Proceedings of Machine Learning Research vol 120:1–10, 2020 2nd Annual Conference on Learning for Dynamics and Control

Improving Input-Output Linearizing Controllers for Bipedal Robots via Reinforcement Learning

Fernando Castañeda[1] FCASTANEDA@BERKELEY.EDU
Mathias Wulfman[1] MATHIAS_WULFMAN@BERKELEY.EDU
Ayush Agrawal[1] AYUSH.AGRAWAL@BERKELEY.EDU
Tyler Westenbroek[2] WESTENBROEKT@BERKELEY.EDU
Claire J. Tomlin[2] TOMLIN@BERKELEY.EDU
S. Shankar Sastry[2] SHANKAR_SASTRY@BERKELEY.EDU
Koushil Sreenath[1] KOUSHILS@BERKELEY.EDU
[1] *Department of Mechanical Engineering, University of California at Berkeley, USA*
[2] *Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, USA*

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

Abstract

The main drawbacks of input-output linearizing controllers are the need for precise dynamics models and not being able to account for input constraints. Model uncertainty is common in almost every robotic application and input saturation is present in every real world system. In this paper, we address both challenges for the specific case of bipedal robot control by the use of reinforcement learning techniques. Taking the structure of a standard input-output linearizing controller, we use an additive learned term that compensates for model uncertainty. Moreover, by adding constraints to the learning problem we manage to boost the performance of the final controller when input limits are present. We demonstrate the effectiveness of the designed framework for different levels of uncertainty on the five-link planar walking robot RABBIT.

Keywords: legged robots, feedback control, reinforcement learning, model uncertainty

1. Introduction

1.1. Motivation

Research on humanoid walking robots is gaining in popularity due to the robots' medical applications as exoskeletons for people with physical disabilities and their usage in dangerous disaster and rescue missions. Model-based controllers have traditionally been applied to obtain stable walking controllers but, in general, they heavily rely on having perfect model knowledge and unlimited torque capacity. In this paper we take a data-driven approach to address these two topics of current research interest which still constitute challenges in bipedal robot control: uncertainty in the dynamics and input saturation.

1.2. Related work

Input-output linearization is a nonlinear control technique that can be used to get the outputs of a nonlinear system to track desired reference trajectories in a simple manner. By introducing an appropriate state transformation, this control technique permits rendering the input-output dynamics

© 2020 F. Castañeda, M. Wulfman, A. Agrawal, T. Westenbroek, C.J. Tomlin, S.S. Sastry & K. Sreenath.

linear. Afterward, linear systems control theory can be used to track the desired outputs. However, input-output linearization requires precise knowledge of the system's dynamics, which directly conflicts with the fact that actual systems' dynamics might have nonlinearities that can be extremely challenging to model precisely. Several efforts have been made to address this issue, using different methods including robust and adaptive control techniques (Nguyen and Sreenath, 2015; Sastry and Bodson, 1989; Craig et al., 1986; Sastry and Isidori, 1989) or, more recently, data-driven learning methods (Taylor et al., 2019; Westenbroek et al., 2019). This paper will take the later approach to address this challenge, specifically combining reinforcement learning (RL) and the Hybrid Zero Dynamics (HZD) method for getting bipedal robots to walk.

The high nonlinearity, underactuation and hybrid nature of bipedal robotic systems pose additional problems that need to be addressed. The virtual constraints and HZD methods (Grizzle et al., 2001; Westervelt et al., 2002; Westervelt, 2003; Morris and Grizzle, 2005) provide a systematic approach to designing asymptotically stable walking controllers if there is full model knowledge. These methods have been very successful in dealing with the challenging dynamics of legged robots, being able to achieve fast enough convergence to guarantee stability over several walking steps. By the HZD method, a set of output functions is chosen such that, when they are driven to zero, a time-invariant lower-dimensional zero dynamics manifold is created. Stable periodic orbits designed on this lower-dimensional manifold are also stable orbits for the full system under application of, for instance, input-output linearizing (Sreenath et al., 2011), or control Lyapunov function (CLF) based controllers (Ames et al., 2014). The later is based on solving online quadratic programs, whereas the former approach does not rely on running any kind of online optimization. The CLF-based method has also been successful in taking into account torque saturation (Galloway et al., 2015), but it assumes perfect model knowledge too. In fact, taking input saturation into account is of major importance and not doing it is one of the main disadvantages of input-output linearization controllers that is often overlooked.

In this work, we build on the formulation proposed in Westenbroek et al. (2019) wherein policy optimization algorithms from the RL literature are used to overcome large amounts of model uncertainty and learn linearizing controllers for uncertain robotic systems. Specifically, we extend the framework introduced in Westenbroek et al. (2019) to the class of hybrid dynamical systems typically used to model bipedal robots using the HZD framework. Unlike the systems considered in Westenbroek et al. (2019), here we must explicitly account for the effects of underactuation when designing the desired output trajectories for the system to ensure that it remains stable. Additionally, we demonstrate that a stable walking controller can be learned even when input constraints are added to the system. By focusing on learning a stabilizing controller for a single task (walking), we are able to train our controller using significantly less data than was used in Westenbroek et al. (2019), where it was trained to track all possible desired output signals.

1.3. Contributions

The contributions of our work thus are:

- We extend the work in Westenbroek et al. (2019) to the case of hybrid, underactuated bipedal robots with input constraints.
- We directly address the challenge of dealing with a statically unstable underactuated system, designing a new training strategy that uses a finite-time convergence feedback controller to track desired walking trajectories.

- We perform Poincaré analysis to claim local exponential stability of our proposed RL-enhanced input-output linearization controller in the presence of torque saturation.

1.4. Organization

The rest of the paper is organized as follows. Section 2 briefly revisits hybrid systems theory for walking and input-output linearization. Section 3 develops the proposed RL framework that improves the input-output linearizing controller when there is a mismatch between the model and the plant dynamics. Section 4 presents simulations on perturbed models of RABBIT, a five-link planar bipedal robot. Finally, Section 5 provides concluding remarks.

2. Input-Output Linearization of Bipedal Robots

2.1. Model Description

Bipedal walking is represented as a hybrid model with single-support continuous-time dynamics and double-support discrete-time impact dynamics (1), with $x \in \mathbb{R}^{2n}$ being the robot state, and $u \in \mathbb{R}^m$ the control inputs. x^- and x^+ represent the state before and after impact, respectively, with $\mathcal{S}$ being the switching surface when the swing leg contacts the ground and Δ being the discrete-time impact map. The constrained continuous-time dynamics are represented in the manipulator form (2), where $q \in \mathbb{R}^n$ is the vector containing the generalized system's coordinates, $D(q)$ is the inertia matrix of the system, $C(q, \dot{q})$ is the matrix representing the centripetal and Coriolis effects, $G(q)$ is the gravitation terms vector, $B(q)$ is the motor torque matrix, $J(q)$ is the Jacobian of the stance foot and λ is the ground contact forces vector. The state variables are $x = [q, \dot{q}]^\top$.

$$\mathcal{H} = \begin{cases} \dot{x} = f(x) + g(x)u, & x^- \notin \mathcal{S}, \\ x^+ = \Delta(x^-), & x^- \in \mathcal{S}. \end{cases} \tag{1}$$

$$\begin{cases} D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(q)u + J^\top(q)\lambda, \\ J(q)\ddot{q} + \dot{J}(q,\dot{q})\dot{q} = 0. \end{cases} \tag{2}$$

2.2. Input-Output Linearization

The output function $y : \mathbb{R}^{2n} \to \mathbb{R}^m$ is defined to represent the walking gait. Supposing y has a vector relative degree two —meaning that the first derivative of y does not depend on the inputs but the second derivative does— the second derivative of y can be written as:

$$\ddot{y} = L_f^2 y(x) + L_g L_f y(x) u. \tag{3}$$

The functions $L_f^2 y$ and $L_g L_f y$ are known as second order Lie derivatives. More information about Lie derivatives and how to compute them can be found in Sastry (1999). Moreover, using the method of Hybrid Zero Dynamics (HZD) the output function and its first derivative are driven to zero, imposing "virtual constraints" such that the system evolves on the lower-dimensional zero dynamics manifold, given by $Z = \{x \in \mathbb{R}^{2n} |\ y(x) = 0,\ \dot{y}(x) = 0\}$. If the vector relative degree is well-defined, then $L_g L_f y(x) \neq 0\ \forall\ x \in D$, with $D \subset \mathbb{R}^{2n}$ being a compact subset of the state space containing the origin. Since $L_g L_f y$ is nonsingular in D, we can use the input-output linearizing control law:

$$u(x) = L_g L_f y^{-1}(x)(-L_f^2 y(x) + v), \tag{4}$$

which yields $\ddot{y} = v$, where v is a virtual input.

Suppose a state transform $\Phi : x \to (\xi, z)$, with $\xi = [y, \dot{y}]^\top$ and $z \in Z$. Then, the closed-loop dynamics become a linear time-invariant system on ξ and the zero-dynamics on z:

$$\begin{cases} \dot{\xi} = A\xi + Bv, \\ \dot{z} = p(\xi, z), \end{cases} \quad \text{with } A = \begin{bmatrix} 0_{m\times m} & I_m \\ 0_{m\times m} & 0_{m\times m} \end{bmatrix} \text{ and } B = \begin{bmatrix} 0_{m\times m} \\ I_m \end{bmatrix}. \tag{5}$$

We define v following Westervelt et al. (2007):

$$v(\xi) = \frac{1}{\epsilon^2}\psi_a(y, \epsilon\dot{y}), \quad \text{with} \quad \begin{cases} \psi_a(y, \epsilon\dot{y}) = -sign(\epsilon\dot{y})|\epsilon\dot{y}|^a - sign(\phi_a(y, \epsilon\dot{y}))|\phi_a(y, \epsilon\dot{y})|^{\frac{a}{2-a}}, \\ \phi_a(y, \epsilon\dot{y}) = y + \dfrac{1}{2-a} sign(\epsilon\dot{y})|\epsilon\dot{y}|^{2-a}, \end{cases} \tag{6}$$

such that v ensures finite time convergence to Z and ϵ controls the rate of convergence.

3. Reinforcement Learning for Uncertain Dynamics

In this section, we study the case in which there is a mismatch between the model and the actual plant dynamics. Now, plant and model are represented by:

(Unknown) Plant Dynamics	(Known) Model Dynamics
$\begin{cases} \dot{x} = f_p(x) + g_p(x)u, \\ y = h_p(x), \end{cases}$ (7)	$\begin{cases} \dot{x} = f_m(x) + g_m(x)u, \\ y = h_m(x). \end{cases}$ (8)

For our application we will be using the same output functions for plant and model, so we could actually set $h_p \equiv h_m$. Furthermore, we assume that both systems have vector relative degree two. Defining an input-output linearizing controller on the model dynamics using the state dependent finite-time convergence feedback controller presented in (6) for the additional input v we get:

$$u(x) = (L_{g_m} L_{f_m} h_m(x))^{-1} (-L_{f_m}^2 h_m(x) + v(x)). \tag{9}$$

However, if the mismatch between the model and the real dynamics is big enough, this controller may not manage to stabilize the plant. In order to address this issue we use an alternative control input:

$$u_\theta(x) = \Big(L_{g_m} L_{f_m} h_m(x)\Big)^{-1}\Big(- L_{f_m}^2 h_m(x) + v(x)\Big) + \alpha_\theta(x)v(x) + \beta_\theta(x), \tag{10}$$

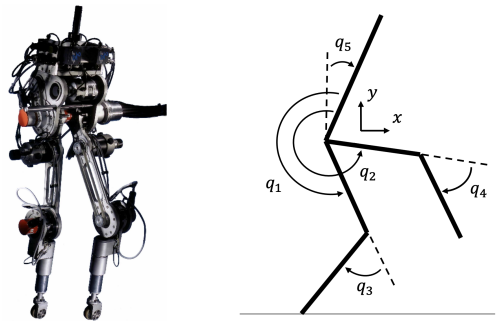
where $\theta \in \mathbb{R}^k$ is a vector of parameters of a neural network that are to be learned. For a specific θ, the policies $\alpha_\theta : \mathbb{R}^n \to \mathbb{R}^{m\times m}$, $\beta_\theta : \mathbb{R}^n \to \mathbb{R}^m$ take the current state as input and serve to define an additive learned term that is affine in v. Note that u_θ maintains the structure of an input-output linearizing controller. Applying the new control law u_θ, the second derivative of the plant's outputs can be rewritten as:

$$\ddot{y} = L_{f_p}^2 h_p(x) + L_{g_p} L_{f_p} h_p(x) \left(\Big(L_{g_m} L_{f_m} h_m(x)\Big)^{-1} \Big(-L_{f_m}^2 h_m(x) + v(x)\Big) + \alpha_\theta(x)v(x) + \beta_\theta(x) \right). \tag{11}$$

In Westenbroek et al. (2019), W_θ is defined as the right hand side of the above equation, such that $\forall x \in \mathbb{R}^{2n}$, $\ddot{y} = W_\theta(x)$. The point-wise loss is then defined on $\mathbb{R}^{2n} \times \mathbb{R}^k$ as:

$$l(x, \theta) = ||v(x) - W_\theta(x)||_2^2, \tag{12}$$

which provides a measure of how well the controller u_θ linearizes the plant at the state x. Since the term W_θ present in the loss function depends on the unknown plant dynamics, we use a finite difference approximation of it by replacing this by the second derivative of the outputs of the plant.



(*a*) RABBIT (*b*) Coordinate system

Figure 1: (a) RABBIT, a planar five-link bipedal robot with nonlinear, hybrid and underactuated dynamics. (b) q_1, q_2 are the relative stance and swing leg femur angles referenced to the torso, q_3, q_4 are the relative stance and swing leg knee angles, q_5 is the absolute torso angle in the world frame, and x and y are the position of the hip in the world frame. Here $q = [x, y, q_1, q_2, q_3, q_4, q_5]^\top$.

Now, we will formulate our problem as a canonical RL problem (Sutton and Barto, 2018). Even though only α_θ and β_θ are learned, for the sake of simplicity let $\pi_\theta : x \mapsto \pi_\theta(x)$ be our policy taking the current state x and returning the control action $u_\theta = \pi_\theta(x)$, and let the reward for a given state x be $R(x, u_\theta) = -l(x, \theta) + R_e(x)$, where $R_e(x)$ is a penalty value if the state x is associated with a fallen robot configuration or a bonus value otherwise. Then, we can define the learning problem

$$\begin{aligned} \max_{\theta} \quad & \mathbb{E}_{x_0 \sim X_0, w \sim \mathcal{N}(0,\sigma^2)} \int_0^T R(x(\tau), u_\theta(\tau)) d\tau, \\ \text{s.t.} \quad & \dot{x} = f(x) + g(x)(\pi_\theta(x) + w_t), \\ & u_{min} \leq \pi_\theta(x) \leq u_{max}, \end{aligned} \tag{13}$$

where X_0 is the initial state distribution, $T > 0$ is the duration of the episode, w is an additive zero-mean noise term and u_{min} and u_{max} are the torque limits. An episode ends when the robot completes an entire step or when it falls. A discrete-time approximation of this problem can be solved using standard on-policy and off-policy RL algorithms. Note that our proposed controller (10) with the chosen loss (12) and the inclusion of input constraints in the optimization (13) addresses the classical challenges of input-output linearization: model uncertainty and input constraints. From now on, we will call *original IO controller* the one of (9) and *RL-enhanced IO controller* the one of (10), with θ chosen by solving (13).

4. Simulation

4.1. System Description

In order to numerically validate our method, we use a model of the five-link planar robot RABBIT (Chevallereau et al., 2003), wherein the stance phase is parametrized by a suitable set of coordinates (Figure 1). RABBIT is a 7 Degrees-of-Freedom (DOF) underactuated system with 4 actuated DOF, with the actuators being located at the four joints (the two hip joints and the two knee joints). The dynamics of this 14-dimensional system are extremely coupled and nonlinear.

4.2. Reference Trajectory Generation

In order to generate a reference trajectory offline, we use the Fast Robot Optimization and Simulation Toolkit (FROST) (Hereid and Ames, 2017). The four actuated DOF (q_1, q_2, q_3 and q_4) are virtually constrained to be Bézier Polynomials of the stance leg angle $\theta = q_5 + q_1 + \frac{q_3}{2}$, which is monotonically increasing during a walking step. This way, the trajectory that has

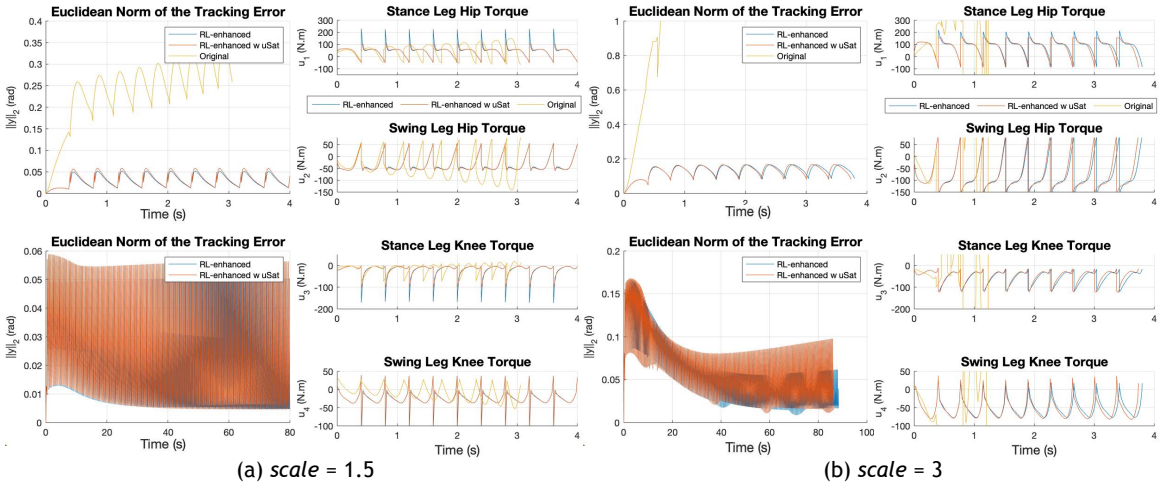


Figure 2: Euclidean norm of the tracking error (for 10 and 200 steps) and joint torques (for 10 steps), for the *original IO controller* (yellow), the *RL-enhanced IO controller* (blue), and the *RL-enhanced IO controller* with torque saturation (red). Torque saturation for the *RL-enhanced IO controller* is set at 105 Nm when $scale = 1.5$, and at 155 Nm when $scale = 3$. There is no torque saturation for the *original IO controller*.

been generated is time-invariant, which makes the controlled system more robust to uncertainties (Westervelt et al., 2007). Taking the difference between the actual four actuated joint angles and the desired ones (coming from the reference trajectory) as output functions y, the system is input-output linearizable with vector relative degree two. Consequently, we can use the *RL-enhanced IO controller* u_θ presented in the previous section.

We train our controller using a Deep Deterministic Policy Gradient Algorithm (DDPG) (Silver et al., 2014). DDPG is used to tune the parameters of the actor and critic feedforward neural networks. They each have two hidden layers of widths 400 and 300 and ReLU activation functions. The actor neural network maps 14 observations, which are the states of the robot, to 20 outputs corresponding to the 4×4 α_θ and the 4×1 β_θ.

4.3. Model-Plant Mismatch and Torque Saturation Results

We introduce model uncertainty by scaling all the masses and inertia values of the plant's links by some factor (*scale*) with respect to the known model. After about twenty minutes of training when the *scale* is 1.5 and about an hour when the *scale* is 3, we obtain the results shown in Figure 2, in which we compare the tracking error and the joint torques when using (i) the *original IO controller*, (ii) the *RL-enhanced IO controller* without torque saturation and (iii) the *RL-enhanced IO controller* when there is torque saturation. For these results we did not need to include torque saturation in the training process, and Figure 2 shows that the *RL-enhanced IO controller* still performs well in the presence of input constraints if they are not too severe. The beneficial effects of including torque saturation constraints during training will be discussed later.

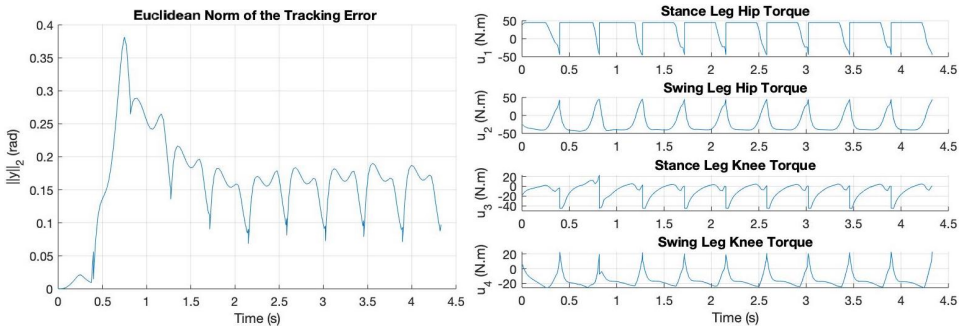


Figure 3: *RL-enhanced IO controller* with torque saturation at $45\ Nm$ and $scale = 1$. Euclidean norm of the tracking error (left) and joint torques (right) for a simulation of 10 walking steps. The *original IO controller* fails after one step and is not shown in this figure.

In Figure 2 it can be observed that the *RL-enhanced IO controller* with and without saturation is able to stabilize the system indefinitely each time, whereas the *original IO controller* accumulates error on the outputs and the robot falls after a few steps. Moreover, the *RL-enhanced IO controller* achieves this without increasing the magnitude of the torques when compared with the *original IO controller*.

The stability of the periodic gait obtained under the *RL-enhanced IO controller* can also be studied by the method of Poincaré. We consider the post-impact double stance surface S as a Poincaré section, and define the Poincaré map $P : S \to S$. We can numerically calculate the eigenvalues of the linearization of the Poincaré map about the obtained periodic gait, which results in a dominant eigenvalue of magnitude 0.67 for $scale = 1.5$ and no torque saturation, 0.78 for $scale = 1.5$ with torque saturation, 0.76 for $scale = 3$ and no torque saturation and 0.83 for $scale = 3$ with torque saturation. The magnitude of the dominant eigenvalue being always less than one means that the designed controllers achieve local exponential stability (Westervelt et al., 2007).

Next, we study the case of having no mismatch between the plant and the model dynamics but, instead, having heavy input constraints in the torques, which make the *original IO controller* fail. By training while taking into account the torque saturation, we obtain a *RL-enhanced IO controller* that achieves stable walking under the presence of severe input constraints, as shown in Figure 3.

4.4. Tracking Untrained Trajectories

Depicted in Figure 4 are the tracking errors and torques produced by the *RL-enhanced IO controller* for a *scale* of 3 when it is trying to follow periodic orbits it was not trained on. These trajectories differ from the one used for the training (*trajectory 1*) in the maximum hip height during a step. As can be seen in the left part of Figure 4, *trajectory 2* and *trajectory 1* are relatively similar, whereas *trajectory 3* constitutes a noticeably different walking gait. From the figures, we can see that the *RL-enhanced IO controller* performs better when tested in *trajectory 2* than in *trajectory 3*. Actually, it will be able to stably track *trajectory 2* for an indefinitely long horizon and not *trajectory 3*. This was expected, since the more different the trajectory is, the farther the state of the robot will be from the distribution of states the DDPG agent has been trained on. Also, the output functions we have defined depend on the Bézier coefficients of the reference trajectory, and so the actual input-output linearizing controller is different for each trajectory. Still, thanks to training the DDPG agent on a stochastic distribution of initial states, we get enough exploration to achieve good

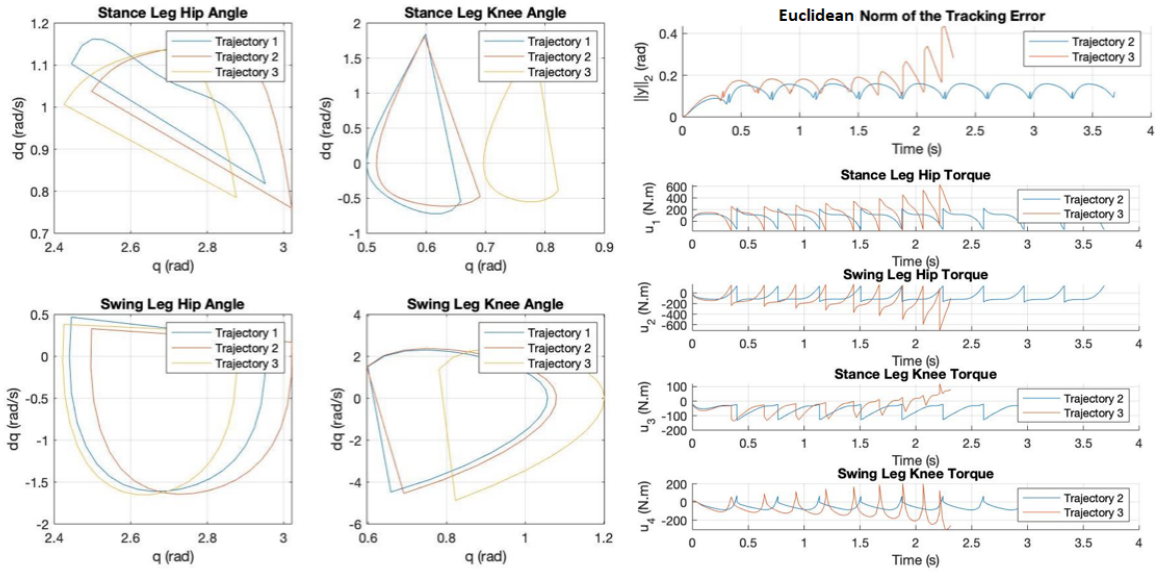


Figure 4: Left: Phase portrait of the periodic orbits. Right: Euclidean norm of the tracking error and joint torques for a simulation of 10 steps on untrained trajectories.

tracking performance on untrained trajectories as long as they are not too different from the one the agent was trained on.

5. Conclusions

In this paper, we deployed a framework for improving an input-output linearizing controller for a bipedal robot when uncertainty in the dynamics and input constraints are present. We demonstrated the effectiveness of this approach by testing the learned controller on the hybrid, nonlinear and underactuated five-link walker RABBIT. For the simulations, different degrees of model-plant mismatch with and without torque saturation were used. Furthermore, the *RL-enhanced IO controller* was able to follow trajectories it was not trained on as long as these trajectories were not too different from the one used for the training. However, a limitation of our work is the need for the *original IO controller* to work for a significant part of a walking step before failing, in order for the training process to converge. For high degrees of uncertainty this could be difficult to guarantee.

Future work would focus on deploying this controller on hardware and on other more complex bipedal walkers, such as Cassie. Moreover, a similar approach could be used to improve Control Lyapunov Function (CLF)-based controllers in the presence of model uncertainty.

Acknowledgments

The work of Fernando Castañeda was supported by a fellowship (code LCF/BQ/AA17/11610009) from ”la Caixa” Foundation (ID 100010434). This work was also partially supported through National Science Foundation Grants CMMI-1931853, IIS-1834557, by Berkeley Deep Drive and by HICON-LEARN (design of HIgh CONfidence LEARNing-enabled systems), Defense Advanced Research Projects Agency award number FA8750-18-C-010.

References

A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 59(4): 876–891, April 2014.

C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. R. Westervelt, C. Canudas-De-Wit, and J. W. Grizzle. Rabbit: a testbed for advanced control theory. *IEEE Control Systems Magazine*, 23(5): 57–79, 2003.

J. Craig, Ping Hsu, and S. Sastry. Adaptive control of mechanical manipulators. *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, 3:190–195, 1986.

K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle. Torque saturation in bipedal robotic walking through control lyapunov function-based quadratic programs. *IEEE Access*, 3:323–332, 2015.

J. W. Grizzle, G. Abba, and F. Plestan. Asymptotically stable walking for biped robots: analysis via systems with impulse effects. *IEEE Transactions on Automatic Control*, 46(1):51–64, 2001.

A. Hereid and A. D. Ames. Frost: Fast robot optimization and simulation toolkit. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 719–726, Vancouver, BC, Canada, September 2017.

B. Morris and J. W. Grizzle. A restricted poincaré map for determining exponentially stable periodic orbits in systems with impulse effects: Application to bipedal robots. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4199–4206, 2005.

Q. Nguyen and K. Sreenath. L1 adaptive control for bipedal robots with control lyapunov function based quadratic programs. *Proceedings of the American Control Conference*, pages 862–867, July 2015.

S. Sastry. *Nonlinear Systems: Analysis, Stability and Control*. Springer Science + Business Media, 1999. ISBN 978-1-4757-3108-8.

S. Sastry and M. Bodson. *Adaptive Control: Stability, Convergence, and Robustness*. Prentice-Hall Inc., 1989. ISBN 0-13-004326-5.

S. S. Sastry and A. Isidori. Adaptive control of linearizable systems. *IEEE Transactions on Automatic Control*, 34(11):1123–1131, November 1989.

D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. *Proceedings of the 31st International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2014.

K. Sreenath, H.-W. Park, I. Poulakakis, and J. Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel. *The International Journal of Robotics Research*, 30:1170–1193, August 2011.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Second Edition, MIT Press, Cambridge, MA, 2018. ISBN 978-0262193986.

A. J. Taylor, V. D. Dorobantu, H. M. Le, Y. Yue, and A. D. Ames. Episodic learning with control lyapunov functions for uncertain robotic systems. *arXiv preprint arXiv:1903.01577*, 2019.

T. Westenbroek, D. Fridovich-Keil, E. Mazumdar, S. Arora, V. Prabhu, S. S. Sastry, and C. J. Tomlin. Feedback linearization for unknown systems via reinforcement learning. *arXiv preprint arXiv:1910.13272*, 2019.

E. Westervelt. *Toward a Coherent Framework for the Control of Planar Biped Locomotion*. PhD thesis, University of Michigan, 2003.

E. R. Westervelt, J.W. Grizzle, and D.E. Koditschek. Zero dynamics of underactuated planar biped walkers. *IFAC Proceedings Volumes*, 35(1):551–556, 2002.

E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris. *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007. ISBN 1-42005-372-8.