Reinforcement learning of potential fields to achieve limit-cycle walking

Feirstein (student), D.S.; Koryakovskiy, Ivan; Kober, Jens; Vallery, Heike

DOI
10.1016/j.ifacol.2016.07.994

Publication date
2016

Document Version
Accepted author manuscript

Published in
Proceedings of the 6th IFAC Workshop on Periodic Control Systems (PSYCO 2016)

Reinforcement Learning of Potential Fields to achieve Limit-Cycle Walking

Denise S. Feirstein * **Ivan Koryakovskiy** * **Jens Kober** **
Heike Vallery *

* *TU Delft Department of BioMechanical Engineering*
** *Delft Center for Systems and Control*

Abstract: Reinforcement learning is a powerful tool to derive controllers for systems where no models are available. Particularly policy search algorithms are suitable for complex systems, to keep learning time manageable and account for continuous state and action spaces. However, these algorithms demand more insight into the system to choose a suitable controller parameterization. This paper investigates a type of policy parameterization for impedance control that allows energy input to be implicitly bounded: Potential fields. In this work, a methodology for generating a potential field-constrained impedance control via approximation of example trajectories, and subsequently improving the control policy using Reinforcement Learning, is presented. The potential field-constrained approximation is used as a policy parameterization for policy search reinforcement learning and is compared to its unconstrained counterpart. Simulations on a simple biped walking model show the learned controllers are able to surpass the potential field of gravity by generating a stable limit-cycle gait on flat ground for both parameterizations. The potential field-constrained controller provides safety with a known energy bound while performing equally well as the unconstrained policy.

Keywords: Machine learning, Energy Control, Limit cycles, Walking, Robot control

1. INTRODUCTION

The demand for robot control that is both safe and energy-efficient is greater than ever with advances in mobile robots and robots that interact in human environments. One such example is the bipedal robot which has applications ranging from home care to disaster relief. Traditional position control, common to industrial robotics, is not suitable for robots that interact in unknown environments because slight position errors can result in high contact forces that can damage the robot and its environment. In the case of humanoid robots which interact in human environments this poses a human-safety issue.

One possible solution is to employ impedance control, which attempts to enforce a dynamic relation between system variables as opposed to controlling them directly (Hogan (1984)). Specifically, impedance control based on potential fields, which inherently bounds the energy exchanged between the robot and the environment. Potential fields can modulate natural dynamics of a system and achieve desired behavior without requiring high-stiffness trajectory tracking. Potential fields have been developed for path planning and motion control by reformulating the objective into a potential function (Koditschek (1987)). Control torques can be represented as a vector field generated by the gradient of the potential field, such that the dimensionality of any number of actuators is essentially reduced to one, the scalar value of the potential function.

Potential fields can only release energy stored inside them, such that they can be classified as a passive control method. Motion control based on passivity generates robust motions not only in real time but also autonomously, while allowing simple task objectives, such as walking speed or reaching targets (Hyon and Cheng (2006)). Contrasting the high energy demand of conventional, fully actuated bipedal robots, passive dynamic walkers have been developed that walk down shallow slopes using only the force of gravity and the robot's natural dynamics (McGeer (1990)). Thus, these mechanisms exploit the natural potential field of gravity. In consequence, they possess an extremely energy-efficient gait that is remarkably similar to that of humans. The stable periodic gait of a passive dynamic walker is referred to as a Limit Cycle (LC). Rendering this gait slope-invariant and improving its disturbance rejection has been the focus of many publications including Hobbelen and Wisse (2007). For example, walking of the so-called simplest walker on flat terrain can be achieved by emulating a slanted artificial gravity field via robot actuators (Asano and Yamakita (2001)). This is a very special case of a potential field.

The design and parameterization of more generic potential fields remains challenging, particularly for systems that exhibit modeling uncertainties or are subjected to unknown disturbances. Reinforcement learning (RL) is a powerful technology to derive controllers for systems where no models are available. Policy search RL methods, also known as actor-only methods, have been found effective for robotic applications due to their ability to handle higher dimensionality and continuous state and action spaces compared to Value-based RL methods (Kober et al. (2013)). Furthermore, policy search methods have been

effectively implemented on bipedal robots (Tedrake et al. (2004)).

In this work, we propose to combine RL and PF-constrained impedance control to improve robot safety for robots that operate in uncertain conditions because:

- PF-constraint provides safety with a known energy bound
- RL provides controllers for systems with modeling uncertainty.

The question arises, can policy search RL be combined with potential fields to achieve LC walking? While the theoretical advantage of a PF-constrained impedance control, specifically energy boundedness, are presented in literature, the sub-question arises, are there limitations when it comes to RL convergence?

As a first step towards answering these questions, this paper presents a methodology for defining a potential field-constrained (PF-constrained) impedance control and improving it via reinforcement learning. To achieve this, we define an impedance control as a parameterized mapping of configurations to control torques, which is analogous to a policy in Reinforcement Learning (RL) algorithms. A PF-constrained and an unconstrained parameterization of an impedance controller are compared before and after RL applied to the bipedal walking problem. These control methods are compared for three cases: the reference case of the simplest walking model (SWM), the slope-modified case of the SWM on flat ground, and the mass-modified case, of the SWM with modified foot mass on flat ground.

This paper is organized as follows: In Section 2, we describe the parameter optimization method for deriving an initial impedance control policy for an unconstrained and a PF-constrained parameterization. In Section 3, we describe a policy search reinforcement learning algorithm that uses the control policies defined in Section 2 as an initial guess. In Section 4 we describe how this method can be applied to the LC walking problem. In sections 5, we present our evaluation protocol for comparing the unconstrained and PF-constrained impedance control for the bipedal walking. In Section 6, we present our results followed by our discussion in Section 7. Finally, in Section 8 we present our conclusions and suggestions for future work.

2. IMPEDANCE CONTROL INITIALIZATION

As opposed to conventional set-point control approaches that directly control system variables such as position and force, impedance control attempts to enforce a dynamic relation between these variables (Hogan (1984)). In this section, an open-loop impedance controller is derived for a fully actuated robot with n Degrees Of Freedom (DOF) using least squares optimization. The controller is open-loop in that it does not use feedback to determine if the output matches a desired value. We assume an accurate model of the robot as well as the ability to measure the position and torque at each joint as well as full collocated actuation. Each configuration of the robot can be described by a unique vector $\boldsymbol{q} = [q_1, q_2, ..., q_n]^T$ where q_n, with index $i = 1...n$, are the generalized coordinates.

If a desired trajectory, $\boldsymbol{x} = \left(\boldsymbol{q}_\mathrm{d}^T,\ \dot{\boldsymbol{q}}_\mathrm{d}^T,\ \ddot{\boldsymbol{q}}_\mathrm{d}^T\right)^T$, is known, the idealistic control torques, $\boldsymbol{\tau}_0$, required to achieve this trajectory can be found using inverse dynamics. A function to approximate the torques applied to the system as a function of the robot's configuration, $\boldsymbol{\tau}(\boldsymbol{q}) \in \mathbb{R}^n$, can be found by formulating the least squares problem

$$\Big(\boldsymbol{\tau}_{0,k}(\boldsymbol{x}_k) - \boldsymbol{\tau}(\boldsymbol{q}_k)\Big)^2 \longrightarrow \min \tag{1}$$

where $\boldsymbol{\tau}_{0,k}(\boldsymbol{x}_k)$, $k = 1...S$, is a set of training data with S samples and $\boldsymbol{\tau}(\boldsymbol{q}_k)$ can be approximated as normalized radial basis functions (RBF) $\boldsymbol{G}(\boldsymbol{q})$, parameterized by weighting vector $\boldsymbol{w}$ such that

$$\boldsymbol{\tau}_k = \boldsymbol{G}(\boldsymbol{q}_k)\boldsymbol{w} \tag{2}$$

The choice of $\boldsymbol{G}(\boldsymbol{q})$ will be discussed in the following subsections.

Defining vector $\boldsymbol{b} = (\tau_{0,1}...\tau_{0,S})$ and matrix $\mathbf{A} = \big(\boldsymbol{G}(\boldsymbol{q}_1)...\boldsymbol{G}(\boldsymbol{q}_S)\big)$, the least squares estimate of $\boldsymbol{w}$, denoted $\hat{\boldsymbol{w}}$ can be formulated as the minimization problem

$$\min_{\hat{\boldsymbol{w}}} \|\boldsymbol{b} - \mathbf{A}\hat{\boldsymbol{w}}\|_\mathbf{Q}^2 \tag{3}$$

which is dependent on the number of training samples, S. The symmetric positive definite weighting matrix $\mathbf{Q}$ contains weights that reflect the importance of certain joints or training samples. The parameter vector $\boldsymbol{w}$ can be found using the pseudoinverse. The solution can also be found recursively if there is a large amount of training data. The procedure for recursive least squares given by Papageorgiou (2012) was modified to include weighting of various parameters such as training data, joints, and torque magnitude.

2.1 Unconstrained Parameterization

The vector function $\boldsymbol{\tau}(\boldsymbol{q})$ can be defined in terms of its components $\tau_i(\boldsymbol{q})$, $i = 1...n$, where n is the number of degrees of freedom, and parameterized as normalized radial basis functions of the form

$$\tau_i(\boldsymbol{q}) = \frac{\sum_{j=1}^N w_{i,j} f_j[r_j(\boldsymbol{q})]}{\sum_{j=1}^N f_j[r_j(\boldsymbol{q})]} = \boldsymbol{g}(\boldsymbol{q})^T \boldsymbol{w}_i \tag{4}$$

where N is the number of basis functions, $w_{i,j}$ is the j^{th} parameter of the i^{th} weighting vector, f_j is an RBF, r_j is a radius function and $\boldsymbol{g}(\boldsymbol{q})$ is a vector function. For the unconstrained case $\boldsymbol{g}(\boldsymbol{q})$ is used as $\boldsymbol{G}(\boldsymbol{q})$ in Equation 2.

Radius functions r_i are scalar functions of the distance vector $\boldsymbol{\delta}_i$ and scaling factor s which defines the size of the radial basis function:

$$r_j(\boldsymbol{q}) = s\,\|\boldsymbol{\delta}_j\|\,. \tag{5}$$

$\boldsymbol{\delta}_j$ describes the distance from the center point $\boldsymbol{c}_j$ of the j^{th} RBF to the joint configurations $\boldsymbol{q}$:

$$\boldsymbol{\delta}_j(\boldsymbol{q}) = \boldsymbol{q} - \boldsymbol{c}_j. \tag{6}$$

For the RBF, f_j, we choose to use compactly supported radial basis functions which allow for the use of a minimal number of center points c_j in the neighborhood of the robot's position to sufficiently compute the function value (Vallery et al. (2009a)). This reduces the computational resources needed during operation.

2.2 Potential Field-constrained Parameterization

Function $\boldsymbol{\tau}(\boldsymbol{q})$ can be constrained to describe a potential field by enforcing that its work is zero for any closed-path trajectory. This implies the control torques are a function of the joint variables $\boldsymbol{q}$ and can be defined as the negative gradient of a potential function $\psi(\boldsymbol{q})$ with respect to $\boldsymbol{q}$:

$$\boldsymbol{\tau}(\boldsymbol{q}) = -\nabla_q \psi(\boldsymbol{q}). \tag{7}$$

This is similar to the method of Generalized Elasticities presented in Vallery et al. (2009a) and Vallery et al. (2009b).

Similar to Equation 4, potential function $\psi(\boldsymbol{q})$ can be parameterized as normalized radial basis functions (RBF) of the form

$$\psi(\boldsymbol{q}) = \frac{\sum_{j=1}^{N} w_j f_j[r_j(\boldsymbol{q})]}{\sum_{j=1}^{N} f_j[r_j(\boldsymbol{q})]} = \boldsymbol{g}(\boldsymbol{q})^T \boldsymbol{w}. \tag{8}$$

Unlike the unconstrained parameterization, which requires a unique weighting vector $\boldsymbol{w}_i$ for each degree of freedom, for the PF-constrained parameterization, the torques can be formulated as the gradient of the potential shown in Equation (7). This can be estimated as the transposed Jacobian of $\boldsymbol{g}(\boldsymbol{q})$:

$$\boldsymbol{\tau}(\boldsymbol{q}) = -\left(\frac{\partial \boldsymbol{g}(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^T \boldsymbol{w}. \tag{9}$$

where $\boldsymbol{G}(\boldsymbol{q}) = -\left(\frac{\partial \boldsymbol{g}(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^T$.

3. POLICY SEARCH REINFORCEMENT LEARNING

Reinforcement Learning (RL) is a machine learning method which attempts to find a control policy, $\pi(\boldsymbol{u}|\boldsymbol{x}, \boldsymbol{w})$, which maps states $\boldsymbol{x}$ to actions $\boldsymbol{u}$. For policy search algorithms, the policy is parameterized by a weighting vector $\boldsymbol{w}$. The policy is analogous to the impedance control laws derived in the previous section where generalized coordinates $\boldsymbol{q}$ are states and control torques $\boldsymbol{\tau}$ are actions.

3.1 Exploration Strategy

The policy space is explored by randomly perturbing the weighting vector $\boldsymbol{w}$. Batch exploration is performed where the policy is independently perturbed from the initial policy a set number of times. The perturbed policies are then evaluated and updated according to the strategies in the following sections.

3.2 Evaluation Strategy

The performance of the policy is numerically evaluated by computing the expected return J, which is a sum of the expected reward R. Based on the expected return J the policy is updated with the objective to find a policy which maximizes the expected return J. The policy evaluation strategy determines how to evaluate the performance of an executed policy by using a reward function, $R(\boldsymbol{x}, \boldsymbol{u})$. For a finite-horizon model, this corresponds to maximizing the expected reward for the horizon H over h steps. The series of states and actions over H steps is called an episode. The expected return is calculated

$$J = E\left\{\sum_{h=0}^{H} R_h\right\}. \tag{10}$$

Episode-based policy evaluation uses the entire episode to assess the quality of the policy used directly (Deisenroth et al. (2011)).

3.3 Update Strategy

The policy is updated based on the performance of the previous policy or set of policies. Policy search methods optimize around an initial policy $\pi(\boldsymbol{u}|\boldsymbol{x}, \boldsymbol{w}_0)$. The policy is iteratively updated using an update strategy that computes changes in the policy parameter in a way that increases the expected return.

Several update strategies for episode-based policy search have been developed. One method developed in Kober and Peters (2011) specifically for motor primitives in robotics is Expectation Maximization Policy learning by Weighted Exploration with the Returns (PoWER). The iterative policy search method with episode-based evaluation is summarized in Algorithm 1.

Algorithm 1 Policy Search using Expectation Maximization PoWER

Initialize: Generate initial episode using policy π_0 parameterized by $\boldsymbol{w}_0$. Compute return J_0.
repeat
Explore: Perform $i = 1 : N$ episodes using perturbed policy parameters $\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$
For each episode compute return J_i
Reweight: Compute importance weights, keep 10 high-importance episodes, discard low-importance episodes.
Update: Compute updated policy
$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \left\langle \sum_i^{10} J_i \right\rangle^{-1} \left\langle \sum_i^{10} \epsilon_i J_i \right\rangle$
until Policy converge

4. APPLICATION TO LC WALKING

4.1 Simplest Walking Model

The simplest walking model (SWM) developed in Garcia et al. (1998) is often used as a tool to study the paradigm of Bipedal Limit-Cycle walking and is detailed in the following sections. A diagram of the SWM is shown in Figure 1.

The model consists of two massless rigid links of length L connected at the hip by a frictionless hinge. The mass is distributed over three point masses at the hip and feet such that the hip mass m_h is much larger than the foot mass m_f. The model is situated on a slope of angle γ and acts only under the force of gravity with acceleration constant g. The configuration of the model is given by the ankle angle θ and hip angle ϕ. The generalized coordinates are $\boldsymbol{q} = (x_\mathrm{c}, y_\mathrm{c}, \theta, \phi)^T$ where the subscripts "c" denotes the contact point of the stance foot with the ground. The model is actuated at the ankle and hip.

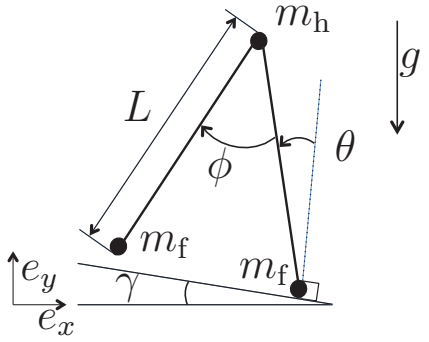

Fig. 1. Diagram of the Simplest Walking Model which consists of two massless links, point mass m_h at the hip and m_f at each foot walking on ground of slope γ. The generalized coordinates θ and ϕ are the angle of the stance leg perpendicular to the slope and the inter-leg (or hip) angle respectively.

4.2 Inverse dynamics

A vector of the global coordinates of the point masses is $\boldsymbol{p} = (x_\mathrm{st}, y_\mathrm{st}, x_\mathrm{hip}, y_\mathrm{hip}, x_\mathrm{sw}, y_\mathrm{sw})^T$ where subscripts "st" and "sw" denote the stance leg and swing leg respectively and subscript "hip" denotes the hip. The generalized coordinates can be transformed to Cartesian positions using transfer function $\boldsymbol{p} = \boldsymbol{F}(\boldsymbol{q})$. The equations of motion can then be found using the virtual power equation

$$\delta\dot{\boldsymbol{p}}^T[\boldsymbol{f} - \mathbf{M}\ddot{\boldsymbol{p}}] = 0. \tag{11}$$

where $\mathbf{M}$ is the global mass matrix defined $\mathbf{M}$ =Diag$(m_\mathrm{f}, m_\mathrm{f}, m_\mathrm{h}, m_\mathrm{h}, m_\mathrm{f}, m_\mathrm{f})$. The resulting equations of motion are

$$[\mathbf{F}_{,\boldsymbol{q}}^T\mathbf{M}\mathbf{F}_{,\boldsymbol{q}}]\ddot{\boldsymbol{q}} = \mathbf{F}_{,\boldsymbol{q}}^T[\boldsymbol{f}_g - \mathbf{M}\mathbf{F}_{,\boldsymbol{qq}}\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}] + \boldsymbol{Q} \tag{12}$$

where the subscript comma operator followed by $\boldsymbol{q}$ denotes partial derivative by $\boldsymbol{q}$, and $\boldsymbol{f}_g$ are the applied forces due to gravity given

$$\boldsymbol{f}_g = \mathbf{M}[\sin\gamma, -\cos\gamma, \sin\gamma, -\cos\gamma, \sin\gamma, -\cos\gamma]^T \tag{13}$$

and $\boldsymbol{Q} = (Q_{x_c}, Q_{y_c}, Q_\theta, Q_\phi)^T$ are the generalized forces. For unactuated cases, Q_θ and Q_ϕ both equal zero. The contact forces at the stance foot Q_{x_c} and Q_{y_c} are only valid for $Q_{y_c} > 0$. In this case $\boldsymbol{Q}$ is known and $\ddot{\boldsymbol{q}}$ can be found using the ordinary differential equation

$$\ddot{\boldsymbol{q}} = \frac{\mathbf{F}_{,\boldsymbol{q}}^T[\boldsymbol{f}_g - \mathbf{M}\mathbf{F}_{,\boldsymbol{qq}}\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}] + \boldsymbol{Q}}{[\mathbf{F}_{,\boldsymbol{q}}^T\mathbf{M}\mathbf{F}_{,\boldsymbol{q}}]}. \tag{14}$$

The unactuated model exhibits an LC gait for a limited set of combinations of initial conditions and slopes, called the basin of attraction. In the case we would like to deviate from the original basin of attraction (for example by modifying the slope and/or model mass) while still maintaining an LC gait, idealistic actuator torques can be derived using inverse dynamics of the known LC joint trajectories to find the generalized forces. Rearranging Equation 12 and replacing $\mathbf{M}$ and $\boldsymbol{f}_g$ with $\mathbf{M}_\mathrm{mod}$ and $\boldsymbol{f}_{g,\mathrm{mod}}$ respectively, gives the inverse dynamics equation

$$\begin{aligned}\boldsymbol{Q}_0 =& [\mathbf{F}_{,\boldsymbol{q}}^T\mathbf{M}_\mathrm{mod}\mathbf{F}_{,\boldsymbol{q}}]\ddot{\boldsymbol{q}}_d \\ &- \mathbf{F}_{,\boldsymbol{q}}^T[\boldsymbol{f}_{g,\mathrm{mod}} - \mathbf{M}_\mathrm{mod}\mathbf{F}_{,\boldsymbol{qq}}\dot{\boldsymbol{q}}_d\dot{\boldsymbol{q}}_d]\end{aligned} \tag{15}$$

where training data $(\dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d)$ is required, $\boldsymbol{f}_{g,\mathrm{mod}}$ corresponds to the applied forces from the modified slope and $\mathbf{M}_\mathrm{mod}$ corresponds to the modified global mass matrix. From this point forward $\boldsymbol{\tau}_0 = [Q_{\theta,0}, Q_{\phi,0}]^T$ will be used to denote the generalized forces at the ankle and hip joints corresponding to the idealistic applied motor torques.

The training data was found by first, scanning the initial conditions $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ for cases in which the SWM converges to an LC and then the associated accelerations $\ddot{\boldsymbol{q}}$ were found using Equation 14. The resulting training data can represented by the vector $\boldsymbol{x} = \left(\boldsymbol{q}_d^T,\ \dot{\boldsymbol{q}}_d^T,\ \ddot{\boldsymbol{q}}_d^T\right)^T = \left(\theta,\ \phi,\ \dot{\theta},\ \dot{\phi},\ \ddot{\theta},\ \ddot{\phi}\right)^T$.

For scanning the initial conditions, the ankle angle was varied between 0.1 and 0.2 rad with a step size of 0.005 rad, and the initial hip angle was set to twice that of the ankle so the model initializes in double support phase. The initial ankle angular velocity was varied between -0.68 and -0.38 rad/s with a step size of 0.005 rad/s, and the initial hip angular velocity was set to 0 rad/s. The result is shown in Figure 2.

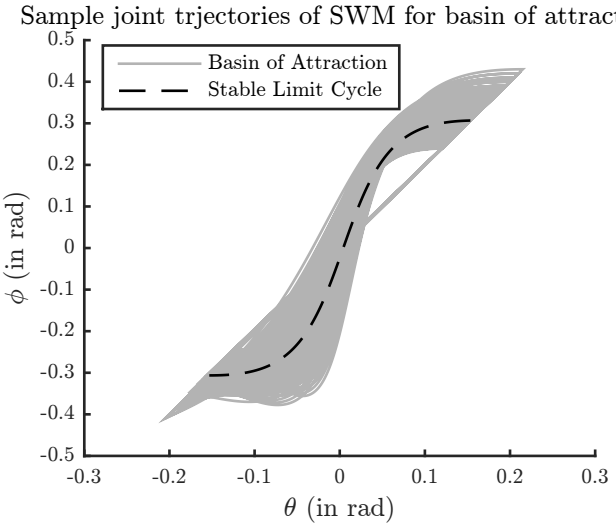

Fig. 2. Joint trajectories of the basin of attraction for the simple walking model with $\gamma = 0.004$ rad, $m_h = 1$ kg, $m_f = 0.001$ kg, $L = 1$ m, and $g = 10$ m/s^2

The torques $\boldsymbol{\tau}_0$ found from training data $\boldsymbol{x}$ can be used to solve the least-squares problem in Equation (3) using the recursive least-squares method described in Section 2 resulting in impedance control laws of the form $\boldsymbol{\tau}(\boldsymbol{q})$.

4.3 Reinforcement Learning

The resulting impedance control laws $\boldsymbol{\tau}(\boldsymbol{q})$ parameterized by vector $\boldsymbol{w}$ are specific to the simplest walking model case and will likely not be effective if the model is modified or more degrees of freedom are added. If this is the case $\boldsymbol{\tau}(\boldsymbol{q})$ parameterized by vector $\boldsymbol{w}_0$ can be used as the initial policy for policy search RL. The policy search with episode-based evaluation strategy described in Section 3 can be used where one episode is H steps of the biped. For a biped robot, the state transitions from the previous state $\boldsymbol{x}$ to the next state $\boldsymbol{x}'$ caused by actions $\boldsymbol{u}$ can be modeled by solving the equations of motion (14) using iterative methods where $\boldsymbol{x} = (\boldsymbol{q}^T,\ \dot{\boldsymbol{q}}^T)^T$ are the states and the generalized forces Q_θ, Q_ϕ are the actions $\boldsymbol{u}$.

To evaluate the quality of parameter vector $\boldsymbol{w}_k$, the return for each episode is calculated

$$J_k = \sum_{h=0}^{H} R_h. \tag{16}$$

The reward function used for each step is

$$\begin{aligned} R_h(\boldsymbol{x}, \boldsymbol{u}) = &+ R_{\text{step}} - R_{\Delta}||\Delta\theta|| - R_{\dot{\Delta}}||\Delta\dot{\theta}|| \\ &- R_t||t_h - t_0|| - R_{\tau,\theta}||\boldsymbol{\tau}_\theta|| - R_{\tau,\phi}||\boldsymbol{\tau}_\phi|| \end{aligned} \tag{17}$$

where $\Delta\theta = \theta_h - \theta_{h-1}$, and $\Delta\dot{\theta} = \dot{\theta}_h - \dot{\theta}_{h-1}$, and R_{step}, $R_{\Delta} = 10$ 1/rad, $R_{\dot{\Delta}} = 10$ s/rad, $R_t = 1$ 1/s, $R_{\tau,\theta} = 10$ 1/Nm and $R_{\tau,\phi} = 100$ 1/Nm are constants. The first term of the reward function is given as a reward for successfully completing a step. The second term penalizes the change in angle and angular velocity of the stance leg at the beginning of each step. This is to encourage a limit-cycle is reached where each step is the same. The third term penalized the change in time of step h from the time of the reference LC step t_0. The fourth term penalizes the magnitude of the control torques to minimize the energy added to the system.

5. EVALUATION PROTOCOL

5.1 Implementation

Simulations were performed in MATLAB to assess the impedance controllers described in the previous section. The simulations used the ODE45 integration algorithm with the following settings: absolute tolerance = 10^{-6}, relative tolerance = 10^{-3} and initial integration-step size $\Delta t = 0.02$ s. In the case of an "odezero (internal error)" the ODE45 settings were temporarily set to: absolute tolerance = 10^{-8}, relative tolerance = 10^{-5}. The event detection was used to determine when a step occurs using the step condition: $\theta - \phi/2 = 0$ and $\dot{\theta} < 0$.

The impedance control laws were implemented on a fully-actuated simple walking model for the three cases: the reference case of the simplest walking model (SWM) on a slope, the slope-modified case of the SWM on flat ground, and the mass-modified case, of the SWM with modified foot mass on flat ground. For all cases the leg length, hip mass and gravity remained constant at $L = 1$ m, $m_{\text{h}} = 1$ kg and $g = 10$ m/s^2 respectively.

For the least squares optimization, 50 RBFs were used. The center locations were determined using a grid step size of 0.05 rad for the ankle angle and 0.1 rad for the hip angle in the area of the ideal trajectory of the SWM as shown in Figure 3.

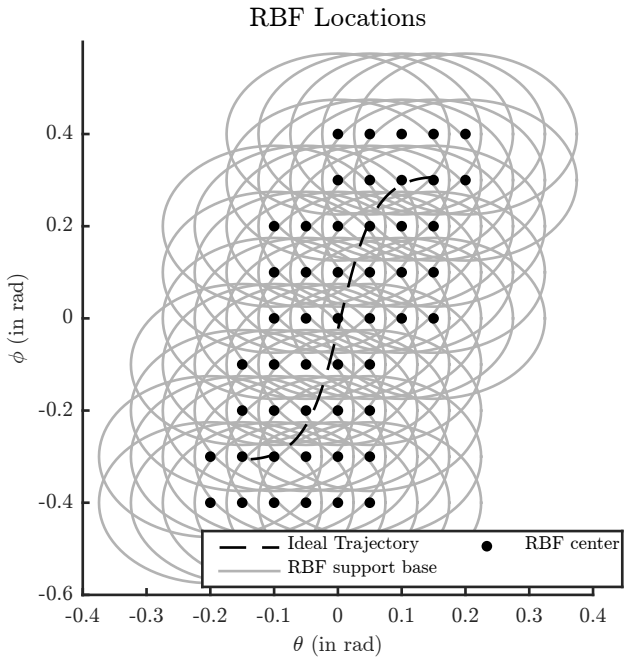

Fig. 3. RBF center locations and support base of 10 degrees.

For the policy search RL, a horizon of $H = 10$ was used corresponding to 10 steps of the robot. For the exploration strategy, a batch size of 100 iterations was used. For the reward function the constants were set to: $R_{\text{step}} = 1$, $R_{\Delta} = 10$, $R_t = 1$ and $R_{\tau} = 5$. The time of the reference LC step was $t_0 = 1.2180$ s. A Gaussian exploration $\epsilon \sim \mathcal{N}(0, \sigma^2)$ was used which was decrease linearly over episodes.

5.2 Experiment Setup

Initial unconstrained and PF-constrained impedance controllers were found using inverse dynamics for each of the three cases described below:

Reference case: For the reference case a slope of $\gamma = 0.004$ rad and foot mass $m_{\text{f}} = 0.001$ kg was used.

Slope-modified case: For the slope-modified case a slope of $\gamma = 0$ rad and foot mass $m_{\text{f}} = 0.001$ kg was used.

Mass-modified case: For the modified-mass case, a foot mass of $m_{\text{f}} = 0.01$ kg was used. This is 10 times the value of the reference case. A slope of $\gamma = 0$ rad was used.

For the Slope and Mass-modified cases, RL was used to attempt to improve the policy for both the unconstrained and the PF-constrained parameterizations. For the reference case, the performance of the controllers can not be improved further using RL based on the evaluation strategy since the control torques cannot decrease further.

5.3 Benchmarking Criteria

The unconstrained and PF-constrained impedance controllers were compared for each of the three cases based on the following benchmarking criteria:

Work and Energy: The energy of the LC of the ideal SWM (unactuated and on a slope) is bounded by the potential field of gravity. The energy bound can be measured as the maximum energy, E of the LC, defined $E = V + T$ where V is the potential energy and T is the kinetic energy. For the LC of the ideal SWM, the total energy is constant at 10.0108 J. At each step kinetic energy is dissipated at impact and an equivalent amount of potential energy is added by the slope. The change in kinetic energy at the end of the step can be seen in Figure 4. The energy added/dissipated at each step is equivalent to 0.0166 J.

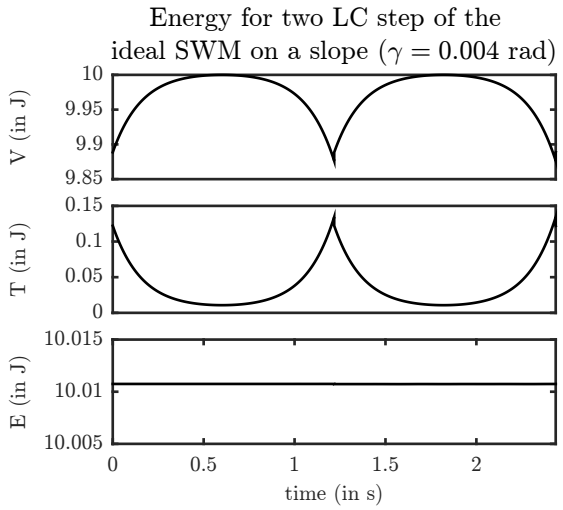

Fig. 4. Energy of the SWM on a slope of $\gamma = 0.004$ rad. The total energy is the sum of the potential (V) and kinetic (T) energy.

Energy consumption can be measured for the actuated model as the work done by the actuators:

$$W = \int_{\boldsymbol{q}_0}^{\boldsymbol{q}_1} \boldsymbol{\tau} \mathrm{d}\boldsymbol{\theta} \tag{18}$$

where $\boldsymbol{q}_0$ is the configuration at the beginning of the step and $\boldsymbol{q}_1$ is the configuration at the end of the step.

Robustness: The robustness of an LC gait can be measured by its velocity disturbance rejection. An angular velocity disturbance is introduced to the stance leg at the beginning of the first step and the maximum disturbance that can be applied without causing the walker to fall is used as a measure for robustness.

RL Performance: The performance of the RL is assessed by plotting the mean performance over the episodes, for several trials, and observing how many episodes it takes to level off.

6. RESULTS

6.1 Reference case

The trajectory phase plots for the Unconstrained and PF-constrained policies derived using inverse dynamics for the reference case are shown in Figures 5 (a) and (b) respectively. The control torque and total energy for the Unconstrained and PF-constrained policy derived using inverse dynamics for the reference case shown in Figure 6 (a) and (b) respectively.

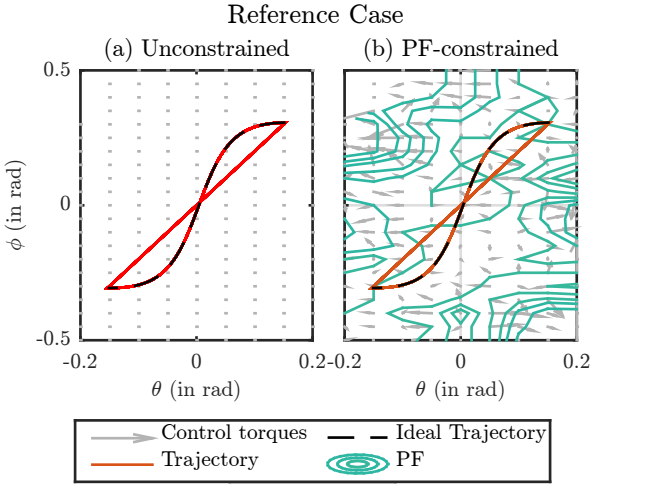

Fig. 5. Trajectory phase plot of the (a) Unconstrained and (b) PF-constrained policies for the Reference Case. The control policies are represented by a vector field and for the PF-constrained policy the contour lines of the potential field are shown.

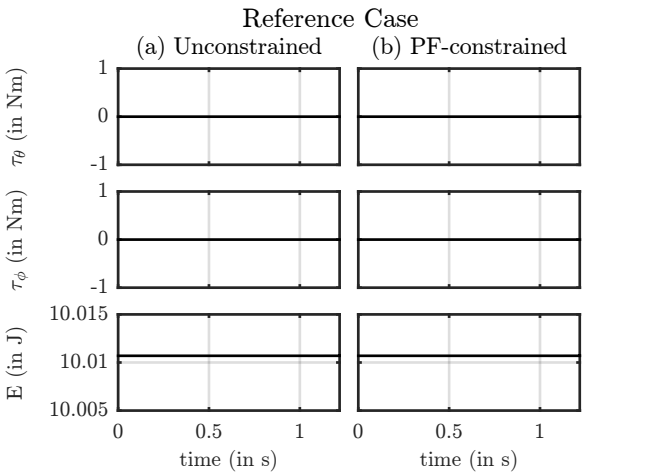

Fig. 6. Control torques and energy of one LC step of the (a) Unconstrained and (b) PF-constrained policies for the Reference Case.

The benchmarking criteria for the energy, work and robustness of the reference case are specified Table 1.

6.2 Slope-modified Case

Initialization The trajectory phase plots for the initial Unconstrained and PF-constrained policies for the Slope-modified case are shown in Figures 7 (a) and (b) respectively. The control torques and total energy for the initial Unconstrained and PF-constrained policies for the Slope-modified case are shown in Figures 8 (a) and (b) respectively.

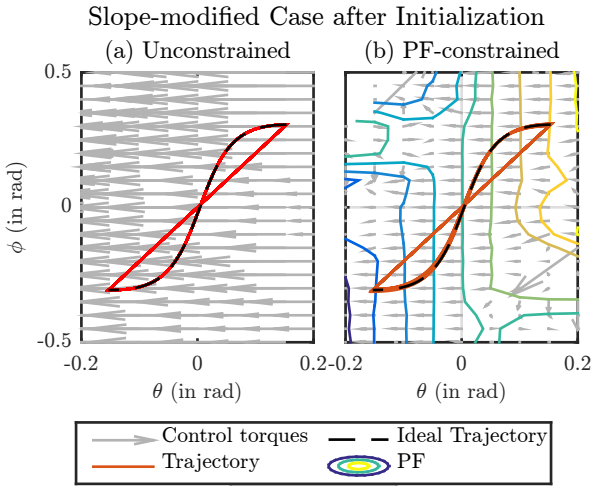

Fig. 7. Trajectory phase plot of the initial (a) Unconstrained and (b) PF-constrained policies for the Slope-modified Case. The control policies are represented by a vector field and for the PF-constrained policy the contour lines of the potential field are shown.

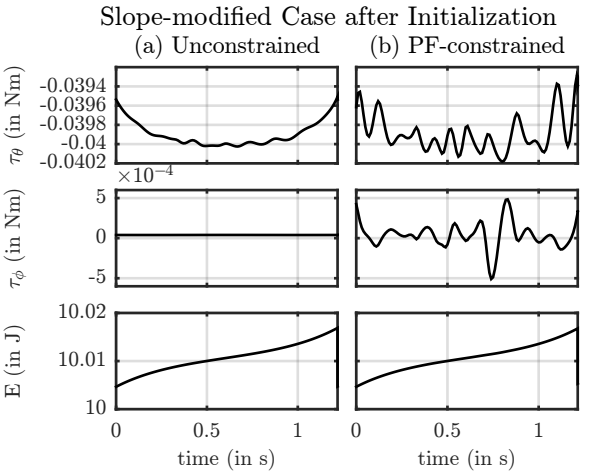

Fig. 8. Control torques and energy of one LC step of the initial (a) Unconstrained and (b) PF-constrained policies for the Slope-modified Case.

Reinforcement Learning Results The mean performance of the RL for the Unconstrained and PF-constrained controllers are shown in Figure 9. The resulting trajectory phase plot for the learned Unconstrained and PF-constrained policies for the Slope-modified case are shown in Figures 10 (a) and (b) respectively. The resulting control torques and energy for the learned Unconstrained and PF-constrained policies for the Slope-modified case are shown in Figures 11 (a) and (b) respectively.

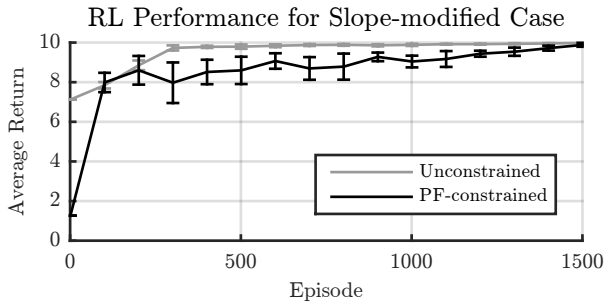

Fig. 9. Mean performance of the RL for the Unconstrained and PF-constrained policies for the Slope-modified case averaged over 10 runs with the error bars indicating the standard deviation. For both policies the exploration variance decreased linearly from 1e-6 to 1e-11 throughout the episodes.

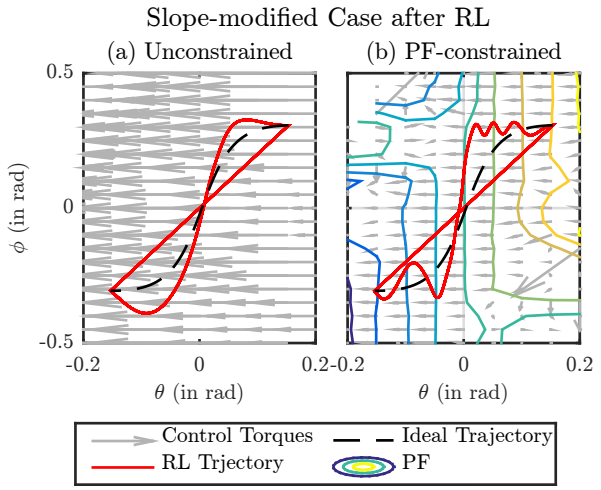

Fig. 10. Trajectory phase plot of the learned (a) Unconstrained and (b) PF-constrained policies for the Slope-modified Case. The control policies are represented by a vector field and for the PF-constrained policy the contour lines of the potential field are shown.

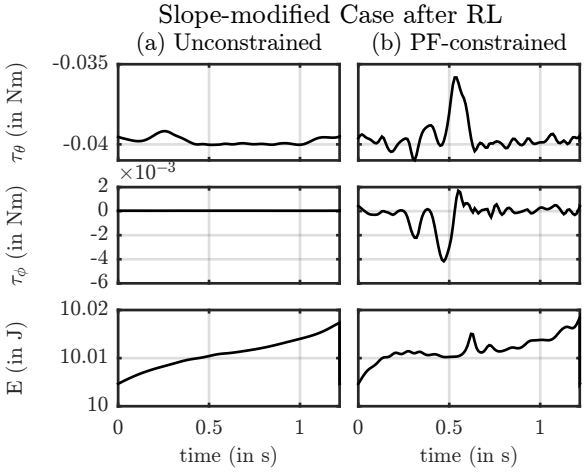

Fig. 11. Control torques and energy of one LC step of the learned (a) Unconstrained and (b) PF-constrained policies for the Slope-modified Case

The benchmarking criteria for the energy, work and robustness of the Slope-modified case are specified Table 1.

6.3 Mass-modified Case

Initialization The trajectory phase plot for the initial Unconstrained and PF-constrained policies for the Mass-modified case is shown in Figure 12 (a) and (b) respectively.

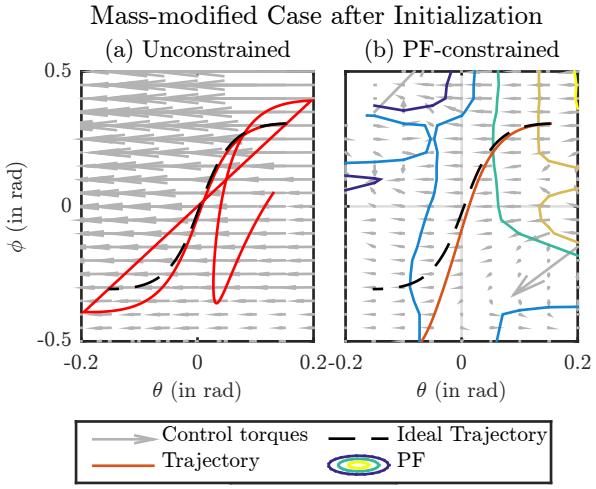

Fig. 12. Trajectory phase plot of the initial (a) Unconstrained and (b) PF-constrained policies for the Mass-modified Case.The control policies are represented by a vector field and for the PF-constrained policy the contour lines of the potential field are shown.

It can be seen that neither policy leads to a stable limit cycle so the corresponding control torque and energy plots are not shown.

Reinforcement Learning The mean performance of the RL for both the PF-constrained and unconstrained case are shown in Figure 13. The resulting trajectory phase plots for the learned Unconstrained and PF-constrained policies for the Mass-modified case are shown in Figures 14 (a) and (b) respectively. The resulting control torques and energy for the learned Unconstrained and PF-constrained policies for the Mass-modified case are shown in Figures 15 (a) and (b) respectively.

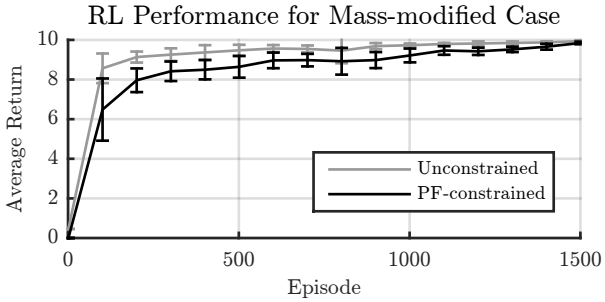

Fig. 13. RL mean performance of the Unconstrained and PF-constrained policies for the Mass-modified case averaged over 10 runs with the error bars indicating the standard deviation. For the Unconstrained policy the exploration variance decreased from 1e-5 to 1e-10 and for the PF-constrained policy the variance decreased from 1e-6 to 1e-10.

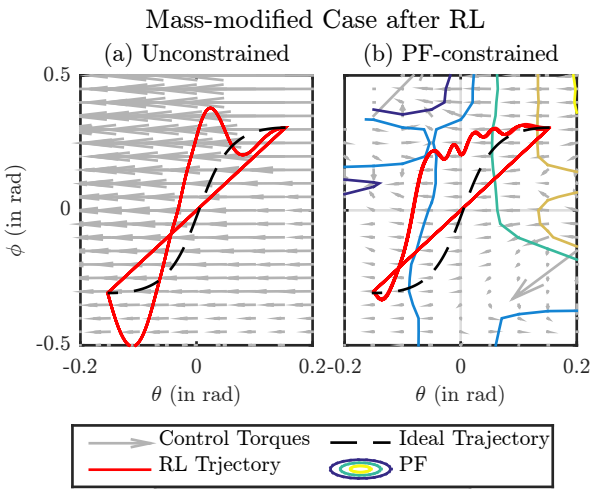

Fig. 14. Trajectory phase plot of the learned (a) Unconstrained and (b) PF-constrained policies for the Mass-modified Case. The control policies are represented by a vector field and for the PF-constrained policy the contour lines of the potential field are shown.

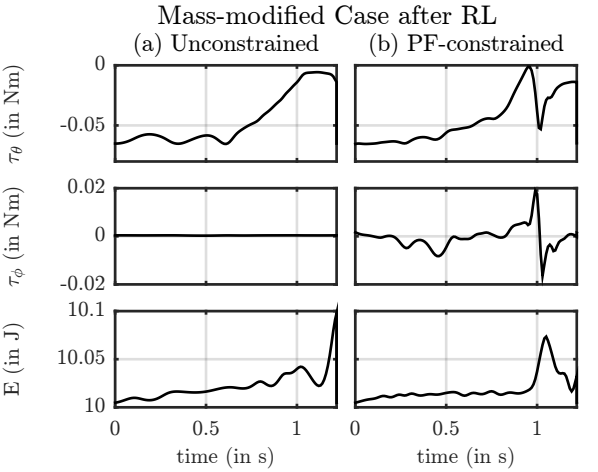

Fig. 15. Control torques and energy of one LC step of the learned (a) Unconstrained and (b) PF-constrained policies for the Mass-modified Case.

The benchmarking criteria for the work, energy and robustness of the Mass-modified case are specified Table 1.

6.4 Results Summary

The benchmarking criteria for the energy bound, work and robustness for each case are summarized in Table 1. The energy bound was determined by the max energy displayed in Figures 6, 8, 11 and 15. The work was calculated using Equation 18. The robustness is given by the max velocity disturbance rejection as described in Section 5.3. The ✗ indicates that an LC was not achieved so there was no benchmarking criteria.

Table 1. Summary of Results

Case	Parameterization	Benchmarking Criteria	Initial Policy	Learned Policy
Reference	Unconstrained	Energy bound (J)	10.0107	-
		Work (J)	0	-
		Max velocity disturbance (rad/s)	-0.05	-
	PF-constrained	Energy bound (J)	10.0107	-
		Work (J)	5e-14	-
		Max velocity disturbance (rad/s)	-0.05	-
Slope-modified	Unconstrained	Energy bound (J)	10.0169	10.0258
		Work (J)	1.5074	1.4877
		Max velocity disturbance (rad/s)	-0.05	-0.03
	PF-constrained	Energy bound (J)	10.0168	10.0185
		Work (J)	1.4948	1.3321
		Max velocity disturbance (rad/s)	-0.06	0
Mass-modified	Unconstrained	Energy bound (J)	✗	10.2146
		Work (J)	✗	1.3110
		Max velocity disturbance (rad/s)	✗	-0.05
	PF-constrained	Energy bound (J)	✗	10.0618
		Work (J)	✗	1.4811
		Max velocity disturbance (rad/s)	✗	-0.02

7. DISCUSSION

For the reference case, it can be seen in the trajectory phase plots, for both the unconstrained and PF-constrained parameterization shown in Figure 5, that the controlled trajectory perfectly follows the ideal trajectory. It can be seen in the corresponding torque and energy plots in Figure 6, that no actuator torques are generated and the energy tracks that of the unactated ideal case, as shown in Figure 4. It can be seen in Table 1 that both controllers have the same energy bound and maximum disturbance rejection as the unactuated ideal case. This serves as a validation for both the impedance controllers derived using inverse dynamics and least squares optimization.

For the slope-modified case, the initial impedance controllers, for both PF-constrained and unconstrained parameterization, allow the biped to achieve an LC gait on a flat surface ($\gamma = 0$) as can be seen in the trajectory phase plots in Figure 7. It can be seen in Table 1 that the velocity disturbance rejections are comparable to the ideal SWM, however, the energy bound is higher than the ideal case for both controllers. The work done by the actuators is similar for both controllers, however, it is almost 100 times the work done by gravity in the ideal case.

As can be seen in Table 1, RL of the initial impedance controllers for the slope-modified case increases the energy bound for both controllers, while decreasing the work done by the actuators. RL also leads to decreased disturbance rejection. As can be seen in Figures 9, the performance of the unconstrained parameterization levels off before the PF-constrained parameterization, indicating the unconstrained parameterization achieves a higher performance with less episodes compared to the PF-constrained parameterization.

For the mass-modified case, the initial impedance controllers, for both PF-constrained and unconstrained parameterizations, do not allow the biped to achieve an LC gait. This can be seen in the trajectory phase plots in Figure 12. The impedance controllers derived from inverse dynamics appear not to be able to compensate for the modified dynamics of the model.

Howerver, RL of these initial policies allows the biped to achieve an LC gait as shown in Figure 14. This validates the use of RL for achieving an LC gait. As can be seen in Table 1, for both controllers the energy bound and work done is greater than the ideal case. While the robustness of the unconstrained controller is comparable to the ideal case, it is reduced for the PF-constrained controller. As can be seen in Figures 13, the performance of the unconstrained parameterization levels off before the PF-constrained parameterization.

For all cases, the energy bound and work done by the actuators was similar for both the PF-constrained and unconstrained controllers. As the implementation of the RL did not converge to a single optimal solution, the variance in the resulting energy and work was too large to draw an accurate comparison.

For all cases, there are no improvements to the robustness of the limit-cycle against velocity disturbances. The reason for this is that the episode (consisting of H steps of the limit-cycle) is a black-box from the perspective of the episode-based RL. Learning is based only on the inputs and outputs of the episode, therefore any unknown disturbances throughout the episode are not accounted for, and consequently the robustness is not improved by the RL. Exploring and learning throughout the episode may be one way to improve the robustness. Additionally, learning could take place in an unknown environment with unknown disturbances.

The scope of these results is limited by the variables of the simple walking model used. The only modifications tested were the ratio of the hip mass to foot mass, and the slope γ.

An interesting observation is the learned behavior of "swing-leg retraction" seen in the learned policy for both cases, as shown in Figures 10 and 14 . This is when the swing leg retracts at the end of a step until it hits the ground. It has been shown in Hobbelen and Wisse (2008) that swing-leg retraction can improve disturbance rejection.

8. CONCLUSION AND FUTURE WORK

In this work we successfully combined potential field control and reinforcement learning to achieve limit-cycle walking for a simple walking model. A limit-cycle was achieved on flat ground, and for a modified hip to foot mass ratio. The results demonstrate that a potential field controller can not only "emulate" the effect of gravity on the simple walking model, but also improve its performance if reinforcement learning is applied. The potential field-constrained controller provides safety by bounding the energy while performing equally well compared to an unconstrained controller. The performance of the RL leveled off faster for the unconstrained case.

Achieving a limit cycle gait on a SMW is trivial compared to more complex models. In future work the method presented in this paper could be applied to higher degree of freedom models. A strength of this method is the ability to bound the energy of the controlled system. In future work it could be explored how to enforce a desired energy bound. Improved tuning of the RL exploration and evaluation strategy could lead to improved policies and more conclusive results for the comparison of the unconstrained and PF-constrained parameterizations. More advanced RL methods could lead to potential fields that further improve performance and even increase robustness.

ACKNOWLEDGMENT

I. Koryakovskiy and H. Vallery were supported by the European project KOROIBOT FP7-ICT-2013-10/611909.

REFERENCES

Asano, F. and Yamakita, M. (2001). Virtual gravity and coupling control for robotic gait synthesis. *Systems, Man, and Cybernetics Part A: Systems and Humans, IEEE Transactions on*, 31(6), 737–745.

Deisenroth, M.P., Neumann, G., and Peters, J. (2011). A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*, 2, 1–142.

Garcia, M., Chatterjee, A., Ruina, A., and Coleman, M. (1998). The Simplest Walking Model: Stability, Complexity, and Scaling. *Journal of Biomechanical Engineering*, 120(2), 281–288.

Hobbelen, D.G.E. and Wisse, M. (2007). Limit Cycle Walking. In M. Hackel (ed.), *Humanoid Robots: Human-like Machines*, June, pages 642–659. Vienna, Austria.

Hobbelen, D.G. and Wisse, M. (2008). Swing-leg retraction for limit cycle walkers improves disturbance rejection. *Robotics, IEEE Transactions on*, 24(2), 377–389.

Hogan, N. (1984). Impedance control: An approach to manipulation. In *American Control Conference, 1984*, 304–313. IEEE.

Hyon, S.H. and Cheng, G. (2006). Passivity-based full-body force control for humanoids and application to dynamic balancing and locomotion. *IEEE International Conference on Intelligent Robots and Systems*, 1, 4915–4922.

Kober, J., Bagnell, J.A., and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32, 1238–1274.

Kober, J. and Peters, J. (2011). Policy search for motor primitives in robotics. *Machine Learning*, 84(1-2), 171–203.

Koditschek, D.E. (1987). Exact robot navigation by means of potential functions: Some topological considerations. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, volume 4, 1–6. IEEE.

McGeer, T. (1990). Passive Dynamic Walking. *The International Journal of Robotics Research*, 9(2), 62–82.

Papageorgiou, M. (2012). *Optimierung: statische, dynamische, stochastische Verfahren*. Springer-Verlag.

Tedrake, R., Zhang, T., and Seung, H. (2004). Stochastic policy gradient reinforcement learning on a simple 3D biped. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3.

Vallery, H., Duschau-Wicke, A., and Riener, R. (2009a). Generalized elasticities improve patient-cooperative control of rehabilitation robots. In *Rehabilitation Robotics, 2009. ICORR 2009. IEEE International Conference on*, 535–541. IEEE.

Vallery, H., Duschau-Wicke, A., and Riener, R. (2009b). Optimized passive dynamics improve transparency of haptic devices. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 301–306. IEEE.