

# Humanoid Gait Optimization Resorting to an Improved Simulation Model

Regular Paper

José L. Lima<sup>1,3,\*</sup>, José A. Gonçalves<sup>1,3</sup>, Paulo G. Costa<sup>2,3</sup> and A. Paulo Moreira<sup>2,3</sup>

1 Department of Electrical Engineering of the Polytechnic Institute of Bragança, Portugal

2 Department of Electrical Engineering and Computers at the Faculty of Engineering of the University of Porto, Portugal

3 INESC TEC (formerly INESC Porto) and the Faculty of Engineering of the University of Porto, Porto, Portugal

\* Corresponding author E-mail: jllima@ipb.pt

Received 14 May 2012; Accepted 30 Oct 2012

DOI: 10.5772/54766

© 2013 Lima et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract** The simulation of a robot with a high number of joints can easily become unstable. Numerical errors on the first joint of the chain are propagated to the other joints. This is a very common problem in humanoid robots. A way to plan the gait for these robots is using simulation and optimization techniques. This paper addresses a new approach to optimizing gait parameter sets using an adaptive simulated annealing optimization algorithm, combined with a new joint model that reduces its instability. The new model and the optimization are implemented in SimTwo (a developed physical robot simulator that is capable of simulating user defined robots in a three-dimensional space, since it includes a physical model based on rigid body dynamics) and results are shown that validate the approach.

**Keywords** Simulation, Modelling, Humanoid Robots

## 1. Introduction

Research on humanoid robots has shown notable progress in terms of design and control. However, gait generation and optimization techniques still remain a

challenge for such high-order, highly-coupled, nonlinear and dynamic systems [1].

Gait generation in a humanoid robot is a good area to apply optimization methods that are able to deliver numerical results. Time consuming tasks should be avoided in the robot's software and hardware. These tasks should be performed off-line. It is also beneficial to be able to adapt the robot gait to different conditions [2].

In order to generate walking patterns for different locomotion kinematics, most existing approaches precompute (off-line) reference trajectories [3]. This paper addresses an overview of our efforts to develop optimized gait planning for humanoid robots based on simulation that can be applied to the real robot. For that purpose, the Adaptive Simulated Annealing approach was used for the optimization technique. It is a generic probabilistic method for the global optimization problem. This approach tries to find the global minimum of a given function in a large search space (as it happens in humanoid gait planning).

However, simulating a robot with a high number of joints that connects different parts (of the same robot) can easily become unstable. Numerical errors on the first joint of the

chain are propagated to the other joints. Due to its composition, this is a critical problem in a realistic humanoid robot simulation. Two solutions were used to solve this problem: a simplified model (reducing joints) was applied and an improved gearbox motor model was included.

The simulations were conducted in SimTwo [4] (available at <http://paginas.fe.up.pt/~paco/pmwiki/index.php?n=SimTwo>. SimTwo), a developed physical robot simulator that is capable of simulating user defined robots in a three-dimensional space, since it includes a physical model based on rigid body dynamics (the ODE - Open Dynamics Engine [5]).

This paper is organized as follows: After the Introduction, Section 2 briefly points out the developed Simulator (SimTwo), while Section 3 presents the improved gearbox model for the humanoid robot. Afterwards, in Section 4, the Adaptive Simulated Annealing approach is addressed to optimize gait planning. Results are presented in Section 5 and finally some conclusions are presented in Section 6.

## 2. Developed Simulator

Studying the robot's behaviour without real hardware is possible due to a physics-based simulator implementation. The physics engine is the key to making simulations useful in terms of high performance robot control [6]. The dynamic behaviour of the robot (or multiple robots) is computed by the ODE (Open Dynamics Engine [4]), which is a free library for simulating rigid body dynamics.

There are several simulators available nowadays, such as Simspark [7], Webots [8], MURoSimF [9], Microsoft Robotics Studio [10], Gazebo [11] and OpenHRP3 [12].

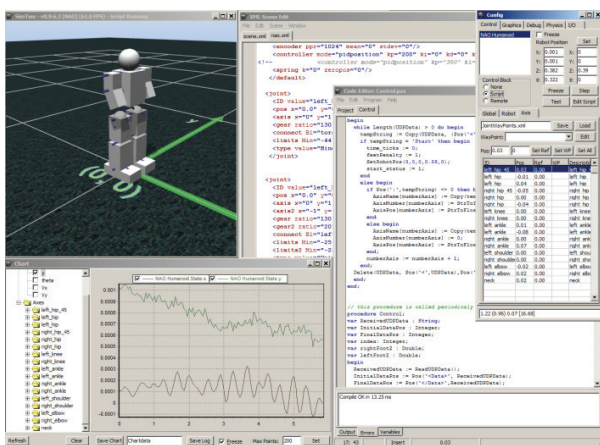


Figure 1. SimTwo simulator with its windows

By using extended xml scene description language, the SimTwo can be configured to accomplish the authors' needs. It includes several sensors and actuator models

that enable a very realistic simulation environment (it is also easy to develop and model new sensors and actuators in SimTwo). The developed realistic simulation system can support several types of robots.

Its main purpose is simulating mobile robots (such as wheeled or legged robots), although industrial robots, conveyor belts, lighter-than-air vehicles and underwater vehicles can also be defined. Basically, any type of terrestrial robot definable with revolution or prismatic joints and/or wheels can be simulated with this software. Figure 1 is a screenshot of the developed simulator with its main windows (scene observer, robot variables chart, scene editor, code editor and configuration parameters).

### 2.1 Simulator control architecture

The communication between high and low level controllers brings some limitations to the control loop, such as lag time, as happens with real robots. The developed simulator enhances these properties, thus increasing the realism of SimTwo.

At the lowest level, the servomotor model includes the control loop, just like real servomotors. At the highest level, a closed loop controller handles the movement of the joints. Figure 2 illustrates the control architecture of SimTwo and its cycle times for each level.

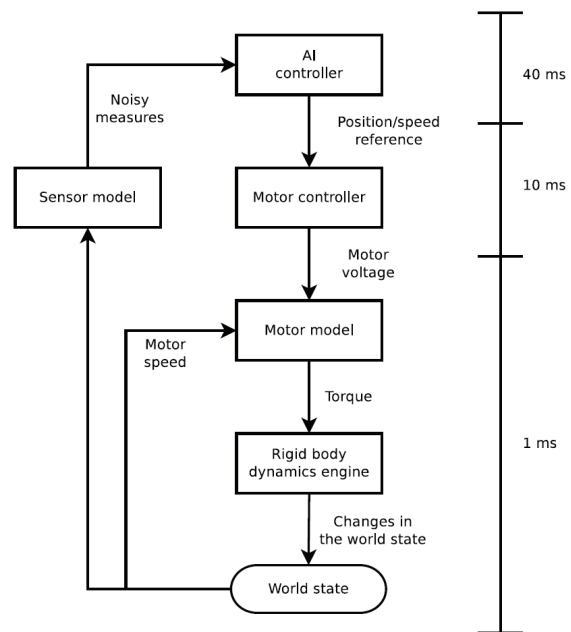


Figure 2. SimTwo controller architecture

### 2.2 Servomotor model

The servomotor can be based on a DC motor model with its main characteristics and with an embedded state space feedback closed-loop controller. The block diagram of the servomotor model can be seen in Figure 3. The variables are explained in Table 1.

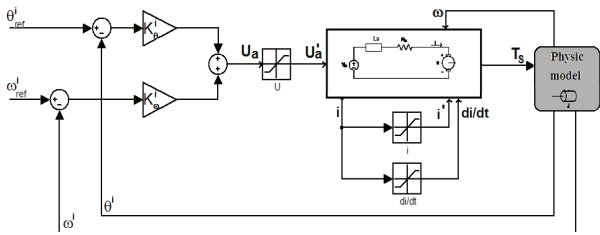


Figure 3. Servomotor model

Symbol	Description
$\theta'_{ref}$	Reference of the angle
$\omega'_{ref}$	Reference of angular speed
$U_a'$	Supply voltage of DC motor (with nonlinearities)
$T_s$	Available torque that will be applied to the gearbox (Electrical torque)

Table 1. Description of servomotor model variables

The motor model parameter estimation has already been presented in previous work [13], as well as the friction constants that were computed based on an optimization method with validation of real hardware. The values are presented in Table 2.

Symbol	Description	Value
$R_a$	Internal resistor	$8 \Omega$
$L_a$	Internal inductance	$5 \text{ mH}$
$K_s$	Motor constant (relates angular speed with back e.m. voltage)	$0.006810 \text{ V.s/rad}$
$B_v$	Viscous constant	$0.01278 \text{ N.m.s/rad}$
$F_c$	Static friction constant	$0.0000171 \text{ N.m}$

Table 2. Motor model values

The low level controller models the closed loop controller in the real robot and is implemented by the servomotor manufacturer. The model and the real servomotor (AX12 from Dynamixel) can be compared in two situations: a step to observe the controller response (Figure 4) and a free fall of an arm to observe the friction comparison (Figure 5). Results validate the simulator approach.

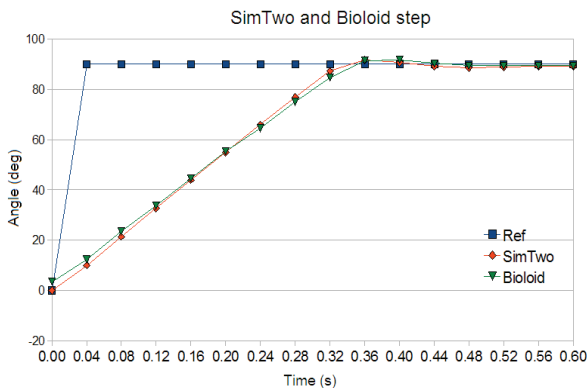


Figure 4. Controller step response comparison

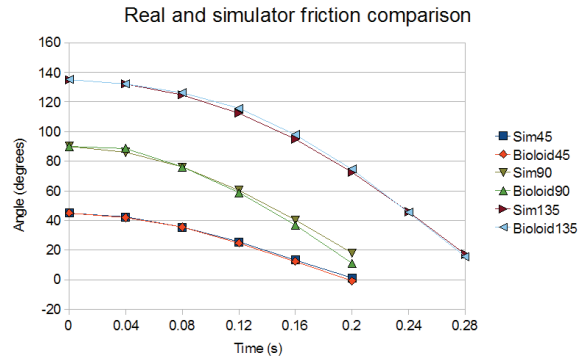


Figure 5. Free arm fall comparison

### 3. Improved gearbox model

The presented model shows some instability for the high values of the gearbox ratio. This happens because an inertia moment of the motor shaft should appear multiplied by the square of the gearbox ratio. However, that property is missing from the model. Both the rotor inertia and the gearbox that transmits torque to the output must be modelled. The model representing the system between the rotor and the load is described in this subsection. This transmission approach is based on a spring damper and makes it possible to increase the stability of the simulation. Moreover, the backlash of the gearbox can be approximated by this model. The improved model is implemented over the “Physics model” presented in Figure 3, thus keeping the servomotor model and the controller unchanged. The motor inertia, plus a spring damper gearbox model, MIPS (Equation 1), are presented in Figure 6 where the  $T_{axis}$  is the available torque at the gearbox output.

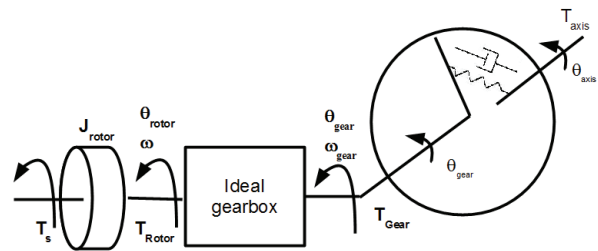


Figure 6. Gearbox model interface

$$\dot{\omega} = \frac{T_s - T_{Bv} + T_{Rotor}}{J_{rotor}} \quad (1)$$

where the  $T_{Bv}$  is the damping friction as presented in Equation 2.

$$T_{Bv} = B_{vr} \cdot \omega \quad (2)$$

$T_{Rotor}$  is the available torque seen by the rotor as presented in Equation 4.  $K_s$  and  $K_B$  are obtained by experimental tests.  $K_s = 0.00001 \text{ N.m/rad}$  and  $K_B = 0.0001 \text{ N.m.s/rad}$ .

$$T_{Gear} = -T_{Axis} \quad (3)$$

$$T_{axis} = K_s \cdot e + K_B \frac{de}{dt} \quad (4)$$

Assuming that  $G_{ratio}$  is the gear ratio constant:

$$T_{Rotor} = \frac{T_{gear}}{G_{ratio}} \quad (5)$$

where

$$e = \frac{\theta_{rotor}}{G_{ratio}} - \theta_{axis\ model} \quad (6)$$

In order to validate this approach, an arm with 7 chained joints was built, as presented in Figure 7.

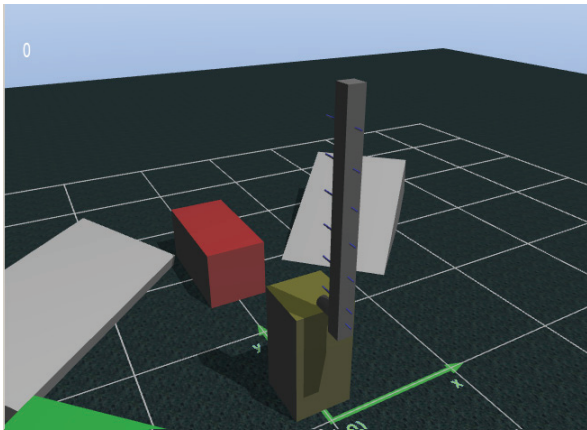


Figure 7. Arm composed by 7 cascade joints

The aim is to keep the lower joint static at 0 degrees (vertical position) so that the error can be measured. In this way, it is possible to calculate the numeric error (instability) for the lower joint. Figure 8 shows screenshots of the old model instability.

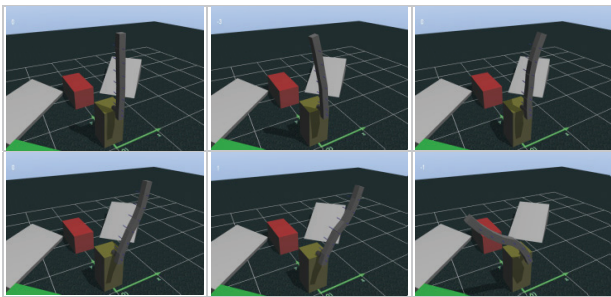


Figure 8. Set of screenshots for the old model

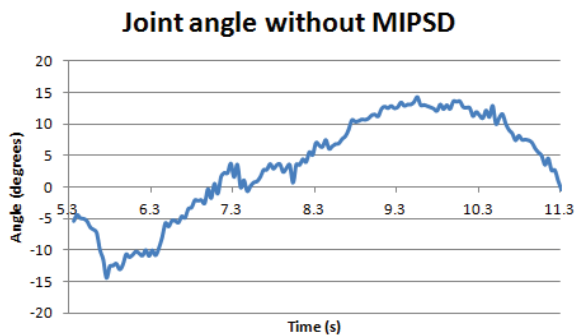


Figure 9. Gearbox model (without MIPSD)

As it can be seen in Figure 9 and Figure 10 the spring-damper model (MIPSD) is more stable than the standard model. The standard error deviation is reduced (from 8.14 rad<sup>2</sup> to 0.0022 rad<sup>2</sup>) with a reference of zero degrees.

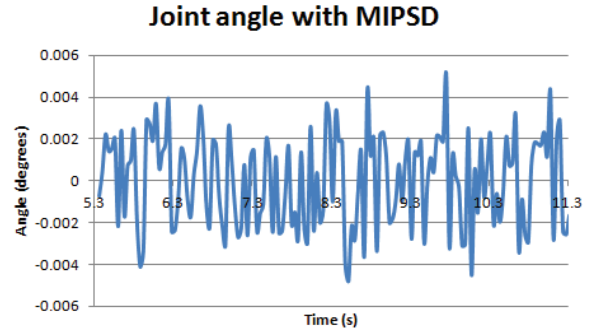


Figure 10. Gearbox model (with MIPSD)

In addition to the stability, the repeatability of measurements is better when the MIPSD model is used. For a series of robot movements (further presented in Section 4 and in Figure 12), it is possible to conclude that for two hundred iterations the energy consumption variance is lower when the MIPSD model is used (3.9% of standard variation against 14.62% without the MIPSD model), as presented in Figure 10.

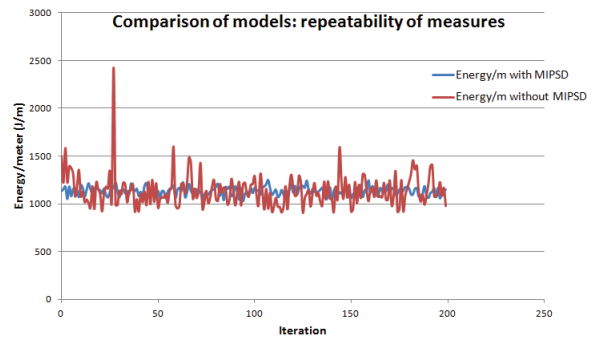


Figure 11. Repeatability of energy measurements for both models

A complete humanoid model can also be used to enhance the MIPSD model. Figure 12 presents a simulated humanoid robot based on a real humanoid. Some joint angles (Left Lateral Shoulder with a reference of -45 degrees, Right Front Shoulder with a reference of 25 degrees, Right Lateral Shoulder with a reference of 25 degrees and Left Elbow with a reference of 45 degrees) are presented for both models in a static posture.

It shows an increase in stability, as presented in Table 3, where  $\delta$  is the standard deviation and Min and Max are the minimum and the maximum reached angles. Standard deviation decreases in all cases and shows the stability of the MIPSD model. Graphical data of a time function with the standard and MIPSD models are presented in Figure 12 for the left elbow joint.

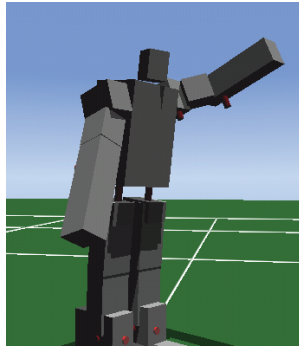


Figure 12. Complete humanoid model

		Left Lat Shoulder	Right Front Shoulder	Right Lat Shoulder	Left Elbow
Standard Model	Max	-45.06	25.00	26.60	48.40
	Min	-47.21	24.13	24.15	44.19
	$\delta$	0.39	0.16	0.50	0.84
MIPSD Model	Max	-44.96	25.04	25.03	45.05
	Min	-45.04	24.97	24.96	44.95
	$\delta$	0.01	0.01	0.01	0.01

Table 3. Joint angles comparison for Standard and MIPSD models with complete humanoid model.

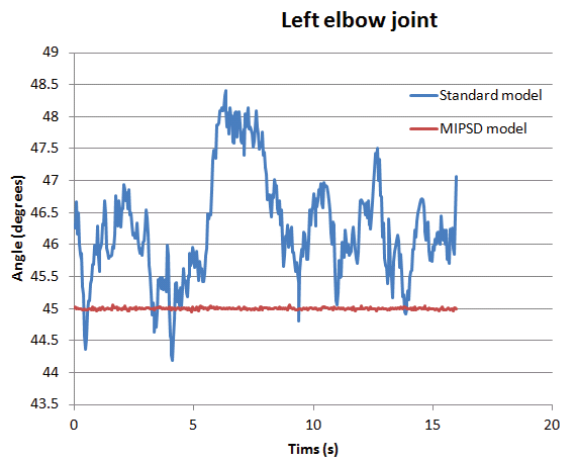


Figure 13. Humanoid left elbow joint time function for standard and MIPSD models.

#### 4. Adaptive simulated annealing

The gait is one of the fundamental problems in humanoid mobile robots [14].

The search space in a problem depends on the complexity of the step. A complete and symmetrical step can be composed of 4 states; further states can be achieved by juxtaposing the first 4 states (repeatability of the gait). Each state is composed of 10 degrees of freedom (each joint axis). Therefore, the optimization should be applied to a dimension (D) of 40. The Adaptive Simulated Annealing algorithm can be implemented in the SimTwo simulator by adding some random noise (dimension D) to each iteration for present state and analyzing the

objective (cost  $f(X_D)$ ) function. If the solution is better than the last solution then it is accepted as the new solution to start a new search. The random vector size (D) depends on the number of missed iterations. The process ends after a predetermined number of iterations without progress. Figure 11 presents the described adaptive simulated annealing optimization algorithm.

**Input:** Initial state  $s_{init}$   
**Output:** Optimized state  $s_{best}$   
 $k \leftarrow 0$ ;  $miss \leftarrow 1$   
**while**  $k < KMAX$  **do**  
     $s_{new} \leftarrow neighbour(s)/miss$   
     $e_{new} \leftarrow E(s_{new})$   
     $miss \leftarrow miss + 1$   
    **if**  $e_{new} < e_{best}$  **then**  
         $s_{best} \leftarrow s_{new}$   
         $e_{best} \leftarrow e_{new}$   
         $miss \leftarrow 1$   
    **end**  
    **if**  $miss = MAXMISS$  **then**  
         $MISS = 0.1$   
    **end**  
     $k \leftarrow k + 1$   
**return**  $s_{best}$   
**end**

Figure 14. Adaptive Simulated Annealing Algorithm

Starting with a first gait approach, based on ZMP biped walking pattern generator [15] and partially presented in the sequence snapshot of Figure 12, several simulations are executed in order to achieve the best cost function.

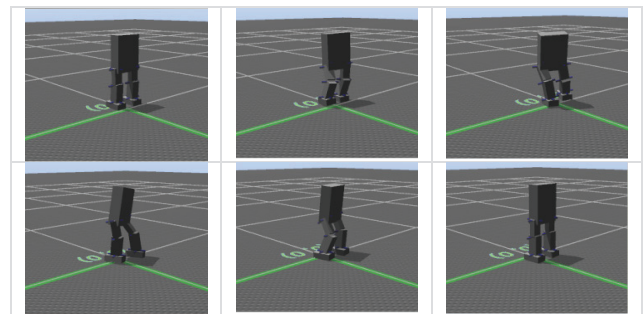


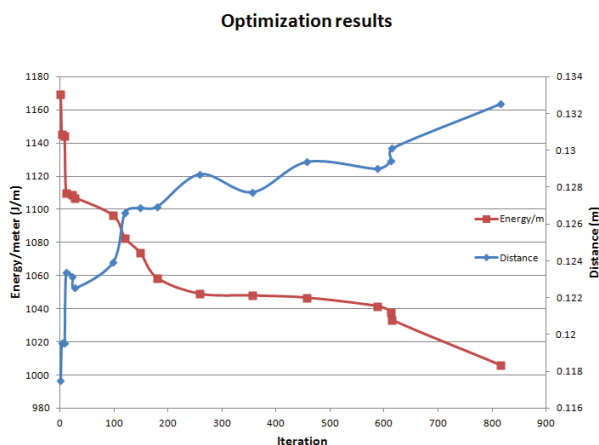
Figure 15. Simplified model complete walking snapshots

The optimization function tries to achieve the lowest energy consumption per distance travelled. This is an important measurement because humanoids are powered by on-board batteries and, therefore, their autonomy depends on how efficient energy consumption is. The trajectory controller should be optimized in order to minimize energy consumption with walking gait optimization [16].

The following section addresses the results and comparison for the new model.

## 5. Results

The accuracy of the new MIPS model makes it possible to obtain more stable results. The presented work optimizes the gait in order to minimize energy consumption per distance travelled. Figure 13 shows the results for three thousand iterations, where optimization takes eight hundred iterations. The older model needs more than 2600 iterations to achieve the final optimized gait [17]. This means that the adaptive simulated annealing and the new model approach convergence to the desired solution significantly faster.



**Figure 16.** Energy per distance optimization results

As can be seen, the initial gait consumption is decreased by 14% (which also increased the step covered distance by 13%).

## 6. Conclusions and future work

This paper addresses a faster and hardware-saving optimization based on the adaptive simulated annealing approach to optimize the gait of a biped humanoid robot. It uses a new gearbox model that makes it possible to increase the simulation stability and to reduce the numerical errors when a chain of joints are connected. The gait parameters were optimized in order to minimize energy consumption. The final results, which help validate the approach, show the optimized gait achieving a walking sequence with lower energy consumption. In future work, the use of a complete model is now possible due to the new MIPS model that exhibits a lower level of numerical errors, more accurate results and most importantly, an improved stability.

## 7. Acknowledgment

This work is financed by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project «FCOMP - 01-0124-FEDER-022701».

## 8. References

- [1] L. Hu, C. Zhou and Z. Sun, Biped Gait Optimization Using Spline Function Based Probability Model, Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida, 2006.
- [2] C. Niehaus, T. R. Röfer, T. Laue. Gait Optimization on a Humanoid Robot using Particle Swarm Optimization, Proceedings of the Second Workshop on Humanoid Soccer Robots in conjunction with the 2007 IEEE-RAS International Conference on Humanoid Robots, 2007.
- [3] S. Sakka and K. Yokoi, Humanoid Vertical Jumping based on Force Feedback and Inertial Forces Optimization, Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005.
- [4] P. Costa, J. Gonçalves, J. Lima and P. Malheiros, SimTwo Realistic Simulator: A Tool for the Development and Validation of Robot Software, International Journal of Theory and Applications of Mathematics & Computer Science, Vol 1, n.1, pp. 17-33, 2011.
- [5] R. Smith, Open Dynamics Engine, <http://www.ode.org>, 2000.
- [6] B. Browning and E. Tryzelaar, Ubersim: A realistic simulation engine for robotsoccer. In Proceedings of Autonomous Agents and Multi-Agent Systems, 2003, AAMAS' 03.
- [7] Simspark, [simspark.sourceforge.net/wiki/](http://simspark.sourceforge.net/wiki/)
- [8] Webots, <http://www.cyberbotics.com/>.
- [9] MURoSimF, [www.dribblers.de/murosimgf/](http://www.dribblers.de/murosimgf/).
- [10] Microsoft Robotics Studio, [www.microsoft.com/robotics/](http://www.microsoft.com/robotics/).
- [11] Gazebo, [playerstage.sourceforge.net/gazebo/gazebo.html](http://playerstage.sourceforge.net/gazebo/gazebo.html)
- [12] Open architecture humanoid robotics platform, <http://www.openrtp.jp/openhrp3/3.0.3/en/index.html>
- [13] J. Lima, J. Gonçalves, P. Costa and A. Moreira, HUMANOID REALISTIC SIMULATOR: The servomotor joint modeling, In 6th International Conference on Informatics in Control, Automation and Robotics, 2005.
- [14] K. Mombaur, Using optimization to create self-stable human-like running, Robotica 2009, Cambridge University Press, 2009.
- [15] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara and H. Hirukawa, Biped walking pattern generator allowing auxiliary ZMP control, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2994-2999, Beijing, China, 2006.

- [16] Z. Tang and C. Zhou and Z. Sun, Humanoid Walking Gait Optimization Using GA-Based Neural Network, *Advances in Natural Computation*, pp. 252-261, Springer Berlin/Heidelberg, Changsha, China, 2005.
- [17] J. Lima, J. Gonçalves, P. Costa and A. Moreira, Humanoid Robot Gait Planning Resorting to an Adaptive Simulated Annealing Algorithm, *10th Conference on Autonomous Robot Systems and Competitions*, Portugal, 2010.