Autonomous Robots manuscript No.
(will be inserted by the editor)

Learning to Exploit Passive Compliance for Energy-Efficient Gait Generation on a Compliant Humanoid

Petar Kormushev · Barkan Ugurlu · Darwin G. Caldwell · Nikos G. Tsagarakis

Received: date / Accepted: date

Abstract Modern humanoid robots include not only active compliance but also passive compliance. Apart from improved safety and dependability, availability of passive elements, such as springs, opens up new possibilities for improving the energy efficiency.With this in mind, this paper addresses the challenging open problem of exploiting the passive compliance for the purpose of energy efficient humanoid walking. To this end, we develop a method comprising two parts: An optimization part that finds an optimal vertical center-of-mass trajectory, and a walking pattern generator part that uses this trajectory to produce a dynamically-balanced gait. For the optimization part, we propose a reinforcement learning approach that dynamically evolves the policy parametrization during the learning process. By gradually increasing the representational power of the policy parametrization, it manages to find better policies in a faster and computationally efficient way. For the walking generator part, we develop a variable-center-of-mass-height ZMP-based bipedal walking pattern generator. The method is tested in real-world experiments with the bipedal robot COMAN and achieves a significant 18% reduction in the electric energy consumption by learning to efficiently use the passive compliance of the robot.

Keywords bipedal walking · energy efficiency · reinforcement learning · passive compliance

P. Kormushev
Dyson School of Design Engineering,
Imperial College London, SW7 2AZ, U.K.
E-mail: petar@kormushev.com

B. Ugurlu (Corresponding author)
Department of Mechanical Engineering,
Ozyegin University, 34794 Istanbul, Turkey.
E-mail: barkan.ugurlu@ozyegin.edu.tr

D. G. Caldwell and N. G. Tsagarakis
Department of Advanced Robotics,
Istituto Italiano di Tecnologia, 16163 Genoa, Italy.
E-mail: {nikos.tsagarakis, darwin.caldwell}@iit.it

1 Introduction

The current state-of-the-art humanoid robots are equipped with passively compliant elements. In addition to inherent safety and enhanced interaction capabilities, availability of passive elements, e.g., springs, opens up new possibilities for improving the energy efficiency [53]. For instance, the springs can be used for temporary energy storage by compressing them, and for energy reuse by releasing the stored energy [9, 48]. The remaining difficult open problem is how to address the best use of the described mechanism. This paper tackles the problem of finding the optimal way to exploit the passive compliance in a walking robot for the purpose of energy efficiency.

The conventional state-action-based reinforcement learning approaches suffer severely from the curse of dimensionality. To overcome this problem, policy-based reinforcement learning approaches were developed. Instead of working in huge state/action spaces, they use a smaller policy space, which contains all possible policies representable with a certain choice of policy parametrization. Thus, the dimensionality is drastically reduced, and the convergence speed is increased.

In order to find a good solution, i.e., a policy that produces a reward very close to the optimal/desired one, the policy parametrization has to be powerful enough to represent a sufficiently large policy space so that a good candidate solution is present in it. If the policy

parametrization is very simple, with only a few parameters, then the convergence is quick, but often a sub-optimal solution is reached. If the policy parametrization is overly complex, the convergence is slow, and there is a higher possibility that the learning algorithm will converge to some local optimum, possibly much worse than the global optimum. The level of sophistication of the policy parametrization should be just the right amount, in order to provide both fast convergence and a sufficiently optimal solution.

Deciding what policy parametrization to use, and how simple/complex it should be, is a very difficult task, often manually performed via trial-and-error sessions by the researchers. This additional overhead is usually not even mentioned in related literature and falls into the category of "empirically tuned" parameters, together with the reward function, decay factor, exploration noise, weights, and so on. Since changing the policy parametrization requires to restart the learning process from the scratch, this approach is slow and inefficient as all the accumulated data needs to be discarded. As a consequence, the search for new solutions often cannot be done directly on real-world robot systems; rather, simulation studies are performed for proof of concept. To remedy these issues, we propose an approach that allows changing the complexity, i.e., the resolution, of the policy representation dynamically while the reinforcement learning is running.

The rest of the paper is organized as follows: Section 2 provides an overview of the state of the art in multiple research areas which are relevant to the interdisciplinary nature of this paper. In Section 3, the evolving policy parametrization approach is introduced, and a prototype implementation using cubic splines is proposed. Moreover, the proposed approach is evaluated via simulation studies. Section 4 explains details concerning the bipedal walking generation scheme, our bipedal robot, and its passively compliant joints. In Section 5, the real-world experiments conducted on our passively compliant bipedal robot are thoroughly described and analyzed. In Section 6, obtained results are discussed and some inevitable limitations are disclosed. Finally, the paper is concluded in 7 by stating the end results and addressing the future directions.

2 Background

2.1 Related work to policy-based RL algorithms

A tremendous effort has been done by researchers in machine learning and robotics to move RL (Reinforcement Learning) algorithms from discrete to continuous domains, thus extending the possibilities for robotic applications [7, 10, 35, 46]. Until recently, policy gradient algorithms such as Episodic REINFORCE [51] and Episodic Natural Actor-Critic eNAC [36] have been well-established approaches to cope with the high dimensionality. Unfortunately, they also have shortcomings; such as, high sensitivity to the learning rate and the exploratory variance. Trying to overcome this drawback, the following two recent approaches were proposed.

Theodorou *et al.* proposed an RL approach for learning parametrized control policies based on the framework of stochastic optimal control with path integrals [45, 46]. They derived update equations for learning so as to avoid numerical instabilities. This is due to the fact that neither matrix inversions nor gradient learning rates are required. The approach demonstrates significant performance improvements over gradient-based policy learning and scalability to high-dimensional control problems, such as control of a quadruped robot.

Abdolmaleki *et al.* introduced the contextual relative entropy policy search concept that adapts the robot walking controller for different contexts through the use of radial basis functions [1]. The method enabled the controller to learn a policy which adjusts control parameters for a simulated NAO humanoid as it walked forward with a continuous set of walking speeds.

Kober *et al.* developed an episodic RL algorithm called Policy learning by Weighting Exploration with the Returns (PoWER), which is based on Expectation Maximization algorithm [19]. One of its major advantages over policy-gradient-based approaches is that it does not require a learning rate parameter. This is desirable because tuning a learning rate is usually difficult to do for control problems, but critical for achieving good performance of policy-gradient algorithms. PoWER also demonstrates high performance in tasks learned directly on real robots, such as underactuated pendulum swing-up, ball-in-a-cup task, and dynamic pancake flipping task [23].

2.2 Related work to adaptive-resolution RL

Adaptive resolution in state space has been studied in various RL algorithms [3]. Moore and Atkeson employed a decision-tree partitioning of state-space and apply techniques from game-theory and computational geometry to efficiently and adaptively concentrate high resolution on critical areas [30]. They address the pitfalls of discretization during reinforcement learning, concluding that in high dimensionality it is essential for the learning not to plan uniformly over the state space. However, in the context of RL, adaptive resolution in the policy parametrization remains largely unexplored

so far. This paper is making a step exactly in this direction.

2.3 Related work to robot trajectory representations

In order to plan and optimize a trajectory, it first needs to be encoded in a certain way. For instance, cubic splines could be utilized to achieve this task. Similar approaches have been investigated in robotics literature and often called as trajectory generation with *via-points*.

As an example, Miyamoto *et al.* used an actor-critic reinforcement learning scheme with via-point trajectory representation for a simulated cart-pole swing up task [29]. The actor incrementally generates via-points at a coarse time scale, while a trajectory generator transforms via-points to primitive action at the lower level.

Morimoto and Atkeson proposed a walking gait learning approach in which via-points are detected from the observed walking trajectories, and RL modulates the via-points to optimize the walking pattern [31]. The system is applied to a planar biped robot fixed to a boom that constrains the robot motion within the sagittal plane. Exploration tries to minimize the torques while keeping the robot above the desired height to prevent it from tipping over.

Wada and Sumita developed a via-points acquisition algorithm based on actor-critic reinforcement learning, where handwriting patterns are reproduced by an iterative and sequential generation of short movements [50]. The approach finds a set of via-points to mimic a reference trajectory by iterative learning, with the help of evaluation values of the generated movement pattern.

Liu *et al.* proposed a behavior-based locomotion controller. The approach includes feed-forward and feedback mechanisms which correspond to motor patterns and reflexes [26]. An optimization module supports the controller to minimize energy consumption while ensuring stability for a simulated humanoid robot.

Rosado *et al.* used the kinematic data that is collected from human walking via VICON system so as to train a set of dynamic movement primitives [38]. These trained motion primitives then used to control a simulated humanoid robot in task space.

Shafii *et al.* utilized central pattern generators to modulate generated bipedal walking trajectories with varying hip height [41]. Covariance matrix adaptation evolution strategy enabled the robot controller to search for feasible hip height patterns and walking parameters in a way to optimize forward velocity.

Koch *et al.* presented a bipedal gait generation method through the use of movement primitives that are learned from dynamically consistent and optimal trajectories [21]. Morphable movement primitives were learned using Gaussian processes and component analysis. The method allowed the fast real-time movement generation for a simulated HRP-2 robot.

2.4 Related work to energy-efficient motion generation

Passive dynamic walkers are known to be energy-efficient mechanisms since they are able to make use of the swinging limbs momentum while walking forward [27]. The downside is that this type of bipedal walking is not able to handle human interaction or disturbance rejection even if the robot is actuated [52]. Moreover, there are application differences between these types of walkers and 3D fully actuated bipedal robots. In contrast, this paper does not focus on energy optimization from the viewpoint of exploiting the passive walking principle.

A few approaches exist for reducing the energy consumption on fully actuated 3D bipedal walkers [2, 28], but not in the context of learning a varying-CoM-height walking, as presented in this paper. Previously, machine learning approaches have been successfully used for learning tasks on bipedal robots, such as dynamic balancing, quadruped gait optimization [22], and whole-body control during kinesthetic teaching [24]. One especially promising approach for autonomous robot learning is reinforcement learning (RL), as demonstrated in [10, 23, 34, 39, 43].

Stulp *et al.* presented a Policy Improvement with Path Integrals (PI^2) RL approach for variable impedance control, where both planned trajectories and gain schedules for each joint are optimized simultaneously [43]. The approach is used to enable the robot to learn how to push and open a door by minimizing the average stiffness gains controlling the individual joints, with the aim to reduce energy consumption and to increase safety.

Kormushev *et al.* presented the use of Expectation-Maximization-based RL for a pancake flipping task to refine the trajectory of the frying pan and the coordination gain parameters in Cartesian space by using a mixture of proportional-derivative systems with full stiffness matrices [23]. Rosenstein *et al.* presented a simple random search approach to increase the payload capacity of a weightlifting robot by exploiting the robot's intrinsic dynamics at a synergy level [39]. Via-points are learned by exploration in the first phase of learning. RL and simple random search are then used to refine the joint coordination matrices initially defined as identity gains.

RL has been applied previously in the context of bipedal walking optimization, as in [4, 8]. However, the

goal for optimization is usually achieving the fastest possible gait without any regard to the energy consumption. In contrast, this paper focuses on the energy efficiency as the main optimization goal while at the same time maintaining the walking pace and speed unchanged.

Certain studies in biomechanics field indicate different aspects of energy-efficient locomotion. Biological systems, for instance, humans, store and release elastic potential energy into/from muscles and tendons during daily activities such as walking [14]. The management of the elastic potential energy that is stored in these biological structures is essential for reducing the energy consumption and for achieving mechanical power peaks. In this connection, vertical CoM movement appears to be a crucial factor in reducing the metabolic cost [33].

Recent advances in robotics and mechatronics have allowed for the creation of a new generation of passively-compliant bipedal robots, such as COMAN [48]. Similar to biological systems, elastic structures in this robot can store and release energy, which can be extremely helpful if properly used. However, it is difficult to pre-engineer an analytical way to utilize the passive compliance for dynamic walking tasks. One possible application could be the utilization of the passive compliance via machine learning for the energy-efficient bipedal walking generation task. In this paper, we present an approach that minimizes the walking energy by learning a varying-CoM-height walking which efficiently uses the passive compliance of the robot. In doing so, an incisive combination of machine learning and biomechanics could be exploited in a way to enhance an existing technology in bipedal locomotion control.

2.5 Novelty

In this paper, we develop a learning-based integrated for learning to minimize the walking energy required for a passively-compliant bipedal robot. The energy minimization problem is challenging due to the difficulties in accurate modeling considering the properties of the springs, the dynamics of the whole robot and various nonlinearities.

The contributions in this paper can be categorized in two fractions: i) Evolving policy parametrization. ii) The first experimentally demonstrated walking energy minimization for fully actuated 3D bipeds, through the utilization of passive compliance.

First, we introduce a novel reinforcement learning technique which allows the use of changeable-over-time policy parametrization. The proposed learning mechanism can incrementally evolve the policy parametrization as necessary, starting from a very simple parametrization and gradually increasing its complexity (i.e. resolution), and therefore, its representational power. We call this mechanism *evolving policy parametrization* and introduce a practical method to implement it using splines.

Second, we exploit the passive compliance built into our bipedal robot, in order to minimize the energy needed for walking. Using the proposed reinforcement learning algorithm, it is possible to find the optimal vertical CoM trajectory which minimizes the consumed energy. To this end, the authors would like to highlight the fact that this paper reports the first experimental results in which the physical compliance is successfully utilized in walking energy minimization task on a fully actuated and compliant 3D bipedal robot.

An early version of this paper containing preliminary experimental results was presented [25]. The current paper is significantly expanded and improved to provide an archival report, which explains evolving policy parametrization technique and elaborates numerous details about the approach, its implementation and application, newly-added experiment results with thorough analyses and exhaustive discussion on the results.

3 Evolving policy parametrization

We present an RL approach that allows to dynamically change the complexity, i.e., resolution, of the policy representation while the reinforcement learning process is running, without losing any portion of the collected data, and without having to restart the learning propose. We propose a mechanism which can incrementally *evolve* the policy parametrization as necessary, starting from a very simple parametrization and gradually increasing its complexity, and thus, its representational power. The goal is to create an adaptive policy parametrization, which can automatically *grow* to accommodate increasingly more complex policies and get closer to the global optimum. A very desirable side effect of this mechanism is that the tendency of converging to a sub-optimal solution is reduced, because in the lower-dimensional representations this effect is less exhibited, and gradually increasing the complexity of the parametrization helps to avoid getting caught in a poor local optimum.

To achieve this goal, the most important property which a policy encoding should provide is backward compatibility. This means that it should be able to represent subsequent policies such that it is backward-compatible with the previously collected data, such as past rollouts, their corresponding policies, and rewards.

In general, it is possible to consider cases in which simplifying the policy parametrization might be useful, but in this work we assume that we only want to *increase* the complexity of the policy over time, and never to reduce it.

3.1 Spline policy representation

One of the simplest representations which have the property of backward compatibility is the geometric splines. For example, if we have a cubic spline with K knots (or via-points), and then we increase the number of knots, we can still preserve the exact shape of the generated curve (trajectory) by the spline. In fact, if we put one additional knot between every two consecutive knots of the original spline, we end up with a $2K-1$ knots and a spline which coincides with the original spline.

Based on this, we propose to use the spline knots as a policy parametrization and use the spline backward compatibility property for evolving the policy parametrization without losing the previously collected data. In order to do this, we need to define an algorithm for evolving the parametrization from K to L knots ($L > K$), which is formulated in Algorithm 1. Without loss of generality, the values of the policy parameters are normalized in the range [0, 1], and appropriately scaled/shifted as necessary later upon use. Fig. 1 illustrates the process of using spline representation for the evolving policy parametrization. Fig. 2 shows an example for a reinforcement learning process using evolving policy parametrization to approximate an unknown function.

Algorithm 1 EvolvePolicy-Spline ($P_{current}$: current policy, L: desired new number of parameters)

1: $K \leftarrow P_{current}.numberOfParameters$
2: $X_{current} \leftarrow [0, \frac{1}{K-1}, \frac{2}{K-1}, ..., 1]$
3: $Y_{current} \leftarrow P_{current}.parameterValues$
4: $S_{current} \leftarrow \text{ConstructSpline}(X_{current}, Y_{current})$
5: $X_{new} \leftarrow [0, \frac{1}{L-1}, \frac{2}{L-1}, ..., 1]$
6: $Y_{new} \leftarrow \text{EvaluateSplineAtKnots}(S_{current}, X_{new})$
7: $S_{new} \leftarrow \text{ConstructSpline}(X_{new}, Y_{new})$
8: $P_{new}.numberOfParameters \leftarrow L$
9: $P_{new}.parameterValues \leftarrow S_{new}.Y_{new}$
10: return P_{new}

3.2 Integrating the evolving policy parametrization into RL

The proposed technique for evolving the policy parametrization can be used with any policy-based RL algo-

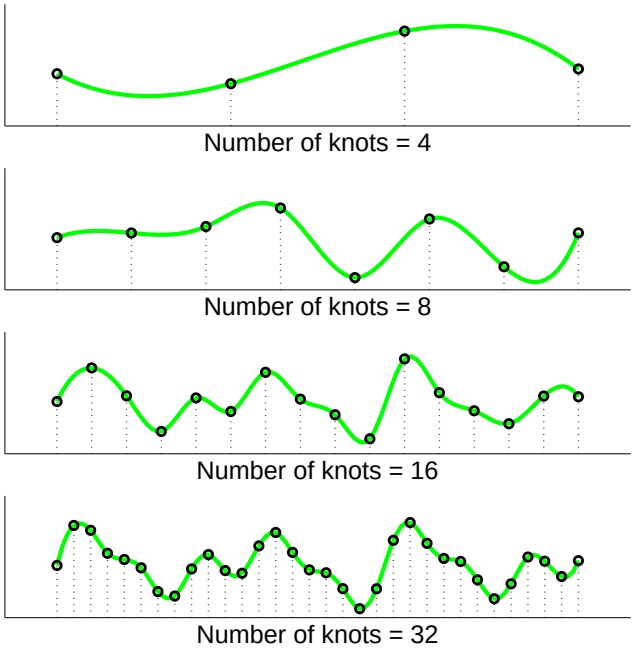


Fig. 1 An example of an evolving policy parametrization based on spline representation of the policy. The set of spline knots is the policy parametrization. The spline values at the knots are the actual policy parameter values. The parametrization starts from 4 knots and evolves up to 32 knots, thus gradually increasing the resolution of the policy.

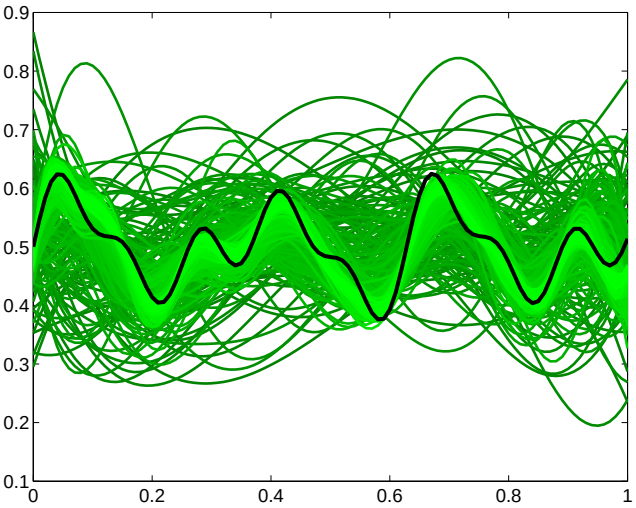


Fig. 2 Reinforcement learning process using evolving policy parametrization. The black trajectory is the unknown global optimum which the reinforcement learning algorithm is trying to approximate. The policy is represented as a trajectory (in green) and is encoded using a spline. The policy evolution is shown by changing the color from dark green for the older policies to bright green for the newer ones. The idea is to gradually evolve the policy, by increasing the number of knots of the spline representation and thus gradually increase the representational power of the policy parametrization. The process is done dynamically while the reinforcement learning algorithm is running.

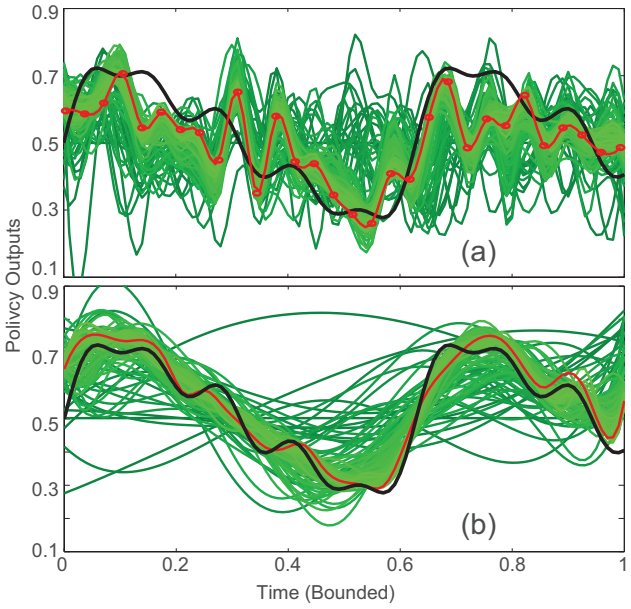


Fig. 3 Simulation Experiment 1: Comparison of the policy outputs from the RL algorithm. a) With fixed policy parametrization (30-knot spline), b) with evolving policy parametrization (from 4-knot to 30-knot spline). In black, the unknown to the algorithm global optimum which it is trying to approximate. In green, all the rollouts performed during the learning process. In red, the current locally-optimal discovered policy by each RL algorithm.

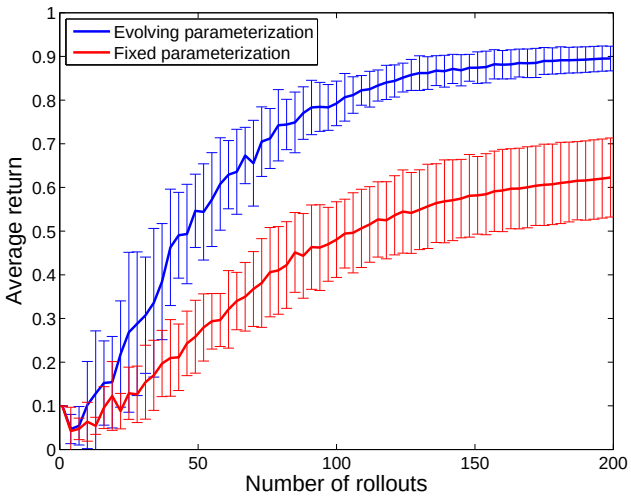


Fig. 4 Simulation Experiment 1: Comparison of the convergence of the RL algorithm with fixed policy parametrization (30-knot spline) versus evolving policy parametrization (from 4-knot to 30-knot spline). The results are averaged over 20 runs of each of the two algorithms in simulation. The standard deviation is indicated with error bars. In addition to faster convergence and higher achieved rewards, the evolving policy parametrization also achieves lower variance compared to the fixed policy parametrization.

rithm. In this paper, we use the state-of-the-art Expectation Maximization-based RL algorithm PoWER [20], due to its fast convergence and a low number of parameters that need tuning. This makes the algorithm appropriate for application directly on the real robot, where it is important to minimize the number of trials, and therefore, the danger of damaging the robot. To further speed up the learning process, we apply the proposed evolving policy parametrization which adaptively changes the resolution of the policy on the fly during the learning process.

In order to minimize the computational time, we evolve the policy parametrization only on those past rollouts which get selected by the importance sampling technique used by the PoWER algorithm. This way, it is not necessary to convert all previous rollouts to the latest policy parametrization, which effectively reduces the computational complexity from $O(N^2)$ to only $O(\sigma N)$, where N is the number of rollouts, and σ is the number of importance sampled rollouts at each RL iteration ($\sigma \ll N$). Usually, σ is a constant number with a value less than 10, which makes the complexity equivalent to $O(N)$, and allows fast execution of the proposed approach for real-time applications.

3.3 Simulation Experiment 1: Function approximation with evolving spline representation

In order to evaluate the proposed reinforcement learning with evolving policy parametrization, we primarily conduct a simulation experiment [1]. The goal is to compare the proposed method with a conventional fixed policy parametrization method that uses the same reinforcement learning algorithm as a baseline. The following synthetic function $\tilde{\tau}$ which is unknown to the learning algorithm is used as the goal for the optimization process.

$$\tilde{\tau}(t) = 0.5 + 0.2\sin(10t) + 0.07\sin(20t) + \\ +0.04\sin(30t) + 0.04\sin(50t), \quad (1)$$

In (1) $\tilde{\tau}$ is with domain $t \in [0, 1]$, and range $\tilde{\tau}(t) \in [0, 1]$. The learning algorithm is trying to approximate $\tilde{\tau}$ by minimizing the difference between it and the policy-generated trajectory.

The reward function used for the simulated experiment is defined as follows:

$$R(\tau) = e^{-\int_0^1 [\tau(t)-\tilde{\tau}(t)]^2 dt}, \quad (2)$$

where $R(\tau)$ is the return of a rollout (trajectory) τ.

[1] https://github.com/petar-kormushev/evolving-policy-parametrization

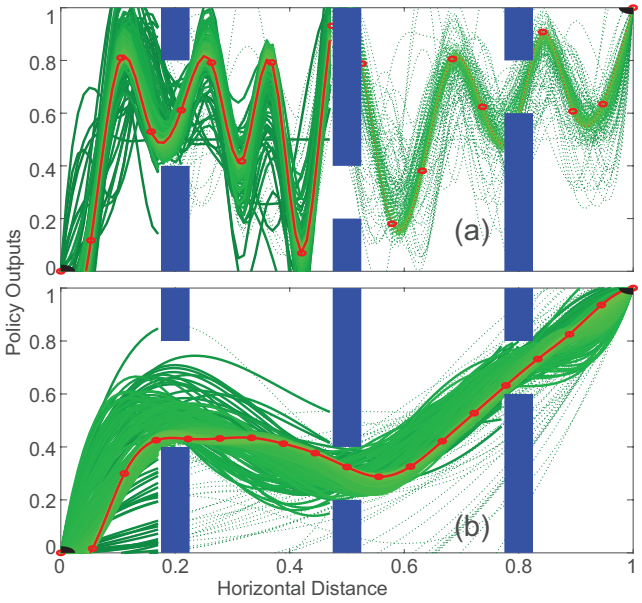


Fig. 5 Simulation Experiment 2: Comparison of the policy outputs from the RL algorithm. a) With fixed policy parametrization (20-knot spline), b) with evolving policy parametrization (from 4-knot to 20-knot spline). Blue bars indicate the obstacles. Green lines represent all the rollouts performed during the learning process. Red lines represent the current locally-optimal discovered policy by each RL algorithm.

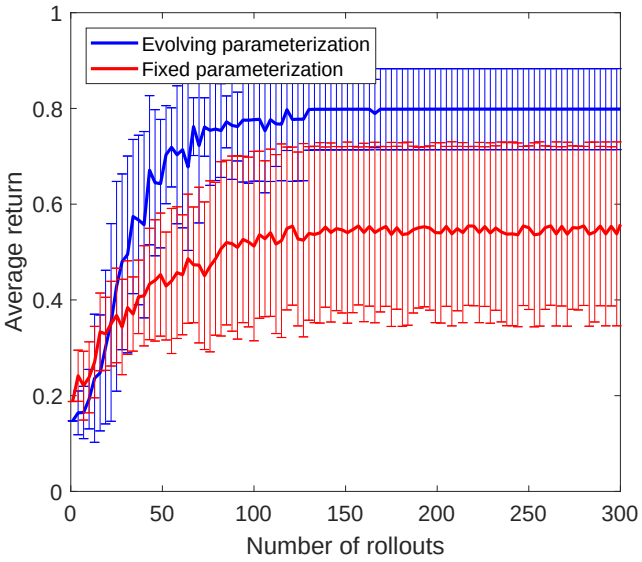


Fig. 6 Simulation Experiment 2: Comparison of the convergence of the RL algorithm with fixed policy parametrization (20-knot spline) versus evolving policy parametrization (from 4-knot to 20-knot spline). The results are averaged over 20 runs of each of the two algorithms in simulation. The standard deviation is indicated with error bars. In addition to faster convergence and higher achieved rewards, the evolving policy parametrization also achieves lower variance compared to the fixed policy parametrization.

Fig. 3 shows a comparison of the generated policy output produced by the proposed evolving policy parametrization method, compared with the output from the conventional fixed policy parametrization method. Due to the lower policy-space dimensionality at the beginning, the evolving policy parametrization approaches much faster the shape of the globally-optimal trajectory. The fixed policy parametrization suffers from inefficient exploration due to the high dimensionality, as well as from overfitting problems, as seen by the high-frequency oscillations of the discovered policies.

Fig. 4 shows that the convergence properties of the proposed method are significantly better than the conventional approach, in terms of faster convergence, higher achieved rewards and lower variance.

3.4 Simulation Experiment 2: Trajectory Planning for Obstacle Avoidance

To further evaluate the proposed RL algorithm with evolving policy parametrization, we have conducted a second, more challenging simulation experiment. In this case, the goal is to perform trajectory planning for obstacle avoidance in 2D space. The simulated environment can be examined in Fig. 5. The starting position is in the bottom-left corner with coordinates (0,0), and the goal position is in the top-right corner with coordinates (1,1). There are 6 obstacles (marked in blue) arranged in a way that creates three narrow openings with progressively smaller sizes in order to produce a challenging motion planning problem. Similarly, the same two methods are being tested and compared: (i) evolving policy parametrization method, and (ii) conventional fixed policy parametrization method. However, this time the reward function does not have the same smoothness properties as in the previous simulation experiments presented in subsection 3.3. This is due to the fact that whenever a trajectory collides with an obstacle, it is terminated at that instant. Therefore, this introduces discontinuities in the reward landscape and is more challenging for the learning algorithm in both cases. Furthermore, the reward function is defined based on the distance from the last reached position before arriving at the goal position. This, again, is more challenging as it introduces multiple local optima in the reward landscape which tends to trap the learning algorithms and makes it harder to reach the global optimum.

Despite these challenges, we show that the proposed evolving policy parametrization method consistently outperforms the conventional fixed policy parametrization method. Fig. 5 displays a comparison of the generated policy output produced by the proposed evolv-

Table 1 Sequence of walking phases

No.	Phase description	Start time[s]	Duration[s]
1	Wait 1	0.00	1.00
2	Initialization	1.00	1.00
3	Wait 2	2.00	5.00
4	Transfer (double)	7.00	0.60
5	Right single	7.60	0.50
6	Double	8.10	0.15
7	Left single	8.25	0.50
8	Double	8.75	0.15
9	Right single	8.90	0.50
10	Double	9.40	0.15

ing policy parametrization method, compared with the output from the conventional fixed policy parametrization method. Due to the lower policy-space dimensionality at the beginning, the evolving policy parametrization is able to more quickly explore the 2D space and is able to navigate around the 6 obstacles in a much smoother way. For comparison, the fixed policy parametrization struggles to go through the second and third opening because of the difficulty to explore the high dimensional policy space. Moreover, it suffers from overfitting problems which produce undesired jitter in the produced trajectories. Finally, the convergence properties of the two methods are compared in Fig. 6 which again confirms that the proposed method performs significantly better than the conventional approach, in terms of faster convergence, higher achieved rewards, and better quality solutions. This makes the proposed method particularly useful for real-world trajectory-planning scenarios, as shown in the following sections on a bipedal walking robot.

4 Bipedal walking energy reduction

For a real-world evaluation of the proposed approach, we tackle the problem of bipedal walking energy minimization. The proposed RL method is used to learn a vertical trajectory for the CoM of the robot such that the potential elastic energy exchange is fully utilized during walking, in order to minimize the energy consumption. A high-level outline of the real-world experiment is shown in Fig. 7.

For the reinforcement learning component, an important difference from the simulated experiments is that here the RL policy (i.e. the vertical CoM trajectory) needs to be cyclic in time. This is necessary because walking motion must be executed periodically over many cycles. A single walking cycle includes a single support phase in which either the left foot or right foot is in swing mode. This phase is followed by a double support phase where both feet are in the stance mode. Subsequently, single support phases are swapped between left and right feet to generate continuous walking motion. In particular, continuous walking was important in our case for the purpose of assessing energy consumption. Duration values for the walking phases, as well as initialization periods, are provided in Table 1.

Table 2 Basic specifications of the robot.

Size	Upper Leg length:	226.63 [mm]
	Lower Leg length:	203.3 [mm]
	Ankle-sole length:	60.3 [mm]
Weight	Each Leg:	6.816 [kg]
	Waist:	4.41 [kg]
	Total:	17.772 [kg]

Fig. 9 illustrates the process of creating a time-cyclic policy out of a single spline in which a single cycle time was contained in the interval [0, 1]. The red line represents the input policy in the form of a spline for a one cycle interval; the values at spline knots were obtained from the policy parametrization values. The green line represents the time-cyclic policy whose spline knots were copied from the policy values at one cycle interval. Since the time-cyclic policy represents the vertical CoM trajectory in bipedal walking, it guarantees that both position and velocity are continuous and in differentiable form.

4.1 Compliant Bipedal Robot COMAN

In order to explore compliant humanoid characteristics, we developed a bipedal robot at the Italian Institute of Technology, as a part of the European AMARSi project. Table 2 summarizes its mechanical specifications. The robot has a total of 15 active DoFs (Degree of Freedom); 6 DoFs in each leg and 3 DoFs at the waist to be able to obtain greater motion flexibility. Each active joint incorporates three position sensors (two absolute and one relative encoders) and one torque sensor. The robot is also equipped with two 6-axis Force/Torque sensors at the ankles and five single-axis load cells on the foot sole. In addition, it has a triaxial rate gyro sensor and an accelerometer, located at the pelvis. In its electronic hardware structure, the main controller is an Intel Core 2 Duo 1.5 GHz dual processor with 3.0 GB RAM, running on a 32-bit GNU/Linux operating system that includesa real-time Xenomai extension. Data communication is performed via a real-time Ethernet protocol called RTnet. Fig. 8 displays the actual robot and its joint configuration.

In the first prototype, only pitch axis ankle and knee joints are equipped with passive compliant elements,

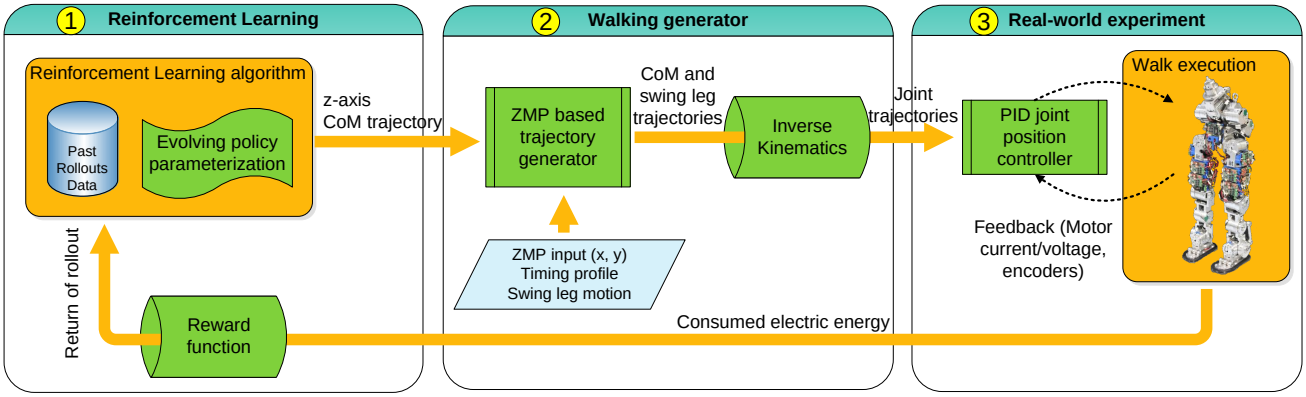


Fig. 7 Outline of the proposed approach for bipedal walking energy consumption minimization, showing details about each of the three components: reinforcement learning, walking generation, and real-world rollout execution. Note that all the components are run in real-time.

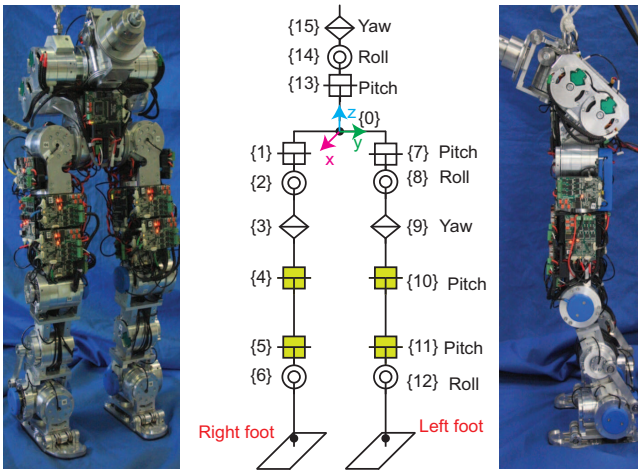


Fig. 8 The mechanical assembly of COMAN and its joint configuration. Joints with yellow color indicate SEA units.

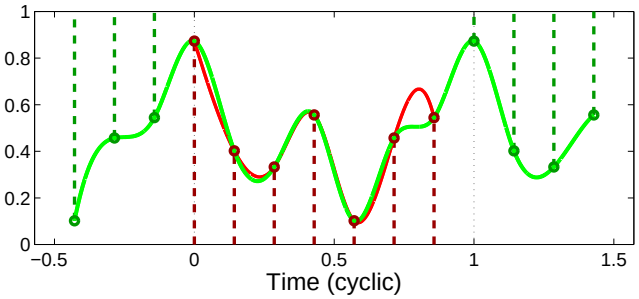


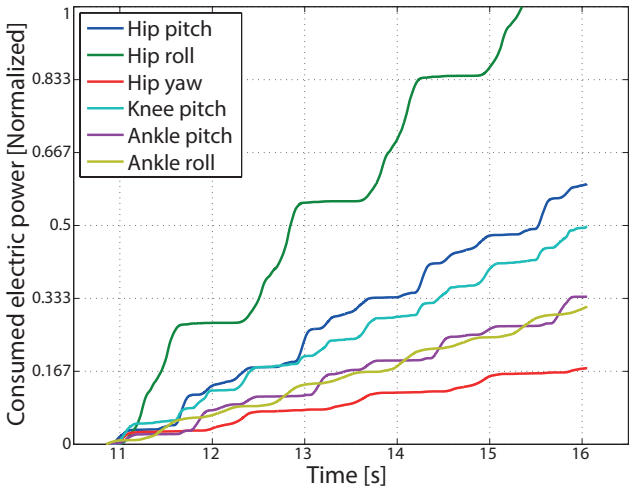
Fig. 9 Illustration of the process for creating a time-cyclic policy out of a single spline. One time cycle is contained in the interval [0, 1]. In red, the input policy in the form of a spline (red line), where the values at the spline knots (red circles) come from the policy parametrization values. In green, the produced time-cyclic policy, where the green knots have values copied from the policy values at one cycle interval. The green spline is the output time-cyclic policy, which guarantees that both position and velocity of the CoM is a continuous and differentiable function.

i.e., springs (see Fig. 8, frames with yellow color). For the compliant actuation system in our bipedal robot, the main objectives are to satisfy dimensional and weight requirements while achieving high rotary stiffness within a compact structure. Regarding these requirements as well as with the previously discussed issues, a series elastic actuator (SEA) module appears to be a very suitable candidate and it is presently implemented in our robot [49]. The rotary stiffness of these modules was set to an approximate value of 156 [Nm/rad] to maximize the walking efficiency while providing sufficient bandwidth for joint position tracking.

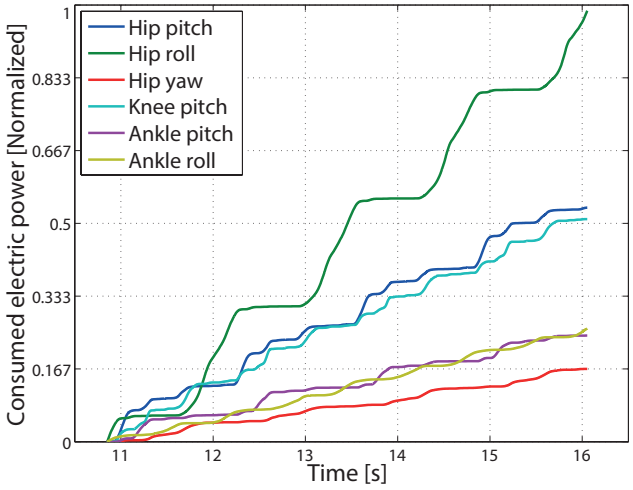
4.2 Evaluation of walking energy consumption

There are many ways in which energy can be measured. One possible approach is to estimate the mechanical energy from motor torque measurements and angular velocities. However, the problem with this approach is that it incorrectly includes the work done by gravity, and can only infer indirectly the actual electric power used for walking. Furthermore, electrical energy is definitely used by the motors even when the mechanical energy is zero, e.g., when the robot is only standing.

We propose, what we think is the best approach, to directly measure the electrical energy used by all the motors of the robot, which allows us to explicitly measure the value that we are trying to minimize. We use the formula $P = IU$, linking the electric power P to the electric current I and the voltage U, and we integrate over time to calculate the consumed electric energy in Joules. The COMAN robot is equipped with both current and voltage sensing units at each motor so that we can accurately measure these values. Fig. 10 shows the accumulated consumed electric energy values for the



(a) Left leg



(b) Right leg

Fig. 10 Electric energy consumption of each leg of COMAN during a 4-cycle walk. Alternating between left and right foot support redistributes the weight on different joint motors and causes differences in the left-right energy consumption. Hip roll joints consume the highest energy due to the fact that they solely support the whole leg weight in the lateral plane while it is in swinging mode.

motor of each individual joint of COMAN, calculated as:

$$E_j(t_1, t_2) = \int_{t_1}^{t_2} I_j(t) U_j(t)\, dt, \tag{3}$$

where j is a selected joint for which the energy consumption is calculated, and $[t_1, t_2]$ is the time interval.

To evaluate the whole walking rollout, we define the energy consumption metric of a given rollout τ to be the average electric energy consumed per walking cycle, and estimate it using the formula.

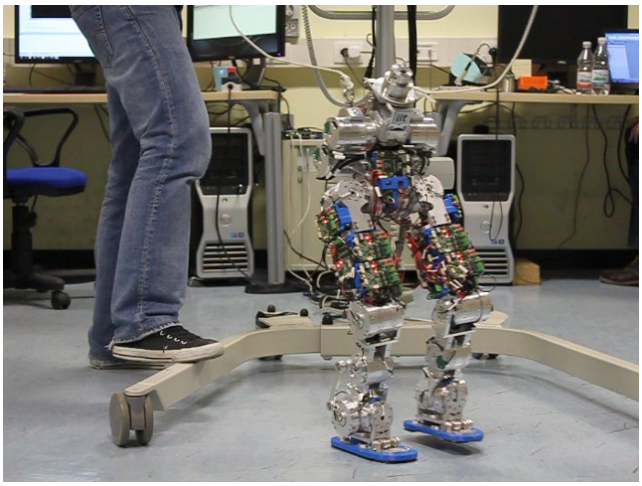


Fig. 11 The experimental setup, showing a snapshot of the bipedal robot COMAN during one walking rollout execution.

$$E(\tau) = \frac{1}{c} \sum_{j \in J} E_j(t_1, t_2), \tag{4}$$

where J is the set of joints in the sagittal plane (hip, knee, and ankle pitch of both legs, 6 in total) whose energy consumption we try to minimize.

In order to reduce the noise effects on the measurement, we make the robot walk for 16 seconds and collect the electric current and voltage measurements of the $c = 4$ consecutive walk cycles (4 repetitions of phases 7 to 10 in Table 1), which contain a total of 8 steps. Therefore, the value of t_1 is the start of phase 7, and the time t_2 is the end of phase 10 in the fourth cycle. Afterward, we average the energy consumption and use this value as the estimate of the electric energy used for this walking rollout.

In this work, the main focus is the exploitation of passive compliance for energy efficiency which may be achieved with the help of springs. Therefore, we use the sum of all electric energy consumed by the motors controlling the motion in the sagittal plane, i.e. the hip, knee, and ankle pitch joints on both legs, in our evaluation metric. Even though hip pitch joints do not include series elasticity, they sufficiently contribute to the vertical CoM trajectory as they are dominant in the sagittal plane together with ankle pitch and knee joints; therefore, they were included in the metric.

Finally, we define the return of a rollout τ as:

$$R(\tau) = e^{-kE(\tau)}, \tag{5}$$

where k is a scaling constant. The lower the energy consumed, the higher the reward is.

5 Real-world experiments

Based on the results of the simulation experiment, the proposed evolving policy parametrization method is chosen for the real-world walking experiment, due to its favorable characteristics for real-time applications. The experimental setup is shown in Fig. 11. The total distance traveled by the robot during our experiments is around 0.5 km. For the evaluation of the energy consumption, we did not include the traveled distance, as the speed of walking was the same for all rollouts because the stride length was fixed.

Fig. 12 shows the convergence results from the walking experiments. The figure shows the convergence of the consumed energy over time during the reinforcement learning. Energy measurements are normalized with the maximum possible energy consumption in mind. Each rollout corresponds to a walking experiment that was executed. For each rollout, the average energy consumed per cycle (averaged over 8 walking steps, i.e. 4 full walk cycles) is shown. At rollout number 126 the lowest energy consumption was achieved, which is 18% lower than the initial energy consumption.

Fig. 13 visualizes the discovered optimal policy by the RL algorithm, as well as all the intermediate 180 rollouts that were performed. Although the single and double support phase periods were determined in advance, the RL algorithm discovered the instant at which the heel strikes the ground (shown with dotted vertical line), and adjusted the trajectory so that the CoM height is bounced off upward in that exact same moment. Note that the CoM height trajectory is normalized by considering the maximum and minimum allowable values which are imposed by the kinematic structure of the robot. All trajectories have been made cyclic in time so that walking can be executed continuously over many cycles.

ZMP response measurements with respect to the inertial frame are displayed in Fig. 15, for 7 consecutive steps. During single support phases, the support polygon is the supporting foot area which is illustrated with rectangles. When the robot is in a double support phase, the area between two feet becomes the support polygon. We illustrated this support polygon only once with a dashed cyan area, in which the robot switches from the first step to the second step. What is more, green areas stand for transition phases in which the robot motion is initiated from a stationary position or vice versa. Steps with odd numbers indicate the right leg's single support phases whereas even numbers stand for the left leg's single support phases. Based on this result, it is possible to examine that the ZMP response is always within the support polygon boundaries. As a result, we obtained dynamically equilibrated and feasible walking cycles throughout the experimentation period. That being the case, we were able to focus solely on the energy minimization problem, without worrying about auxiliary issues, such as the dynamic balance.

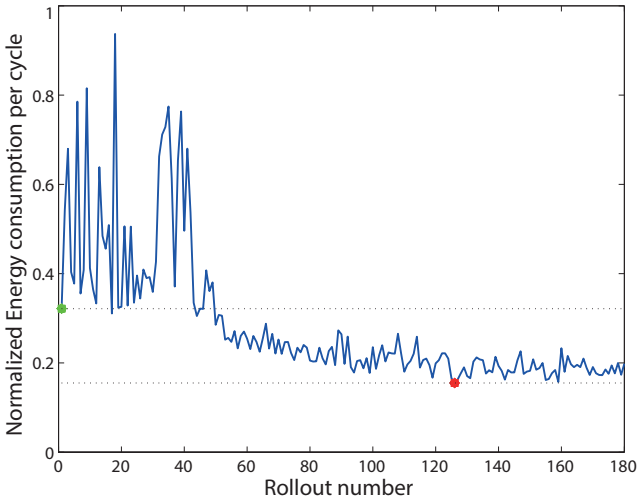


Fig. 12 Results from the real-world minimization of the consumed energy for walking.

As previously stated, each compliant joint includes separate encoders to measure both link side (after the spring) and motor side angles. This feature enables us to record spring deflection variations, in a reliable way. To this end, right leg's knee and ankle joint deflections are respectively illustrated in Fig. 16 and Fig. 17. In these figures, solid purple lines show the deflection variations while varying CoM height is generated by the RL algorithm after it learns to use the passive compliance efficiently to minimize the energy. Solid green lines indicate deflection values when CoM height is fixed. Analogous trends are observed for the left leg as well and therefore not plotted.

6 Discussion of results

6.1 Discussions on Evolving Policy parametrization

A major advantage demonstrated by the proposed approach is the low variance of the generated policies. The lower exploratory variance combined with the faster convergence is the key factor for achieving higher rewards than the fixed parametrization.

With respect to the learning, the focus of the paper is not on the encoding scheme (splines), but on the evolving policy parametrization. Spline-based techniques have well-known limitations such as providing a non-autonomous (time-based) control policy, discarding variability and synergy information in the repre-

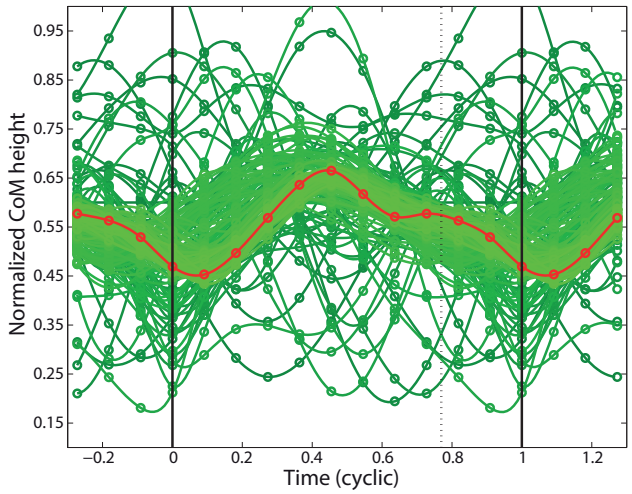


Fig. 13 The discovered optimal policy (in red) by the reinforcement learning. Among all tried 180 CoM trajectories (in green), which were executed on the real robot.

sentation, and having difficulty to cope with unforeseen perturbations [37, 40]. Being aware of their limitations, splines provided us with a simple encoding scheme to be used as a first step to study the possibility of dynamic evolution of the policy parametrization during the learning. In addition, splines provided us with a straightforward way to implement a cyclic policy which spans continuously over many time cycles and is convenient for robot walking applications.

In this study, the knots were increased along the way in a heuristic manner for the sake of simplicity. By observing the convergence rates, it is possible to devise a systematic method for the addition of knots along the iteration. This additional feature deserves further investigation and is addressed as a future work.

6.2 Discussions on Bipedal Walking and Energy Minimization

The passive compliance of our robot was recently exploited to generate periodic jumping patterns [48]. In this study, the base resonance frequency of the overall system is identified to be within $0.925 \sim 1.04$ [Hz] frequency band. Even though the stiffness in robot joint is constant, the base resonance frequency depends on the configuration, so that it varies within a frequency band. When the robot is vertically excited within a close proximity to the base resonance frequency, joint deflections are expected to be maximized. This enables us to maximize elastic potential energy stored in the springs. That being the case, it is possible to obtain walking cycles with lower energy demands.

Throughout the learning process, the RL algorithm produced vertical motions with relatively higher frequencies as seen in Fig. 13, resulting in bad score in energy minimization goal. Finally, the vertical CoM movement which was eventually learned by the RL algorithm produced cyclic motions with a frequency within the $0.925 \sim 1.04$ [Hz] band; approximately 1.0 [Hz]. This can be justified in Fig. 16 and Fig. 17. Spring deflections are measured to be about -11 degrees, which is 97% of the maximum allowable[2] values. Therefore, the algorithm achieved bipedal walking energy minimization goal, as it successfully found the optimal vertical CoM trajectory.

The end result of the energy minimization is computed to be 18%, which may be regarded as a crucial value when considering real-time operation duration of bipedal robots. The authors would like to highlight the fact that this paper reports the first experimental results of a bipedal walking energy minimization task, achieved on a fully actuated 3D robot with spring-supported passively compliant joints. Furthermore, it allows us to operate COMAN in real-time for approximately 4.3 more hours while using Li-Ion on-board batteries. Due to this fact, the robot demands less for battery recharge and become more environment-friendly by effectively using the limited power source.

We would like to highlight the fact that the dynamic balance is guaranteed by the ZMP-based motion generator as it outputs dynamically balanced and consistent walking trajectories for a given set of feasibly designated ZMP inputs, regardless of the CoM height variance. In other words, the vertical CoM trajectory is given by the RL algorithm beforehand and utilized in the ZMP-based walking generator to induce dynamically balanced horizontal CoM trajectories. Therefore, we may focus on the bipedal walking energy minimization task without having any concern related to the dynamic balance issue.

In the current configuration, spring deflections are already maximized as illustrated in Fig. 16 and Fig. 17, thanks to the RL algorithm. Therefore, 18% energy minimization appears to be the direct consequence of maximizing spring deflections. The amount of energy minimization may be further improved if springs in the joint are replaced with their softer counterparts. In doing so, elastic energy storage can be increased, however, the robot may suffer undesired vertical oscillations. Therefore, there is a trade-off between the energy minimization and dynamic balance, in choosing the spring stiffness profile. We handled this prob-

[2] Spring deflections are mechanically limited within 11.25 degrees in COMAN.

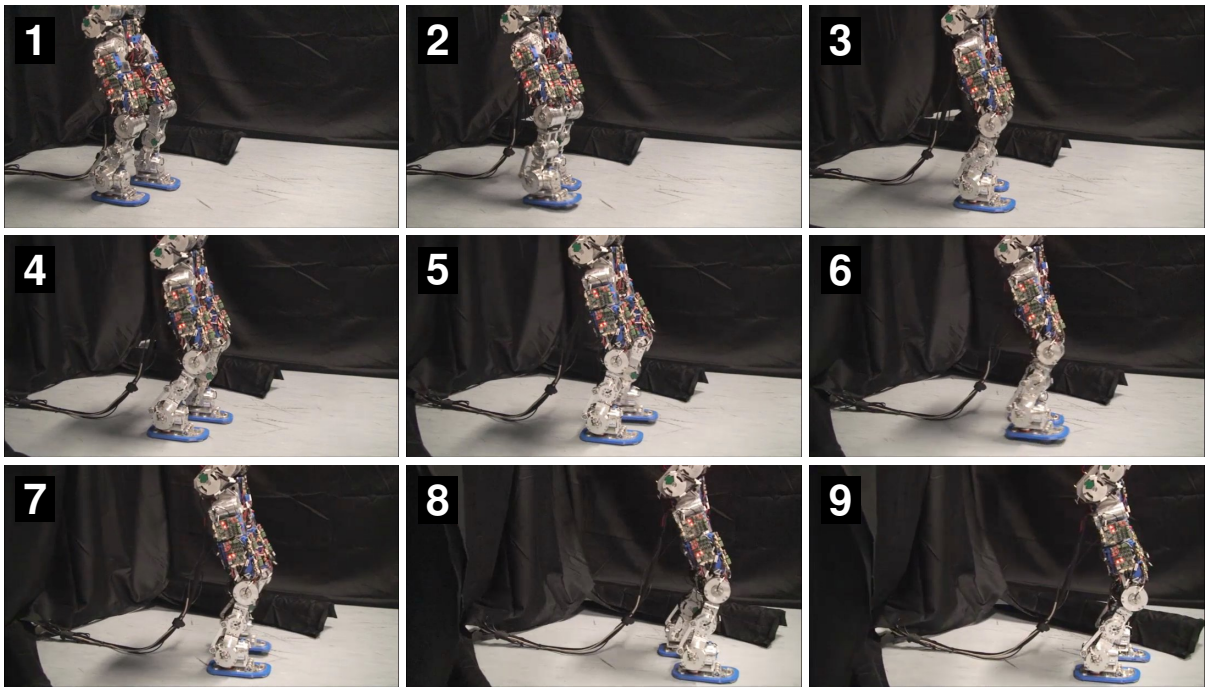


Fig. 14 A sequence of video snapshots from the real-world experiment with the lower body of the COMAN robot.

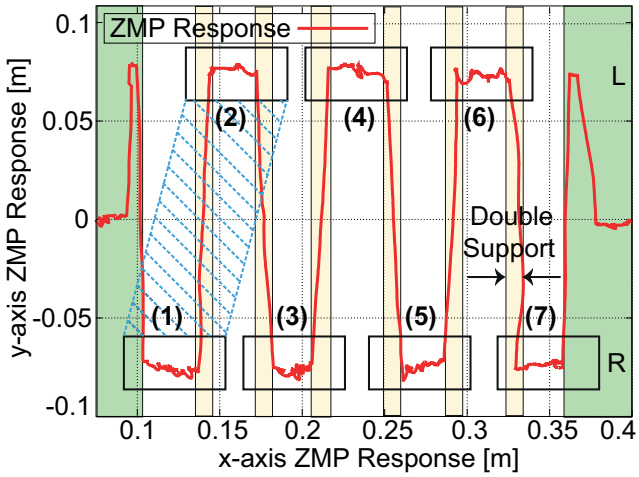


Fig. 15 Actual ZMP measurements with respect to the inertial frame. In this experiment, the robot walked 7 steps ahead. Foot positions are also indicated with rectangles.

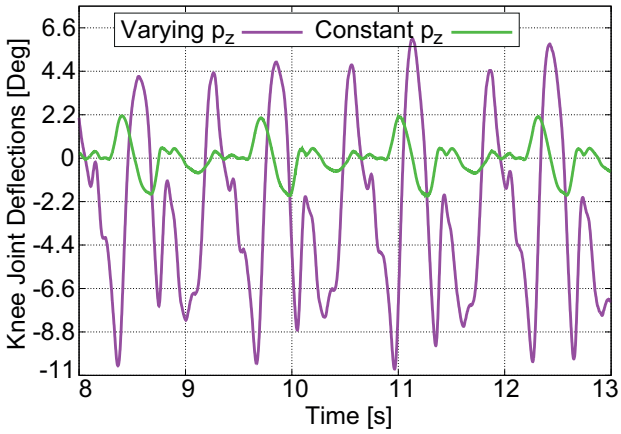


Fig. 16 Right knee springs deflection during walking, measured in angular offset. Similar deflection variations were also observed in the left knee joint.

lem throughout the design process by empirically trying various springs with different stiffness coefficients.

Variable stiffness actuators may remedy the stiffness adjustment problem of SEAs through the active regulation of the passive compliance in real-time [16]. Variable stiffness regulation plays an important role in human walking; humans actively change the joint stiffness to explore optimal walking patterns [9, 13]. That being said, these actuators are still large-sized and may not be applicable to power humanoids in their current form. Therefore, learning variable stiffness for the legged robot control will be investigated once the necessary hardware improvements are introduced.

Due to hardware limitations, the current version of COMAN had passive compliance only in pitch axis knee and ankle joints. Therefore, the overall energy minimization is provided solely by 4 joints, whereas the rest of the 8 joints (roll axis joints, yaw axis joints, pitch axis hip joints) could not contribute to this task due to the lack of passive compliance. Currently, our design engineering team is working on the second generation COMAN bipedal robot which will have passive compliance utilized in all the joints. In principle, the pro-

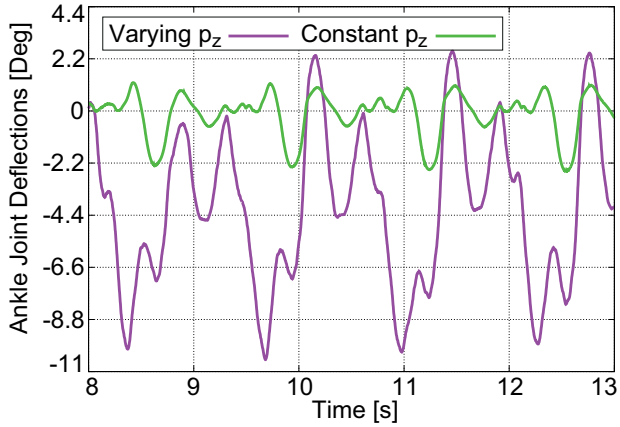


Fig. 17 Right ankle springs deflection during walking, measured in angular offset. Similar deflection variations were also observed in the left ankle joint.

posed method may perform even better when conducting walking motion on a robot with passive compliance in all joints.

In this work, the idea of generating efficient walking pattern through the use of potential energy management has its roots from studies in biomechanics [9,15]. Therefore, we used an abstracted model for the humanoids so as to fully focus our attention to exploit passive compliance for energy efficiency. At this stage, While useful in its own right, abstracted models may have limitations in describing the complete robot behavior. With the advent of centroidal dynamics [32], efficient locomotion controllers were proposed [5,12]. An extension of centroidal dynamics for robots with passive compliance may be investigated as a future work to further improve the performance.

Energy minimization may also be achieved by altering bipedal walking generator parameters. Since the main focus of this paper was the exploitation of passive compliance, the additional investigation of energy-minimization via learning the optimized bipedal walking parameters will be a future work. That being said, our research group previously implemented the proposed learning algorithm for the efficient quadruped gaits. For details refer to [42].

7 Concluding Remarks

We proposed a reinforcement learning approach that can evolve the policy parametrization dynamically during the learning process. We showed that the gradually increasing representational power of the policy parametrization helps to find better policies faster than a fixed parametrization. We successfully applied it to a bipedal walking energy minimization task by utilizing a variable-CoM-height ZMP-based bipedal walking generator. The method achieved 18% reduction in energy consumption by learning to use efficiently the passive compliance of the robot, which is the first reported experimental walking energy minimization results in the state-of-the-art humanoid robotics.

As a future work, we plan to extend this work for more powerful movement representations, based on a superposition of basis motion fields [23]. Another interesting direction for extension is towards learning of variable stiffness control, which is of particular interest in the context of energy minimization.

Appendix: Bipedal Walking Gait Generator

Given the z-axis CoM trajectory, we utilized the ZMP concept for x-axis and y-axis CoM trajectories, in order to obtain walking patterns with dynamic balance. To generate real-time bipedal walking patterns which use the vertical CoM trajectory generated by the RL component, we adopted the resolution method explained in [17], using Thomas Algorithm [47]. Considering the one mass model, CoM position and ZMP position are described as $P = (p_x, p_y, p_z)$ and $Q = (q_x, q_y, 0)$, respectively. As described in [6,11,18,44], the abstracted x-axis ZMP equation takes the following form,

$$q_x = p_x - \frac{\ddot{p}_x}{\ddot{p}_z + g} p_z, \tag{6}$$

where g is the gravitational acceleration. The vertical CoM position (p_z) and acceleration ($\ddot{p}_z$) are provided by the learning algorithm for all times as previously stated. As next step, (6) is discretized for p_x as follows:

$$\ddot{p}_x(t) = \frac{p_x(i+1) - 2p_x(i) + p_x(i-1)}{\Delta t^2}, \tag{7}$$

where Δt is the sampling period, i is the discrete event. i starts from 0 to n which is the total number of discrete events. Inserting (7) into (6), we obtain the following:

$$p_x(i+1) = \frac{b(i)}{c(i)} p_x(i) - p_x(i-1) + \frac{q_x(i)}{c(i)}; \tag{8}$$

$$b(i) = 1 - 2c(i); \qquad c(i) = \frac{-p_z(i)}{(\ddot{p}_z(i) + g)\Delta t^2}. \tag{9}$$

In order to solve this tridiagonal equation efficiently, we employ Thomas Algorithm [47]. To do so, initial and final position of x-axis CoM ($p_x(0)$ and $p_x(n)$) must be given in advance. Therefore, for a given set of reference ZMP trajectory, initial conditions, and final conditions, we are able to calculate CoM trajectory. For that purpose, the tridiagonal equation is re-arranged as below.

$$p_x(i) = e(i+1)p_x(i+1) + f(i+1). \quad (10)$$

In (10), $e(i+1)$ and $f(i+1)$ can be defined as follows:

$$e(i+1) = -\frac{c(i)}{c(i)e(i) + b(i)}, \quad (11)$$

$$f(i+1) = \frac{q_x(i) - c(i)f(i)}{c(i)e(i) + b(i)}. \quad (12)$$

Combining (10), (11) and (12), (13) is yielded.

$$p_x(i) = -\frac{c(i)}{c(i)e(i) + b(i)}p_x(i+1) + \frac{q_x(i) - c(i)f(i)}{c(i)e(i) + b(i)} \quad (13)$$

Recall that $p_x(0) = x_0$ and $p_x(n) = x_n$, $e(1)$ and $f(1)$ are determined as 0 and x_0, respectively. Utilizing Thomas Algorithm for the solution of this tridiagonal equation, we can obtain the CoM trajectory's x-axis component. If an identical approach is also executed for y-axis CoM position, we could derive all the components of the CoM trajectory in real-time since vertical CoM position is previously determined by the RL algorithm.

ACKNOWLEDGEMENTS

This work was partially supported by the EU project AMARSi, under the contract FP7-ICT-248311.

References

1. A. Abdolmaleki, N. Lau, L. P. Reis, J. Peters, and G. Neumann. Contextual policy search for linear and nonlinear generalization of a humanoid walking controller. *Journal of Intelligent & Robotic Systems*, 83(3):393–408, 2016.
2. C. A. Amran, B. Ugurlu, and A. Kawamura. Energy and torque efficient zmp-based bipedal walking with varying center of mass height. In *IEEE Intl. Workshop on Advanced Motion Control*, pages 408–413, 2010.
3. A. Bernstein and N. Shimkin. Adaptive-resolution reinforcement learning with polynomial exploration in deterministic domains. *Machine Learning*, 81(3):359–397, 2010.
4. Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. An experimental comparison of bayesian optimization for bipedal locomotion. In *Proceedings of 2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
5. J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard. A versatile and efficient pattern generator for generalized legged locomotion. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 1–6, Stockholm, Sweden, May 2016.
6. Y. Choi, D. Kim, Y. Oh, and B. You. Posture/walking control for humanoid robot based on resolution of com jacobian with embedded motion. *IEEE Transactions on Robotics*, 23(6):1285–1293, 2007.
7. A. Coates, P. Abbeel, and A. Y. Ng. Apprenticeship learning for helicopter control. *Commun. ACM*, 52(7):97–105, 2009.
8. M. P. Deisenroth, R. Calandra, A. Seyfarth, and J. Peters. Toward fast policy search for learning legged locomotion. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1787–1792. IEEE, 2012.
9. H. Geyer, A. Seyfarth, and R. Blickhan. Compliant leg behaviour explains basic dynamics of walking and running. *Proceedings of the Royal Society B: Biological Sciences*, 273(1603):2861–2867, 2006.
10. F. Guenter, M. Hersch, S. Calinon, and A. Billard. Reinforcement learning for imitating constrained reaching movements. *Advanced Robotics*, 21(13):1521–1544, 2007.
11. K. Harada, S. Kajita, K. Kaneko, and H. Hirukawa. An analytical method on real-time gait planning for a humanoid robot. *Intl. Journal of Humanoid Robotics*, 3(1):1–19, 2004.
12. A. Herzog, S. Schaal, and L. Righetti. Structured contact force optimization for kino-dynamic motion generation. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 1–6, Daejeon, Korea, October 2016.
13. Y. Hu, M. Felis, and K. Mombaur. Compliance analysis of human leg joints in level ground walking with an optimal control approach. In *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, pages 881–886, Madrid, Spain, November 2014.
14. M. Ishikawa, P. V. Komi, M. J. Grey, V. Lepola, and P.G. Bruggemann. Muscle-tendon interaction and elastic energy usage in human walking. *Journal of Appl. Physiology,*, 99(2):603–608, 2005.
15. M. Ishikawa, V. Komi, M. J. Grey, V. Lepola, and P. Bruggemann. Muscle-tendon interaction and elastic energy usage in human walking. *Journal of Appl. Physiol.*, 99(2):603–608, 2005.
16. A. Jafari, N. G. Tsagarakis, and D. G. Caldwell. A novel intrinsically energy efficient actuator with adjustable stiffness (AwAS). *Mechatronics, IEEE/ASME Transactions on*, 18(1):355–365, 2013.
17. S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, T. Inaba, and H. Inoue. A fast dynamically equilibrated walking trajectory generation method of humanoid robot. *Autonomous Robots*, 2(1):71–82, 2002.
18. S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 1620–1626., Taipei, Taiwan, May 2003.
19. J. Kober and J. Peters. Learning motor primitives for robotics. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 2112–2118, May 2009.
20. J. Kober and J. Peters. Policy search for motor primitives in robotics. *Machine Learning*, 84(1-2):171–203, 2011.
21. K. H. Koch, D. Clever, K. Mombaur, and D. Endres. Learning movement primitives from optimal and dynamically feasible trajectories for humanoid walking. In *Proc. IEEE-Ras Intl Conf. on Humanoid Robots (Humanoids)*, pages 866–873, Seoul, Korea, November 2015.
22. N. Kohl and P. Stone. Machine learning for fast quadrupedal locomotion. In *Proc. National Conference on Artificial Intelligence*, pages 611–616. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.
23. P. Kormushev, S. Calinon, and D. G. Caldwell. Robot motor skill coordination with EM-based reinforcement

learning. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 3232–3237, Taipei, Taiwan, October 2010.
24. P. Kormushev, D.N. Nenchev, S. Calinon, and D.G. Caldwell. Upper-body kinesthetic teaching of a free-standing humanoid robot. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, Shanghai, China, 2011.
25. P. Kormushev, B. Ugurlu, S. Calinon, N. G. Tsagarakis, and D. G. Caldwell. Bipedal walking energy minimization by reinforcement learning with evolving policy parameterization. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 318–324, San Francisco, USA, September 2011.
26. Q. Liu, J. Zhao, S. Schutz, and K. Berns. Adaptive motor patterns and reflexes for bipedal locomotion on rough terrain. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 3856–3861, Hamburg, Germany, September 2015.
27. T. McGeer. Passive dynamic walking. *Intl. Journal of Robotics Research*, 9(2):62–82, 1990.
28. H. Minekata, H. Seki, and S. Tadakuma. A study of energy-saving shoes for robot considering lateral plane motion. *IEEE Transactions on Industrial Electronics*, 55(3):1271–1276, 2008.
29. H. Miyamoto, J. Morimoto, K. Doya, and M. Kawato. Reinforcement learning with via-point representation. *Neural Networks*, 17:299–305, April 2004.
30. A. W. Moore and C. G. Atkeson. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. *Machine Learning*, 21:199–233, December 1995.
31. J. Morimoto and C. G. Atkeson. Learning biped locomotion: Application of poincare-map-based reinforcement learning. *IEEE Robotics and Automation Magazine*, 14(2):41–51, 2007.
32. D. E. Orin, A. Goswami, and S.-H. Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35(2):161–176, 2013.
33. J. D. Ortega and C. T. Farley. Minimizing center of mass vertical movement increases metabolic cost in walking. *Journal of Appl. Physiol.*, 581(9):2099–2107, 2005.
34. Pastor P., Kalakrishnan M., Chitta S., Theodorou R., and Schaal S. Skill learning and task outcome prediction for manipulation. In *Intl Conf. on Robotics and Automation (ICRA)*, Shanghai, China, 2011.
35. J. Peters and S Schaal. Policy gradient methods for robotics. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2006.
36. J. Peters and S. Schaal. Natural actor-critic. *Neurocomput.*, 71(7-9):1180–1190, 2008.
37. J. Peters and S. Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, 2008.
38. J. Rosado, F. Silva, and V. Santos. Biped walking learning from imitation using dynamic movement primitives. In L. P. Reis, A. P. Moreira, P. U. Lima, L. Montano, and V. Munoz Martinez, editors, *Advances in Intelligent Systems and Computing*, pages 185–196. Springer International Publishing, Switzerland, 2015.
39. M. T. Rosenstein, A. G. Barto, and R. E. A. Van Emmerik. Learning at the level of synergies for a robot weightlifter. *Robotics and Autonomous Systems*, 54(8):706–717, 2006.
40. S. Schaal, A. Ijspeert, and A Billard. Computational approaches to motor learning by imitation. *Philosophical Transaction of the Royal Society of London: Series B, Biological Sciences*, 358(1431):537–547, 2003.
41. N. Shafii, N. Lau, and L. P. Reis. Learning to walk fast: Optimized hip height movement for simulated and real humanoid robots. *Journal of Intelligent & Robotic Systems*, 80(3):555–571, 2015.
42. H. Shen, J. Yosinski, P. Kormushev, D. G. Caldwell, and H. Lipson. Learning fast quadruped robot gaits with the rl power spline parameterization. *Bulgarian Academy of Sciences, Cybernetics and Information Technologies*, 12(3):66–75, 2012.
43. F. Stulp, J. Buchli, E. Theodorou, and S. Schaal. Reinforcement learning of full-body humanoid motor skills. In *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, pages 405–410, Nashville, TN, USA, December 2010.
44. T. Sugihara and Y. Nakamura. Boundary condition relaxation method for stepwise pedipulation planning of biped robot. *IEEE Transactions on Robotics*, 25(3):658–669, 2009.
45. E. Theodorou, J. Buchli, and S. Schaal. A Generalized Path Integral Control Approach to Reinforcement Learning. *The Journal of Machine Learning Research*, 11:3137–3181, December 2010.
46. E. Theodorou, J. Buchli, and S. Schaal. Reinforcement learning of motor skills in high dimensions: a path integral approach. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2010.
47. B. Ugurlu, T. Hirabayashi, and A. Kawamura. A unified control frame for stable bipedal walking. In *IEEE Intl. Conf. on Industrial Electronics and Control*, pages 4167–4172, Porto, Portugal, 2009.
48. B. Ugurlu, J. A. Saglia, N. G. Tsagarakis, S. Morfey, and D. G. Caldwell. Bipedal hopping pattern generation for passively compliant humanoids: Exploiting the resonance. *IEEE Transactions on Industrial Electronics*, 61(10):5431–5443, 2014.
49. B. Ugurlu, N. G. Tsagarakis, E. Spyrakos-Papastravridis, and D. G. Caldwell. Compiant joint modification and real-time dynamic walking implementation on bipedal robot cCub. In *IEEE Intl. Conf. on Mechatronics*, 2011.
50. Y. Wada and K. Sumita. A reinforcement learning scheme for acquisition of via-point representation of human motion. In *Proc. of the IEEE Intl Conference on Neural Networks*, volume 2, pages 1109–1114, July 2004.
51. R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3-4):229–256, 1992.
52. M. Wisse, A. L. Schwab, R. Q. van der Linde, and F. C. T. van der Helm. How to keep from falling forward: Elementary swing leg action for passive dynamic walkers. *IEEE Transactions on Robotics*, 21(3):393–401, 2005.
53. Y. Xiaoxiang and F. Iida. Minimalistic models of an energy-efficient vertical-hopping robot. *IEEE Transactions on Industrial Electronics*, 61(2):1053–1062, 2014.