Supervised Learning for Stabilizing Underactuated Bipedal Robot Locomotion, with Outdoor Experiments on the Wave Field

Xingye Da, Ross Hartley, and Jessy W. Grizzle*

***Abstract*— Supervised learning is used to build a control policy for robust, stable, dynamic walking of an underactuated bipedal robot. The training and testing sets consist of controllers based on a full dynamic model, virtual constraints, and parameter optimization to meet torque limits, friction cone, and environmental conditions. The controllers are designed to induce locally exponentially stable periodic walking gaits at various speeds, both forward and backward, and for various constant ground slopes. They are also designed to induce aperiodic gaits that transition among a subset of the periodic gaits in a fixed number of steps. In experiments, the learned policy allows a 3D bipedal robot to recover from a significant kick. It also enables the robot to walk down a 22 degree slope and walk on sinusoidally varying terrain, all without using a camera. During the development of these results, it is demonstrated that supervised learning of locally exponentially stable controllers can result in a loss of stability and a means to avoid this is suggested.**

I. INTRODUCTION

For many control tasks, real-time constrained optimization is becoming an important means of designing and implementing feedback control policies. With current computational power, it is not possible to achieve highly dynamic motions (e.g., running or jumping) or to respond to large perturbations with this approach. One alternative is to precompute a set of controllers and build an explicit control policy [1].

This paper proposes an offline approach to design an explicit model-based feedback control policy using ideas from parameter optimization and Machine Learning (ML). The control design process begins by using parameter optimization to generate both training and testing sets of controllers that induce walking gaits in a bipedal robot model. Virtual constraints provide a convenient parametrization of the feedback control laws and corresponding gaits [2]. The training and testing sets include locally exponentially stable periodic walking gaits at various speeds, both forward and backward, and for various constant ground slopes, flat ground, uphill and downhill. They also include aperiodic gaits that transition among a subset of the periodic gaits in a fixed number of steps.

Supervised learning is then used to train a state-variable feedback control policy. The feature space for the supervised learning includes parameters from a reduced-order biped model (e.g., initial stance leg angle and average speed), exogenous signals (target walking speed is used here, but turning angle could be used as well) and perception input (e.g., terrain height or slope). This policy is compared with a testing set of optimal gaits in simulation and is subsequently evaluated on the 3D underactuated robot MARLO. In a simulation of stepping in place, the learned policy takes at most one more step than an optimal gait to recover from initial velocity and position errors. In experiments, the learned policy allows MARLO to recover from $\approx$ 200 N kick. It also enables MARLO to walk down a 22 deg slope and walk on the Wave Field, which presents sinusoidally varying ground height (see Fig. 1).

*The authors are with College of Engineering at University of Michigan, Ann Arbor, MI, USA, 48109. {xda, rosshart, grizzle}@umich.edu

A. Stability

In the course of this work, it will be shown that stability can be lost when applying supervised learning to a training set of locally exponentially stable controllers. This observation and a *suggestion* for how to avoid instability can help to explain recent work of NVIDIA, where supervised learning was applied in an "end-to-end" fashion to design controllers for a self-driving car [3]. In NVIDIA's on-road training process, a human driver plays the role of an exponentially stabilizing controller, while constant speed, center of the lane driving is analogous to the periodic gaits studied in this paper. NVIDIA required an offline handcrafted step to stabilize their steering to the center of the lane: they did that by using off-center camera views and computer-generated corrective steering actions. An analogous handcrafted approach (foot placement design) was discussed in our previous work [4]. Importantly, in this paper, we eliminate this handcrafted step by enriching the training set through aperiodic gaits that

Fig. 1: Bipedal robot MARLO walked on the University of Michigan's Wave Field, a sinusoidally varying grass terrain. Photo was taken by Roger Hart.

represent control solutions that steer back to equilibria and reject disturbances, which speak to the essence of stability.

B. Literature Overview

One of the earliest applications of online optimization in bipedal walking was done on a 5-degree-of-freedom simulation model of RABBIT [5], [6]; the computation time for each sampling period was 37.08 s. More recently, Model Predictive Control (MPC) was applied in the DARPA Virtual Robotics Challenge [7]. In that work, the computation time of the MPC solver was important, and a "real-time implementation" on a full-order dynamic model of Atlas was achieved through the use of a novel physics engine and a relaxed contact map. Experimental results on a humanoid robot HRP-2 were reported in [8]. The robot did not walk, but could balance while standing and track a ball with its hands. MPC was applied to the full kinematics and centroidal dynamics of Atlas in [9], and 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 are achieved in [10] using online Hybrid Zero Dynamics (HZD) gait generation. The online optimization generates a new controller based on the commanded speed and updates it at the beginning of the next step. Average computational time is 0.4964 s.

Online computational burden has been reduced by using reduced-order models to compute CoM trajectories and swing foot positions. 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 [11], [12], [13]. Though a reduced-order model may provide fundamental insight into the dynamics of a robot [14], it limits the achievable motions of the robot, and different tasks, such as walking and running, typically require different models.

Another means to get around the limitations of online computation is to pre-compute a set of controllers and design a control policy to "stitch" them together. The most common policies in the literature involve switching, finite-state machines, and interpolation. Switching based on only the commanded task (target walking speed, running vs walking, stairs vs flat ground) is used in [15], [16], [17]. A hand-designed, finite-state machine is used in [18] for rough terrain. More sophisticated finite-state-machines are designed using offline reinforcement learning to handle rough terrain [19] and to reduce settling time to a commanded walking speed [20]. Interpolation has been used to design transition gaits among a finite set of controllers for walking at constant speeds in [16] and to create a continuous family of gaits in [21], [4]. Supervised learning has been applied in [22], [23] for gait synthesis; experimental results for quasi-static walking is reported in [24]. Nearest neighbor is used in [25] to enlarge the basin of attraction. A review of machine learning algorithms in bipedal robot control is given in [26].

C. Contributions of the Paper

In many cases, it is computationally expensive to build a good training set for supervised learning [26]. In previous work [4], parameter optimization and virtual constraints are used to design a set of controllers for fixed speeds and a simple interpolation method was used to build a control policy. Here, supervised learning is used to build the control policy from a larger family of controllers, including those for aperiodic walking.

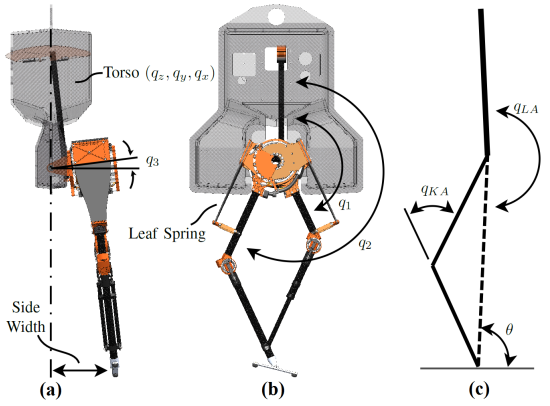


Fig. 2: Biped coordinates. (a) Lateral plane. (b) Sagittal plane. (c) Equivalent sagittal model.

The novel contributions of the work include:

- using supervised learning to approximate the optimal gaits from a finite set;
- the training and testing sets are selected from controllers that induce periodic gaits, aperiodic gaits that effect transitions among a subset of the periodic gaits, and perturbations of periodic gaits;
- the feature space for the supervised learning is richer than standard reduced-order models; indeed it includes initial conditions from a reduced-order biped model, exogenous command or reference signals, and quantities deduced from onboard sensors;
- suggestions are made that relate closed-loop stability to properties of both the training and feature sets;
- compared to previous work in [4], this control policy significantly improves the ability to reject perturbations and to walk on uneven terrain.

II. ROBOT DESCRIPTION

A. Robot Configuration

The bipedal robot shown in Fig. 2, called MARLO, is the Michigan copy of an ATRIAS series robot and is capable of 3D walking. The configuration variables for the robot can be defined as $q^{3D} := (q_z, q_y, q_x, q_{1R}, q_{2R}, q_{3R}, q_{1L}, q_{2L}, q_{3L}) \in \mathbb{R}^9$ where the leaf springs are sufficiently stiff and have been deliberately neglected from the model. The variables (q_z, q_y, q_x) correspond to the world frame rotation angles: yaw, roll, and pitch; the variables $(q_{1R}, q_{2R}, q_{3R}, q_{1L}, q_{2L}, q_{3L})$ refer to local coordinates. These local coordinates are each actuated by a DC motor, resulting in 6 degrees of actuation $u \in \mathbb{R}^6$ and 3 degrees of underactuation. A more complete description is available in [27].

B. Planar Representation

All optimization and control policy designs in this paper are based on a planar model for simplicity and a fast optimizer of Ames's group was not yet available [28]. Experiments on the 3D robot are done by augmenting a controller designed on a planar model with a lateral controller given in [4]. A planar representation is obtained from the 3D model by constraining $(q_y, q_z, q_{3L}, q_{3R})$ to zero, [27, Sec. 4.5]. The remaining configuration variables $q := (q_x, q_{1R}, q_{2R}, q_{1L}, q_{2L}) \in \mathbb{R}^5$ can also be written as $q := (\theta, q_{LA}^{\text{right}}, q_{LA}^{\text{left}}, q_{KA}^{\text{right}}, q_{KA}^{\text{left}})$ for control purposes, where the leg angles are $q_{LA} := \frac{1}{2}(q_1 + q_2)$ and knee angles are $q_{KA} := q_2 - q_1$. The absolute stance leg angle θ is underactuated.

III. CONTROL POLICY OVERVIEW

The control policy proposed here relates a vector of features to a set of control parameters. This policy will be constructed using supervised learning techniques from a carefully designed training dataset. The process includes:

1) choosing features and control parameters;
2) generating the datasets through optimization;
3) fitting the control policy using a training set with supervised learning algorithms; and
4) assessing the policy with a testing set and simulations.

The steps are specified in the following sections for individual policies. Figure 3 shows an overview of the policy design process.

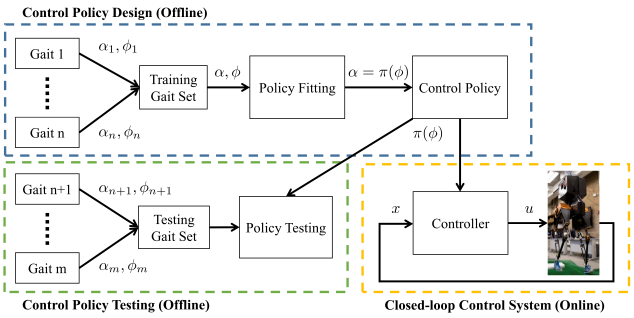


Fig. 3: Control Policy Design and Implementation

A. Control Policy

A control policy $\pi : \Phi \to \mathcal{A}$ is a function that maps a feature vector $\phi \in \Phi$ to a vector of control parameters $\alpha \in \mathcal{A}$. In this paper, α is a set of Bézier coefficients inducing a desired trajectory, $q^d(t)$. A low-level feedback controller is then used to minimize the tracking error. The specifics of the feedback controller derivation are given in previous work [4]. The focus of this paper is to build the control policy.

B. Dataset Generation Through Optimization

Parameter optimization [2, Sec. 6.3] is used to build a dataset for supervised learning. Each optimization provides a single dynamically feasible path $q^d(t)$ over one or more steps. α and ϕ are extracted at each step. Here, the dataset is constructed from as few as seven to as many as a hundred optimizations, selected to represent the small number of behaviors that the control policy is to learn. All optimizations are set up to respect constraints given in Table I and to minimize the sum of squared torques. Other constraints implemented depend on the nature of the control policy that is to be learned.

TABLE I: Optimization constraints

Motor Toque $\lvert u \rvert$	< 5 Nm
Step Duration T	$= 0.35$ s
Friction Cone μ	< 0.6
Impact Impulse Fe	< 15 Ns
Vertical Ground Reaction Force	> 300 N
Mid-step Swing Foot Clearance	> 0.18 m

C. Machine Learning Methods

Once the dataset has been generated, various machine learning techniques can be used to regress the control policy $\pi(\cdot)$. This paper compares three fitting methods: linear interpolation (LI), support vector machines (SVMs), and neural networks (NNs). The three methods show similar performance in fitting quality and speed tracking. Detailed discussion is available in the simulation section.

1) Linear Interpolation: When the feature ϕ is a scalar, linear interpolation can be used as

$$\pi_{LI}(\phi) = (1 - \zeta(\phi))\alpha_i + \zeta(\phi)\alpha_{i+1} \tag{1}$$

$$\zeta(\phi) = \frac{\phi - \phi_i}{\phi_{i+1} - \phi_i}, \tag{2}$$

where ϕ_i and ϕ_{i+1} are features in the training set between the input ϕ. It can be extended to bilinear interpolation (BiLI) if the feature has two variables. Since the method only uses local data, it is good to fit an evenly distributed data set.

2) Support Vector Machines: Support vector machines (SVMs) are a common ML technique that can be used for function regression (also known as SVR). The SVM algorithm can be used to regress a nonlinear function by applying the "kernel trick" [29]. In this paper, the regression was learned using the LIBSVM toolbox with the radial basis function kernel.

3) Neural Networks: Neural Networks (NNs) are an increasingly used method for nonlinear function approximation. They rely on a series of connected "neurons", usually sigmoid functions, and a set of weights that can be learned [30]. In this paper, the learning is implemented using MATLAB's Neural Network Toolbox with 5 hidden layers. The networks are trained using the default Levenberg-Marquardt algorithm.

D. Training and Testing

The training and testing datasets are built separately. For each of the learning algorithms listed, a control policy $\pi(\cdot)$ is learned using only the training dataset. Each resulting control policy is assessed using the separate testing dataset. The coefficient of determination (R^2) and the root mean square

error (RMSE) provide one way to evaluate how closely the output of the control policy matches the test data. The utility of the control policy is further verified by running simulations.

IV. SPEED REGULATION POLICY

Previous work in [4] designed a gait library for speed tracking via optimization. The discrete set of gaits was then interpolated to produce a continuously defined feedback controller. Even when the discrete gaits were (locally) exponentially stable, the resulting closed-loop system was at best neutrally stable. Subsequently, exponential stability was recovered with a supplemental foot placement policy, which allowed MARLO to walk forwards and backwards at a variety of speeds. This section reformulates the design procedure as a supervised learning problem.

A. Dataset Generation

To generate the training dataset, 13 separate parameter optimizations are run. Each optimization generates a periodic gait at different sagittal velocities v_{avg}. The set of gaits is denoted by

$$\mathcal{A}_{\text{train}} = \{\alpha(v_{\text{avg}}) \mid -1.2 \leq v_{\text{avg}} \leq 1.2\}, \tag{3}$$

where v_{avg} increases in steps of 0.2 m/s. A similar testing set of gaits $\mathcal{A}_{\text{test}}$ is designed for the same speed range but at a finer grid of 0.05 m/s. The optimization is set up to respect constraints given in Table I and to minimize the sum of squared torques. Additional constraints for periodicity and the average velocity are also included.

B. Feature Selection

The only difference among the optimizations is the average velocity. Therefore, a logical feature choice is $\phi = \{v_{\text{avg}}\}$.

C. Training Methods

Since ϕ is a scalar quantity, linear interpolation (LI) can be used to fit the control policy $\pi(\cdot)$. This is what was used in [4]. For comparison, support vector machines (SVMs) and neural networks (NNs) are also used.

D. Stability Remark

When SVM and NNs are used in place of LI, the closed-loop system is also at best neutrally stable. As in [4], this is checked with a Poincaré map.

V. TRANSITION GAIT POLICY

The speed regulation policy discussed in the last section "teaches" MARLO how to walk along a steady state, periodic gait. This section proposes a novel optimization setup to add the transitions between various periodic gaits into the learning process.

A. Dataset Generation

Let $x_i := [q, \dot{q}]_i^\top$ and $x_j := [q, \dot{q}]_j^\top$ be two points in the robot's state space corresponding to double support. Denote by $\alpha^{x_i \to x_j}$ the control parameters, if they exist, that effect a transition in one step from x_i to x_j. When $x_i = x_j$, we have a periodic gait, and we also denote the control parameters by $\alpha(v^i_{\text{avg}})$, as one of the element in (3), inducing a periodic gait at velocity v_{avg}. The corresponding state is denoted by $x^*(v^i_{\text{avg}})$.

To handle a wide range of transitions, we also consider the case where two points in the state space cannot be joined in one step. Specifically, given two points x_i and x_j, we also design controllers that effect transitions in three steps[1]. Optimization is used to compute two intermediate states $x_a^{i\to j}$ and $x_b^{i\to j}$, and corresponding control parameters, such that, the robot transitions are

$$x_i \to x_a^{i\to j} \to x_b^{i\to j} \to x_j. \tag{4}$$

In the language of capture points [32], [33] , x_i above is in the 3-step viable-capture basin of x_j. The 3-step transition gaits are computed for

$$x_j = x^*(v^j_{\text{avg}}), \text{for } v^j_{\text{avg}} \in \{-0.4, -0.2, 0, 0.2, 0.4\} \tag{5}$$

$$x_i \in \{x^*(v^i_{\text{avg}}) \mid -0.6 + v^j_{\text{avg}} \leq v^i_{\text{avg}} \leq 0.6 + v^j_{\text{avg}}\}. \tag{6}$$

When $v^i_{\text{avg}} = v^j_{\text{avg}}$, it is noted that $\alpha^{x_i \to x_j} = \alpha(v^i_{\text{avg}})$, the control parameters for the periodic gait at speed v^i_{avg}, given in (3). To be clear, each 3-step optimization provides three controllers that are included in the training set.

In this initial study on supervised learning, the testing set focuses on stepping in place. The three-step optimization process as in (4) is used to compute controllers given the terminal point $x_j = x^*(0)$ (stepping in place) and initial points x_i that has perturbations of stepping in place. These perturbations correspond to the robot being in double support, in which the support leg angle θ is perturbed $\pm 15\,\text{deg}$, the support leg angle rate $\dot{\theta}$ is perturbed $\pm 34\,\text{deg}/s$, and the swing leg angle rate $\dot{q}^{sw}_{LA}$ is perturbed $\pm 114\,\text{deg}/s$, all independently.

[1]The number three is motivated by [31]. In case the transition can be done in two steps, $x_b^{i\to j} = x_j$; similarly for one step.

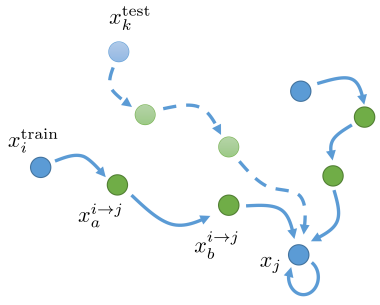


Fig. 4: A graph of three-step optimization. Given x_i and x_j, the optimization will find a path $x_i \to x_a^{i\to j} \to x_b^{i\to j} \to x_j$ if exists. Blue dots are specified in the optimization while green dots and path are generated from the optimization.

B. Feature Selection

In the speed regulation policy design, the only changed optimization constraint is the average speed. This led to a logical choice of the feature being v_{avg}. In contrast, when optimizing transition gaits, all of the states change. The feature vector could potentially use the full states, but this may require a large training dataset. Instead, a small set of features $\phi = \{v_{\text{avg}}, \theta_{\text{init}}, v_{\text{tgt}}\}$ is proposed. Inspired from the inverted pendulum model, these features capture the two crucial underactuated degrees of freedom as well as the target velocity. Kernel principal component analysis (PCA) may be used in the future to find a low dimensional representation of the state space to extract features from.

C. Training Methods

The transition control policies are trained using SVMs and NNs. These policies are assessed by simulating from the initial states x_i in the testing dataset. The simulation results are compared against the optimized gaits in the testing dataset.

D. Stability Remark

For periodic gaits, current and desired speed are identical. For transitioning among periodic gaits, or when rejecting a perturbation, these two speeds are different which is why we design aperiodic gaits. With the richer feature set $\{v_{\text{avg}}, v_{\text{tgt}}\}$, and with the richer training set, {periodic, aperiodic}, Poincaré analysis verifies that (local) exponential stability is recovered.

VI. TERRAIN ADAPTION POLICY

This section adds periodic gaits for different terrain heights or slopes to design a terrain adaption policy. It will enhance the speed tracking performance on sloped terrain and robustness over uneven terrain.

Since the MARLO does not have any vision sensors to foresee the terrain, proprioceptive sensors are used to measure the positions of the feet in the double support phase to estimate the terrain profiles.

A. Dataset Generation

To generate the datasets, a 2D grid of gaits is optimized. The training dataset includes gaits where v_{avg} ranges [-1.2, 1.2] m/s in 0.2 m/s steps, and h ranges [-0.1, 0.1] m in 0.05 m steps. The testing dataset is designed on the same range but at a finer grid, 0.1 m/s increments for v_{avg} and 0.02 m increments for h.

B. Feature Selection

1) Sagittal Terrain Adaption: Since the dataset was generated using varying velocities and step heights, the empirical choice for the feature vector is $\phi = \{v_{\text{avg}}, h\}$. The controller is designed using the planar model, thus h is measured as sagittal terrain height.

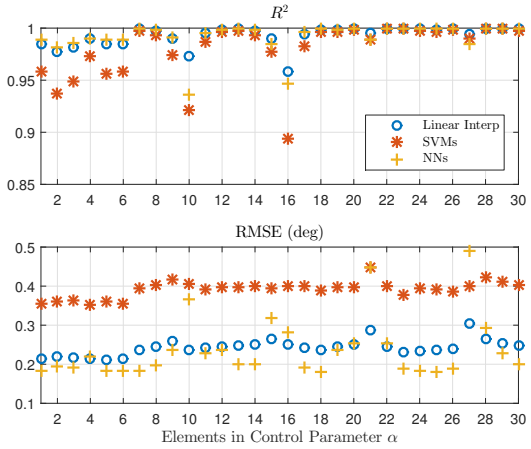


Fig. 5: Fitting quality of each element in control parameters α. Every six of them construct a trajectory of configure variable in $q^d(t)$

2) Lateral Terrain Adaption: In the 3D model, the feet height and side width in the double support phase can be also used to estimate the lateral terrain slope β_{lateral}. This paper shows the preliminary use of this feature in the experiments Section VIII-C. More sophisticated terrain profile estimation could be used, though the design of control policy remains the same.

C. Training Methods

Since the dataset was constructed uniformly on a grid, a simple bilinear interpolation can be implemented. More advanced regression methods (SVMs, NNs, etc.) could be used, but the authors did not pursue them for this control policy. However, a unified control policy, which is described in Section VIII-D, is fit using a neural network. It combines terrain adaption with speed regulation and transition gaits

VII. SIMULATION

The control policies are evaluated in two ways: comparing the control parameters with the testing data, and assessing the control performance in simulated planar walking.

A. Speed Regulation Policy

The speed regulation policy is generated by three regression methods: π_{LI}, π_{SVM} and π_{NN}. The elements in α show a strong correlation with the testing data in Fig. 5, where the lowest coefficient of determination R^2 is 0.9 and the biggest root mean square error (RMSE) is 0.5 deg. These 30 elements are 5 sets of Bézier coefficient that induce $q^d(t)$. The biggest RMSE error of $q^d(t)$ between the control policies and optimization is 0.4 deg, where the position tracking error in experiments is 5 deg on average. The control policies are subsequently evaluated by tracking a target velocity in Fig. 6. The three methods give consistent results indicating that the supervised learning approach proposed in this paper is not limited to a certain method. Small speed tracking error comes from a low-level feedback controller.

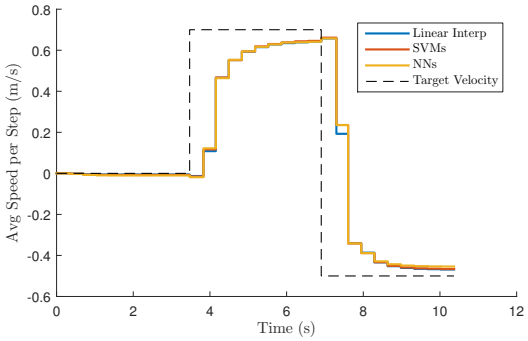
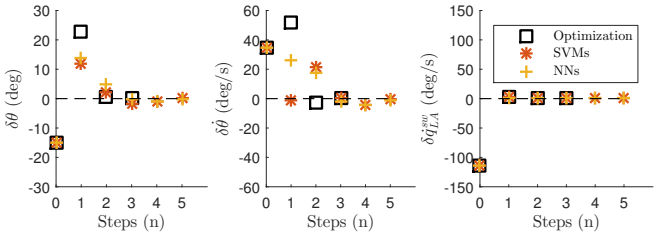


Fig. 6: All three fitting methods show consistent speed tracking performance indicating the fitting method is not limited to any specific one.



(a) (b) (c)

Fig. 7: (a) has 15 deg initial error on θ. (b) has 34 deg$/s$ error on $\dot{\theta}$. (c) has 114 deg$/s$ error on q_{LA}^{sw}. The transition policy takes at most one more step than the gaits from optimization to recover from these initial errors.

B. Transition Policy

The fitting quality of the transition policy is deteriorated because the features are extracted from a reduced order model in Section V-B, where the biggest RMSE is 8 deg. Due to page limitations, more discussion will be included in a journal paper. The control policies are analyzed by simulating from the three largest initial state perturbations in the testing dataset $\{\delta\theta = -15 \text{ deg}, \delta\dot{\theta} = 34 \text{ deg}/s, \delta\dot{q}_{LA}^{sw} = -114 \text{ deg}/s\}$, shown in Fig. 7. The three-step optimization gives optimal controllers that converge within three steps. The control policy learned from the training set takes at most one more step to recover.

The transition policy is compared with the speed regulation policy through a perturbation rejection test. The push force is 200 N in one step (0.35 s), shown in Fig. 8. The transition policy converges back to the target velocity faster and with less overshoot.

C. Terrain Policy

Since the training set includes gaits that function correctly for sloped ground, the terrain adaption policy improves speed regulation on both uphill and downhill walking. In Fig. 9, both the speed regulation policy and terrain adaption policy are applied to walking downhill and uphill. The 10 degree slope is the steepest that the speed regulation policy can handle, though it gains considerable speed going downhill. In contrast, the terrain adaption policy maintains roughly the same velocity throughout.

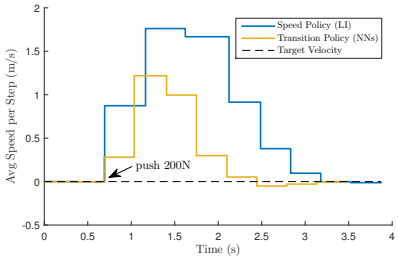


Fig. 8: After subjecting to a 200N push in one step (0.35 s), the transition policy has shorter settling time and smaller overshoot than the speed policy.

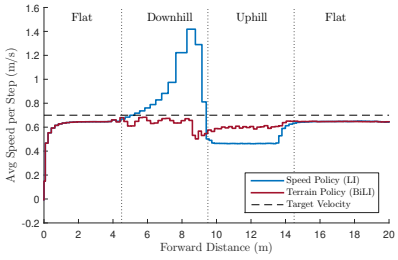


Fig. 9: The terrain adaption policy chooses a controller based on the current velocity and terrain height, which gives a more consistent speed tracking result than the speed regulation policy. The slope is ± 10 deg.

VIII. EXPERIMENTS AND DISCUSSION

For simplicity, the supervised learning policies presented in Sections III - VII concern the planar model of MARLO. The control polices are augmented with a lateral controller as in [4], [34] for implementation on the physical 3D robot. Controllers for speed regulation were presented in [4]. The new experiments for this paper are numbers 3 - 11 in Table II.

A. Speed Regulation and Transition

To understand the utility of including the transition gaits in the learning sets, a first control policy is designed using only periodic gaits (see Section IV). The asymptotic stability of the closed-loop system is assured with a foot placement controller given in [4], [34]. A second policy is then designed using the same set of periodic gaits augmented with transitions (see Section V), no extra controller is needed. Transition gaits represent transient conditions that have been designed to respect the physical limitations of the robot, and hence the resulting control policy is better able to avoid foot slippage during transients than the policy built on steady-state (periodic) walking. This is demonstrated by comparing the light push in Experiment 2 to the much stronger kick in Experiment 3.

B. Sagittal Terrain Adaptation

Experiment 1 uses the speed controller, without transitions, discussed above. At the end of Experiment 1, MARLO encounters a 7 deg upward slope, slips, and falls. A new policy is designed that focuses on ground slope changes (see Section VI-B.1), and is used in Experiments 4 and 5. Experiment 4 demonstrates the robot walking down a long,

TABLE II: Experiment Videos

Number	Gait Policy	Experiment	Link
1	Speed Regulation [4]	A Long Walk	https://youtu.be/eSllkIptlK0
2	Speed Regulation [4]	Light Push	https://youtu.be/iOltRR0RqiM
3	Transition	Kick MARLO	https://youtu.be/YXJQJtcXX4E
4	Sagittal Terrain	Walking Down 22 Degree Slope	https://youtu.be/gHpXTmyG4mE
5	Sagittal Terrain	Random Terrain	https://youtu.be/iW9SWPQmYh0
6	Sagittal Terrain	Wave Field (First Attempt)	https://youtu.be/YErF0cyPI-g
7	Lateral Terrain	Practice for the Wave Field	https://youtu.be/vEQa1e7lzjQ
8	Lateral Terrain	Wave Field (Second Attempt)	https://youtu.be/TDFz_0Avc2A
9	A Unified Policy	Walking	https://youtu.be/xPHMgFiSeu0
10	A Unified Policy	Pushing, Random Terrain	https://youtu.be/VovWti_wKRU
11	A Unified Policy	Walking in the Forest	https://youtu.be/uYD99f01aek

22 deg, steep slope. The average walking speed is about 0.2 m/s; the safety gantry gets stuck at several points, keeping the average speed quite low. Walking up the slope has not been demonstrated because pushing the gantry up a steep hill is impossible. Walking down is often more challenging because the robot will gain speed from gravity. Even though the control policy was designed for constant slopes, in Experiment 5 the robot is challenged to walk indoors over randomly varying terrain. In these experiments, the ground slope is estimated by relative foot height during double support, which is one of the features used in determining the control policy for the next step. If the terrain changes dramatically over a step, a camera is needed to preview the terrain and this information must be added to the feature set during supervised learning.

C. Lateral Terrain Adaptation

Without a camera, and using only relative foot height information in double support, it is not possible to distinguish between a slope in the sagittal direction, the lateral direction, or a combination. Using the same control policy as in Experiments 4 and 5, the robot was taken to the Wave Field on the University of Michigan Campus; see Experiment 6. The most frequent failure mode was the robot's swing leg hitting the ground prematurely because, when moving the leg laterally, it assumed the slope was zero. A new policy was designed under the assumption the relative changes in foot height are due to a lateral slope only (in the sagittal direction, the ground is assumed flat); see Section VI-B.2. Experiment 7 tests the control policy indoors and Experiment 8 is performed on the Wave Field. In the latter, the robot is able to make two complete passes in the troughs between the crests, whereas in Experiment 6, it never made it more than half way down any one of them.

D. Unified Policy

Here, the control policy is designed using periodic gaits, transition gaits among a subset of them, and terrain slope changes in the sagittal direction. In Experiment 9, MARLO walks outdoors at speeds varying from standing to 0.5 m/s. In Experiment 10, the robot traverses a pile of rubble in the laboratory. When taken to a a section of woods on the campus in Experiment 11, the robot walks down sloped terrain, covered with branches, and encounters stumps. After about five minutes, MARLO trips on a lateral slope because of the same failure mechanism in Experiment 6: when moving the swing leg laterally, premature impact with the ground occurs.

IX. CONCLUSIONS AND NEXT STEPS

Supervised learning was used to design control polices for the complete planar model of an underactuated 3D bipedal robot. The training and testing sets included periodic gaits on flat and sloped ground and transition gaits. The control policy designed with supervised learning increased the robustness of the robot's gait in comparison to previous control solutions that focused on asymptotically stable walking at a constant speed [35], or a solution built by interpolating controllers from a library of such gaits.

Part of the enhanced robustness comes from including transient control solutions in the training set. These provide a means for returning to a target speed after a perturbation, while satisfying constraints on peak torque, friction cone and motor speed. Additional robustness comes from including gaits that functioned correctly on sloped ground. The supervised learning formulation allowed a collection of behaviors to be addressed in a unified manner, when the feature set was expanded to include initial states of a reduced-order model, exogenous command signals, and terrain information gleaned from sensors.

Future work includes extending the method to address the full 3D dynamic model of the robot. This was not done here because, when the work was initiated, the fast optimizer of Ames's group was not yet available [28]. A camera has been purchased for the robot to allow more sophisticated terrain information to be included in the feature set. To date, a very small number of controllers has been used in the training sets. It will be interesting to explore the utility of including many more periodic and aperiodic solutions for different dynamic behaviors. When this advance is taken, it is worthwhile to use automatic feature extraction tools.

ACKNOWLEDGMENT

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 help of PhD student Omar Harib and Post doc Brent Griffin (Univ. of Michigan). The work in this paper is supported by the National Science Foundation through NSF grants EECS-1525006, ECCS-1343720 and ECCS-1231171.

REFERENCES

[1] U. Maeder, R. Cagienard, and M. Morari, "Explicit model predictive control," in *Advanced strategies in control systems with input and output constraints*, pp. 237–271, Springer, 2007.

[2] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Control and Automation, Boca Raton, FL: CRC Press, June 2007.

[3] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[4] X. Da, O. Harib, R. Hartley, B. Griffin, and J. W. Grizzle, "From 2D design of underactuated bipedal gaits to 3D implementation: Walking with speed tracking," *IEEE Access*, vol. 4, pp. 3469–3478, 2016.

[5] C. Azevedo, P. Poignet, and B. Espiau, "Moving horizon control for biped robots without reference trajectory," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 3, pp. 2762–2767, 2002.

[6] C. Azevedo, P. Poignet, and B. Espiau, "Artificial locomotion control: from human to robots," *Robotics and Autonomous Systems*, vol. 47, no. 4, pp. 203–223, 2004.

[7] T. Erez, K. Lowrey, Y. Tassa, V. Kumar, S. Kolev, and E. Todorov, "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, Oct 2013.

[8] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the HRP-2 humanoid," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 3346–3351, IEEE, 2015.

[9] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[10] A. Hereid, S. Kolathaya, and A. D. Ames, "Online hybrid zero dynamics optimal gait generation using legendre pseudospectral optimization," in *To appear in: IEEE Conference on Decision and Control (CDC)*, IEEE, 2016.

[11] J. Pratt, T. Koolen, T. de Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based analysis and control of legged locomotion, Part 2: Application to M2V2, a lower-body humanoid," *The International Journal of Robotics Research*, pp. 1117–1133, Aug.

[12] M. Krause, J. Englsberger, P.-B. Wieber, and C. Ott, "Stabilization of the capture point dynamics for bipedal walking based on model predictive control," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 165–171, 2012.

[13] S. Faraji, S. Pouya, C. G. Atkeson, and A. J. Ijspeert, "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)*, pp. 1943–1950, IEEE, 2014.

[14] R. Full and D. Koditschek, "Templates and anchors: Neuromechanical hypotheses of legged locomotion on land," *Journal of Experimental Biology*, vol. 202, pp. 3325–3332, December 1999.

[15] K. Sreenath, H.-W. Park, I. Poulakakis, and J. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on mabel," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 324–345, 2013.

[16] A. E. Martin, D. C. Post, and J. P. Schmiedeler, "Design and experimental implementation of a hybrid zero dynamics-based controller for planar bipeds with curved feet," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 988–1005, 2014.

[17] M. J. Powell, A. Hereid, and A. D. Ames, "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*, pp. 4803–4810, IEEE, 2013.

[18] H. Park, A. Ramezani, and J. W. Grizzle, "A finite-state machine for accommodating unexpected large ground height variations in bipedal robot walking," *IEEE Transactions on Robotics*, vol. 29, no. 29, pp. 331–345, 2013.

[19] C. O. Saglam and K. Byl, "Meshing hybrid zero dynamics for rough terrain walking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5718–5725, IEEE, 2015.

[20] S. Apostolopoulos, M. Leibold, and M. Buss, "Settling time reduction for underactuated walking robots," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 6402–6408, IEEE, 2015.

[21] K. R. Embry, D. J. Villarreal, and R. D. Gregg, "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, 2016.

[22] J.-G. Juang and C.-S. Lin, "Gait synthesis of a biped robot using backpropagation through time algorithm," in *Neural Networks, 1996., IEEE International Conference on*, vol. 3, pp. 1710–1715, IEEE, 1996.

[23] J.-G. Juang, "Locomotion control using environment information inputs," in *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*, pp. 196–201, IEEE, 1999.

[24] J. P. Ferreira, M. Crisostomo, A. P. Coimbra, and B. Ribeiro, "Simulation control of a biped robot with support vector regression," in *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, pp. 1–6, IEEE, 2007.

[25] C. Liu, C. G. Atkeson, and J. Su, "Biped walking control using a trajectory library," *Robotica*, vol. 31, no. 02, pp. 311–322, 2013.

[26] S. Wang, W. Chaovalitwongse, and R. Babuska, "Machine learning algorithms in bipedal robot control," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 5, pp. 728–743, 2012.

[27] A. Ramezani, J. W. Hurst, K. Akbari Hamed, and J. W. Grizzle, "Performance Analysis and Feedback Control of ATRIAS, A Three-Dimensional Bipedal Robot," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 2, 2014.

[28] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "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)*, 2016.

[29] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, ACM, 1992.

[30] H. Demuth and M. Beale, "Neural network toolbox for use with matlab," 1993.

[31] P. Zaytsev, S. J. Hasaneini, and A. Ruina, "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, May 2015.

[32] J.-P. Aubin, J. Lygeros, M. Quincampoix, S. Sastry, and N. Seube, "Impulse differential inclusions: a viability approach to hybrid systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 1, pp. 2–20, 2002.

[33] J. Pratt and R. Tedrake, "Velocity-based stability margins for fast bipedal walking," in *Fast Motions in Biomechanics and Robotics* (M. Diehl and K. Mombaur, eds.), vol. 340 of *Lecture Notes in Control and Information Sciences*, pp. 299–324, Springer Berlin Heidelberg, 2006.

[34] S. Rezazadeh, C. Hubicki, M. Jones, A. Peekema, J. Van Why, A. Abate, and J. W. Hurst, "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)*, p. 23, 2015.

[35] B. G. Buss, K. A. Hamed, B. A. Griffin, and J. W. Grizzle, "Experimental results for 3D bipedal robot walking based on systematic optimization of virtual constraints," in *American control conference*, 2016.