2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)
November 28-30, 2022. Ginowan, Japan.

Dynamic Bipedal Turning through Sim-to-Real Reinforcement Learning

Fangzhou Yu, Ryan Batke, Jeremy Dao, Jonathan Hurst, Kevin Green, and Alan Fern

2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids) | 979-8-3503-0979-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/HUMANOIDS53995.2022.10000225

***Abstract*— For legged robots to match the athletic capabilities of humans and animals, they must not only produce robust periodic walking and running, but also seamlessly switch between nominal locomotion gaits and more specialized transient maneuvers. Despite recent advancements in controls of bipedal robots, there has been little focus on producing highly dynamic behaviors. Recent work utilizing reinforcement learning to produce policies for control of legged robots have demonstrated success in producing robust walking behaviors. However, these learned policies have difficulty expressing a multitude of different behaviors on a single network. Inspired by conventional optimization-based control techniques for legged robots, this work applies a recurrent policy to execute four-step, 90° turns trained using reference data generated from optimized single rigid body model trajectories. We present a training framework using epilogue terminal rewards for learning specific behaviors from pre-computed trajectory data and demonstrate a successful transfer to hardware on the bipedal robot Cassie.**

I. INTRODUCTION

Animals exhibit a multitude of dynamic behaviors, such as squirrels leaping from treetops and birds taking off and landing. Moreover, they are also able to seamlessly transition between behaviors. Robots that match human and animal athletic capability will require a control architecture that enables them to transition between behaviors with the same fluidity. As legged robots evolved over the past few decades to become more agile and dynamic, their control algorithms became more sophisticated to take advantage of advances in mobile computing [1]. Recent developments in the realm of legged locomotion controls prominently feature the use of model predictive control (MPC) and optimization techniques to generate trajectories for quadrupedal jumps and backflips [2], [3]. Using reinforcement learning (RL) to train neural network locomotion controllers has also shown to be a promising alternative avenue of research, enabling Cassie, a human-scale bipedal robot, to perform dynamic gaits ranging from walking, running, skipping, and stair climbing on real-world hardware [4], [5]. This work extends upon the periodic reward composition method of learning bipedal gait policies for Cassie [4] by including trajectory data derived offline using a single rigid body model (SRBM) [6] with the aim of producing policies that learn to transfer the trajectories of the reduced-order model (ROM) to Cassie in the real world. The challenge of successfully transitioning between different policies is addressed during the training process with the concept of an epilogue terminal reward. To prove the viability of our proposed technique implemented on real world hardware, we demonstrate the successful sim-to-real transfer of Cassie performing a four-step 90 degree right turn using a policy trained with trajectory data that successfully transitions between another policy performing a running gait developed from our previous work.

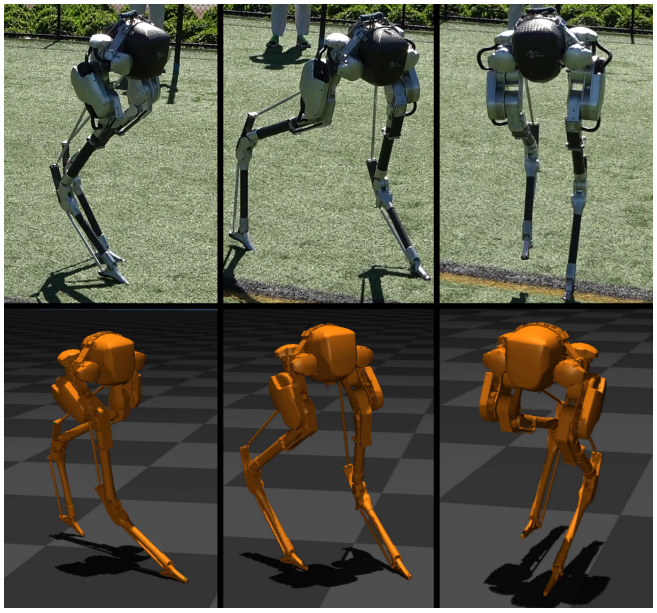

Fig. 1. A Cassie robot executing a four-step 90° right turn. **(Top Row)** Hardware field test of the full-reference turning policy initialized from a commanded heading speed of 2.0m/s on artificial turf. **(Bottom Row)** Cassie running the full-reference turning policy in simulation initialized from a target heading speed of 2.5 m/s.

II. RELATED WORKS

A. Learning Locomotion Skills

RL has shown to be a promising alternative to model-based control of legged robots. Recent work on RL control policies on the Cassie and ANY-mal robots are able to out perform model-based controllers in published research, while being computationally cheaper to evaluate at runtime [4], [7], [8]. However, most published work on the application of RL to legged locomotion focuses on performing cyclic gaits, while in this work we are concerned with more dynamics one-off maneuvers. Prior work on performing different behaviors with learned methods use RL to train a singular policy to execute all desired behaviors instead of training separate policies for each individual behavior. This causes the behavior space of the policies to be limited by

*This work is supported by the NSF Grants No. IIS-1849343, NSF Grants 1314109-DGE and DARPA Contract W911NF-16-1-0002.

All authors are with Collaborative Robotics and Intelligent Systems Institute, Oregon State University, Corvallis, Oregon, 97331, USA. Email: {yufangzh, batker, daoje, jonathan.hurst, greenkev, alan.fern}@oregonstate.edu.

the richness of a singular reward function, making them ill-suited to learning a wide variety of different behaviors. This work addresses this issue by training policies per behavior and switching between them. [9] uses a RL-based approach similar to our prior work to produce robust locomotion policies for the ANY-mal quadrupedal robot that learns to track commanded body velocities and yaw rate better than any previously existing controller. They also demonstrate a fall-recovery policy trained with the same methodology that can successfully get up from difficult configurations, although policy transitions occur only at static robot configurations. [4] demonstrated a learning framework capable of reproducing all common bipedal gaits for Cassie on a single policy that does not use expert reference trajectories. The resulting policy could continuously transition between different bipedal gaits by adjusting a left and right foot cycle offset parameter. However, this framework is unable to express maneuver sequences that fall outside what can be expressed by varying a single gait cycle offset parameter.

B. Learning Behavior Transitions

Data-driven learning approaches has also been used to learn different locomotion skills as well as smooth transitions between them, although research in this area is mostly confined to impressive simulation results, and published hardware results in this area is rather preliminary. [10] demonstrated successful sim-to-real transfer of switching between forward and backward walking on a single policy trained with atomic, task-specific reward functions. Similarly, learned locomotion control policies have shown to be capable of assuming different gait behaviors to negotiate terrain obstacles and gaps. [11] demonstrated the emergence of robust obstacle clearing behavior for torque-controlled legged agents by training control policies using simple reward functions in obstacle-rich simulation environments. These examples of prior work engineer the agent-environment interactions to encourage the emergence of multiple locomotion modes, which limits the amount of control one can have over the possible actions and strategies adopted by the agent, and may also lead to unexpected environment exploitation. Related work from the computer graphics community has demonstrated impressive results in motion synthesis for animated characters by learning from motion capture datasets [12], [13]. [14] used RL to train an physically-simulated agent to mimic behaviors from reference motion capture data on a single network. Their results were able to reproduce a diverse array of behaviors from the sample data, and was also able to generalize to some behaviors unseen during training. Similarly, [15] used RL with a generative adversarial motion-prior component to train a simulated character for motion synthesis of complex and highly dynamic behavior sequences. [16] demonstrated that the RL framework of [15] can transfer to hardware on a quadrupedal robot, but only basic locomotion behaviors were shown to transfer to hardware after training to mimic dog motion capture data.

C. Model Based Methods

Trajectory optimization (TO) is a widely used technique in motion planning for modern dynamic legged locomotion [3], [17]–[19]. In this context, trajectory optimization is a tool used to yield a plan for future robot states given an initial state, such as contact wrenches and *center-of-mass (CoM)* positions. The fidelity of the model used for TO varies from detailed, full-order dynamic and kinematic representations of actual hardware [18] to reduced-order dynamic, full-order kinematic models [20], down to minimal centroidal/SLIP models amenable for MPC [2], [21]. In this work, we choose to use the single rigid body model for its ability to capture linear and rotational dynamics while being easy to describe mathematically. TO-based control techniques for legged robots have also shown success in composing behaviors in recent work. [2] used a MPC strategy to execute running leaps for the Cheetah 2 robot over obstacles. [22] used offline TO to generate jumping trajectories and a separate MPC style landing controller that targeted an optimal distribution of foot contact forces to perform jumps and successful landings for Cheetah 3. [23] demonstrates jumping behaviors on Cassie using a model based controller that tracks a reference trajectory generated by applying direct collocation to the hybrid dynamics of a jump sequence on a reduced order spring-mass model. [24] achieved smooth transitions between a large variety of different behaviors by using TO to generate a pre-computed motion library, and targeted sequences of desired library motions using MPC that plan over shorter time horizons. This approach is similar to the approach taken by Boston Dynamics [25] for recent work on the Atlas bipedal robot. Model-based methods are capable of producing complex, dynamic behavior on legged robots, but they are challenging to implement, and expensive to evaluate. In comparison, training control policies using RL algorithms are a more straightforward method of transferring dynamic motion plans to controllers for the full-order robot.

III. Methods

A four step 90° right turn was selected as the target behavior for the control policies in this work because its aperiodic and highly dynamic nature marks a significant departure from the dynamical regime of regular walking. We anticipate that such behaviors will be difficult to learn and thus choose to rely on reference trajectories in order to guide the policy learning. Control policies trained to execute the turning behavior are expected to start from a pretrained walking policy, follow the reference turning trajectory, and should transition back to the walking policy at the end of the turning maneuver. Matching the terminal state of the reference trajectory is not guaranteed to permit successful transitions back to walking policies, so turning policies must learn how to deviate away from tracking the reference data to facilitate successful transitions. This challenge is addressed using epilogue rewards detailed in Section III-E, and is a novel component of our proposed learning framework. We train recurrent control policies to perform four-step, 90° turns using Proximal Policy Optimization (PPO) in simulation

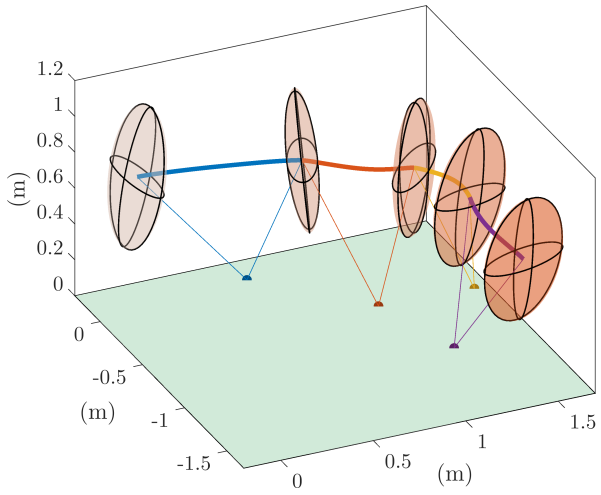

Fig. 2. Plot of the reference trajectory for a 2.5 m/s, four-step turn from the optimized single rigid-body model moving left to right. The thick line represents the center of mass path, with different colors showing the different stance phases. Thin lines show leg positions at the start and end of stance phases.

[26]. The simulator we use is MuJoCo, extended with the robot's state estimator and noise models[1]. Previous work has shown that highly accurate simulations such as this are effective at producing control policies that transfer to hardware with no additional adaptation [4], [10], [27].

A. Reference Trajectory Optimization

Dynamic legged maneuvers require abrupt changes in linear and angular momentum while heavily constrained by underactuation constraints. We hypothesize that in these contexts reference information could be more useful than it was previously shown to be in nominal, steady-state locomotion. To provide a rich library of reference motions we perform trajectory optimization with an SRBM, representing a reduced-order model of locomotion. The SRBM approximates the complex multibody dynamics of a robot into a single rigid-body with dynamics that are manipulated via ground reaction forces applied at footholds. We apply a widely-used, prescribed contact sequence, direct collocation trajectory optimization method [28]. This allows the optimization to adjust foot timings, but not the sequence of contacts. This is not overly restrictive as bipedal robots have only a small space of feasible contact patterns. Our contact pattern for four-step turns is a grounded run consisting of alternating phases of single-stance with instant transfer. We apply a set of transferability constraints which ensure the resulting trajectories are more directly applicable to the target Cassie robot. These include maximum ground reaction force, friction cones, maximum yank (time rate of change of force), leg length limits, and foot placement constraints to prevent leg crossing. More details on the library generation method can be found in [6].

The resulting library of turn references spans from 0.0 to 2.5 m/s. The 2.5 m/s turning trajectory is shown in Fig. 2. The trajectories have smoothly varying body motions, footstep locations, ground reaction forces, and step timing.

[1]Simulation available at https://github.com/osudrl/cassie-mujoco-sim

Policy Input	Size
Pelvis Orientation Quaternion	4
Pelvis Angular Velocity	3
Pelvis Translational Acceleration	3
Joint Positions and Velocities	28
Maneuver Progression	1
Clock Signal	2
Target Forward Speed	1

TABLE I

THE INPUTS INTO THE LEARNED CONTROL POLICIES. ALL STATE INFORMATION IS ESTIMATED FROM REAL OR SIMULATED SENSOR DATA.

B. Policy Network Design

The policy architecture used in this work is derived from previous work on applying LSTM networks to bipedal locomotion control [29]. Both actor and critic networks are LSTM RNNs of size 128x128. The state space inputs to our control policy concatenates information from the robot state estimator along with a maneuver progression counter, two periodic clock waveforms, and a target forward heading speed for a total input space size of 42. The breakdown of the state space is shown in Table I.

The action space of the policy consists of position targets for all 10 actuated joints on Cassie. The actions are updated at our nominal policy control rate of 40hz, which are then sent to joint-level PD controllers running at 2 kHz.

C. Reward Function Formulation

To support an ablation study, we trained policies using different reward functions, *Full Reference, Subset Reference, Foot Timing,* and *No Reference*, each with a different set of additive reward components that capture different aspects of reference information. Table II gives the individual component weights for each of the four reward functions. All weights are rounded the nearest percentage point.

The following reward components are common across all four reward function variations:

- A contact mode reward r_{contact}, which specifies when each foot should be in swing or in stance with a piecewise linear clock function. The gait parameters that define such a function (stepping frequency and swing ratio) is calculated from the reference information. We refer readers to previous work [27] for further details.
- Action smoothness, torque cost, and motor velocity costs on the hip roll and yaw motors make up r_{ctrl}. These terms, along with self collision avoidance rewards r_{coll} help promote successful sim-to-real transfer. To implement r_{coll}, we use $\mathbf{s_{site}}$ as the distance between points of interest on Cassie's legs, such as the inner edge of the heelsprings and the ankle joint crank. This measurement defines the collision avoidance heuristic that penalizes the policy for small $\mathbf{s_{site}}$ values.

$$r_{\text{coll}} = \exp\left(-\left|\frac{-100\mathbf{s_{site}}}{6} + 3\right|\right) \tag{1}$$

The four reward functions are summarized below.

Reward	Full Ref.	Sub Ref.	Foot Timing	No Ref.
r_{ψ}^{ref}	6	3	-	-
r_{ψ}^{interp}	-	-	20	20
$r_{\mathbf{v_{xy}}}$	6	22	-	-
$r_{\mathbf{v_z}}$	3	-	-	-
$r_{\mathbf{pose_{xy}}}$	13	22	-	-
$r_{\mathbf{pose_z}}$	6	-	-	-
$r_{\mathbf{L}}$	9	-	-	-
r_{contact}	38	32	53	53
r_{ctrl}	19	22	27	27

TABLE II

REWARD COMPONENT COMPOSITION AND WEIGHTING PERCENTAGES.

1) Full Reference: The tracking components of the reward function include pelvis yaw angle (ψ), pelvis linear velocity (**v**), pelvis angular momentum (**L**), and the relative distance vector between the pelvis COM and the stance foot (**pose**). They are given by

$$r_{\psi}^{\text{ref}} = \exp\left(-\left|(3(\psi_{\text{pelv}} - \psi_{\text{pelv}}^{\text{ref}})\right|\right) \tag{2}$$

$$r_{\mathbf{v}} = \exp\left(-\left\|2(\mathbf{v}_{\text{pelv}} - \mathbf{v}_{\text{pelv}}^{\text{ref}})\right\|_1\right) \tag{3}$$

$$r_{\mathbf{L}} = \exp\left(-\left\|\mathbf{L}_{\text{body}} - \mathbf{L}_{\text{body}}^{\text{ref}}\right\|_1\right) \tag{4}$$

$$r_{\mathbf{pose}} = \exp\left(-\left\|5(\mathbf{p}_{\text{pose}} - \mathbf{p}_{\text{pose}}^{\text{ref}})\right\|_1\right) \tag{5}$$

2) Subset Reference: Includes only a subset of full reference rewards, specifically $r_{\psi}^{\text{ref}}, r_{\mathbf{v_{xy}}}, r_{\mathbf{pose_{xy}}}, r_{\text{contact}}, r_{\text{ctrl}}$. Notably, this omits the angular momentum tracking term in equation (4), as well as tracking only the planar x, y components of $r_{\mathbf{v}}$ and $r_{\mathbf{pose}}$. This particular reward function was chosen because tracking angular momentum was found to have no qualitative effects on the behavior of the resulting policies.

3) Foot Timing: Omits all tracking rewards (2) to (5) and only consists of $r_{\psi}^{\text{interp}}, r_{\text{contact}}, r_{\text{ctrl}}$. The only reference information present in the reward is the gait parameters for the contact mode reward term. r_{ψ}^{interp} replaces r_{ψ}^{ref} and tracks a yaw target that linearly interpolates between 0 and $-\pi/2$ within the timespan of the reference turning maneuvers instead of the optimized yaw trajectory $\psi_{\text{pelv}}^{\text{ref}}$.

4) No Reference: A reference-free policy similar to *Foot Timing* that also omits the tracking rewards (2) to (5). It uses $r_{\psi}^{\text{interp}}, r_{\text{contact}}, r_{\text{ctrl}}$ exclusively, but in contrast to *Foot Timing*, the gait parameters for r_{contact} are set by a hand-tuned heuristic. Thus, this policy uses **no** information from the reference trajectory.

D. Episode Initialization

The beginning of a training episode for turning needs to be reset to a configuration that is a close match to the starting SRBM configuration specified by the turning trajectory. For this purpose, a set of initialization poses $P_{\text{init}}(v, \theta)$ is generated by executing a pre-trained running policy in simulation for a sweep of commanded speeds that match the speeds v of the trajectories in the reference library. The configurations $[q, \dot{q}]$ of Cassie within a range of gait phases θ before and after a left-foot swing apex (the starting point of the reference trajectories) are saved to P_{init}. On every reset, the configuration state of Cassie is uniformly sampled from the set of poses in P_{init} for a desired initial speed.

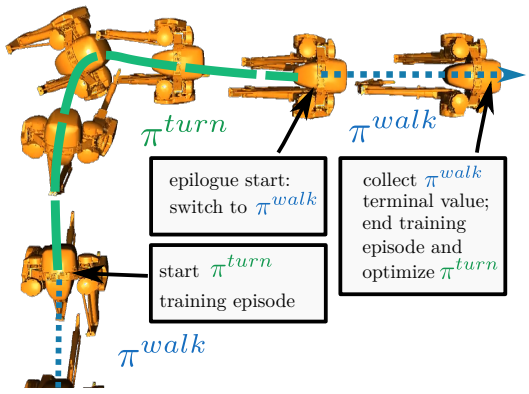

Fig. 3. Visualization of a PPO rollout during training. After being initialized from a π^{walk} pose, π^{turn} is evaluated until the end of the turning maneuver. If π^{turn} completed the turning maneuver, π^{walk} subsequently takes over to generate the epilogue reward.

E. Epilogue Reward

Since we want to transition back to walking after executing a turn, the turning policy should fulfill the terminal objective of ending in a state that can successfully initialize walking in order to return to a nominal locomotion gait. We introduce the novel concept of training with an epilogue reward as a component of our training framework in order to allow turning policies π^{turn} to successfully switch back to the nominal locomotion policies π^{walk} once the turning policies have reached the end of the reference trajectory states.

At the end of a standard PPO rollout, the critic value for the final state $V_{\pi}(S_T)$ is used as the terminal value in the discounted chain of rewards received at each episode step to estimate the sum of any future discounted rewards [26]. This is analogous to calculating the n-step temporal-difference (TD) returns, where the final value is an estimate for the uncollected rewards beyond the n-step horizon [30]. The epilogue reward is an alternative terminal value computed at the end of a π^{turn} training episode. It is the discounted sum of returns of the epilogue episode which starts when the turning policy has successfully reached the end of its maneuver. During the epilogue episode, the walking policy π^{walk} takes over from the last state of the turning episode (normal PPO rollout of the π^{turn} turning policy), and is evaluated deterministically for k simulation steps. In essence, rather than just using $V_{\pi^{\text{walk}}}$ to estimate how well the turning policy can transition back to the walking policy, we actually rollout the walking policy for some steps and use the resulting rewards as a terminal value estimate. Formally, the epilogue reward is

$$V_{\pi^{\text{turn}}}(S_T) = \left(\sum_{i=T}^{T+k} \gamma^{i-T} R_{i+1}\right) + \gamma^k V_{\pi^{\text{walk}}}(S_{T+k+1}) \tag{6}$$

where k is the length of the epilogue, T is the length of the turning maneuver, R_{i+1} is based on the reward function

Parameter	Range	Unit
Policy Control Rate	[0.95,1.05] × default	Hz
Joint Encoder Noise	[-0.05, 0.05]	rad
Joint Damping	[0.8, 2.5] × default	Nms/rad
Link Mass	[0.9, 1.5] × default	kg
Friction Coefficient	[0.45, 1.3]	-
External Force Magnitude	[0, 40]	N
External Force Dir. (Azimuth)	[0, 2π]	rad
External Force Dir. (Elevation)	[0, $\frac{\pi}{4}$]	rad
Initial Pelvis Velocity (x)	[-0.3,0.3] + default	m/s
Initial Pelvis Velocity (y)	[-0.4,0.4]	m/s

TABLE III
RANDOMIZATION RANGE PARAMETERS

used to train π^{walk}, and $V_{\pi^{\text{walk}}}$ is the critic trained for the walking policy. To the best of the authors knowledge, the use of non-standard terminal values such as our epilogue reward has not been explored in prior work within the domain of RL-based controllers for legged locomotion. Modifying the estimate of future returns in this manner incentivizes π^{turn} to terminate in a configuration $[q, \dot{q}]$ amenable for the execution of π^{walk} by maximizing the epilogue returns for continued walking. As a control for the epilogue, we also compare turning policies trained using a k value of 120 against their no-epilogue versions trained using the same parameters, but with k set to zero.

F. Dynamics Randomization

We applied dynamics randomization as described in [29] during the training process of our turning controller to help close the sim-to-real gap and enable a successful transfer to real hardware. In addition, we also apply a constant perturbance force to the robot pelvis over the course of a training episode with a randomly sampled magnitude and direction in order to promote the emergence of robust turning behaviors. The details of our randomization parameters can be found in Table III.

IV. RESULTS

To evaluate the utility and necessity of our optimized SRBM trajectories and the epilogue reward, we assess and compare the set of policies proposed in Section III-C and Section III-E in simulation for their performance and turning behavior characteristics.

We also present successful sim-to-real transfer of a selection of the policies tested in simulation in our submission video.

A. Simulation Results

1) Sample Efficiency: We plot the learning curves for each policy in Fig. 4 to compare the sample efficiency of our turning policies. Since each policy is trained with different reward functions, the reward values attained by each policy can not be used to form conclusions about their relative performance. Instead, we compare policies by the number of samples to convergence, shown for each turning policy by star symbols in Fig. 4 that mark when each policy first surpassed 97% of the maximum reward value experienced during training. Notably, the policies that use less information from the optimized trajectories converge slightly faster than the policies that follow the SRBM reference trajectory more faithfully. We hypothesize that the learning speed disparity may be attributed to model differences between the SRBM and Cassie's dynamics leading to conflicting interactions between the tracking reward terms. This may cause policies that track more of the reference data to require more samples in order to learn to optimize for multiple conflicting objectives before convergence.

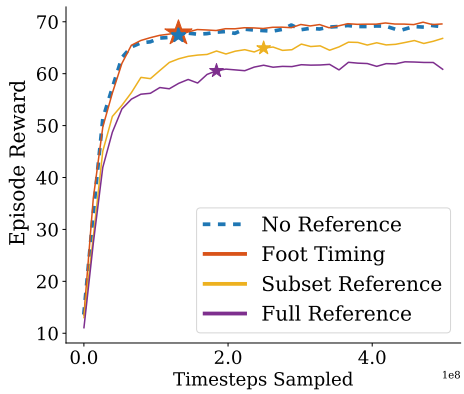

Fig. 4. Comparison of sample efficiency for our proposed turning policies. Note that the absolute scale of the different curves are not necessarily comparable since each reward function include different reward components. The star symbols mark the time to convergence for each policy, which is the first point on the learning curve that exceeds 97% of the maximum reward seen during training for the first time.

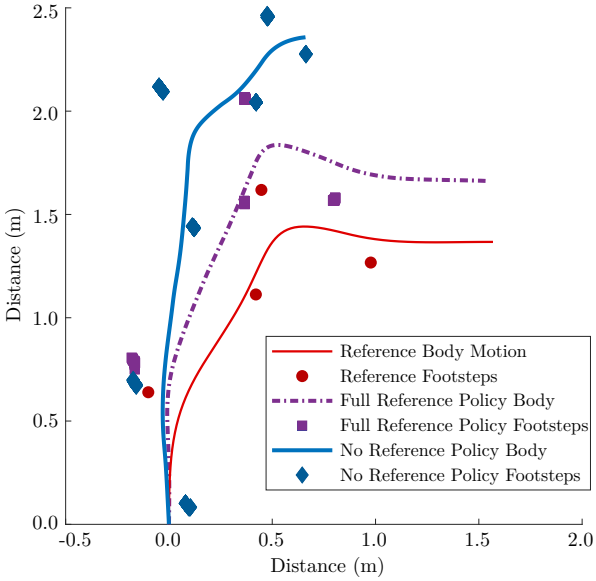

Fig. 5. Plot of footstep touchdown locations and pelvis trajectory for the reference data, *Full Reference* and *No Reference* policies for a turning maneuver executed at 2.5m/s, showing how the turning behavior differs between policies trained with and without the use of TO data. All reference-based policies used in this work executed similar footstep touchdown locations and pelvis trajectories, so only *Full Reference* is compared against *No Reference* for clarity. The ability of the SRBM to serve as a useful model for Cassie can be seen by how closely turning maneuvers executed by policies trained using reference data matched the reference trajectory.

2) Turning Behavior: Fig. 5 compares the turning trajectory of the *Full Reference* and *No Reference* policies for

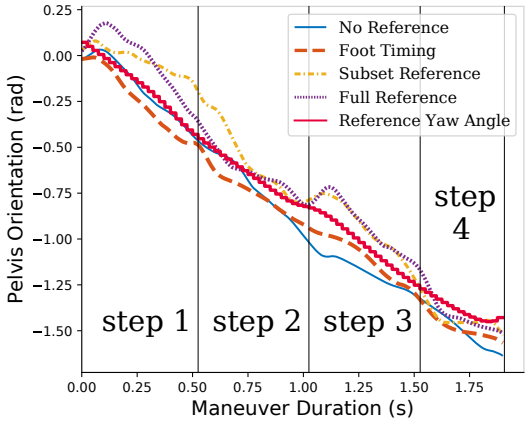

Fig. 6. Simulated pelvis yaw angles for various turning policies over the course of a turning maneuver. The Foot Timing and No Reference policies were not trained to track the reference data yaw trajectory shown in solid red. The Full and Subset Reference policies deviate the most from the target yaw angle in order to fulfill competing reward objectives, and their motions also qualitatively appear the best in simulation.

a single sample trial in simulation against the trajectory prescribed by the reference data. Since the *No Reference* policy is trained to match a footstep contact schedule set by a heuristic instead of following the trajectory data, it completes the 90° turn in seven steps rather than four. As a result, the pelvis trajectory and footstep placements differ from the reference data since it is trained to not track the reference data. This is in contrast to the *Full Reference* policy turning behavior, where the features of the pelvis trajectory is similar to that of the reference data, and the placement of its stance feet relative to the body also closely match those of the reference. Policies trained on subsets of the tracking rewards all produce four-step turning behaviors similar to the results of the *Full Reference* policy, indicating that the only necessary reference trajectory information for training policies to perform four-step 90° turns is a feasible footstep contact schedule. However, our results explained in the next section suggest that the choice of data tracked by the reward function from the trajectory data seem to affect the robustness of policies trained using it. Fig. 6 illustrates the change in orientation of the robot pelvis over the course of a turning maneuver, which is not communicated by the pelvis COM trajectories illustrated in Fig. 5. The *Full Reference* and *Subset Reference* policies are the two policies rewarded to track the optimized body yaw angle trajectory, but deviate noticeably from the target yaw trajectories at the beginning of the first and third footsteps. This is likely caused by the policies learning to maximize rewards of multiple conflicting objectives from the reference data, such as pelvis linear velocity and body yaw angles. The *No Reference* and *Foot Timing* policies on the other hand do not use the reference yaw data.

3) Policy Robustness: We simulate 1000 trials of 2.5m/s turning maneuvers for each turning policy to assess its ability to complete a turn and switch back to π^{walk} successfully in the presence of a constant perturbance force applied to the body during the execution of the turning maneuver. A

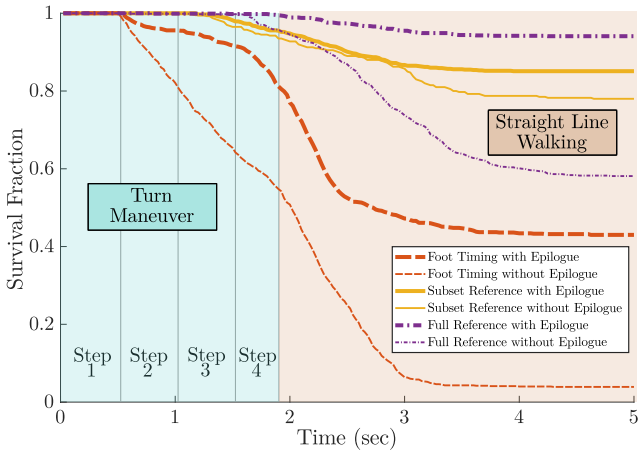

Fig. 7. Robustness comparison between *Full Reference*, *Subset Reference* and *Foot Timing* policies trained with epilogue and the same policies trained without using the same method as Fig. 8. The epilogue improved the likelihood of successful walking transitions for all turning policies.

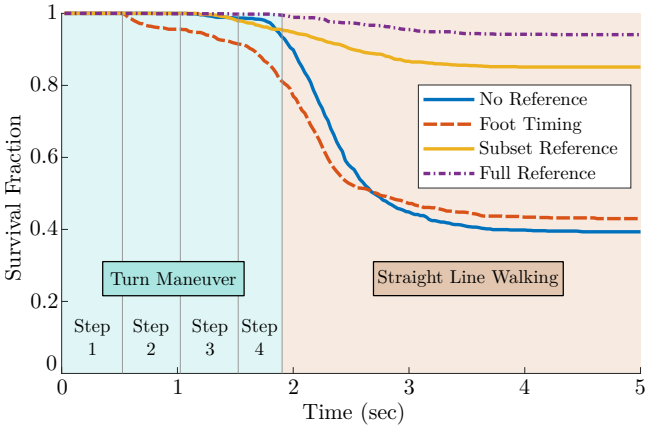

Fig. 8. Robustness comparison between our proposed turning policies conducted for 1000 turning maneuver trials at 2.5m/s. The step labels denote the reference data step progression timings produced by TO. Since the *No Reference* policy is trained to follow a contact schedule set by a heuristic, the step labels do not apply to this policy.

random direction is sampled before each turning maneuver trial, and a constant force of 35N is applied in the chosen direction. The time at which the the policy falls over is logged for each trial to produce the policy survival plots shown in Figs. 7 and 8. From Fig. 8, policies trained with more of the reference trajectory data are more robust at rejecting perturbance forces, with the exception that the *No Reference* policy outperforms the *Foot Timing* policy during the turn maneuver. The *Full Reference* policy was the most successful at completing the four step turn without falling. Fig. 7 compares the effects of training with and without the use of epilogue rewards in simulation. Policies trained without the epilogue used a terminal value $V_\pi(S_T)$ computed by simply evaluating the walking policy π^{walk} critic network. Across all reward functions, policies trained using the epilogue are observed to be more likely to successfully switch to walking than their counterparts trained without.

B. Hardware Results

During our outdoor hardware tests, we were able to demonstrate successful turning maneuvers and policy switching on artificial turf with the *No Reference* and *Full Reference* policies. The *Subset Reference* policy was also tested, but we were unable to switch back to the walking policy without falling. Due to the logistical challenges of running field tests for dynamic maneuvers, we are unable to provide quantitative reliability data and performance metrics for turning policies on hardware. We also successfully tested the *Foot Timing* policy indoors on multiple low speed turns performed in succession. While our simulation results indicate that the *Full Reference* policy should perform more consistently than the *No Reference* policy on real hardware, we observed that the *No Reference* policy was more consistent than the *Full Reference* policy at turning and transitioning in our outdoors tests. On hardware, the *Full Reference* and *Subset Reference* policies modulate the pelvis pitch, in contrast to the *No Reference* and *Foot Timing* policies, which keep the pelvis fairly level throughout the turn. This is consistent with what we see in simulation, although the *Full Reference* pelvis pitching motion in simulation appeared much more dynamic and continuous than our corresponding hardware results, where an awkward downward angle is maintained throughout the entire turn. We also found that the pelvis orientation state estimates deviated significantly from the actual orientation during turning attempts of *Full Reference* and *Subset Reference* policies. The *No Reference* and *Foot Timing* policies did not appear to suffer this state estimation bug, suggesting an issue with the sensor readings that feed our orientation state estimates under certain situations. This may be related to why our hardware results differ from what our simulation results predict. Despite this, the *Full Reference* policy performed the best on hardware out of the reference based policies for turning at speed. We refer readers to the attached video for full hardware results.

V. CONCLUSIONS

In this work, we present a novel learning framework to generate aperiodic behaviors using SRBM trajectories on the bipedal robot Cassie. From our simulation rewards, we see that the *Full Reference* policy trained to track all the data available from the reference trajectory is the most robust policy tested in sim. Policies trained to track minimal to no reference data are significantly less capable than the *Full Reference* at rejecting disturbances. From our results, the availability of reference turning trajectories aid policies to discover good strategies for executing the desired maneuver. Epilogue rewards introduced in this work have the potential to significantly improve the probability of successful transitions between different locomotion policies. However, our tests also indicate that the efficacy of the epilogue may be dependent on the choice of reward function. While our methods exhibited promising results in simulation, we encountered difficulties with sim-to-real and were unable to fully transfer the success of our simulation results to hardware field trials. The *Full Reference* still performed the best on hardware out of the reference based policies, but was not able to execute transitions back to the walking policy as consistently as the *No Reference* policy. A possible reason for the performance gap might stem from issues with the hardware state estimator producing inaccurate orientation estimates when large pelvis accelerations are experienced during the execution of turning maneuvers at higher speeds. Our reference-based policies command much larger pelvis pitch angles over the course of a turn than the *No Reference* and *Foot Timing* policies which may have exacerbated the state estimation problems.

Although we only address learning a 4-step right 90° turn maneuver in this work, our methods can clearly be applied to learning other singular turning maneuvers at different angles. We can also see our framework being able to learn multiple turns given a more diverse trajectory library set that includes turns at varying angles and different number of steps. Such a policy could likely interpolate between the turn maneuvers in the library and learn to generalize, though it is unclear how many different turns a single policy could handle.

One drawback of this learning framework is its tedious implementation procedure and lack of ability to scale to handle diverse behavior trajectory libraries, as there are at most $n(n-1)$ number of transition policies to train for a library of n behaviors. Future avenues of research could build upon this work by investigating how to effectively switch between large sets of individual behavior policies in order to allow for the execution of more complex dynamic routines such as dancing or parkour. Quantitatively identifying the limit to behaviors that can be produced by a single policy network would be a useful insight to know when switching between multiple policies should be considered. Using data driven methods to learn motion skill embeddings that are then used to train control policies similar to [12] are also promising next steps.

REFERENCES

[1] S. Kim and P. M. Wensing, "Design of dynamic legged robots," *Foundations and Trends® in Robotics*, vol. 5, no. 2, pp. 117–190, 2017. [Online]. Available: http://dx.doi.org/10.1561/2300000044

[2] H.-W. Park, P. M. Wensing, and S. Kim, "Jumping over obstacles with mit cheetah 2," *Robotics and Autonomous Systems*, vol. 136, p. 103703, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889020305431

[3] B. Katz, J. D. Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6295–6301.

[4] J. Siekmann, Y. Godse, A. Fern, and J. Hurst, "Sim-to-real learning of all common bipedal gaits via periodic reward composition," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 7309–7315.

[5] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind Bipedal Stair Traversal via Sim-to-Real Reinforcement Learning," in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.

[6] R. Batke, F. Yu, J. Dao, J. Hurst, R. L. Hatton, A. Fern, and K. Green, "Optimizing bipedal maneuvers of single rigid-body models for reinforcement learning," 2022. [Online]. Available: https://arxiv.org/abs/2207.04163

[7] T. Haarnoja, A. Zhou, S. Ha, J. Tan, G. Tucker, and S. Levine, "Learning to walk via deep reinforcement learning," *ArXiv*, vol. abs/1812.11103, 2019.

[8] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abc5986

[9] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, 2019.

[10] R. Hafner, T. Hertweck, P. Klöppner, M. Bloesch, M. Neunert, M. Wulfmeier, S. Tunyasuvunakool, N. Heess, and M. A. Riedmiller, "Towards general and autonomous learning of core skills: A case study in locomotion," in *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, vol. 155. PMLR, 2020, pp. 1084–1099. [Online]. Available: https://proceedings.mlr.press/v155/hafner21a.html

[11] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. A. Eslami, M. Riedmiller, and D. Silver, "Emergence of locomotion behaviours in rich environments," 2017. [Online]. Available: https://arxiv.org/abs/1707.02286

[12] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters," *ACM Trans. Graph.*, vol. 41, no. 4, jul 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530110

[13] S. Starke, I. Mason, and T. Komura, "Deepphase: Periodic autoencoders for learning motion phase manifolds," *ACM Trans. Graph.*, vol. 41, no. 4, jul 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530178

[14] N. Chentanez, M. Müller, M. Macklin, V. Makoviychuk, and S. Jeschke, "Physics-based motion capture imitation with deep reinforcement learning," *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*, 2018.

[15] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Trans. Graph.*, vol. 40, no. 4, jul 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459670

[16] A. Escontrela, A. Iscen, J. Peng, K. Goldberg, P. Abbeel, T. Zhang, and W. Yu, "Adversarial motion priors make good substitutes for complex reward functions," 2022.

[17] A. Hereid, O. Harib, R. Hartley, Y. Gong, and J. W. Grizzle, "Rapid trajectory optimization using c-frost with illustration on a cassie-series dynamic walking biped," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4722–4729.

[18] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014. [Online]. Available: https://doi.org/10.1177/0278364913506757

[19] S. Kuindersma, R. Deits, M. F. Fallon, A. K. 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, pp. 429–455, 2016.

[20] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 295–302.

[21] T. Apgar, P. Clary, K. Green, A. Fern, and J. W. Hurst, "Fast online trajectory optimization for the bipedal robot cassie," *Robotics: Science and Systems XIV*, 2018.

[22] Q. Nguyen, M. J. Powell, B. Katz, J. D. Carlo, and S. Kim, "Optimized jumping on the mit cheetah 3 robot," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7448–7454.

[23] X. Xiong and A. Ames, "Bipedal hopping: Reduced-order model embedding via optimization-based control," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3821–3828, 2018.

[24] M. Bjelonic, R. Grandia, M. Geilinger, O. Harley, V. S. Medeiros, V. Pajovic, E. Jelavic, S. Coros, and M. Hutter, "Offline motion libraries and online mpc for advanced mobility skills," *The International Journal of Robotics Research*, June 2022. [Online]. Available: https://doi.org/10.1177/02783649221102473

[25] C. Hennick, "Leaps, bounds, and backflips," https://blog.bostondynamics.com/atlas-leaps-bounds-and-backflips, 2021, accessed: 2022-05-19.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[27] J. Dao, K. Green, H. Duan, A. Fern, and J. Hurst, "Sim-to-real learning for bipedal locomotion under unsensed dynamic loads," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022.

[28] M. S. Jones, "Optimal control of an underactuated bipedal robot," Master's thesis, Oregon State University, 2014.

[29] J. Siekmann, S. Valluri, J. Dao, F. Bermillo, H. Duan, A. Fern, and J. Hurst, "Learning Memory-Based Control for Human-Scale Bipedal Locomotion," in *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020.

[30] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. The MIT Press, 2020.