# Coordinación de Arbotix, Raspberry Pi y motores Dynamixel Ax-12+ con el objetivo de la construcción de un robot humanoide que busque y patee pelotas

Jennifer Dos Reis
Grupo De Inteligencia Artificial
Universidad Simón Bolívar
Caracas, Venezuela
08-10323@usb.ve

Juliana León
Grupo De Inteligencia Artificial
Universidad Simón Bolívar
Caracas, Venezuela
juliana@ac.labf.usb.ve

Carolina Chang
Grupo De Inteligencia Artificial
Departamento de Computación y Tecnología de la Información
Universidad Simón Bolívar
Caracas, Venezuela
cchang@usb.ve

***Abstract*—This article presents Debupa (Detección Búsqueda Pateo) a small humanoid (38 cm high) built with the Bioloid Premium Kit [?] from the manufacturer ROBOTIS [?]. The CM-510 card has been excluded to be replaced by the Arbotix controller card, which is used to control the 16 motors Dynamixel AX-12 + (for moving the robot) and 2 analog servo (to move the camera). It has been also added a Raspberry Pi mini computer, with its camera [?], so that the robot can detect and track the ball autonomously.**

**All these components must be coordinated to accomplish the tasks of detection, tracking and kicking of the ball. Thus, communication between the Arbotix and Raspberry Pi is necessary. The tool used for this is the ROS (Ros Operating System) framework [?].**

**In the Raspberry Pi the c++ language is used. A program captures the camera image, filters and processes it to find the ball, decides the action to take, and requests the Arbotix to execute the movements of the motors. To capture the camera image the raspicam_cv library [?] was used. To filter and process the image the OpenCv libraries have been used [?].**

**The Arbotix, besides from moving engines, is responsible for monitoring the robot balance, for it uses the Robotis Gyro sensor [?] If it detects an imbalance of a certain magnitude, then the robot has fallen, and tries stand up.**

***Keywords*—Arbotix ,Raspberry Pi, Dynamixel, Humanoid, Soccer, OpenCV.**

***Resumen*—En este artículo se presenta a Debupa (Detección Búsqueda Pateo) un humanoide de tamaño pequeño (38 cm de alto) construido con las piezas del kit Bioloid [?] del fabricante ROBOTIS [?]. Del kit se ha excluido la tarjeta CM-510 para sustituirla por la tarjeta controladora Arbotix, que será la que controle los 16 motores Dynamixel Ax-12+ (para mover al robot) y 2 servomotores analógicos (para mover la cámara). Además se ha agregado un mini computador Raspberry Pi, con su cámara [?], para que el robot pueda detectar y seguir la pelota de forma autónoma.**

**Todos estos componentes deben ser coordinados para que se logre cumplir la tarea de detectar, seguir y patear la pelota. Por ello se hace necesaria la comunicación entre la Arbotix y la Raspberry Pi. La herramienta empleada para ello es el framework ROS (Ros Operating System) [?].**

**En la Raspberry Pi se usa el lenguaje c++ y se ejecuta un solo programa encargado de captar la imagen de la cámara, filtrar y procesar para encontrar la pelota, tomar la decisión de la acción a ejecutar y hacer la petición a la Arbotix para que de la orden a los motores de ejecutar el movimiento. Para captar la imagen de la cámara se ha utilizado la librería raspicam_cv [?]. Para filtrar y procesar la imagen se ha usado las librerías de OpenCv [?].**

**La Arbotix, además de controlar los motores, se encarga de monitorizar que el robot se encuentre balanceado, para ello usa el sensor Gyro de ROBOTIS [?]. Si detecta un desbalance de un cierto tamaño puede saber si se ha caído y levantarse.**

***Palabras Clave*—Arbotix, Raspberry Pi, Dynamixel, Robot Humanoide, Fútbol, OpenCV.**

## I. Introducción

RobotCup [**?**] es una competencia de fútbol iniciada desde 1997 donde contribuyen las áreas de robótica, investigación e inteligencia artificial. Entre sus categorías se encuentra RobotCup Soccer [**?**], la cual consiste en la participación de pequeños robots humanoides que se enfrentan a otro equipo para jugar fútbol. El objetivo de esta competencia es lograr que en el año 2050 el equipo campeón logre vencer al ganador del año en la copa mundial de la FIFA (International Federation of Association Football). Las destrezas de robots con forma de humanos (como son caminar, percibir el mundo y tomar alguna acción sobre él) suelen ser más complejas de lo que se puede pensar. Una de las más avanzadas muestras en el área es el robot ASIMO [**?**], creado por la compañía Honda, cuyos últimos avances incluyen la predicción de trayectoria de objetos para poder esquivarlos.

En este artículo se presenta un robot humanoide (Debupa) de tamaño pequeño (38 cm de altura) que es capaz de detectar una pelota de un color único en el ambiente, buscarla y, al llegar hasta ella, patearla. En artículos relacionados de este mismo enfoque se puede encontrar el trabajo de Sven Behnke

cuyo título es “See, walk, and kick: Humanoid robots start to play soccer” donde se describe la construcción del equipo de robots que participaron en la RobotCupSoccer en el año 2006. El artículo cubre el diseño mecánico y electrónico, además el software utilizado para la percepción, control de comportamiento, comunicación y simulación de los robots [**?**].

Existen equipos que han participado durante varios años consecutivos en la competencia Robocup, logrando mejoras en sus diseños y técnicas; tal es el caso del equipo MRL que ha participado en los años 2011, 2012, 2013 y 2014 en la categoría “Humanoid League”, han iniciado con el hardware del robot DARwIn-OP y con el tiempo han modificado los componentes electrónicos para agregar eficiencia y estabilidad. Para el balance han utilizado un sensor Gyro y otros sensores de aceleración. Para la visión han empleado una cámara conectada por USB al CPU principal [**?**].

En el desarrollo de habilidades más específicas con respecto a la competencia RobotCup Soccer, en el artículo de investigación de Seung-Joon Yi, Stephen McGill y Daniel D. Lee [**?**], se refieren a dos posibles estrategias para el pateo de la pelota donde los factores fundamentales para un buen desempeño es la fuerza y la rapidez con que se patea. Los investigadores ponen en práctica dos estrategias de pateo en distintas circunstancias del juego basado en la cinemática y dinámica de equilibrar el cuerpo al momento de realizar el pateo.

En la sección II se describen los componentes de hardware usados para construir el humanoide Debupa; luego en la sección III se explica cómo se unieron esas piezas. Con respecto a la parte de programación, en la sección IV, se explica cómo se logró constituir los movimientos necesarios para que el humanoide cumpla sus objetivos, mientras que en la sección V se muestran los resultados experimentales. Las herramientas y técnicas que permitieron lograr la detección de la pelota se detallan en la sección VI. También se explica la discretización del ambiente para reducir el número de estados. La comunicación de las tarjetas Arbotix y Raspberry Pi, para que puedan trabajar en conjunto, se explica en la sección VII y consideraciones especiales en la sección IX.

## II. Componentes de *Hardware*

En este capítulo se describen los principales componentes utilizados para armar la estructura del robot.

Arbotix: El controlador Arbotix es una solución de control avanzado para manejar servos Dynamixel AX/MX/RX/EX. Incorpora un potente microcontrolador AVR, radio inalámbrica XBEE, conductores de motor dual, y cabeceras de estilo servo de 3 pines para E/S digital y analógica [**?**].

Raspberry Pi: La Raspberry Pi es una computadora del tamaño de una tarjeta de crédito a la que se puede conectar un televisor y un teclado. Puede ser utilizada en proyectos de electrónica, y para muchas de las tareas que una PC de escritorio hace, como hojas de cálculo, procesadores de texto y juegos [**?**].

Motores Dynamixel Ax-12+: Son actuadores inteligentes y modulares que incorporan un reductor de engranajes, un motor DC de presión y un circuito de control con funcionalidad de red, todo en un solo paquete [**?**].

Micro Servo Analógico (actuador rotatorio): Es un motor eléctrico que permite ser controlado tanto en velocidad como en posición [**?**].

## III. Construcción de la estructura

Para la construcción del robot se ha utilizado el kit de piezas Bioloid Premium, de marca ROBOTIS, el cual incluye motores Dynamixel Ax-12+, una tarjeta controladora CM-510, un sensor Gyro, un manual, entre otros elementos. El manual incluye las instrucciones de como armar varios modelos de humanoide, el utilizado en este proyecto es el tipo B, haciendo uso de 16 motores.

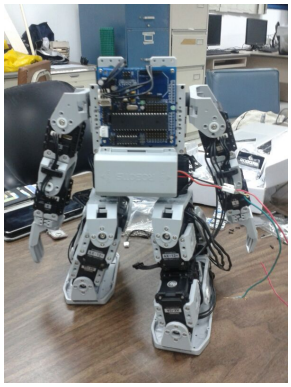

Figura 1. Parte trasera del robot con la Arbotix

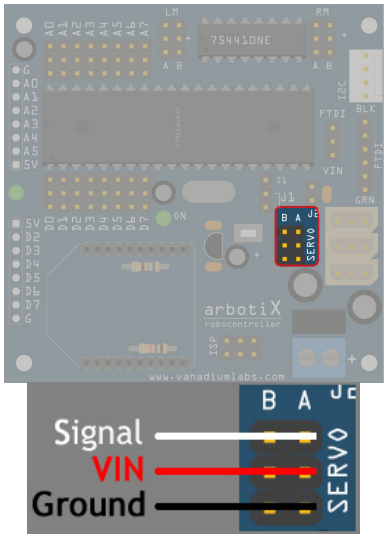


Figura 2. Ilustración de los puertos Hobby de la Arbotix

En lugar de la utilización de la tarjeta CM-510 se ha decidido usar la tarjeta controladora Arbotix, particularmente en este proyecto el cambio se ha debido a que la controladora CM-510 no acepta la incorporación actuadores o dispositivos adicionales. La Arbotix permite la incorporación de nuevos actuadores y más dispositivos con sencillez, la programación, basada en el lenguaje Processing, se ha realizado bajo el IDE de Arduino. En la figura 1 se puede observar la estructura del robot con la Arbotix incorporada. En la parte interna del tronco del robot se sitúa el sensor Gyro.

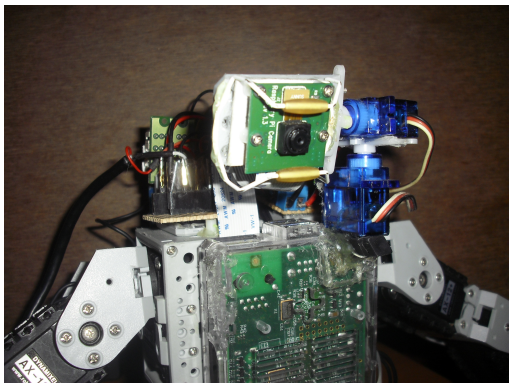

Figura 3. Vista delantera del robot con la cámara y servomotores instalados

El movimiento de la cámara se ha logrado incorporando dos servomotores, uno para el movimiento horizontal y otro para el vertical. La conexión es pin a pin en los puertos especiales para ese tipo de motores ('Hobby servo') [**?**] (ver figura 2). La cámara ha sido conectada a la Raspberry Pi en el puerto CSI. El resultado de estas tres piezas instaladas en el robot se puede apreciar en la figura 3.

## IV. Movimiento

Para que Debupa pueda cumplir sus objetivos debe ser capaz de ejecutar los siguientes movimientos:

- Caminar hacia adelante
- Girar a la izquierda
- Girar a la derecha
- Levantarse cuando ha caído boca abajo
- Levantarse cuando ha caído boca arriba
- Patear con la pierna derecha e izquierda
- Mover la cámara en sentido horizontal y vertical

Se ha logrado realizar estos movimientos a través de la fijación de poses con la utilización del software Pypose, una aplicación para modelado de poses, que permite la lectura simultánea de las posiciones de todos los motores. De esta manera se ha fijado y guardado, para Debupa, un conjunto de poses necesarias para caminar, girar, patear y levantarse. Luego se ha podido generar en Pypose una secuencia de poses para cada movimiento. Finalmente se ha exportado el archivo de las poses y secuencias para ser utilizadas en el programa a ser ejecutado en Arbotix.

Para detectar una posible caída del robot se ha incorporado el sensor giroscopio Gyro. Los valores de este sensor son leídos constantemente desde la Arbotix para corregir al robot en caso de una caída.

## V. Experimentos de movimientos

Se realizaron varios experimentos para comprobar la movilidad del robot. Primero se debe definir los movimientos que representan una unidad. Cada una de las siguientes acciones constituyen una unidad: caminar hacia adelante, girar hacia la derecha, girar hacia la izquierda, patada con la pierna derecha, patada con la pierna derecha, levantarse en la posición supino, levantarse en posición prono.

El primer experimento consistió en verificar el desempeño de las unidades de acciones, para ello se procedió a realizar la ejecución de cada unidad 50 veces, 250 ejecuciones en total. Se tomó nota de las veces que lograba realizar el movimiento sin fallas, es decir, de forma exitosa y de aquellas que lograba completar el movimiento pero con fallas. Los resultados obtenidos fueron un 99.6 % de casos exitosos y un 0.4 % de casos en los que pudo recuperarse de fallas. El 0.4 % representa un solo caso en el que el robot se ha caído y levantado.

En el segundo experimento se evaluó la ejecución de combinaciones de varias unidades de movimiento. Las combinaciones elegidas fueron:

- Caminar hacia adelante y patear con la pierna derecha
- Caminar hacia adelante y patear con la pierna izquierda
- Caminar hacia adelante, girar hacia la derecha y patear con la pierna derecha
- Caminar hacia adelante, girar hacia la derecha, patear con la pierna izquierda
- Caminar hacia adelante, girar hacia la izquierda y patear con la derecha
- Caminar hacia adelante, girar hacia la izquierda y patear con la pierna izquierda
- Girar hacia la izquierda y patear con la pierna derecha
- Girar hacia la derecha y patear con la izquierda
- Girar hacia la derecha y patear con la pierna derecha
- Girar hacia la izquierda y patear con la pierna izquierda

Se realizaron 30 ejecuciones de cada una de estas combinaciones, 300 ejecuciones en total. Los resultados se muestran en el gráfico de la figura 4. El resultado ha sido satisfactorio, obteniendo un 96 % de casos exitosos y un 4 % de casos con fallas, de las que ha podido recuperarse. La mayoría de las fallas se presentaron cuando al patear se caía, sin embargo lograba recuperarse.

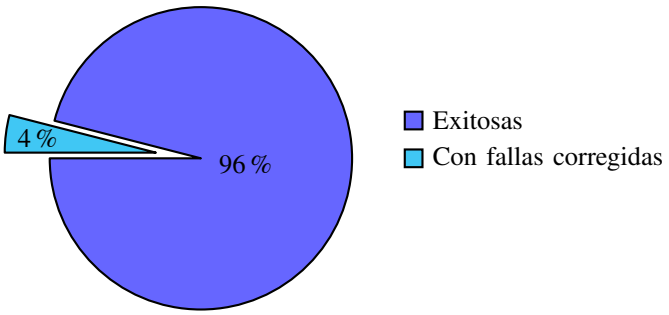


Figura 4. Experimento 2: 300 ejecuciones de acciones combinadas

## VI. Detección

En esta sección se presentan las herramientas y estrategias con las cuales se ha abordado el problema de la detección de la pelota.

### *VI-A. Integración la Raspberry Pi y la cámara*

Se ha incorporporado un mini computador Raspberry Pi 2011.12 y una cámara Raspberry Pi 1.3 que actúe como "los

ojos" del robot y así lograr obtener un modelo parcial del ambiente. La cámara ha sido instalada sobre dos servo-motores RC, otorgándole dos grados de libertad y un rango visibilidad de 160 grados en una posición fija del robot.

Estos motores se controlan desde la Arbotix usando la librería HServo [**?**], que aunque sólo puede ser usada para los motores conectados en los puertos Hobby (12 y 13), brinda la ventaja de un control más preciso, evitando que los motores tiemblen.

### *VI-B. Detección de la pelota con OpenCV*

Para lograr conocer la ubicación de la pelota en un momento dado y de forma autónoma se ha decidido aplicar detección por 'blobs', o reconocimiento de regiones, esta técnica consiste en filtrar la imagen por color. Por ello es importante que el color de la pelota no se repita en el ambiente y así poder obtener la posición de la pelota dentro de la imagen. Se han utilizado las librerías de OpenCV para apoyar esta tarea.

La imagen es captada en el modelo de color RGB y se transforma al HSV. Luego se aplica la función inRange de OpenCv para obtener una imagen en blanco y negro, derivada de la imagen original, en la que se identifica con blanco la región ocupada por la pelota y el resto de la imagen en negro.

Para disminuir el ruido y los posibles elementos aislados que pueda tener la imagen se han aplicado los siguientes filtros o transformaciones: morfología en apertura y en clausura. La morfología en apertura es una transformación de una imagen que consiste en aplicar la operación de erosión seguida de la operación de dilatación [**?**]. La morfología de clausura es una transformación que aplica la dilatación seguida de la erosión.

### *VI-C. Discretización de la imagen y acciones del robot*

El hecho de que la cámara tenga dos grados de libertad para moverse es una gran ventaja, ya que se puede obtener un mayor rango de visión. Debupa puede mirar hacia la derecha o izquierda sin tener que mover sus pies, también puede mirar hacia abajo para verificar si la pelota está tan cerca como para patearla, o hacia arriba para ubicar la pelota a mayor distancia. La desventaja es que agrega mayor complejidad.

Debupa debe tomar una acción diferente dependiendo de la posición de la cámara y de la pelota en la imagen. Sin embargo esto genera una gran cantidad de estados, por lo cual se ha decidido discretizarlos de la siguiente manera:

- La cámara tiene 9 (3x3) posibles posiciones. La visión horizontal abarca 3 cuadros, aproximadamente 160 grados. La visión vertical también abarca 3 cuadros, llega a captar la imagen desde sus pies hasta más de 2 metros hacia adelante.
- Desde cada posición de la cámara se obtiene una imagen y, dependiendo de la posición, la imagen se divide en diferentes regiones.

Las imágenes de la cámara obtenidas en la posición central, y en la posicioń central inferior (ver figura 5) son las más importantes y prioritarias, pues si la pelota se detecta en ellas significa que el robot está cerca de poder patearla. Estas dos imágenes se dividen en las regiones que pueden ser apreciadas

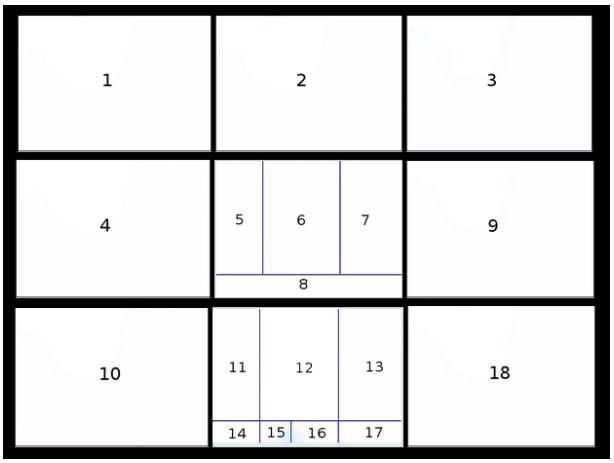


Figura 5. Campo de visión del robot con el número de cada región

en la figura 5. Cada región ayuda a identificar la acción que Debupa debe tomar. Por ejemplo, en el cuadro central inferior, cuando la pelota se encuentra del lado derecho de la pantalla (región 13 o 17) el robot debe girar a la derecha para situarse de frente a la pelota. El área de pateo se encuentra en las regiones 15 y 16.

A continación se especifica la acción a tomar en cada región:

- Girar a la Izquierda: Debupa debe girar a la izquierda cuando la pelota se encuentre en alguna de las siguientes regiones: 1, 4, 10, 5, 11, 14.
- Girar a la Derecha: debe girar a la derecha cuando la pelota se ubique en alguna de las siguientes regiones: 7, 13, 17, 3, 9, 18.
- Caminar hacia adelante: cuando la pelota se ubique en alguna de las siguientes regiones: 12, 6, 2.
- Patear con la pierna izquierda: cuando la pelota se encuentre en la región 15.
- Patear con la pierna derecha: cuando la pelota se encuentre en la región 16.

## VII. Integración de los componentes

La Raspberry Pi procesa la información de la cámara y la Arbotix controla los actuadores. Para coordinar los movimientos del robot según la posición de la pelota se ha establecido una forma de comunicación entre ambas tarjetas.

Se seleccionado la Arbotix como servidor de peticiones y a la Raspberry Pi como cliente. Dentro de la Raspberry Pi se ejecuta el proceso de decidir qué acción debe tomar el robot. Una vez determinada la acción se envía la petición a la Arbotix para que esta la ejecute. Este proceso es bidireccional y síncrono, es decir, la Raspberry envía la petición y se bloquea hasta que la Arbotix retorne la respuesta de su culminación.

Para la implementación de la comunicación se ha usado ROS (Robot Operating System) con la utilización de los servicios que no es más que un método de comunicación basado en el paradigma de request/reply con el concepto de maestro esclavo.

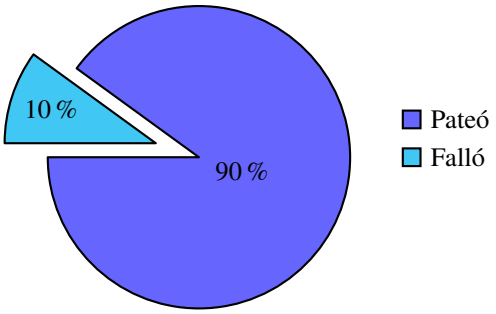


Figura 6. Experimento 3: Pruebas de movimientos integrados CASO I

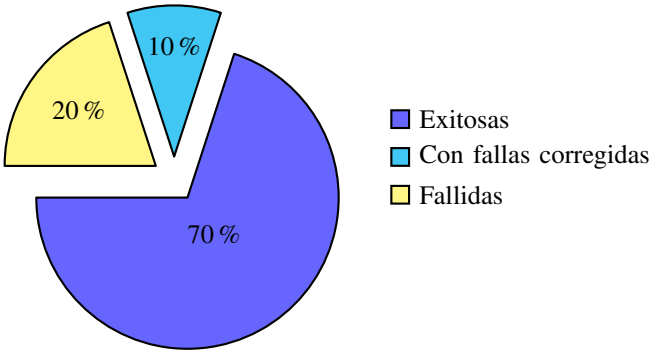


Figura 7. Experimento 3: Pruebas de movimientos integrados CASO II

## VIII. Experimentos con comportamientos integrados

El tercer experimento consistió en la realización de pruebas de desempeño en donde se observó el comportamiento global del robot con todos sus comportamientos integrados (detección, búsqueda y pateo).

El primer caso (CasoI) consistió en la colocación de la pelota en la zona de pateo (las regiones 15 y 16 de la figura 5). El número de pruebas en este primer caso fue de 10, cuya duración aproximada de cada prueba fue de 30 segundos. En la figura 6 se puede observar los resultados obtenidos de esta prueba.

El segundo caso (CasoII) consistió en en la colocación inicial de la pelota a una distancia de 50 cm en línea recta al robot. Se realizaron 10 pruebas, siendo la duración promedio de 2 minutos con 12 segundos. Los resultados de dicho caso se puedo observar en la figura 7.

El tercer caso (CasoIII) consistió en en la colocación inicial de la pelota a una distancia de 50 cm en línea recta al robot y 10 cm a la izquierda formando asi una diagonal. Se realizaron 10 pruebas, siendo la duración promedio de 3 minutos con 29 segundos. Los resultados de dicho caso se puedo observar en la figura 8.

El cuarto caso (Caso IV) consistió en en la colocación inicial de la pelota a una distancia de 50 cm en línea recta al robot y 10 cm a la derecha. Se realizaron 10 pruebas, siendo la duración promedio de 4 minutos con 48 segundos. Los resultados de dicho caso se puedo observar en la figura 9.

Con un total de 40 pruebas de comportamiento integrado los resultaron obtenidos fueron satifactorios con un porcentaje

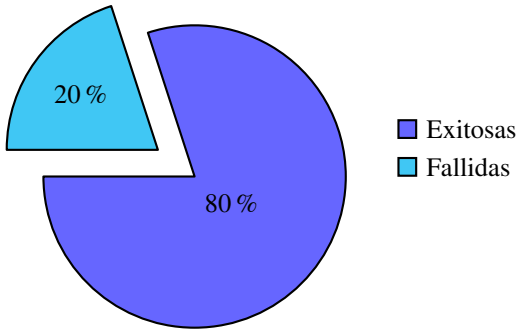


Figura 8. Experimento 3: Pruebas de movimientos integrados CASO III

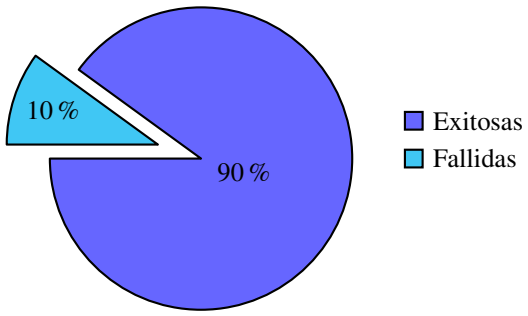


Figura 9. Experimento 3: Pruebas de movimientos integrados CASOIV

de logro del 82 % mas un 5.5 % en el cual se logro recuperar y finalizar la tarea, con 12.5 % de fallo en la tarea se pudo observar de manera general que la razon de dicha falla es debido a problemas de bateria.

## IX. Consideraciones especiales

El uso de los motores Dynamixel Ax-12+ debe ser cuidadoso dado que por el exceso de esfuerzo, si no se controla la temperatura máxima o el torque máximo permitido, suelen quemarse con facilidad. El sensor de temperatura es capaz de medir valores entre -5°C y 80°C, y el torque puede ir desde 512 hasta 1023 kgf-cm. La solución aplicada ha sido establecer en el chip de cada motor una cota máxima de 30°C de temperatura y un torque máximo de 800 kgf-cm. En caso de superar alguna de estas cotas los motores se apagan automáticamente evitando así un daño mayor. Para lograr esto se ha tenido que modificar la librería Ax12 agregando algunos procedimientos que permitieran establecer la temperatura y el torque máximo.

Para ampliar información sobre este proyecto se encuentra disponible el blog de Debupa en donde se presenta la descripción de algunas de las tareas que han presentado problemas con sus soluciones (https://sites.google.com/site/arbotixml3j/home).

## X. Conclusiones

La presente investigación ha nacido de la motivación por hacer que en Venezuela se incursione en proyectos que involucren humanoides autónomos e inteligentes. Se ha inspirado especialmente en la categoría Robocup soccer de la competencia internacional Robocup. Desde 1997, fecha en la que inició la competencia, Venezuela nunca ha participado en categorías

con humanoides, mientras que países latinoamericanos como México, Brasil y Colombia sí han tenido avances en este campo. Si bien este proyecto no cumple con todas las reglas de la competencia se espera que éste pueda dar pie a continuar investigaciones dentro de Venezuela.

Los componentes utilizados en este proyecto son relativamente económicos comparados con otros en el mercado. La integración del kit Bioloid Premium con la Arbotix y la Raspberry Pi, ha hecho posible construir un humanoide inteligente sin tener que invertir exorbitantes cantidades de dinero. Una de las contribuciones más importantes es la coordinación y paralelismo exitoso entre todos los componentes utilizados.

Las mejoras que se pueden incorporar al proyecto podrían ser: la inclusión de aprendizaje por reforzamiento para patear de forma exitosa, incluir que la patada sea en dirección al arco e incluso añadir aprendizaje por reforzamiento para hacer que el robot pueda predecir la posición de una pelota en movimiento para que pueda patearla en el momento indicado.