Sensor and actuator stochastic modeling of the Lego Mindstorms NXT educational Kit

José Gonçalves, José Lima, Paulo Malheiros and Paulo Costa

***Abstract*— This paper describes the sensor and actuator stochastic modeling of Lego Mindstorms NXT educational kit. It will be also presented the experimental setups applied to obtain each sensor and actuator parameters. The introduction of an industrial robot in some of the experimental setups allowed to increase the speed, repeatability and to reduce errors in the process of sensor data collecting.**

I. Introduction

Lego Mindstorms is a powerful educational tool [1] [2] [3], being used by the authors in many activities. It has been used for teaching mobile robots introductory concepts to secondary school and undergraduate students and in several demonstrations. The secondary school students while attending summer courses prototype their own robots and develop robot software based on the Lego Mindstorms educational software. The undergraduate students while attending robotics classes also prototype their own robots, with the difference that the developed robot software is done resorting to high level programming languages. It has also been used in demonstrations with the goal of captivate and inform secondary school students, concerning the areas that involve technology. As an example it is shown in Figure 1 a demonstration at the Live Science Center of Bragança where are shown two mobile robot prototypes based on the Lego Mindstorms NXT kit.

The goal of the sensor and actuator modeling of the Lego Mindstorms kit is to develop its realistic simulation in order to produce robot software without accessing to real hardware. For the simulation of the NXT robots it was used the *SimTwo* (available for download at [4]), being a versatile robot simulation environment that allows rapid test and design of differential, omnidirectional, industrial robots, humanoids, etc., developed in Object Pascal. Design behavior without real hardware is possible due to a physics-based simulator implementation, the dynamic behavior of the simulated robot is computed by the ODE (Open Dynamics Engine), a free library for simulating rigid body dynamics [5] [6]. The simulator architecture is based on the real Lego NXT robot. The body masses and dimensions are accurately measured in order to build a realistic simulated robot. *SimTwo* has a set of predefined components such as motors where the model specified in Section II is inputed. The robots look

José Gonçalves and José Lima are with the Polytechnic Institute of Bragança, Department of Electrical Engineering, Bragança, Portugal {goncalves,jllima}@ipb.pt

Paulo Malheiros and Paulo Costa are with the Faculty of Engineering of University of Porto, DEEC, Porto, Portugal {paulo.malheiros,paco}@fe.up.pt

All the authors are with INESC PORTO

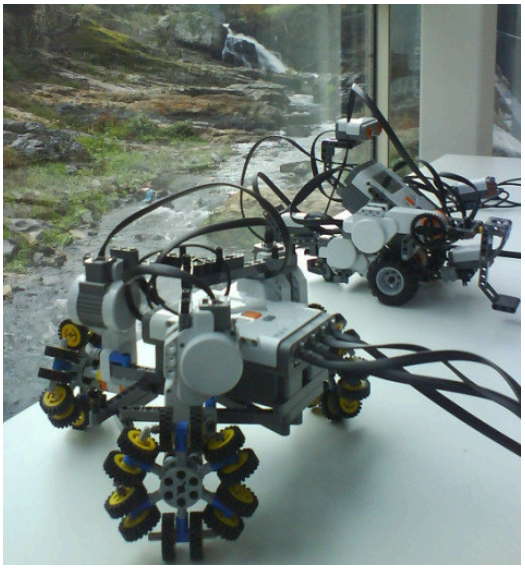

Fig. 1. Demonstration at the Live science center of Bragança

and behavior are defined in XML format files. The virtual world is represented using the GLScene components [7], these provide a simple implementation of OpenGL. The NXT robots are defined using boxes and cilynders, as shown in Figure 2. These are the objects used by the ODE to determine the collisions and friction. *SimTwo* allows the replacement of the solids by any 3DS model which gives a realistic look to the robots, as shown in Figure 3.

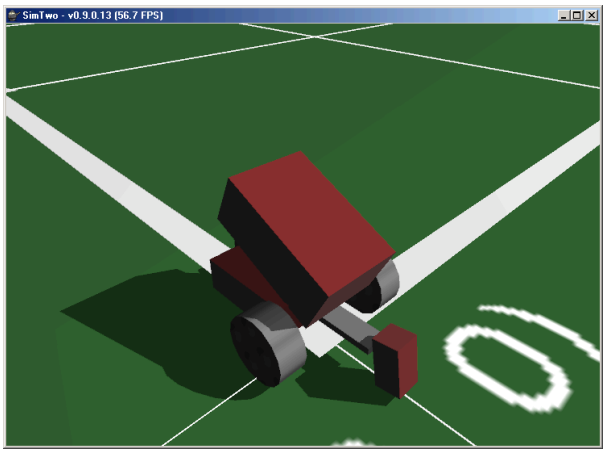

Fig. 2. NXT robot modeled in the *SimTwo*

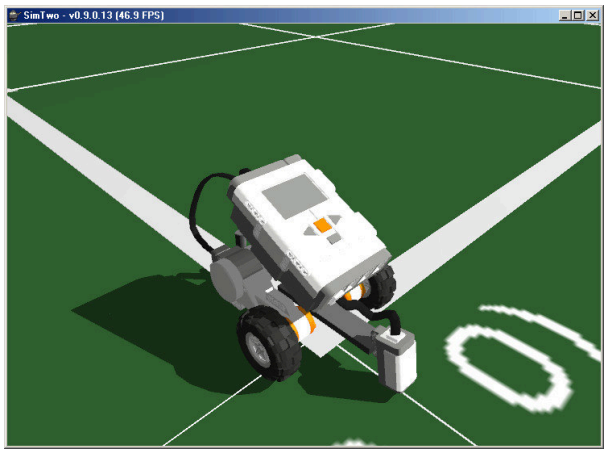

Fig. 3. NXT robot modeled in the *SimTwo* with 3DS models

II. Actuator Modeling

The Lego Mindstorms NXT kit provides three servomotors with built-in rotation sensor and a gear ratio of 1:48. A Lego Mindstorms NXT servomotor is shown in Figure 4.

Fig. 4. Lego Mindstorms NXT servomotor.

The servomotor can be resumed to a DC motor model, presented in Figure 5, 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 (electromotive force) voltage as expressed by equation (1) [8].

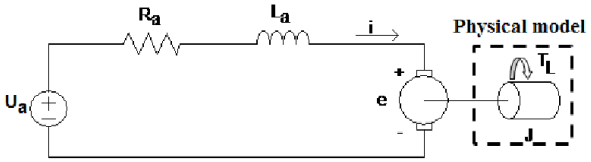

Fig. 5. DC motor electric model.

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

The motor can supply a torque T_L and the load has a moment of inertia J that will be computed by the physical model of the ODE. Current i_a can be correlated with the developed torque T_d through equation (2) and the back emf voltage can be correlated 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 II-.1 [9].

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

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

In fact, the real developed torque (useful) that will be applied to the load (T_L) is the developed torque subtracted by the friction torque (T_c).

$$T_L = T_d - T_c \quad (4)$$

1) DC motor model measurements: It was used the NXT Lego Mindstorms motor as the base of the simulator. The R_a and L_a values can be directly measured (R_a=7.6 Ω and L_a= 4.88 mH). The K_s motor parameter can be found by an indirect measure. For several angular speeds, it can be measured the emf voltage while the motor is in open circuit. Table I presents the data for 7 measures.

rot (rot/min)	rot (rad/s)	e (V)	k=e/rot
0	0	0	
66	6.91	3.4	0.491
115	12.04	6.0	0.498
155	16.23	7.9	0.487
195	20.42	9.8	0.479
233	24.39	11.9	0.487
265	27.75	13.85	0.499

TABLE I
SPEED AND EMF VOLTAGE.

Figure 6 shows graphical data of the K_s line and its trend line. The average value for K_s (line slope) is about 0.4919 $V.rad/s$. It is possible to observe that the error in the origin for the presented approximation it is negligible, being only 0.0218 Volt. Despite being a small error it is possible to increase the precision in the K_s parameter estimation, forcing the trend line to pass in the origin. The obtained value for K_s is nearly 0.4908 V.rad/s.

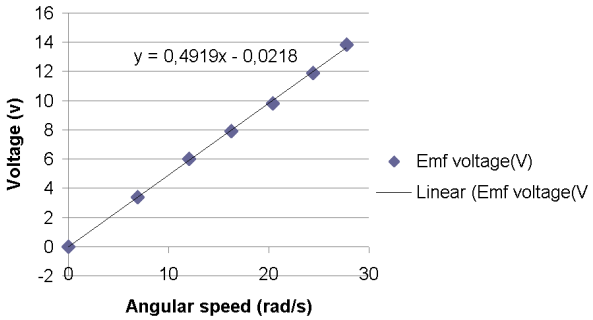

Fig. 6. K_s value for DC motor model.

In order to calculate the friction it was obtained the motor angular velocity for different voltages at its terminals. After a linear regression it was obtained equation (5).

$$rot = 1.9248v - 0.2519 \quad (5)$$

Where rot is the angular velocity in rad/s and v is the voltage at the motor terminals.

Assuming that the developed torque (T_d) is equal to the torque generated by the static friction (T_c) it is possible, resorting to the linear regression presented in equation (5), to obtain the necessary voltage to equal the static friction (v_c), by replacing rot by zero. For this situation it was not necessary to include in the model the viscous friction, because it only exists for angular velocities different from zero. As there is no movement there is no voltage generated by the emf and as the current has no variation it is possible to obtain the produced torque by the static friction resorting to equations (1) and (2), resulting in equation (6). The torque due to the static friction has a numerical value of 3.558E-4 Nm.

$$T_c = K_s \frac{v_c}{R_a} \tag{6}$$

After obtaining the static friction there is only left, to complete the motor model, the parameter estimation that relates the viscous friction with the motor angular velocity, as shown in equation (7).

$$T_\omega = B\omega \tag{7}$$

Placing the motor spinning without load the developed torque will given by equation (8).

$$T_d = T_c + B\omega \tag{8}$$

As the developed torque is proportional to the current as shown in equation (2) it is possible to conclude, from equation (8), that the current can be given by equation (9).

$$i_a = \frac{T_c + B\omega}{K_s} \tag{9}$$

The voltage applied to the motor terminals can be given by equation (1), but as in steady state the current variations can be despised, it is obtained equation (10) for the motor terminal voltages.

$$U_a = K_s\omega + R_a i_a \tag{10}$$

where e was replaced by $K_s\omega$ as exemplified in equation (3).

From equations (9) and (10) it is obtained equation (11).

$$\omega = \frac{U_a - \frac{R_a T_c}{K_s}}{\frac{R_a B}{K_s} + K_s} \tag{11}$$

Minimizing the sum of the absolute error between the real angular velocity and the obtained by the model (equation (11)), it is obtained a numerical value for B (1.92E−3).

The motor estimated parameters are shown in Table II.

The different obtained motor models and the errors obtained for each model can be observed in the graphics shown in Figure 7 and 8.

Parameters	Value SI units
K_s	0.4908
T_c	3.558E-4
B	1.92E−3
R_a	7.6
L_a	4.88E−3

TABLE II
MOTOR PARAMETERS

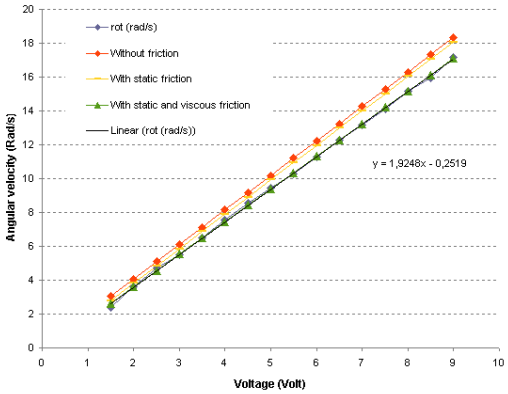

Fig. 7. Different obtained models.

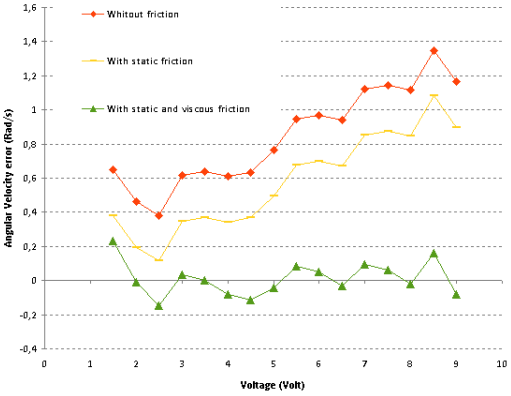

Fig. 8. Angular velocity error.

2) DC motor nonlinearities: In a way to map the reality closer, where variables cannot assume all values, the model must have some limitations. The first one, the voltage applied to the motor terminals U_a. This voltage should be limited to the batteries voltage. Further, current i should be limited once it is related to the torque through equation (2). Figure 9 shows the limitations presented in the servomotor model.

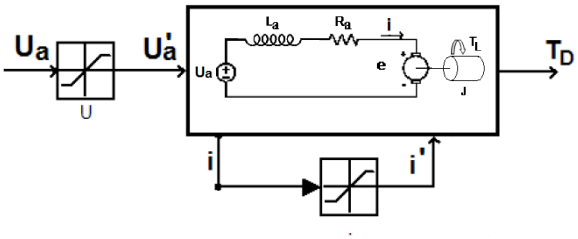

Fig. 9. Servomotor model nonlinearities.

III. SENSORS MODELING

Lego Mindstorms NXT provides four different sensors, a touch sensor, a sound sensor, a light sensor and an ultrasonic sensor (Figure 10). The light sensor can detect reflected and ambient light, it can create a robot capable of moving along a line drawn on the ground or a robot able to sort by color. For this reason, a red ball and blue ball are supplied with the kit. The ultrasonic sensor measures distance to an obstacle and the touch sensor detects when it is being pressed by something and when it is released again. Finally, the sound sensor measures the decibel level of the room. In this section it will be presented the modeling of the NXT sensors with the exception of the sound sensor.

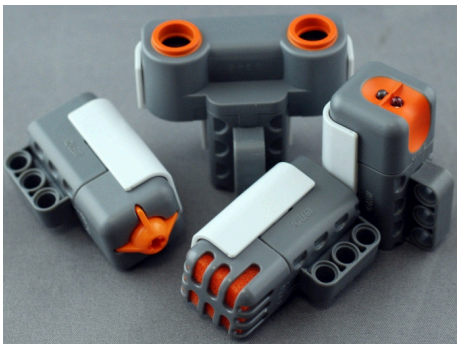

Fig. 10. Lego NXT Mindstorms sensors.

A. Light sensor

The light sensor enables the robot to distinguish between light and dark. It can read the light intensity in a room and measure the light intensity of colored surfaces. The light sensor reacts as the characteristic shown in Figure 11, for measures across a gradient printed in an A4 page (Figure 12). It was observed that the rate that the sensor data evolves remains always the same, but an offset value must be added depending on the room lighting.

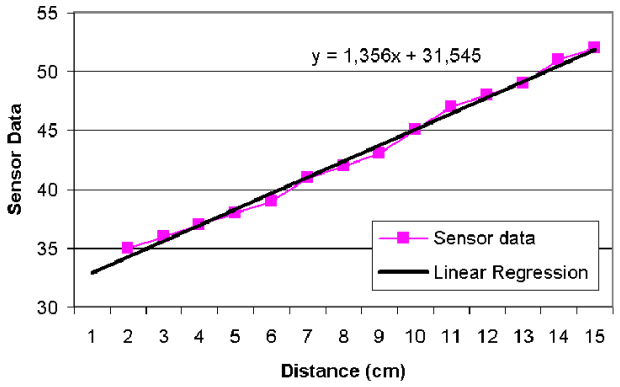

Fig. 11. Light sensor characteristic.

Fig. 12. Robot reading gradient.

B. Touch sensor

In order to model the touch sensor it was necessary to collect a considerable amount of data, for this task it was used an industrial robot to move the sensor. Industrial robots allow executing repetitive operations normally performed by human operators, without getting bored and without loosing precision [14] [15]. The used experimental setup is shown in Figure 13. The used industrial robot is the HP6 from Motoman with the NX100 controller available at the Laboratory of Control and Robotics of the Faculty of Engineering of the University of Porto. The robot controller provides a remote mode with the communication based in the protocol described in [16], which uses a standard TCP/IP connection. The touch sensor is pressed against a weight balance in order to model the necessary force and the sensor spring compression to force the sensor to a state change. It was observed that there are significant differences from sensor to sensor but a high repeatability for samples of the same sensor. The applied force to obtain a state transition is presented in grams in Tables III and IV. The necessary spring compression necessary to force a sensor state transition is presented in mm in Tables V and VI. For the compression the standard deviation is only different from zero for the state change off to on for the sensor 8 and for the state change on to off for the sensor 6, being for both cases $1.6\text{E}{-3}$. It was observed that there is hysteresis in the state change, as an example it is shown the state change for sensor 1 in Figure 14.

C. Ultrasonic sensor

The ultrasonic sensor is one of the most common forms of distance measurement used in mobile robotics and a variety of other applications [10] [11] [12] [13]. A speaker (transducer) is used to emit a short burst of sound (ping). The sound wave travels through the air and reflects off a target back to the transducer (echo). By measuring the time of flight between ping and echo detection, one can calculate the distance between the target and transudcer.

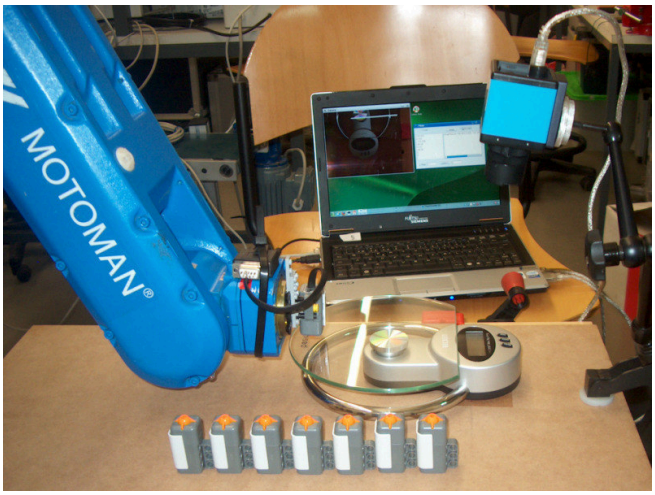

Fig. 13. Industrial robot moving the touch sensor.

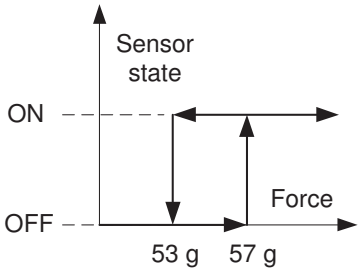

Fig. 14. State transition sensor 1.

In order to obtain the necessary data to model the distance sensor it was also used an industrial robot for the same reasons as presented in the previous section. The experimental setup is shown in Figure 15. The industrial robot moved the sensor for different distances to the floor, from 10 to nearly 1550 mm in one mm steps, as shown in Figure 16. It was observed that between 250 and 1500 mm the sensor is able to determine correctly the measures, increasing the measure noise with distance. The sensor characteristic for this interval can be modeled by the trend line $y = mx + b$, with $m = 1.0055$ and $b = 0.013$. For distances lower than 40 mm the sensor is unable to measure correctly. From 40 to 250 mm the sensor characteristic it is not linear, where most of the measurements present an offset error of 20 mm.

The sensor behavior is affected by its position angle when facing an obstacle. To model this feature a test was made where the sensor was rotated, while facing an obstacle, for a constant distance. The measurements were made between the distances 138 and 1538 mm, spaced by 100 mm. For each of these distances the angle was incremented by 5 degrees until no valid echoes were received by the sensor receptor. For each sensor position 64 samples where registered. As a result of this experiment the ultrasonic sensor beam pattern was obtained as shown in Figure 17. The beam pattern increases for distances from 138 to 438 mm, decreasing progressively from this point to the last measured distance.

Samples	1	2	3	4	5	6	7	8	μ	σ
Sensor 1	57	57	57	57	57	57	57	57	57	0
Sensor 2	62	62	63	63	63	63	63	63	62.6	0.38
Sensor 3	71	71	71	71	71	71	71	71	71	0
Sensor 4	66	67	67	67	67	67	67	67	66.9	0.33
Sensor 5	62	62	62	62	63	62	63	62	62.3	0.43
Sensor 6	71	71	71	71	71	71	71	71	71	0
Sensor 7	50	49	49	50	49	50	50	50	49.6	0.48
Sensor 8	52	52	52	52	52	53	52	52	52.1	0.33

TABLE III

APPLIED FORCE STATE TRANSITION OFF TO ON

Samples	1	2	3	4	5	6	7	8	μ	σ
Sensor 1	53	53	53	53	53	53	53	53	53	0
Sensor 2	60	60	60	60	60	60	60	60	60	0
Sensor 3	67	67	67	67	67	67	67	68	67.1	0.11
Sensor 4	66	66	66	66	66	66	66	66	66	0
Sensor 5	58	58	58	58	58	58	57	58	57.9	0.11
Sensor 6	67	67	67	67	67	67	67	68	67.1	0.11
Sensor 7	48	48	46	48	48	48	48	48	48.8	0.44
Sensor 8	52	52	52	52	52	52	52	52	52	0

TABLE IV

APPLIED FORCE STATE TRANSITION ON TO OFF

The best sensor performance was registered for 438 mm, where valid measurements were obtained for a maximum angle of 55 degrees.

An extra test was made in order to determine the position of the sensor emitter and receptor, using two ultrasonic sensors working simultaneously. Having the sensor emitter blocked, if a valid echo is received from the other sensor, then the receiver is found and consequently the emitter. The ultrasonic distance sensor is shown in Figure 18.

IV. CONCLUSIONS AND FUTURE WORK

This paper described the sensor and actuator stochastic modeling of Lego Mindstorms NXT educational kit. It also

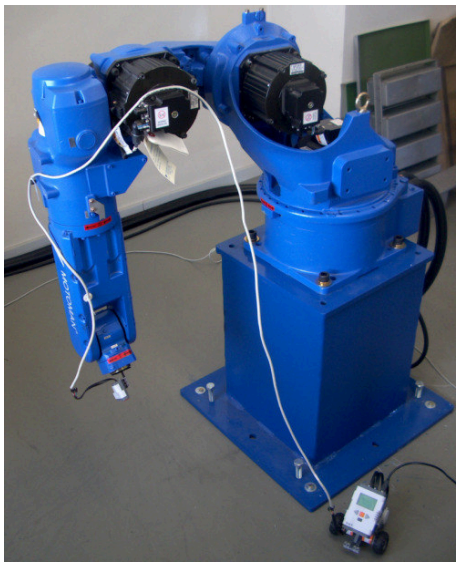

Fig. 15. Industrial robot placing the ultrasonic sensor.

Samples	1	2	3	4	5	6	7	8
Sensor 1	2.11	2.11	2.11	2.11	2.11	2.11	2.11	2.11
Sensor 2	2.22	2.22	2.22	2.22	2.22	2.22	2.22	2.22
Sensor 3	2.22	2.22	2.22	2.22	2.22	2.22	2.22	2.22
Sensor 4	2.02	2.02	2.02	2.02	2.02	2.02	2.02	2.02
Sensor 5	2.11	2.11	2.11	2.11	2.11	2.11	2.11	2.11
Sensor 6	2.22	2.22	2.22	2.22	2.22	2.22	2.22	2.22
Sensor 7	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
Sensor 8	2.21	2.21	2.21	2.21	2.31	2.21	2.21	2.21

TABLE V

SPRING COMPRESSION STATE TRANSITION OFF TO ON

Samples	1	2	3	4	5	6	7	8
Sensor 1	1.80	1.80	1.80	1.80	1.80	1.80	1.80	1.80
Sensor 2	2.01	2.01	2.01	2.01	2.01	2.01	2.01	2.01
Sensor 3	1.91	1.91	1.91	1.91	1.91	1.91	1.91	1.91
Sensor 4	1.81	1.81	1.81	1.81	1.81	1.81	1.81	1.81
Sensor 5	1.80	1.80	1.80	1.80	1.80	1.80	1.80	1.80
Sensor 6	2.02	2.00	2.00	2.00	2.00	2.00	1.91	2.00
Sensor 7	1.70	1.70	1.70	1.70	1.70	1.70	1.70	1.70
Sensor 8	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00

TABLE VI

SPRING COMPRESSION STATE TRANSITION ON TO OFF

presented the experimental setups applied to obtain each sensor and actuator parameters. The introduction of an industrial robot in some of the experimental setups allowed to increase the speed, repeatability and reduced errors in the process of sensor data collecting.

The correct modeling of the sensors and actuators enables a more realistic simulation. For the simulation of the NXT Robot it was used the *SimTwo*, being a versatile robot simulation environment that allows rapid test and design of differential, omnidirectional, industrial robots, humanoids, etc., developed in Object Pascal. The actuator and the light sensor are already implemented in the *SimTwo* platform. As future work it will also be implemented the touch and ultrasonic sensors.

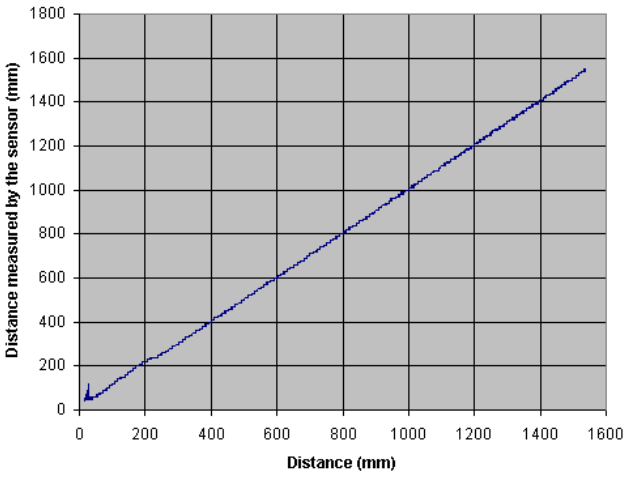

Fig. 16. Ultrasonic distance sensor characteristic.

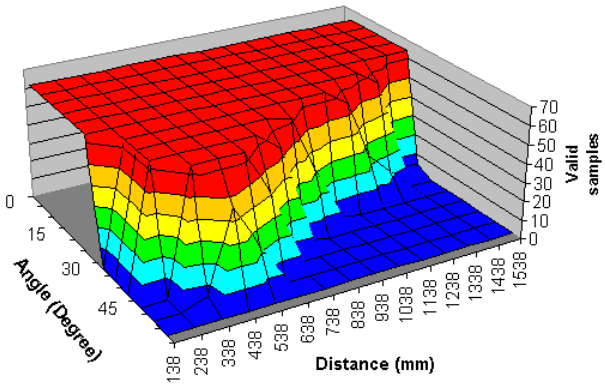

Fig. 17. Ultrasonic sensor beam pattern.

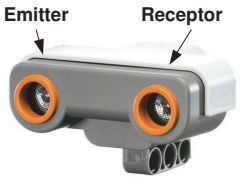

Fig. 18. Ultrasonic sensor.

REFERENCES

[1] J. Gonçalves, J. Lima, P. Costa, *Rapid prototyping of mobile robots extending Lego mindstorms platform,* 7th IFAC Symposium on Advances in Control Education, Madrid, 2006.

[2] Á. Valera, M. Weiss, M. Vallés, J. Díez, *Proposal of a low-cost mobile robot control laboratory experiment,* 7th IFAC Symposium on Advances in Control Education, Madrid, 2006.

[3] P. Gawthrop, E. McGookin, *Using Lego In Control Education,* Plenary Lecture 7th IFAC Symposium on Advances in Control Education, Madrid, 2006.

[4] "*SimTwo*" http://www.fe.up.pt/∼paco/wiki, 2009.

[5] "ODE" http://www.ode.org/, 2009.

[6] B. Browning, E. Tryzelaar, *UberSim: A Realistic Simulation Engine for RobotSoccer,* Proceedings of Autonomous Agents and Multi-Agent Systems, Melbourne, 2003.

[7] "GLScene" http://glscene.sourceforge.net/wikka/HomePage, 2009.

[8] A. Conceição, A. Moreira, P. Costa, *Dynamic Parameters Identification of an Omni-directional Mobile Robot,* Proceedings of the International Conference on Informatics in Control, Automation and Robotics, Setbal 2006.

[9] R. Bishop, *The Mechatronics Handbook,* CRC Press, New York, 2002.

[10] J. Borenstein, H. Everett, L. Feng, *'Where am I?' Sensors and Methods for Mobile Robot Positioning,* Technical Report, The University of Michigan, 1996.

[11] P. Silva, *Navegação Acústica em Ambientes Estruturados,* Master Thesis, FEUP, 1995.

[12] F. Zhao, H. Guo, K. Abe, *Mobile robot localization using two sonar sensors and one natural landmark* Proceedings of the 37th SICE Annual Conference, 1998.

[13] H. Yang, J. Borenstein, D. Wehe, *Sonar-based obstacle avoidance for a large, non-point, omnidirectional mobile robot,* in Proc of the International Conference in Nuclear and Hazardous Waste Management, Sept. 2000.

[14] M. Groover, M. Weiss, R. Nagel, N. Odrey, *Industrial Robotics: Technology, Programming, and Applications* McGraw-Hill, 1995.

[15] H. Colestock, *Industrial Robotics: Selection, design and maintenance* McGraw-Hill, 2005.

[16] Yaskawa Electric Corporation (Japan), *NX 100 applications options - Instructions for data transmission function*, August 2004.