

# HUMANOID REALISTIC SIMULATOR: *The servomotor joint modeling*

José L. Lima, José A. Gonçalves

*Polytechnic Institute of Bragança, Bragança, Portugal*  
*jllima@ipb.pt, goncalves@ipb.pt*

Paulo G. Costa, A. Paulo Moreira

*Faculty of Engineering of University of Porto, Porto, Portugal*  
*paco@fe.up.pt, amoreira@fe.up.pt*

Keywords: Humanoid, Servomotor, Modeling, Simulation.

Abstract: This paper presents a humanoid servomotor model that can be used in simulations. Once simulation is a tool that reduces the software production time, it was developed a realistic simulator that own the humanoid features. Based on a real platform, the simulator is validated when compared with the reality.

## 1 INTRODUCTION

Recent research in biped robots has resulted in a variety of prototypes that resemble their biological counterparts. Legged robots have several advantages, they can move in rugged terrains, they have the ability to choose optional landing points, and two legged robots are more suitable to move in human environment (Suzuki and Ohnishi, 2006).

The simulator should capture the essential characteristics of the real system. In this paper, the servomotor model that powers the real humanoid joints is addressed. The model of a Dynamixel AX-12 servomotor and its characteristics are found by an iterative method based on a realistic simulator, the *SimTwo* (Costa, 2009).

There are several simulators with humanoid simulation capability, like Simspark, Webots, MURoSimF, Microsoft Robotics Studio, YARP: Yet Another Robot Platform (Wang et al., 2006) and OpenHRP3 (Ope, 2009), meanwhile, the *SimTwo*, as a generic simulator, allows to simulate different types of robots and allows the access to the low level behaviour, such as dynamical model, friction model and servomotor model in a way that can be mapped to the real robot, with a minimal overhead. This simulator deals with robot dynamics and how it reacts for several controller strategies and styles. Using a realistic simulator can be the key for reducing the development time of robot control, localization and navigation software. It is not an easy task to develop such simulator due to the in-

herent complexity of building realistic models for the robot, its sensors and actuators and their interaction with the world (Browning and Tryzelaar, 2003).

The purpose of developing such simulator is to produce a personalized and versatile tool that will allow the development and validation of the robot's software thereby reducing considerably the development time.

The paper is organized as follows: Initially, the real robot (which is the basis of the simulator) and its main control architecture are presented. Then, section 3 presents the developed simulator where the servomotor model was developed. Further, section 4 presents the validation of the simulator by comparing its results with the real robot. Finally, section 5 rounds up with the conclusions and future work.

## 2 REAL HUMANOID

There are several humanoid robots kits available. The commercially available Bioloid robot kit, from Robotis, is the basis of the used humanoid robot and the overview of the proposed biped robot is shown in Figure 1.

The servo motors are connected to the central processing unit through a serial 1Mbps network. Next subsection presents the physical robot in which the developed humanoid simulator was based.

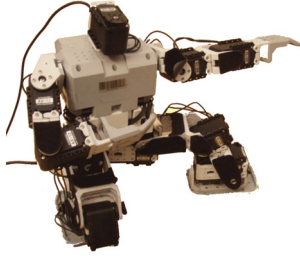


Figure 1: Real humanoid robot (from Bioloid).

## 2.1 Main architecture

The presented humanoid robot is driven by 19 servo motors (AX-12): six per leg, three in each arm and one in the head. Three orthogonal servos set up the 3DOF (degree of freedom) hip joint. Two orthogonal servos form the 2DOF ankle joint. One servo drives the head (a vision camera holder). The shoulder is based on two orthogonal servos allowing a 2DOF joint and elbow has one servo allowing 1DOF. The total weight of the robot (without camera and on board computer) is about 2 kg and its height is 38 cm.

To control the real humanoid robot a high level application (presented in Figure 2) was developed. This application is independent from the simulator, although it allows to acquire and share some real robot data with simulator.

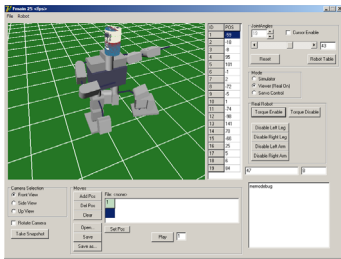


Figure 2: Real humanoid robot control application.

## 3 Simulation Model

Designing the robot's behaviour without real hardware is possible due to a physics-based simulator implementation. The physics engine is the key to make simulation useful in terms of high performance robot control (Browning and Tryzelaar, 2003). The dynamic behaviour of robot (or multiple robots) is computed by the ODE (Open Dynamics Engine), a free library for simulating rigid body dynamics.

## 3.1 Simulator architecture

The simulator architecture is based on the real humanoid robot. The body masses and dimensions are used to build a humanoid simulator similar to the real one. The communication architecture in real robot brings some limitations to control loop such as lag time. The developed simulator enhances these properties. The same architecture levels of the real robot are implemented in the simulator. At the lowest level, the servo motor model includes the control loop, just like the real servomotors. At the highest level, some predefined joint states are created based on several methods presented on literature (Kajita et al., 2006) and (Zhang et al., 2008). At the middle level, an optimized trajectory controller that allows to minimize the energy consumption is introduced as presented in Figure 3 (Lima et al., 2008b).

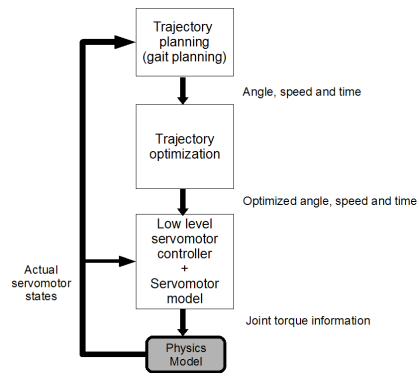


Figure 3: Simulator main architecture.

Next subsections describe the servomotor model that is used in the simulator where the electrical, friction and controller models are presented.

## 3.2 DC Motor model

The servomotor can be modeled by a DC motor model, presented in Figure 4, where  $U_a$  is the converter output,  $R_a$  is the equivalent resistor,  $L_a$  is the equivalent inductance and  $e$  is the back *emf* voltage as expressed by equation 1 (Conceição et al., 2006).

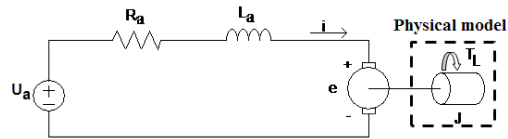


Figure 4: DC motor electric model.

$$U_a = e + R_a i_a + L_a \frac{\partial i_a}{\partial t} \quad (1)$$

The motor can deliver a  $T_S$  torque and its load has a  $J$  moment of inertia that will be presented by the physical model ODE. Current  $i_a$  can be related with developed torque  $T_D$  through equation 2 and the back *emf* voltage can be related with angular speed through equation 3, where  $K_s$  is a motor parameter that can be found by an experimental setup as presented in subsection 3.3 (Bishop, 2002).

$$T_D(t) = K_s i(t) \quad (2)$$

$$e(t) = K_s \omega(t) \quad (3)$$

In fact, the real developed torque (useful) that will be applied to the load ( $T_S$ ) is the motor torque subtracted by the friction torque ( $T_F$ ) as presented in equation 4. The friction torque is discussed in subsection 3.5.

$$T_S = T_D - T_F \quad (4)$$

### 3.3 DC motor model measurements

It was used the AX-12 servomotor from Dynamixel as the base of the humanoid simulator articulations. The  $R_a$  and  $L_a$  values can be directly measured ( $R_a=8 \Omega$  and  $L_a=5 mH$ ). The  $K_s$  motor parameter can be found by an indirect estimation. For several angular speeds, it can be measured the *emf* voltage while motor is open circuit.

Figure 5 shows the graphical data of the  $K_s$  line and its trend line. The average value of 13 measures for  $K_s$  (line slope) is about  $0.006810 V.s/rad$ .

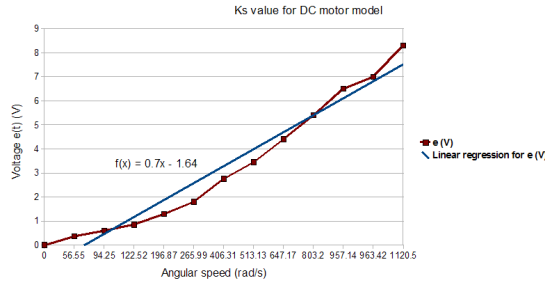


Figure 5:  $K_s$  value for DC motor model.

### 3.4 DC motor nonlinearities

In a way to better model the real system, where variables cannot assume all values, there must be some limits applied to some quantities. The first one, is the voltage applied to the supply terminals  $U_a$ . This voltage is limited by the batteries voltage. Further, current  $i$  is limited by the drive electronics once it is related

to the torque through equation 2 (torque limit is programmed in the real servomotor). Current gradient is also implicitly limited by the presence of  $L_a$ . Figure 6 shows the block diagram computed by the simulator. There is also a internal gradient limitation to ensure some numerical stability specially when the integration step period is significantly slower than electric dynamic.

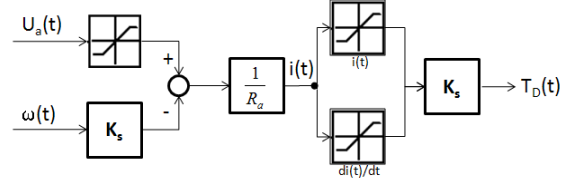


Figure 6: Servomotor model block diagram computation.

### 3.5 Friction model

The friction model has two terms: the static and dynamic friction as presented in equation 5.

$$T_F = F_c \text{sign}(\omega) + B_v \omega \quad (5)$$

The first one can be modeled as the sign function (with  $F_c$  constant) and the second one can be modeled as a linear function with slope  $B_v$ .

The sum of this two components ( $T_F$ ), the final friction model, is shown in figure 7.

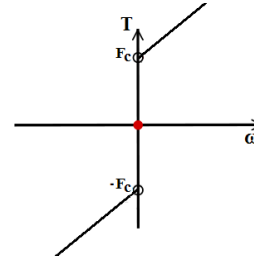


Figure 7: Servomotor model nonlinearities.

Figure 8 shows the friction model implemented in the servomotor model.

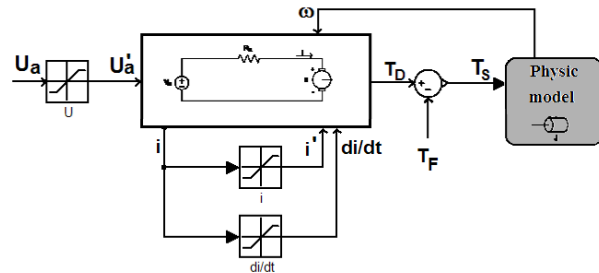


Figure 8: Servomotor model with friction model.

The  $F_c$  and  $B_v$  constants are found using simulator scanning several possible values minimizing the error

with the real system during an arm fall from 90 to 0 degrees. The error surface function can be minimized.

As result,  $Bv=0.01278 \text{ N.m.s/rad}$  and  $Fc=0.0000171 \text{ N.m}$  where found as the best values. These constants allow the simulator to follow reality very closely as presented in figure 9 where an arm falls from 45, 90 and 135 degrees for both robots.

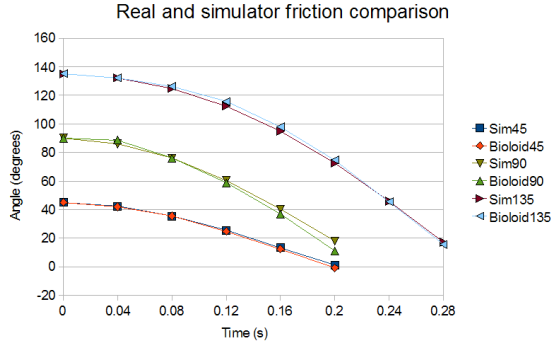


Figure 9: Real robot and simulator friction comparison.

### 3.6 Low-level controller

The low level controller resembles the closed loop controller in the real robot implemented by the servomotor manufacturer. This controller accepts, from the higher level, the angle and angular speed references that define a trajectory. The controller type present in the Dynamixel isn't specified. However, there are some possible models for the controller, such as PID or state feedback. Considering a state feedback controller and assuming  $K_{\theta}^i$  the position error gain and  $K_{\omega}^i$  the speed error gain for each  $i$  joint, the equation 6 defines the servomotor input ( $U_a$ ) that keeps the desired reference conditions for each  $i$  joint. The output torque is computed by the physic model as presented in Figure 10.

$$U_a^i(t) = K_{\theta}^i(\theta_{ref}^i(t) - \theta^i(t)) + K_{\omega}^i(\omega_{ref}^i(t) - \omega^i(t)) \quad (6)$$

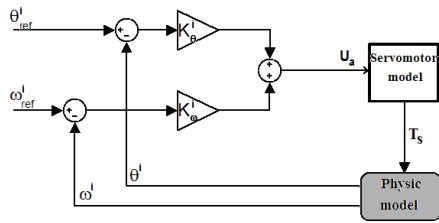


Figure 10: Low level controller.

The position error and the speed error gains can be computed resorting to a least squares approach. Hav-

ing the real robot arm step response (from 0 to 90 degrees), it is possible to scan several gains and to determine the quadratic error between the real servo and the humanoid simulator joint for each solution. The one that fulfill the lowest quadratic error is the chosen gains to the controller. Figure 11 shows the quadratic error for the several solutions.

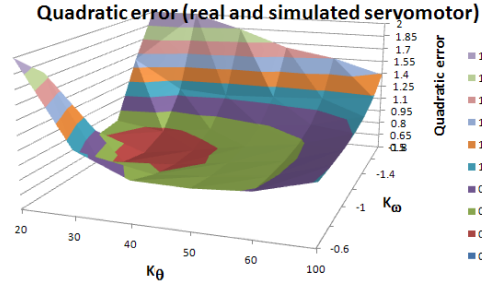


Figure 11: Quadratic error for real and simulator servomotor.

The lowest quadratic error ( $0.66 \text{ degrees}^2$ ) occurs for  $K_{\theta}^i$  equals 30 and for  $K_{\omega}^i$  equals -1.2. These gains allow to obtain the step response very close to the real servo. Figure 12 shows both step responses.

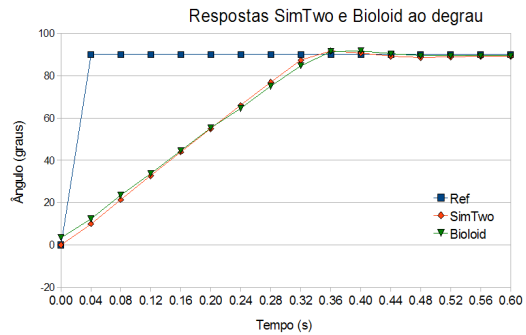


Figure 12: Real and simulator servomotor step response.

### 3.7 Servomotor model

With the previous results, it is possible to create the servomotor model including the DC motor, the friction and the controller models (with its nonlinearities) that will represent the real servo motor in the simulator. The final model is presented in figure 13.

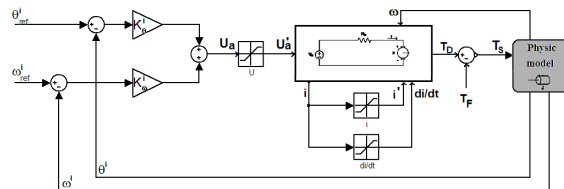


Figure 13: Simulator servomotor model.

## 4 Simulator validation

To validate the humanoid simulator model it is required to implement the same control signal to both robots and to analyze the behaviors. Predefined trajectory states, that allow robot to walk, are based on the Zero Moment Point (ZMP) method. Figure 14 shows the sequence during walk movements for both robots (real at left and simulator at right). It is possible to observe that both robots exhibit very similar behaviours.

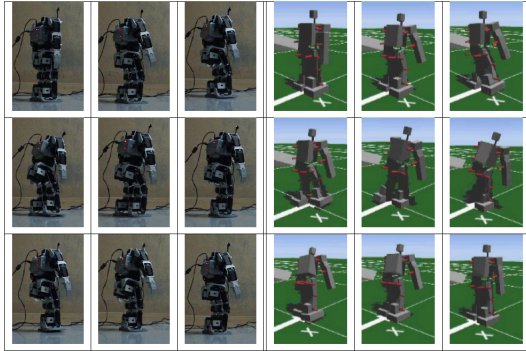


Figure 14: Real and simulator robots walking with the same predefined gaits.

With this walking movement, it can be acquired all joint angles for both robots. Figure 15 shows a knee angle for real and simulated robots, in a walk movement, that shows simulator behaves as real robot. Moreover, the power consumption comparison between real and simulated robot is presented in (Lima et al., 2008a).

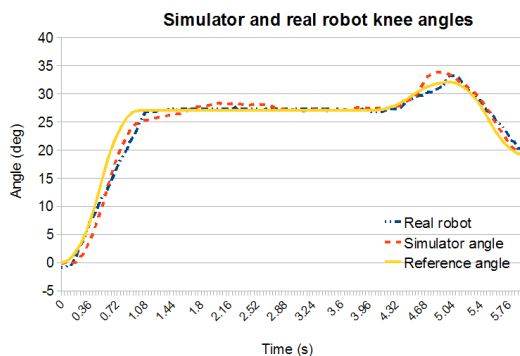


Figure 15: Real and simulator robots knee angles during a walk movement.

## 5 Conclusions and future work

A simulator that allows a humanoid robot simulation capability is addressed and validated. The joints that emulate the real articulations are based on a realistic servomotor model. The proposed servomotor

model was implemented in the developed simulator, *SimTwo*. This simulator is based in a real platform. The friction model and closed loop controller gains are found based on the real robot behaviour. It allows to search the optimal values for friction and controller gains based on a heuristic approach. The validation with the real humanoid robot allows to confirm the proposed servomotor model. As future work, the simulator can be useful to find several parameters that optimize a desired condition such as energy consumption in the walk movement and further applied to the real robot.

## REFERENCES

- (2009). Open architecture humanoid robotics platform. <http://www.openrtp.jp/openhrp3/en/index.html>.
- Bishop, R. (2002). *The Mechatronics Handbook*. CRC Press, New York.
- Browning, B. and Tryzelaar, E. (2003). Ubersim: A realistic simulation engine for robotsoccer. In *Proceedings of Autonomous Agents and Multi-Agent Systems, AAMAS'03*.
- Conceição, A., Moreira, A., and Costa, P. (2006). Dynamic parameters identification of an omni-directional mobile robot.
- Costa, P. (2009). Simtwo webpage. <http://www.fe.up.pt/~paco/wiki/>.
- Kajita, S., Morisawa, M., Harada, K., Kaneko, K., Kanehiro, F., Fujiwara, K., and Hirukawa, H. (2006). Biped walking pattern generator allowing auxiliary zmp control. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2994–2999.
- Lima, J., Gonçalves, J., Costa, P., and Moreira, A. (2008a). Realistic behaviour simulation of a humanoid robot. In *8th Conference on Autonomous Robot Systems and Competitions*.
- Lima, J., Gonçalves, J., Costa, P., and Moreira, A. (2008b). Realistic humanoid robot simulation with an optimized controller: a power consumption minimization approach. In *11th International Conference on Climbing and Walking Robots*, pages 1242–1248.
- Suzuki, T. and Ohnishi, K. (2006). Trajectory planning of biped robot with two kinds of inverted pendulums. In *Proceedings of 12th International Power Electronics and Motion Control Conference*, pages 396–401.
- Wang, X., Lu, T., and Zhang, P. (2006). Yarp: Yet another robot platform. *International Journal of Advanced Robotic Systems*.
- Zhang, L., Zhou, C., and Xiong, R. (2008). A lie group formulation for realtime zmp detection using force/torque sensor. In *Proceedings of the 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, pages 1250–1257.