Evaluation of physical damage associated with action selection strategies in reinforcement learning

Koryakovskiy, Ivan; Vallery, Heike; Babuška, Robert; Caarls, Wouter

DOI
10.1016/j.ifacol.2017.08.1218

Publication date
2017

Document Version
Final published version

Published in
IFAC-PapersOnLine

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 6928–6933

Evaluation of physical damage associated with action selection strategies in reinforcement learning [⋆]

Ivan Koryakovskiy [*] Heike Vallery [*] Robert Babuška [**] Wouter Caarls [***]

[*] *Department of BioMechanical Engineering, TU Delft, Netherlands ({i.koryakovskiy, h.vallery}@tudelft.nl)*
[**] *Delft Center for Systems and Control, TU Delft, Netherlands (r.babuska@tudelft.nl)*
[***] *Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro, Brazil (wouter@caarls.org)*

Abstract: Reinforcement learning techniques enable robots to deal with their own dynamics and with unknown environments without using explicit models or preprogrammed behaviors. However, reinforcement learning relies on intrinsically risky exploration, which is often damaging for physical systems. In the case of the bipedal walking robot Leo, which is studied in this paper, two sources of damage can be identified: fatigue of gearboxes due to backlash re-engagements, and the overall system damage due to falls of the robot. We investigate several exploration techniques and compare them in terms of gearbox fatigue, cumulative number of falls and undiscounted return. The results show that exploration with the Ornstein-Uhlenbeck (OU) process noise leads to the highest return, but at the same time it causes the largest number of falls. The Previous Action-Dependent Action (PADA) method results in drastically reduced fatigue, but also a large number of falls. The results reveal a previously unknown trade-off between the two sources of damage. Inspired by the OU and PADA methods, we propose four new action-selection methods in a systematic way. One of the proposed methods with a time-correlated noise outperforms the well-known ϵ-greedy method in all three benchmarks. We provide guidance towards the choice of exploration strategy for reinforcement learning applications on real physical systems.

Keywords: Reinforcement learning control, Fault detection and diagnosis, Analysis of reliability and safety, Adaptation and learning in physical agents, Autonomous robotic systems

1. INTRODUCTION

Until recently, robotic applications were mostly limited to controlled and well-predictable environments such as factories or space. However, currently scientists and engineers strive to bring robots to uncontrolled, partially observable and human-friendly environments. Despite the existence of advanced software and hardware, many challenges remain in the integration of robots into our society.

Machine learning techniques enable robots to deal with unknown environments without using explicit models or preprogrammed policies. In simulations, impressive results were obtained with deep learning in the actor-critic setting (Lillicrap et al., 2015). The authors use a deep neural network for learning both from low-dimensional state descriptions and high-dimensional renderings of the environment. In both cases, they have shown the ability of their approach to scale to complex tasks such as control of a seven-degree-of-freedom arm and bipedal locomotion, reaching a good control policy in at most 2.5 million steps.

However, the application of learning on real robots can be very costly. For example, our robot Leo, shown in Figure 1, can learn to walk by first observing a preprogrammed controller and then improving the observed policy using reinforcement learning (RL) (Schuitema, 2012). Without the preprogrammed controller, Leo's gearboxes can only withstand five minutes of learning as a direct result of the aggressive nature of its learning strategy, involving large and rapidly changing motor torques (Meijdam, 2013). Therefore, in this article we investigate possibilities of reducing the damage while learning.

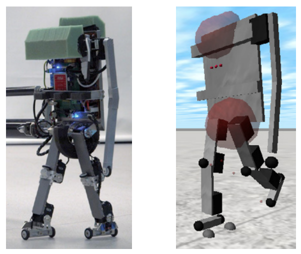
Fig. 1. 7 DoF robot Leo (*left*) and its model (*right*).

Garcia and Fernandez (2015) give an overview of Safe RL. Perhaps the most prominent method of limiting damage is to define specific parameterized policies that are benign to the hardware at hand and then to learn the parameters only. This can, for example, be done from

[⋆] I. Koryakovskiy, H. Vallery and R.Babuška were supported by the European project KOROIBOT FP7-ICT-2013-10/611909.

2405-8963 © 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.1218

optimal control roll-outs (Levine and Koltun, 2013) or kinesthetic teach-in (Kober and Peters, 2011). In general, they can achieve good-quality policies within dozens of episodes but require a few human demonstrations for each task that needs to be learned.

An exploration method by Moldovan and Abbeel (2012) requires a model with known uncertainty in the dynamics. It restricts a set of policies to ergodic ones, which are policies that intrinsically encode the possibility of returning to an initial state from any other state.

On the hardware level, multiple contact dynamics were used in order to dissipate impacts with a minimum damaging effect on the robot (Ha and Liu, 2015). This planning strategy requires a model and explicit formulation of damage measures.

When aiming at higher robot autonomy and better generalization to unknown environments and new tasks, learning to control fragile systems in a model-free setting is essential. Only a few methods have been proposed that explicitly consider safe exploration in this setting. For instance, trust region policy optimization (Schulman et al., 2015) generates near-monotonic improvements of a policy by choosing sufficiently small step sizes. Unfortunately, as mentioned by Lillicrap et al. (2015), it appears to be less data-efficient than unconstrained policies.

Another method, proposed by Gehring and Precup (2013), identifies areas of high randomness in the rewards or transitions and avoids those during exploration. It was shown that the approach can scale to high-dimensional problems and noisy state information.

Finally, superior results regarding mean time before failure (MTBF) were achieved by the Previous Action-Dependent Action (PADA) algorithm of Meijdam (2013), where the author constrained a set of possible actions to remain within a fixed distance from a previous action. Our work can be seen as a continuation of this research. We select four commonly used exploration methods (Greedy, ϵ-greedy, PADA, OU) for the comparison on the bipedal robot Leo. Earlier experiments (Meijdam, 2013) indicated that robot falls and foot impacts also contribute significantly to the MTBF. To distinguish these two sources, we compute the cumulative number of falls in addition to calculation of fatigue, MTBF and undiscounted return. The obtained results reveal a previously unknown trade-off between the number of falls and gearbox fatigue. Furthermore, by proposing four new exploration methods, we bridge the gap between the methods mentioned above and provide a better insight into the influence of exploration on the damage of Leo. As an outcome, we provide guidance towards a choice of exploration strategy for physical RL applications.

2. REINFORCEMENT LEARNING

2.1 The Markov decision process

Reinforcement learning can deal with unmodelled and noisy environments. The dimension of the state space is n_x with $\mathcal{X} \subset \mathbb{R}^{n_x}$ being the set of possible states. The dimension of the action space (the space of the control signals) is n_u with $\mathcal{U} \subset \mathbb{R}^{n_u}$ being the set of possible actions. Then a Markov decision process is defined as the quadruple $\langle \mathcal{X}, \mathcal{U}, \mathcal{P}, \mathcal{R} \rangle$, where $\mathcal{P} : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \to [0, 1]$ is a transition function that defines the probability of ending in state $x_{k+1} \in \mathcal{X}$ after executing action $u_k \in \mathcal{U}$ in state $x_k \in \mathcal{X}$. The reward function $\mathcal{R} : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \to \mathbb{R}$ gives a real-valued reward $r_{k+1} = \mathcal{R}(x_k, u_k, x_{k+1})$ for the particular transition between states. A Markov decision process satisfies the Markov property, which assumes that the current state x_k provides enough information to determine an optimal action u_k.

A deterministic control policy $\pi : \mathcal{X} \to \mathcal{U}$ defines an action u_k taken in a state x_k. The goal of learning a *continuing* task is to find an optimal control policy π^* that maximizes the discounted return,

$$G(x_k) = \mathbb{E}\left\{ \sum_{i=0}^{\infty} \gamma^i r_{k+i+1} \right\},$$

where the immediate rewards are exponentially decayed by the discount rate $\gamma \in [0, 1)$ – rewards further in the future contribute less to the return.

The state-action value function $Q^\pi(x_k, u_k)$ denotes the expected return assuming that the system starts in the state x_k with the action u_k and then follows a prescribed control policy π. The optimal control policy maximizes the value for each state-action pair.

In this article, we solve a bipedal walking task using the well-known model-free temporal-difference RL algorithm SARSA (Sutton and Barto, 1998). The value function is represented by a linear function approximator using binary features defined by tile coding (Albus, 1975). A discrete action u_k is selected in state x_k according to one of the action-selection methods, and then the value function is updated according to

$$\begin{aligned} Q^\pi(x_{k-1}, u_{k-1}) &= Q^\pi(x_{k-1}, u_{k-1}) \\ &\quad + \alpha(r_k + \gamma Q^\pi(x_k, u_k) - Q^\pi(x_{k-1}, u_{k-1})). \end{aligned}$$

We implement standard accumulating eligibility traces for speeding up the convergence of SARSA.

In RL, exploration is achieved either by taking suboptimal actions with a certain probability or by initializing the value function optimistically, that is with values higher than the expected return. This causes visited states to become less attractive than states that have not been visited yet (Matignon et al., 2006). In this article, we only focus on methods of suboptimal action selection and do not consider optimistic initialization.

2.2 Action-selection methods

All studied action-selection methods and the relations between them are summarized in Figure 2. In the following, we explain details of each method.

Greedy. This method always takes the expected best possible action

$$u_k = \arg\max_{u \in \mathbb{U}} Q^\pi(x_k, u),$$

where $\mathbb{U} \subset \mathcal{U}$ is a discrete subset of possible actions.

ϵ-greedy. This method takes a greedy action most of the time, but with a small probability $\epsilon > 0$ it samples a random action from a uniform distribution,

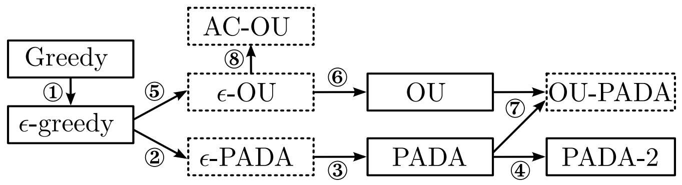

Fig. 2. A relation between conventional (solid line) and proposed (dashed line) exploration methods. ① Take a random action with probability ϵ. ② Select a random action within a Δ interval. ③ Select greedy and random actions within the Δ interval. ④ Include $u_{k-1} \pm 2\Delta$ actions to the action selection set. ⑤ Add a time-correlated noise to a greedy action taken with probability ϵ. ⑥ Add a time-correlated noise to a greedy action taken with probability 1. ⑦ Add a time-correlated noise to a greedy action constrained by the action selection set. ⑧ With probability ϵ, take an action correlated with a previous action.

$$u_k = \begin{cases} \arg\max\limits_{u \in \mathbb{U}} Q^\pi(x_k, u), & \text{with probability } 1-\epsilon \\ \text{uniform}(\mathbb{U}), & \text{otherwise.} \end{cases}$$

PADA. Greedy and ϵ-greedy methods choose a future action independently from the previous action. However, it was shown by Meijdam (2013) that selection of a new action from a subset of actions defined around the previous action dramatically reduces the MTBF of RL. In case of Leo, the authors used a previous action and two neighboring actions:

$$u_k = \begin{cases} \arg\max\limits_{u \in \tilde{\mathbb{U}}(u_{k-1})} Q^\pi(x_k, u), & \text{with probability } 1-\epsilon \\ \text{uniform}(\tilde{\mathbb{U}}(u_{k-1})), & \text{otherwise,} \end{cases}$$

where the set of neighboring actions is defined as $\tilde{\mathbb{U}}(u_{k-1}) = \{u_{k-1} - \Delta, u_{k-1}, u_{k-1} + \Delta\}$, and Δ is equal to the discretization step of controls.

In the case of the PADA-2 method, the set of neighboring actions is extended with actions located $\pm 2\Delta$ away from a previous action.

OU. Rather than taking an entirely random action such as with the ϵ-greedy method, the Ornstein-Uhlenbeck process (Lillicrap et al., 2015) adds time-correlated noise to a greedy action. The OU exploration term n_k is the integral over a Gaussian noise signal $g_k \sim \mathcal{N}(0, 1)$, but pulled towards an asymptotic mean μ,

$$n_k = n_{k-1} + \theta(\mu - n_{k-1}) + \sigma g_k$$
$$u_k = \arg\max_{u \in \mathbb{U}} Q^\pi(x_k, u) + C n_k.$$

The three parameters, $\theta > 0$, $\sigma > 0$ and μ, influence the dynamics of the process, and C scales the noise to the values of admissible actions.

We establish a connection between the described methods by introducing four new action-selection methods.

ϵ-PADA. The method selects a greedy action at exploitation steps and a random action within $\pm\Delta$ bound at exploration steps, therefore bridging ϵ-greedy and PADA methods.

$$u_k = \begin{cases} \arg\max\limits_{u \in \mathbb{U}} Q^\pi(x_k, u), & \text{with probability } 1-\epsilon \\ \text{uniform}(\tilde{\mathbb{U}}(u_{k-1})), & \text{otherwise.} \end{cases}$$

ϵ-OU. The method bridges the gap between ϵ-greedy and OU by only adding the Ornstein-Uhlenbeck process noise at exploration steps,

$$n_k = n_{k-1} + \theta(\mu - n_{k-1}) + \sigma g_k$$
$$u_k = \begin{cases} \arg\max\limits_{u \in \mathbb{U}} Q^\pi(x_k, u), & \text{with probability } 1-\epsilon \\ \arg\max\limits_{u \in \mathbb{U}} Q^\pi(x_k, u) + C n_k, & \text{otherwise.} \end{cases}$$

OU-PADA. The method adds the Ornstein-Uhlenbeck process noise to the greedy action selected within $\pm\Delta$ bounds, therefore bridging OU and PADA methods.

$$n_k = n_{k-1} + \theta(\mu - n_{k-1}) + \sigma g_k$$
$$u_k = \arg\max_{u \in \tilde{\mathbb{U}}(u_{k-1})} Q^\pi(x_k, u) + C n_k$$

AC-OU. Inspired by the Ornstein-Uhlenbeck process, we introduce an Action-Correlated Ornstein-Uhlenbeck (AC-OU) action-selection method. As in the ϵ-greedy method, we separate exploratory and greedy actions. An exploratory action is selected based on the previous action so that it does not stress the system as much as a random action would do. As in the Ornstein-Uhlenbeck process, we add a θ-multiplied term, which works as an action regularization,

$$u_k = \begin{cases} \arg\max\limits_{u \in \mathbb{U}} Q^\pi(x_k, u), & \text{with probability } 1-\epsilon \\ u_{k-1} + \theta(\mu - u_{k-1}) + \sigma g_k, & \text{otherwise.} \end{cases}$$

Note that here σ and θ are applied on the action level and do not require scaling.

In addition to the above-described methods, we tried the Softmax action-selection method (Sutton and Barto, 1998), but there was no temperature for which it performed better than the Greedy method. For this reason, we excluded Softmax from further investigation.

Table 1 gives the parameters of the methods presented. Those used for ϵ-greedy (Schuitema, 2012) and PADA (Meijdam, 2013) were taken from the corresponding articles, while for the other methods we tested a range of values and selected the ones that led to the highest undiscounted return. Additionally, a SARSA learning rate $\alpha = 0.2$, a discount rate $\gamma = 0.9962$, an eligibility trace decay rate of 0.8582 and a sampling period of 0.033 s of Leo's controller were taken from Schuitema (2012).

3. LEO SIMULATIONS RESULT

We evaluate properties of the described action-selection methods using the Leo dynamics simulator. Following Schuitema (2012), we exploit the symmetry of the bipedal walking problem to reduce the state and action space dimensions to ten and three, respectively. Actions from a voltage range of $[-10.7\,\text{V}, 10.7\,\text{V}]$ are discretized into seven linearly spaced values. We selected $C = 10.7\,\text{V}$ to account for the whole range of admissible actions. The reward was constructed with the goal of promoting a fast but energy-efficient forward walking. The simulator includes a realistic model of the Dynamixel RX-28 motor with the last gear of the gearbox made of anodized aluminum. Torque τ applied to the last gear is calculated from voltage U, the motor's torque constant K_τ, gearbox

Table 1. Parameters of action-selection methods.

Method	Parameter values			
ϵ-greedy	$\epsilon = 0.050$			
PADA	$\epsilon = 0.050$;	$\Delta = 3.570$		
PADA-2	$\epsilon = 0.050$;	$\Delta = 3.570$		
OU	$\mu = 0.000$;	$\theta = 0.001$;	$\sigma = 0.020$	
ϵ-PADA	$\epsilon = 0.050$;	$\Delta = 3.570$		
ϵ-OU	$\epsilon = 0.050$;	$\mu = 0.000$;	$\theta = 0.001$;	$\sigma = 0.020$
OU-PADA	$\mu = 0.000$;	$\theta = 0.001$;	$\sigma = 0.020$;	$\Delta = 3.570$
AC-OU	$\mu = 0.000$;	$\theta = 0.100$;	$\sigma = 2.000$;	$\epsilon = 0.050$

ratio K_G, the joint velocity $\dot{\phi}$ and the winding resistance R by

$$\tau = K_\tau K_G \frac{U - K_\tau K_G \dot{\phi}}{R}.$$

Following Meijdam (2013), we use torque amplitude to estimate the number N_k of completely reversed cycles withstood before failure. The completely reversed stress cycle is the cycle with zero mean and an equal magnitude of positive and negative stress. Assuming that each of the 45 teeth of the last gear is equally stressed, the fatigue J of the gear is calculated by

$$J = \sum_{k=1}^{K} \frac{1}{45 N_k},$$

where K is the number of gear re-engagements during learning. Note that our measure of fatigue accounts only for the cases when the torque sign changes, and fatigue is not influenced by falls of the robot. MTBF during learning is predicted as the time when $J \geq 1$.

Figure 3 shows control trajectories of the left hip before and after learning. In the final policy, the ϵ-greedy and ϵ-OU methods showed high-frequency oscillations involving a change of voltage polarity. Greedy, OU, ϵ-PADA methods showed moderate voltage oscillations, and PADA, PADA-2, OU-PADA and AC-OU showed the least ones.

Table 2 summarizes the performance of the methods in terms of gearbox fatigue, MTBF at the beginning of learning and final MTBF after learning (i.e., when only greedy actions are applied), the cumulative number of falls of Leo and the undiscounted return obtained. A careful comparison of fatigue and MTBF during learning results of ϵ-greedy, ϵ-PADA and ϵ-OU with the help of Figure 4a reveals the difference between these benchmarks. The rate of fatigue accumulation was nonlinear and slowed down after approximately 25 min since the beginning of learning. This value can be regarded as an average number of gear replacements during learning. Therefore, fatigue gives a more accurate estimation of loss during learning comparing to MTBF, which only accounts for a fail-free learning time at the beginning of a simulation. To avoid clutter in plots, we decided to present the curves of the five most characteristic methods, Greedy, PADA, OU, OU-PADA and AC-OU.

PADA and OU-PADA methods resulted in a remarkably low fatigue, leaving behind all other methods. Extending the action selection set with just two more actions (PADA-2) already increased fatigue caused by the change of a torque sign, and most noticeably reduced final MTBF by more than four times. It also significantly decreased the cumulative number of falls.

All action-selection methods succeeded in learning a walking gait and reaching reasonable rewards, see Figure 4b. PADA and OU-PADA rising slopes were slightly less steep comparing to other methods, but OU-PADA reached a much higher level of end performance comparing to PADA. Table 2 shows that OU significantly outperformed the other methods.

The cumulative number of falls encountered during learning is shown in Figure 4c. The smallest number of falls was achieved by the Greedy method, which was closely followed by OU-PADA and then AC-OU. PADA and OU methods resulted in approximately 2.5 and 8 times larger numbers of falls compared to the Greedy method, respectively.

In this article, we do not experiment with the real robot, because that would incur a continuing damage. Meijdam (2013) demonstrated the increase of MTBF by limiting the changes in a control signal applied to the real Dynamixel RX-28 motor. This fact correlates well with our results.

4. DISCUSSION

PADA significantly outperformed all exploration methods in terms of MTBF and fatigue. However, during learning under this action-selection method, the simulated Leo underwent a significant number of falls and achieved the worst performance. While the decrease in performance was already described, the trade-off between number of falls and MTBF was previously unknown. The explanation of this could be the following: PADA always selects an action that is the same as or close to the previous one. This reduces fatigue because gear re-engagements happen much more rarely. However, the prevention of falls may require an immediate reaction, which may involve a rapid change of the control signal sign. This hypothesis closely correlates with the fact that PADA resulted in the smallest consecutive change of control signal among all studied methods. Reducing the constraints on actions as in PADA-2 also supports this hypothesis, because the cumulative number of falls was reduced at the expense of larger fatigue.

However, the absence of any constraint also led to more damage, which can be observed in the results of OU and OU-PADA. The OU explores very well in physical environments, but in the experiment it was the most demanding with respect to hardware endurance. Constraining actions as in OU-PADA not only reduced the fatigue, but also reduced the number of falls, at the cost of decreased walking performance.

It is important to note the difference between uniform noise (ϵ-greedy) and time-correlated noise (ϵ-OU) during exploration. The results in Table 2 demonstrate that time-correlated noise reduced the number of falls by more than 40%, leaving all other benchmark values within the confidence intervals of ϵ-greedy. ϵ-PADA and AC-OU showed similar results with a slight shift towards a lower fatigue, but a higher number of falls.

Both Greedy and AC-OU showed intermediate performance. Greedy underwent the lowest number of falls during learning, but AC-OU outperformed Greedy in terms

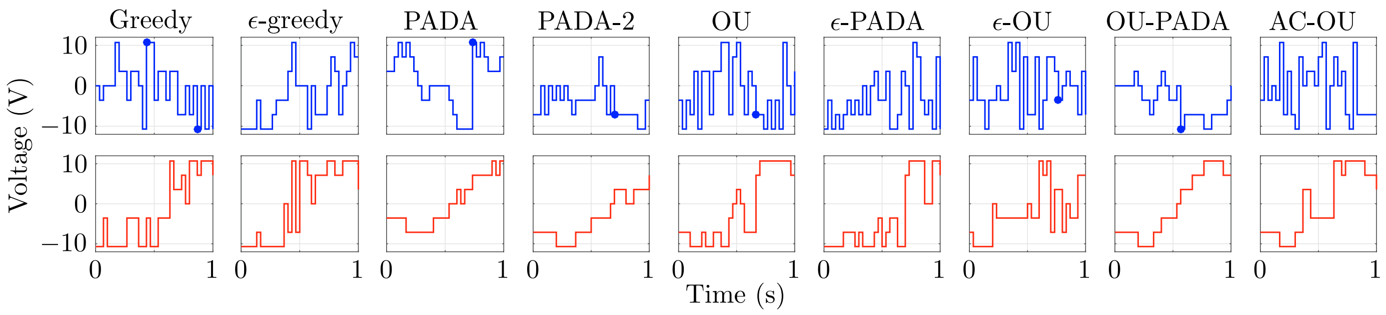

Fig. 3. Initial (*top*) and final (*bottom*) control signals. Solid dots (●) denote the beginnings of new episodes.

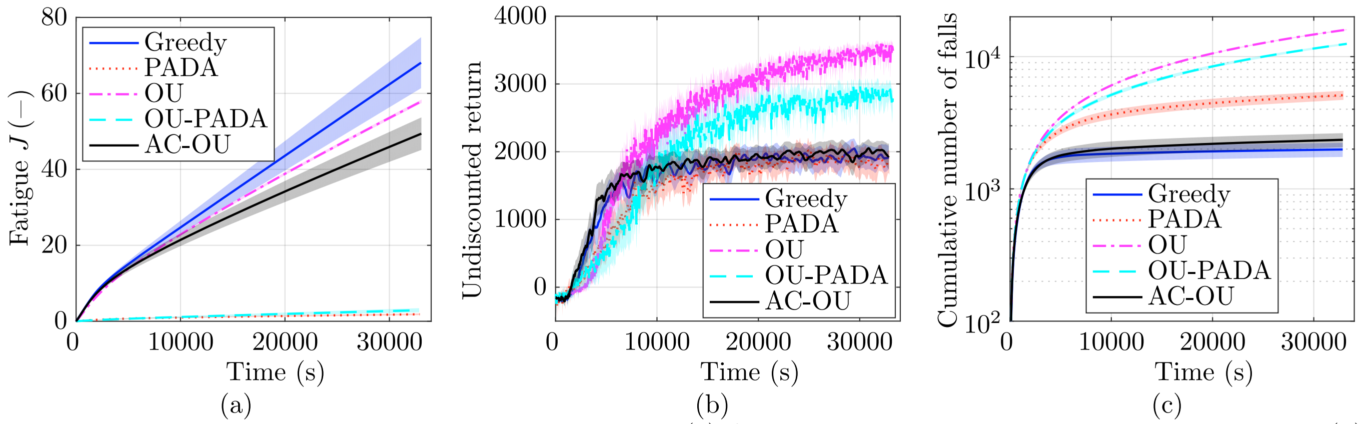

Fig. 4. During learning three benchmarks are calculated: (a) fatigue accumulated due to gear re-engagements, (b) undiscounted return, and (c) cumulative number of falls. Means with upper and lower 95% confidence limits are shown for 50 samples.

of fatigue and MTBF. Interestingly, AC-OU obtained the lowest MTBF among methods that did not constrain actions during the exploitation step.

For a clear overview of the results, we summarize them in Table 3. First, we note that none of the methods surpassed others in both fatigue and number of falls of the robot. This suggests that to minimize damage from both sources, a faster learning algorithm is required. In the context of exploration strategies, faster learning may be achieved by a problem-driven high-level guided exploration. Second, exploration based on time-correlated noise outperformed the ϵ-greedy method, therefore for actual experiments with a robot, the ϵ-greedy strategy is not advised. Finally, no definite conclusion can be drawn about which exploration method is better for a generic physical system. Nevertheless, some insight can be provided. If the falls are highly damaging, then either Greedy, ϵ-PADA, ϵ-OU or AC-OU should be used. On the other hand, if the robot can withstand falls, but the gear re-engagements are damaging, then PADA, PADA-2 or OU-PADA methods are advisable. This is the case for the robot Leo, whose gears are made of aluminum and can easily be damaged by random exploration. Gears made of hardened steel instead of aluminum are more robust against gear re-engagements. Thus, when the amount of damage induced by crashes is little, it would be practical to use OU or OU-PADA, as they achieve high performance.

Further reduction of falling or fatigue can be achieved by a time-dependent decay schedule applied to ϵ or σ. We expect that such strategies will only affect the benchmark results relatively, and our conclusions will still hold.

It is worth mentioning that in addition to the above factors, the damage depends on the configuration of the environment, the protection of the robot, the severity of contact impacts, and other factors. For example, visual observation of Leo's gait after learning with OU (Figure 5) exhibited high lifts of a swing leg, therefore large steps and presumably high damage due to higher swing leg velocities right before heel strikes, compared to ϵ-greedy. The figures of fatigue in Table 2 do not account for this source of damage. We expect that our future experiments with real Leo will unveil the contribution of the described factors to the total damage of the robot.

Finally, we note that there might not be a single supreme exploration strategy when controlling physical systems, but exploration can rather be system- and task-driven. Similar findings were made in neuroscience, where dynamic regulation of exploration strategies has been observed in human and animals. Wu et al. (2014) provide experimental support for the hypothesis that motor variability is centrally driven and is regulated according to the nature of the task.

5. CONCLUSION

In this article, we studied properties of several conventional and newly proposed action-selection methods in terms of their performance and the damage they cause to motor gears on the one hand and to the overall system on the other hand. We showed that none of the methods was capable of minimizing both sources of damage. Based on the quantitative comparison, we characterized conditions

Table 2. Mean and 95 % confidence interval of fatigue, MTBF, cumulative number of falls and undiscounted return obtained by each studied method averaged over 50 independent runs.

Method	Learning fatigue J	MTBF at start (in min)	Final MTBF (in min)	Cumulative # of falls	Return
Greedy	68.68 ± 34.00	4.38 ± 0.72	12.67 ± 8.89	$\mathbf{1984 \pm 241}$	1908 ± 193
ϵ-greedy	79.13 ± 21.18	4.41 ± 0.71	11.68 ± 4.66	3529 ± 351	2109 ± 122
PADA	$\mathbf{1.86 \pm 1.11}$	$\mathbf{338.73 \pm 250.86}$	$\mathbf{699.20 \pm 356.65}$	5099 ± 399	1824 ± 180
PADA-2	4.92 ± 1.00	73.33 ± 36.32	166.09 ± 57.21	2962 ± 206	1930 ± 150
OU	58.27 ± 2.27	5.19 ± 0.48	54.36 ± 27.30	15919 ± 144	$\mathbf{3501 \pm 110}$
ϵ-PADA	55.47 ± 22.08	4.40 ± 0.60	18.45 ± 10.85	2478 ± 294	2193 ± 129
ϵ-OU	63.24 ± 30.75	4.40 ± 0.68	15.77 ± 12.29	$\mathbf{2098 \pm 246}$	2012 ± 154
OU-PADA	$\mathbf{2.94 \pm 2.73}$	$\mathbf{377.67 \pm 295.50}$	$\mathbf{1292.41 \pm 855.44}$	12435 ± 227	2811 ± 174
AC-OU	49.73 ± 21.66	4.38 ± 0.70	21.07 ± 10.87	2348 ± 288	1951 ± 176

Table 3. A simplified overview of benchmark performances of action-selection methods.

Method	Minimizes gear re-engagements	Minimizes cumulative number of falls	Maximizes return
Greedy	−	+	−
ϵ-greedy	−	+/−	+/−
PADA	+	−	−
PADA-2	+	+/−	−
OU	+/−	−	+
ϵ-PADA	+/−	+	+/−
ϵ-OU	+/−	+	+/−
OU-PADA	+	−	+
AC-OU	+/−	+	−

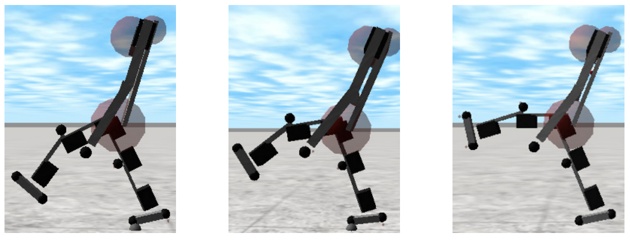
$0.66 \pm 0.16\,\mathrm{m\,s^{-1}}$
$0.68 \pm 0.24\,\mathrm{m\,s^{-1}}$
$0.87 \pm 0.22\,\mathrm{m\,s^{-1}}$

Fig. 5. Maximum raise of the swing leg after learning with ϵ-greedy (*left*), ϵ-OU (*middle*) and with OU (*right*). Swing leg velocities and standard deviations right before heel strikes obtained after five independent learning runs are shown below each picture.

required for the selection of a certain method for learning in a physical system. Results indicate that uniform exploration, commonly achieved by the well-known ϵ-greedy exploration method, was not a good choice for learning on a physical robot. Our simulation results demonstrated that exploration based on the time-correlated noise (ϵ-OU) achieved similar performance and fatigue levels, but additionally it reduced the number of falls of the robot. In contrast, limiting the action set (OU-PADA) resulted in better performance and much less fatigue, but a larger number of falls.

REFERENCES

Albus, J.S. (1975). A new approach to manipulator control: the cerebellar model articulation controller (CMAC). *J. Dyn. Sys., Meas., Control*, 97(3), 220–227.

Garcia, J. and Fernandez, F. (2015). A Comprehensive Survey on Safe Reinforcement Learning. *Journal of Machine Learning Research*, 16, 1437–1480.

Gehring, C. and Precup, D. (2013). Smart exploration in reinforcement learning using absolute temporal difference errors. In *Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, 1037–1044.

Ha, S. and Liu, C.K. (2015). Multiple Contact Planning for Minimizing Damage of Humanoid Falls. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

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

Levine, S. and Koltun, V. (2013). Guided Policy Search. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Matignon, L., Laurent, G., and Le Fort-Piat, N. (2006). Reward Function and Initial Values: Better Choices for Accelerated Goal-Directed Reinforcement Learning. volume 4131 of *Lecture Notes in Computer Science*, 840–849.

Meijdam, H. (2013). Learning while preventing mechanical failure due to random motions. Master's thesis, Delft University of Technology.

Moldovan, T.M. and Abbeel, P. (2012). Safe exploration in markov decision processes. In *Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML)*, 1451–1458.

Schuitema, E. (2012). *Reinforcement Learning on autonomous humanoid robots*. Ph.D. thesis, TU Delft.

Schulman, J., Levine, S., Abbeel, P., Jordan, M.I., and Moritz, P. (2015). Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 1889–1897.

Sutton, R.S. and Barto, A.G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Wu, H.G., Miyamoto, Y.R., Castro, L.N.G., Olveczky, B.P., and Smith, M.A. (2014). Temporal structure of motor variability is dynamically regulated and predicts motor learning ability. *Nature Neuroscience*, 17(2), 312–321.