BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **12**, No 3

Sofia • 2012

On Global Optimization of Walking Gaits for the Compliant Humanoid Robot, COMAN Using Reinforcement Learning

Houman Dallali, Petar Kormushev, Zhibin Li, Darwin Caldwell

Department of Advanced Robotics, Istituto Italiano di Tecnologia, via Morego 30, 16163 Genova
Emails: houman.dallali@iit.it petar.kormushev@iit.it zhibin.li@iit.it Darwin.caldwell@iit.it

Abstract: *In ZMP trajectory generation using simple models, often a considerable amount of trials and errors are involved to obtain locally stable gaits by manually tuning the gait parameters. In this paper a 15 degrees of Freedom dynamic model of a compliant humanoid robot is used, combined with reinforcement learning to perform global search in the parameter space to produce stable gaits. It is shown that for a given speed, multiple sets of parameters, namely step sizes and lateral sways, are obtained by the learning algorithm which can lead to stable walking. The resulting set of gaits can be further studied in terms of parameter sensitivity and also to include additional optimization criteria to narrow down the chosen walking trajectories for the humanoid robot.*

Keywords: *Humanoid robot walking, compliance, reinforcement learning.*

1. Introduction

Walking trajectory generation for a humanoid robot is a challenging control problem. Humanoid robots have many Degrees of Freedom (DoF), with unstable, nonlinear and underactuated dynamics. Moreover, humanoid robots need to adapt their walking gait based on the environment, for instance to avoid an obstacle the robot needs to adapt its walking gait to take a different step size while keeping the same average speed.

Due to the complexity of the walking model several studies in the literature have approached the walking problem using machine learning techniques [1]. For example, in [2] a model-based reinforcement learning method was used for bipedal walking on a planar 5 DoF robot fixed to a rotating boom. Learning was applied to learn the Poincare return map of the bipedal robot. The learning algorithm was used

to minimize the torques while keeping a certain height to avoid falling. In [3] Central Pattern Generators (CPG) for QRIO humanoid were tuned using a policy gradient method and stable gaits were shown in simulation and in experiments. The robot's pelvis states were used to describe the motion of the robot. In [4] a stochastic policy gradient reinforcement learning was applied to a toy robot with a carefully designed passive mechanical dynamics and resulted in stable walking mainly due to the passive dynamics design. In [5] a map was constructed offline for a given trajectory to verify the feasibility of an step size. The set of all possible steps was considered to be a 6 dimensional space. The set was constrained to have a one to one correspondence between each trajectory and step size. Geometric and Zero Moment Point (ZMP) constraints where used to verify the feasibility of the desired walking step. In [6] reinforcement learning was used on a compliant bipedal legs to reduce the electrical energy consumption by changing the centre of mass of the robot during walking.

In terms of walking models, often simplified ones such as the inverted pendulum model [7-8] or the compass gait model [9] are used in trajectory generation and stability analysis while detailed and accurate models are used in simulation studies.

In addition to use of centre of mass, limit cycle criterion and capture point, one of the most commonly used stability criteria for humanoid walking is Zero Moment Point (ZMP) which is often formulated as a closed form solution of the linear inverted pendulum model of walking [10-11]. In a dynamically stable gait ZMP of a robot is the same as the centre of pressure.

Despite using simplified models for walking trajectory generation, a considerable effort in manual tuning is needed before the generated trajectory can be applied in practice on an actual humanoid robot. In order to address this problem, in this study an automatic way of tuning a walking gait using reinforcement learning methods is investigated. The goal of learning is to find dynamically stable gait parameters for a desired walking speed. In summary, reinforcement learning varies the walking gait parameters such as the step size and lateral sway amplitude to generate walking trajectories using the inverted pendulum model [8]. The produced gaits are verified in terms of stability walking speed using dynamic simulation of the compliant humanoid robot with 15 DoF. The simulation output, i.e. multibody stability and achieved walking speed are evaluated as a reward function for the learning algorithm to better tune the gait parameters. Applying this method on the robot is time consuming and can be risky, but in simulation we can obtain many stable solutions that can be further narrowed down using additional criteria such as energy efficiency. Hence, using the simulator the gaits are tested in advance, which creates a more reliable and feasible gait for the experiments.

In this paper, instead of using state-action based reinforcement learning which suffers from the curse of dimensionality, direct policy search reinforcement learning is used. This method works in a low dimensional policy space which bypasses the dimensionality problem of the state-action space. The contributions of this study are twofold. Firstly, the method performs global search in the walking gait parameter

space to yield multiple alternative solutions which can then be narrowed down using further optimization criteria such as energy. Secondly, it investigates the importance of different parameters on the walking gait.

This paper is organized as follows. In Section 2, the mechanical overview of COMAN humanoid robot and the dynamic walking simulator are described. In Section 3 the trajectory generation method is briefly described. The reinforcement learning algorithm is described in Section 4. The results of using this algorithm are presented in Section 5, and finally the conclusions and future work are briefly discussed.

2. Dynamic model of the humanoid robot COMAN

In this section, first, the mechanical description of the compliant humanoid robot COMAN is given. Next, an overview of the dynamic walking simulator and its features are discussed.

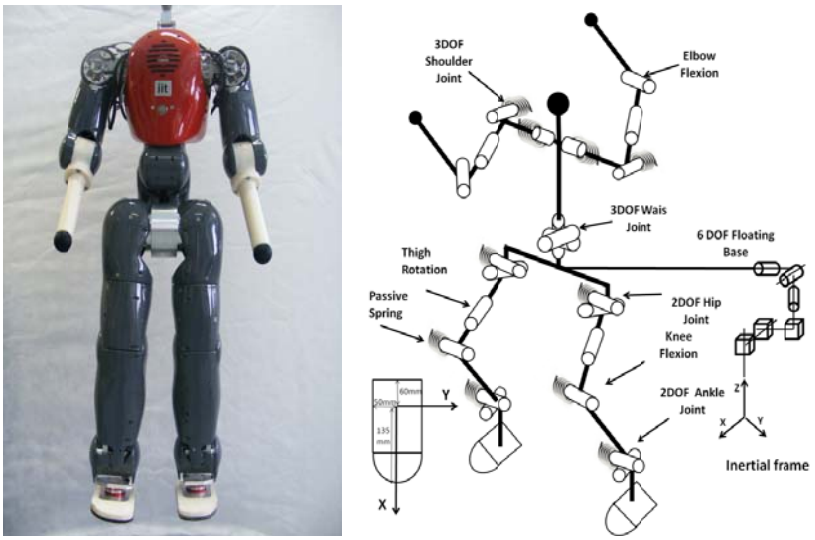
2.1. Overview of the mechanical model

COMAN (stands for COmpliant huMANoid) is powered by series elastic actuators and is being developed within the AMARSI European project [12] at the Italian Institute of Technology (IIT) as a derivative of the original iCub, and cCub [13] which added passive compliance in the major joints of the legs (see Fig. 1 (a)). The use of passive compliance provides shock protection, robust locomotion, safer interaction and potentially energy efficient locomotion. Currently, COMAN has 23 DoF, with passive compliance in the pitch joints in the legs, the waist, and the pitch and roll shoulder joints. In addition, COMAN uses brushless DC motors and harmonic drives, which are modeled in the dynamic simulator described in Subsection 2.2. In terms of control software architecture, currently COMAN uses a decentralized PID control architecture. Further details about the first prototype of COMAN, cCub are available in [13] with the major kinematic difference being the addition of passive compliance in the hip pitch and the orders of the ankle and the waist pitch and roll joints being swapped.

The coupled mechanical model of the robot is described in (1), where *M, C* and *G* are mass-inertia, Coriolis and gravity matrices. q, $\dot{q}$ and $\ddot{q}$ are positions, velocities and accelerations of all joints in vector form. Similarly, q_m, $\dot{q}_m$ and $\ddot{q}_m$ are the positions, velocities and accelerations of the motors. J and B_m are the motors' inertia and damping matrices. B_s and K_s are the passive compliance damping and stiffness matrices. τ_m is the motors' torque expressed in vector form,

$$\begin{cases} M\ddot{q} + C\dot{q} + Gq = B_s\left(\dot{q}_m - \dot{q}\right) + K_s\left(q_m - q\right) \\ J\ddot{q}_m + B_m\dot{q}_m + B_s\left(\dot{q}_m - \dot{q}\right) + K_s\left(q_m - q\right) = \tau_m \end{cases}. \quad (1)$$

In the next section, the dynamic walking simulator under Matlab is described.



(a) (b)

Fig. 1. COMAN humanoid robot (a) and the mechanical diagram with a floating base used to model the walking simulator (b)

2.2. Dynamic walking simulator

The dynamic walking simulator used in this paper considers 15 DoF of COMAN (legs and waist) and it is based on the floating base representation of legged robots as proposed in the literature [14-16]. A floating base is represented by additional six DoF attached between the humanoid robot and the world inertial frame to describe the free motion of the robot with respect to the inertial frame, as shown in Fig. 1 (b). The 6DoF consist of 3 translational joints and 3 rotational joints about the XYZ axis. This formalism unifies all the phases of legged locomotion, including a single support, double support and flight phase, as well as the fallen phase into one single model, and simplifies the simulation models by removing the switches between various phases of walking.

Since the floating base joints are not actuated, a realistic ground model must be introduced to simulate the robots balance and locomotion. The ground model is defined in the simulator using linear springs and dampers with limited friction to allow slipping on the floor. The impacts with the ground are modeled as compliant impacts as opposed to rigid body impacts. The tangential force of the ground is modeled in (2), where μ is the friction coefficient, and Δx is the deflection from the first point of contact. The normal force is given in (3), where F_z is the vertical ground reaction force, $\Delta z = z - z_0$ denotes the penetration in the ground which is always positive,

(2)
$$F_x = \begin{cases} -K\Delta x - D\Delta \dot{x}, & (-K\Delta x - D\Delta \dot{x}) \le \mu F_z \\ -\mathrm{sgn}(\Delta \dot{x})\,\mu F_z, & (-K\Delta x - D\Delta \dot{x}) > \mu F_z \end{cases},$$

(3) $$F_z = \begin{cases} 0 & z \geq z_0 \\ -K_G \Delta z - D_G \Delta \dot{z} & z < z_0 \end{cases}.$$

There are four points at each corner of the foot where the ground models are introduced as external forces as shown in Fig. 2. The parameters of the ground model are given in Table 1.

Table 1. Ground model coefficients

Symbol	Description	Value	Units
K_G	Vertical spring	150 000	N/m
D_G	Vertical damper	150	(N.s)/m
K_F	Friction stiffness	150 000	N/m
D_F	Friction damper	150	(N.s)/m
μ	Friction coefficient	0.9	⟨unitless⟩

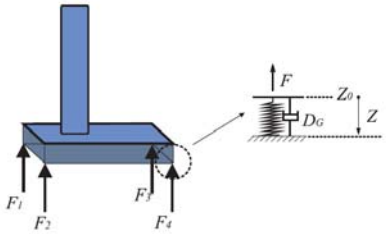


Fig. 2. Four ground contact point is introduced under each foot

Having developed a realistic simulation of walking for COMAN, the next section discusses the method used for walking trajectory generation.

3. Trajectory generation

A trajectory generator computes the reference trajectory for the robot's joints. This control system translates the desired walking parameters such as foot locations, step length, walking speed and walking direction into feasible and stable joints' trajectories. In this paper, the preview control method of ZMP based on the cart table model, proposed in [10], is used. The advantages of using this method are the ability of the robot to modify the reference trajectories according to the walking path and low computational cost which makes it suitable for online calculations. A brief description of this method is given below.

In this method, the cart-table linear model (Fig. 3) is used to formulate the relation between the centre of mass motion and the ZMP. This model has linear and decoupled dynamics in sagittal and lateral planes due to the constraint on the height of the centre of mass, which moves along a plane. This simplification in the nonlinear dynamics of an inverted pendulum results in derivation of the closed form equation between centre of mass and zero moment point $p_x = x - \frac{z_c}{g}\ddot{x}$, where x denotes the position of centre of mass and p_x is the position of the zero moment point in x direction. Since the dynamics are decoupled (constant height), the same equation holds for the y direction.

In order to generate the motion of the centre of the mass using (1) the initial position of the CoM and the ZMP should coincide. Based on the foot paths of the robot, a reference ZMP is derived as shown in Fig. 4 which shows the trajectory for three steps with 0.1 m step size. The continuous time based ZMP reference can be designed by using either linear or spline interpolation of a set of ZMP points with respect to time. The overall pattern generation scheme as well as the dynamic simulator and the learning algorithm are shown in Fig. 5. The objective locomotion parameters such as the walking speed and foothold planning are assumed to be given. Further details about the trajectory generator are provided in [10].

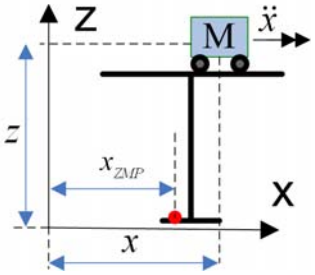


Fig. 3. Cart table model

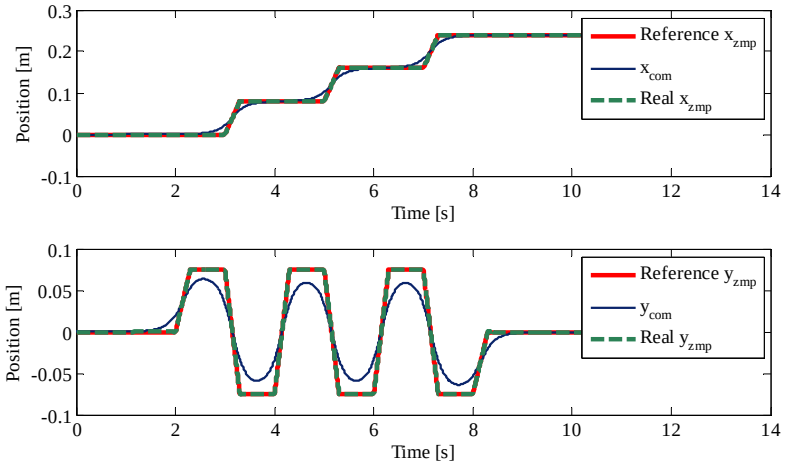


Fig. 4. The centre of mass and ZMP trajectories in sagittal and lateral planes

However, this method has a number of limitations. Firstly, the cart-table model only considers the overall CoM therefore the multi-body dynamics is not considered. Secondly, the control scheme assumes ideal position tracking and the dynamic effect of the springs in the compliant joints is not included. Therefore, the process of generating models using the simple cart table model and applying it to the full multibody system with compliance involves a considerable number of trial and errors. In other words, this method has only been applied before to find a locally stable gait, while for the first time in this paper, reinforcement learning for global search is proposed to explore the whole parameter space using the accurate dynamic walking simulation of COMAN. The result of the search yields multiple

stable solutions which can then be studied further for walking sensitivity analysis to gait parameter variations and also to include additional optimization criteria such as energy efficiency to choose among the dynamically feasible gaits. The learning algorithm is described in the next section.

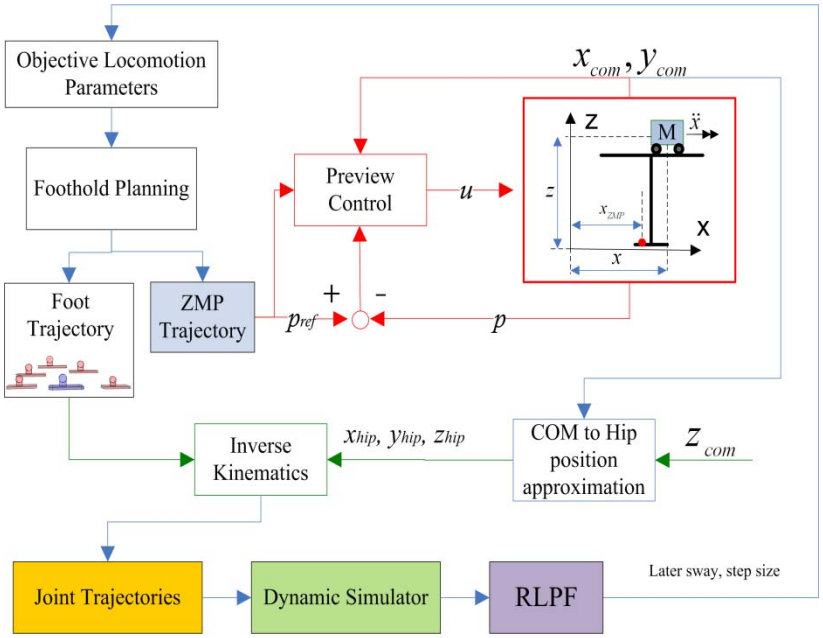


Fig. 5. Diagram of the overall learning based trajectory generation method is illustrated. The Cartesian position of the CoM of the robot is defined by x_{com}, y_{com} and z_{com}. The reference ZMP trajectory is p_{ref} and p is the ZMP feedback signal from the Cart table model which consists of x_{ZMP} and y_{ZMP}. The objective locomotion parameters namely, lateral sway and step size are provided by RLPF for a desired walking speed

4. Reinforcement learning

This section describes a recently proposed Reinforcement Learning algorithm based on Particle Filters (RLPF) for global search in policy space, which is capable of finding multiple alternative optimal policies [1]. The algorithm performs global search in the policy space, therefore eliminating the dependency on the policy initialization, and it has the ability to find the globally optimal policy.

Linking particle filters and RL is explained by the following observation. The landscape, defined by the reward function $R(\theta) \in \mathbb{R}$ over the whole continuous domain of the parameter space $\theta \in \Theta$, can be viewed as defining an Improper Probability Density Function (IPDF). An IPDF is similar to probability density function except that the integral of it does not have to be equal to one. This is possible even if the reward function $R(\theta)$ has negative values in its range, because a constant positive number can be added to the reward to obtain a non-negative reward function $R'(\theta)$ which has exactly the same set of optimizers $\theta^* \in \Theta$ as

$R(\theta)$. Hence, optimizing $R'(\theta)$ will also optimize $R(\theta)$. Based on this assumption that $R(\theta)$ is an IPDF, the RL problem can be reformulated as follows. Each trial $\tau(\pi(\theta))$ (τ denotes each trial and π is the policy for a given parameter vector θ) can be considered as an independent sample from the unknown IPDF. The RL algorithm chooses a finite number of sample points to find the values and modes of the unknown IPDF.

The main idea of RLPF is to use particle filtering as a method for choosing the sampling points, i.e. for calculating a parameter vector θ for each trial, which consist of the step size and lateral sway. A policy particle p_i is defined to be the tuple $p_i = \langle \theta_i, \quad \tau_i, \quad R_i \quad \omega_i \rangle$, where the particle p_i represents the outcome of a single trial τ_i performed by executing an RL policy $\pi(\theta_i)$, where θ_i is a vector of policy parameter values modulating the behavior of the RL policy π. The policy particle also stores the value of the reward function evaluated for this trial $R_i = R(\tau_i(\pi(\theta_i)))$. The variable τ_i contains task-specific information recorded during the trial depending on the nature of the task. The information in τ_i is used by the reward function to perform its evaluation. The variable ω_i is the importance weight of this policy particle, and the way of its calculation is explained as follows.

It is assumed that the set of particles $\{p_i\}$ is an approximate implicit representation of the underlying unknown IPDF defined by $R(\theta)$. Therefore, in order to select a new particle consistent with the real IPDF distribution, the samples are taken from the approximate distribution while correcting for the discrepancy. The mechanism for this correction is provided by the importance weights $\{\omega_i\}$.

Firstly, each policy particle p_i is assigned a scalar importance weight ω_i derived from its corresponding reward R_i using a transformation function g, such that $\omega_i \propto g(R_i)$. In the simplest case, $g(\cdot)$ could be the identity, but in the general case, it could be an arbitrary non-negative function. The function g is applied in such a way, that the importance weights are normalized, in the sense that $\forall \omega_i \; 0 < \omega_i < 1$, and also $\sum \omega_i = 1$. Secondly, an auxiliary function $h(u) = \int_{-\infty}^{u} \omega_u du$ is constructed, which takes the form $h(k) = \sum_{j=1}^{k} \omega_j$ in the discrete case. This function can be thought of as the (approximate) Cumulative Density Function (CDF) of the unknown PDF. Indeed, due to the way we create the importance weights, it follows directly that $\int_{-\infty}^{+\infty} \omega_u du = 1$, and thus $h(u)$ is a proper CDF. This is important because,

given that $\omega_i > 0$, it guarantees that $h(u)$ is strictly monotonically increasing and therefore the inverse function h^{-1} exists.

Algorithm 1 Reinforcement Learning based on Particle Filters (RLPF)

Input: parameterized policy π, policy parameter space Θ, reward function $R(\theta)$ where $\theta \in \Theta$, reward transformation function g, total number of trials N, initialization number of trials $L < N$, initial noise ε_0, noise decay factor λ, maximum number of particles σ.

Let $S = \{\}$ {A set of policy particles}
for $l = 1$ **to** L **do**
  Draw $\theta_l \sim U(\Theta)$ {Sample L initial particles}
  Perform trial $\tau_l(\pi(\theta_l))$
  Create new policy particle $p_l = \langle \theta_l, \tau_l, R_l, \omega_l \rangle$
  $S = S \cup \{p_l\}$
end for
for $n = L + 1$ **to** N **do**
  Let $h(0) = 0$
  for $i = 1$ **to** $|S|$ **do**
    $\omega_i = \frac{g(R_i)}{\sum_{j=1}^{|S|} g(R_j)}$ {Calc. importance weights}
    $h(i) = h(i-1) + \omega_i$ {Calc. aux. function}
  end for
  Draw $z \sim U(0,1)$
  Let $y = h^{-1}(z)$
  Let $k = \lceil y \rceil$ {$\lceil . \rceil$ is the ceiling function}
  Select policy particle $p_k = \langle \theta_k, \tau_k, R_k, \omega_k \rangle$
  Let $\varepsilon_n = \varepsilon_0 \lambda^{(n-L-1)}$ {noise with exp. decay λ}
  Let $\theta_n = \theta_k + \varepsilon_n$
  Perform trial $\tau_n(\pi(\theta_n))$
  Create new policy particle $p_n = \langle \theta_n, \tau_n, R_n, \omega_n \rangle$
  $S = S \cup \{p_n\}$
  if $|S| > \sigma$ **then**
    {Remove policy particle with smallest reward}
    $j = \arg\min_{p_j \in S} g(R_j)$
    $S = S \setminus \{p_j\}$
  end if
end for

Thirdly, a random variable z is introduced which is uniformly distributed in the interval (0, 1). Now, it can be shown that the random variable y defined as $y = h^{-1}(z)$ is distributed (approximately) according to the desired unknown PDF, see e.g. [17].

The goal of RLPF is not to approximate the expectation of a function, but rather, to find the mode (or modes) of the unknown function $R(\theta)$. The pseudo-code for RLPF is given in Algorithm 1. In the results section the reward function, number of trials, and gait parameter space are provided.

5. Simulation results

In this section, the results of applying RLPF algorithm to find optimal gait parameters for two desired walking speeds are presented. The reward function is defined in (4) where θ denotes the gait parameters, namely the step size and the lateral sway, B is a Boolean variable which is zero when the robot falls in the last trial and one otherwise, V_d is the desired average walking speed and V is the achieved walking speed in the dynamic simulator. The coefficients c_1 and c_2 are chosen to be 1000, and c_3 is 100. The goal of the reward function in (4) is to distinguish between the dynamically stable and unstable gaits and to feed back the achieved walking speed to the learning algorithm.

$$R(\theta) = c_1(1 - B) + c_2 e^{c_3 (V_d - V)^2}. \tag{4}$$

Initially the desired speed was set to 0.05 m/s and learning was used in 120 trials, where both a stable gait and the desired walking speed were achieved. The reward function is shown in Fig. 7. It can be seen that after the 30th trial the robot has not fallen since the reward is above 1000 points and the algorithm is only adjusting the walking speed. Moreover, the reward of each trial is color coded and plotted in Fig. 8 which shows that two clusters of parameters are found which gives the highest rewards (i.e., gaits which are stable and close to the desired walking speed). These two clusters of stable gaits have a vertical spread which suggests the lateral sway parameter has less sensitivity on stability compared to the step length. The lateral sway parameter is related to the dynamics of the robot in the lateral plane which has stiff joints (no passive compliance) and the step length parameter is directly related to the sagittal dynamics of the robot with passive compliance in the ankles, knees and the hips. Therefore, for a fixed passive compliance and using 120 trials a set of step sizes are derived which are between 1-3 cm. increasing the number of trials can further explore the parameter space and provide walking gaits with larger step sizes as shown in second experiment of this section. Also, this effect is due to the joint servo designs which are controlling the motors' positions to track the walking trajectories. Designing the servo controllers to control the link positions can improve the range of step sizes, while the achieved walking speed will depend on the bandwidth of the servo controllers. The robot will only be able to walk with trajectories speeds which are within its tracking bandwidth. The snapshot

of the optimal walking gait with the highest reward is shown in Fig. 6, where the robot takes four steps with speed of 0.0499 m/s.

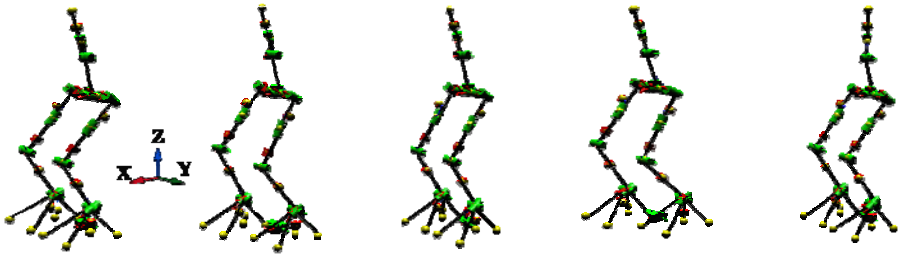


Fig. 6. Snapshots from the simulation of the gait computed in trial 120

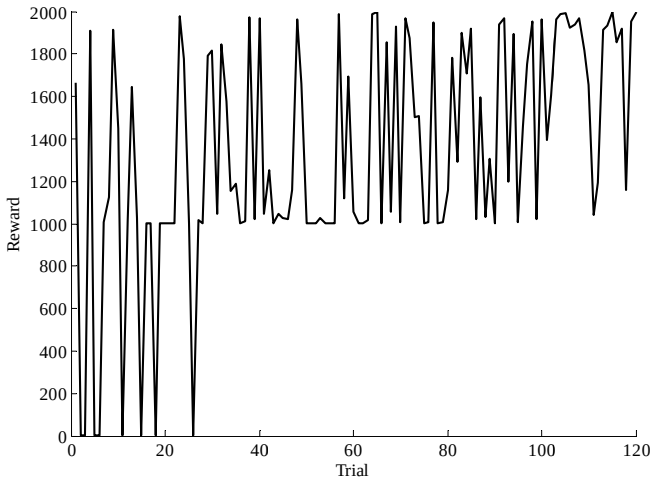


Fig. 7. Reward values during learning for desired speed of 0.05 m/s

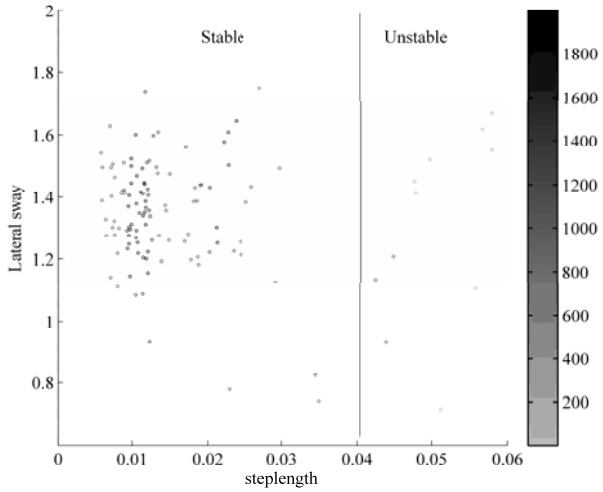


Fig. 8. Reward values in the walking gait parameter space with desired speed of 0.05 m/s

Moreover, the desired walking speed was set to 0.15 m/s and the RLPF algorithm was applied to 400 trials. This speed is above the tracking bandwidth of the current PID controllers set in the simulation and the robot achieved stable walking but the maximum achieved walking speed was lower (0.75 m/s).

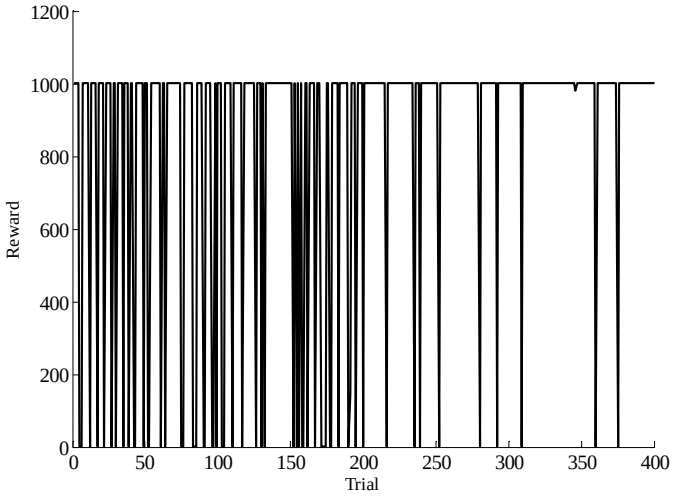


Fig. 9. Reward values during learning for desired speed of 0.15 m/s

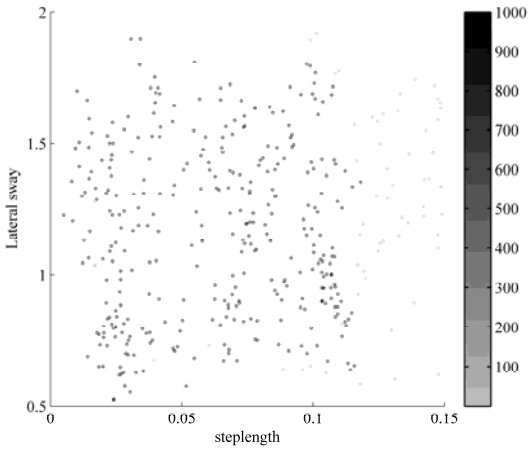


Fig. 10. Reward values in the walking gait parameter space with desired speed of 0.15 m/s

The reward values are shown in Fig. 9, which essentially distinguishes between the stable and unstable gaits, since the extra reward of walking close to the desired speed of 0.15 m/s is not obtained in the dynamic simulations. However, it can be seen that the falling frequency is decreasing with the number of trials and the learning is converging to higher rewards. The reward values in the parameter space are shown in Fig. 10, which shows the robot has taken steps sizes between 1-12 cm and has been stable.

6. Conclusion

In this paper the common problem of ZMP trajectory generation using simplified cart-table model which involves considerable amount of trials and errors to find locally stable gaits was considered. A reinforcement learning algorithm was combined with a dynamic walking simulator to perform automated global search for stable gaits in the walking parameter space. The algorithm was tested on two relatively low (0.05 m/s) and high (0.15 m/s) walking speeds and it found multiple sets of stable walking for a given speed with different step lengths and lateral sway. The result of this algorithm can be used to study the sensitivity of the gaits to parameter changes as well as including additional optimization criteria (such as energy efficiency) to narrow down the set of stable gaits.

In the future work, the designed walking gaits will be applied on the real robot, and in a possible future study to investigate the energy efficiency of a certain gait using simple models and learning methods to predict the optimal step length for a given walking speed. Also the simple reward function used in this study can be improved to better distinguish among different type of gaits.

Acknowledgements.: This work is supported by the FP7 European project AMARSi (ICT-248311).

References

1. Kormushev, P., D. G. Caldwell. Simultaneous Discovery of Multiple Alternative Optimal Policies by Reinforcement Learning. – In: IEEE International Conference on Intelligent Systems 2012, Sofia, Bulgaria (Accepted).
2. Morimoto, J., G. Cheng, C. G. Atkeson, G. Zeglin. A Simple Reinforcement Learning Algorithm For Biped Walking. – In: IEEE International Conference on Robotics and Automation'2004, New Orleans, LA, USA.
3. Endo, G., J. Morimoto, T. Matsubara, J. Nakanishi, G. Cheng. Learning CPG-Based Biped Locomotion with a Policy Gradient Method: Application to a Humanoid Robot. – The International Journal of Robotics Research, Vol. **27**, 2008, No 2, 213-228.
4. Tedrake, R., T. W. Zhang, H. S. Seung. Stochastic Policy Gradient Reinforcement Learning on a Simple 3D Biped. – In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, Sendai, Japan. 2849-2854.
5. Wada, Y., K. Sumita. A Reinforcement Learning Scheme for Acquisition of Via-Point Representation of Human Motion. – In: IEEE International Joint Conference on Neural Networks'2004, Vols 1-4, 2004, 1109-1114.
6. Kormushev, P., B. Ugurlu, S. Calinon, N. G. Tsagarakis, D. G. Caldwell. Bipedal Walking Energy Minimization by Reinforcement Learning with Evolving Policy Parameterization. – In: IEEE/RSJ International Conference on Intelligent Robots and Systems'2011, San Francisco, CA, USA.
7. Kajita, S., F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, H. Hirukawa. Biped Walking Pattern Generation by a Simple Three-Dimensional Inverted Pendulum Model. – Advanced Robotics, Vol. **17**, 2003, No 2, 131-147.
8. Kajita, S., F. Kanehiro, K. Kaneko, K. Yokoi, H. Hirukawa. The 3D Linear Inverted Pendulum Mode: A Simple Modeling for a Biped Walking Pattern Generation. – In: IEEE/RJS International Conference on Intelligent Robots and Systems (IROS), 2001, 239-246.
9. Goswami, A., B. Thuilot, B. Espiau. A Study of the Passive Gait of a Compass-Like Biped Robot: Symmetry and Chaos. – International Journal of Robotics Research, Vol. **17**, 1998, No 12, 1282-1301.

10. Kajita, S., F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa. Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point. – In: IEEE International Conference on Robotics and Automation (ICRA), 2003. 1620-1626.
11. Kajita, S., M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, K. Yokoi. Biped Walking Stabilization Based on Linear Inverted Pendulum Tracking. – In: IEEE/Rsj International Conference on Intelligent Robots and Systems (IROS), 2010.
12. AMARSI. Adaptive Modular Architectures for Rich Motor Skills. EU Supported FP7 Project. **http://www.amarsi-project.eu/**
13. Tsagarakis, G., N., Z. Li, J. Saglia, D. G. Caldwell. The Design of the Lower Body of the Compliant Humanoid Robot "cCub". – In: IEEE International Conference on Robotics and Automation (ICRA), 2011, Shanghai, China.
14. Buchli, J., M. Kalakrishnan, M. Mistry, P. Pastor, S. Schaal. Compliant Quadruped Locomotion Over Rough Terrain. – In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2009, St. Louis, USA.
15. Featherstone, R. Rigid Body Dynamics Algorithms. 1st Ed. New York, Springer 2008, Science+Business Media, LLC. 280.
16. Mistry, M., J. Nakanishi, G. Cheng, S. Schaal. Inverse Kinematics with Floating Base and Constraints for Full Body Humanoid Robot Control. – In: IEEE-RAS International Conference on Humanoid Robots, 2008, Daejeon, Korea.
17. Bishop, C. M. Pattern Recognition and Machine Learning. New York, Springer, 2006.