2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Reinforcement Learning Control for Biped Robot Walking on Uneven Surfaces

Shouyi Wang, Jelmer Braaksma, Robert Babuška
Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft, the Netherlands

Daan Hobbelen
Delft Biorobotics Laboratory
Delft University of Technology
Mekelweg 2, 2628 CD Delft, the Netherlands

Abstract— **Biped robots based on the concept of (passive) dynamic walking are far simpler than the traditional fully controlled walking robots, while achieving a more natural gait and consuming less energy. However, lightly actuated dynamic walking robots, which rely on the natural limit cycle of their mechanical structure, are very sensitive to ground disturbances. Already a very small step down can cause the robot to lose stability. In this paper, we investigate the use of reinforcement learning to make a dynamic walking robot more robust against ground disturbances. The learning controller is applied to a simulated twolink biped which is an abstraction of a mechanical prototype developed at the Delft Biorobotics Laboratory. The learning controller has been designed such that it can be applied as a straightforward extension of the proportionalderivative (PD) controller currently used to drive the robot's pneumatic actuators. The learning controller is therefore suitable for the future implementation in the robot hardware. Simulation results demonstrate that the biped quickly learns to overcome stepdown disturbances on the floor up to 10% of the leg length, without compromising the natural walking style provided by the PD controller, which was optimized for walking on an even surface.**

I. Introduction

Since McGeer introduced the concept of passive dynamic walking [1], a number of energyefficient and naturally walk ing machines have been built. However, the stability and robustness of this kind of biped robots are still not satisfactory. This is due to the inherent properties of dynamic walking, which exploits the natural limit cycle of the kinematic struc ture, as opposed to trajectorybased humanoid robots (such as the Asimo robot by Honda).

To enhance the robustness of dynamic walking, learning control methods have been investigated with some promising results [2], [3], [4]. Two types of learning are commonly used: supervised learning and reinforcement learning. Imitation is an example of a supervised learning method commonly applied to humanoid robots in order to achieve the desired motion pattern. Nakanishi et. al [5] used a humandemonstrated tra jectory to train a 5link planar biped robot walking on a flat surface. The main drawback of imitation learning is that it is difficult to find proper target trajectories, mainly because of the mechanical limitations and the potentially large difference between simulations and the real robot.

Another type of supervised learning for biped walking is based on the Zero Moment Point (ZMP) concept [6]. ZMP controllers are popular because they can ensure stability over a large range of conditions. The robot learns to follow a predefined ZMP trajectory during walking. However, ZMP based methods require accurate mathematical models and a suitable trajectory for each joint of the biped. Their main drawback is that the resulting gait is energy inefficient and appears unnatural.

Reinforcement learning (RL) algorithms have also been studied in the context of bipedal walking. They have two main advantages over supervised learning: 1) The algorithm can automatically learn a good control solution, without providing an accurate robot model. 2) The learning process can continue in order to account for changes in the robot dynamics or terrain properties. The control performance therefore improves with increasing experience. Several directions can be distinguished within the RL control paradigm:

1) Some walking robots use Central Pattern Generators (CPG) to generate walking gaits. A CPG controller is often realized as a recurrent neural network (RNN) whose weights are optimized by RL [7].
2) Fuzzy control based on RL for bipedal walking has been studied in [8], [9]. Initial fuzzy rules were designed to capture intuitive knowledge on human walking. Dy namic walking experience is then incorporated through RL. Reinforcement learning has also been used to tune parameters of other manually designed controllers [10].
3) Tedrake [4] applied a policygradientbased reinforce ment learning algorithm to a 3D biped, starting with a passive walker without any control. The robot learned to walk on a flat surface in about 20 minutes.

Although dynamically walking bipeds have been extensively studied, most of the results only apply to walking on an even surface. The main contribution of this paper is the development of a RLbased control scheme for a dynamic biped walking on an uneven surface. The RL controller has been designed to enhance the disturbance rejection properties of a well tuned PD controller, without compromising the natural gait or excessively increasing the energy consumption.

The remainder of this paper is structured as follows. In Section II, we introduce the considered biped and the PD controller. Section III details the structure of the learning controller and the RL algorithm. In Section IV, the simulation

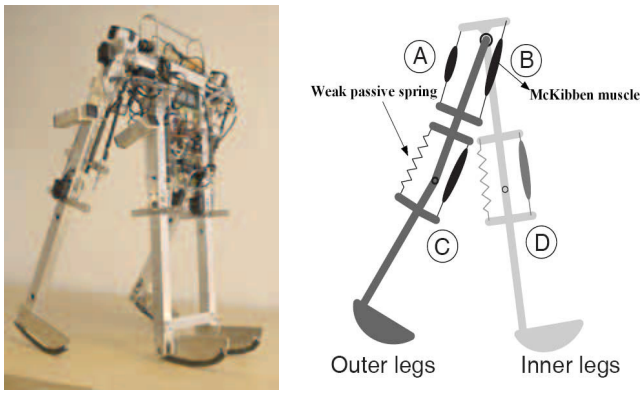
results are reported and Section V concludes the paper.

II. PDC ONTROLLED DYNAMIC WALKER

Passive dynamic walkers can walk in a smooth and stable manner down a shallow slope without any actuation or control. They exhibit a steady walking pattern (limit cycle) which is energy efficient and natural. Walking on a flat surface can be achieved by properly timed light actuation of the hip or ankle joints [11], [12].

A. Mechanical Prototype

In this study, we consider the 2D bipedal walker 'Mike' shown in Fig. 1a. It is powered by lightweight pneumatic actuators (McKibben muscles) at the hip. The hip joint is actuated by an antagonistic pair of muscles which generate the desired joint torque. Each knee is extended by one McKibben muscle which is counteracted by a weak passive spring, see Fig. 1b. There is no ankle actuation.



(a) The prototype. (b) Structure schematic.

Fig. 1. 'Mike' is a 2D dynamic walking robot with pneumatic actuation. The control system switches the equilibrium of the McKibben muscles according to the foot contact signal [12].

The dynamic simulation model is a twodimensional biped model with two rigid legs and curved feet (the knees are not simulated). It is implemented in MATLAB, using the following parameter values: leg length 0.4 m, foot radius 0.1 m, leg mass 1.0 kg and leg inertia 0.01 kgm^2.

B. PD Controller

The hip actuators are controlled by a PD controller, which is represented by the spring and damper system in the mechanical scheme of Fig. 2. The spring generates hip torque and drives the swing leg towards the desired relative angle ϕ_d.

The hip torque τ is generated according to the following proportionalderivative control law:

$$\tau(t) = k_p\big(\phi_d - \phi_r(t)\big) - k_d\dot{\phi}_r(t) \qquad (1)$$

where ϕ_r is the relative angle between the two legs (which can be easily measured by a potentiometer or encoder), k_p, k_d are the PD parameters and ϕ_d is the desired relative angle between the legs. Once the current swing leg touches the ground and becomes the new stance leg, the sign of ϕ_d is inverted and the

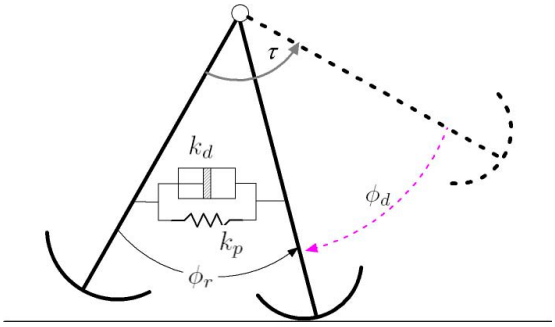


Fig. 2. The PDcontrolled biped. The spring and damper system represents the PD controller, which drives the swing leg towards the desired relative angle ϕ_d indicated by the dashed line.

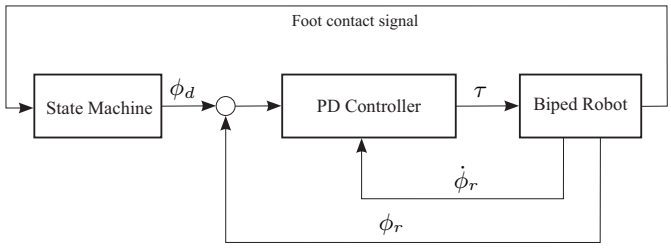


Fig. 3. The PD control scheme of the biped robot.

controller drives the new swing leg forward. The PD control scheme is shown in Fig. 3.

Tuning the PD parameters is a nontrivial exercise; on an even surface, a wide range of different gaits can be obtained within the controller's stability region. Figure 4 shows the stable ranges of k_p and k_d for three different desired angles ϕ_d: 0.3 rad, 0.5 rad and 1.0 rad.

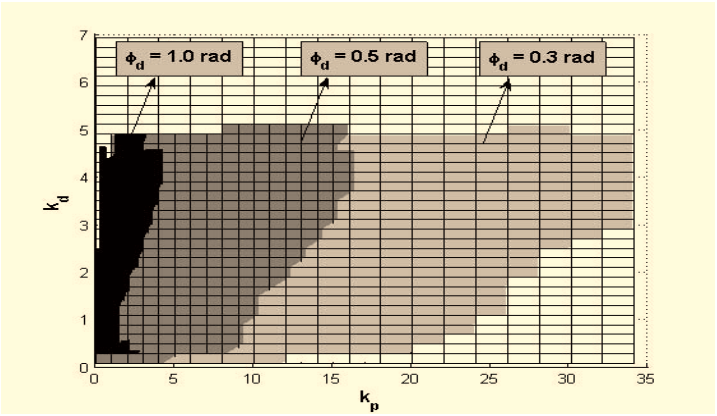


Fig. 4. Stable ranges for k_p, k_d with ϕ_d=0.30 rad, 0.50 rad and 1.0 rad.

The PD parameters were finetuned within the stable range for $\phi_d = 0.5$ rad, which gave the most natural gait. The fine tuning was based on a cost function that makes a tradeoff between the walking speed and the energy cost per unit distance [11]. During this tuning, we found that the PD controller cannot achieve natural and energyefficient walking simultaneously with strong disturbance rejection properties.

The learning controller described in next section is designed to improve the rejection of ground disturbances.

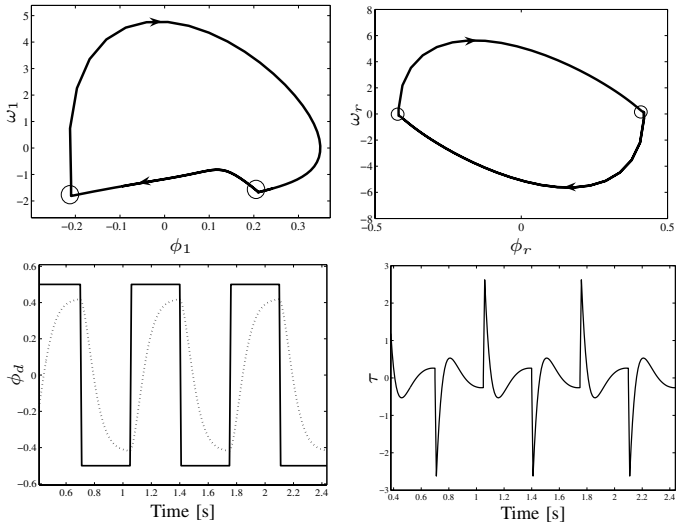


Fig. 5. The walking limit cycle generated by the optimized PD controller. Topleft: phase plot of leg 1 angle; topright phase plot of relative angle; bottomleft: ϕ_d (solid) and ϕ_r (dotted); bottomright: torque τ.

Figure 5 shows the walking cycle generated by the opti mized PD controller. The circles in the upper two phase plots indicate the foot contact instants. The angles are denoted by ϕ and the angular velocities by ω.

III. Learning Control Scheme

In order to select an appropriate signal as an input of the learning controller, the effect of ground disturbances on the limit cycle has been thoroughly analyzed. We found that the difference ω_{dv} between the measured angular velocity of the stance leg ω_{sto} (measured right after the foot contact) and its nominal value ω^*_{sto} on an even surface carries enough information on the deviation from the nominal walking cycle. The angular velocity difference ω_{dv} has therefore been selected as the input of the learning agent.

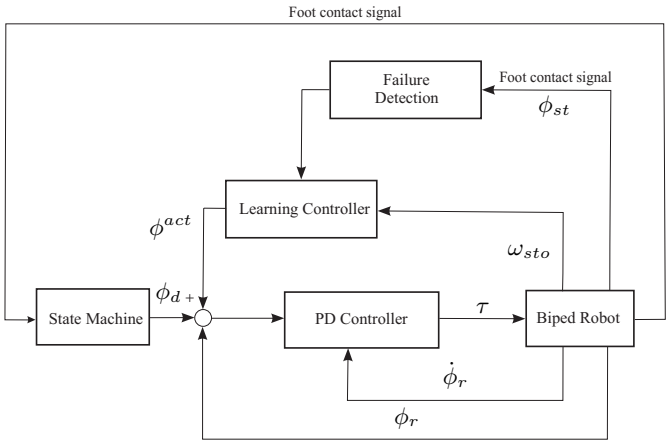


Fig. 6. The learning control scheme of the biped robot.

The output of the learning controller ϕ^{act} is a short pulse added to the PD controller's setpoint ϕ_d (see Fig. 6). This results in extra hip torque to move the swing leg more forward and prevent tripping. This simple, oneinput, oneoutput con trol structure facilitates quick learning convergence and allows future realtime implementation in the robot hardware. The purpose of the failuredetection block in Fig. 6 is to monitor in simulations the foot contact and the angle of the stance leg ϕ^{act} in order to detect whether walking failed. A failure means that the robot fell either backward or forward or that it started running (no foot in contact with ground). This information is used to compute the rewards, as shown in Fig. 7 and further detailed in Section IIIB.

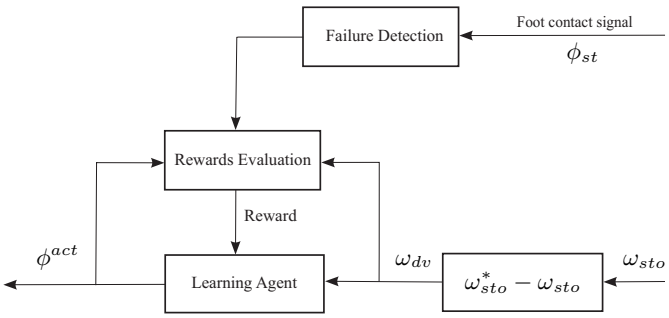


Fig. 7. Structure of the learning controller.

A. The Reinforcement Learning Algorithm

Reinforcement learning is suitable for online adaptation through interaction with the process. At each time step, the learning agent receives information on the process' state, the reward (indicating a success or failure of the control task) and chooses an action to perform. The goal of RL is to map the states to optimal actions such that the expected future reward is maximized [13]. In this work, we adopt the Sarsa(λ) algorithm [14], [15], [13], which is an onpolicy temporal difference (TD) algorithm, briefly outlined below.

Denote by s the system's state and by a the control action; $Q(s,a)$ is the estimated stateaction value (Qvalue). The Q value is stored in a Qtable, which is a lookup table indexed by discrete stateaction pairs (continuous states and actions must be discretized). The expected future reward under the current control policy π is defined as:

$$Q_\pi(s_k,a_k) = E\left(\sum_{n=0}^{\infty}\gamma^k r_{k+n+1}\right), \text{ with } 0<\gamma<1 \quad (2)$$

with E the expectation operator, r_{k+n+1} the future rewards, and γ a discount factor in the range of $(0,1)$. The optimal Qvalue function (2) cannot be computed, but it can be approximated through the following iterative updates:

$$Q_{k+1}(s_k,a_k) = Q_k(s_k,a_k) + \alpha\Big(r_{k+1} + \gamma Q_k(s_{k+1},a_{k+1}) - Q_k(s_k,a_k)\Big) \quad (3)$$

where r_{k+1} is the reward received for the transition from state s_k to s_{k+1} as a result of action a_k and $\alpha \in (0,1]$ is the

learning rate. Note that this basic 'Sarsa' (stateactionreward stateaction) rule only updates the Qvalues corresponding to the most recently visited state. As this typically leads to long learning times in tasks with delayed reward, eligibility traces are employed to speed up the learning process. The eligibility trace is a memory variable associated with each state, which decays in time. The basic idea behind the eligibility trace is that earlier stateaction pairs are given less credit for the current reward. The decay factor of the trace is denoted by λ, hence the term Sarsa(λ) algorithm.

The control policy is obtained by selecting the action corresponding to the largest Qvalue for the given state:

$$\pi(s) = \arg\max_a Q(s,a) \tag{4}$$

This greedy policy is optimal in the case that the Qvalues converged to the optimal ones. In order to discover an optimal control strategy, the RL agent must explore its environment. This means that it must also select other actions than dictated by the greedy policy. We use the MaxBoltzmann exploration strategy [16], which explores with probability ϵ using the 'Boltzmann selection' and takes the optimal action with prob ability $1-\epsilon$. The learning algorithm is summarized in the following table:

Set learning rate α, discount factor γ, eligibility trace decay factor λ, ϵgreedy exploration rate ϵ, Boltzmann exploration constant T.
Initialize Q(ω_{dv}, ϕ^{act}) $\leftarrow 0$, $\forall\ \omega_{dv}$, ϕ^{act}
Repeat (for each trial):
  e $\leftarrow 0$ (clear the eligibility traces)
  Initialize the biped to a stable gait
  Repeat (for each step):
    Take action ϕ_k^{act}, observe r_{k+1}, $\omega_{dv,k+1}$
    Choose ϕ_{k+1}^{act} on the basis of $\omega_{dv,k+1}$, compute :
    $\delta_k = r_k + \gamma Q(\omega_{dv,k+1}, \phi_{k+1}^{act}) - Q(\omega_{dv,k}, \phi_k^{act})$
    Update the replacing eligibility traces:
      $e(\omega_{dv}, \phi^{act}) \leftarrow \gamma\lambda e(\omega_{dv}, \phi^{act})\ \ \forall\ \omega_{dv}, \phi^{act}$
      $e(\omega_{dv,k}, \phi_k^{act}) \leftarrow 1$
    Update the Qvalue $\forall\ \omega_{dv}, \phi^{act}$:
      $Q(\omega_{dv}, \phi^{act}) \leftarrow Q(\omega_{dv}, \phi^{act}) + \alpha\delta_k e(\omega_{dv}, \phi^{act})$
  Until trial terminated.
Until maximum number of trials is reached.

B. Reward Function for the Biped

The RL algorithm is applied on a stepbystep basis, which means that the sampling period of the learning agent equals to the period of one step. Moreover, the learning agent is only triggered when $\omega_{dv} > 0.1$ rad/s. If the biped successfully makes a step forward, the agent receives a small positive reward, otherwise, a large negative reward is given. More specifically, the reward after each step is defined by:

$$r_k = \begin{cases} 5 - \phi_{k-1}^{act} - \omega_{dv,k}, & \text{succesful step forward} \\ -100, & \text{otherwise} \end{cases} \tag{5}$$

Here ϕ_{k-1}^{act} is the action taken at the previous step. The $-\omega_{dv,k}$ term encourages ω_{dv} to converge to zero. The $-\phi_k^{act}$ term encourages the learning agent to take energyefficient actions: the smaller the magnitude of ϕ_k^{act}, the higher the reward r_k. The constant 5 makes the reward positive (the maximum action $\phi_{max}^{act} = 2$ rad/s, and the maximum value of the state $\omega_{dv,max} = 2$ rad/s).

The learning trial is terminated when the biped either falls or when it manages to successfully walk 10 steps with $|\omega_{dv}| < 0.1$ rad (this is regarded as convergence to the nominal walking cycle). A series of several trials (e.g. 20) is called an experiment. Within an experiment, the walking performance will typically improve. For the sake of reliable statistical evaluation, an experiment is repeated several times.

The continuous input ω_{dv} is discretized by using interval boundaries $\{-2.0, -1.9, \ldots, 2.0\}$ and the control action ϕ^{act} takes on discrete values from the set $\{0, 0.1, \ldots, 2\}$. The learning rate was selected as $\alpha = 0.5$, the eligibility traces decay factor $\lambda = 0.5$, and the constant of Boltzmann selection $T = 10$. The exploration probability ϵ is set to 0.3 at the beginning of learning and reset to zero after 15 trials in order to switch off exploration and evaluate the strategy learned.

IV. Simulation Results and Discussion

A. Performance Evaluation

The *average reward per step* is used as a performance criterion:

$$\bar{r}_a = \frac{1}{N}\sum_{k=1}^{N} r_k \tag{6}$$

where N is the total number of steps within one trial. As the exploration involves randomness, the evaluation must be based on statistical analysis of several learning experiments. We take the mean of $\bar{r}_a$ over 20 experiments:

$$P = \frac{1}{20}\sum_{i=1}^{20} \bar{r}_a^i \tag{7}$$

Higher values of P represent better performance.

B. Fixed StepDown Disturbance

First, only a single step of 36 mm down is considered. The learning result over 20 experiments is shown in Fig. 8. Each of the experiments consists of 20 trials. At the start of learning, the performance index P is negative, the biped falls nearly each time after the disturbance occurs. As the number of trials increases, the biped gains more experience and learns to overcome the disturbance. After the 15th trial, at which the exploration is switched off, the performance index stays positive, which implies that the biped is able to overcome the disturbance every time.

C. Random StepDown Disturbances

The setting is the same as above, however, in each trial, the height of the step is randomly chosen from the set $\{12,\ 16,\ 20,\ 24,\ 28,\ 32,\ 36,\ 40\}$ mm, which corresponds to the range 3 to 10% of the leg length (40 cm). Note that the PD controller without learning is able to overcome only the 12 mm stepdown disturbance. Again, 20 experiments are run, now each experiment consists of 150 trials. The parameters are the same as above, the exploration probability is reset to zero after 130 trials. Figure 9 shows the evolution of the

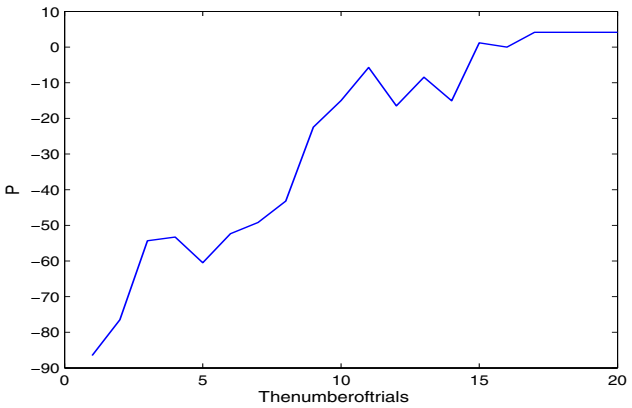


Fig. 8. Performance index P for a 36 mm stepdown disturbance, averaged over 20 experiments.

performance index P over the trials (again averaged over 20 experiments).

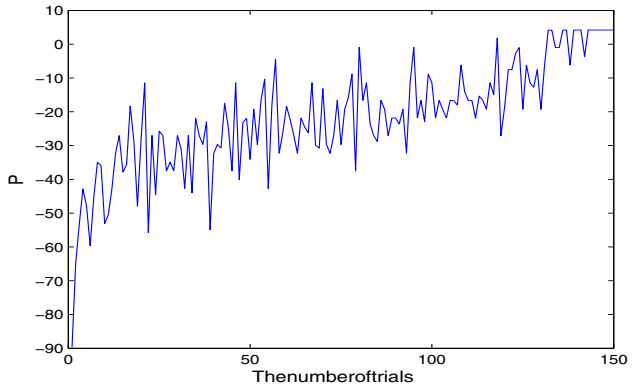


Fig. 9. The performance index P for a random stepdown disturbance, averaged over 20 experiments.

D. Performance in a Test Scenario

In order to test the robustness of the learning controller, a test scenario with 16 steps was designed. The height of the steps monotonically increases from 12 mm to 40 mm with a 2 mm interval. The first step occurs at $t = 2$ s, and the subsequent 13 steps after every two seconds.

If ω_{dv} is smaller than 0.1 rad/s for 10 successive steps, the bipedal gait is then considered as having converged to the original stable limit cycle. In the test scenario, the exploration is turned off (by setting $\epsilon = 0$). The agent uses the control policy it has learned during one of the experiments with 150 trials, as described above.

The foot contact points with the floor are marked by asterisks in Fig. 10. The feedback state ω_{dv} that the agent receives in the test scenario is shown in Fig. 11. The learning controller adaptively tunes the setpoint ϕ_d, as shown in Fig. 12.

These results demonstrate that the learned control policy helps the biped overcome all the stepdown disturbances of

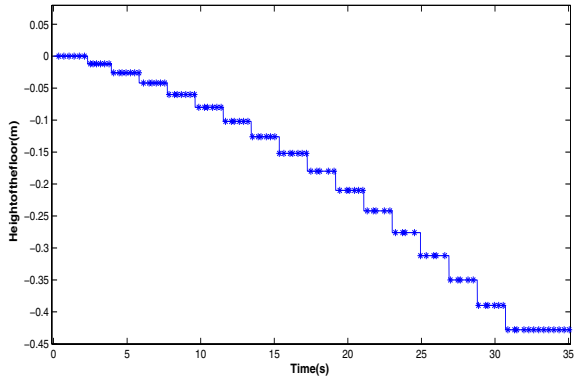


Fig. 10. Foot contacts during the test scenario 16 steps.

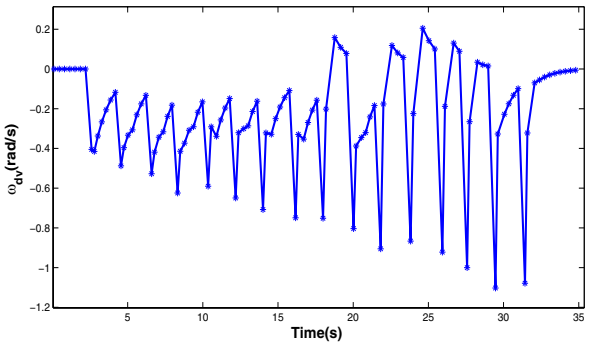


Fig. 11. The deviation ω_{dv} during the test scenario. The '$*$' markers denote the discretized input of the learning agent.

the test scenario, in which the maximum step height is 4 cm (about 10% of the leg length). The proposed learning algorithm is effective in making the biped robust against stepdown disturbances on the floor.

V. Conclusions

We proposed a RLbased controller to help a dynamic walking robot overcome stepdown disturbances on the walking surface. The learning agent learns to apply additional hip torque to adapt the gait after a stepdown disturbance occurs. The effectiveness of the learning algorithm was demonstrated by two simulation experiments. In the first experiment, the agent learned a control policy to overcome a single step down on the floor within only 20 trials. In the second experiment, the learning agent learned to overcome a random height stepdown within 150 trials. Although the biped was trained by only one step in each trial, it is able to walk through a test scenario with 16 different stepdown disturbances. This demonstrates the good generalization properties of the learning controller. The learned control policy is also effective in the presence of multiple subsequent disturbances. The Sarsa(λ) algorithm proved to be suitable for this task. Our future research will include an extension to a more complex simulation model, RL based on function approximation techniques and finally

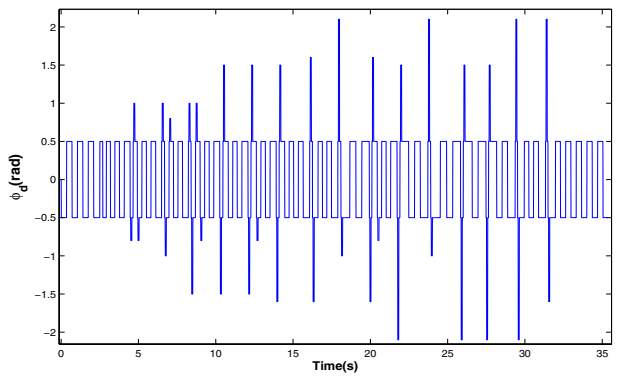


Fig. 12. The adaptively modified setpoint ϕ_d during the test scenario.

an implementation in the robot hardware.

REFERENCES

[1] T. McGeer, "Passive dynamic walking," *International Journal of Robotics Research*, vol. 2, pp. 62–82, 1990.

[2] H. Benbrahim and J. Franklin, "Biped dynamic walking using reinforce ment learning," *Robotics and Autonomous Systems*, vol. 22, pp. 283–302, 1997.

[3] J. Morimoto, J. Nakanishi, G. Endo, G. Cheng, C. Atkeson, and G. Zeglin, "Poincarémapbased reinforcement learning for biped walk ing," in *Proceedings IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 2005, pp. 2381–2386.

[4] R. L. Tedrake, "Applied optimal control for dynamically stable legged locomotion," PhD Dissertation, Massachusetts Institute of Technology, 2004.

[5] J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, vol. 47, pp. 79–91, 2004.

[6] M. Vukobratović, "Zeromomentpoint – thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

[7] T. Mori, Y. Nakamura, M. aki Sato, and S. Ishii, "Reinforcement learning for CPGdriven biped robot," in *Proceedings Nineteenth National Con ference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence*, San Jose, USA, July 2004, pp. 623–630.

[8] C. Zhou, "Robot learning with GAbased fuzzy reinforcement learning agents," *Information Sciences*, vol. 145, no. 12, pp. 45–68, 2002.

[9] C. Zhou and Q. Meng, "Dynamic balance of a biped robot using fuzzy reinforcement learning agents," *Fuzzy Sets and Systems*, vol. 134, no. 1, pp. 169–187, 2003.

[10] C.M. Chew and G. A. Pratt, "A general control architecture for dynamic bipedal walking," in *Proceedings 2000 IEEE International Conference on Robotics & Automation*, San Francisco, USA, Apr. 2000, pp. 3989–3995.

[11] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passivedynamic walkers." *Science*, vol. 307, pp. 1082–1085, 2005.

[12] M. Wisse, "Essentials of dynamic walking: analysis and design of two legged robots," PhD Dissertation, Delft University of Technology, Delft, the Netherlands, 2004.

[13] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[14] G. Rummery and M. Niranjan, "Online Qlearning using connectionist systems," Cambridge University, Engineering Department, Tech. Rep., Oct 1994.

[15] S. P. Singh and R. S. Sutton, "Reinforcement learning with replacing eligibility traces," *Machine Learning*, vol. 22, pp. 123–158, 1996.

[16] A. Kølle, "The nature of learning – a study of reinforcement learning methodology," PhD Dissertation, University of Copenhagen, Denmark, 2003.