arXiv:1910.01748v1 [cs.RO] 3 Oct 2019

Hybrid Zero Dynamics Inspired Feedback Control Policy Design for 3D Bipedal Locomotion using Reinforcement Learning

Guillermo A. Castillo[1], Bowen Weng[1], Wei Zhang[2], and Ayonga Hereid[3]

Abstract— **This paper presents a novel model-free reinforcement learning (RL) framework to design feedback control policies for 3D bipedal walking. Existing RL algorithms are often trained in an end-to-end manner or rely on prior knowledge of some reference joint trajectories. Different from these studies, we propose a novel policy structure that appropriately incorporates physical insights gained from the hybrid nature of the walking dynamics and the well-established hybrid zero dynamics approach for 3D bipedal walking. As a result, the overall RL framework has several key advantages, including lightweight network structure, short training time, and less dependence on prior knowledge. We demonstrate the effectiveness of the proposed method on Cassie, a challenging 3D bipedal robot. The proposed solution produces stable limit walking cycles that can track various walking speed in different directions. Surprisingly, without specifically trained with disturbances to achieve robustness, it also performs robustly against various adversarial forces applied to the torso towards both the forward and the backward directions.**

I. Introduction

3D bipedal walking is a challenging problem due to the multi-phase and hybrid nature of legged locomotion. Properties like underactuation, unilateral ground contacts, nonlinear dynamics, and high degrees of freedom significantly increase the model complexity. Existing approaches on bipedal walking can be roughly grouped into two categories: model-based and model-free methods. In [1], the authors provide a comprehensive review of model-based methods, feedback control, and open problems of 3D bipedal walking. One of the main challenges for model-based methods is the limitation of mathematical models that capture the complex dynamics of a 3D robot in the real world. This results in non-robust controllers that require additional heuristic compensations and tuning processes, which can be time-consuming and requires experiences.

Reduced order models, such as Linear Inverted Pendulum and its variants [2], have been studied extensively in the literature. For these simple models, stable walking conditions can be stated in terms of the ZMP (zero moment point) [3]–[5] or CP (capture point) [6], [7], which can significantly simplify the control design. However, these approaches rely on some strong assumptions that often lead to quasi-static and unrealistic walking behaviors. Optimization-based methods such as Linear Quadratic Regulator (LQR) [8], Model Predictive Control (MPC) [9], [10], and Hybrid Zero Dynamics (HZD) [11], [12] use the full order model of the robot to capture the underlying dynamics more accurately, which yields more natural dynamic walking behaviors. In particular, HZD is a formal framework for the control of bipedal robots with or without underactuation through the design of nonlinear feedback controllers and a set of virtual constraints. It has been successfully implemented in several physical robots, including many underactuated robots [13]–[16]. Nonetheless, these methods are computationally expensive and sensitive to model parameters and environmental changes. Particularly for 3D walking, additional feedback regulation controllers are required to stabilize the system [17]–[19]. Notably, recent work has successfully realized robust 3D bipedal locomotion by combining Supervised Learning with HZD [20].

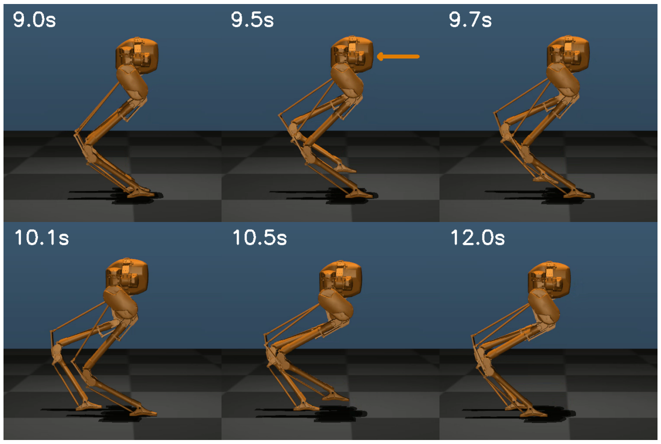

Fig. 1: Cassie in simulation: gait recovering from the backward adversarial force with our proposed method.

With recent progress on deep learning, Reinforcement Learning (RL) has become a popular tool in solving challenging control problems in robotics. Existing RL methods often rely on end-to-end training without considering the underlying physics of the particular robot. A NN function is trained with policy gradient methods that directly maps the state space to a set of continuous actions [21], [22]. Despite the empirical success, such methods are often sampling inefficient (millions of data samples) and are usually over-parameterized (thousands of tunable parameters). They may also lead to non-smooth control signals and unnatural motions that are not applicable to real robots. Through incorporating HZD with RL training, [23] generated feasible trajectories that are tracked by PD controllers to produce

*This work was supported in part by the National Science Foundation under grant CNS-1552838 and the OSU M&MS Discovery Theme Initiative.

[1]Electrical and Computer Engineering, Ohio State University, Columbus, OH, USA; {castillomartinez.2, weng.172}@osu.edu.

[2]SUSTech Institute of Robotics, Southern University of Science and Technology, China; zhangw3@sustech.edu.cn.

[3]Mechanical and Aerospace Engineering, Ohio State University, Columbus, OH, USA. hereid.1@osu.edu.

sustainable walking gaits at different speeds. However, this method only works for a simple 2D robot model. For the more complex 3D bipedal walking, [24] adopts RL methods as part of the feedback control. The method relies on prior knowledge of a good joint reference trajectory and only learns small compensations added to the known reference trajectory, which does not provide the overall control solution. An imitation learning inspired method is proposed for a 3D robot [25], which also requires a known walking policy and gradually improves the policy through learning.

In this paper, we propose a novel hybrid control structure for robust and stable 3D bipedal locomotion. It harnesses the advantages of parameterized policies obtained through RL while exploiting the structure of the intuitive yet powerful additional regulations commonly used in 3D bipedal walking. The regulation terms embedded in the design of policy structure and reward functions differentiate our proposed method from the previous work of [23] that is also inspired by HZD. We evaluate the performance of our method on Cassie, which is also a much more challenging robot than the 2D Rabbit [23]. We further summarize the primary contributions of the present paper as follows:

- **Model-free:** The trained policy naturally learns a feasible walking gait from scratch without the need of a given reference trajectory or the model dynamics. To the best of our knowledge, this is the first time that a variable speed controller for a 3D robot is learned without using previously known reference trajectories or training separate policies for different speeds.
- **Efficiency:** By incorporating the physical insight of bipedal walking, such as its hybrid nature, symmetric motion, and heuristic compensation, into the control structure and learning process, we significantly simplify the design to a shallow neural network (NN) with only 5069 trainable parameters. To the best of our knowledge, this is the smallest NN ever reported for Cassie. As a result, the NN policy is easy to train. It is also fast enough for real-time control with 1 kHz frequency on single process CPU (the low-level PD controller runs at 2 kHz).
- **Robustness:** The learned controller is capable of stably tracking a wide range of walking speeds in both longitudinal and lateral directions with just one trained policy. The robustness of the controller is also evaluated by several disturbance rejection tests in simulation.

II. PROBLEM FORMULATION

In this section, we will review the classic HZD based feedback controllers for dynamic 3D walking robots. Inspired by the HZD framework, we then propose to study a model-free control problem using RL techniques.

A. *Existing Challenges of the HZD Framework*

One of the main challenges of 3D bipedal walking is to find feasible trajectories that render stable and robust limit walking cycles while keeping certain desired behaviors, such as walking speeds of the system. In the HZD framework, to obtain such trajectories, an offline optimization problem is solved using the full-order model, and virtual constraints are introduced as a means to synthesize feedback controllers that realize stable and dynamic locomotion. By designing virtual constraints that are invariant through impact, an invariant sub-manifold is created—termed the *hybrid zero dynamics surface*—wherein the evolution of the system is dictated by the reduced-dimensional dynamics of the under-actuated degrees of freedom of the system [11], [12].

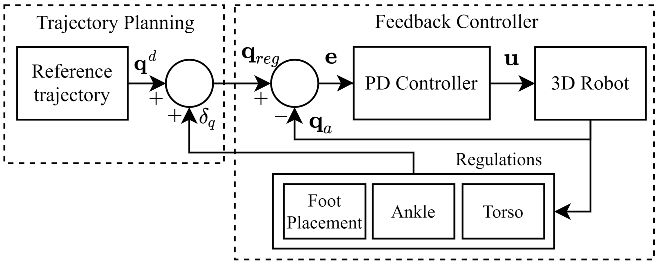

Fig. 2: An example of 3D bipedal walking controller with heuristic feedback regulations.

However, the mathematical models used in this optimization cannot completely capture the complex dynamics of a 3D bipedal robot. Consequently, additional heuristic compensation controllers or regulators are often required on top of the PD tracking controllers to stabilize the robot [18], [19]. An example of such a control structure with feedback regulations for a 3D walking robot is shown in Fig. 2. The compensations δ_q will either modify the original reference trajectories $\mathbf{q}^d$, or exert extra feed-forward torques based on additional feedback information, such as the lateral hip velocity or torso orientation, to improve the stability and robustness of the walking gaits in experiments. The new regulated reference trajectory $\mathbf{q}_{reg}$ is then tracked by PD controller through the control action $\mathbf{u}$.

B. *Structure of HZD-Based Feedback Controller*

From a high-level abstraction, the problem of 3D bipedal walking can be divided in two stages: trajectory planning and feedback control.

Virtual Constraints Let $\mathbf{q}$ be the vector of joint coordinates of a general 3D bipedal robot, and $\tau(t) \in [0,1]$ be a time-based phase variable (see (3) for explicit definition), then the virtual constraints are defined as the difference between the actual and desired outputs of the robot [26]:

$$\mathbf{y}_2 := \mathbf{y}_2^a(\mathbf{q}) - \mathbf{y}_2^d(\tau(t),\alpha), \tag{1}$$

where $\mathbf{y}_2^d$ is a vector of desired outputs defined in terms of 5th order Bézier polynomials parameterized by the coefficients α, given as:

$$\mathbf{y}_2^d(\tau(t),\alpha) := \sum_{k=0}^{5} \alpha[k] \frac{M!}{k!(M-k)!} \tau(t)^k (1-\tau(t))^{M-k}. \tag{2}$$

In this paper, we choose $\tau(t)$ to be the scaled relative time with respect to step time interval, i.e.,

$$\tau(t) = \frac{t-t^-}{t_{step}}, \tag{3}$$

where t_{step} is the duration of one walking step, and t^{-} is the time at the beggining of the step. It is important to denote that by properly choosing the coefficients of these Bézier polynomials, one can achieve different walking motions.

In the HZD framework, the Bézier coefficients are obtained from the solution of an optimization problem whose cost function and constraints are determined by the desired behavior of the robot. Then, the Bézier Polynomials define the desired trajectories to be tracked in order to drive the virtual constraints to zero. However, for 3D bipedal walking robots, simply tracking desired trajectories is not enough to achieve a stable walking motion. Therefore, the following regulations are added to the controller: foot placement, torso regulation, and ankle regulation.

Foot placement controller has been widely used in 3D bipedal walking robots with the objective of improving the speed tracking and the stability and robustness of the walking gait [17], [19], [27]. Longitudinal speed regulation, defined by (4), sets a target offset in the swing hip pitch joint, whereas lateral speed regulation (5) do the same for the swing hip roll angle. In these equations, $v_x[k]$ and $v_y[k]$ are the average longitudinal and lateral speeds of the robot at the middle of step k, v_x^d, v_x^d are the reference speeds, and $K_{p_x}, K_{d_x}, K_{p_y}, K_{d_y}$ are manually tuned gains.

$$\delta^{sw}_{hpitch}[k] = K_{p_x}(v_x[k] - v_x^d) + K_{d_x}(v_x[k] - v_x[k-1]), \quad (4)$$

$$\delta^{sw}_{hroll}[k] = K_{p_y}(v_y[k] - v_y^d) + K_{d_y}(v_y[k] - v_y[k-1]). \quad (5)$$

Torso regulation is applied to keep the torso in an upright position, which is desired for a stable walking gait. Assuming that the robot has a rigid body torso, simple PD controllers defined by (6) and (7) can be applied respectively to the hip roll and hip pitch angle of the stance leg:

$$u^{st}_{hroll} = K_{p_{troll}}(\phi - \phi^d) + K_{d_{troll}}(\dot{\phi} - \dot{\phi}^d), \quad (6)$$

$$u^{st}_{hpitch} = K_{p_{tpitch}}(\theta - \theta^d) + K_{d_{tpitch}}(\dot{\theta} - \dot{\theta}^d), \quad (7)$$

where ϕ and θ are the torso roll and pitch angles, and $K_{p_{troll}}, K_{d_{troll}}, K_{p_{tpitch}}, K_{d_{tpitch}}$ are manually tuned gains.

Ankle regulation is applied to keep the swing foot flat during the whole swinging phase, including the landing moment. For Cassie, this can be done by using forward kinematics for the reference trajectory of the swing ankle joint, given as

$$\gamma^{sw} = \theta - 13\,\text{deg} - 50\,\text{deg}, \quad (8)$$

where γ^{sw} is the ankle joint corresponding to the pitch angle of the swing foot. In addition, to stabilize the walking gait, especially when walking on soft surfaces [19], the stance foot pitch angle of the robot will be set to be passive.

It is important to denote that the speed and torso regulations presented above are fixed, intuitive, and applicable to any general 3D bipedal walking robot. However, given the decoupled structure of the controller used for the different regulations, there are several gains that need to be manually tuned in order to achieve improved stability, which is time-consuming and requires experience. However, this process can be easily automated within an RL framework.

C. Tackling the Problem with Reinforcement Learning

Inspired by the nice properties of HZD (low-dimensional space, accounting of the hybrid nature of walking, virtual constraints), we propose an RL framework that incorporates those properties in the learning process to solve the complex problem of 3D walking. The main objective is to create a unified policy that can handle both problems presented in Fig. 2: trajectory planning and feedback control.

By this, we aim to address two specific challenges generally present in the current methods for 3D bipedal walking (i) eliminating the hideous task of manually tuning the gains of the feedback regulations by including them into the learning process, and (ii) improving the data efficiency of the RL method by reducing significantly its number of parameters.

To validate the proposed approach, we use Cassie-series bipedal robot, designed by Agility Robotics, as our test-bed in this paper. This underactuated biped has 20 degrees of freedom (DOF) in total. Each leg has seven joints, in which five of them are directly actuated by electrical motors and the other two joints are connected via specially designed leaf-spring four-bar linkages for additional compliance. When supported on one foot during walking, the robot is underactuated due to its narrow feet. Agility Robotics has released dynamic simulation models of the robot in MuJuCo [28], which will be used later in this paper.

III. APPROACH

In this section, we propose a non-conventional RL framework that combines a learning structure inspired by the HZD with the foot placement, torso and ankle regulation introduced in section II. We incorporate useful insights from traditional control framework into the learning process of the control policy. By HZD-inspired, we refer to the fact that the learning structure uses a low dimensional representation of the state and a time-based phase variable to command the behavior of the whole system while enforcing invariance of the virtual constraints through impact and symmetry conditions of the walking gait.

A. Overall Framework

We formally present a non-conventional RL framework that uses a low dimensional state of the robot to learn a robust control policy able to track different walking speeds while maintaining the stability of the walking limit cycle. The proposed framework first establishes a NN function that maps a reduced order of the robot's state to (i) a set of coefficients of the Bézier polynomials that define the trajectory of the actuated joints, and (ii) a set of gains corresponding to the derivative gain of the joints PD controller, as well as the gains for the foot placement and torso regulations described in section II-B. Independent low level PD controllers are then used to track the desired output for each joint, which enforces the compliance of the HZD virtual constraints.

A diagram of the overall RL framework is presented in Fig. 3. At each time step, the trained policy maps the inputs of the desired walking velocity, the average actual velocity, the average velocity tracking error, and the torso

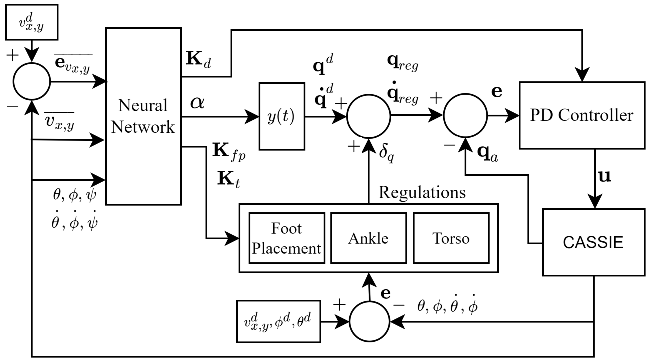

Fig. 3: Overall structure of the RL framework.

orientation and angular velocity to the set of coefficients α, the derivative gains of the joint PD controller, and the compensation gains for the foot placement and torso regulation. A detailed explanation of the NN structure will be given in Section III-B. Then, α is used jointly with the phase variable $\tau(t)$ to compute each joint's desired position and velocity. Compensation gains are used in the foot placement regulation and torso regulation to compute the trajectory compensation for hip roll and pitch angle of the stance and swing leg of the robot. In addition, the ankle regulation computes the trajectory for the swing and stance leg ankle joints. The PD controller uses the tracking error between the desired and actual value of the output to compute the torque of each actuated joint, which is the input of the dynamic system that represents the walking motion of the robot. To close the control loop, the measurement of the robot's states are used as feedback for the inner and outer control loops.

It is worth mentioning that the reference trajectories are learned from scratch and naturally obtained by the proposed RL framework. This is in contrast to some existing studies of RL [29], [24], [25], which rely on some given working policy providing the joints reference trajectories.

B. Neural Network Structure

Fig. 4 shows the structure of the NN implemented for the learning process. Cassie's dynamics model contains 40 states, the robot's pelvis position, velocity, orientation, and angular velocity, plus the angle and angular velocity of all the active and passive joints of the robot. However, the proposed NN only contains 12 dimensional reduced-order state: desired longitudinal and lateral velocity (v^d_x, v^d_y), average longitudinal and lateral velocity ($\overline{v_x}$ $\overline{v_y}$), average longitudinal and lateral velocity error ($\overline{e_{v_x}}$, $\overline{e_{v_y}}$), roll, pitch and yaw angles (ϕ, θ, ψ), and roll, pitch and yaw angular velocities ($\dot\phi, \dot\theta, \dot\psi$). Here, we consider the average speed as the speed during one walking step of the robot, which takes about 350 *ms*. All the inputs are normalized in the interval $[-0.5, 0.5]$. The value of the desired velocity is uniformly sampled from a continuous space interval from -0.5 to 1.0 m/s.

The output of the NN corresponds to the coefficients of the Bézier polynomials, denoted by α, and the set of gains of the PD controller, foot placement and torso compensations, denoted by $\mathbf{K}_d$, $\mathbf{K}_{fp}$ and $\mathbf{K}_t$, respectively. Initially, since the robot has ten actuated joints and each Bézier polynomial is of degree 5, the total size of the set of parameters α should be 60 for the right stance and 60 for the left stance. In addition, the same derivative gains $\mathbf{K}_d$ are used for the corresponding joints of the left and right legs, which is possible because of the symmetric nature of the walking gait. Finally, by equations (4)-(6), $\mathbf{K}_{fp}$ and $\mathbf{K}_t$ are of the form

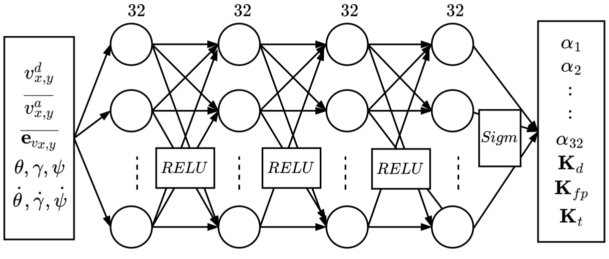

Fig. 4: Structure of the neural network.

$$\begin{aligned}\mathbf{K}_{fp} &= [K_{p_x}, K_{d_x}, K_{p_y}, K_{d_y}],\\ \mathbf{K}_t &= [K_{p_{troll}}, K_{d_{troll}}, K_{p_{tpitch}}, K_{d_{tpitch}}].\end{aligned} \tag{9}$$

Then $\mathbf{K}_d$ is of dimension 5, and both, $\mathbf{K}_{fp}$ and $\mathbf{K}_t$, are of dimension 4. This results in a total of 133 outputs. Nonetheless, by considering the physical insight of the dynamic walking, we can significantly reduce the number of outputs of the NN from 128 to 45. This will be explained in detail in Section III-C. The number of hidden layers of the NN is 4, each with 32 neurons. ReLU activation functions are used between hidden layers, whereas the final layer employs a sigmoid function to limit the range of the outputs. As compared with other methods in the literature [21], [24], the proposed NN is much smaller in size, making the overall RL method sample efficient and easy to implement.

Finally, due to the properties of the family of polynomials used to parameterize the joints trajectories, the set of Bézier coefficients accurately define the upper and lower bounds of the desired output trajectories. That is, for each set of Bézier coefficients α_i and desired trajectory q^d_i associated with the i^{th} joint, we have

$$q^{min}_i < \alpha^{min}_i < q^d_i < \alpha^{max}_i < q^{max}_i. \tag{10}$$

Therefore, the output range of the set of parameters can be limited by the physical constraint of each actuated joints, or even more, by the expected behavior of the robot (q^{min}_i, q^{max}_i). This critical feature significantly reduces the continuous interval of the output, which decreases the complexity of the RL problem and improves the efficiency of the learning process.

C. Reduction of Output Dimension

For a general walking pattern, there exists a symmetry between the right and left stance. Therefore, given the set of coefficients for the right stance $\alpha_R \in \mathbb{R}^{6x10}$, where each column represents the Bézier coefficients for a desired joint trajectory, we can easily obtain the set of coefficients for the left stance $\alpha_L \in \mathbb{R}^{6x10}$ by

$$\alpha_L = \alpha_R \mathbf{T} \tag{11}$$

where $\mathbf{T} \in \mathbb{R}^{10x10}$ is a very sparse transformation matrix that represents the symmetry between the joints of the right and left legs of the robot.

To encourage the smoothness of the control actions after the ground impact, we enforce that at the beginning of every step the initial point of the Bézier polynomial coincides with the current position of the robot's joints. That is, for each joint i with Bzier coefficients $\alpha_R \in \mathbb{R}^{6x10}$, we have

$$\alpha_i[0] = q_i(\tau(0)). \tag{12}$$

In addition, to encourage the invariance of the virtual constraints through impact, we enforce the position of the hip joints and knee joints to be equal at the beginning and end of the step ($\tau(t) = 0$ and $\tau(t) = 1$ respectively).

Finally, the ankle regulation enforces the stance ankle to be passive and the trajectory of the non-stance ankle to be defined by forward kinematics accordingly to (8). Therefore, we do not need to find the Bézier coefficients for the stance and swing ankle joints.

D. Learning Procedure

Provided the NN policy structure and the reduced desired output of actions, the network is then trained with the evolution strategies (ES) [30]. Note that our proposed method is not limited to a particular training method. It can be trained using any RL algorithms that can handle continuous action space, including evolution strategies (ES), proximal policy optimization (PPO) [22], and deterministic policy gradient methods [31].

In this paper, we adopt the following reward function in training for Cassie:

$$r = \mathbf{w}^T \mathbf{r}, \tag{13}$$

with a vector of 8 customized rewards $\mathbf{r}$ and the weights $\mathbf{w}$. Specifically,

$$\mathbf{r} = [r_{v_x}, r_{v_y}, r_h, r_u, r_{COM}, r_{ang}, r_{angvel}, r_{fd}]^T. \tag{14}$$

This encourages better velocity tracking (through r_{v_x}, r_{v_y}), height maintenance (r_h), energy efficiency (r_u) and natural walking gaits ($r_{COM}, r_{ang}, r_{angvel}, r_{fd}$). Starting from a random initial state that is close to a "stand-up" position with zero velocity and a uniformly sampled desired velocity, we collect a trajectory of states, actions and rewards, referred as an episode. The episode length is 10000 simulation steps and it has an early termination if any of the following conditions is violated:

$$\begin{gathered} \psi| < 0.5, \quad |\theta| < 0.5, \quad |\phi| < 0.5, \\ 0.75 < p_z < 1.1, \quad \Delta_f < 0.05, \end{gathered} \tag{15}$$

where p_z is the height of the robot's pelvis and Δ_f is the distance between the feet.

IV. SIMULATION RESULTS

To validate the proposed method, a customized environment for Cassie was built in MuJoCo physics engine [32]. We used the model information of Cassie robot provided by Agility Robotics and the Oregon State University, which is publicly available [28]. The number of trainable parameters for the NN is 5069, and the training time is about 10 hours using a single 12-core CPU machine. Visualized results of the learning process and evaluation of the policy in simulation can be seen in the accompanying video submission [33]. This section presents the performance of the control policy obtained from the training in terms of (i) speed tracking, (ii) disturbance rejection, and (iii) the convergence of stable periodic limit cycles.

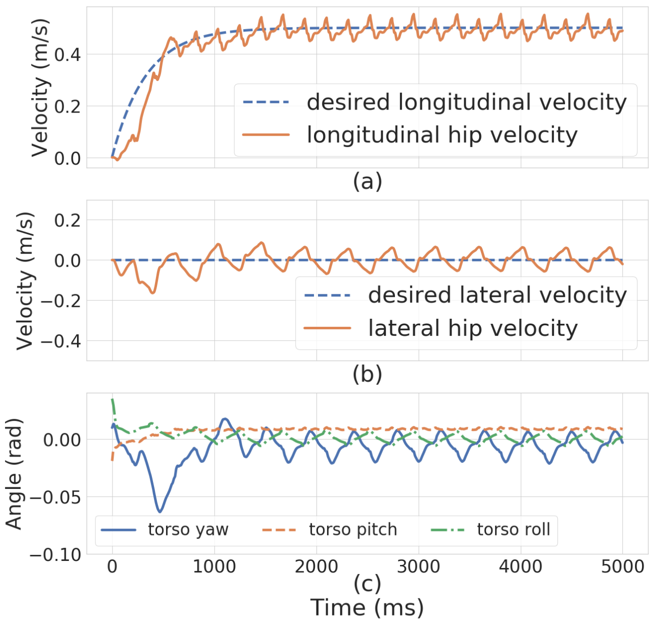

Fig. 5: Performance of the learned policy while tracking a fixed desired longitudinal walking speed.

A. Speed Tracking

Due to the decoupled structure, the learned controller can effectively track a wide range of desired walking speeds in both longitudinal and lateral directions. The performance of tracking a fixed desired speed of 0.5 m/s in the forward direction is shown in Fig. 5. From Fig. 5(c), one can see the controller is capable of keeping the upright position of the torso while walking. This particular behavior is encouraged by the reward function during the training process, and it also contributes to the stability of the walking gait.

The performance of continuously tracking various desired speeds is shown in Fig. 6, in an interval from -0.5 to 1.0 m/s longitudinally (v_x), and an interval from -0.3 to 0.3 m/s in the lateral direction (v_y). Note that walking at negative v_x means the robot is walking backward, whereas moving at positive v_y implies the robot is taking side steps to the left. In both cases, the controller is able to handle any speed change without falling or losing track of the reference, even after steep changes in the desired velocity.

B. Disturbance Rejection

To evaluate the robustness of our controller, we applied an adversarial force directly at the robot's pelvis in both the forward and the backward directions. It is worth emphasizing

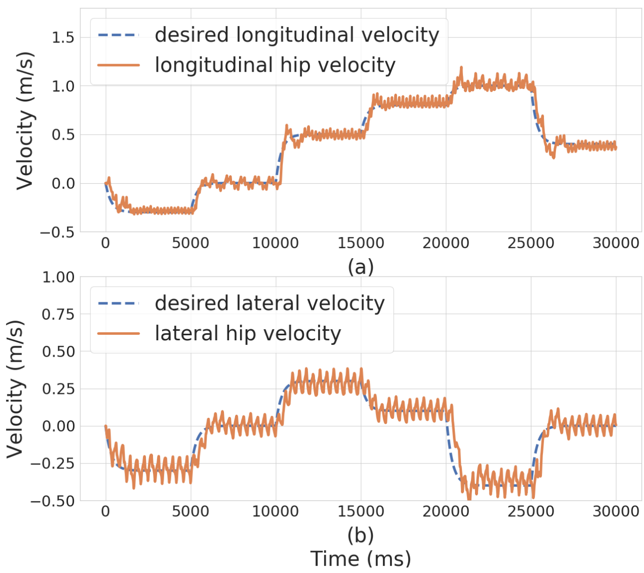

Fig. 6: Performance of the learned policy while tracking varying desired longitudinal and lateral walking speeds.

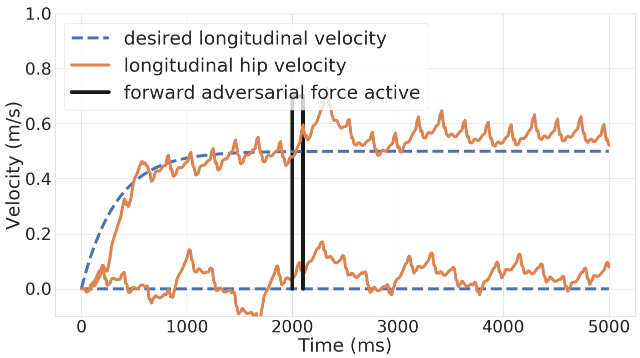

Fig. 7: Robustness of the controller when an adversarial force is applied in the forward direction.

that we do not inject any torso disturbance throughout the training process. The robustness of the policy is achieved naturally through the constant updates of the Bézier coefficients, the derivative gains of the adaptive PD controller, and the gains of the additional foot placement, torso and ankle regulators. In the results shown in Fig. 7 and Fig. 8, we adopt the adversarial force with the same magnitude of 25 N in both directions. It is applied 2 seconds after starting the test and lasts for 0.1 seconds. Throughout our adversarial tests, the robot can handle up to 40 N in the forward direction and 45 N in the backward direction without falling, but the speed tracking may take a long time to recover with an external force of high magnitude.

Fig. 7 illustrates the response of the controller for a forward adversarial force when the robot is walking in place and walking forward at 0.5 m/s. Fig. 8 illustrates the response of the controller when the same force of 25 N is applied in the backward direction while the robot is walking forward at 0 and 0.8 m/s. Throughout the four tests, the robot never falls and always closely recovers to the desired velocity.

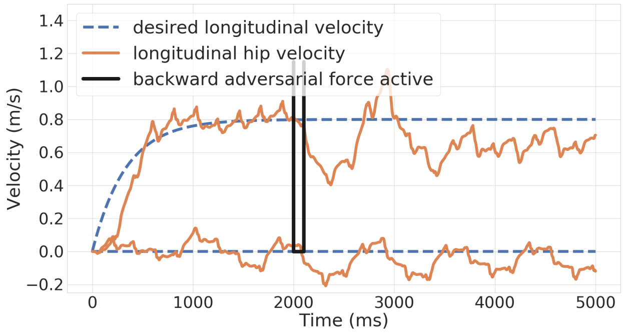

Fig. 8: Robustness of the controller when an adversarial force is applied in the backward direction.

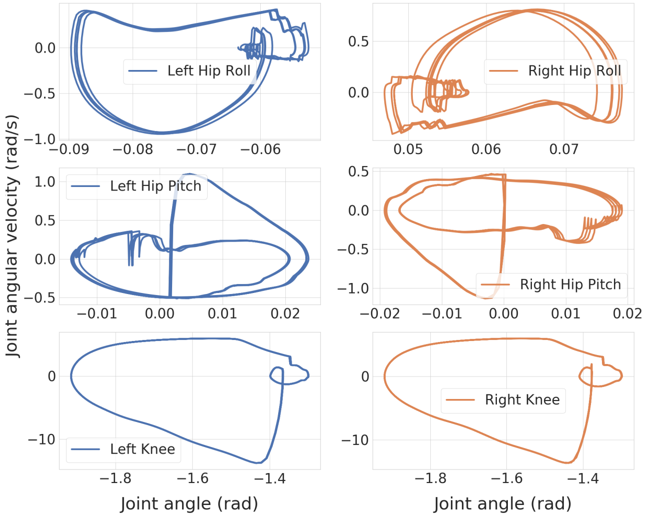

Fig. 9: Walking limit cycle of the learned policy with the desired longitudinal velocity of 0.5 m/s.

C. *Periodic Stability of the Walking Gaits*

Periodic stability is one of the most important metrics for assessing the stability of walking gaits. In this paper, we only empirically evaluated the stability by observing the joint limit cycles of a periodic walking gait. Fig. 9 shows that the convergence of several representative robot actuated joints to periodic limit cycles during a fixed speed walking. Moreover, the orbit described by the left and right joints demonstrates the symmetry of walking gaits. This is due to the specific feature we encouraged in the design of the control policy.

V. Conclusion

This paper presents a novel model-free RL approach for the design of feedback controllers for 3D bipedal robots. The unique decoupled structure of the learned control policy incorporates the physical insights of the dynamic walking and heuristic compensations from classic 3D walking controllers. The result is a data-efficient RL method with a reduced number of parameters in the NN that can learn stable and robust dynamic walking gaits from scratch, without any reference motion or expert guidance. The learned policy demonstrates good velocity tracking and disturbance rejection performances on a 3D bipedal robot.

References

[1] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames, "Models, feedback control, and open problems of 3D bipedal robotic walking," *Automatica*, vol. 50, no. 8, pp. 1955–1988, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109814001654

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1, Oct 2001, pp. 239–246 vol.1.

[3] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life," *I. J. Humanoid Robotics*, vol. 2, pp. 225–227, 2004.

[4] E. Yoshida, C. Esteves, I. Belousov, J. Laumond, T. Sakaguchi, and K. Yokoi, "Planning 3-d collision-free dynamic robotic motion through iterative reshaping," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1186–1198, Oct 2008.

[5] B. J. Stephens and C. G. Atkeson, "Dynamic balance force control for compliant humanoid robots," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 1248–1255.

[6] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, Dec 2006, pp. 200–207.

[7] 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*, vol. 31, no. 10, pp. 1117–1133, 2012.

[8] M. Posa, S. Kuindersma, and R. Tedrake, "Optimization and stabilization of trajectories for constrained dynamical systems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1366–1373.

[9] 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)*, Oct 2013, pp. 292–299.

[10] 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 *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3346–3351.

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

[12] A. D. Ames, "Human-inspired control of bipedal walking robots," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1115–1130, May 2014.

[13] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. R. Westervelt, C. C. De-Wit, and J. W. Grizzle, "RABBIT: a testbed for advanced control theory," *IEEE Control Systems*, vol. 23, no. 5, pp. 57–79, Oct. 2003.

[14] K. Sreenath, H. Park, I. Poulakakis, and J. W. Grizzle, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[15] H. Zhao, A. Hereid, W. Ma, and A. D. Ames, "Multi-contact bipedal robotic locomotion," *Robotica*, vol. 35, no. 5, pp. 1072–1106, Apr. 2017.

[16] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames, "Dynamic humanoid locomotion: a scalable formulation for HZD gait optimization," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 370–387, Apr. 2018.

[17] S. Rezazadeh, C. Hubicki, M. Jones, A. Peekema, J. Van Why, A. Abate, and J. Hurst, "Spring-mass walking with atrias in 3d: robust gait control spanning zero to 4.3 kph on a heavily underactuated bipedal robot," in *ASME 2015 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2015, pp. V001T04A003–V001T04A003.

[18] J. P. Reher, A. Hereid, S. Kolathaya, C. M. Hubicki, and A. D. Ames, "Algorithmic foundations of realizing multi-contact locomotion on the humanoid robot DURUS," in *the 12th International Workshop on the Algorithmic Foundations of Robotics (WAFR)*. San Francisco: Springer, Dec. 2016. [Online]. Available: http://wafr2016.berkeley.edu/

[19] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, "Feedback control of a Cassie bipedal robot: walking, standing, and riding a segway," *American Control Conference (ACC)*, 2019.

[20] X. Da and J. Grizzle, "Combining trajectory optimization, supervised machine learning, and model structure for mitigating the curse of dimensionality in the control of bipedal robots," *The International Journal of Robotics Research*, vol. 38, no. 9, pp. 1063–1097, 2019. [Online]. Available: https://doi.org/10.1177/0278364919859425

[21] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *CoRR*, vol. abs/1509.02971, 2015.

[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[23] G. A. Castillo, B. Weng, A. Hereid, Z. Wang, and W. Zhang, "Reinforcement learning meets hybrid zero dynamics: A case study for rabbit," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 284–290.

[24] Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. van de Panne, "Feedback control for cassie with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1241–1246.

[25] Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, and M. van de Panne, "Iterative Reinforcement Learning Based Design of Dynamic Locomotion Skills for Cassie," *arXiv e-prints*, p. arXiv:1903.09537, Mar 2019.

[26] A. D. Ames, "Human-inspired control of bipedal robots via control lyapunov functions and quadratic programs," in *Proceedings of the 16th international conference on Hybrid systems: computation and control*, C. Belta and F. Ivancic, Eds., ACM. ACM, 2013, pp. 31–32.

[27] 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.

[28] "A simulation library for agility robotics' cassie robot using mujoco," https://github.com/osudrl/cassie-mujoco-sim, accessed: 2019-09-15.

[29] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne, "Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 41:1–41:13, July 2017.

[30] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[31] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," 2014.

[32] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: a physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 5026–5033.

[33] "Simulation results for Cassie in MuJoCo," https://youtu.be/GOT6bnxqwuU, accessed: 2019-09-15.