WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia IJCNN

Challenges for the Policy Representation when Applying Reinforcement Learning in Robotics

Petar Kormushev, Sylvain Calinon, Darwin G. Caldwell
Department of Advanced Robotics,
Istituto Italiano di Tecnologia, via Morego 30, 16163 Genova, Italy
Email: {petar.kormushev, sylvain.calinon, darwin.caldwell}@iit.it

Barkan Ugurlu
Toyota Technological Institute,
468-8511 Nagoya, Japan
Email: barkanu@toyota-ti.ac.jp

***Abstract*—A summary of the state-of-the-art reinforcement learning in robotics is given, in terms of both algorithms and policy representations. Numerous challenges faced by the policy representation in robotics are identified. Two recent examples for application of reinforcement learning to robots are described: pancake flipping task and bipedal walking energy minimization task. In both examples, a state-of-the-art Expectation-Maximization-based reinforcement learning algorithm is used, but different policy representations are proposed and evaluated for each task. The two proposed policy representations offer viable solutions to four rarely-addressed challenges in policy representations: *correlations*, *adaptability*, *multi-resolution*, and *globality*. Both the successes and the practical difficulties encountered in these examples are discussed.**

I. INTRODUCTION

It has been a long-recognized fact that robots need more than a fixed repertoire of skills - they need the ability to learn new tasks. Over the years, many approaches for teaching new skill to robots have been proposed and implemented. Currently, there are at least four well-established types of approaches: direct programming, kinesthetic teaching, imitation learning, and reinforcement learning. All of these approaches are being actively used, and each one has its own advantages and disadvantages, and is preferred in certain environments.

For example, *direct programming* is still being actively used in industrial settings, where the environment is well-structured and is crucial to control precisely the movement of the robot. *Kinesthetic teaching*, i.e. manually moving the robot and recording its motion, usually works only for smaller, lightweight robots, or robots driven by gravity-compensation controllers [1]. In these cases, it is possible to directly guide the robot's end-effector through the task, record and replay its motion, assuming that neither the task nor the objects change. *Imitation learning* provides more adaptability to changes, and has been successfully applied many times for learning movement tasks on robots, for which the human teacher can demonstrate a successful execution [2]–[4].

Reinforcement learning (RL, [5]) may not be the most widespread machine learning approach, but it has created a well-defined niche for its application in robotics [6]–[10]. The main motivation for using reinforcement learning to teach robots new skills is that it offers three previously missing abilities:

- to learn new tasks which even the human teacher cannot physically demonstrate (e.g. jump three meters high, lift heavy weights, move very fast, etc.);
- to learn to achieve optimization goals of difficult problems which have no analytic formulation or no known closed form solution, when even the human teacher does not know which the optimum is, by using only a known cost function (e.g. minimize the used energy for performing a task, or find the fastest gait, etc.);
- to learn to adapt a skill to a new, previously unseen version of a task (e.g. learning to walk from flat ground to slope, learning to generalize a task to new previously unseen parameter values, etc.). Some imitation learning approaches can also do this, but in a much more restricted way (e.g. by adjusting parameters of a learned model, without being able to change the model itself).

In the following Section II, we present an overview of the most important recent RL algorithms that are being successfully applied in robotics. Then, in Section III we identify numerous challenges posed by robotics on the RL policy representation. To illustrate some of these challenges, and to propose some example solutions to them, in two consecutive Sections V and VI we give two representative examples for real-world application of RL in robotics. Both of them are based on the same RL algorithm, but each faces different policy representation problems and therefore requires different solutions. Finally, in Section VII we conclude with a brief peek into the future of robotics, revealing in particular the potential wider need for RL.

II. STATE-OF-THE-ART REINFORCEMENT LEARNING ALGORITHMS IN ROBOTICS

Robot systems are naturally of high-dimensionality, having many degrees of freedom (DoF), continuous states and actions, and high noise. Because of this, traditional RL approaches based on MDP/POMDP/discretized state and action spaces have problems scaling up to work in robotics, because they suffer severely from the curse of dimensionality. The first partial successes in applying RL to robotics came with the function approximation techniques, but the real "renaissance" came with the policy-search RL methods.

In policy-search RL, instead of working in the huge state/action spaces, a smaller policy space is used, which contains all possible policies representable with a certain choice of policy parameterization. Thus, the dimensionality is drastically reduced, and the convergence speed is increased.

Until recently, *policy-gradient algorithms* (such as Episodic Natural Actor-Critic eNAC [11] and Episodic REINFORCE [12]) have been a well-established approach for implementing policy-search RL [8]. Unfortunately, policy-gradient algorithms have certain shortcomings, such as high sensitivity to the learning rate and the exploratory variance.

To avoid such problems, Kober *et al* proposed in [13] an episodic RL algorithm called PoWER (*Policy learning by Weighting Exploration with the Returns*). It is based on Expectation-Maximization algorithm (EM) and has one major advantage over policy-gradient-based approaches: 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 superior performance in tasks learned directly on a real robot, by applying importance sampling technique to reuse efficiently previous experience.

Another state-of-the-art policy-search RL algorithm, called *PI^2* (*Policy Improvement with Path Integrals*), was proposed by Theodorou *et al* in [14], for learning parameterized control policies based on the framework of stochastic optimal control with path integrals. They derived update equations for learning which avoid numerical instabilities because 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 dog.

III. CHALLENGES FOR THE POLICY REPRESENTATION IN ROBOTICS

Only having a good policy-search RL algorithm is not enough for solving real-world problems in robotics. Before any given RL algorithm can be applied to learn a task on a robot, an appropriate *policy representation* (also called *policy encoding*) needs to be devised. This is important, because the choice of policy representation determines what in principle can be learned by the RL algorithm (i.e. the policy search space), analogous to the way a hypothesis model determines what kind of data a regression method can fit well. In addition, the policy representation can have significant influence on the RL algorithm itself, e.g. it can help or impede the convergence, or influence the variance of the generated policies.

However, creating a good policy representation is not a trivial problem, due to a number of serious *challenges*, posed by the high requirements from a robotic system, such as:

- *smoothness* - the policy representation needs to encode smooth, continuous trajectories, without sudden accelerations or jerks, in order to be safe for the robot itself, and also to reduce its energy consumption;
- *safety* - the policy should be safe not only for the robot (in terms of joint limits, torque limits, work space restrictions, obstacles, etc.), but also for the people around it;
- *gradual exploration* - the representation should allow gradual, incremental exploration, so that the policy does not suddenly change by a lot; e.g. in state-action based policies, changing the policy action at only a single state could cause a sudden dramatic change in the overall behavior of the system when following this new branch of the policy, which is not desirable neither for the robot, nor for the people around it;
- *scalability* - to be able to scale up to high dimensions, and for more complex tasks; e.g. a typical humanoid robot nowadays has well above 50 DoF;
- *compactness* - despite the high-DoF of robots, the policy should use very compact encoding, e.g. it is impossible to directly use all points on a trajectory as policy parameters;
- *adaptability* - the policy parameterization should be adaptable to the complexity and fidelity of the task, e.g. lifting weights vs. micro-surgery;
- *multi-resolution* - different parts of the policy parameterization should allow different resolution/precision;
- *prior/bias* - the policy parameterization should work without prior knowledge about the solution being sought, and without restricting unnecessarily the search scope for possible solutions (i.e. unbiasedness);
- *regularization* - the policy should allow to incorporate regularization to guide the exploration towards desired types of policies;
- *autonomy* - this is also called time-independence, and the idea is that the policy should not depend on precise time or position, in order to cope with unforeseen perturbations;
- *embodiment-agnostic* - the representation should not depend on any particular embodiment of the robot, e.g. joint-trajectory based policies cannot be transferred to another robot easily;
- *invariance* - the policy should be an invariant representation of the task (e.g. rotation-invariant, scale-invariant, position-invariant, etc.);
- *correlations* - the policy should encapsulate correlations between the control variables (e.g. actuator control signals), similar to the motor synergies found in animals;
- *globality* - the representation should help the RL algorithm to avoid local minima.

A good policy representation should provide solutions to all of these challenges. However, it is not easy to come up with such a policy representation that satisfies all of them. In fact, the existing state-of-the-art policy representations in robotics cover only subsets of these requirements, as highlighted in the next section.

IV. STATE-OF-THE-ART POLICY REPRESENTATIONS IN ROBOTICS

Traditionally, explicit time-dependent approaches such as cubic splines or higher-order polynomials were used as policy representations. These, however, are not autonomous, in the sense that they cannot cope easily with perturbations (unexpected changes in the environment). Currently, there are a number of efficient state-of-the-art representations available to address this and many of the other challenges mentioned earlier. We give three examples of such policy representations below:

• Guenter *et al* explored in [15] the use of *Gaussian Mixture Model* (GMM) and *Gaussian Mixture Regression* (GMR) to respectively encode compactly a skill and reproduce a generalized version of it. The model was initially learned by demonstration through *Expectation-Maximization* techniques. RL was then used to move the Gaussian centers in order to alter the reproduced trajectory by regression. It was successfully applied to the imitation of constrained reaching movements, where the learned movement was refined in simulation to avoid an obstacle that was not present during the demonstration attempts.

• Kober and Peters explored in [16] the use of *Dynamic Movement Primitives* (DMP) [17] as a compact representation of a movement. The DMP framework was originally proposed by Ijspeert *et al* [18], and further extended in [17], [19]. In DMP, a set of attractors is used to reach a target, whose influence is smoothly switched along the movement. The set of attractors is first learned by imitation, and a proportional-derivative controller is used to move sequentially towards the sequence of targets. RL is then used to explore the effect of changing the position of these attractors. The proposed approach was demonstrated with pendulum swing-up and ball-in-a-cup tasks [20].

• Pardo *et al* proposed in [21] a framework to learn coordination for simple rest-to-rest movements, by taking inspiration of the motor coordination, joint synergies, and the importance of coupling in motor control [22]–[24]. The authors suggested to start from a basic representation of the movement by considering point-to-point movements driven by a proportional-derivative controller, where each variable encoding the task is decoupled. They then extended the possibilities of movement by encapsulating coordination information in the representation. RL was then used to learn how to efficiently coordinate the set of variables which were originally decoupled.

Although these policy representations work reasonably well for specific tasks, neither one of them manages to address all of the challenges listed in the previous section, but only a different subset. In particular, the challenges of *correlations*, *adaptability*, *multi-resolution*, and *globality* are rarely addressed by the existing policy representations.

In the following two sections we give two concrete examples of tasks that pose such rarely-addressed challenges for the policy representation, and we propose some possible solutions to them. The two examples are: pancake flipping task and bipedal walking energy minimization task. In both examples, the same EM-based RL algorithm is used (PoWER), but different policy representations are devised to address the specific challenges in the task at hand. Videos of the two presented robot experiments are available online at [25].

V. EXAMPLE A: PANCAKE FLIPPING TASK

This example addresses mainly the *correlations*, *compactness*, and *smoothness* challenges described in Section III. We present an approach allowing a robot to acquire new motor skills by learning the couplings across motor control variables. The demonstrated skill is first encoded in a compact form through a modified version of DMP which encapsulates correlation information. RL is then used to modulate the mixture of dynamical systems initialized from the user's demonstration via weighted least-squares regression. The approach is evaluated on a torque-controlled 7-DoF Barrett WAM robotic arm. More implementation details can be found in [26].

A. Task description

The goal of the pancake flipping task is to first toss a pancake in the air, so that it rotates 180 degrees, and then to catch it with the frying pan. Due to the complex dynamics of the task, it is unfeasible to try learning it directly with *tabula rasa* RL. Instead, a person presents a demonstration of the task first via kinesthetic teaching, which is then used to initialize the RL policy. The experimental setup is shown in Fig. 1.

The pancake flipping task is difficult to learn from multiple demonstrations because of the high variability of the task execution, even when the same person is providing the demonstrations. Extracting the task constraints by observing multiple demonstrations is not appropriate in this case for two reasons:

- when considering such skillful movements, extracting the regularities and correlations from multiple observations would take too long, as consistency in the skill execution would appear only after the user has mastered the skill;
- the generalization process may smooth important acceleration peaks and sharp turns in the motion. Therefore, in such highly dynamic skillful tasks, early trials have shown that it was more appropriate to select a single successful demonstration (among a small series of trials) to initialize the learning process.

A noticeable problem with all of the existing policy representations is the lack of any coupling between the different variables. To address this problem, we propose an approach which builds upon the works above by taking into consideration the efficiency of DMP to encode a skill with a reduced number of states, and by extending the approach to take into consideration local coupling information across the different variables.

B. Proposed compact encoding with coupling

The proposed approach represents a movement as a superposition of basis force fields, where the model is initialized from weighted least-squares regression of demonstrated trajectories. RL is then used to adapt and improve the encoded skill by learning optimal values for the policy parameters. The proposed policy parameterization allows the RL algorithm to learn the coupling across the different motor control variables.

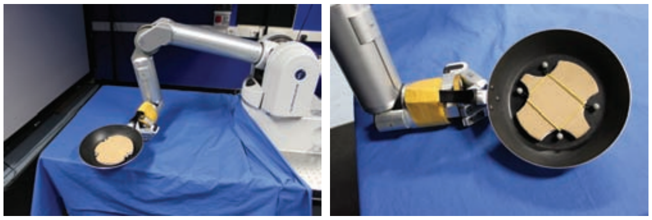


Fig. 1. Experimental setup for the pancake flipping task. A torque-controlled 7-DoF Barrett WAM robot learns to flip pancakes in the air and catch them with a real frying pan attached to its end-effector. Artificial pancakes with passive reflective markers are used to evaluate the performance of the learned policy.

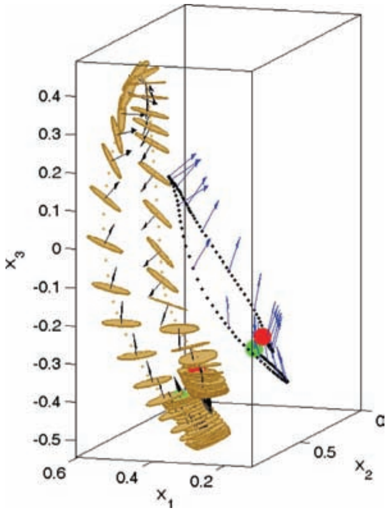


Fig. 2. Visualization of a real-world pancake flipping rollout (trial) performed by the robot. The pancake (in yellow) was successfully tossed and caught with the frying pan, and it rotated 180 degrees (for better visibility of the pancake's trajectory, the frying pan is not displayed here). The trajectory of the end-effector is displayed with black dots, and its orientation (represented by the normal vector) with blue arrows. The normal vectors perpendicular to the pancake are shown with black arrows.

A demonstration consisting of T positions x in 3D, velocities $\dot{x}$ and accelerations $\ddot{x}$ is shown to the robot. By considering flexibility and compactness issues, we propose to use a controller based on a mixture of K proportional-derivative systems:

$$\hat{\ddot{x}} = \sum_{i=1}^{K} h_i(t) \Big[K_i^{\mathcal{P}} (\mu_i^x - x) - \kappa^{\mathcal{V}} \dot{x} \Big]. \qquad (1)$$

The above formulation shares similarities with the DMP framework. We extend here the use of DMP by considering synergy across the different motion variables through the association of a full matrix $K_i^{\mathcal{P}}$ with each of the K primitives (or states) instead of a fixed $\kappa^{\mathcal{P}}$ gain.

The superposition of basis force fields is determined in (1) by an implicit time dependency, but other approaches using spatial and/or sequential information could also be used [27]. Similarly to DMP, a decay term defined by a canonical system $\dot{s} = -\alpha s$ is used to create an implicit time dependency $t = -\frac{ln(s)}{\alpha}$, where s is initialized with $s = 1$ and converges to zero. We define a set of Gaussians $\mathcal{N}(\mu_i^{\mathcal{T}}, \Sigma_i^{\mathcal{T}})$ in time space $\mathcal{T}$, with centers $\mu_i^{\mathcal{T}}$ equally distributed in time, and variance parameters $\Sigma_i^{\mathcal{T}}$ set to a constant value inversely proportional to the number of states. α is fixed depending on the duration of the demonstrations. The weights are defined by:

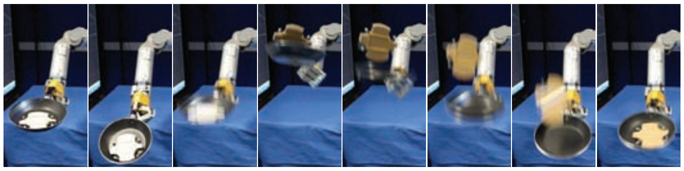


Fig. 3. Sequence of video frames showing a successful pancake flipping, performed on the WAM robot.

$$h_i(t) = \frac{\mathcal{N}(t;\ \mu_i^{\mathcal{T}}, \Sigma_i^{\mathcal{T}})}{\sum_{k=1}^{K} \mathcal{N}(t;\ \mu_k^{\mathcal{T}}, \Sigma_k^{\mathcal{T}})}. \qquad (2)$$

In (1), $\{K_i^{\mathcal{P}}\}_{i=1}^{K}$ is a set of full stiffness matrices, which we refer to as *coordination matrices*. Using the full coordination matrices (not only their diagonal elements) allows us to consider different types of synergies across the variables, where each state/primitive encodes local correlation information. Both attractor vectors $\{\mu_i^x\}_{i=1}^{K}$ and coordination matrices $\{K_i^{\mathcal{P}}\}_{i=1}^{K}$ in Eq. (1) are initialized from the observed data through weighted least-squares regression (see [26] for details).

C. Experiment

Custom-made artificial pancakes are used, whose position and orientation are tracked in real-time by a reflective marker-based *NaturalPoint OptiTrack* motion capture system.

The return of a *rollout* τ (also called trial) is calculated from the timestep reward $r(t)$. It is defined as a weighted sum of two criteria (orientational reward and positional reward), which encourage successful flipping and successful catching of the pancake

$$R(\tau) = w_1 \Big[\frac{\arccos(v_0 . v_{t_f})}{\pi} \Big] + w_2 e^{-||x^p - x^F||} + w_3 x_3^M, \qquad (3)$$

where w_i are weights, t_f is the moment when the pancake passes with downward direction the horizontal level at a fixed height Δ_h above the frying pan's current vertical position, v_0 is the initial orientation of the pancake (represented by a unit vector perpendicular to the pancake), v_{t_f} is the orientation of the pancake at time t_f, x^P is the position of the pancake center at time t_f, x^F is the position of the frying pan center at time t_f, and x_3^M is the maximum reached altitude of the pancake. The first term is maximized when the pancake's orientation (represented as a normal vector) at time t_f points in the opposite direction of the initial orientation, which happens in a successful flip. The second term is maximized when the pancake lands close to the center of the frying pan.

To learn new values for the coordination matrices, the RL algorithm PoWER is used. The policy parameters θ_n for the RL algorithm are composed of two sets of variables: the first set contains the full 3×3 coordination matrices $K_i^{\mathcal{P}}$

with the positional error gains in the main diagonal and the coordination gains in the off-diagonal elements; the second set contains the vectors μ_i^x with the attractor positions for the primitives.

D. Experimental results

In practice, around 60 rollouts were necessary to find a good policy that can reproducibly flip the pancake without dropping it. Fig. 2 shows a recorded sample rollout from the RL exploration, during which the pancake rotated fully 180 degrees and was caught successfully with the frying pan. Video frame sequence from a successful 180-degree flipping rollout is shown in Fig. 3.

It is interesting to notice the up-down bouncing of the frying pan towards the end of the learned skill, when the pancake has just fallen inside of it. The bouncing behavior is due to the increased compliance of the robot during this part of the movement. This was produced by the RL algorithm in an attempt to catch the fallen pancake inside the frying pan. Without it, a controller being too stiff would cause the pancake to bounce off from the surface of the frying pan and fall out of it. Such unintentional discoveries made by the RL algorithm highlight its important role for achieving adaptable and flexible robots.

In summary, the proposed policy parameterization based on superposition of basis force fields demonstrates three major advantages:

- it provides a mechanism for learning the couplings across multiple motor control variables, thus addressing the *correlations* challenge;
- it highlights the advantages of applying probabilistic approaches in RL for reducing the size of the representation, thus addressing the *compactness* challenge;
- it demonstrates that even fast, dynamic tasks can still be represented and executed in a safe-for-the-robot manner, addressing the *smoothness* challenge.

VI. EXAMPLE B: BIPEDAL WALKING ENERGY MINIMIZATION TASK

In this example, we address mainly the *adaptability*, *multi-resolution*, and *globality* challenges described in Section III.

Adaptive resolution methods *in state space* have been studied in RL before (see e.g. [28]). They address the pitfalls of discretization during reinforcement learning, and show that in high dimensions it is better if the learning does not plan uniformly over the state space. For example, in [29] Moore *et al* employed a decision-tree partitioning of state-space and applied techniques from game-theory and computational geometry to efficiently and adaptively concentrate high resolution on critical areas.

However, in the context of RL, adaptive resolution *in policy parameterization* remains largely unexplored so far. To address this challenge, we present a policy parameterization that can evolve dynamically while the RL algorithm is running without losing information about past experience. We show that the gradually increasing representational power of the policy parameterization helps to find better policies faster than a fixed parameterization. The particular problem at hand is an energy minimization problem for bipedal walking task. More implementation details can be found in [30].

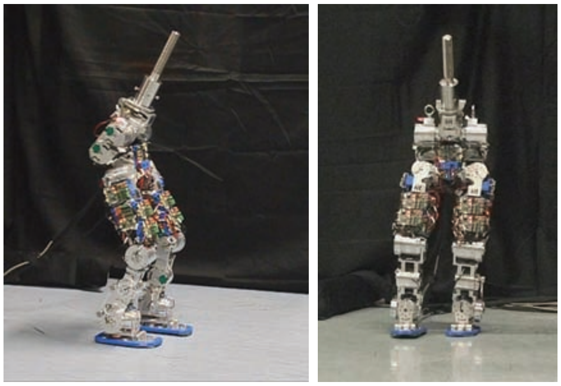


Fig. 4. The experimental setup for the bipedal walking energy minimization task, showing a snapshot of the lower body of the compliant humanoid robot COMAN during one walking rollout.

A. Energy minimization problem

Recent advances in robotics and mechatronics have allowed for the creation of a new generation of passively-compliant robots, such as the humanoid robot COMAN (derived from the cCub bipedal robot [31]) shown in Fig. 4.

Such robots have springs which can store and release energy and are essential for reducing the energy consumption and for achieving mechanical power peaks. However, it is difficult to manually engineer an optimal way to use the passive compliance for dynamic and variable tasks, such as walking. For instance, the walking energy minimization problem is very challenging because it is nearly impossible to be solved analytically, due to the difficulty in modeling accurately the properties of the springs, the dynamics of the whole robot and various nonlinearities of its parts. In this section, we apply RL to learn to minimize the energy consumption required for walking of this passively-compliant bipedal robot.

The vertical center of mass (CoM) movement is a crucial factor in reducing the energy consumption. Therefore, the proposed RL method is used to learn an optimal vertical trajectory for the center of mass (CoM) of the robot to be used during walking, in order to minimize the energy consumption. In order to apply RL in robotics to optimize the movement of the robot, first the trajectory needs to be represented (encoded) in some way. This particular experiment is based on cubic splines. Similar approaches have been investigated before in robotics under the name *via-points* [32]–[34].

However, there is a problem with applying a fixed policy parameterization RL to such a complex optimization problem.

B. Problems with fixed policy parameterization

In policy-search RL, in order to find a good solution, the policy parameterization has to be powerful enough to represent a large enough policy space, so that a good candidate solution is present in it. If the policy parameterization is too simple, with only a few parameters, then the convergence is quick, but often a sub-optimal solution is reached. If the policy parameterization 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 parameterization should be just the right amount, in order to provide both fast convergence and good enough solution.

Deciding what policy parameterization to use, and how simple/complex it should be, is a very difficult task, often performed via trial-and-error manually by the researchers. This additional overhead is usually not even mentioned in reinforcement learning papers, and falls into the category of "empirically tuned" parameters, together with the reward function, decay factor, exploration noise, weights, etc. Since changing the policy parameterization requires to restart the learning from scratch, throwing away all accumulated data, this process is slow and inefficient. As a consequence, the search for new solutions often cannot be done directly on a real-world robot system, and requires instead the use of simulations, which are not accurate enough.

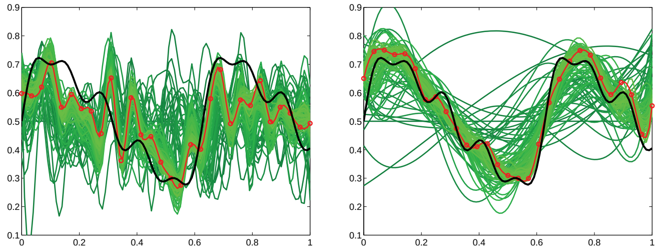
C. *Evolving policy parameterization*

To solve this problem, we propose an approach that allows to change the complexity of the policy representation dynamically while the reinforcement learning is running, without losing any of the collected data, and without having to restart the learning. We propose a mechanism which can incrementally "evolve" the policy parameterization as necessary, starting from a very simple parameterization and gradually increasing its complexity and thus, its representational power. The goal is to create an adaptive policy parameterization, which can automatically "grow" to accommodate increasingly more complex policies and get closer to the global optimum.

A very desirable side effect of this is that the tendency of converging to a sub-optimal solution will be reduced, because in the lower-dimensional representations this effect is less exhibited, and gradually increasing the complexity of the parameterization helps not to get caught in a poor local optimum.

The main difficulty to be solved is providing backward compatibility, i.e. how to design the subsequent policy representations in such a way, that they are backward-compatible with the previously collected data, such as past rollouts and their corresponding policies and rewards.

One of the simplest representations which have the property of backward compatibility, are the geometric splines. For example, if we have a cubic spline with K knots, and then we increase the number of knots, we can still preserve the 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, the idea we propose is to use the spline knots as a policy parameterization, and use the spline backward compatibility property for evolving the policy parameterization without losing the previously collected data.



(a) Fixed policy parameterization (b) Evolving policy parameterization

Fig. 5. Comparison of the policy output from RL with fixed policy parameterization (30-knot spline) versus evolving policy parameterization (from 4- 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. Due to the lower policy-space dimensionality at the beginning, the evolving policy parameterization approaches much faster the shape of the globally-optimal trajectory. The fixed policy parameterization 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.

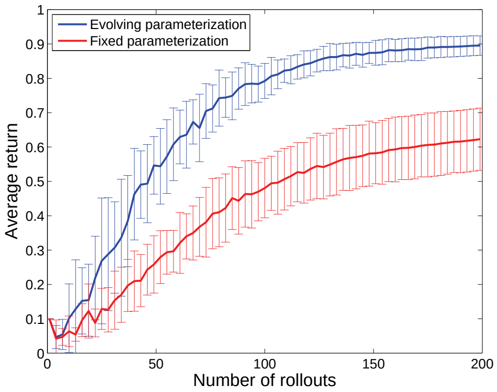


Fig. 6. Comparison of the convergence of the RL algorithm with fixed policy parameterization (30-knot spline) versus evolving policy parameterization (from 4- to 30-knot spline). The results are averaged over 20 runs of each of the two algorithms. The standard deviation is indicated with error bars. In addition to faster convergence and higher achieved rewards, the evolving policy parameterization also achieves lower variance compared to the fixed policy parameterization.

The proposed technique for evolving the policy parameterization can be used with any policy-search RL algorithm. For this particular implementation, we use the PoWER, due to its low number of parameters that need tuning.

Different techniques can be used to trigger the increase in the number of knots of the spline representation. For this example, we used a fixed, pre-detemined trigger, activating at regular time intervals.

D. *Evaluation of evolving policy parameterization*

In order to evaluate the proposed evolving policy parameterization, we conduct a function approximation experiment. The goal is to compare the proposed method with a conventional fixed policy parameterization method that uses the same reinforcement learning algorithm as a baseline.

For this experiment, the reward function is defined as follows:

$$R(\tau) = e^{-\int_0^1 [\tau(t)-\tilde{\tau}(t)]^2 dt}, \tag{4}$$

where $R(\tau)$ is the return of a rollout (the policy-generated trajectory) τ, and $\tilde{\tau}$ is the (unknown to the learning algorithm) function that the algorithm is trying to approximate.

Fig. 5 shows a comparison of the generated policy output produced by the proposed evolving policy parameterization method, compared with the output from the conventional fixed policy parameterization method. Fig. 6 shows that the convergence properties of the proposed method are significantly better than the conventional approach.

E. Bipedal walking experiment

For the real-world bipedal walking experiment we use the lower body of the passively-compliant humanoid robot COMAN which has 17 DoF. The experimental setup is shown in Fig. 4.

To generate trajectories for the robot joints, we use a custom variable-height bipedal walking generator. Given the z-axis CoM trajectory provided by the RL, we use ZMP (Zero Moment Point) concept for deriving the x- and y-axis CoM trajectories.

To calculate the reward, we measure the actual electrical energy used by the motors of the robot. The return of a rollout τ depends on the average electric energy consumed per walking cycle, and is defined as:

$$R(\tau) = e^{-k\frac{1}{c}\sum_{j\in J} E_j(t_1,t_2)}, \tag{5}$$

where J is the set of joints whose energy consumption we try to minimize, $E_j(t_1, t_2)$ is the accumulated consumed electric energy for the motor of the j-th individual joint of COMAN, and k is a scaling constant. To reduce the effect of noise on the measurement, for each rollout the robot walks for 16 seconds (from time t_1 to t_2), which corresponds to 8 steps ($c = 4$ walking cycles).

The learning converged after 150 rollouts. The total cumulative distance traveled by the robot during our experiments was 0.5 km. The discovered optimal policy by the RL algorithm, for which the lowest energy consumption was achieved, consumes 18% less energy than a conventional fixed-height walking, which is a significant improvement.

In summary, the proposed evolving policy parameterization demonstrates three major advantages:

- it achieves faster convergence and higher rewards than the fixed policy parameterization, using varying resolution for the policy parameterization, thus addressing the *adaptability* and *multi-resolution* challenges;
- it exhibits much lower variance of the generated policies, addressing the *gradual exploration* challenge;
- it helps to avoid local minima, thus addressing the *globality* challenge.

VII. CONCLUSION

We summarized the state-of-the-art for RL in robotics, in terms of both algorithms and policy representations. We identified a significant number of the existing challenges for policy representations in robotics. We showed two examples for extensions of the capabilities of policy representations, on two real-world tasks: pancake flipping and bipedal walking. In these examples we proposed solutions to four rarely-addressed challenges in policy representations: *correlations*, *adaptability*, *multi-resolution*, and *globality*.

What does the future hold for RL in robotics? This seems difficult to predict *for RL*, but it is relatively easier to predict the near-future trend *for robotics*. Robotics is moving towards higher and higher DoF robots, having more nonlinear elements, variable passive compliance, variable damping, flexible joints, reconfigurability, fault tolerance, independence, power autonomy, etc. Robots will be progressively going out of the robot labs and into everyday life.

As the robot hardware complexity increases to higher levels, the conventional engineering approaches and analytical methods for robot control will start to fail. Therefore, machine learning (and RL in particular) will inevitably become a more and more important tool to cope with the ever-increasing complexity of the physical robotic systems. And the future RL candidates will have to address an ever-growing number of challenges accordingly.

ACKNOWLEDGEMENTS

This work was partially supported by the AMARSi European project under contract FP7-ICT-248311, and by the PANDORA European project under contract FP7-ICT-288273.

REFERENCES

[1] 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, pp. 3970–3975.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Secaucus, NJ, USA: Springer, 2008, pp. 1371–1394.

[3] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, 2009.

[4] P. Kormushev, S. Calinon, and D. G. Caldwell, "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input," *Advanced Robotics*, vol. 25, no. 5, pp. 581–603, 2011.

[5] R. S. Sutton and A. G. Barto, *Reinforcement learning : an introduction*, ser. Adaptive computation and machine learning. Cambridge, MA, USA: MIT Press, 1998.

[6] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal, "Skill learning and task outcome prediction for manipulation," in *Intl Conf. on Robotics and Automation (ICRA)*, Shanghai, China, 2011.

[7] 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)*, Nashville, TN, USA, December 2010, pp. 405–410.

[8] J.Peters and S.Schaal, "Policy gradient methods for robotics," in *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[9] A. Coates, P. Abbeel, and A. Y. Ng, "Apprenticeship learning for helicopter control," *Commun. ACM*, vol. 52, no. 7, pp. 97–105, 2009.

[10] P. Kormushev, S. Calinon, R. Saegusa, and G. Metta, "Learning the skill of archery by a humanoid robot iCub," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, Nashville, TN, USA, December 2010, pp. 417–423.

[11] J. Peters and S. Schaal, “Natural actor-critic,” *Neurocomput.*, vol. 71, no. 7-9, pp. 1180–1190, 2008.

[12] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Mach. Learn.*, vol. 8, no. 3-4, pp. 229–256, 1992.

[13] J. Kober and J. Peters, “Learning motor primitives for robotics,” in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, May 2009, pp. 2112–2118.

[14] E. Theodorou, J. Buchli, and S. Schaal, “A Generalized Path Integral Control Approach to Reinforcement Learning,” *The Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, December 2010.

[15] F. Guenter, M. Hersch, S. Calinon, and A. Billard, “Reinforcement learning for imitating constrained reaching movements,” *Advanced Robotics*, vol. 21, no. 13, pp. 1521–1544, 2007.

[16] J. Kober and J. Peters, “Policy search for motor primitives in robotics,” in *Advances in Neural Information Processing Systems*, 2009, vol. 21, pp. 849–856.

[17] S. Schaal, P. Mohajerian, and A. J. Ijspeert, “Dynamics systems vs. optimal control a unifying view,” *Progress in Brain Research*, vol. 165, pp. 425–445, 2007.

[18] A. J. Ijspeert, J. Nakanishi, and S. Schaal, “Trajectory formation for imitation with nonlinear dynamical systems,” in *Proc. IEEE Intl Conf. on Intelligent Robots and Systems (IROS)*, 2001, pp. 752–757.

[19] H. Hoffmann, P. Pastor, D. H. Park, and S. Schaal, “Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance,” in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2009, pp. 2587–2592.

[20] J. Kober, “Reinforcement learning for motor primitives,” Master’s thesis, University of Stuttgart, Germany, August 2008.

[21] D. Pardo, “Learning rest-to-rest motor coordination in articulated mobile robots,” PhD thesis, Technical University of Catalonia (UPC), 2009.

[22] E. Todorov and M. I. Jordan, “Optimal feedback control as a theory of motor coordination,” *Nature Neuroscience*, vol. 5, pp. 1226–1235, 2002.

[23] R. Huys, A. Daffertshofer, and P. J. Beek, “The evolution of coordination during skill acquisition: the dynamical systems approach,” in *Skill Acquisition in Sport: Research, Theory and Practice*, A. M. Williams and N. J. Hodges, Eds. Routledge, 2004, pp. 351–373.

[24] M. Bernikera, A. Jarcb, E. Bizzic, and M. C. Trescha, “Simplified and effective motor control based on muscle synergies to exploit musculoskeletal dynamics,” in *Proc. Natl Acad. Sci. USA*, vol. 106, no. 18, 2009, pp. 7601–7606.

[25] Videos accompanying this paper. [Online]. Available: http://kormushev.com/research/videos/

[26] 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)*, Taipei, Taiwan, October 2010, pp. 3232–3237.

[27] S. Calinon, F. D’halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, “Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression,” *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

[28] A. Bernstein and N. Shimkin, “Adaptive-resolution reinforcement learning with polynomial exploration in deterministic domains,” *Machine Learning*, vol. 81, no. 3, pp. 359–397, 2010.

[29] A. W. Moore and C. G. Atkeson, “The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces,” *Machine Learning*, vol. 21, pp. 199–233, December 1995.

[30] P. Kormushev, B. Ugurlu, S. Calinon, N. 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)*, San Francisco, USA, September 2011, pp. 318–324.

[31] 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.

[32] H. Miyamoto, J. Morimoto, K. Doya, and M. Kawato, “Reinforcement learning with via-point representation,” *Neural Networks*, vol. 17, pp. 299–305, April 2004.

[33] J. Morimoto and C. G. Atkeson, “Learning biped locomotion: Application of poincare-map-based reinforcement learning,” *IEEE Robotics and Automation Magazine*, vol. 14, no. 2, pp. 41–51, 2007.

[34] 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*, vol. 2, July 2004, pp. 1109–1114.