

Robótica en el Contexto de las Tecnologías de la Educación

Gustavo Pereira¹, Jorge Ierache¹, Norberto Mazza¹, Juan Iribarren¹, Rodolfo Meda¹, Diego Avila¹, Emiliano Lorusso¹, Sofia Fasce¹

Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica (ISIER)¹
Laboratorio de sistemas de información avanzados Facultad de Ingeniería Universidad de Buenos Aires²

ISIER, Facultad de Informática Ciencias de la Comunicación y Técnicas Especiales
Universidad de Morón, Cabildo 134, (B1708JPD) Morón, Buenos Aires, Argentina
54 11 5627 200 int 189
jierache@yahoo.com.ar

Resumen

Mediante la implementación y ampliación de kits de robótica o dispositivos que son considerados juguetes se pueden realizar experiencias en el contexto de las Tecnologías de la Educación con alumnos de niveles iniciales e inclusive con alumnos de niveles más avanzados.

Los alumnos iniciales pueden inclusive incorporarse al mundo de la programación al observar de una forma mucho más clara y atrayente el resultado de su código. No es lo mismo para un alumno inicial aprender una estructura de programación observando en pantalla que ver como un robot reacciona en un ambiente. En el caso de los estudiantes avanzados los juguetes autónomos y kits de robots son excelentes herramientas para probar nuevos paradigmas o metodologías aplicadas a la resolución de problemas simples como por ejemplo el juego de “piedra libre” [1] o sistemas cooperativos más complejos como fútbol de robots.

Palabras clave: robótica, educación.

1. Introducción

Los diferentes kits de material de robots disponibles nos brindan actualmente diversas oportunidades de aplicación de estas tecnologías en la formación de alumnos, incentivando a los alumnos tanto iniciales como avanzados en informática mediante la

construcción de un robot físico y la programación de su comportamiento en ambientes reactivos, cooperativos y competitivos.

En trabajos anteriores Ierache et al [2], [3],[4] se trató la aplicación de un Laboratorio de Sistemas Autónomos de Robots (LSAR) el que colabora con distintas áreas en los procesos de formación de alumnos en el marco del contexto propuesto en figura 1, se observa el Paradigma bajo el cual el alumno desarrolla su trabajo, la Metodología que se aplica bajo el paradigma seleccionado, Técnicas que facilitan el desarrollo de fases y etapas de la metodología aplicada, Herramientas sobre las cuales son aplicadas las técnicas, los Lenguajes de programación en los cuales se pueden aplicar las diversas primitivas y procedimientos, y Robots que ejecutaran todas estos procedimientos.

Para los estudiantes iniciales el paradigma principal, es el de objetos, con sus metodologías, técnicas como diagramas de Nassi Seiderman [5], UML [6], con lenguajes de programación semejantes a C en particular NQC [7] para los Lego RCX o NXC [8] para los Lego NXT [9], lenguajes derivados de JAVA, como LeJOS [10],[11] o librerías .NET, para el desarrollo del software que correrá sobre robots NXT.

Para estudiantes avanzados se considera el paradigma multiagente, aplicando

metodologías como MaSE [12], y técnicas tales como Agent-UML.

No es lo mismo que un alumno observe los resultados de su código de programación en una pantalla que verlo reflejado en el comportamiento de un Sistema Autónomo Robot (SAR) que puede moverse por una habitación esquivando obstáculos, juegue al fútbol, realice rescate o inclusive ver cómo responde a nuestros movimientos.

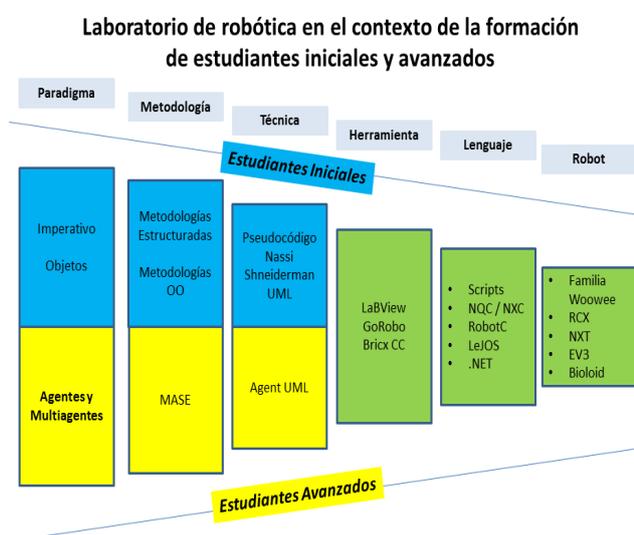


Figura 1 Contexto de la formación de estudiantes

El LSAR contribuye con el mejoramiento de procesos constructivos en un ambiente de trabajo en grupos de alumnos los cuales son motivados por el desarrollo de sus robots.

Muchos proyectos involucran un control centralizado clásico, la computadora le dice al motor uno que se encienda, gire en sentido de las agujas del reloj, a media potencia bajo una secuencia planificada de acciones, pero también se puede aplicar el mismo agente robot para explorar sistemas descentralizados y de conductas auto-organizadas [13], [14], [15].

Si consideramos un robot que deambula por un

ambiente en el que cuenta con áreas iluminadas y áreas oscuras, nuestro agente cuenta con dos reglas, una que le indica al agente moverse hacia adelante cuando detecta áreas iluminadas y moverse hacia atrás cuando detecta áreas oscuras, el agente deambula hasta que llega a una sombra, entonces retrocede hasta que sale de ella y luego hacia adelante otra vez, sigue así oscilando en el borde de la sombra; en este caso podemos considerar a nuestro agente robot como una criatura que detecta bordes, en sí esta capacidad no está explícitamente declarada en sus dos reglas, en realidad es una conducta grupal que surge de la interacción de las dos reglas, algo similar a como la conducta de una bandada surge de la interacción entre los pájaros [16].

2. Robots de Lego

El objetivo de esta sección es brindar una orientación general de los Kits de Robótica económicos actuales de Lego el NXT sus herramientas de programación en el ambiente de LabView [17], RobotC [18], NQC, NXC, librerías .NET [19],[20],[21] , LeJOS, entre otras.

2.1 Lego mindstorms NXT

El NXT (Figura 2) es la segunda generación de robots de Lego mindstorms, se caracteriza por una mayor capacidad de cómputo que el RCX. El NXT incluye funciones para testear los sensores, la personalización de los sonidos que puede reproducir, tiene tres puertos para motores, cuatro puertos para sensores.



Figura 4 Kit Lego NXT

Se equipa con un sensor de luz, un sensor de sonido para detectar sonidos, dos sensores de tacto, que pueden detectar tres estados (presionado, libre, rebote), un sensor de ultrasonido que actúa como un radar permitiendo la detección de objetos, al que se lo puede configurar para detectar objetos lejanos o cercanos, detecta objetos entre 0 y 255 cm con una precisión de ± 3 cm, además cuenta con un parlante de alta fidelidad, mejorado y tres servomotores, mejorados con respecto a la versión RCX.

Los servomotores tienen integrados sensores de rotación que permiten movimientos precisos, controlados y una perfecta sincronización de los motores. El NXT posee un puerto USB, para la descarga de programas. Soporta comunicación bluetooth inalámbrica, permitiendo descargar programas de este modo, así como también la interacción con celulares, con PCs y laptops, etc.

La comunicación con otros NXT también es vía bluetooth. Estos robots puede ser

programables en entornos gráficos nativos, para RCX y LabView para NXT.

En cuanto a LabView cabe destacar que fue desarrollado por National Instruments y utilizado por la NASA para monitorear y controlar al robot Sojourner Rover, en la misión de exploración de la superficie de Marte [22].

Estos entornos utilizan bloques que se ensamblan para formar un programa completo. Estos bloques incluyen control de motores (avance, retroceso, encendido y apagado), ciclos repetitivos (while, repeat), estructuras de control (if else), adquisición de datos de los sensores, utilización de variables, constantes y timers.

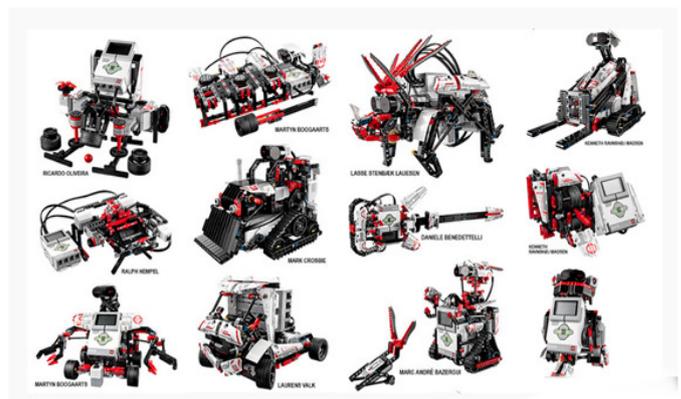


Figura 5 Kit Lego EV3 piezas y configuraciones

2.3 Lego mindstorms EV3

El kit mindstorms EV3 [23] es la tercera generación de kit de robótica LEGO, a diferencia de su antecesor el NXT que utiliza

un procesador ARM7, la potencia de procesamiento del EV3 está dada por un ARM9 ejecutando Linux con 64 MB de RAM y 16 MB de memoria Flash junto con la posibilidad de expandir el almacenamiento hasta 32 GB mediante un slot microSDHC.

Los sensores y motores del NXT pueden ser conectados en un EV3 permitiendo la reutilización de piezas adquiridas anteriormente.

2.3 Lenguajes y herramientas

Además de los entornos gráficos, existen una serie de programas que permiten su programación en códigos más tradicionales, como por ejemplo en Java. Es el caso del API LeJOS para RCX y para NXT el LeJOS NXJ, iCommand un paquete java para el control de comunicaciones sobre conexiones Bluetooth [24].

Uno de los programas más utilizados y que aumenta enormemente las posibilidades de programación es el NQC de Dave Baum, utilizado para programar el RCX y el NXT en un lenguaje similar al C. Para el NXT, existe un programa llamado RobotC, que es mucho más completo que el NQC del RCX, y que además incluye un firmware propio, que lo hace muy potente.

Para la programación en JAVA de la familia mindstorms se encuentra el LeJOS. Para el cual es necesario instalar un firmware alternativo en el ladrillo, se caracteriza por:

[a] permitir realizar programas en JAVA para controlar robots.

[b]funcionar en entorno Windows y Linux.

[c] comunicarse con el NXT y EV3 vía USB.

[d] soportar comunicación con Bluetooth, los firmwares del NXT y EV3 permiten una configuración de tipo Amo-Esclavo para la comunicación con bluetooth. Mediante bluetooth se pueden comunicar hasta tres

ladrillos.

Se han desarrollados diversas librerías en lenguaje .NET que permiten la manipulación de un ladrillo NXT mediante la conexión bluetooth, la mayor ventaja de estas librerías es que si la potencia de procesamiento del ladrillo no es suficiente para la aplicación deseada, con esto se permite utilizar el procesamiento de una PC.

Entre las librerías de mayor utilización se encuentran: Mindsqualls [19], NXT.NET [20]. y Aforge.NET [21].

Como herramienta soporte para el desarrollo y testing de robots NXT, existe la aplicación RoboSim [25] (Figura 7) la cual presenta un mundo virtual similar al que se podría implementar en un laboratorio. Este tipo de aplicaciones permite a los alumnos probar sus desarrollos sin necesidad de estar presentes en el laboratorio, o de tener acceso al agente físico.

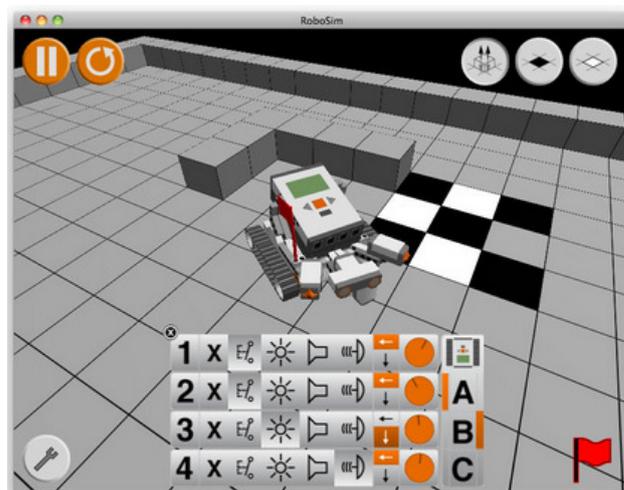


Figura 6 Robosim

3 Juguetes Autónomos

En materia de Juguetes autónomos se detallarán las características más relevantes de “Robosapien”. (robot bípedo) y de “Robopet, Robotail, Roboraptor” (robots cuadrúpedos), Roboquead (robots exopodos) de la familia Woowee [26] . Se consideran además las interfaces de comunicación y herramientas de

programación, en particular GoRobo. Aunque se venden como juguetes, ofrecen unas prestaciones tan avanzadas que hacen de éstos un medio excelente para experimentar con la robótica

3.1 Bípedos: estos robots, poseen sensores de sonido estéreo, visión infrarroja, y también sensores de tacto para detectar obstáculos y un importante número de grados de libertad. Dentro de esta categoría encontramos: [a] Robosapien V1 es una versión con menores prestaciones a nivel de sensores que el V2, no incorpora visión, las capacidades de desplazamiento son similares en sus funcionalidades, si bien el RSV1 al ser más pequeño tiene un mejor desplazamiento, [b] Robosapien V2, además de las características antes mencionadas incluye también sensores táctiles en los guantes, y en las palmas de las manos, (Figura 7) lo que le permite tomar objetos con ellas. Además posee una cámara que le permite reconocer colores [27]. [c] Robosapien Multimedia, [28], aumenta aún más las capacidades de RSV2, incluyendo como característica más importante una memoria mini SD, en donde se lo puede programar directamente, a través de un editor de código de tipo gráfico, que es propio de esta versión. Por default tiene cuatro personalidades que el usuario puede modificar. Además puede grabar videos, sonidos en formato mp3, tomar fotos, y luego reproducirlos en su display LCD.

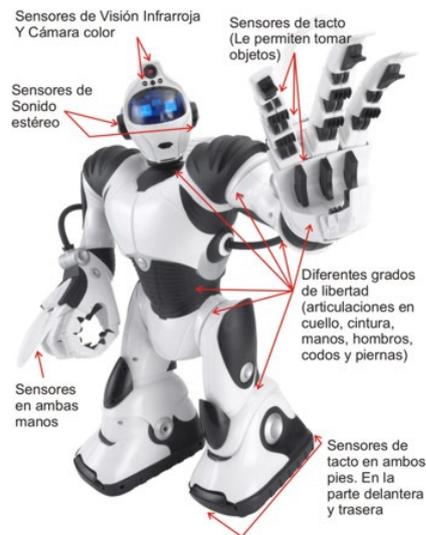


Figura 7 Robosapien V2

3.2 Cuadrúpedos. Estos robots, también están equipados con visión infrarroja, sensores de sonido estéreo y motores. Entre éstos se destacan: [a] Robopet, (Figura 8.a) además de lo dicho anteriormente es capaz de interactuar con Robosapien, y también puede detectar los bordes de por ejemplo una mesa [29]. [b] Robotail, (Figura 8.b.) posee un sensor de tacto en su lomo, que cuando es presionado, hace que tenga comportamientos diferentes. Además cuando está “hambriento”, se vuelve muy “agresivo”, y sólo se calma “al encontrar comida” [30]. [c] Roboraptor, (Figura 9.a.) es el roboreptil [31] que también es capaz de interactuar con Robosapien.

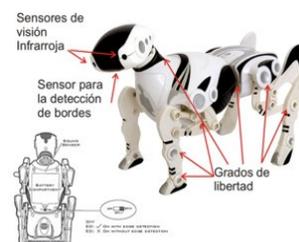


Figura 8.a Robopet



Figura 8.b Robotail



Figura 9.a Roboraptor

Figura 9.b
Roboquad

3.3 Antrópodos: En este grupo presentamos a Roboquad [32] (Figura 9b), tiene cuatro patas con un chasis que está diseñado para moverse en cualquier dirección con 3 velocidades diferentes se caracteriza por identificar movimientos a una distancia de alrededor de tres metros, una vez identificado puede seguir el movimiento del objeto. Tiene sensores de bordes capaces de detectar marcos de puertas, bordes de mesas y sillas.

3.4 Lenguajes y herramientas de programación familia WowWee

El entorno de programación “GoRobo” [33], permite controlar a la mayoría de los robots mencionados de la familia WowWee [34]. (Roboraptor, Robopet, Roboreptile, RSV2 y RS Multimedia). El lenguaje de programación que utiliza se denomina GRIDscript (Go-Robo ID script). Utiliza una sintaxis de programación simple y consistente basado en las prácticas modernas de productos comerciales de alto nivel (Visual Basic, Java, etc). GRIDscript utiliza sintaxis básica de programación (While/EndWhile, For, If/Else/Endif, Repeat/EndRepeat), para la creación de procedimientos y el uso de variables. El estudiante inicial puede utilizar este lenguaje para definir procedimientos simples que posteriormente pueden combinarse para crear otros más complejos. Además los robots pueden programarse para interactuar entre sí, ya que el software permite el control de seis de ellos de forma simultánea. Las órdenes se transmiten a través de una torre infrarroja que a modo de intérprete envía los comandos a cada robot, distinguido por su tipo a través de un torre infrarroja, como USB-

UIRT [35]. y RedRat3 [36]. Algunas de las cosas que se pueden hacer con GoRobo son: utilizar bloques condicionales y de repetición de instrucciones, utilizar eventos condicionados por timers, se puede introducir ejecución aleatoria de código. Este lenguaje, se ha diseñado para adaptarse a cualquier edad y utilizarse tanto en un contexto educativo como en uno más profesional, donde se genera un ambiente de interacción de los lenguajes clásicos y formales de programación con los comandos nativos de los robots utilizados. Incorpora además un editor de escenas al que se le puede sumar sonido. Otras opciones de Programación, para el Robosapien como Robopet, reciben los comandos vía IR por control remoto, de este modo hay quienes han realizado un mapeo de dichos comandos a hexadecimal [37], esto hace posible la programación de Robosapien descargando el código con la torre de IR de Lego Mindstorms. El problema es que hay una limitante con la cantidad de instrucciones que el RS puede recibir, que son veinte como máximo. Otra opción más radical es el transplante de cerebro al Robosapien, hay quienes han optado por reemplazar la cabeza del Robosapien V1 por una Palm [38], de este modo, se elimina el problema de la cantidad de instrucciones que se pueden enviar al RS, ampliando enormemente la capacidad de cómputo para Robosapien V1.



Figura 10 Interface Go-Robo

4. Robots Multimodulares

El Kit de Robot Bioloid [39], Figura 11, se caracteriza por disponer de un total de 18 servomotores, sensores de infrarrojos en la cabeza para comunicarse con otros robots y sensores de proximidad hacia delante y hacia los lados, micrófono y altavoz. Bioloid Comprehensive es el kit de la plataforma robótica modular adecuado para construir robots avanzados de hasta 18 grados de libertad como los humanoides. Es adecuado para aprendizaje, hobby, investigación y competición. El kit es como una versión superior de Meccano y está hecho de muchos bloques constructivos que el usuario puede unir y encajar con tornillos. Su lenguaje de programación es C.



Figura 11 Robot Bioloid

5. Robotgroup

El proyecto Argentino Robotgroup presenta diversos kits destinados a la enseñanza de la robótica, especialmente el kit avanzado N8 [40] el cual posee un controlador Duinobot, compatible con Arduino y que puede ser programado tanto por IDE Arduino como por el entorno gráfico miniBloq propio de robotgroup. El kit viene con sensores diversos como luz, táctil, infrarrojo, ultrasónico y sensor de IR un receptor de IR. El kit también incluye dos servomotores y varias piezas para el armado de múltiples diseños de robots.

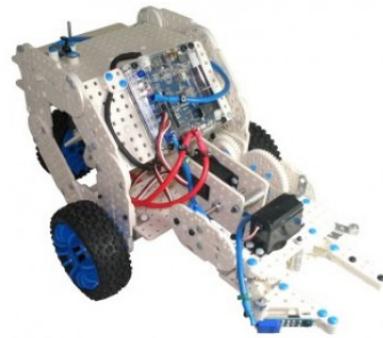


Figura 12 Kit RobotGroup N8

6. Fútbol Robot

El fútbol robot plantea un problema de cooperación entre agentes robóticos individuales que deben interactuar entre sí para obtener un fin en común, los robots comparten un ambiente y deben evitar que los oponentes consigan hacer goles. Para esto es necesario la programación de navegación, estrategias de juego, interacción, manejo de ambientes y variables físicas.

El primer contacto con el fútbol robot es mediante el simulador de FIRA (SimuroSot), en donde cada equipo se programa en lenguajes LINGO o C++. Cada equipo se compila en una DLL que es cargada en el simulador y de acuerdo a las situaciones del partido reacciona como fue programada.



Figura 13 SimuroSot

Para la implementación de equipos de fútbol robot físicos (F-180 Robocup) se pueden adquirir kits diseñados exclusivamente para ello o utilizar un kit multipropósito como el NXT o el EV3. En las experiencias realizadas siempre se utilizó el kit Lego NXT con las librerías Aforge.NET para el desarrollo de equipos físicos.



Figura 14 Partido de Fútbol Robot Físico

8. Bibliografía

- [1] B. Bauer, J. P. Muller, J. Odell. "Agent UML: A Formalism for Specifying Multiagent Software Systems". Proc. ICSE 2000 Workshop on Agent-Oriented Software Engineering AOSE 2000, Limerick, 2000, Agent-UML
- [2] Laboratorio de Robots Autónomos en el Contexto de las Carreras de Informática Experiencias para su Implementación, Ierache J. Bruno M., Mazza N. 2007.
- [3] Experiencia de laboratorio en Robótica: De las Tortugas Simuladas a los Robots y Juguetes Autónomos una Oportunidad Para la Enseñanza de Programación, Ierache J. Bruno M., Mazza N. 2007.
- [4] Ierache, J., García-Martínez, R., De Giusti, A. (2009)., "A Proposal of Autonomous Robotic Systems Educative Environment", R. J.-H. Kim et al. (Eds.): FIRA 2009, CCIS 44, pp. 224–231, 2009. © Springer-Verlag Berlin Heidelberg 2009.
- [5] Nassi, I.; Shneiderman, B.: Flowchart

7. Conclusiones y futuras líneas de investigación

El presente trabajo presenta las características más distintivas de kits de robots y juguetes autónomos de bajo costo que son empleados en la enseñanza de la robótica, acompañados de sus lenguajes de programación que facilitan el desarrollo de experiencia en el área y contribuyen a la enseñanza de programación de alumnos principiantes, como así también facilitan por parte de alumnos avanzados el desarrollo de agentes autónomos, orientando al desarrollo en este caso a una amplia variedad de modelos computacionales descentralizados, como por ejemplo, las redes neuronales, arquitecturas de subsunción y autómatas celulares, entre otros. En el marco de las líneas de trabajo futuras se orientan al empleo de sistemas de visión globales y locales para el mejoramiento de aplicaciones en el contexto de robótica situada.

techniques for structured programming, SIGPLAN Notices XII, August 1973

[6] UML <http://www.uml.org/>

[7] NQC – Not Quite C.:

<<http://bricxcc.sourceforge.net/nqc/index.html>>

[8] NXC <http://bricxcc.sourceforge.net/nbc/>

[9] NXT <http://mindstorms.lego.com/>

[10] Lejos NXJ

http://lejos.sourceforge.net/p_technologies/nxt/nxj/nxj.php

[11] LeJOS RCX

http://lejos.sourceforge.net/p_technologies/rcx/lejos.php

[12] Elaboración de una Aproximación Metodológica para el desarrollo de Software Orientado a Sistemas Multiagente. Ierache J, 2003.

[13] CASTELO, C; FARSI, H; SCARPETTINI, F. Tesis de Licenciatura "Fútbol de Robots: Revisión del Estado del Arte y Desarrollo del Equipo UBASot de Simulación". Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires. 2002

[14] CLARO, M.; GAZOLLI, A;

- POTENZA, A; VISCUSO, Germán;
IERACHE, Jorge. “*El equipo Moraso*”.
Facultad de Informática, Universidad de
Morón. 2003
- [15] RESNICK, Mitchel. Tortugas, Termitas y
Atascos de Tráfico, Gedisa, 2001
- [16] MORROLLON, M; SEGOVIANO, A. *1,*
2, 3... Logo (Ideas e Imaginación). Centro de
- [17] National Instruments. *LabVIEW*. [en
línea]. [ref. Abril de 2007]. Disponible en
Web:
<<http://www.ni.com/academic/mindstorms/>>
- [17] ROGER, Chris. LEGOS, ROBOLAB,
and LabVIEW: Designing, Programming, and
Collecting Data. University of Wisconsin-
- [22]NI :
<http://www.ni.com/academic/mindstorms/>
- [23] Lego mindstorm EV3
<http://www.lego.com/en-us/mindstorms/products/ev3/31313-mindstorms-ev3/>
- [24] Icommand NXT
http://lejos.sourceforge.net/p_technologies/nxt/icommand/icommand.php
- [25] Robosim:
<http://schuelerlabor.informatik.rwth-aachen.de/simulator>
- [26] <http://www.woowe.com>
- [27] ROBOSAPIEN.tk, the first unofficial
robosapien hacks and mods site.
<<http://home.planet.nl/~pruim006/main.htm>>
- [28]WowWee
<<http://www.rsmediaonline.com/>>
- [29] Robopet: www.robopetonline.com
- [30] Robotail: www.roboreptileonline.com
- [31] Roboraptor : www.roboraptoronline.com
- [32] WowWee Roboquad
< <http://www.roboquadonline.com/>>
- Madison. The Institute on Learning
Technology. National Institute of Science
Education. EEUU.
<http://www.wcer.wisc.edu/archive/cl1/ilt/extra/download/solution/rogerscw97.doc>>
- [18] QUICK START GUIDE, Robotics
Academy, CarnegieMellonUniversity,
<http://www.robotc.net/>>
- [19]MindSqualls
<<http://www.mindsqualls.net/>>
- [20] NXT.NET <<http://nxtnet.codeplex.com/>>
- [21] AForge.NET <
http://www.aforge.net.com/framework/features/lego_robotics.html>
- [33]GOROBO:
<http://www.q4technologies.com/>
- [34] <http://www.woowe.com>
- [35] USB-UIRT: <<http://www.usbuirt.com/>>
- [36] RedRat3 – USB Universal Remote
Control for
<<http://www.redrat.co.uk/RedRat3/index.html>
>
- [37]Controlling RoboSapien using LEGO IR-
Tower.Trondheim-Bratislava,
<<http://www.robotika.sk/maine.php>>
- [38] Sven., Behnke et al., Using Handheld
Computers to Control Humanoid Robots
Proceedings Dextrous Autonomous Robots
and Humanoids, 2005.
- [39] Tribotix Bioloid Constructive Kid
<http://www.tribotix.com/Products/Robotis/Bioloid/Bioloid_info1.htm>
- [40] RobotGroup N8
<http://robotgroup.com.ar/index.php/productos/217-kit-n8#n8-kit-avanzado-de-tecnología-y-robótica>