

A new educational tool for Bioloid Kit

Gonzalo Zabala¹, Ricardo Moran¹, Sebastián Blanco¹

¹ Centro de Altos Estudios en Tecnología Informática, Universidad Abierta Interamericana,
Buenos Aires, Argentina
{Gonzalo.Zabala, Ricardo.Moran, Sebastian.Blanco}@uai.edu.ar

Abstract. The aim of this paper is to show the implementation of a new educational tool within Physical Etoys that enables the use of the Bioloid kit in the classroom. A new firmware was made and some of the objects of the kit were virtually represented with Physical Etoys in order to create a more intuitive approach. In addition, a simplified example of inverse kinematics with a humanoid for high school students is proposed.

Keywords: physical etoys; educational robotics; bioloid; inverse kinematics; trigonometry

1 Introduction

Since the emergence of educational robotics kits, the presence of particular technological materials in the classrooms has been growing. In Argentina, there are three provinces that have already implemented the use of this resource in their classrooms. To achieve the same aim, the rest of the provinces begun with pilot programs. Another proof of the presence of such materials is the surging of both nationals and internationals contests such as the Roboliga, the Argentinian Robotics Olympics and the Robocup Junior, which is an international competition where students from around the world gather to share their experiences in educational robotics. In the Robocup Junior, particularly, there is a category known as Dance [1], that brings together the largest number of participants in which is common the use of humanoids within a developed choreography.

In summary, the use of robots as an educational source inside the classrooms is no longer a novelty. As an innovation, we propose the inclusion of humanoids in the kits used at school. What special qualities does this kind of robots offer us as teaching material? On the one hand, the interest and motivation that they cause on children is much greater than the one caused by any other type of robotic material. On the other hand, the design and programming of a humanoid's behavior implies a careful consideration about the human being itself.

Following our aim of introducing robotics in the educational system, we can use this robot in other disciplines such as natural sciences, where it allows us to study the body of a human being and its various systems. It also offer us a new set of challenges related to calculations, trigonometry and physics as we will see in a further example.

At the same time, developing any intelligent behavior of the robot as a response to the captured data by the sensors implies studying aspects of communication, collaborative behavior and so on.

Researchers have begun to investigate, within special education, the use of robots, humanoids in particular, for the treatment of autism. The most important job in this field is the one that has been developed since 1998 by the Aurora Project of the Adaptive Systems Research Group at the University of Hertfordshire. [2] [3]

Various studies allow us to affirm that its therapeutic use in imitative interactive games help the development of social skills in autistic children. In the last few years inexpensive and highly versatile humanoids have emerged in the market which allows its mass usage at school and/or health institutions. Unfortunately, the programming of these devices requires complex languages, which impedes a simple use on behalf of professors or the kids themselves.

This is why we decided to expand our graphic platform of robots programming, Physical Etoys, to be able to control the most popular low-price humanoid in the market: Bioloid robot from the Korean firm named Robotis.

2 Physical Etoys

Physical Etoys is an extension of Etoys, a media-rich authoring environment and visual programming system made by the very same people who created Smalltalk, and it inherits all its educational potential [4]. The purpose of Physical Etoys is to allow kids to program robotic kits in an environment specially designed for them, following Papert's constructionism ideas [5]. Within this environment, physical objects are represented graphically and the students can directly interact with those entities, instantly seeing the consequences in the real world.

Physical Etoys is free, open source, it works in several operating systems, and it is translated to many languages such as Spanish, English, French, and Portuguese and others. Comparing Physical Etoys to other popular programming platforms for kids, such as Scratch, falls beyond the scope of this paper. However, we chose Physical Etoys because of its distinct characteristics: the entire Smalltalk programming environment is available in case the user needs to add more complex behaviours to his projects; and it provides different ways of expressing solutions to its problems through the use of multiple programming models (for instance, you can express your problem using a spreadsheet or a state-finite machine).

Among the robotic kits supported by Physical Etoys we find two big referents widely used by the education community: the Lego Mindstorms NXT and the Arduino board [6]. Since version 2.0, Physical Etoys also supports an argentinian robotic kit called DuinoBot, which is used in several schools from Argentina. Moreover, Physical Etoys includes a module system that allows the user to extend it in order to support many different electronic devices such as Microsoft Kinect, Nintendo Wiimote, Orbotix Sphero, among others.

Physical Etoys exposes a graphical user interface in which the real objects used in each kit (including motors, sensors, controllers, and wires) are represented by virtual objects. The user can create scripts by dragging and assembling tiles. These scripts run virtually at the same time, implicitly exposing the student to concurrent programming in an easy way.

3 Graphical User Interface of the tool

The Bioloid module of Physical Etoys has two objects: the motor and the CM-510 controller. They represent graphically some of the real objects that the Bioloid kit has in order to reduce the complexity of the interaction with it.

The properties and behavior of these objects can be programmed with these tiles:

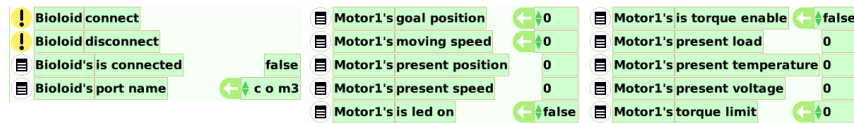


Fig. 1. Important tiles to program the Bioloid controller and Motor objects.

For example we can create a script in which the Bioloid Humanoid waves. We can take advantage of changing the script's ticking rate to one time per second in order to wait for the motor to reach its final position before the next movement:

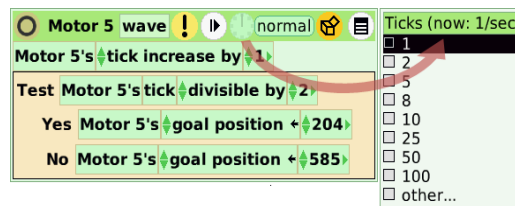


Fig. 3. Scripts to make the robot wave.

The objective of this virtual representation is to let the user focus specifically in the problem to solve instead of dealing with the complexity of the abstract thinking. If the user demands new challenges to program that cannot be done with tiles, he can switch to Smalltalk within the same environment.

4 Description of the Firmware

The original firmware included in the Bioloid kit has a few problems that we needed to solve before being able to use the kit with Physical Etoys. Among these problems, the Bioloid's firmware does not allow to easily access the data from the sensors and it provides little to no control over the possible errors produced in any part of the system (being the CM-510 controller, the servos, the communication between the controller and the servos, or between the controller and the computer).

Moreover, the original firmware includes a request-response communication protocol, which is not very suitable for Physical Etoys typical usage. Since in Physical Etoys all slots are constantly updating its value, this constant update would mean a lot of useless requests sent to the robot's controller.

For these reasons, the original firmware provided by the Bioloid kit was replaced by a new firmware that solves the abovementioned issues. This new firmware is based on the communication protocol used in Physical Etoys to communicate to the Arduino board, which is in turn inspired by Firmata [7].

This protocol message format is very simple and quick to parse, which makes it easy to implement and efficient for most use cases. In this protocol, each message contains a byte denoting its type followed by a series of arguments. Each type of message expects a fixed amount of arguments, so all messages have a fixed size known beforehand. Thus, when receiving a message, reading only its first byte is enough to know how many arguments to expect next. The “type” bytes and the “argument” bytes are marked distinctly using their first bit (being 1 for “arguments”, and 0 for “type”). This allows them to be easily identified and helps preventing communication errors. However, it means that the maximum number of messages allowed is 127 and all arguments sent must be split in packets of 7-bits. In practice, this disadvantage is insignificant.

To control the Bioloid servos from the computer only two type of messages are necessary: RQ_SERVO_WRITE_BYTE and RQ_SERVO_WRITE_WORD.

With this two messages any attribute from the servo can be modified, being the most interesting: Goal position, Led, Moving speed, Torque enabled, and Torque limit.

From the other side of the communication, the robot controller sends at regular intervals the data from each servo. This data is limited to only a few of the servo’s attributes in order to prevent overloading the communication channel with useless information. The attributes we most care about are: Present load, Present position, Present speed, Present temperature, and Present voltage.

Additionally, in order to avoid receiving data from servos we do not need to monitor another type of message was implemented, called:

RQ_SERVO_ACTIVATE_SAMPLING.

5 Inverse kinematics example

5.1 Description

As an example of this educational tool we have developed an application that takes advantage of the humanoid structure provided by the Bioloid kit and the skeleton information given by Microsoft Kinect.

This example can be used to teach the following concepts: basic trigonometry relations, coordinate systems, and linear functions.

By taking advantage of the information provided by Kinect, the application performs the necessary calculations in order to translate the positions of each skeleton’s node to appropriate angles for each Bioloid’s motor, so that the robot can mimic the human position in ACAACA near real-time.

We will be using the type A humanoid structure provided by the Bioloid Premium Kit. In this structure the controller is attached to the back of the robot and it does not interfere with its movements. Furthermore, the connection from the controller to the computer is done wirelessly using the zigbee module (provided by the Bioloid kit as well).

The modular design of Physical Etoys allows installing the Kinect and the Bioloid controllers separately. The sources of this modules as well as this entire example can be found in Physical Etoys' website¹.

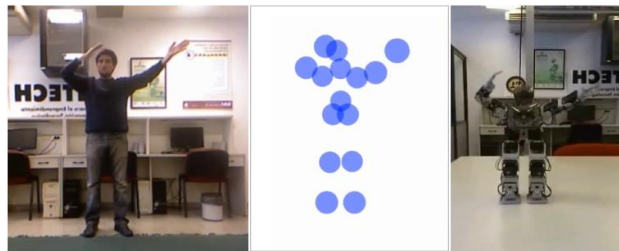


Fig. 3. From left to right: the position of the human, the skeleton produced by Kinect, the final

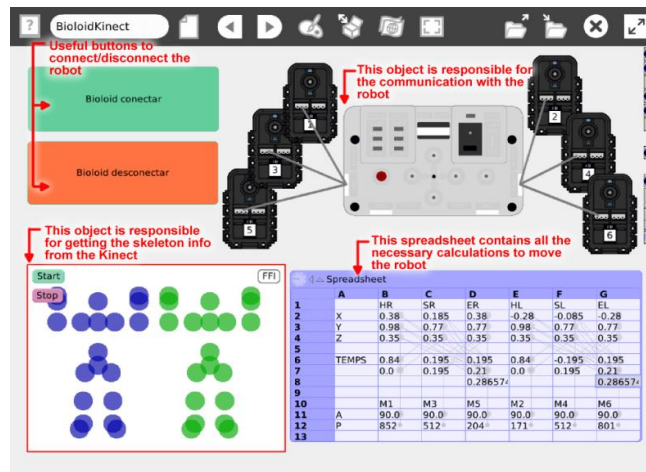


Fig. 4. The application and its parts.

The problem presented by this example can be classified as a dataflow programming problem, which in turn can be naturally expressed with a spreadsheet. Although Physical Etoys' programming model is essentially imperative, it includes a Spreadsheet object developed by Takashi Yamamiya [8], which is especially useful to express programs in a functional style.

5.2 Implementation

The problem of moving a robotic arm to a specified location is known as "Inverse kinematics". In our case, we do not need to provide a high degree of precision; instead

¹ <http://tecnodacta.com.ar/gira/projects/physical-etoys/>

we only need the movement to look as closely as we can to the movement of the human.

So, for this example, we have chosen a trigonometric solution that relies on the law of cosines because it involves high-school math and its implementation is relatively simple. [9].

For the moment, we have only implemented the movement of both arms. Controlling the legs and the body requires solving an additional set of problems, which falls beyond the scope of this work.

Each arm has 3 degrees of freedom. We will briefly describe the necessary calculations to solve the movement of the right arm, knowing that the left arm can be solved in a similar way. By applying the law of cosines to each triangle formed by the different joints, we can calculate the appropriate angle for each motor (as shown in fig. 9).

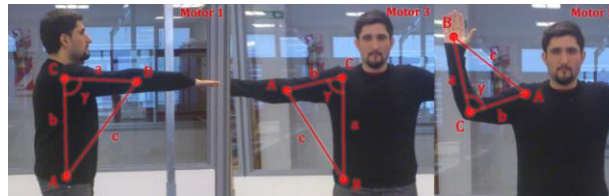


Fig. 5. The angles we need to solve for each motors.

Once we have the angle for each joint we need to calculate the final value (between 0 and 1023) that we will deliver to the servo. This can be solved with a simple linear function for each motor, calculated empirically.

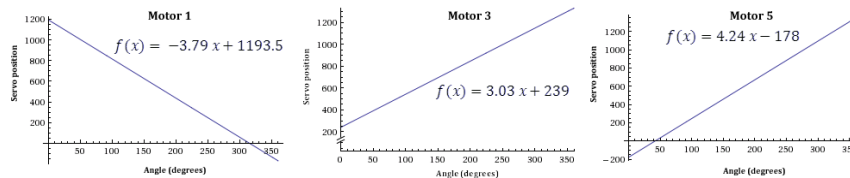


Fig. 6. Linear functions to translate the calculated angle to positions for each servo.

All these formulas were coded inside a Physical Etoys' spreadsheet, as shown in figure 7. The input cells are linked to the "x", "y", "z" slots from the corresponding joints in the Kinect object, so that their value will be updated automatically whenever the Kinect camera sends new information. The output cells contain the formulas described above and are linked to the "goal position" slot from the corresponding Bioloid motors, so that the motor's position would get updated automatically with every tick.

	A	B	C	D	E	F	G
Input cells		HR	SR	ER	HL	SL	EL
2	X	0.38	0.185	0.38	-0.28	-0.085	-0.28
3	Y	0.98	0.77	0.77	0.98	0.77	0.77
4	Z	0.35	0.35	0.35	0.35	0.35	0.35
5							
Temporary values	TEMPS	0.84	0.195	0.195	0.84	-0.195	0.195
7		0.0	0.195	0.21	0.0	0.195	0.21
8				0.286574			0.286574
9							
Output cells		M1	M3	M5	M2	M4	M6
11	A	90.0	90.0	90.0	90.0	90.0	90.0
12	P	852	512	204	171	512	801
13							

Fig. 7. The spreadsheet used to calculate the angles.

6 Future work

The future priority tasks of this project can be categorized in two different ways: technical and pedagogical.

Regarding the technical tasks, even though we have a functional version of the project, only the motors and the controller were virtually modeled. We have to add the sensors that belong to the kit in our software and we need to implement an appropriate way to report errors that may occur during the use of the robot, especially to increase its durability in the classroom.

In the particular example abovementioned, the robot was moving its arms. The tool supports many motors but only six are used. We have to consider that there are motor positions for the Bioloid humanoid that are physically invalid; therefore, we have to develop a basic sense of proprioception for it because the robot can be damaged while someone is interacting with it. Finally a study will be made of the new possibilities that Kinect 2.0 will bring to see if it could improve the experience with Physical Etoys and Bioloid Kit within the educational environment.

Regarding pedagogical tasks, we will invite teachers to think other practical exercises using this environment not only for students but also for their colleagues and their courses.

7 Conclusion

Although we have laid the foundation for the development of an environment that promulgates a new way of interaction with the Bioloid Kit in the classroom, it should be noted that there is still a significant way to go in terms of future work. It is also important to note that nowadays, robotics offers transversal content in a more accessible way to supplement the subjects of the educational programs. We believe that it is important to inspire children with a useful and attractive activity at an early age to prepare them for a world that tends to be robotic.

8 References

1. A. Eguchi, «How to turn competitions into International collaboration through educational robotics—a case of RoboCupJunior Dance tournament», in Society for Information Technology & Teacher Education International Conference, 2010, vol. 2010, pp. 4002–4005.

2. K. Dautenhahn y A. Billard, «Games children with autism can play with Robota, a humanoid robotic doll», *Universal access and assistive technology*, pp. 179–190, 2002.
3. Robins, K. Dautenhahn, R. T. Boekhorst, y A. Billard, «Robotic assistants in therapy and education of children with autism: can a small humanoid robot help encourage social interaction skills?», *Univ Access Inf Soc*, vol. 4, n.º 2, pp. 105-120, dic. 2005.
4. A. Kay, «Children Learning by Doing: Squeak Etoys on the OLPC XO», *Viewpoints Research Institute, Research Note RN-2007-006-a*, 2007.
5. S. Papert, *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc., 1980.
6. G. Zabala, R. Morán, y S. Blanco, «Arduino Etoys: a programming platform for Arduino on Physical Etoys», in *1st International Conference on Robotics in Education (RIE 2010)*, Bratislava, Eslovaquia, 2010.
7. H. Steiner "Firmata: Towards making microcontrollers act like extensions of the computer", In *Proc. NIME '09*, pp. 52--53.
8. Takashi Yamamiya, "Skeleton — Easy Simulation System," c5, pp.50-53, *Second International Conference on Creating, Connecting and Collaborating through Computing (C5'04)*, 2004.
9. R. Müller-Cajar, R. Mukundan, 'Triangulation: A New Algorithm for Inverse Kinematics', *Proceedings of Image and Vision Computing New Zealand 2007*, pp. 181–186, Hamilton, New Zealand, December 2007.