIP or others in Figure 2), which equipped with a foot-placement strategy for stability [Kajita et al. \(1992\)](#); [Raibert \(1986b,a\)](#); [Pratt and Tedrake \(2006\)](#). The robot is then controlled in such a way that its center of mass approximately follows the target dynamics. The many challenges associated with this more common approach include: achieving stable solutions in the full model; exploiting the full capability of the machine, especially in light of physical constraints of the hardware or environment; deciding how to associate the states of the low-dimensional pendulum with the full-order system; and finally, even deciding upon the appropriate pendulum model for a given task is not evident: what is the correct model for turning while stepping off a platform?

For these reasons, we do not rely on a pre-specified pendulum model to encode the walking motion. In this introduction, we ask that the reader allow us to sketch the

main ideas of our approach without worrying too much about technicalities. Later developments will be more formal.

For the sake of simplicity, let’s pretend the model of the robot can be captured by an ordinary differential equation,

$$\dot{x} = F(x, u), \quad (1)$$

with state variables  $x \in \mathcal{X}$  and control inputs  $u \in \mathcal{U}$ . The design process begins with construction of a periodic solution meeting relevant constraints. Denote the period by  $T_p > 0$  and the initial condition by  $\xi^*$ . The next step is to make an initial selection of a low-dimensional set, let’s call it  $Z_0 \subset \mathcal{X}$ , such that  $\xi^* \in Z_0$ . Now begins the real work; we seek to design open-loop trajectories of the full-order model

$$(x_\xi(t), u_\xi(t)), \quad 0 \leq t \leq T_p, \quad \xi \in Z_0 \quad (2)$$

that over the interval  $[0, T_p]$ , “approach” the periodic solution. Specifically, for some  $0 \leq c < 1$ , and for each  $\xi \in Z_0$ , we have

$$\varphi_\xi(T_p) \in Z_0 \quad \text{and} \quad \|\varphi_\xi(T_p) - \xi^*\| \leq c\|\xi - \xi^*\|. \quad (3)$$

After designing the trajectories, we seek to construct a low-dimensional subsystem that realizes them, namely,

$$\begin{aligned} \dot{z}(t) &= G(t, z) \\ x(t) &= H(t, z), \end{aligned} \quad (4)$$

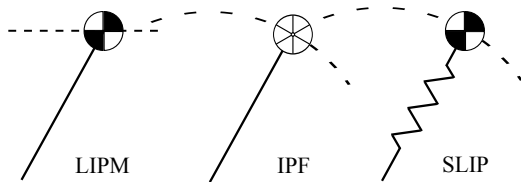
with  $z \in Z \subset \mathcal{X}$ , such that (a) for  $\xi \in Z_0$ ,

$$z(0) = \xi \Rightarrow x_\xi(t) = H(t, z(t)),$$

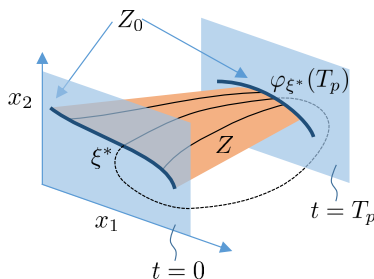
and (b) the periodic motion is a “locally exponentially stable output” of the model. If this can be done, we would argue that (4) is a more desirable target model than a typical pendulum because the target has been constructed directly from the full-order model and its “specification”, that is, the constraints imposed when designing the trajectories.

To turn this into a viable feedback design process for the original system, we have to address the following issues:

- (i) How to compute the low-dimensional model (4) for realistic bipedal robots and the surface<sup>1</sup>  $Z \supset Z_0$  on which it is defined?
- (ii) As with any low-dimensional target model, how to embed it in the full-order model with stability? In other words, how to design a feedback for the original system (1) that does two things: (a) creates an invariant surface  $Z$  in its state space with restriction dynamics given by (4), thereby encoding the desired stable



**Figure 2.** Adapted from (Grizzle et al. 2014, Figure 5). Three low-dimensional models that are frequently used as approximate representations of walking robots. From left to right: the Linear Inverted Pendulum (LIP) lumps the mass of the robot at a point moving at a constant height and assumes massless legs; the Inverted Pendulum with Flywheel (IPF) relaxes the assumption on constant height and adds a flywheel to account for internal angular momentum; and the Spring-Loaded Inverted Pendulum (SLIP) adds a spring to model a robot’s legs as a massless pogo stick. There is no obvious way to embed these low-dimensional models into the full model of a robot.



**Figure 3.** Based on model structure, the system’s state is decomposed into  $x = (x_1, x_2)$ , where the dimension of  $x_1$  is much smaller than the dimension of  $x_2$ . The surface  $Z_0$ , shown here as a line, is the set of initial conditions used to build a set of trajectories that will fill out the surface  $Z$ . By construction, this model, if it exists, will be easy to embed in the robot’s full state space.

walking motion; and (b) solutions of the closed-loop system starting near the surface asymptotically converge to the surface, thereby realizing the walking motion in a stable manner in the overall system.

The remainder of the paper is dedicated to addressing these challenges for a class of hybrid models and tasks of interest to bipedal locomotion. Section 3 develops the basic ideas in the simpler setting of ordinary differential equations. The results are of independent interest for tasks such as rising from a sitting position or standing in place. Trajectory optimization is used to generate the low-dimensional set of open-loop trajectories (2) that includes a metric for attractivity to a periodic solution, a family of periodic solutions, or transitions among such solutions. Model structure and Supervised Machine Learning are proposed as a means to extract functions from the open-loop trajectories to build the low-dimensional model (4). Finally, the appeal is made once again to model structure to embed the low-dimensional model in the full-order model while guaranteeing local exponential attractivity.

Section 4 illustrates the design process on the well-known inverted pendulum on a cart. This will allow the reader to explore the method on a simple model. Section 5 develops the results for hybrid models, preparing the ground for the simulations and experiments reported in Section 6 for the bipedal robot of Figure 1. The discussion and conclusions are given in Section 8 and all proofs are given in Appendix C.

## 2 Related Literature

In the following, related work in the legged robotics literature is summarized and contrasted to the work in the present paper. Appendix D provides a more technical comparison of several nonlinear control methods, namely, backstepping, zero dynamics, and immersion and invariance.

### 2.1 Online Optimization

One of the earliest uses of online optimization for bipedal walking was done on a *simulation model* of the planar robot RABBIT Azevedo et al. (2002, 2004). The controller could only be done in simulation because its computation time was approximately forty times slower than the duration of a step. More recently, Model Predictive Control (MPC) was applied in the *DARPA Virtual Robotics Challenge* Erez et al. (2013). In that work, a real “simulation-time” implementation on a full-order model of Atlas was achieved through the use of a novel physics engine and a relaxed contact map. MPC was applied on Kuindersma et al. (2016) to a simplified model of Atlas that captured the kinematics and centroidal dynamics; this resulted in walking at 0.4 m/s. On a planar biped, higher walking speeds from 0.43 m/s to 0.97 m/s were achieved in Hereid et al. (2016b) using online Partial Hybrid Zero Dynamics (PHZD) gait generation. Average computational time was 0.5 s. To stabilize a robot that has to make and break multiple contacts with the environment, a piecewise affine approximation of a nonlinear hybrid system was controlled using a mix of online approach and explicit MPC Marcucci et al. (2017), or a Piecewise-Affine Quadratic Regulator Han and Tedrake (2017). These optimizations are more effective than the purely online MPC, but the model so far has not taken the centroidal angular momentum into account.

### 2.2 Pendulum Models

The pendulum models illustrated in Figure 2 are ubiquitous means in the bipedal robotics literature to reduce the online computational burden. The LIP model is especially prevalent for the design of flat-footed walking gaits based on the Zero Moment Point criterion Kajita and Tani (1991); Yamaguchi et al. (1999); Ogura et al. (2006); Sakagami et al. (2002);

Kaneko et al. (2011); Park et al. (2005); Pfeiffer et al. (2002). The optimization or closed-form compute the CoM trajectories and swing foot positions on the reduced-order models. A low-level controller and inverse kinematics then realize these on the full-order model or robot. Recent experimental uses of this approach can be found in Pratt et al. (2012); Krause et al. (2012); Faraji et al. (2014); Rezazadeh et al. (2015).

The bottom line, however, is that when a pendulum model is pre-specified as a template Full and Koditschek (1999) the full order model needs to compromise its achievable motions to follow the template. Moreover, for each different task of the robot, such as walking or running, the designer is faced with the selection of the “best” target model. In our approach, a low-dimensional model is generated from the full-order model and the task. It is dynamically feasible and uses the full capability of the robot to accomplish the task.

### 2.3 Pre-computed Gaits

A means to get around the limitation of online computation is to pre-compute a set of controllers and design a control policy to “stitch” them together. A policy that switches the task (target walking speed, running vs. walking, stairs vs. flat ground) is employed in Sreenath et al. (2013); Martin et al. (2014); Powell et al. (2013). Finite-state machines or motion primitives are used in Park et al. (2013); Saglam and Byl (2015); Manchester and Umenberger (2014) for rough terrain and in Apostolopoulos et al. (2015) for reducing settling time. Interpolation among gaits has been used to create a continuous family of gaits in Embry et al. (2016); Da et al. (2016); Nguyen et al. (2017, 2016). Transient trajectories that approach the nominal periodic orbit were added in Liu et al. (2013); Da et al. (2017) to enlarge the basin of attraction. The current paper provides a formal mathematical framework for the work in Da et al. (2017) and increases its applicability.

### 2.4 Hybrid Zero Dynamics

The work in the present paper is related to the method of virtual constraints and hybrid zero dynamics (HZD) proposed in Westervelt et al. (2002). In that method, a monotonically increasing function of a robot’s generalized coordinates  $q$  is first identified, often denoted by  $\theta(q)$ , and then a family of virtual constraints of the form

$$y = h_0(q) - h_d(\theta(q), \alpha) \quad (5)$$

are posted, where  $\dim(y) = \dim(u)$ , the number of inputs,  $h_0(q)$  represents quantities to be regulated, such as knee

angles and hip angles, and  $h_d(\theta(q), \alpha)$  is a vector of splines representing the two to be determined desired evolution of  $h_0(q)$ . A parameter optimization problem is posted to select the values of  $\alpha$  (if they exist) so that  $y \equiv 0$  along a periodic solution of the model, torque bounds are met, as are ground contact forces. If the outputs  $y$  have vector relative degree two (Westervelt et al. 2007, pp. 119), the robot model with outputs (5) is input-output linearizable, and hence a feedback controller can be designed that drives the virtual constraints asymptotically to zero. If the surface defined by the outputs being zero is invariant in the hybrid model of the robot, then (Westervelt et al. 2007, Ch. 5) provides strong stability theorems for the closed-loop system.

While this theory has been successfully implemented on many robots Ames et al. (2012); Ames (2014); Buss et al. (2014); Chevallereau et al. (2003); Hereid et al. (2014); Martin et al. (2014); Sreenath et al. (2011, 2012); Reher et al. (2016b), lower-limb prostheses Gregg et al. (2014); Aghasadeghi et al. (2013); Zhao et al. (2015) and exoskeleton Agrawal et al. (2017), it has important limitations that the current paper overcomes:

- In basic HZD, only one optimization is done, namely the determination of the periodic orbit. Hence, only that solution is guaranteed to be feasible. Here, we build a low-dimensional surface of feasible solutions.
- The stability theorems in Westervelt et al. (2007) only apply to robots with one degree of underactuation. Here, multiple degrees of underactuation can be handled. Moreover, unlike the method proposed in Buss et al. (2016); Hamed et al. (2016) which works with the linearization of the Poincaré map to achieve local exponential stability for robots with more than one degree of underactuation, here, at least on the low-dimensional surface, large perturbations away from the periodic orbit can be included in the controller design, providing better robustness.
- For robots with one degree of underactuation, the stability mechanism in (Westervelt et al. 2007, Ch. 5.4) is through energy loss at impacts, similar to the stability proofs for passive robots walking down a slope. Here, more general stability mechanisms in bipedal robots Pratt and Tedrake (2006), such as posture adjustments through foot placement and knee bend, automatically arise.
- Only gaits for which a monotonic variable can be identified can be treated with the current HZD method. Here, a much richer set of locomotion primitives can be realized, such as stepping in place or transitioning from walking forward to walking backward. As in

Wang and Chevallereau (2011), Reher et al. (2016a), Da et al. (2016), the feedback is allowed to be time-dependent, enriching the set of possible closed-loop solutions.

- Similar in spirit to Griffin and Grizzle (2016), which added conjugate momenta of the underactuated coordinates into the virtual constraints, this paper can include velocity as well as positions in the low-dimensional surface that eventually defines a generalized hybrid zero dynamics manifold; see also Hartley et al. (2017).

### 3 Presentation of Main Ideas

For the class of robot problems of interest to us, optimal gaits can now be computed in minutes Jones (2014); Hereid et al. (2016a), but not in tens of milliseconds, which is what would be required for online use. In the simpler setting of a non-hybrid system, this section develops our main ideas for mitigating the curse of dimensionality in optimization-based controller design. Section 3.2 provides the first example of conditions for a family of open-loop trajectories of a model from which a realization can be extracted and its equilibrium will be locally exponentially stable; see also Coron (1994); Schürmann and Althoff (2017). The process of building the realization from the trajectories is based on regression, namely Supervised Machine Learning. The size of the model that can be treated with these initial results is limited by both the number of optimizations it takes to create the family of open-loop trajectories and the number of features that can be included in the Supervised Machine Learning. Section 3.3 extends the design process to building reduced-order target model for a high-dimensional model. Importantly, the design is based on a far smaller set of open-loop trajectories. Sections 3.4 and 3.5 then provide conditions for embedding the target model in the full-order model such that the origin of the full-order model is locally exponentially stable. The proofs of the results developed in the section are given in Appendix C. The relationship to other controller design methods is addressed in Appendix D.

In presenting the main ideas, we will deliberately organize them as a design philosophy. We choose to let the user rely on his or her wits to meet our conditions, rather than muddying the waters with a set of highly technical sufficient conditions that no one would ever check. We know as well as the readers know that optimization problems are very tricky: it is easy to paint oneself into a corner that only yields non-smooth solutions. On the other hand, many problems, such as the examples worked out in the paper, seem to have very nicely

behaved solutions. We are confident that we did not cherry-pick the only nice problems and that the readers will find a host of further interesting examples.

#### 3.1 Model Assumptions

To keep the connection to stabilizing periodic gaits in bipeds, we consider a periodically time-varying nonlinear system with equilibrium point at the origin,

$$\dot{x} = f(t, x, u). \quad (6)$$

The simple coordinate transformation required for shifting a periodic solution of a nonlinear model to the origin is provided in (Khalil 2002, pp. 147). An equilibrium point is treated as a special case of a periodic orbit where the period can be any number  $T_p > 0$ ; in particular, when discussing periodicity,  $T_p$  is not required to be a fundamental period.

The ODE (6) is assumed to satisfy the following conditions.

**A-1**  $f : [0, \infty) \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is locally Lipschitz continuous in  $x$  and  $u$ , piecewise continuous in  $t$ , and there exists  $T_p > 0$  such that  $\forall (t, x, u) \in [0, \infty) \times \mathbb{R}^n \times \mathbb{R}^m$ ,  $f(t + T_p, x, u) = f(t, x, u)$ .

**A-2**  $f(t, 0, 0) = 0$  for all  $t \geq 0$ .

**A-3** The user has selected an open ball about the origin,  $B \subset \mathbb{R}^n$ , a positive-definite, locally Lipschitz-continuous function  $V : B \rightarrow \mathbb{R}$ , and constants  $0 < \alpha_1 \leq \alpha_2$  such that,  $\forall x \in B$

$$\alpha_1 x^\top x \leq V(x) \leq \alpha_2 x^\top x.$$

■

As alluded to above, the reader is encouraged to view the assumptions made throughout this section as requirements to impose on an open-loop trajectory optimizer. We have found them straightforward to meet when using the optimizers of Jones (2014); Hereid et al. (2016a). In many cases, the positive-definite function indicated in A-3 comes “for free” from the optimization problem used to compute trajectories; this is standard in model predictive control Mayne et al. (2000). Because the Lyapunov condition in A-4 below can also be included as a constraint in the trajectory generation process, the user has much freedom in its selection, even something as simple as the 2-norm squared could be used.



### 3.2 Extracting a Feedback from Open-loop Trajectories

Two feedback controllers can be constructed from the following solutions of the model (6). The obvious relation to MPC is discussed in the remarks following Prop. 1.

**A-4** There is a constant  $0 \leq c < 1$ , such that, for each initial condition  $\xi \in B$ , there exists a continuous input  $u_\xi : [0, T_p] \rightarrow \mathbb{R}^m$  and a corresponding solution of the ODE,  $\varphi_\xi : [0, T_p] \rightarrow \mathbb{R}^n$  satisfying  $\varphi_\xi(T_p) \in B$ , and

$$V(\varphi_\xi(T_p)) \leq cV(\xi); \quad (7)$$

moreover, for  $\xi = 0$ ,  $u_\xi(t) \equiv 0$ . For clarity, solutions are taken in the sense of equation (C.2) of (Khalil 2002, pp. 657), namely

$$\varphi_\xi(t) = \xi + \int_0^t f(\tau, \varphi_\xi(\tau), u_\xi(\tau)) d\tau. \quad (8)$$

■

A  $T_p$ -periodic Continuous-Hold (CH) feedback is defined by periodic extension of  $u_\xi(t)$ , namely,

$$u^{ch}(t, \xi) = u_\xi(\hat{t}), \quad \hat{t} = t \bmod T_p. \quad (9)$$

Jumps are allowed at multiples of the period, with continuity taken from the right. Due to the reset or hold-nature of the above feedback, the stability of solutions of (6) in closed-loop with (9) should be studied as a sampled-data system, that is, the solutions should be evaluated at times  $t_k = kT_p$ . We will not analyze its stability, however, because it will clearly perform poorly in the face of perturbations occurring between samples, where the system is open loop.

We proceed directly instead to a feedback controller that allows continual updates in the state variables, and yet, under certain conditions, can be built from the open-loop trajectories given in Assumption A-4. To understand the feedback controller, a *thought experiment* is helpful: Suppose at time  $t_0 = 0$  the system's initial state value is  $\xi^0$ , and the continuous-hold feedback (9) is being applied. Then the system is evolving along the trajectory  $\varphi_{\xi^0}(t)$ . Suppose subsequently at time  $0 < t_d < T_p$ , an "impulsive disturbance" affects the system, displacing the system's state to a value  $x(t_d) \neq \varphi_{\xi^0}(t_d)$ . What input might be applied, given the information in A-4? If there exists a  $\xi^d \in B$  such that  $x(t_d) = \varphi_{\xi^d}(t_d)$ , then applying the input  $u_{\xi^d}(t)$  for  $t_d \leq t < T_p$  will move the system toward the equilibrium in the sense that  $V(\varphi_{\xi^d}(T_p)) \leq cV(\xi^d)$ . The next result builds

Features		Labels
$t_j$	$x^{j,i} = \varphi_{\xi^i}(t_j)$	$\mu^{j,i} = u_{\xi^i}(t_j)$
$t_0 = 0$	$x^{0,1}$	$\mu^{0,1}$
$t_0 = 0$	$x^{0,2}$	$\mu^{0,2}$
$\vdots$	$\vdots$	$\vdots$
$t_0 = 0$	$x^{0,M}$	$\mu^{0,M}$
$t_1$	$x^{1,1}$	$\mu^{1,1}$
$\vdots$	$\vdots$	$\vdots$
$t_1$	$x^{1,M}$	$\mu^{1,M}$
$\vdots$	$\vdots$	$\vdots$
$t_N = T_p$	$x^{N,1}$	$\mu^{N,1}$
$\vdots$	$\vdots$	$\vdots$
$t_N = T_p$	$x^{N,M}$	$\mu^{N,M}$

**Table 1.** Conceptual arrangement of the data in A-4 from which a controller satisfying (10) may be determined by "regression". In practice, not only time must be discretized, but also the initial conditions  $\xi^i \in B$ . This is where the "curse of dimensionality" rears its ugly head. Placing ten points per dimension leads to  $10^n$  optimizations to compute, which quickly becomes impractical.

on this idea; see also Coron (1994); Schürmann and Althoff (2017).

**Proposition 1.** Assume the open-loop system (6) satisfies Assumptions A-1 to A-4. Assume in addition there exists an open set  $B^e \supset B$  and a feedback

$$\mu : [0, \infty) \times B^e \rightarrow \mathbb{R}^m$$

that is piecewise continuous in  $t$ ,  $T_p$ -periodic, locally Lipschitz continuous in  $x$ , and, such that, for  $0 \leq t < T_p$  and  $\xi \in B$ ,

$$\mu(t, \varphi_\xi(t)) = u_\xi(t). \quad (10)$$

Then the origin of the closed-loop system,

$$\dot{x} = f^{cl}(t, x) := f(t, x, \mu(t, x)), \quad (11)$$

is locally exponentially stable, uniformly in  $t$ , and the trajectories in A-4 are solutions of (11). Said another way, (11) is a realization of the trajectories in A-4. ■

Returning to (4) in the Introduction, the key point is that when a function can be found that is compatible with the open-loop trajectories in A-4 in the sense that the "learning condition" (10) is satisfied, then (11) provides an "exponentially stable realization" of the trajectories; this idea will be extended to a lower-dimensional system in the next subsection. Secondly, if at time  $0 \leq t \leq T_p$  there exists  $\xi$  such that  $x(t) = \varphi_\xi(t)$ , the value of the "learned function"

is being set to

$$\mu(t, x(t)) := u_\xi(t).$$

In practice, the trajectories will only be computed for a finite grid of initial conditions  $\xi^j$ ,  $j \in J$ . Hence, (10) is an *interpolation of the data* from the trajectory optimizations. For  $x(t) \in B^e$  for which there does not exist  $\xi$  such that  $x(t) = \varphi_\xi(t)$ , the function in (10) is an *extrapolation of the data*. In Coron (1994), the fitting of a function to trajectory data was done *in principle* in closed form and by hand; here, the fitting is being done with Supervised Machine Learning. Moreover, the learning algorithms available today provide easy tools for checking the quality of a fit and hence for checking how closely a function was found that meets the learning condition (10).

**Remark 1.** *Because the solutions in A-4 will be computed via a trajectory optimization algorithm, it is useful to understand how the assumptions on  $\mu$  relate to requirements on the trajectories.*

- (i) Suppose  $T_h > T_p$  and that for  $\xi \in B$ ,  $u_\xi^o : [0, T_h] \rightarrow \mathbb{R}^m$  minimizes a cost function of the form

$$J(\xi) = \min_u \int_0^{T_h} L(\varphi_\xi(t), u_\xi(t)) dt + N(\varphi_\xi(T_h)) \quad (12)$$

where, as before,  $\varphi_\xi(t)$  is the solution of (6) with initial condition  $\xi$  at  $t_0 = 0$ , and suppose furthermore that  $u_\xi : [0, T_p] \rightarrow \mathbb{R}^m$  is the restriction of  $u_\xi^o$  to  $[0, T_p]$ , that is,

$$u_\xi = u_\xi^o|_{[0, T_p]}.$$

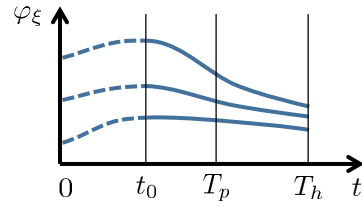
By the principle of optimality, for  $0 \leq t_0 < T_p$ ,

$$u_\xi^o|_{[t_0, T_h]}$$

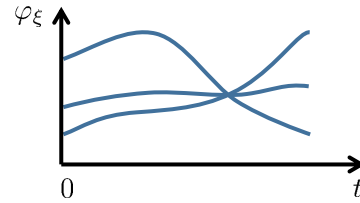
is a minimizer of

$$J(x_0) = \min \int_{t_0}^{T_h} L(\varphi_\xi(t), u_\xi(t)) dt + N(\varphi_\xi(T_h)), \quad (13)$$

where,  $\varphi_\xi(t)$  is the solution of (6) with initial condition  $\varphi_\xi(t_0)$  at  $t_0$ . Hence, the condition (10) can be interpreted as arising from an MPC-style controller with a shrinking horizon,  $[t_0, T_h]$ , for  $0 \leq t_0 \leq T_p$  and fixed final-time  $T_h$ . This control strategy is visualized in Figure 4. The condition (10) comes from the shrinking of the optimization horizon; it will be essential in allowing a judiciously chosen set



**Figure 4.** Principle of Optimality. If the system is initialized at  $\varphi_\xi(t_0)$  and the cost function is modified from (12) to (13), then  $\varphi_\xi : [t_0, T_h] \rightarrow \mathbb{R}^n$  is optimal.



**Figure 5.** This shows the trajectories crossing one another, which means that the mapping  $\Psi_t : B \rightarrow \mathbb{R}^n$  in (14) is not injective at certain moments of time. In this case, a feedback function cannot be extracted from the data.

of open-loop trajectories to be realized with a low-dimensional state-variable model.

- (ii) Supervised Machine Learning will be used to extract the function  $\mu(t, x)$  in Prop. 1 from the trajectories and control inputs given in A-4. The method is sketched in Table 1. An example is given in Sect. 4.
- (iii) The local Lipschitz continuity of  $\mu$  imposes conditions on the solutions given in A-4. Indeed, for each  $t \in [0, T_p]$ , the mapping

$$\Psi_t : B \rightarrow \mathbb{R}^n, \text{ by } \Psi_t(\xi) := \varphi_\xi(t) \quad (14)$$

must be injective. This follows by the Gronwall-Bellman inequality (Khalil 2002, pp. 651); see also (Khalil 2002, Exercise 3.17). Hence, the optimization problem must be set up so as to avoid the existence of trajectories that cross one another, which can easily occur as shown in Figure 5. For example, if the user selected  $T_h = T_p$  and imposed that the origin be attained at  $T_h$ , that is, dead-beat control, then a locally Lipschitz continuous  $\mu$  would not exist.

- (iv) Conditions are known under which the value function (12) meets the Lyapunov conditions in A-3 and A-4. Roughly speaking, they require that the terminal weight  $N$  either be replaced with finite-time convergence to the origin or, the terminal weight be selected as  $\beta N(x(T_h))$ , where  $N(x)$  is positive definite and  $\beta > 0$  is sufficiently large. Hence, in practice, the Lyapunov constraint can be replaced by careful formulation of the trajectory optimization

problem. In our limited experience, it is never an active constraint.

- (v) There is a long history of work in the nonlinear control literature that relates asymptotic controllability to an equilibrium point and the existence of stabilizing feedback controllers. The reader is referred to [Coron \(1994\)](#); [Coron et al. \(1995\)](#); [Clarke et al. \(1997\)](#); [Shim and Teel \(2003\)](#) and references therein. The methods employed are not nearly as constructive as the present paper.

### 3.3 Building a Reduced-Order Target Model

To begin the construction of a reduced-order target model as in (4), we now assume that the system (6) is decomposed in the form

$$\begin{aligned}\dot{x}_1 &= f_1(t, x_1, x_2) \\ \dot{x}_2 &= f_2(t, x_1, x_2, u),\end{aligned}\quad (15)$$

where  $x_1 \in \mathbb{R}^{n_1}$  and  $x_2 \in \mathbb{R}^{n_2}$ . For clarity of exposition, the input is assumed not to appear in  $f_1$ ; the changes required to include inputs in  $f_1$  are given in Sect. 3.5. Assumptions A-1 through A-3 are assumed to hold for (15).

Because of how the decomposition will arise in the case of bipeds, we think of the  $x_1$ -states as the “weakly actuated part” of the system and the  $x_2$ -states as the “strongly actuated part” of the system. With the model expressed in this form, it is clear that the  $x_2$ -states are virtual controls for the  $x_1$ -states. We will continue to build open-loop trajectories by the full-order model, except now the trajectories will be computed for a *reduced set of initial conditions defined by the  $x_1$ -subsystem*.

**Definition 1.** An insertion map,  $\gamma : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_2}$ , is a function that preserves the equilibrium point, namely  $\gamma(0) = 0$ . ■

The insertion map specifies initial conditions for  $x_2$  as a function of  $x_1$ ; in other words, it specifies the surface  $Z_0$  in the Introduction, just before (2).

**A-5** There is a constant  $0 \leq c < 1$ , such that, for each initial condition  $\xi = (\xi_1, \gamma(\xi_1)) \in B$ , there exists a piecewise continuous input  $u_{\xi_1} : [0, T_p] \rightarrow \mathbb{R}^m$  and a corresponding solution of the ODE,  $\varphi_{\xi_1} : [0, T_p] \rightarrow \mathbb{R}^n$  satisfying  $\varphi_{\xi_1}(T_p) \in B$ , and

$$V((\varphi_{1\xi_1}(T_p), \gamma(\varphi_{1\xi_1}(T_p)))) \leq cV(\xi_1, \gamma(\xi_1)), \quad (16)$$

where the solution of the  $(n_1 + n_2)$ -dimensional model (15) has been decomposed as

$$\varphi_{\xi_1}(t) =: (\varphi_{1\xi_1}(t), \varphi_{2\xi_1}(t)).$$

Features		Labels
$t_j$	$x_1^{j,i} = \varphi_{1\xi_1^i}(t_j)$	$\nu^{j,i} = \varphi_{2\xi_1^i}(t_j)$
$t_0 = 0$	$x_1^{0,1}$	$\nu^{0,1}$
$t_0 = 0$	$x_1^{0,2}$	$\nu^{0,2}$
$\vdots$	$\vdots$	$\vdots$
$t_0 = 0$	$x_1^{0,M}$	$\nu^{0,M}$
$t_1$	$x_1^{1,1}$	$\nu^{1,1}$
$\vdots$	$\vdots$	$\vdots$
$t_1$	$x_1^{1,M}$	$\nu^{1,M}$
$\vdots$	$\vdots$	$\vdots$
$t_N = T_p$	$x_1^{N,1}$	$\nu^{N,1}$
$\vdots$	$\vdots$	$\vdots$
$t_N = T_p$	$x_1^{N,M}$	$\nu^{N,M}$

**Table 2.** Conceptual arrangement of the data in A-5 from which a controller satisfying (17) may be determined by “regression”. Since only the  $x_1$ -component of the state is sampled, far fewer optimizations are required. The number of time samples remains the same. ■

**Proposition 2.** Assume the open-loop system (15) satisfies Assumptions A-1 to A-3 and A-5, and define  $B_1 := \{\xi_1 \in \mathbb{R}^{n_1} \mid (\xi_1, \gamma(\xi_1)) \in B\}$ . Assume in addition there exists an open set  $B_1^e \supset B_1$  and a function

$$\nu : [0, \infty) \times B_1^e \rightarrow \mathbb{R}^{n_2}$$

that is piecewise continuous in  $t$ ,  $T_p$ -periodic, locally Lipschitz continuous in  $x_1$ , and, such that, for  $0 \leq t < T_p$  and  $\xi_1 \in B_1$ ,

$$\nu(t, \varphi_{1\xi_1}(t)) = \varphi_{2\xi_1}(t). \quad (17)$$

Then the origin of the reduced-order system

$$\dot{x}_1 = f_{\text{red}}^{\text{cl}}(t, x_1) := f_1(t, x_1, \nu(t, x_1)), \quad (18)$$

is locally uniformly exponentially stable, and the trajectories in A-5 are solutions of (18). ■

**Remark 2.**

- (i) Assumption A-5 and Prop. 2 represent our first result to mitigate the curse of dimensionality. Assumption A-5 leads to a greatly reduced training set for building a realization than A-4 because, in many practical examples,  $n_1 \ll (n_1 + n_2)$ . Proposition 2 says that this reduced training set can encode a stabilization goal that is a feasible action of the full-order model.



The next section embeds the target model (18) in the full-order model, completing our basic plan for mitigating the curse of dimensionality.

- (ii) The numerical burden of developing the “training sets” for  $\nu(t, x_1)$  is exponential in the dimension of  $x_1$ , at least if a uniform grid is used to sample  $B_1$ .
- (iii) Table 2 shows how to extract the function  $\nu(t, x_1)$  from the optimization data.
- (iv) The local Lipschitz continuity of  $\nu(t, x_1)$  imposes stronger conditions on the solutions given in A-5 than those encountered in A-4. This is because the mapping defined by, for each  $t \in [0, T_p]$ ,

$$\Psi_{1t} : B_1 \rightarrow \mathbb{R}^{n_1}, \quad \Psi_{1t}(\xi_1) := \varphi_{1\xi_1}(t) \quad (19)$$

being injective is stronger than the mapping

$$\Psi_t : B_1 \rightarrow \mathbb{R}^n, \quad \Psi_t(\xi_1) := \begin{bmatrix} \varphi_{1\xi_1}(t) \\ \varphi_{2\xi_1}(t) \end{bmatrix} \quad (20)$$

being injective. If  $\Psi_t$  is continuously differentiable and full rank, then there does exist a new choice of  $x_1$ -coordinates for which the corresponding mapping  $\Psi_{1t}$  is full rank and hence is locally injective. This will be illustrated on the cart-pendulum model in Sect. 4.

- (v) Under the assumptions of Prop. 2, for each  $t \in [0, T_p]$ ,  $\Psi_t : B_1 \rightarrow \mathbb{R}^n$  is a homeomorphism onto its image. It follows that  $\Psi_e : [0, T_p] \times B_1 \rightarrow \mathbb{R}^{n+1}$ , by

$$\Psi_e(t, \xi_1) := \begin{bmatrix} t \\ \Psi_t(\xi_1) \end{bmatrix},$$

is also a homeomorphism onto its image. After augmenting the state with time in the usual manner, the low-dimensional model (4) discussed in the Introduction can be seen as evolving on the surface

$$Z_{T_p} := \Psi_e([0, T_p] \times B_1), \quad (21)$$

with the dynamics and output given by

$$\begin{aligned} \dot{\tau} &= 1 \\ \dot{x}_1 &= f_1(\tau, x_1, \nu(\tau, x_1)) \\ x &= \begin{bmatrix} x_1 \\ \nu(\tau, x_1) \end{bmatrix}. \end{aligned} \quad (22)$$

At this point, the direct relation with trajectories of the original model is only true for  $0 \leq t < T_p$ .

- (vi) In Sect. 3.6, we will provide a concrete way to select the insertion map. For now, we propose an insertion

map inspired by backstepping

$$\gamma(x_1) := Kx_1, \quad (23)$$

that the equilibrium of the reduced-order model,

$$\dot{x}_1 = f_1(t, x_1, \gamma(x_1)) \quad (24)$$

is stable. Other relations to backstepping are noted in Appendix D.

- (vii) Suppose the system (15) is time invariant, so that one is stabilizing a trivial periodic orbit (i.e., an equilibrium). Then  $T_p > 0$  is a free parameter available to the designer. How to choose it? If the insertion map actually stabilizes the equilibrium of the reduced-order model (24), then in principle,  $T_p$  can be taken to be arbitrarily small, subject to choosing  $c > 0$  and the positive function in (16) properly. Otherwise, if the system is locally asymptotically controllable to the origin Clarke et al. (1997), a larger  $T_p$  makes it easier to meet the Lyapunov contraction condition.

### 3.4 Embedding the Target Dynamics in the Original System

Consider the system (15) with the assumptions and notation of Prop. 2. Assume there exists a feedback  $u(t, x_1, x_2)$  such that in the coordinates

$$y := x_2 - \nu(t, x_1), \quad (25)$$

the closed-loop system has the form

$$\begin{aligned} \dot{x}_1 &= f_1(t, x_1, \nu(t, x_1) + y) \\ \dot{y} &= Ay, \end{aligned} \quad (26)$$

with  $A$  Hurwitz. Then the surface  $y \equiv 0$  is invariant and the restriction dynamics is given by (18). While  $\nu(t, x_1)$  is  $T_p$ -periodic, its limits from the left and the right are not necessarily equal at  $T_p$ , and  $y$  in (25) inherits this property. Hence, without further assumptions, it cannot be a solution of the ODE (26), in the usual sense (Khalil 2002, pp. 657), namely

$$y(t) = y(t_0) + \int_{t_0}^t Ay(\tau) d\tau.$$

In the following, we impose continuity in  $t$  on  $\nu(t, x_1)$ , and then after the theorem, analyze what this means in terms of the trajectories coming out of the optimizer.

**Theorem 1.** Assume the open-loop system (15) satisfies Assumptions A-1 to A-3, and A-5, and define  $B_1 := \{\xi_1 \in \mathbb{R}^{n_1} \mid (\xi_1, \gamma(\xi_1)) \in B\}$ . Assume in addition there exists an

open set  $B_1^e \supset B_1$  and a feedback

$$\nu : [0, \infty) \times B_1^e \rightarrow \mathbb{R}^{n_2} \quad (27)$$

that is continuous in  $t$ ,  $T_p$ -periodic, locally Lipschitz continuous in  $x_1$ , and, such that, for  $0 \leq t < T_p$  and  $\xi_1 \in B_1$ ,

$$\nu(t, \varphi_{1\xi_1}(t)) = \varphi_{2\xi_1}(t). \quad (28)$$

Then any feedback  $u(t, x_1, x_2)$ , piecewise continuous in  $t$  and locally Lipschitz continuous in  $x_1$  and  $x_2$  that transforms the system to (26), with  $A$  Hurwitz, renders the origin of (26) locally uniformly exponentially stable. Moreover, the surface defined by

$$Z_e := \{(t, x_1, x_2) \mid (t \bmod T_p, x_1, x_2) \in Z_{T_p}\}, \quad (29)$$

is invariant with restriction dynamics given by (22). ■

**Remark 3.**

- (i) In fact, (29) is the Isidori-Byrnes *Isidori (1995) Zero Dynamics Manifold* and the Zero Dynamics is given by

$$\begin{aligned} \dot{\tau} &= 1 \\ \dot{x}_1 &= f_1(\tau, x_1, \nu(\tau, x_1)). \end{aligned}$$

The output that is being “zeroed” is

$$y = x_2 - \nu(t, x_1)$$

as long as the domain is properly specified.

- (ii) If  $\nu$  in (27) is continuous in  $t$ , then for all  $x_1 \in B_1^e$ ,  $\nu(T_p, x_1) = \nu(0, x_1)$ . How does this relate to the trajectories in the training set used to generate  $\nu$ ? Because  $V$  in (16) is positive definite, and  $V$  decreases after  $T_p$  seconds, there exists an open ball  $B_2$  contained in  $B_1$  such that  $\xi_1 \in B_2 \Rightarrow \varphi_{1\xi_1}(T_p) \in B_1$ . Because  $\hat{\xi}_1 := \varphi_{1\xi_1}(T_p) \in B_1$ ,

$$\varphi_{2\hat{\xi}_1}(0) = \gamma(\hat{\xi}_1) := \gamma(\varphi_{1\xi_1}(T_p)). \quad (30)$$

From (17), because  $\hat{\xi}_1 \in B_1$ ,

$$\nu(0, \varphi_{1\hat{\xi}_1}(0)) = \varphi_{2\hat{\xi}_1}(0), \quad (31)$$

and because  $\varphi_{1\hat{\xi}_1}(0) = \hat{\xi}_1$ , we have

$$\nu(0, \varphi_{1\hat{\xi}_1}(0)) = \nu(0, \hat{\xi}_1). \quad (32)$$

From the continuity of  $\nu$  and using the definition of  $\hat{\xi}_1$ ,

$$\nu(0, \varphi_{1\xi_1}(T_p)) = \nu(T_p, \varphi_{1\xi_1}(T_p)). \quad (33)$$

From (17) again,

$$\nu(T_p, \varphi_{1\xi_1}(T_p)) = \varphi_{2\xi_1}(T_p). \quad (34)$$

Putting these together, the corresponding condition on the trajectories used in the training set for  $\nu$  is given in A-6.

**A-6** The solutions in A-5 also satisfy

$$\gamma(\varphi_{1\xi_1}(T_p)) = \varphi_{2\xi_1}(T_p). \quad (35)$$

■

Section 4 will illustrate these ideas on a simple low-dimensional example to make it easy for the interested reader to reproduce the results. The true benefits of the approach will not be clear until Sect. 6, where it will be applied to a high-dimensional hybrid model of a bipedal robot, and subsequently implemented in hardware on the robot of Figure 17.

### 3.5 Extended class of models

We discuss the case with the input appearing in both blocks. To keep the presentation brief and simple, it is supposed that the model has the form

$$\begin{aligned} \dot{x}_1 &= f_1(t, x_1, x_2, u) \\ \dot{x}_2 &= f_2(t, x_1, x_2, u), \end{aligned} \quad (36)$$

with

$$x_2 = \begin{bmatrix} x_{2a} \\ x_{2b} \end{bmatrix} \text{ and } f_2 = \begin{bmatrix} x_{2a} \\ u \end{bmatrix}.$$

**Corollary 1.** Assume the open-loop system (36) satisfies Assumptions A-1 to A-3, A-5, and A-6, and define  $B_1 := \{\xi_1 \in \mathbb{R}^{n_1} \mid (\xi_1, \gamma(\xi_1)) \in B\}$ . Assume in addition there exists an open set  $B_1^e \supset B_1$ , a function

$$\nu : [0, \infty) \times B_1^e \rightarrow \mathbb{R}^{n_2} \quad (37)$$

satisfying the conditions of Theorem 1, and a second function

$$\mu : [0, \infty) \times B_1^e \rightarrow \mathbb{R}^m \quad (38)$$

that is piecewise continuous in  $t$ ,  $T_p$ -periodic, locally Lipschitz continuous in  $x_1$ , and such that, for  $0 \leq t < T_p$  and  $\xi_1 \in B_1$ ,

$$\mu(t, \varphi_{1\xi_1}(t)) = u_{\xi_1}(t). \quad (39)$$

Then for all  $n_2/2 \times n_2/2$  positive definite matrices  $K_p$  and  $K_d$ , the origin of

$$\begin{aligned}\dot{x}_1 &= f_1(t, x_1, x_2, u) \\ \dot{x}_2 &= f_2(t, x_1, x_2, u) \\ u &= \mu(t, x_1) - [K_p \ K_d](x_2 - \nu(t, x_1))\end{aligned}\quad (40)$$

is locally exponentially stable, uniformly in  $t_0$ . Moreover, the surface defined by (29) is invariant with restriction dynamics given by

$$\begin{aligned}\dot{\tau} &= 1 \\ \dot{x}_1 &= f_1(t, x_1, \nu(t, x_1), \mu(t, x_1)) \\ x &= \begin{bmatrix} x_1 \\ \nu(\tau, x_1) \end{bmatrix}.\end{aligned}\quad (41)$$

#### Remark 4.

- (i) There is no extra boundary condition, such as A-6, associated with (28) because the term  $\mu$  arises from the inputs instead of the states of the ODE, as in the case of  $\nu$ . In particular,  $\mu$  can be piecewise continuous in  $t$ . The learning of  $\mu$  is done the same as for  $\nu$  in Table 2.
- (ii) Most systems will require a pre-feedback to arrive at the form (36); this must be taken into account when implementing the feedback indicated in (40).

### 3.6 Orbit Library and Design of the Insertion Map

The objective of this section is to provide a systematic means for designing the insertion map in a way that takes into account the ‘‘physics’’ of a model.

**Definition 2.** An orbit library  $\mathcal{L}$  is a set of periodic trajectories of the model (36) that are parameterized by the  $x_1$ -states. We denote the library consisting of the periodic solutions by

$$\mathcal{L} := \{\varphi_{\xi_1} : [0, T_p] \rightarrow \mathbb{R}^n \mid \xi_1 \in B_1\}, \quad (42)$$

with  $B_1$  as A-5.

**Definition 3.** An insertion map associated to an orbit library (42) is a function  $\gamma_{\mathcal{L}} : B_1 \rightarrow \mathbb{R}^{n_2}$  such that

$$\gamma_{\mathcal{L}}(\xi_1) := \varphi_{2\xi_1}(0). \quad (43)$$

One should think of the above insertion map as taking the states of the  $x_1$ -coordinates, associating them to periodic

orbits of the full model, and then defining the initial condition of the  $x_2$ -coordinates (in the trajectory optimization) to be its value at a point on the associated periodic orbit. Hence, the overall model is being initialized in a physically meaningful manner. Moreover, the trajectories in A-6 can now be interpreted as affecting a transition *from* a family of periodic solutions *to* a desired periodic solution *in a way that* leads to stabilization of the desired periodic solution, via Theorem 1 or Corollary 1. The authors have found this to be very useful on bipedal robots.

## 4 Inverted Pendulum on a Cart

- This section will illustrate the controller designs of Section 3 on the well-known inverted pendulum on a cart model. The MATLAB code for the calculations is available for download in Extension 1. The optimization setup and the learning method used here are nearly identical to what will be implemented on a bipedal robot in Section 6; the only significant change involves the hybrid aspect of a biped model.

### 4.1 System Model

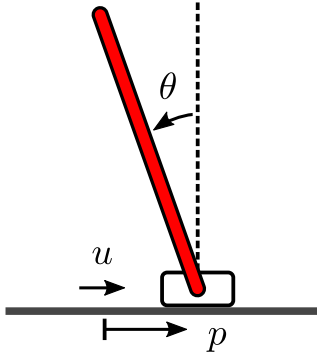
The system consists of a unit length, uniformly distributed unit-mass pendulum attached via a revolute joint to a planar unit-mass cart, shown in Figure 6. A driving force is applied on the cart, and there is no torque acting on the revolute joint of the pendulum. The motion of the cart and the pendulum are free of friction forces. The configuration variable  $q := (p, \theta)$  consists of the cart position and the pendulum angle.

The system is written in state variable form as

$$\begin{aligned}\dot{x}_1 &= \begin{bmatrix} \dot{p} \\ \frac{2 \sin(\theta) \dot{\theta}^2 - 3g \cos(\theta) \sin(\theta) - 4u}{3 \cos(\theta)^2 - 8} \end{bmatrix} \\ \dot{x}_2 &= \begin{bmatrix} \dot{\theta} \\ \frac{3 \cos(\theta) \sin(\theta) \dot{\theta}^2 - 12g \sin(\theta) - 6 \cos(\theta) u}{3 \cos(\theta)^2 - 8} \end{bmatrix},\end{aligned}\quad (44)$$

where  $u$  is the force acting on the cart and the system state  $x$  is decomposed into  $x_1 = (p, \dot{p})$  and  $x_2 = (\theta, \dot{\theta})$ . The equilibrium point of the upright pendulum is  $x^* = 0$  and  $u^* = 0$ . Assumptions A-1 and A-2 are then trivially satisfied.

The overall control objective will be to locally exponentially stabilize a continuum of periodic motions with a common period  $T_p = 2$  seconds. We first illustrate the control design method on a trivial periodic orbit corresponding to the pendulum upright and the cart at the origin.



**Figure 6.** An inverted pendulum on a cart model is used to illustrate the controller designs of Sect. 3. The objective is to stabilize a continuum of periodic motions, including a trivial periodic orbit corresponding to the pendulum upright and the cart at the origin. In part of the analysis, a barrier is imposed.

## 4.2 Stabilizing the Upright Equilibrium While Respecting a Barrier

The presentation follows the basic steps of the design, from learning a full-state feedback as in Prop. 1 to embedding a target model as in Corollary 1.

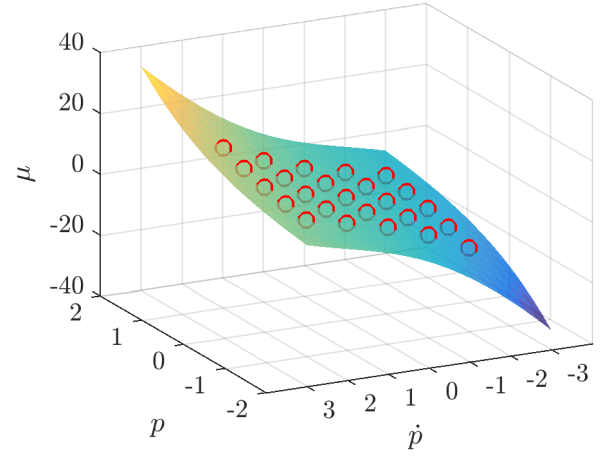
**4.2.1 Trajectory Generation and Learning for the Full-Order Model** The set  $B$  and positive definite function of A-3 are discussed shortly. For an initial state  $\xi \in B$ , the direct collocation algorithms of Jones (2014); Hereid et al. (2016a) are used to generate a trajectory  $\varphi_\xi(t)$  and corresponding input  $u_\xi(t)$  over an interval  $[0, T_p]$  to meet the conditions of A-4. To emphasize the ability to handle interesting constraints in the control design, the cart position is heavily penalized if it moves out of a “safe region”  $[-p_b, p_b]$ , with  $p_b = 2$ .

The cost function for determining the trajectories is a standard quadratic form with an additional penalty for the safety region:

$$J(\xi) = \min_u \int_0^{T_h} (\|x\|_Q^2 + \|u\|_R^2 + L(p, p_b)) dt \quad (45)$$

$$L(p, p_b) = wp^2(e^{p-p_b} + e^{-p-p_b}).$$

The weights  $Q$  and  $R$  are taken as identity matrices and the penalty weight is  $w = 10$ . The optimization is subjected to the system dynamics constraints(44) and the terminal constraint  $x(T_h) = 0$ , with  $T_h = 3T_p = 6$  seconds. One could also use a terminal cost  $N(x(T_h))$  in place of the terminal constraint. Even though a terminal constraint may make the optimization problem infeasible for some initial conditions, we have found it to be quite practical in bipedal robots. As discussed earlier, the cost function in (45) can often used as a Lyapunov function meeting the conditions in A-4. Here, we do not add this as a constraint



**Figure 7.** A slice of the function  $\mu(t, p, \dot{p}, \theta, \dot{\theta})$ , with  $t = 0$ ,  $\theta = 0$  and  $\dot{\theta} = 0$ . The presence of the (soft-penalty) barrier is most evident near  $p = -2$  and  $\dot{p} = -3$ . The circles are training and validation data. Both interpolation and extrapolation can be seen in the surface.

**Table 3.** MATLAB Neural Network Fitting Parameters

Hidden neurons	50
Training Ratio	80%
Validation Ratio	20%
Training Algorithm	Bayesian Regularization
Max Iteration	4000

to the optimization and will illustrate the satisfaction of the Lyapunov condition.

The function  $\mu(t, x)$  in Prop. 1 is learned for the ball of initial conditions

$$B = \{-1 \leq p \leq 1, -\frac{\pi}{6} \leq \theta \leq \frac{\pi}{6}, \quad (46)$$

$$-2 \leq \dot{p} \leq 2, -2 \leq \dot{\theta} \leq 2\},$$

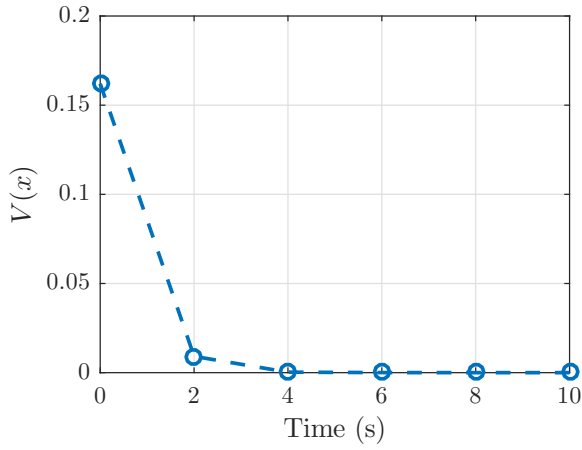
with samples  $\xi^i \in B$  selected from a uniform grid of the state. Five points are used in each dimension, for a total of 625 input sequence  $u_{\xi^i}(t)$  and solutions  $\varphi_{\xi^i}(t)$ . At each time

$$t_j \in \{t \mid j \frac{T_p}{40}, j = 0, 1, \dots, 40\}, \quad (47)$$

the time-state pair  $(t_j, x^{j,i})$  is a feature and the input  $u^{j,i}$  is a label. The complete list of features and labels is shown in Table. 1 in Section 3.2. We use the MATLAB Neural Network Fitting Toolbox to approximate  $\mu$ . The fitting setup is shown in Table 3. The mean squared error of the validation set is around  $10^{-4}$ .

We show there exists a Lyapunov function

$$V(x) := x^\top P x \quad (48)$$



**Figure 8.** The plot shows the typical evolution of the optimization-cost function, confirming that it serves as a Lyapunov function. It is to be noted that  $V(t)$  is only required to monotonically decrease from sample to sample, that is, from  $kT_p$  to  $(k+1)T_p$ , with  $T_p = 2$ .

as required in A-4 that is built from the cost function in (45). The matrix

$$P = \begin{bmatrix} 0.04 & -0.11 & 0.03 & -0.03 \\ -0.11 & 0.94 & -0.12 & 0.18 \\ 0.03 & -0.12 & 0.03 & -0.03 \\ -0.03 & 0.18 & -0.03 & 0.04 \end{bmatrix} \quad (49)$$

is from a regression of  $J(x)$ . The matrix is positive definite. We next find the constant  $c$  in A-4 using the data set to train  $\mu(t, x)$  that  $c$  satisfies

$$c \geq \max_i \frac{V(\varphi_{\xi^i}(T_p))}{V(\varphi_{\xi^i}(0))}. \quad (50)$$

The maximum ratio over 625 points of  $\xi^i$  is 0.22, then we set  $c = 0.25$ . Notice when  $\xi^i$  is the equilibrium point,  $V(\varphi_{\xi^i}(T_p)) = V(\varphi_{\xi^i}(0)) = 0$ , which has to be ignored when finding  $c$ . Figure 8 shows that  $V(\varphi_{\xi}(kT_p))$  in the simulation is exponentially decreased.

**4.2.2 Comparison of Continuous Hold vs Learned Feedback** Figure 9 compares  $u^{ch}$  in (9) and  $\mu$  in (10). The continuous-hold feedback  $u^{ch}$  updates the state every  $T_p = 2$  seconds. This type of MPC-style controller is guaranteed to perform poorly in the face of disturbances occurring within the sample period. Figure 9-(a) shows the continuous-hold controller stabilizing the system from the initial condition

$$(p, \dot{p}, \theta, \dot{\theta}) = (-1, 0, \pi/12, 0). \quad (51)$$

Figure 9-(b) shows that the learned feedback  $\mu$  performs identically to  $u^{ch}$  given the same initial condition and a perfect model. The difference is obvious when it comes to disturbance rejection. A constant external force  $d = 1N$  is

applied to the cart for  $t \in [11.5, 12]$ . The continuous-hold feedback  $u^{ch}$  has to wait till  $t = 12$  to update the state and respond to the disturbance; on the other hand, the learned feedback  $\mu$  updates the state continuously and hence reacts to the disturbance immediately.

**4.2.3 Building a Reduced-Order Target Model** The previous subsection illustrated how feedback is extracted from data via Supervised Machine Learning and will provide a benchmark for later designs. Because the cart-pendulum system has only four states, the number of trajectories required for training in the previous subsection was quite manageable, and with random sampling techniques Parisini and Zoppoli (1995), it could be further reduced. Eventually, however, the number of required optimizations will become untenable. Here we illustrate Prop. 2 for a two-dimensional subsystem of the cart-pendulum system.

Recall that the system state decomposition was already shown in (44). The insertion map used here is inspired by backstepping as in Remark 2. Linearizing the  $x_1$ -subsystem with  $u = 0$  and selecting  $x_2$  as a stabilizing linear feedback yields

$$\gamma(x_1) = \begin{bmatrix} 0.03 & 0.1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \end{bmatrix}. \quad (52)$$

We do not further explore the choice of  $\gamma$  because we will primarily use the orbit library  $\gamma_{\mathcal{L}}$  of Definition 3 in the remainder of the paper, including the bipedal robot section.

With this insertion map, the trajectories required by A-5 are determined via optimization with

$$B_1 = \{-1 \leq p \leq 1, -2 \leq \dot{p} \leq 2\}. \quad (53)$$

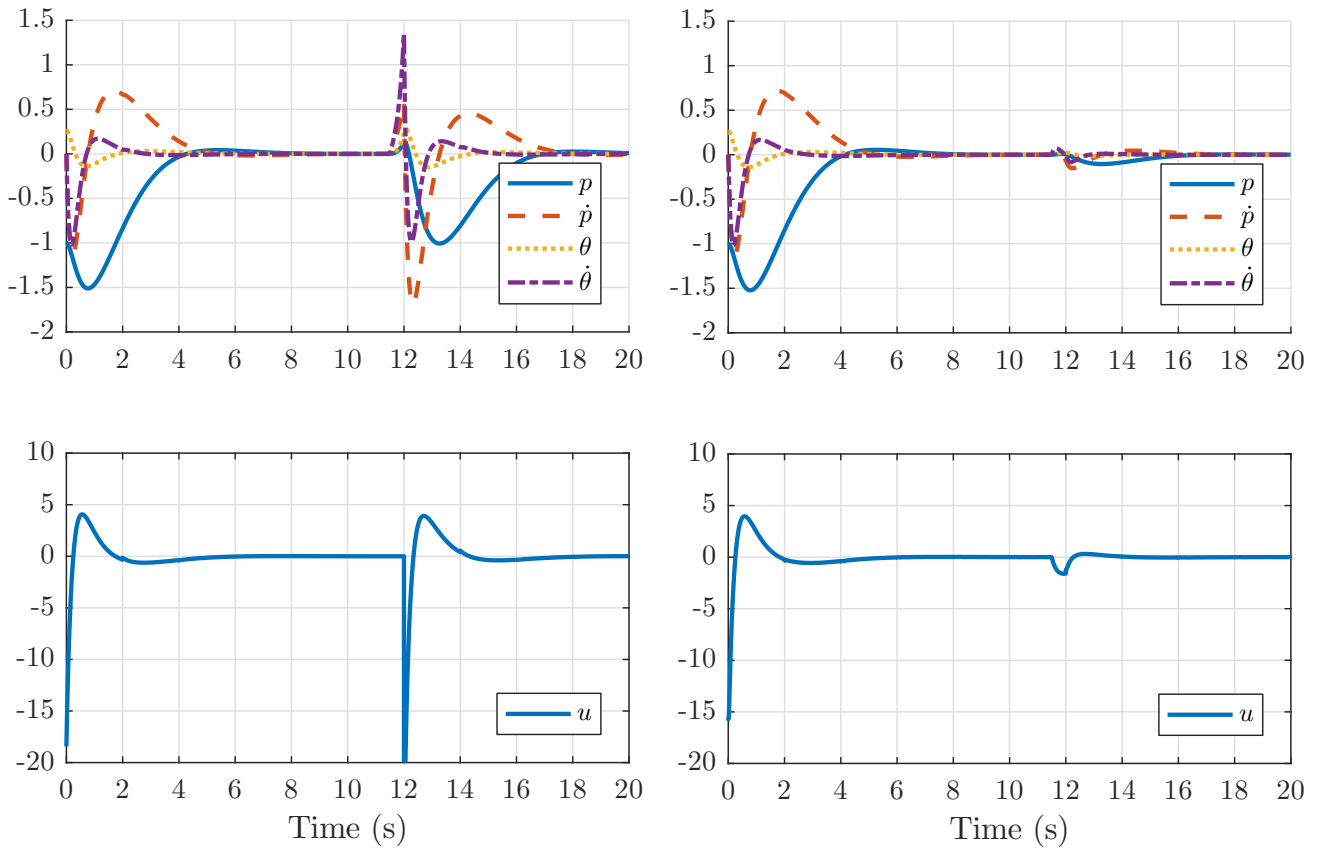
In anticipation of using the results here in Corollary 1, the boundary condition of A-6 is also imposed.

The set of initial conditions  $\xi_1^j$ ,  $j \in J$  now has 25 points instead of 625 points. The mapping (19) is checked to be injective by evaluating the numerical rank of the  $x_1$ -features in Table 2 via SVD. Just as before, the function  $\nu(t, x_1)$  is obtained from the data via the MATLAB Neural Network Fitting Toolbox, again with the parameters indicated in Table 3. The same holds for the function  $\mu(t, x_1)$ . An example of the fitting of  $\nu$  is shown in Figure 10.

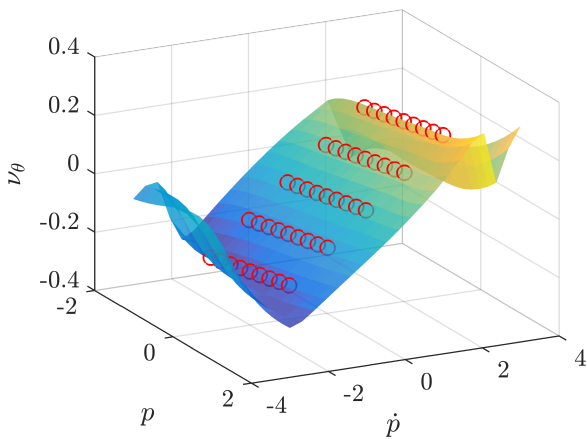
The evolution of the target model is shown in the next subsection when a disturbance is applied after  $y = x_2 - \nu(t, x_1)$  has nearly converged to zero.

**4.2.4 Embedding the Target Dynamics in the Original System** The learned functions from the reduce-order optimization are now used to stabilize the full-order system based on Theorem 1 and Corollary 1. To place the system in



(a) Continuous-hold controller  $u^{ch}$ (b) Learned feedback  $\mu$ 

**Figure 9.** Comparison of the continuous-hold controller in (a), versus the learned controller in (b). In both cases, the initial condition is  $(p, \dot{p}, \theta, \dot{\theta}) = (-1, 0, \pi/12, 0)$ . A disturbance is applied for  $t \in [11.5, 12]$  seconds. The classical MPC plot is in an Appendix-E



**Figure 10.** A slice of the function  $\nu(t, p, \dot{p})$ , with  $t = 0$ . This is the  $\nu$  associated with  $x_2$  coordinate  $\theta$ . The circles are training and validation data. The interpolation is smooth while the extrapolation may be not.

the form (36), a pre-feedback is applied

$$\bar{u} := \frac{3 \cos(\theta) \sin(\theta) \dot{\theta}^2 - 12g \sin(\theta) - 6 \cos(\theta) u}{3 \cos(\theta)^2 - 8} \quad (54)$$

resulting in

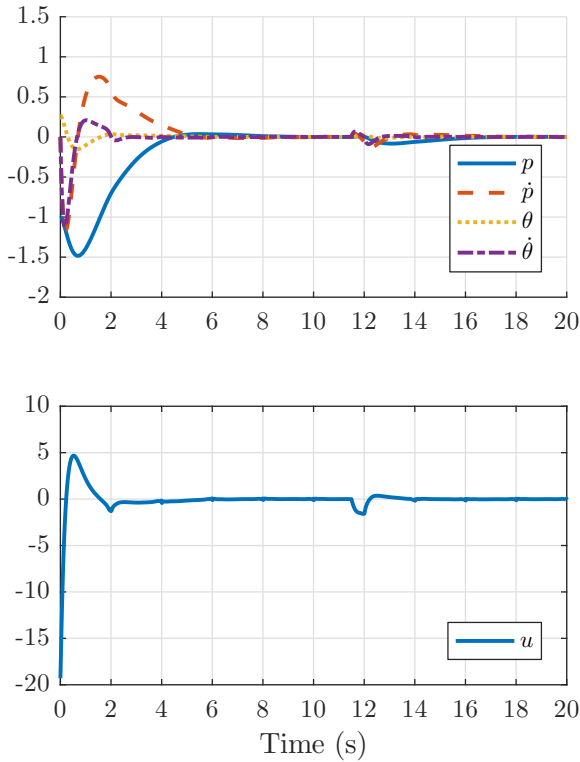
$$\ddot{\theta} = \bar{u}.$$

The original input  $u$  can be computed from  $\bar{u}$  because (54) is invertible in the operational range of interest, namely  $-\pi/2 < \theta < \pi/2$ . While the function  $\bar{\mu}(t, x)$  of Corollary 1 can be recovered from  $\mu(t, x_1)$  and (54), it is just as easy to learn it with the features  $(t_j, x_1^{j,i})$  and label  $\bar{u}^{j,i}$ . In the full model,

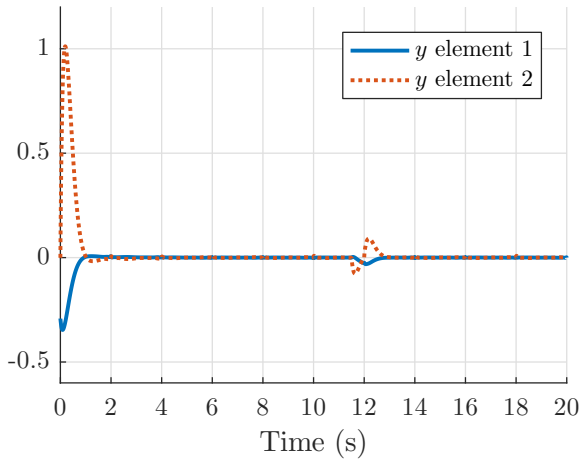
$$\bar{u} = \bar{\mu}(t, x_1) - [K_p \ K_d](x_2 - \nu(t, x_1)), \quad (55)$$

with  $K_p = 50$  and  $K_d = 15$ .

Figure. 11 shows the response of the closed-loop system with the same initial condition and perturbation of Figure. 9. The settling time and disturbance rejection performance is similar to the full state learned feedback. Figure 12 illustrates the attractiveness of the surface  $x_2 = \nu(t, x_1)$  by showing that the output error in (25) of the full-order system decays exponentially to zero. When the disturbance is applied for  $11.5 \leq t < 12$ , the output is driven away from zero and then decays back quickly when the disturbance is removed.



**Figure 11.** Response of the reduced-order model (41). The states of the model are  $p$  and  $\dot{p}$ , while  $\theta$  and  $\dot{\theta}$  are outputs. The initial condition and disturbance are as in Figure 9.



**Figure 12.** Showing the convergence of the output error in (25). Comparing this figure with Figure 11 shows that the system converges to the zero dynamics surface more quickly than it converges to the periodic orbit. The disturbance initially drives the system away from the surface.

### 4.3 Orbit Library and Transitioning Among Periodic Orbits

The last subsection has gone through the control design process for a trivial periodic orbit where the pendulum is upright, and the cart is at the origin. This subsection designs a controller for a set of periodic orbits, illustrate an insertion map  $\gamma_{\mathcal{L}}$  arising from an orbit library, and shows the possibility of the mapping (19) not being injective. To

simplify matters, we work directly with the cart-pendulum system *after* the pre-feedback (54) has been applied.

**4.3.1 Orbit Library** For  $T_p = 2$  seconds, define a set of periodic motions of the cart by

$$p(t) = p_0 + \frac{\dot{p}_0}{\pi} \sin(\pi t),$$

for  $(p_0, \dot{p}_0) \in B_1$  in (53). The trajectory for  $p(t)$  fixes the acceleration of the cart, which in turn gives trajectories for  $\theta(t)$ ,  $\dot{\theta}(t)$ , and  $u(t)$ . Moreover, imposing  $-\pi/2 < \theta < \pi/2$  selects among the two possible solutions for the model. These considerations define an orbit library  $\mathcal{L}$ , with solutions indexed by  $(p_0, \dot{p}_0)$ . Denote the set of initial conditions of the orbit library as  $(\xi_1^{\mathcal{L}}, \xi_2^{\mathcal{L}})$ . Recalling Definition 3, an insertion map associated to the orbit library is

$$\gamma_{\mathcal{L}}(\xi_1^{\mathcal{L}}) = \xi_2^{\mathcal{L}}.$$

To make it more explicit, for this example, we use linear regression to find  $\gamma_{\mathcal{L}} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  as

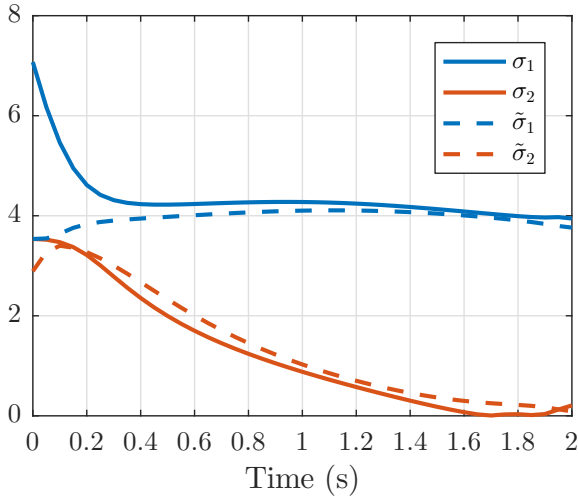
$$\begin{bmatrix} \theta_0 \\ \dot{\theta}_0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0.5911 \end{bmatrix} \begin{bmatrix} p_0 \\ \dot{p}_0 \end{bmatrix}. \quad (56)$$

**Remark 5.** *The reader may be wondering why we bring up the orbit library as a means of computing a new insertion function, especially when the ‘backstepping-inspired’ insertion map worked so well? The point is that for a robot, where the dimension of  $x_2$  may be twenty, one has no idea how to design a ‘backstepping-inspired’ insertion map, whereas the concept of an orbit library extends naturally as will be seen later in the paper.*

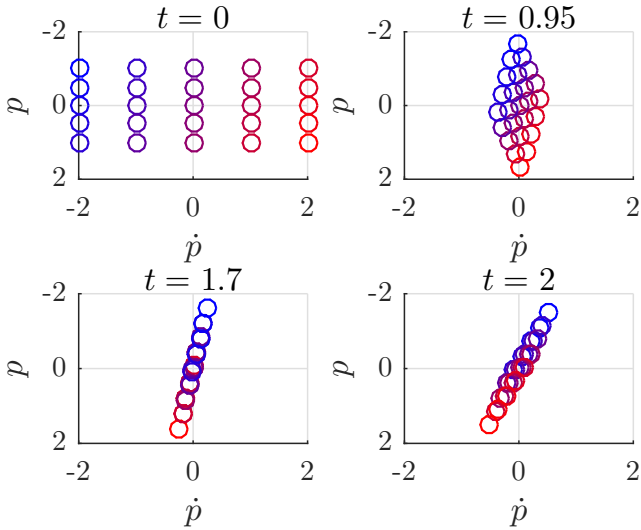
**4.3.2 Loss and Recovery of Injectivity** Using the new insertion function (56), trajectory generation is performed exactly as in Sect. 4.2, with  $x_1$  and  $x_2$  as given in (44). The mapping (19) is checked not to be injective. Indeed, for  $t \simeq 1.8$ , the cart trajectories pass through a one-dimensional surface as shown in Figures 13 and 14, while the mapping (20) remains injective. Hence, one expects the existence of a set of coordinates in which the design can proceed. It can be checked that the new coordinates<sup>2</sup>

$$\tilde{x}_1 = (p - \theta, \dot{p} - \dot{\theta})$$

are “full rank” as shown in Figure 13. It is not necessary to redo the optimization because the new coordinates correspond to a new  $\tilde{B}_1$  and a new insertion map,  $\tilde{\gamma}_{\mathcal{L}}$ , and these are not explicitly required in the computation of  $\nu$  and  $\mu$ . The important thing is the feature set for the Supervised



**Figure 13.** The singular values of the matrix formed by the sampled trajectories versus time the insertion function is given by (56). The solid lines are for  $(p(t), \dot{p}(t))$ , with  $\Delta t = 0.05$ , while the dashed lines are for  $(p(t) - \theta(t), \dot{p}(t) - \dot{\theta}(t))$  for the same time samples. For the choice of the insertion function arising from backstepping in (52), the minimum value of  $\sigma_2$  was 0.58.



**Figure 14.** Another perspective on the information in Figure 13. The initial conditions are taken from a grid, as can be seen at  $t = 0$ . At subsequent times, the grid is transformed into a parallelogram at  $t = 0.95$  and a line at  $t = 1.7$ , where the mapping  $\Psi_{1t} : B_1 \rightarrow \mathbb{R}^{2n}$  in (19) loses rank. We have not yet observed this problem in the case of bipedal robots.

Machine Learning is now indexed by  $(t_j, \tilde{x}_1^{j,i})$  rather than  $(t_j, x_1^{j,i})$ .

**4.3.3 Transitioning Between Periodic Orbits** Next, we use the library insertion map  $\gamma_{\mathcal{L}}$  and the new state  $\tilde{x}_1$  to design a controller for transitioning between periodic orbits; this is analogous to transitioning between walking gaits of various speeds or direction for a bipedal robot. The cost function used in A-5 is modified to include the target orbit

$A := (p_0, \dot{p}_0) \in B_1$ , per (57) to

$$J(\xi_1, A) = \min_u \int_0^{T_h} (\|x - \varphi^A\|_Q^2 + \|u - u^A\|_R^2) dt \quad (57)$$

subject to  $x(T_h) = \varphi^A(T_p)$  and  $x(0) = (\xi_1, \gamma_{\mathcal{L}}(\xi_1))$ . The boundary condition A-6 is also applied as  $\gamma_{\mathcal{L}}(x_1(T_p)) = x_2(T_p)$ . Here, the target trajectory and its corresponding input are denoted as  $\varphi^A(t)$  and  $u^A(t)$ . To simplify the problem, the cost function excludes the barrier penalty  $L(p, p_b)$  in (45). The set  $B_1$  is still given by (53).

With this choice of the insertion map, the boundary condition A-6 means that each cycle in the transition moves the cart-pendulum from one periodic orbit to the next; this is because  $(\varphi_{1\xi_1}(T_p), \gamma_{\mathcal{L}}(\varphi_{1\xi_1}(T_p)))$  is an initial condition for a periodic solution of the model. Denote the family of solutions to the optimization problem by

$$\begin{aligned} \tilde{x}_1^{i,j,k} &:= \varphi_{1\xi_1^{i,j,k}}^{A_k}(t_j) - \varphi_{2\xi_1^{i,j,k}}^{A_k}(t_j) \\ \nu^{i,j,k} &:= \varphi_{2\xi_1^{i,j,k}}^{A_k}(t_j) \\ \bar{\mu}^{i,j,k} &:= \bar{u}_{\xi_1^{i,j,k}}^{A_k}(t_j). \end{aligned} \quad (58)$$

The feature set for the Supervised Machine Learning is taken as  $(t_j, \tilde{x}_1^{i,j,k}, A_k)$  and the labels are  $(\nu^{i,j,k}, \bar{\mu}^{i,j,k})$ . Figure 15 shows a orbit transition

$$(-1, 0.5) \rightarrow (0, 0) \rightarrow (0, 1.2) \quad (59)$$

of the target orbit  $(p_0, \dot{p}_0)$  at  $t = 20$  and  $t = 40$ . A constant external force  $d = 20N$  is applied to the cart for  $t \in [69.5, 70]$ .

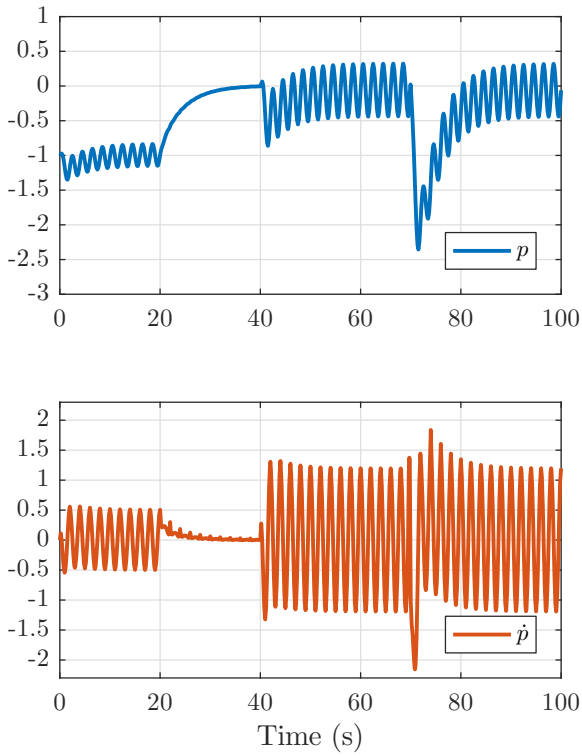
**Remark 6.**

- (i) Orbit transition from set  $B_1$  to a target orbit  $A$  can also be reviewed as rejecting state disturbances in  $B_1$ . The distance from a state in  $B_1$  to  $A$  is not necessarily “small”, indicating the region of attraction for this controller could be “large”.
- (ii) There may exist two orbits  $A_m$  and  $A_n$  in  $B_1$  for which a transition cannot be achieved over  $[0, T_h]$ . However, one may think of transitions in  $B_1$  as a graph so that if there exists an orbit  $A_k$  such that

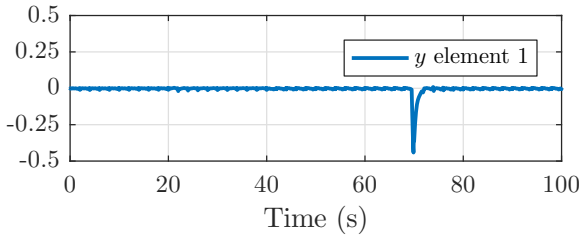
$$A_m \rightarrow A_k \rightarrow A_n$$

is possible, then the orbits are connected.

- (iii) If the target orbit is modified at multiples of  $T_p$ , there are no jumps in  $\nu$ ; this is because the orbit-library insertion map transitions the system from one periodic orbit to another as shown in Figure 16.



**Figure 15.** Plots of  $p(t)$  (top) and  $\dot{p}(t)$  (bottom) as the closed-loop system transitions from one periodic orbit to another, as given in (59), with a disturbance applied for  $t \in [69.5, 70]$ .



**Figure 16.** This shows that there is no jump in the output when the transition point takes place at a multiple of  $T_p$ . The jump corresponds to the disturbance in Figure 15. Only the first component of (25) is shown as the other component is the derivative of this one.

## 5 Hybrid Model and Control

This section describes an extension of the control policy developed in Sect. 3 to systems with impulse effects Grizzle et al. (2014); Westervelt et al. (2007); Bainov and Simeonov (1989), a special class of hybrid models that arises in bipedal robots. The control goals for the hybrid system corresponds to stabilizing periodic walking gaits for various speeds, and to transitioning among these gaits. The robot should also be able to reject a range of force perturbations.

### 5.1 Hybrid Model

We consider a hybrid system with one continuous-time phase as follows

$$\Sigma : \begin{cases} \dot{x} = f(x, u) & x^- \notin \mathcal{S} \\ x^+ = \Delta(x^-) & x^- \in \mathcal{S}, \end{cases} \quad (60)$$

in which  $x \in \mathcal{X}$  and  $\mathcal{X} \subset \mathbb{R}^n$  denote the *vector of state variables* and *n-dimensional state manifold*, respectively. The continuous-time control input is represented by  $u \in \mathcal{U}$ , where  $\mathcal{U} \subset \mathbb{R}^m$  is an open *set of admissible control values*. In addition,  $f : \mathcal{X} \times \mathcal{U} \rightarrow T\mathcal{X}$  is assumed to be continuously differentiable ( $\mathcal{C}^1$ ) so that a Poincaré map can be computed later when checking stability. For each  $u \in \mathcal{U}$ ,  $f(\cdot, u)$  is a vector field in  $T\mathcal{X}$ , the *tangent bundle* of the state manifold  $\mathcal{X}$ .

The *switching hypersurface*  $\mathcal{S}$  is an  $(n - 1)$ -dimensional manifold

$$\mathcal{S} := \{x \in \mathcal{X} \mid p(x) = 0\}, \quad (61)$$

on which the state solutions are allowed to undergo a sudden jump according to the *re-initialization rule*  $x^+ = \Delta(x^-)$ . Here,  $p : \mathcal{X} \rightarrow \mathbb{R}$  is a  $\mathcal{C}^1$ -*switching function* which satisfies  $\frac{\partial p}{\partial x}(x) \neq 0$  for all  $x \in \mathcal{S}$ . Moreover,  $\Delta : \mathcal{X} \rightarrow \mathcal{X}$  denotes the  $\mathcal{C}^1$  *reset map*.  $x^-(t) := \lim_{\tau \nearrow t} x(\tau)$  and  $x^+(t) := \lim_{\tau \searrow t} x(\tau)$  represent the left and right limits of the state trajectory  $x(t)$ , respectively. As in Westervelt et al. (2003), the solution of the hybrid system (60) is assumed to be right continuous. In particular, it is constructed by piecing together the flow of  $\dot{x} = f(x, u)$  such that the discrete transition takes place when this flow intersects the switching hypersurface  $\mathcal{S}$ . The new initial condition for  $\dot{x} = f(x, u)$  is then determined by the reset map  $x^+ = \Delta(x^-)$ .

### 5.2 Setting up the Optimization Problem

We now start the translation of the Assumptions A-1 to A-6 to the hybrid setting for the purpose of designing a feedback controller to locally exponentially stabilize a periodic solution. For bipedal robots, mid-step is a good time to make adjustments to the gait: (a) impact transients have had a chance to settle out; (b) the swing foot is safely away from the ground; and (c) there is still adequate time to steer the swing leg to a favorable configuration for impact. Hence, we will use mid-step of the periodic orbit understudy for setting the beginning and end of the trajectories that we compute via optimization.

The hybrid model (60) is assumed to satisfy the following conditions.

**H-1**  $f : \mathcal{X} \times \mathcal{U} \rightarrow T\mathcal{X}$  and the reset map  $\Delta\mathcal{X} \rightarrow \mathcal{S}$  are Lipschitz continuous. This will allow the stability analysis tools of Ames et al. (2014) to be applied later on.

**H-2** There exists  $T_p > 0$ ,  $x_m^* \in \mathbb{R}^n$ , a piecewise continuous input  $u^* : [0, T_p] \rightarrow \mathcal{U}$  and a solution  $\varphi^*(t)$  of (60) satisfying:

- a)  $\varphi^{*+}(0) = x_m^*$ ;
- b)  $\varphi^{*-}(T_p/2) \in \mathcal{S}$ , (swing foot touches the ground);
- c)  $\forall t \neq T_p/2, \varphi^*(t) \notin \mathcal{S}$ , (does so only once); and
- d)  $\varphi^{*-}(T_p) = x_m^*$  (periodicity).

It is noted that by the definition of  $\mathcal{S}$ , the periodic solution is transversal to  $\mathcal{S}$ , namely  $\frac{d}{dt}p(\varphi^{*-}(T_p/2)) < 0$ . And yes, the motion is being ‘‘clocked’’ with the middle of the step. ■

The point  $x_m^*$  is the midpoint of the periodic trajectory, as measured by time. The controller we build will start from mid-stance, follow the Lagrangian model, undergo impact, and then once again evolve according to the Lagrangian model. To formulate the trajectory designs and the closed-loop system, we need to split the continuous phase of the model (60) into part-(i), after mid-stance, and part-(ii), the first half of the ensuing stance phase.

$$\Sigma_i : \begin{cases} \dot{\tau} = 1, \\ \dot{x} = f(x, u), & x^- \notin \mathcal{S} \\ \tau^+ = \tau^- \\ x^+ = \Delta(x^-), & x^- \in \mathcal{S} \end{cases} \quad (62)$$

$$\Sigma_{ii} : \begin{cases} \dot{\tau} = 1, & \tau^- < T_p \\ \dot{x} = f(x, u), \\ \tau^+ = 0 & \tau^- = T_p \\ x^+ = x^- \end{cases}$$

The guard condition on the phase- $i$  depends only on the state  $x$ , whereas the guard condition on the phase- $ii$  depends only on ‘‘time’’ as measured by  $\tau$ .

**H-3** The user has selected an open ball  $B \subset \mathbb{R}^n$  about  $x_m^*$ , a positive-definite, locally Lipschitz-continuous function  $V : B \rightarrow \mathbb{R}$ , and constants  $0 < \alpha_1 \leq \alpha_2$  such that,  $V(x_m^*) = 0$  and  $\forall x \in B$ ,

$$\alpha_1(x - x_m^*)^\top(x - x_m^*) \leq V(x) \leq \alpha_2(x - x_m^*)^\top(x - x_m^*).$$

**H-4** There is a constant  $0 \leq c < 1$ , such that, for each initial condition  $\xi \in B$ , there exists a piecewise continuous input  $u_\xi : [0, T_p] \rightarrow \mathbb{R}^m$  and a corresponding solution  $\varphi_\xi : [0, T_p] \rightarrow \mathbb{R}^n$  of the hybrid model (62) satisfying

- a)  $\varphi_\xi^+(0) = \xi$ ,
- b)  $\varphi_\xi^-(T_p/2) \in \mathcal{S}$ ,
- c)  $\forall t \neq T_p/2, \varphi_\xi(t) \notin \mathcal{S}$ ,
- d)  $\varphi_\xi^+(T_p) \in B$  and there is exponential convergence toward the periodic orbit, namely,

$$V(\varphi_\xi^+(T_p)) \leq cV(\xi), \quad (63)$$

and

- e)  $\xi = x_m^* \Rightarrow u_\xi = u^*$ .

**Proposition 3.** Assume the open-loop hybrid model (60) satisfies Assumptions H-1 to H-4. Assume in addition there exist open sets  $B_i^e$  and  $B_{ii}^e$  that contain  $B$ , a  $\delta > 0$ , and two feedbacks

$$\mu_i : [0, T_p/2 + \delta] \times B_i^e \rightarrow \mathbb{R}^m$$

$$\mu_{ii} : [T_p/2 - \delta, T_p] \times B_{ii}^e \rightarrow \mathbb{R}^m$$

that are piecewise continuous in  $t$ , locally Lipschitz continuous in  $x$ , and, such that, for  $0 \leq t < T_p$  and  $\xi \in B$ ,

$$\mu_i(t, \varphi_\xi(t)) = u_\xi(t), \quad 0 \leq t < T_p/2$$

$$\mu_{ii}(t, \varphi_\xi(t)) = u_\xi(t), \quad T_p/2 \leq t < T_p. \quad (64)$$

Then  $\varphi^* : [0, T_p] \rightarrow \mathbb{R}^n$  is a locally exponentially stable periodic solution of the closed-loop system

$$\Sigma_i^{cu} : \begin{cases} \dot{\tau} = 1, \\ \dot{x} = f(x, \mu_i(\tau, x)), & x^- \notin \mathcal{S} \\ \tau^+ = \tau^- \\ x^+ = \Delta(x^-), & x^- \in \mathcal{S} \end{cases} \quad (65)$$

$$\Sigma_{ii}^{cu} : \begin{cases} \dot{\tau} = 1, & \tau^- < T_p \\ \dot{x} = f(x, \mu_{ii}(\tau, x)), \\ \tau^+ = 0 & \tau^- = T_p \\ x^+ = x^- \end{cases}$$



### 5.3 Generalized Hybrid Zero Dynamics

Following Appendix-B, assume now that the continuous phase of the hybrid model has been decomposed as

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, u) \\ \dot{x}_2 &= f_2(x_1, x_2, u), \end{aligned} \quad (66)$$

with

$$x_2 = \begin{bmatrix} x_{2a} \\ x_{2b} \end{bmatrix} \text{ and } f_2 = \begin{bmatrix} x_{2b} \\ u \end{bmatrix}.$$

Let  $\gamma : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_2}$  be a locally Lipschitz continuous insertion function that preserves the periodic orbit, namely, writing  $x_m^* =: (x_{1m}^*; x_{2m}^*)$ , it follows that  $\gamma(x_{1m}^*) = x_{2m}^*$ .

**H-5** There is a constant  $0 \leq c < 1$ , such that, for each initial condition  $\xi = (\xi_1, \gamma(\xi_1)) \in B$ , there exists a continuous input  $u_{\xi_1} : [0, T_p] \rightarrow \mathbb{R}^m$  and a corresponding solution  $\varphi_{\xi_1} : [0, T_p] \rightarrow \mathbb{R}^n$  of the hybrid model (62) satisfying,

- a)  $\varphi_{\xi_1}^+(0) = (\xi_1; \gamma(\xi_1))$ ,
- b)  $\varphi_{\xi_1}^-(T_p/2) \in \mathcal{S}$ ,
- c)  $\forall t \neq T_p/2, \varphi_{\xi_1}(t) \notin \mathcal{S}$ ,
- d)  $\varphi_{\xi_1}^+(T_p) \in B$  and there is exponential convergence toward the periodic orbit, namely,

$$V((\varphi_{1\xi_1}^+(T_p), \gamma(\varphi_{1\xi_1}^+(T_p))) \leq cV(\xi_1, \gamma(\xi_1)), \quad (67)$$

and

- e)  $(\xi_1, \gamma(\xi_1)) = x_m^* \Rightarrow u_{\xi_1} = u^*$ .

where a solution of the  $(n_1 + n_2)$ -dimensional model (66) has been decomposed as  $\varphi_{\xi_1}(t) =: (\varphi_{1\xi_1}(t), \varphi_{2\xi_1}(t))$ . ■

The following result generalizes the hybrid zero dynamics defined in Westervelt et al. (2003); Morris and Grizzle (2009); Westervelt et al. (2007). Even in the case of one degree of underactuation, one is able to achieve exponential stability with this method for gaits that could not be rendered stable with the previous formulation of virtual constraints. See Appendix D.2 for an example. More important that this fact, however, the new formulation allows a systematic approach to robot models with more than one degree of underactuation. This is illustrated in Sect. 6.

**Proposition 4.** Assume the open-loop hybrid system (60) with  $f$  given by (66) satisfies Hypotheses H-1 to H-3 and H-5, and define  $B_1 := \{\xi_1 \in \mathbb{R}^{n_1} \mid (\xi_1, \gamma(\xi_1)) \in B\}$ . Assume in addition there exist open sets  $B_{1,i}^e$  and  $B_{1,ii}^e$  that contain

$B_1$ , a  $\delta > 0$ , and two feedbacks

$$\begin{aligned} \nu_i &: [0, T_p/2 + \delta] \times B_{1,i}^e \rightarrow \mathbb{R}^{n_1} \\ \nu_{ii} &: [T_p/2 - \delta, T_p] \times B_{1,ii}^e \rightarrow \mathbb{R}^{n_1} \end{aligned}$$

and

$$\begin{aligned} \mu_i &: [0, T_p/2 + \delta] \times B_{1,i}^e \rightarrow \mathbb{R}^m \\ \mu_{ii} &: [T_p/2 - \delta, T_p] \times B_{1,ii}^e \rightarrow \mathbb{R}^m \end{aligned}$$

that are piecewise continuous in  $t$ , locally Lipschitz continuous in  $x$ , and, such that, for  $0 \leq t < T_p$  and  $\xi \in B$ ,

$$\begin{aligned} \nu_i(t, \varphi_{1\xi_1}(t)) &= \varphi_{2\xi_1}(t), \quad 0 \leq t < T_p/2 \\ \nu_{ii}(t, \varphi_{1\xi_1}(t)) &= \varphi_{2\xi_1}(t), \quad T_p/2 \leq t < T_p \end{aligned} \quad (68)$$

and

$$\begin{aligned} \mu_i(t, \varphi_{1\xi_1}(t)) &= u_{\xi_1}(t), \quad 0 \leq t < T_p/2 \\ \mu_{ii}(t, \varphi_{1\xi_1}(t)) &= u_{\xi_1}(t), \quad T_p/2 \leq t < T_p. \end{aligned} \quad (69)$$

Then  $x_1^* : [0, T_p] \rightarrow \mathbb{R}^{n_1}$  is a locally exponentially stable periodic solution of the reduced-order hybrid system

$$\Sigma_i : \begin{cases} \dot{\tau} = 1, \\ \dot{x}_1 = f_1(x_1, \nu_i(\tau, x_1), \mu_i(\tau, x_1)), \\ \text{when } \begin{bmatrix} x_1^- \\ \nu_i(\tau^-, x_1^-) \end{bmatrix} \notin \mathcal{S} \\ \tau^+ = \tau^-, \\ x_1^+ = \Delta_1(x_1^-, \nu_i(\tau^-, x_1^-)), \\ \text{when } \begin{bmatrix} x_1^- \\ \nu_i(\tau^-, x_1^-) \end{bmatrix} \in \mathcal{S} \end{cases} \quad (70)$$

$$\Sigma_{ii} : \begin{cases} \dot{\tau} = 1, & \tau^- < T_p \\ \dot{x}_1 = f(x, \nu_{ii}(\tau, x_1), \mu_{ii}(\tau, x_1)), \\ \tau^+ = 0, & \tau^- = T_p \\ x^+ = x^- \end{cases}$$

**Remark 7.**

(i) In principle,  $\tau^* : [0, T_p] \rightarrow \mathbb{R}$  needs to be defined to complete the periodic orbit, but clearly, the trivial solution,  $\tau^*(t) = t$ , is the only possibility.

(ii) As in the non-hybrid case, using the trajectories in H-5, define

$$\Psi_t : B_1 \rightarrow \mathbb{R}^n, \quad \Psi_t(\xi_1) := \begin{bmatrix} \varphi_{1\xi_1}(t) \\ \varphi_{2\xi_1}(t) \end{bmatrix} \quad (71)$$

and  $\Psi_e : [0, T_p) \times B_1 \rightarrow \mathbb{R}^{n+1}$  by

$$\Psi_e(\tau, \xi_1) := \begin{bmatrix} \tau \\ \Psi_\tau(\xi_1) \end{bmatrix}. \quad (72)$$

By H-5-b),  $\forall \xi_1 \in B_1$ ,  $\Psi_e^-(T_p/2, \xi_1) \in \mathcal{S}$ . Hence, the loss of dimension is in the  $\tau$ -component, and therefore

$$\dim(\Psi_e^-(T_p/2, B_1) \cap \mathcal{S}) = \dim(B_1).$$

(iii) The Generalized Hybrid Zero Dynamics Manifold (G-HZD) is therefore<sup>3</sup>

$$Z_e := \Psi_e([0, T_p), B_1), \quad (73)$$

which has two components,

$$Z_{e,i} := \Psi_e([0, T_p/2), B_1)$$

and

$$Z_{e,ii} := \Psi_e([T_p/2, T_p), \Psi_e^-(T_p/2, B_1) \cap \mathcal{S}).$$

(iv) The corresponding restriction dynamics is given by (70), which is then the G-HZD.

## 5.4 Stabilizing the Original Model

We can now obtain and explain the controller we use on bipeds. Similar to Sect. 3.5, assume the continuous phase of the hybrid model has the form

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, u_1) \\ \dot{x}_2 &= f_2(x_1, x_2, u_2), \end{aligned} \quad (74)$$

with

$$x_2 = \begin{bmatrix} x_{2a} \\ x_{2b} \end{bmatrix} \text{ and } f_2 = \begin{bmatrix} x_{2b} \\ u_2 \end{bmatrix},$$

and  $u = (u_1, u_2)$ . The reason to split the input and not allow the  $u_2$ -component to enter the  $x_1$ -dynamics will be clear shortly. We allow the  $u_1$ -component to be empty.

**H-6** The solutions in H-5 also satisfy

$$\gamma(\varphi_{1\xi_1}(T_p)) = \varphi_{2\xi_1}(T_p). \quad (75)$$

**Theorem 2.** Assume the open-loop hybrid system (60) with  $f$  given by (74) satisfies Hypotheses H-1 to H-3, H-5 and H-6. Define  $B_1 := \{\xi_1 \in \mathbb{R}^{n_1} \mid (\xi_1, \gamma(\xi_1)) \in B\}$ . Assume in addition there exist open sets  $B_{1,i}^e$  and  $B_{1,ii}^e$  that contain  $B_1$ ,

a  $\delta > 0$ , and feedbacks

$$\begin{aligned} \nu_i &: [0, T_p/2 + \delta] \times B_{1,i}^e \rightarrow \mathbb{R}^{n_1} \\ \nu_{ii} &: [T_p/2 - \delta, T_p] \times B_{1,ii}^e \rightarrow \mathbb{R}^{n_1} \end{aligned}$$

and

$$\begin{aligned} \mu_i &: [0, T_p/2 + \delta] \times B_{1,i}^e \rightarrow \mathbb{R}^m \\ \mu_{ii} &: [T_p/2 - \delta, T_p] \times B_{1,ii}^e \rightarrow \mathbb{R}^m \end{aligned}$$

that are piecewise continuous in  $t$ , locally Lipschitz continuous in  $x$ , and, such that, for  $0 \leq t < T_p$  and  $\xi_1 \in B_1$ ,

$$\begin{aligned} \nu_i(t, \varphi_{1\xi_1}(t)) &= \varphi_{2\xi_1}(t), \quad 0 \leq t < T_p/2 \\ \nu_{ii}(t, \varphi_{1\xi_1}(t)) &= \varphi_{2\xi_1}(t), \quad T_p/2 \leq t < T_p \end{aligned} \quad (76)$$

and

$$\begin{aligned} \mu_i(t, \varphi_{1\xi_1}(t)) &= u_{\xi_1}(t), \quad 0 \leq t < T_p/2 \\ \mu_{ii}(t, \varphi_{1\xi_1}(t)) &= u_{\xi_1}(t), \quad T_p/2 \leq t < T_p. \end{aligned} \quad (77)$$

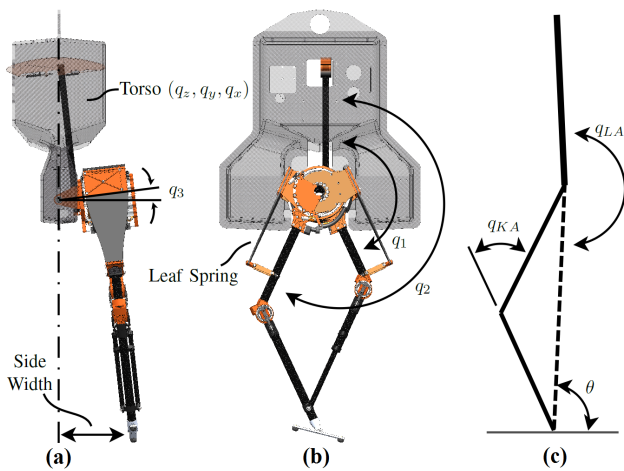
Then for all  $\frac{n_2}{2} \times \frac{n_2}{2}$  positive definite matrices  $K_p$  and  $K_d$ ,  $\exists \epsilon^* > 0$ , such that  $\forall 0 < \epsilon \leq \epsilon^*$ ,  $x^* : [0, T_p] \rightarrow \mathbb{R}^{n_1+n_2}$  is a locally exponentially stable periodic solution of the closed-loop hybrid system

$$\Sigma_i : \begin{cases} \dot{\tau} = 1 & x^- \notin \mathcal{S} \\ \dot{x} = f(x_1, x_2, u_1, u_2) \\ u_1 = \mu_{1i}(\tau, x_1) \\ u_2 = \mu_{2i}(\tau, x_1) - \left[ \frac{K_p}{\epsilon^2} \frac{K_d}{\epsilon} \right] (x_2 - \nu_i(\tau, x_1)) \\ \tau^+ = \tau^- \\ x^+ = \Delta(x^-), & x^- \in \mathcal{S} \end{cases}$$

$$\Sigma_{ii} : \begin{cases} \dot{\tau} = 1 & \tau^- < T_p \\ \dot{x} = f(x_1, x_2, u_1, u_2) \\ u_1 = \mu_{1ii}(\tau, x_1) \\ u_2 = \mu_{2ii}(\tau, x_1) - \left[ \frac{K_p}{\epsilon^2} \frac{K_d}{\epsilon} \right] (x_2 - \nu_{ii}(\tau, x_1)) \\ \tau^+ = 0 & \tau^- = T_p \\ x^+ = x^-. \end{cases} \quad (78)$$

Moreover, the closed-loop system possesses a G-HZD and it is given by (70).

■ **Remark 8.** The control was split so that the high-gain part of the feedback does not directly enter the states of the zero dynamics, namely  $x_1$ . This allows the system to conform with existing theorems Ames et al. (2014) for establishing the exponential stability of the periodic orbit — in the full-order hybrid model — on the basis of its stability in the



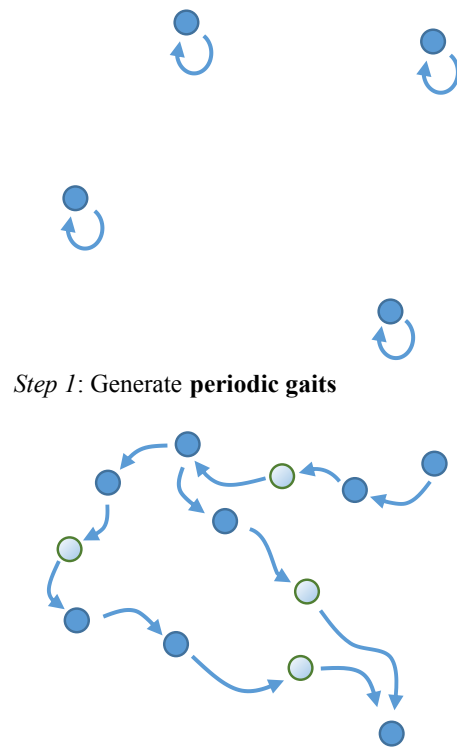
**Figure 17.** Based on (Da et al. 2017, Figure 2). Coordinate representation for MARLO, the Michigan copy of an Atrias-series robot.

zero dynamics, (70). Isolating the action of the high-gain controller to the  $x_2$ -dynamics was not necessary in the case of ODEs.

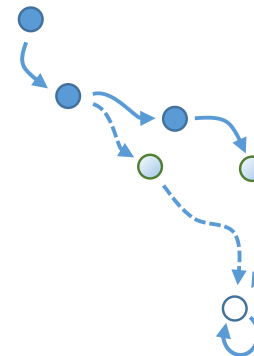
## 6 Bipedal Walking Gaits

This section applies the Generalized Hybrid Zero Dynamics (G-HZD) developed in Section 5 to a bipedal robot, namely, the University of Michigan copy of an ATRIAS-series 3D robot that we call MARLO Ramezani et al. (2014); see Figure 17. As shown in Buss et al. (2016); Griffin and Grizzle (2016); Da et al. (2017); Hartley et al. (2017), the robot is capable of walking forward and backward at various speeds and over challenging terrain, both indoors and outdoors. The robot can step in place, but it cannot stand in place because its feet are passive. In the experiments, shoes are placed over the passive feet to prevent excessive yawing about the point of contact. The robot's hips have 2 DoF (pitch and roll). Because the hips lack yaw motion, turning is not one of the robot's strengths!

The control laws developed here will illustrate stepping in place, walking forward and backward, and transitions among such gaits. The work illustrates a theoretically sound method for gait design that unifies and significantly extends many of our previous results. The first control designs will rely on the optimization package in Jones (2014), which can only handle a planar model of the robot. For these designs, lateral stability is achieved via a heuristic foot placement policy. Since March of 2017, we have had access to the trajectory optimization package Hereid et al. (2016a), which easily handles the full 3D model of the robot. Figure 18 summarizes our control design process.



*Step 2:* Design **transient gaits** between periodic gaits; here, we are showing three-step transitions. Similar to MPC, only the first step of the transition is retained for the learning set. From this set of trajectories, **Supervised Machine Learning** is used to extract the controller



*Step 3:* The hollow dot is the target periodic gait. It can be reached in three steps, though the **learned controller** re-plans at each step to reach the target in three additional steps, leading to exponential convergence.

**Figure 18.** This can be thought of as an alternative representation of Figure 1 when the surface  $Z_0$  is built from periodic solutions of the full-order model. The light dots represent transient trajectories while the other dots (solid or hollow) are periodic orbits.

### 6.1 MARLO

The robot is described in detail in Ramezani et al. (2014). For the planar model, the configuration variables are joint angles and one absolute coordinate. The angle  $\theta$ , the absolute stance leg angle, is unactuated because the feet are passive. Hence, we define

$$x_1 = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}.$$

For the purposes of controller design, the regulated quantities are  $q_a = (q_x, q_{sw,LA}, q_{st,KA}, q_{sw,KA})$ , that is, the torso, stance knee, swing leg, and swing knee angle, and hence

$$x_2 = \begin{bmatrix} q_a \\ \dot{q}_a \end{bmatrix}.$$

In the simulation and control design, we constrain the stance foot to remain in contact with the ground with no foot slip. In the experiments, we estimate the ground reaction forces through the deflection of the leaf springs to decide whether to control the torso angle or the stance leg angle  $q_{st,LA}$ , as in [Rezazadeh et al. \(2015\)](#). The model decomposition is done as in [Appendix B](#).

## 6.2 Design of Planar Periodic Gaits and Transition Trajectories via Optimization

For robots, an orbit library is called a gait library. We first design a gait library

$$\mathcal{L} := \{\bar{v} \mid -0.8 \leq \bar{v} \leq 0.8\} \quad (79)$$

consisting of periodic gaits for various average walking speeds satisfying [H-2](#). In this example, we reuse the gaits described in [Da et al. \(2017\)](#), where each gait has period  $T_p = 0.4$ , and the cost function is

$$J = \int_{-T_p/2}^{T_p/2} \|u(t)\|^2 dt. \quad (80)$$

Denote the trajectory of the periodic gait and the corresponding input as  $\varphi^{\bar{v}}(t)$  and  $u^{\bar{v}}(t)$ , respectively, and the midpoint of the periodic trajectory as  $x_m^{\bar{v}}$ . The insertion function is built from the gait library and is denoted  $\gamma_{\mathcal{L}}$ .

For a given periodic orbit in  $\mathcal{L}$ , we define

$$B_1^{\bar{v}} = \{\xi_1 := (\theta, \dot{\theta}) \mid -\frac{\pi}{12} \leq |\theta - \theta_m^{\bar{v}}| \leq \frac{\pi}{12}, \\ -0.8 \leq |\dot{\theta} - \dot{\theta}_m^{\bar{v}}| \leq 0.8\}, \quad (81)$$

a sliding window<sup>4</sup> about the target speed  $\bar{v}$ . For  $\xi_1 \in B_1^{\bar{v}}$ , trajectories are generated as in [H-5](#) using optimization with cost function

$$J(\xi_1, \bar{v}) = \sum_{k=1}^6 \int_{(k-1)\frac{T_p}{2}}^{k\frac{T_p}{2}} (\|x - \varphi^{\bar{v}}\|_Q^2 + \|u - u^{\bar{v}}\|_R^2) dt \quad (82)$$

for a horizon of length  $T_h = 3T_p$ . The optimization is performed subject to the hybrid dynamics describing MARLO, the physical constraints shown in [Table 4](#), and the terminal constraint  $x(3T_p) = \varphi^{\bar{v}}(T_p)$ . So that the trajectories

**Table 4.** Optimization constraints

Motor Torque $ u $	$< 5 \text{ Nm}$
Step Duration $T$	$= 0.4 \text{ s}$
Friction Cone $\mu$	$< 0.6$
Impact Impulse $Fe$	$< 15 \text{ Ns}$
Vertical Ground Reaction Force	$> 300 \text{ N}$
Mid-step Swing Foot Clearance	$> 0.15 \text{ m}$

can be used to stabilize the full-order model, the boundary constraints in [H-6](#) is also imposed. The resulting trajectory and input for the given  $\bar{v}$  and selected  $\xi_1$  are denoted  $\varphi_{\xi_1}^{\bar{v}}$  and  $u_{\xi_1}^{\bar{v}}(t)$ , respectively.

**Remark 9.** The initial condition set  $B_1^{\bar{v}}$  is related to the notion of 3-step capture region defined in [Koolen et al. \(2012\)](#); [Zaytsev et al. \(2015\)](#). In our experience, three-steps is a reasonable balance between planning horizon and computational burden.

## 6.3 Controller Design via Machine Learning

The gait library [\(79\)](#) is assumed to be discretized by 5 evenly spaced average speeds  $\bar{v}_k$ , each  $\xi_1^i := (\theta^i, \dot{\theta}^i)$  is drawn from a uniformly spaced grid of 25 points, and time interval  $[0, T_p]$  is evenly sampled into 21 points,  $t_j$ . The combined training and validation data set is therefore denoted by

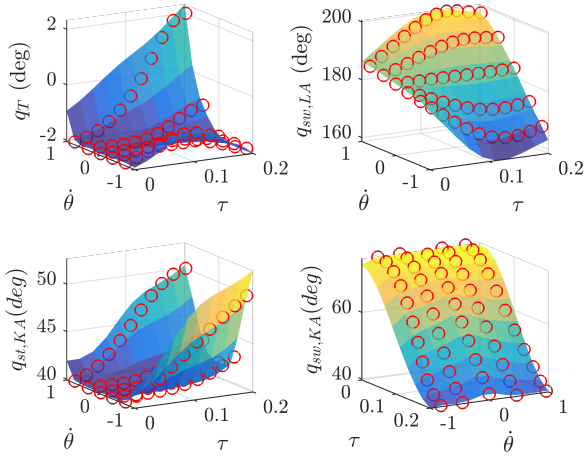
$$\begin{aligned} x_1^{i,j,k} &:= \varphi_{1\xi_1^i}^{\bar{v}_k}(t_j) \\ \nu^{i,j,k} &:= \varphi_{2\xi_1^i}^{\bar{v}_k}(t_j) \\ \mu^{i,j,k} &:= u_{\xi_1^i}^{\bar{v}_k}(t_j). \end{aligned} \quad (83)$$

Any infeasible optimization problems<sup>5</sup> are removed from the data set before processing it by Supervised Machine Learning.

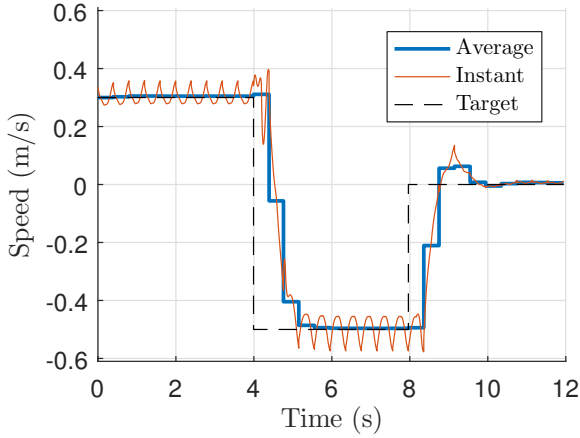
We next learn the functions in [Theorem 2](#). The features are  $(t_j, x_1^{i,j,k}, \bar{v}_k)$  and the labels are  $(\nu^{i,j,k}, \mu^{i,j,k})$  and the data base is split at  $t = T_p/2$  so that the functions

$$\begin{aligned} \nu_1(t, x_1, \bar{v}) \\ \nu_{ii}(t, x_1, \bar{v}) \\ \mu_i(t, x_1, \bar{v}) \\ \mu_{ii}(t, x_1, \bar{v}) \end{aligned}$$

are learned individually. Part of the fitting is shown in [Figure 19](#). These functions are enough to construct the G-HZD in [\(70\)](#). To complete the control design as in [\(78\)](#), the feedback gains  $K_p$ ,  $K_d$  and  $\epsilon$  must be selected. While in principle these last gains may have to vary with  $\bar{v}$ , for MARLO, we have found that a single set of gains<sup>6</sup> suffices.



**Figure 19.** The fitting results of the Supervised Machine Learning. The features  $\theta = \theta_m^{\bar{v}}$  and  $\bar{v} = 0$  are fixed at constant values, while  $\tau \in [0, 0.2]$  is from mid-step to ground contact. The plots show four of the components in  $\nu_i(t, x_1, \bar{v})$ .



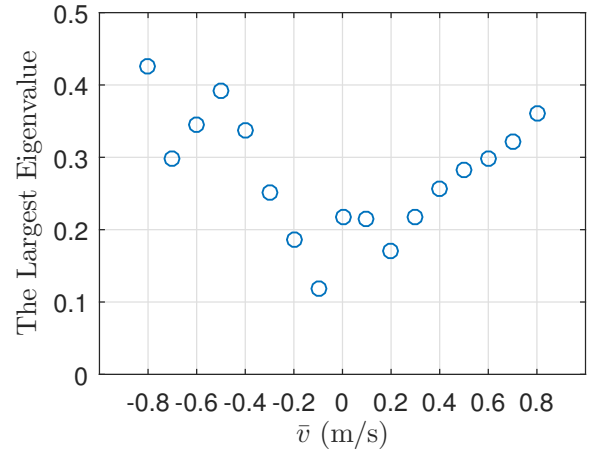
**Figure 20.** The target speed  $\bar{v}$  changes from 0.3 m/s to  $-0.5$  m/s and to 0 m/s. The gait transition takes less than five steps to reach the target speed. The error between target and average speed is small.

The transition among different target speeds are shown in Figure 20.

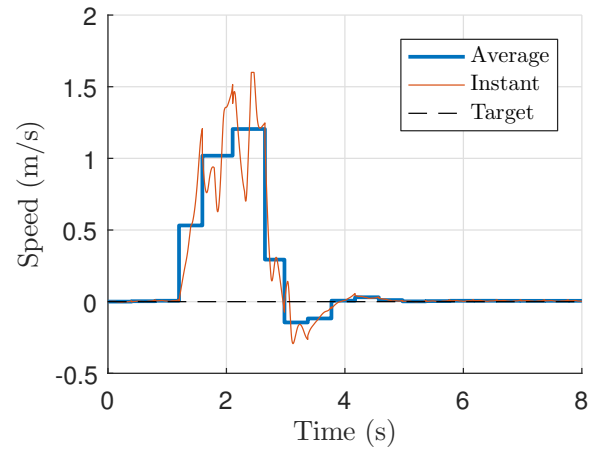
## 6.4 Example Performance Analysis

**6.4.1 Stability Analysis** We know that the periodic orbits in the full-order model should be locally exponentially stable by the results in Sect. 5. We formally verify this by numerically evaluating the Jacobian of the Poincaré map for twenty evenly-spaced points in the interval  $-0.8 \leq \bar{v} \leq 0.8$ . The magnitude of the largest eigenvalue is shown in Figure 21, which proves local exponential stability for each fixed target speed  $\bar{v}$ . Note that only five of these points were in the training data. The learned feedback functions have provided stable gaits for a continuum of target speeds.

The stability of the overall closed-loop system is further illustrated by applying force perturbations, which is a more “realistic” test. We apply a longitudinal force on the hip at



**Figure 21.** The largest eigenvalue of the Poincaré map is given for some target speeds. This indicates the target speed in the gait library (79) is exponentially stable.

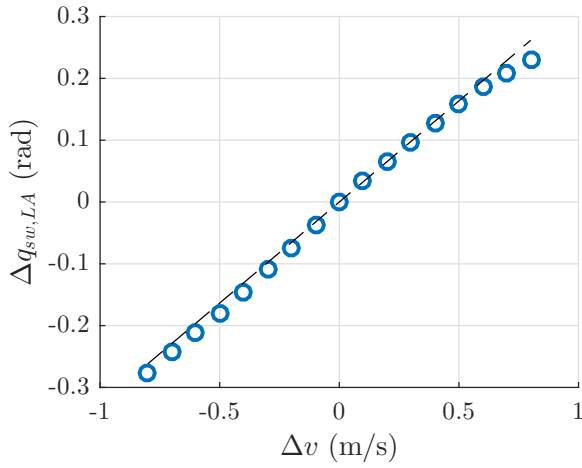


**Figure 22.** A perturbation is applied from 1.2 seconds to 2 seconds in a magnitude of 150 N. The maximum speed is larger than the maximum speed (0.8 m/s) in the training set. The extrapolation may be the cause of the speed oscillation. Even though, the speed convergence near the target speed in less than five steps.

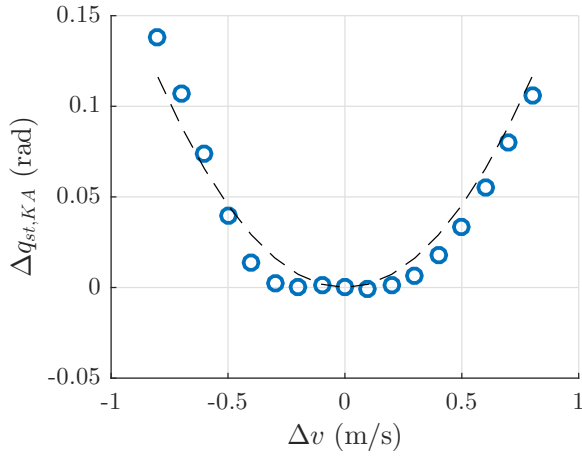
1.2 seconds for 0.8 second (i.e.,  $2T_p$ ) and examine the time to recover the nominal gait. For the stepping-in-place gait  $\bar{v} = 0$ , the largest force from which the robot can recover without violating the physical constraints is 150 N. Figure 22 shows the resulting longitudinal velocity of the robot. The peak speed is approximately 1.5 m/s, which is beyond the maximum training speed of 0.8 m/s. The speed is once again less than 0.05 m/s within five steps. The convergence rate is relatively fast given that the optimizer uses a horizon of three steps.

**6.4.2 Interpretation of the posture changes employed by the controller** We now provide some physical intuition for how the controller coordinates the links to achieve stability. In fact, we evaluate  $\nu(T_p/2, x_1)$  with  $x_1 = (\pi/12, \dot{\theta})$ ,  $-0.8 \leq \dot{\theta} \leq 0.8$ . Figures 23 and 24 show the changes in the swing leg angle and the stance knee angle at touchdown. The swing





**Figure 23.** Change in swing leg angle vs change in velocity. One part of the learned-optimal strategy is a standard linear leg-angle adjustment policy as in Pratt and Tedrake (2006); Dunn and Howe (1996).



**Figure 24.** Change in stance knee angle vs change in velocity. This is not part of the standard recommendations in Pratt and Tedrake (2006); Dunn and Howe (1996).

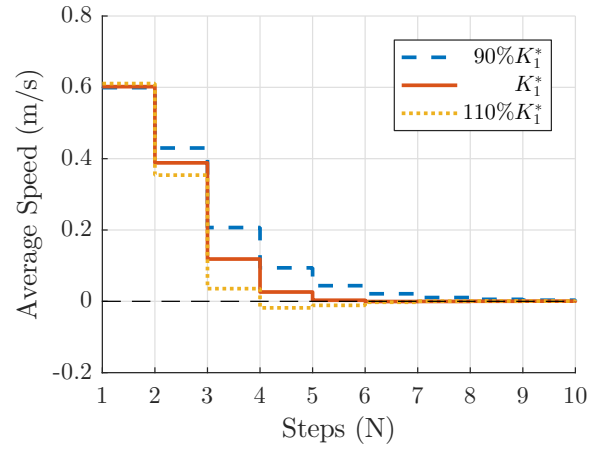
leg is seen to obey an approximately linear relationship with respect to velocity, just as in the foot placement controllers in Pratt and Tedrake (2006); Dunn and Howe (1996) designed on the basis of an inverted pendulum model or a linear inverted pendulum model, viz

$$\Delta q_{sw,LA} = K_1 \Delta v,$$

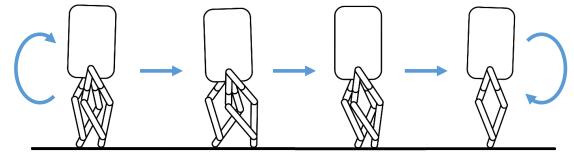
and the scalar  $K_1$  is constant. Denote the regressed linear fit in Figure 23 by  $K_1^*$ . We add  $\pm 10\%$  to  $K_1^*$  and compare the resulting foot placement strategies in Figure 25. It is seen that with the smaller gain the velocity takes longer to settle whereas with the larger gain, there is overshoot.

The learned controller is more than just providing foot placement. Figure 24 also shows a quadratic relationship for knee angle versus velocity just before touchdown, viz

$$\Delta q_{st,KA} = K_2 (\Delta v)^2.$$



**Figure 25.** When the effective linear policy from the learned-optimal strategy is modified by  $\pm 10\%$ , the convergence to the nominal speed of zero either slows down or overshoots.



**Figure 26.** Stick figure showing the coordinated action of torso angle, knee bend, and leg angle provided by the learned-three-step optimization.

As the velocity moves away from zero in either direction, the stance knee angle increases. Perhaps this is to lower the center of mass and to make it easier for the swing foot to touch the ground. Furthermore, in addition to changing the swing leg angle, the learned controller also straightens the swing knee angle, thereby extending the foot further out. Finally, it also leans the torso backward, keeping the center of mass over the stance toe, as shown in Figure 26. These adjustments are all coordinated by the optimization and automatically extracted from the transition trajectories by the supervised learning. Unlike a classical foot placement controller that adjusts only swing leg angle, the learned controller uses the many degrees of freedom of the robot to achieve better performance.

## 7 Experiments on a 3D Bipedal Robot

This section extends the learning controller of the last section to the full-3D model of MARLO. Hence, both the sagittal and lateral planes of the robot are included, while yaw rotations are assumed to be small due to the foot. This control design will allow the physical 3D-robot to walk and step in place. The 3D-controller design will mostly follow the process of the planar example that was illustrated through simulations, though some modifications have been made during the experimental implementation to deal with model

uncertainty, impact model uncertainty, and signal noise; these will be clearly explained and justified.

### 7.1 3D-MARLO Configuration

We still use Figure 17 to describe the generalized coordinates on the 3D robot. Let  $(p_x, p_y)$  denote the sagittal and lateral position of the center of mass, and be  $(v_x, v_y)$  be the corresponding velocities. We define

$$x_1 = \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \end{bmatrix}$$

as the “weakly actuated” states. The regulated angles are  $q_a = (q_y, q_x, q_{sw,3}, q_{sw,LA}, q_{st,KA}, q_{sw,KA})$ , that is, the torso roll and pitch, swing hip, swing leg, stance knee and swing knee angles, and hence the “strongly actuated” states are

$$x_2 = \begin{bmatrix} q_a \\ \dot{q}_a \end{bmatrix}.$$

We note that the coordinates  $(x_1, x_2)$  describe the robot dynamics in (66), or (94) after a coordinate change.

### 7.2 Optimization

We first use optimization, with a cost function as in (80), to build a periodic gait library

$$\mathcal{L} := \{(\bar{v}_x, \bar{v}_y) \mid -0.6 \leq \bar{v}_x \leq 0.6, -0.4 \leq \bar{v}_y \leq 0.4\}, \quad (84)$$

as a grid in two-dimensional Cartesian space; i.e., longitudinal and lateral speed of the robot. The gaits are designed, without loss of generality, such that the associated 2D-Cartesian positions  $(p_x, p_y)$  are equal to the origin at a nominal point in each of the gaits. The insertion function associated to the gait library is constructed using linear regression as in the pendulum model (56) and in the planar biped examples; specifically,

$$x_2 = \gamma_{\mathcal{L}}(x_1) = a_0 + a_1 x_1, \quad (85)$$

where  $a_0$  and  $a_1$  are constant vectors. A linear fitting is good enough for 3D MARLO. While one could do a more sophisticated fit, the maximum root-mean-square-error (RMSE) for all joints is less than 1 degree, and for all joint velocities it is less than 4 degrees per second, even with the linear fit. In part, this is a benefit of using the middle of the gait for building the controller.

The next step is to design transition trajectories from periodic orbits in the library  $\mathcal{L}$  to a target periodic orbit. In this paper, we will only illustrate the target orbit as the stepping-in-place gait, that is,  $(\bar{v}_x^*, \bar{v}_y^*) = 0$ . Further details are not given because they follow the planar example of the last section. One can also refer to work in Da et al. (2017) for the design of transition gaits for different ground slopes.

We perform three-step trajectory optimizations as in (82) and denote the collection of transition trajectories to the target speed of zero as  $\varphi_{\mathcal{L} \rightarrow 0}$ . The orbit library is evenly sampled per  $\bar{v}_x = \{-0.6, -0.4, -0.2, \dots, 0.6\}$  and  $\bar{v}_y = \{-0.4, -0.3, -0.2, \dots, 0.4\}$ , so that the total number of transition trajectories  $\varphi^i_{\mathcal{L} \rightarrow 0}$  is 63. The time interval  $[0, 0.4]$  is evenly sampled into 21 points,  $t_j$ .

### 7.3 Machine Learning

For the 3D robot, we illustrate a different philosophy in building the feature set. Recall that in the planar examples, we included time and all of  $x_1$ -coordinates in the feature vector. Here, we will select the feature vector as simply time and the velocity coordinates, namely,

$$(t, v_x, v_y),$$

and leave out the Cartesian positions  $(p_x, p_y)$ . There are several reasons for this:

1. The impact map in the hybrid model (62) resets the Cartesian position variables to a constant, assumed to the origin. Hence, these variables do not need to be stabilized.
2. On the robot MARLO, we are placing the torso sagittal and roll angles in the  $x_2$ -coordinates, and hence if these are kept upright, the position of the center of mass does not provide significant additional information.
3. It keeps the dimension of the feature set as low as possible, which allows a smaller training data set.

On 3D MARLO, the labels are taken as

$$\nu = \varphi_{2, \mathcal{L} \rightarrow 0},$$

which represents the  $x_2$ -coordinates of the optimized trajectories. The control input  $\mu$  is not learned because, as in previous experiments Griffin and Grizzle (2016); Da et al. (2017) on this robot, we use PD control

$$u_{PD} = [K_p, K_d] (x_2 - \nu(t, v_x, v_y)) \quad (86)$$

to regulate the joint angles, without a feedforward term. The feedforward torque is not applied because of uncertainty in the model. Specifically, the model does not include the motor drive friction, which consumes about 20% of the torque in nominal operation (stepping in place and walking), nor does the model include backlash or compliance in the harmonic drives. Finally, the leaf springs in Figure 17 are excluded from the model; they deflect about 5 degrees when supporting the robot's weight.

Since the impact happens at  $T_p/2 = 0.2$ , the functions

$$\begin{aligned} v_i(t, v_x, v_y) \\ v_{ii}(t, v_x, v_y) \end{aligned}$$

are learned individually.

#### 7.4 Experimental Implementation

Another difference between the model and the physical robot occurs in the velocity signal. Due to spring deflection, impacts, and joint compliance, the estimated Cartesian velocities  $(v_x, v_y)$  are noisy even if each of the individual joint angular velocity signals is relatively “clean”. We thus use a strong<sup>7</sup> first-order filter to clean up the Cartesian velocity signals  $(v_x, v_y)$  appearing in  $\nu(t, v_x, v_y)$ , the reference for low-level PD controllers (86).

The filtered signal, shown in Figure 27, is relatively clean, but causes phase lag. Moreover, the energy loss at impact is less on the robot than in the design model because the springs store energy at impact release it throughout a step. This two factors lead the learned controller to generate overshoot in the Cartesian velocities. We mitigate the overshoot by introducing a speed-damping term on the torque of the swing leg and the swing hip,

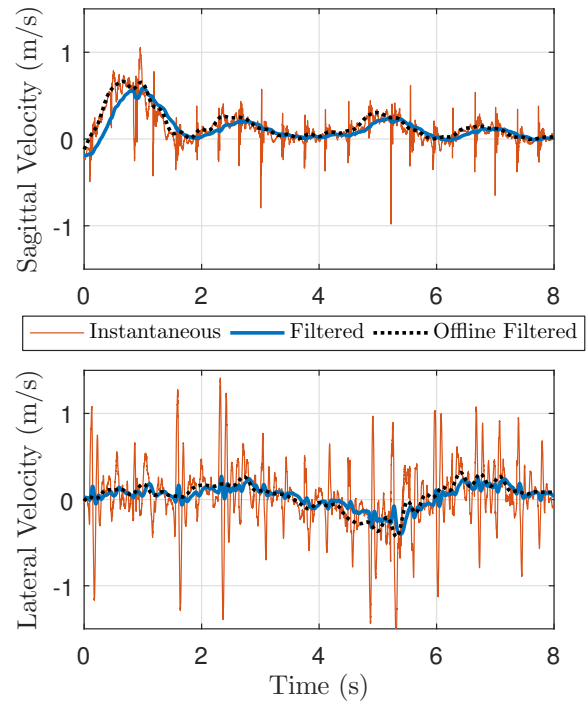
$$\begin{aligned} u_d^{sw,LA} &= N_{x,d}(v_x - v_x^k) \\ u_d^{sw,3} &= N_{x,d}(v_y - v_y^k), \end{aligned} \quad (87)$$

which is the same term used in Rezazadeh et al. (2015). The overall torque is

$$u = u_{PD} + u_d. \quad (88)$$

#### 7.5 Results

The learned controller for stabilizing the stepping-in-place gait was implemented on MARLO. The nominal Cartesian velocities are thus zero. Forces were applied in the sagittal and lateral directions, or a mix of the two, by the experimenter applying a push or a kick to the robot. The amount of force has not been estimated, but the reader can

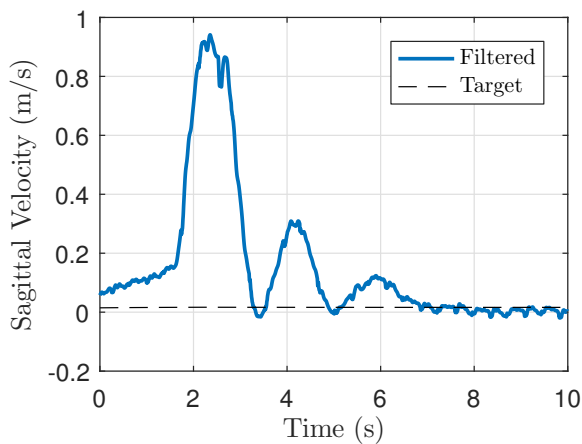


**Figure 27.** A perturbation is applied from 1.2 seconds to 2 seconds in a magnitude of 150 N. The maximum speed is larger than the maximum speed (0.8 m/s) in the training set. The extrapolation may be the cause of the speed oscillation. Even though, the speed convergence near the target speed in less than five steps.

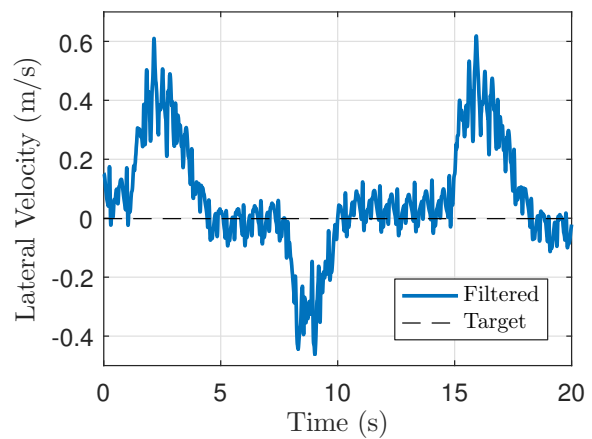
judge of his-or-herself based on the experiment video (see also Extension 2).

In the first experiment, five successive kicks were applied to MARLO in the forward (sagittal) direction. MARLO consistently recovered from the disturbances. The peak speed varied from 0.8 to 1 m/s, as shown in Figure 28. A harder kick was not applied since the training set only includes speed up to 0.6 m/s. After reaching the peak speed, MARLO slowed down to 0.1 m/s in less than 5 seconds. The robot acted as an underdamped spring-load system. This may be caused by the leg springs in the physical which compressed and unloaded on the second step, while the model did not include this effect. We have added the damping term in (87) to mitigate the overshoot effect. A larger derivative gain will further reduce the overshoot, but will so increase the settling time. Still, the speed slowed down to 0.1 m/s in less than 5 seconds. The leg motor torque is shown in Figure 28. The torque bound (5 Nm) was reached when robot moved around the peak speed. This could explain why the optimization can only find the solution of transition gait from 0.6 m/s to zero.

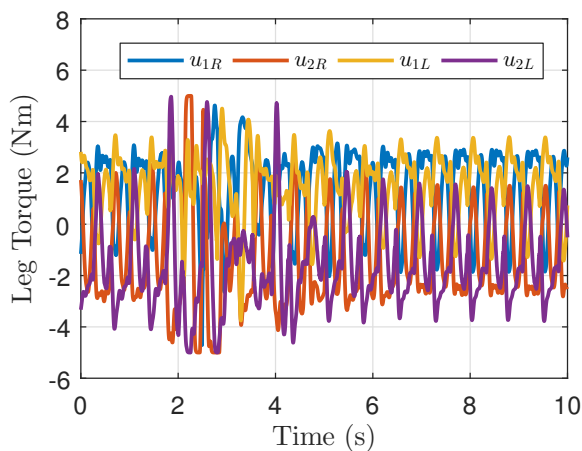
The second experiment was to push MARLO in the lateral direction. Because of the parallelogram shape of the legs, one foot cannot place across the other, which limits the available range of foot placement. Plus the weaker motor on the hip,



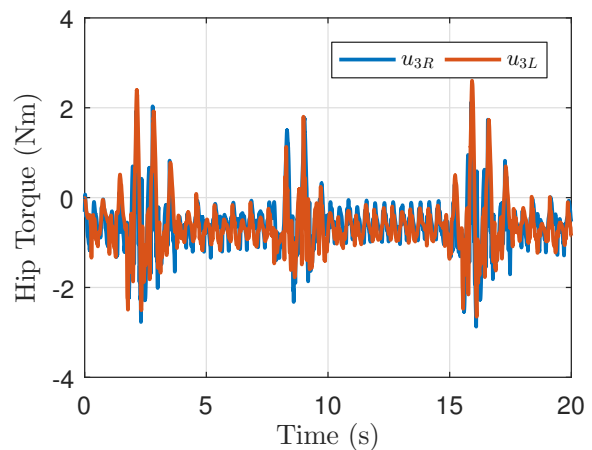
**Figure 28.** An example of the velocity response for a kick in the forward (sagittal) direction. The perturbation is applied at around 1 second driving the robot to peak speed of 0.9 m/s.



**Figure 30.** An example of the velocity response for multiple pushes in the lateral direction. The positive sign is the right direction whereas the negative is the left.



**Figure 29.** The leg torques are applied before a 1:50 gear transmission. The torques ( $u_{1R}$ ,  $u_{2R}$ ,  $u_{2L}$ ,  $u_{1L}$ ) are associated with the leg joints ( $q_{1R}$ ,  $q_{2R}$ ,  $q_{1L}$ ,  $q_{2L}$ ) in the robot configuration, Figure 17.



**Figure 31.** The hip torques are applied before a 1:27.5 gear-belt drive. The positive torque is to move leg inward whereas the positive torque is to move leg outward.

the lateral stability of MARLO is weaker than the sagittal direction. The push drove the lateral speed to 0.6 m/s, which is higher than the training speed 0.4 m/s. The push was applied on both left and right sides, shown in Figure 30. The hip motor torque is shown in Figure 31. The torque bound is 3 Nm.

Random direction pushes and kicks were included in the last part of the experiment video. We applied force to move MARLO backward and to turn around.

## 8 Discussion and Conclusions

### 8.1 Strategy

This paper is building on the recent revolution in open-loop trajectory optimization. It is now possible to compute in minutes gaits that used to take us hours or more. Armed with a set of open-loop trajectories, the question we posed was, how to turn them into a feedback controller?

Our strategy was to attempt to build a surface from the trajectories and to induce a vector field on that surface that had desirable properties, such as (1) it contained a periodic solution of the model that met important physical constraints; (2) trajectories on the surface, by design, converged to the periodic solution; and (3), we could find a feedback controller for the complete model of the system that would render this surface exponentially attractive.

We have used Supervised Machine Learning as a “universal” function approximation, in other words, glorified regression. The functions we sought were implicitly contained in the data, and to our knowledge, closed-form solutions seem unlikely to exist. Hence, they had to be computed numerically in one fashion or another, and we believe an important contribution of the paper is to show how the functions needed to build a feedback controller can be extracted from a set of trajectories over a fixed time interval.

## 8.2 Curse of Dimensionality

The method we use to build a vector field from open-loop trajectories works, at least in principle, in large dimensions. Even with the large strides made in optimization, high-dimensional state spaces are still the bottleneck, at least with our approach. Hence, it was important to find a structural property in our robot models that would allow us to focus the optimization effort on a low-dimensional portion of the system. We chose to exploit the local input-output linearizability of the actuators associated with knees and hips for example and put into the “weakly-actuated category” things like the global orientation of the body and possibly ankle joints. This allowed us to build trajectories of the full-model parameterized by initial conditions of a small subsystem, without making any approximations. In the end, we do build the control law for the full model around a low-dimensional model, just as advocates of pendulum models do, however, and this is important, all of the solutions of our low-dimensional model are feasible solutions of the robot itself and they meet whatever constraints were included in the design of the trajectories.

## 8.3 Original HZD vs G-HZD

Once one understands how the G-HZD tool works, it’s hard to believe how much the previous method could accomplish with a single optimization. The work presented in [Westervelt et al. \(2007\)](#) uses a single optimization to design the periodic orbit. And that is it. For robots with one degree of underactuation and for which a “mechanical phase variable” can be found, that is a strictly monotonically increasing generalized coordinate along the periodic orbit, the basic HZD result in [Westervelt et al. \(2007\)](#) shows how to build an invariant surface, relate stability of the periodic orbit in the surface to a physical property of the periodic orbit<sup>8</sup>, and how to render the surface exponentially attractive. G-HZD can handle more than one degree of underactuation. As in [Reher et al. \(2016a,b\)](#); [Kolathaya and Ames \(2017\)](#), G-HZD includes “time” as a monotonically increasing generalized coordinate. What makes it quite distinct from these references is that the equivalent of “G-HZD virtual constraints”, the function  $\nu(t, x_1)$ , depends on time and the full state of the zero dynamics, thereby enriching the set of possible closed-loop behaviors.

With the previous work on HZD and its extensions in [Buss et al. \(2016\)](#); [Hamed et al. \(2016\)](#); [Griffin and Grizzle \(2016\)](#), we were unable to handle challenging terrain such as the Michigan Wave Field . This motivated the introduction of a family of periodic orbits in [Da et al. \(2016\)](#) and a first

attempt at including transitions among the periodic orbits in [Da et al. \(2017\)](#) as a means of building in stability. While this latter paper also used Supervised Machine Learning to design a feedback function, it also made some false steps relying on analogies with model predictive control that were not supported by deeper analysis. The present paper is our attempt to provide a consistent design framework. An attractive feature of the original HZD approach is that it has an easily verifiable set of sufficient conditions for its many results. We hope in some future paper, a similarly clean analytical framework will be developed for G-HZD.

## 8.4 Stability Mechanism

Not only do the new results handle higher degrees of underactuation but even in the case of one degree of under actuation, the way stability is achieved is quite different in G-HZD vs. HZD. As discussed in the Introduction, with G-HZD, one does not have to count on the impacts to create stability. More general stability mechanisms, such as foot placement, or as shown in [Figure 24](#), lowering the center of mass, naturally arise.

## 8.5 Future Work

We see this paper as a first cut in developing a happy marriage among trajectory optimization, machine learning, and geometric nonlinear control. We hope the results in the paper can be reinforced with easy-to-check sufficient conditions for our many assumptions. Beyond these technical considerations, we also see several other directions. The recent work in [Consolini et al. \(2016\)](#) may provide a geometric formulation for building the invariant surface, which would also clarify the choice of coordinates for making the mapping  $\Psi$  in [\(20\)](#) and its projection to be full rank. We believe the feedback linearization assumption on the “strongly actuated” part of the dynamics can be weakened considerably. Replacing the terminal condition in the optimization with a terminal penalty is another direction that needs to be investigated.

## Acknowledgements

Professor Jonathan Hurst (Oregon State University), Mikhail Jones (Agility Robotics) and the whole team in the Dynamic Robotics Laboratory (Oregon State University) are sincerely thanked for sharing their copy of ATRIAS. The experiments reported here have also benefited greatly from the participation of Ph.D. students Omar Harib, Ross Hartley, and Yukai Gong (University of Michigan). The optimization and gait generation for the 3D walking and



experiments were contributed by a postdoctoral researcher Ayonga Hereid (University of Michigan).

### Declaration of conflicting interests

The Authors declare that there is no conflict of interest.

### Funding

This work was supported in part by NSF Grants CPS-1239037 and NRI-1525006, and in part by Toyota Research Institute under award TRI-N021515; however this article solely reflects the opinions and conclusions of its authors and not any of the funding entities.

### Notes

1.  $Z_0$  only contains the initial conditions for the trajectories. The evolution of the trajectories will determine  $Z$ . The vector field and output map in (4) will be computed through a combination of Supervised Machine Learning and model structure.
2. Almost any linear combination of  $x_1$  and  $x_2$  works because it takes rows from the bottom of (20) and adds them to the top rows, making (19) full rank, and hence locally injective. There is nothing magic about our choice.
3. Modulo  $T_p$  is not required here because  $\tau$  is reset at  $T_p$ , whereas in the non-hybrid case, it was required in (29).
4. Because the legs are 1 m long, average walking speed and angular rate at the middle of the step are nearly the same.
5. For example, due to torque limits, there is no solution for  $\bar{v} = 0.8$  and  $\xi_1 = (\theta_m^{\bar{v}} + \pi/12, 0.8 + \dot{\theta}_m^{\bar{v}})$ .
6. They essentially correspond the low-level PD gains which are straightforward to tune on MARLO. In simulation,  $K_p = 800$ ,  $K_d = 40$  and  $\epsilon = 1$
7. The time constant is 0.2 second.
8. The velocity of the robots center of mass should point downward at the end of the step.

### References

- Aghasadeghi N, Zhao H, Hargrove LJ, Ames AD, Perreault EJ and Bretl T (2013) Learning impedance controller parameters for lower-limb prostheses. In: *Intelligent robots and systems (IROS), 2013 IEEE/RSJ international conference on*. IEEE, pp. 4268–4274.
- Agrawal A, Harib O, Hereid A, Finet S, Masselin M, Praly L, Ames AD, Sreenath K and Grizzle JW (2017) First steps towards translating HZD control of bipedal robots to decentralized control of exoskeletons. *IEEE Access* 5: 9919–9934. DOI: 10.1109/ACCESS.2017.2690407.
- Ames A, Galloway K, Grizzle JW and Sreenath K (2014) Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control* 59(4): 876–891.
- Ames AD (2014) Human-inspired control of bipedal walking robots. *IEEE Transactions on Automatic Control* 59(5): 1115–1130. DOI:10.1109/TAC.2014.2299342.
- Ames AD, Cousineau EA and Powell MJ (2012) Dynamically stable bipedal robotic walking with NAO via human-inspired hybrid zero dynamics. In: *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*. ACM, pp. 135–144.
- Apostolopoulos S, Leibold M and Buss M (2015) Settling time reduction for underactuated walking robots. In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, pp. 6402–6408.
- Astolfi A and Ortega R (2003) Immersion and invariance: a new tool for stabilization and adaptive control of nonlinear systems. *IEEE Transactions on Automatic Control* 48(4): 590 – 606.
- Azevedo C, Poignet P and Espiau B (2002) Moving horizon control for biped robots without reference trajectory. In: *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 3. pp. 2762–2767. DOI:10.1109/ROBOT.2002.1013650.
- Azevedo C, Poignet P and Espiau B (2004) Artificial locomotion control: from human to robots. *Robotics and Autonomous Systems* 47(4): 203–223.
- Bainov D and Simeonov P (1989) *Systems with Impulse Effects : Stability, Theory and Applications*. Chichester: Ellis Horwood Limited.
- Buss BG, Hamed KA, Griffin BA and Grizzle JW (2016) Experimental results for 3D bipedal robot walking based on systematic optimization of virtual constraints. In: *American control conference*.
- Buss BG, Ramezani A, Akbari Hamed K, Griffin, B A, Galloway KS and Grizzle JW (2014) Preliminary walking experiments with underactuated 3D bipedal robot MARLO. In: *Intelligent Robots and Systems (IROS)*. IEEE, pp. 2529–2536.
- Chevallereau C, Abba G, Aoustin Y, Plestan F, Westervelt ER, Canudas-De-Wit C and Grizzle JW (2003) Rabbit: a testbed for advanced control theory. *IEEE Control Systems* 23(5): 57–79. DOI:10.1109/MCS.2003.1234651.
- Clarke FH, Ledyav YS, Sontag ED and Subbotin AI (1997) Asymptotic controllability implies feedback stabilization. *IEEE Transactions on Automatic Control* 42(10): 1394–1407.
- Consolini L, Costalunga A and Maggiore M (2016) A coordinate-free theory of virtual holonomic constraints. *arXiv preprint arXiv:1709.07726 [math.OC]*.
- Coron JM (1994) On the stabilization of controllable and observable systems by an output feedback law. *Mathematics*

- of Control, Signals, and Systems (MCSS) 7(3): 187–216.
- Coron JM, Praly L and Teel A (1995) Feedback stabilization of nonlinear systems: Sufficient conditions and Lyapunov and input-output techniques. In: *Trends in control*. Springer, pp. 293–348.
- Da X, Harib O, Hartley R, Griffin B and Grizzle JW (2016) From 2D design of underactuated bipedal gaits to 3D implementation: Walking with speed tracking. *IEEE Access* 4: 3469–3478. DOI:10.1109/ACCESS.2016.2582731.
- Da X, Hartley R and Grizzle JW (2017) Supervised learning for stabilizing underactuated bipedal robot locomotion, with outdoor experiments on the wave field. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 3476–3483. DOI:10.1109/ICRA.2017.7989397.
- Dunn ER and Howe RD (1996) Foot placement and velocity control in smooth bipedal walking. In: *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1. IEEE, pp. 578–583.
- Embry KR, Villarreal DJ and Gregg RD (2016) A unified parameterization of human gait across ambulation modes. In: *Submit to: International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE.
- Erez T, Lowrey K, Tassa Y, Kumar V, Koley S and Todorov E (2013) An integrated system for real-time model predictive control of humanoid robots. In: *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. pp. 292–299. DOI:10.1109/HUMANOIDS.2013.7029990.
- Faraji S, Pouya S, Atkeson CG and Ijspeert AJ (2014) Versatile and robust 3D walking with a simulated humanoid robot (Atlas): a model predictive control approach. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1943–1950.
- Full R and Koditschek D (1999) Templates and anchors: Neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology* 202: 3325–3332.
- Gregg RD, Lenzi T, Hargrove LJ and Sensinger JW (2014) Virtual constraint control of a powered prosthetic leg: From simulation to experiments with transfemoral amputees. *IEEE Transactions on Robotics* .
- Griffin B and Grizzle J (2016) Nonholonomic virtual constraints and gait optimization for robust walking control. *The International Journal of Robotics Research* : 0278364917708249.
- Grizzle JW, Chevallereau C, Sinnet RW and Ames AD (2014) Models, feedback control, and open problems of 3D bipedal robotic walking. *Automatica* 50(8): 1955–1988.
- Hamed KA, Buss BG and Grizzle JW (2016) Exponentially stabilizing continuous-time controllers for periodic orbits of hybrid systems: Application to bipedal locomotion with ground height variations. *The International Journal of Robotics Research* 35(8): 977–999. DOI:10.1177/0278364915593400.
- Han W and Tedrake R (2017) Feedback design for multi-contact push recovery via LMI approximation of the piecewise-affine quadratic regulator. In: *Under Review*.
- Hartley R, Da X and Grizzle JW (2017) Stabilization of 3D underactuated biped robots: Using posture adjustment and gait libraries to reject velocity disturbances. In: *IEEE Conference on Control Technology and Applications (CCTA)*.
- Hereid A, Cousineau EA, Hubicki CM and Ames AD (2016a) 3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. In: *IEEE International Conference on Robotics and Automation (ICRA)*.
- Hereid A, Kolathaya S and Ames AD (2016b) Online hybrid zero dynamics optimal gait generation using Legendre pseudospectral optimization. In: *To appear in: IEEE Conference on Decision and Control (CDC)*. IEEE.
- Hereid A, Kolathaya S, Jones MS, Van Why J, Hurst JW and Ames AD (2014) Dynamic multi-domain bipedal walking with ATRIAS through SLIP based human-inspired control. In: *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control, HSCC '14*. New York, NY, USA: ACM. ISBN 978-1-4503-2732-9, pp. 263–272. DOI: 10.1145/2562059.2562143.
- Isidori A (1995) *Nonlinear Control Systems*. Third edition. Berlin: Springer-Verlag.
- Jones MS (2014) *Optimal control of an underactuated bipedal robot*. Master's Thesis, Oregon State University, ScholarsArchive@OSU.
- Kajita S and Tani K (1991) Study of dynamic biped locomotion on rugged terrain-theory and basic experiment. Pisa, pp. 741–746.
- Kajita S, Yamamura T and Kobayashi A (1992) Dynamic walking control of biped robot along a potential energy conserving orbit. *IEEE Transactions on Robotics and Automation* 8(4): 431–37.
- Kaneko K, Kanehiro F, Morisawa M, Akachi K, Miyamori G, Hayashi A and Kanehira N (2011) Humanoid robot HRP-4 – humanoid robotics platform with lightweight and slim body. pp. 4400–4407.
- Karagiannis D, Astolfi A and Ortega R (2005) Nonlinear stabilization via system immersion and manifold invariance: survey and new results. *Multiscale Modeling & Simulation* 3(4): 801–817.
- Khalil H (2002) *Nonlinear Systems - 3rd Edition*. Upper Saddle River, NJ: Prentice Hall.
- Kolathaya S and Ames AD (2017) Parameter to state stability of control Lyapunov functions for hybrid system models of robots.

- Nonlinear Analysis: Hybrid Systems* 25: 174–191.
- Koolen T, de Boer T, Rebula J, Goswami A and Pratt J (2012) Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models. *The International Journal of Robotics Research* 31(9): 1094–1113. DOI:10.1177/0278364912452673.
- Krause M, Engelsberger J, Wieber PB and Ott C (2012) Stabilization of the capture point dynamics for bipedal walking based on model predictive control. *IFAC Proceedings Volumes* 45(22): 165–171.
- Kuindersma S, Deits R, Fallon M, Valenzuela A, Dai H, Permenter F, Koolen T, Marion P and Tedrake R (2016) Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous Robots* 40(3): 429–455.
- Liu C, Atkeson CG and Su J (2013) Biped walking control using a trajectory library. *Robotica* 31(02): 311–322.
- Manchester IR and Umenberger J (2014) Real-time planning with primitives for dynamic walking over uneven terrain. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, pp. 4639–4646.
- Marcucci T, Deits R, Gabiccini M, Bicchi A and Tedrake R (2017) Approximate hybrid model predictive control for multi-contact push recovery in complex environments. In: *Under Review*.
- Martin AE, Post DC and Schmedeler JP (2014) Design and experimental implementation of a hybrid zero dynamics-based controller for planar bipeds with curved feet. *The International Journal of Robotics Research* 33(7): 988–1005. DOI:10.1177/0278364914522141.
- Mayne DQ, Rawlings JB, Rao CV and Scokaert PO (2000) Constrained model predictive control: Stability and optimality. *Automatica* 36(6): 789–814.
- Morris B and Grizzle JW (2009) Hybrid invariant manifolds in systems with impulse effects with application to periodic locomotion in bipedal robots. *IEEE Transactions on Automatic Control* 54(8): 1751–1764.
- Nguyen Q, Agrawal A, Da X, Martin WC, Geyer H, Grizzle JW and Sreenath K (2016) Dynamic walking on stepping stones with gait library and control barrier. In: *Workshop on Algorithmic Foundations of Robotics (WAFR)*.
- Nguyen Q, Agrawal A, Da X, Martin WC, Geyer H, Grizzle JW and Sreenath K (2017) Dynamic walking on randomly-varying discrete terrain with one-step preview. In: *Robotics: Science and Systems (RSS)*.
- Ogura Y, Aikawa H, Shimomura K, Morishima A, Lim H and Takanishi A (2006) Development of a new humanoid robot WABIAN-2. In: *Proc. of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL*. pp. 76–81.
- Parisini T and Zoppoli R (1995) A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica* 31(10): 1443–1451.
- Park H, Ramezani A and Grizzle JW (2013) A finite-state machine for accommodating unexpected large ground height variations in bipedal robot walking. *IEEE Transactions on Robotics* 29(29): 331–345.
- Park I, Kim J, Lee J and Oh J (2005) Mechanical design of humanoid robot platform KHR-3 (KAIST Humanoid Robot 3: HUBO). In: *Proc. 5th IEEE—RAS Int. Conf. Humanoid Robots*. pp. 321–326.
- Parker TS and Chua LO (1989) *Practical Numerical Algorithms for Chaotic Systems*. New York, NY: Springer-Verlag.
- Pfeiffer F, Löffler K and Gienger M (2002) The concept of jogging Johnnie. In: *IEEE International Conference on Robotics and Automation*. Washington, DC.
- Powell MJ, Hereid A and Ames AD (2013) Speed regulation in 3D robotic walking through motion transitions between human-inspired partial hybrid zero dynamics. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, pp. 4803–4810.
- Pratt J, Koolen T, de Boer T, Rebula J, Cotton S, Carff J, Johnson M and Neuhaus P (2012) Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower-body humanoid. *The International Journal of Robotics Research* 31(10): 1117–1133. DOI:10.1177/0278364912452762.
- Pratt J and Tedrake R (2006) Velocity-based stability margins for fast bipedal walking. In: Diehl M and Mombaur K (eds.) *Fast Motions in Biomechanics and Robotics, Lecture Notes in Control and Information Sciences*, volume 340. Springer Berlin Heidelberg. ISBN 978-3-540-36118-3, pp. 299–324. DOI: 10.1007/978-3-540-36119-0\_14.
- Raibert MH (1986a) Legged robots. *Communications of the ACM* 29(6): 499–514.
- Raibert MH (1986b) *Legged robots that balance*. Mass.: MIT Press.
- Ramezani A, Hurst JW, Akbari Hamed K and Grizzle JW (2014) Performance Analysis and Feedback Control of ATRIAS, A Three-Dimensional Bipedal Robot. *Journal of Dynamic Systems, Measurement, and Control* 136(2).
- Reher J, Cousineau EA, Hereid A, Hubicki CM and Ames AD (2016a) Realizing dynamic and efficient bipedal locomotion on the humanoid robot DURUS. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 1794–1801. DOI:10.1109/ICRA.2016.7487325.
- Reher JP, Hereid A, Kolathaya S, Hubicki CM and Ames AD (2016b) Algorithmic foundations of realizing multi-contact locomotion on the humanoid robot DURUS. In: *The*

- International Workshop on the Algorithmic Foundations of Robotics (WAFR)*.
- Reyhanoglu M, van der Schaft A, McClamroch N and Kolmanovskiy I (1999) Dynamics and control of a class of under-actuated mechanical systems. *IEEE Transactions on Automatic Control* 44(9): 1663–1671.
- Rezazadeh S, Hubicki C, Jones M, Peekema A, Van Why J, Abate A and Hurst JW (2015) Spring-mass walking with ATRIAS in 3D: Robust gait control spanning zero to 4.3 kph on a heavily underactuated bipedal robot. *ASME Dynamic Systems and Control Conference (DSCC)* : 23.
- Saglam CO and Byl K (2015) Meshing hybrid zero dynamics for rough terrain walking. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 5718–5725.
- Sakagami Y, Watanabe R, Aoyama C, Matsunaga S, Higaki N and Fujimura K (2002) The intelligent ASIMO: system overview and integration. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Lausanne, Switzerland, pp. 2478–83.
- Schürmann B and Althoff M (2017) Convex interpolation control with formal guarantees for disturbed and constrained nonlinear systems. In: *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*. ACM, pp. 121–130.
- Shim H and Teel AR (2003) Asymptotic controllability and observability imply semiglobal practical asymptotic stabilizability by sampled-data output feedback. *Automatica* 39(3): 441–454.
- Spong MW (1994) Partial feedback linearization of underactuated mechanical systems. In: *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Munich, Germany*. pp. 314–321.
- Spong MW (1996) Energy based control of a class of underactuated mechanical systems. In: *Proc. of IFAC World Congress, San Francisco, CA*. pp. 431–435.
- Sreenath K, Park H and Grizzle JW (2012) Design and experimental implementation of a compliant hybrid zero dynamics controller with active force control for running on MABEL. In: *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*. pp. 51–56. DOI:10.1109/ICRA.2012.6224944.
- Sreenath K, Park H, Poulakakis I and Grizzle J (2011) A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *International Journal of Robotics Research* 30(9): 1170–1193.
- Sreenath K, Park HW, Poulakakis I and Grizzle J (2013) Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL. *The International Journal of Robotics Research* 32(3): 324–345.
- Vidyasagar M (2002) *Nonlinear systems analysis*. SIAM.
- Wang L, Forni F, Ortega R, Liu Z and Su H (2017) Immersion and invariance stabilization of nonlinear systems via virtual and horizontal contraction. *IEEE Transactions on Automatic Control* 62(8): 4017–4022.
- Wang T and Chevallereau C (2011) Stability analysis and time-varying walking control for an under-actuated planar biped robot. *Robotics and Autonomous Systems* 59(6): 444 – 456. DOI:https://doi.org/10.1016/j.robot.2011.03.002.
- Westervelt E, Grizzle JW and Koditschek D (2003) Hybrid zero dynamics of planar biped walkers. *IEEE Transactions on Automatic Control* 48(1): 42–56.
- Westervelt ER, Grizzle JW, Chevallereau C, Choi J and Morris B (2007) *Feedback Control of Dynamic Bipedal Robot Locomotion*. Control and Automation. Boca Raton, FL: CRC Press.
- Westervelt ER, Grizzle JW and Koditschek DE (2002) Zero dynamics of planar biped walkers with one degree of under actuation. In: *IFAC World Congress*. Barcelona, Spain.
- Yamaguchi J, Soga E, Inoue S and Takanishi A (1999) Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking. In: *Proc. of the 1999 IEEE International Conference on Robotics and Automation, Detroit, MI*. pp. 368–374.
- Zaytsev P, Hasaneini SJ and Ruina A (2015) Two steps is enough: No need to plan far ahead for walking balance. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 6295–6300. DOI:10.1109/ICRA.2015.7140083.
- Zhao H, Horn J, Reher J, Paredes V and Ames AD (2015) A hybrid systems and optimization-based control approach to realizing multi-contact locomotion on transfemoral prostheses. In: *IEEE Conference on Decision and Control (CDC)*. pp. 1607–1612. DOI:10.1109/CDC.2015.7402440.

## Appendix A Index to Multimedia Extensions

Extension	Type	Description
1	Code	Inverted pendulum example
2	Video	3D robot experiments

## Appendix B Normal Forms for Mechanical Models

Consider a standard mechanical model

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu$$

and let

$$\Omega(q, \dot{q}) := C(q, \dot{q})\dot{q} + G(q).$$

We suppose the system is *underactuated*, that is, there are fewer independent actuators than generalized coordinates. In fact, we suppose there exists a partition of the coordinates in which the model takes the form

$$\begin{aligned} D_{11}(q)\ddot{q}_1 + D_{12}(q)\ddot{q}_2 + \Omega_1(q, \dot{q}) &= 0 \\ D_{21}(q)\ddot{q}_1 + D_{22}(q)\ddot{q}_2 + \Omega_2(q, \dot{q}) &= B_2u, \end{aligned} \quad (89)$$

with  $B_2$  square and invertible. Because  $D(q)$  is positive definite, by the Shur Complement Lemma,  $D_{11}(q)$ ,  $D_{22}(q)$ , and

$$\bar{D}(q) := D_{22}(q) - D_{21}(q)D_{11}^{-1}(q)D_{12}(q) \quad (90)$$

are all positive definite as well.

Following [Spong \(1994\)](#), define

$$\begin{aligned} J^{\text{norm}}(q) &:= D_{11}^{-1}(q)D_{12}(q) \\ \bar{\Omega}_1(q, \dot{q}) &:= -D_{11}^{-1}(q)\Omega_1(q, \dot{q}) \\ \bar{\Omega}_2(q, \dot{q}) &:= \Omega_2(q, \dot{q}) - D_{21}(q)D_{11}^{-1}(q)\Omega_1(q, \dot{q}), \end{aligned} \quad (91)$$

Then the (regular) feedback

$$u = B_2^{-1}(q) (\bar{D}(q)v + \bar{\Omega}_2(q, \dot{q})), \quad (92)$$

results in the Spong normal form:

$$\begin{aligned} \ddot{q}_1 &= \bar{\Omega}_1(q, \dot{q}) - J^{\text{norm}}(q)v, \\ \ddot{q}_2 &= v. \end{aligned} \quad (93)$$

Defining  $x_1 = (q_1, \dot{q}_1)$ ,  $x_{2a} = q_2$ , and  $x_{2b} = \dot{q}_2$ , it follows that the model can be expressed as

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2, v) \\ \dot{x}_{2a} &= x_{2b} \\ \dot{x}_{2b} &= v. \end{aligned} \quad (94)$$

An alternative form is developed in ([Westervelt et al. 2007](#), pp. 62), which uses the conjugate momenta that arises from Lagrange's equations. It has the advantage that the input only

appears in the second row of the model. Define

$$\sigma_1 := D_{11}(q)\dot{q}_1 + D_{12}(q)\dot{q}_2 \quad (95)$$

$$\dot{D}(q, \dot{q}) := \frac{d}{dt}D(q) \quad (96)$$

Then the model can also be written as

$$\begin{aligned} \dot{q}_1 &= D_{11}^{-1}(q) [\sigma_1 - D_{12}(q)\dot{q}_2] \\ \dot{\sigma}_1 &= \kappa_1(q, \sigma_1, \dot{q}_2) \\ \ddot{q}_2 &= v \end{aligned} \quad (97)$$

where

$$\begin{aligned} \kappa_1(q, \sigma_1, \dot{q}_2) &:= \left( \dot{D}_{11}(q, \dot{q})\dot{q}_1 + \dot{D}_{12}(q, \dot{q})\dot{q}_2 - \right. \\ &\quad \left. \Omega_1(q, \dot{q}) \right) \Big|_{\dot{q}_1 = D_{11}^{-1}(q)[\sigma_1 - D_{12}(q)\dot{q}_2]} \end{aligned} \quad (98)$$

With  $x_2$  defined as above and  $x_1 := (q_1, \sigma_1)$ , the model takes the form

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2) \\ \dot{x}_{2a} &= x_{2b} \\ \dot{x}_{2b} &= v. \end{aligned} \quad (99)$$

Various authors prefer one of (94) and (99) to the other; both are useful.

## Appendix C Proofs

### C.1 Proof of Proposition 1

The proof is most easily done using the method of Poincaré sections [Parker and Chua \(1989\)](#). By A-1 and the assumption on  $u^{cu}$ , the closed-loop system (11) has period  $T_p > 0$  and the origin is an equilibrium point. Due to the time-varying nature of the closed-loop system, we make time a state variable, and because the system is  $T_p$ -periodic, we add in a time-based reset map

$$\begin{cases} \dot{\tau} = 1, & \tau^- < T_p \\ \dot{x} = f^{cu}(\tau, x) := f(\tau, x, u^{cu}(\tau, x)), \\ \tau^+ = 0 & \tau^- = T_p \\ x^+ = x^- \end{cases} \quad (100)$$

The notation  $\tau^-$ ,  $\tau^+$ ,  $x^-$  and  $x^+$  is explained in Section 5. Because the state reset map is trivial, namely  $x^+ = x^-$ , the solutions of (11) and (100) are identical. Define a Poincaré section by

$$\mathcal{S}_n := \{(\tau, x) \in \mathbb{R}^{n+1} \mid \tau = T_p, x \in B\}, \quad (101)$$

which is an  $n$ -dimensional hypersurface in the state space of the model. Then, by construction of the closed-loop system,



for  $\xi \in \mathcal{S}_n$ , the Poincaré map  $P : \mathcal{S}_n \rightarrow \mathcal{S}_n$  is given by

$$P(\xi) = \varphi_\xi(T_p). \quad (102)$$

Indeed, for  $\xi \in B$ ,

$$\begin{aligned} \varphi_\xi(t) &= \xi + \int_0^t f(\varphi_\xi(s), u_\xi(s)) ds \\ &= \xi + \int_0^t f(\varphi_\xi(s), u^{cu}(s, \varphi_\xi(s))) ds, \end{aligned}$$

due to (10). By A-4,  $\xi^* = 0$  is a fixed point of the Poincaré map. Also by A-4,  $P$  is a contraction because for each  $\xi \in B$ ,  $V \circ P(\xi) \leq cV(\xi)$ , and hence by induction,

$$V \circ P^k(\xi) \leq c^k V(\xi),$$

and by A-3,

$$\|P^k(\xi)\|^2 \leq c^k \frac{\alpha_2}{\alpha_1} \|\xi\|^2 \xrightarrow{k \rightarrow \infty} 0,$$

proving local exponential stability of the fixed point. The uniformity in  $t_0$  follows from periodicity. ■

### C.2 Proof of Proposition 2

Without loss generality, we assume that  $B_{1cu}$  is bounded so that its closure is compact. Then there exists  $L$ , a Lipschitz constant, such that

$$\|\gamma(x_1)\|_2 \leq L\|x_1\|_2$$

for all  $x_1 \in B_{1cu}$ . Define  $V_1 : B_1 \rightarrow \mathbb{R}$  by  $V_1(x_1) := V(x_1, \gamma(x_1))$ . It follows that

$$\alpha_1 x_1^\top x_1 \leq V_1(x_1) \leq \alpha_2 (1 + L^2) x_1^\top x_1,$$

and hence  $V_1$  is positive definite, with quadratic lower and upper bounds. From (16),

$$V_1(\varphi_{1\xi_1}(T_p)) \leq cV_1(\xi_1).$$

From here, the proof of Prop. 1 can be repeated and the result follows. ■

### C.3 Proof of Theorem. 1

From the hypotheses of the Theorem and Prop. 2, the closed-loop system (26) is a cascade of two locally exponentially systems, namely, the second row of (26) and the reduced-order system (18). By standard results, the overall system is locally exponentially stable. By (Vidyasagar 2002, Thm.

43, Section 5.1, pp. 143), because the system is periodically time-varying, the stability is uniform in  $t_0$ . ■

### C.4 Proof of Corollary. 1

Defining  $y$  as in (25) results in the closed-loop system having the form

$$\begin{aligned} \dot{x}_1 &= f_1(t, x_1, \nu(t, x_1) + y, \mu(t, x_1) - [K_p \ K_d]y) \\ \dot{y} &= Ay, \end{aligned} \quad (103)$$

with  $A$  Hurwitz. Hence, the proof of Theorem 1 can be repeated and we are done. ■

### C.5 Proof of Proposition 3

The proof is very similar to that of Prop. 1. Define a Poincaré section by

$$\mathcal{S}_n := \{(\tau, x) \in \mathbb{R}^{n+1} \mid \tau = T_p, x \in B\}, \quad (104)$$

which is an  $n$ -dimensional hypersurface in the state space of the model. Then, by construction of the closed-loop system, for  $\xi \in \mathcal{S}_n$ , the Poincaré map  $P : \mathcal{S}_n \rightarrow \mathcal{S}_n$  is given by

$$P(\xi) = \varphi_\xi(T_p). \quad (105)$$

By H-4,  $\xi^* := x_m^*$  is a fixed point of the Poincaré map. Also by H-4,  $P$  is a contraction because for each  $\xi \in B$ ,  $V \circ P(\xi) \leq cV(\xi)$ , and hence by induction,

$$V \circ P^k(\xi) \leq c^k V(\xi),$$

and by H-3,

$$\|P^k(\xi)\|^2 \leq c^k \frac{\alpha_2}{\alpha_1} \|\xi - \xi^*\|^2 \xrightarrow{k \rightarrow \infty} 0,$$

proving local exponential stability of the fixed point. Because the closed-loop system is locally Lipschitz continuous, local exponential stability of the fixed point implies exponential attractivity of the orbit

$$\mathcal{O} := \{(\tau, \varphi_{\xi^*}(\tau) \mid 0 \leq \tau < T_p\}.$$

Because  $\tau(t) = t$ , we have local exponential stability of the periodic solution. ■

### C.6 Proof of Proposition 4

Without loss generality, we assume that  $B_{1cu}$  is bounded so that its closure is compact. Then there exists  $L$ , a Lipschitz

constant, such that

$$\|\gamma(x_1 - x_1^*)\|_2 \leq L\|x_1 - x_1^*\|_2$$

for all  $x_1 \in B_{1cu}$ . Define  $V_1 : B_1 \rightarrow \mathbb{R}$  by  $V_1(x_1) := V(x_1, \gamma(x_1))$ . It follows that

$$\begin{aligned} \alpha_1(x_1 - x_1^*)^\top (x_1 - x_1^*) &\leq V_1(x_1) \leq \\ \alpha_2(1 + L^2)(x_1 - x_1^*)^\top (x_1 - x_1^*), & \end{aligned}$$

and hence  $V_1$  is positive definite, with quadratic lower and upper bounds. From (16),

$$V_1(\varphi_{1\xi_1}(T_p)) \leq cV_1(\xi_1).$$

From here, the proof of Prop. 3 can be repeated and the result follows. ■

### C.7 Proof of Theorem. 2

The Poincaré section is defined as in (104). References Grizzle et al. (2014) and (Westervelt et al. 2007, Chap. 4) show how to reduce the stability analysis of a hybrid model with two continuous phases to that of an equivalent hybrid system with a single continuous phase. We build the equivalent hybrid system with the continuous phase from  $\Sigma_{ii}$  and a reset map  $\Delta_{eq}$  that captures the flow of  $\Sigma_i$ , viz

$$\left\{ \begin{array}{l} \dot{\tau} = 1 \\ \dot{x} = f(x_1, x_2, u_1, u_2) \\ u_1 = \mu_{1ii}(\tau, x_1) \\ u_2 = \mu_{2ii}(\tau, x_1) - \left[ \frac{K_p}{\epsilon^2} \quad \frac{K_d}{\epsilon} \right] (x_2 - \nu_{ii}(\tau, x_1)) \\ \left[ \begin{array}{c} \tau \\ x \end{array} \right]^+ = \Delta_{eq}(\tau^-, x^-), \end{array} \right. \quad \begin{array}{l} \tau^- < T_p \\ \\ \\ \\ \tau^- = T_p \end{array} \quad (106)$$

With this construction and Prop. 4, the zero dynamics manifold is

$$\mathcal{Z} := \{(\tau, x_1, x_2) \mid x_2 = \nu_{ii}(\tau, x_1)\},$$

and the restricted Poincaré map  $\rho : \mathcal{S}_e \cap \mathcal{Z} \rightarrow \mathcal{S}_e \cap \mathcal{Z}$  has  $x_1^*$  as a locally exponentially stable fixed point. The equivalent hybrid system (106) therefore satisfies all the hypotheses of (Ames et al. 2014, Thm. 2), and hence the periodic orbit

$$\mathcal{O} := \{(\tau, \varphi_{\xi_1^*}(\tau) \mid 0 \leq \tau < T_p\}$$

is locally exponentially stable. ■

## Appendix D Relation to Backstepping, Zero Dynamics, and Immersion and Invariance

For definiteness, consider a standard Lagrangian dynamical model where  $q \in \mathbb{R}^n$  is a set of generalized coordinates and  $u \in \mathbb{R}^m$  is a vector of torques,

$$D(q)\ddot{q} + H(q, \dot{q}) = B(q). \quad (107)$$

Assume the system is underactuated, that is,  $n > m$ , and that the coordinates have been decomposed as

$$q := \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad (108)$$

in which the model takes the form

$$\begin{aligned} D_{11}(q)\ddot{q}_1 + D_{12}(q)\ddot{q}_2 + H_1(q, \dot{q}) &= 0 \\ D_{21}(q)\ddot{q}_1 + D_{22}(q)\ddot{q}_2 + H_2(q, \dot{q}) &= B_2(q)u. \end{aligned} \quad (109)$$

with  $B_2(q)$  square and full rank. References Spong (1996); Reyhanoglu et al. (1999); Westervelt et al. (2002) show that there is a regular feedback that places the system in the form

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, q_2, \dot{q}_2) \\ \ddot{q}_2 &= v, \end{aligned} \quad (110)$$

with

$$x_1 := \begin{bmatrix} q_1 \\ \sigma_1 \end{bmatrix} \text{ and } \sigma_1 := D_{11}(q)\dot{q}_1 + D_{12}(q)\dot{q}_2,$$

the generalized momentum conjugate to  $q_1$ .

### D.1 Backstepping

To begin the backstepping process in (110), one needs a feedback

$$\begin{bmatrix} q_2 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \nu_a(x_1) \\ \nu_b(x_1) \end{bmatrix}$$

that renders the origin of the reduced-order system

$$\dot{x}_1 = f_1(x_1, \nu_a(x_1), \nu_b(x_1)) \quad (111)$$

locally exponentially stable with a known Lyapunov function. However, to pull this feedback through the double integrator, it must be true that  $\nu_b(x_1) = \frac{d}{dt}\nu_a(x_1)$ , that is

$$\nu_b(x_1) = \left[ \frac{\partial}{\partial x_1} \nu_a(x_1) \right] f_1(x_1, \nu_a(x_1), \nu_b(x_1)). \quad (112)$$

Backstepping does not provide any systematic means to meet the required integrability condition. Of course, if the system has the form

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, q_2) \\ \ddot{q}_2 &= v,\end{aligned}\quad (113)$$

then there is no integrability constraint and backstepping can be done, assuming one is clever enough to find a feedback  $q_2 = \nu_a(x_1)$  that renders the origin of

$$\dot{x}_1 = f_1(x_1, \nu_a(x_1))$$

locally exponentially stable. The solution we presented in Sect. 3.4 uses trajectory optimization to automatically build a feedback that satisfies the integrability condition (112) and provides for local exponential stability. Moreover, bounds on inputs and other constraints can potentially be included in the trajectory optimization process, whereas they are challenging to incorporate into backstepping.

## D.2 Zero Dynamics

The method of Hybrid Zero Dynamics as developed in Westervelt et al. (2002) exploits the structure of  $f_1$  in (111), namely

$$\frac{d}{dt} \begin{bmatrix} q_1 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \\ f_{1b}(q_1, \dot{q}_1, q_2, \dot{q}_2) \end{bmatrix}$$

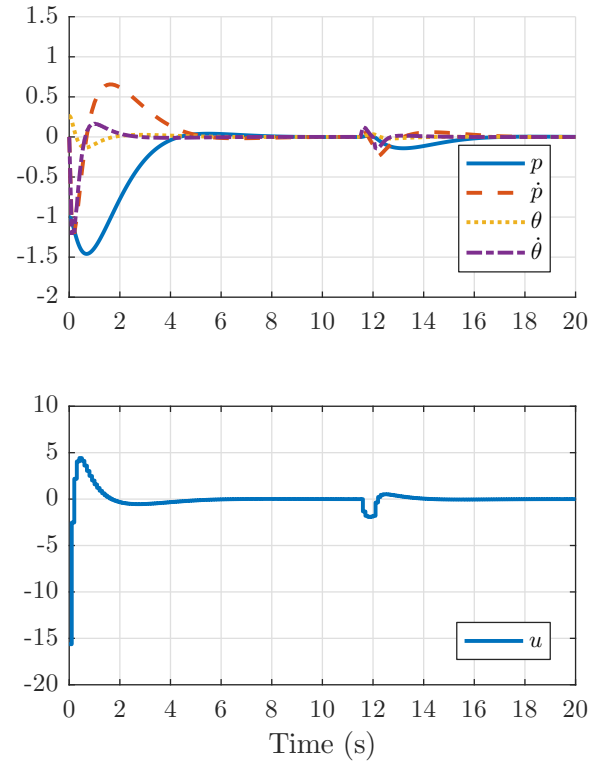
and

$$\dot{q}_1 = D_{11}^{-1}(q) [\sigma_1 - D_{12}(q)\dot{q}_2],$$

to solve for a solution of the form

$$\begin{bmatrix} q_2 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} h_d(q_1) \\ \left( \frac{\partial}{\partial q_1} h_d(q_1) \right) \dot{q}_1 \end{bmatrix},$$

so that the integrability condition (112) is automatically met. When the computational method in Jones (2014); Hereid et al. (2016a) does produce a solution, it does not come with a Lyapunov function and hence input-output linearization is often used to “pull” the virtual constraints back through the double integrators. Moreover, conditions for the *virtual constraints*  $q_2 = h_d(q_1)$  to stabilize (a hybrid version of) (111) are only known when  $q_1$  is a scalar. The solution we have given in Sect. 5 works for vector valued  $q_1$ , hence for models with more than one degree of underactuation. Moreover, even for one degree of underactuation, it provides a more general solution to the boundary value problem in that it naturally produces solutions of the form  $h_d(t, q_1, \dot{q}_1)$ , that is, the controller depends in a non-trivial way on the full state of the  $x_1$ -subsystem.



**Figure 32.** A classic MPC controller is applied to the same system as in Figure 9

## D.3 Immersion and Invariance

The method of immersion and invariance (I&I) presented in Astolfi and Ortega (2003); Karagiannis et al. (2005); Wang et al. (2017) is more general than backstepping and can provide alternative cascade realizations to the simple one used in (26). However, I&I still requires a target system to be provided, such as, (18), which is what our method is constructing. In other words, once a feedback satisfying Prop. 2 has been constructed, I&I can be used to build alternatives to the feedback used in Thm. 1, but it will not replace the design of the reduced-order model.

## Appendix E Standard MPC and Relative to This Work

Figure 32 shows the standard MPC with Zero-Order-Hold (ZOH) condition,

$$u^{zoh}(t, \xi) = u_\xi(0), t \in [0, T_p]. \quad (114)$$

We would like to implement this controller to the high dimensional system. However, it does not scale well. The learned feedback  $\mu(t, x)$  shows the similar performance as the classic MPC. Sect. 3.3 illustrates how to apply it to a reduced-order model while embedding it to the full-order model. In this sense, the curse of dimensionality has been mitigated.