

# Implementación de un sistema inalámbrico de adquisición de datos en tiempo real para un droide caminante de 4 gdl

***José Luis Monay Arredondo***

Instituto Tecnológico de Ensenada

*jimonay@ite.edu.mx*

***Iliana Marlen Meza Sánchez***

Instituto Tecnológico de Ensenada

*imeza@ite.edu.mx*

## **Resumen**

En el presente trabajo se presenta el desarrollo de un sistema de adquisición de datos en tiempo real utilizando un protocolo inalámbrico de comunicación ZigBee para retroalimentar e implementar tareas en un robot caminante de 4gdl (grados de libertad). Para ello se han diseñado bloques para establecer comunicación inalámbrica, contruidos a partir de funciones S (S-functions), y programadas en lenguaje C que permiten crear modelos para la implementación de tareas con ejecución en tiempo real (modo normal) en Simulink de Matlab. La efectividad del esquema propuesto es evaluada experimentalmente a través del diseño de una tarea de caminata para el sistema robótico.

**Palabras Claves:** adquisición inalámbrica de datos, Matlab, Simulink, sistema robótico bípedo, tiempo real.

## 1. Introducción

El interés en el desarrollo de aplicaciones de tiempo real para robots tipo humanoides se ha incrementado en los últimos años. Tareas como la generación de movimiento para robots bípedos, planeación de trayectorias para lograr caminata, algoritmos basados en energía, sincronización de articulaciones para lograr transiciones suaves o lograr patrones de movimiento específicos, entre otras, se encuentran bajo estudio. Asimismo, la búsqueda de soluciones factibles que permitan lograr resultados experimentales que apoyen los desarrollos teóricos en la línea de generación del conocimiento de la Robótica ha llevado a que diversas compañías desarrollen plataformas educativas, comerciales e industriales (Véase por ejemplo [1]).

En el presente artículo se describe un sistema inalámbrico de adquisición de datos para implementar retroalimentación sobre un sistema robótico bípedo de cuatro articulaciones denominado Droide Caminante (Walking Droid).

La principal contribución corresponde a un resultado que no ha sido reportado en la literatura. Se pretende que pueda ser utilizado en el ámbito educativo para el entrenamiento de estudiantes de licenciatura en ingeniería para la comprensión de áreas relacionadas con la robótica y el control tales como análisis de movimiento, cinemática, mecatrónica, sistemas de control retroalimentados, teoría clásica de control, entre otros.

El software como parte de un sistema embebido se utiliza para controlar los productos electrónicos a nivel lógico [2]. Con el desarrollo de estos sistemas se pretende que los programadores, que no estén acostumbrados, entiendan de manera clara y precisa lo que este software significa. También se pretende que sea de ayuda para la dar solución a problemas que impliquen un desarrollo tecnológico, ya que se ha vuelto muy común el uso de sistemas embebidos como productos inteligentes.

Este trabajo está organizado de la siguiente forma. El sistema inalámbrico de adquisición de datos incluyendo la descripción de las características del sistema robótico, requerimientos del sistema, instalación y configuración, además de los alcances del esquema propuesto se presentan en la sección 2. La tarea de caminata para el robot

bípedo incluyendo resultados experimentales es presentada en la sección 3. Finalmente, conclusiones y trabajo futuro se enuncian en la sección 4.

## **2. Sistema inalámbrico de adquisición de datos**

### **2.1 Descripción del sistema robótico**

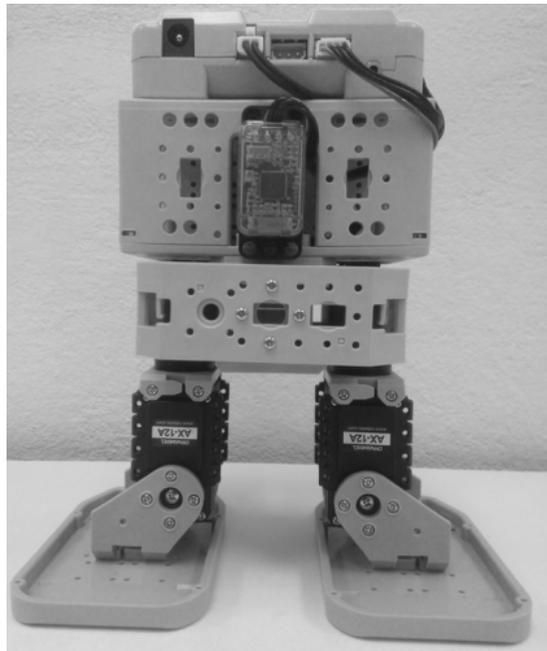
Bioid [3] de la compañía Robotis es un kit robótico educacional utilizado en varias instituciones e instalaciones de ingeniería científica para enseñar los principios de la cinemática y la dinámica de los movimientos robóticos con el objetivo de desarrollar e implementar estrategias de control y modelado matemático con validación experimental. Además, la comunidad en general los está utilizando en competiciones de robótica y por aficionados debido a la facilidad de la programación y el software proporcionado por el fabricante.

Este kit robótico puede ser configurado de varias formas, las cuales van aumentando el grado de complejidad dependiendo de los componentes que se utilicen. En Fig. 1 se muestran algunos ejemplos de configuración extraídos del manual del fabricante.



**Fig. 1. Algunos tipos de configuración del kit robótico Bioloid [7].**

Se propone la utilización del kit robótico Bioloid bajo la configuración de un droide caminante (Walking Droid) de 4 grados de libertad (Véase Fig. 2.).



**Fig. 2. Configuración de droide caminante de 4 articulaciones.**

Los elementos y características que describen este sistema robótico son:

- a) Controlador CM-510 (microcontrolador ATmega 2561).
- b) 4 Servomotores de corriente directa Dynamixel AX-12+.
- c) Comunicación inalámbrica mediante el protocolo ZigBee.
- d). Ejecución en Tiempo Real (Modo Normal) con Simulink de Matlab.

### **Controlador CM-5/CM-510/CM-700/CM-530.**

Uno de los componentes principales es el controlador, que en un robot desempeña el papel del cerebro. Aquí es donde se le programan los códigos de control al robot. Para el kit robótico Bioloid existen varios controladores CM-5/CM-510/CM-700/CM-530 de manera cronológica. Estos controladores están diseñados sobre un microcontrolador:

CM-5 basado en un microcontrolador AVR ATmega 128, CM-510/CM-700 basado en AVR ATmega 2561 y el CM-530 basado en ARM Cortex STM32F103RE.

Estos controladores pueden ser programados en lenguaje C, y cada uno tiene su software para generar el código que se le programará. Se utilizan principalmente para mantener ciertos ángulos en virtud de algunas condiciones tales como las articulaciones en movimiento, moviendo dispositivos de dirección, etc.

### **Servomotores de corriente directa Dynamixel.**

Son los actuadores del sistema robótico; mediante su programación se genera el movimiento de la tarea a realizar. Para los controladores antes mencionados existen los servomotores Dynamixel, de los cuales los Dynamixel de las series AX y MX\_TTL son soportados por estos controladores.

Cada servomotor Dynamixel de la serie AX tiene un rango de 0 a 300 grados de movimiento o de 0 a 1023 pulsos, esto significa que tiene una resolución de 0.29 grados por pulso si es configurado en modo articulación, y un rango de giro sin fin si es configurado en modo rueda. Cada servomotor consta de una única ID, y un protocolo de comunicación serial asíncrona semidúplex (half-duplex); esto significa que la transmisión y recepción de datos utilizan el mismo bus y por lo tanto no puede ser simultánea.

Otra característica de estos servomotores es que tienen una tabla de control que consiste en datos sobre el estado actual y el funcionamiento, la cual se encuentra programada en el microcontrolador interno. El usuario puede controlar el Dynamixel cambiando los datos de la tabla de control a través de paquetes de instrucciones.

### **Comunicación inalámbrica**

Dentro del sistema robótico Bioloid es posible establecer comunicación inalámbrica a través de Bluetooth y ZigBee, los cuales la cual hacen posible la comunicación serial (UART) del Bioloid con la PC.

Bluetooth presenta características similares al dispositivo ZigBee, como el ancho de banda. Pero ZigBee presenta algunas ventajas sobre Bluetooth, como menor consumo de energía, mayor número de nodos en una red, mayor rango de operación [6].

Para hacer posible esta comunicación se utiliza la interface USB2Dynamixel, que es un dispositivo que se utiliza para operar el servomotor directamente desde el PC. Esta interface se puede utilizar para cambiar de puerto USB a puerto serie de la PC cuando el dispositivo Zig2Serial se conecta al puerto USB; de esta forma, es posible controlar los robots de forma inalámbrica a través de ZigBee. ZigBee puede ser controlado a través de la librería ZigBee SDK en lenguaje C, disponible por el fabricante.

## 2.2. Descripción y requerimientos

El diagrama a bloques del sistema propuesto se muestra en Fig. 3.

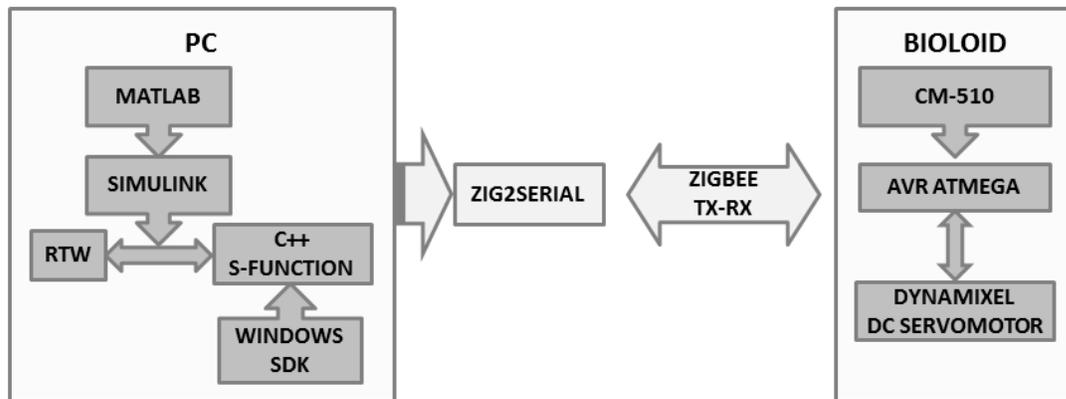


Fig. 3. Diagrama a bloques del esquema propuesto.

El software para implementar el esquema propuesto es el siguiente:

- a) **Roboplus, Embeded\_c (para el Controlador CM-510) y ZigBee SDK.**  
Corresponde al software y librerías proporcionadas por el fabricante para verificar el funcionamiento del sistema robótico, enviar comando a cada servomotor y cargar el firmware del microcontrolador.
- b) **Atmel Studio y WinAVR.** Software y compilador en lenguaje de programación C para generar el firmware que se descargara al sistema robótico mediante Roboplus.
- c) **Visual Studio Express, Windows SDK 7.1 y .NET Framework 4.0.**  
Compiladores de lenguaje de programación C que permitirá la generación de funciones para ser embebidas en bloques de Simulink de Matlab.
- d) **Matlab/Simulink.** Software para computación numérica, visualización y programación. Es posible utilizarlo para análisis de datos, desarrollo de algoritmos y creación de modelos y aplicaciones.

Es importante señalar que todo el software del esquema propuesto es gratuito a excepción de Matlab que necesita licencia.

### 2.2.1 Instalación y configuración

Para el diseño del sistema de adquisición se debe tomar en cuenta las compatibilidades entre Matlab/Simulink, Visual Studio Express, .NET Framework y Windows SDK [4]. Es importante señalar que se debe consultar la compatibilidad correspondiente a la versión de Matlab que se vaya a utilizar. En el sistema de adquisición propuesto se utilizó Matlab 2012a, Visual Studio Express 2010, Microsoft Windows SDK 7.1 y .NET Framework 4.0 en un sistema Windows 7 de 64 bits.

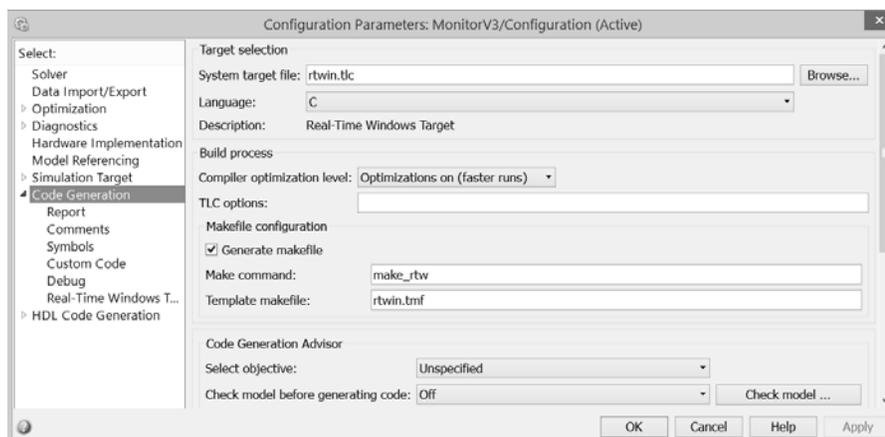
Para verificar que tenemos instalados el kernel de tiempo real de Matlab (rtwintgt), en la ventana de comandos de Matlab escribimos **rtwho** y nos mostrara la versión que tenemos instalada. Si no se encuentra instalado el kernel de tiempo real, en la ventana de comandos escribimos **rtwintgt –install** y procedemos a su instalación.

Para instalar el compilador que utilizaremos en Matlab, primero instalaremos Visual Studio Express (Visual C++), y luego Windows SDK 7.1. Una vez instalados, escribimos **mex –setup** en la ventana de comandos de Matlab y seleccionamos el compilador SDK (Microsoft Software Development Kit (SDK) 7.1).

Antes de instalar el software de Atmel Studio, se recomienda registrar las librerías de ZigBee SDK en las variables de entorno de Windows e instalar WinAVR. Asimismo, es recomendable instalar la versión más reciente de Roboplus.

### 2.2.2 Ejecución en tiempo real (modo normal de Simulink).

La ejecución de aplicaciones en tiempo real para Simulink de Matlab puede realizarse bajo dos configuraciones: modo normal y modo externo [5, 6]. En ambos casos, la generación de la aplicación es realizada por el kernel de tiempo real denominado Real-Time Windows Target. De esta forma, la creación de un modelo de aplicación necesita configurarse con estas características y para acceder a ella es necesario seleccionar **Simulation → Model Configuration Parameters → Code Generation** del menú superior para cada modelo creado. Una vez seleccionada esta opción, la configuración para ejecución debe ser tal como se muestra en Fig. 4.



**Fig. 4. Configuración del modelo en Simulink para ejecución en tiempo real (Real-Time Windows Target).**

En el caso del sistema de adquisición de datos que se presenta en este trabajo, se utiliza el modo de ejecución normal. La razón de esto es que las librerías para la comunicación proporcionadas por el fabricante (zigbee.lib, dynamixel.lib) utilizan funciones de Windows (windows.h) que no pueden ser implementadas bajo la configuración de modo externo en Simulink de Matlab. De esta forma, el tiempo real es garantizado mediante la inclusión de un bloque denominado "Real-Time Synchronization" que permite establecer un control sobre los plazos de ejecución no cumplidos (missed ticks).

### 2.2.3 Diseño de aplicaciones

El sistema inalámbrico de adquisición propuesto permite el diseño de aplicaciones de diversa naturaleza sobre el sistema robótico. Las características que cada aplicación creada debe cumplir para poder ser ejecutadas en tiempo real son:

- a. **Funciones S programadas en lenguaje C.** Estas funciones se construyen de acuerdo a la aplicación para envío y recepción de datos inalámbricamente. Dado que se embeben dentro de un modelo de Simulink es posible construir una aplicación dinámica utilizando los bloques de diversas herramientas. Toda función desarrollada debe incluir el siguiente encabezado:

```
#pragma comment (lib, "zigbee.lib")
#include "dynamixel.h"
#include "serial.h"
#include "zigbee.h"
```

Dentro de estas librerías destacan las funciones: a) `dxl_write_word(int, int, int)` que sirve para mover cada servomotor y b) `zgb_rx_check()`, `zgb_rx_data()` y

zgb\_tx\_data(int) que son funciones necesarias para envío y recepción inalámbricos de datos.

- b. Configuración para ejecución en tiempo real.** Cada aplicación diseñada en Simulink implica la necesidad de configurar la ejecución del modelo no sólo seleccionando el kernel de tiempo real (Real-Time Windows Target) sino que también es necesario especificar tiempo de muestreo y la inclusión del bloque “Real-Time Synchronization”.
- c. Firmware del microcontrolador del robot.** La ejecución de tareas de manera autónoma por parte del sistema robótico es a través de la programación de los movimientos y funciones deseadas y su posterior descarga al controlador que corresponda. En este trabajo se utilizó el CM-510 y su programación fue desarrollada con WinAVR/Atmel Studio. Sin embargo, la programación puede ser extendida a los demás de manera directa considerando que es en lenguaje C y es posible descargar ejemplos consultando el manual en línea del fabricante [7].
- d. Diseño de la tarea.** Esta opción no es necesaria pero si recomendable. Todo sistema robótico posee limitaciones; por lo tanto, la primera prueba de los movimientos a realizar se recomienda que sea probada mediante Roboplus. De esta forma, es posible detectar errores de implementación: restricciones de las articulaciones y movimientos no posibles.

### **3. Experimento: Caminata con retroalimentación inalámbrica**

La configuración utilizada en esta aplicación fue solo de monitoreo, donde los pasos de la tarea se encuentran programados en el microcontrolador. El droide caminante trae una tarea de caminata diseñada por el fabricante, esta tarea consta de seis pasos que se estarán repitiendo indefinidamente una vez que se inició la tarea. Los valores de esta tarea fueron adaptados para utilizarse en una comunicación inalámbrica a través del protocolo ZigBee entre la PC y el droide caminante. La computadora solo controla los tiempos en los que se va a realizar la tarea comunicándose inalámbricamente con el robot a través de ZigBee.

La tarea consta de 8 pasos, en donde el paso cero corresponde a la posición inicial (central) de todos los servomotores Dynamixel (Véase Tabla 1). El valor de cada paso está dada en pulsos, y los identificadores (ID) de los Dynamixel fueron previamente configuradas con Roboplus.

Dynamixel [ID]	Pasos [Pulsos]							
	0	1	2	3	4	5	6	7
1	512	646	646	448	377	372	587	646
2	512	650	650	437	375	367	590	650
3	512	512	444	412	512	643	682	512
4	512	512	359	326	512	572	607	512

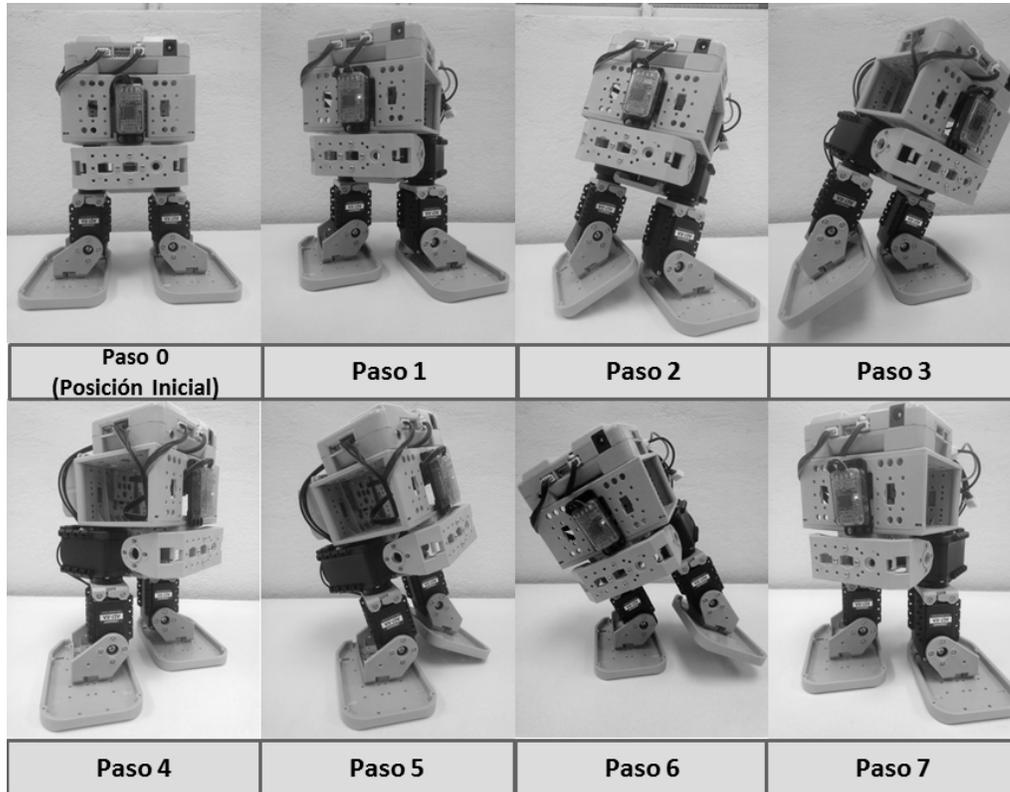
**Tabla 1. Tarea para caminata del robot droide caminante.**

En Fig. 5 se muestra como la tarea de caminata para el robot caminante en forma de pasos. Todas las posturas mostradas corresponden a la finalización de cada paso. Si se deseara realizar una caminata más larga, solo bastaría con repetir desde el paso 1 hasta el paso 7.

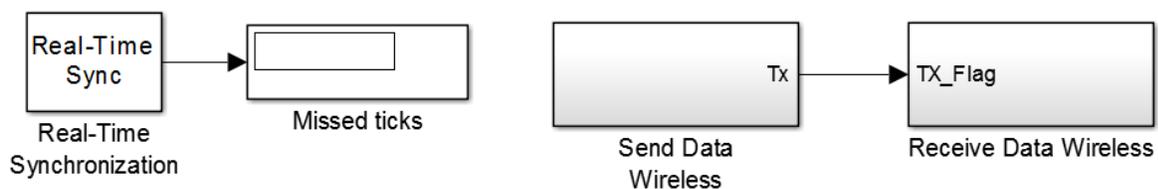
El modelo construido en Simulink de Matlab para realizar la tarea propuesta se muestra en Fig. 6. De esta forma, el bloque de envío de datos corresponde a una bandera de activación para el bloque de recepción de datos.

Como se había mencionado anteriormente, el bloque “Real-Time Synchronization” se encarga de contabilizar los plazos de ejecución no cumplidos de la aplicación (missed

ticks) cuando el modelo a ejecutarse no puede mantenerse a la par con la ejecución del kernel de tiempo real.



**Fig. 5. Movimiento de caminata del robot droide caminante.**



**Fig. 6. Diagrama a bloques de Simulink.**

Los subsistemas del modelo en Simulink de Fig. 6 denominados “Send Data Wireless” y “Receive Data Wireless” se muestran en Fig. 7 y 8, respectivamente. Obsérvese que el

subsistema “Receive Data Wireless” consta de cuatro bloques donde cada uno corresponde a la lectura de la posición de un servomotor.

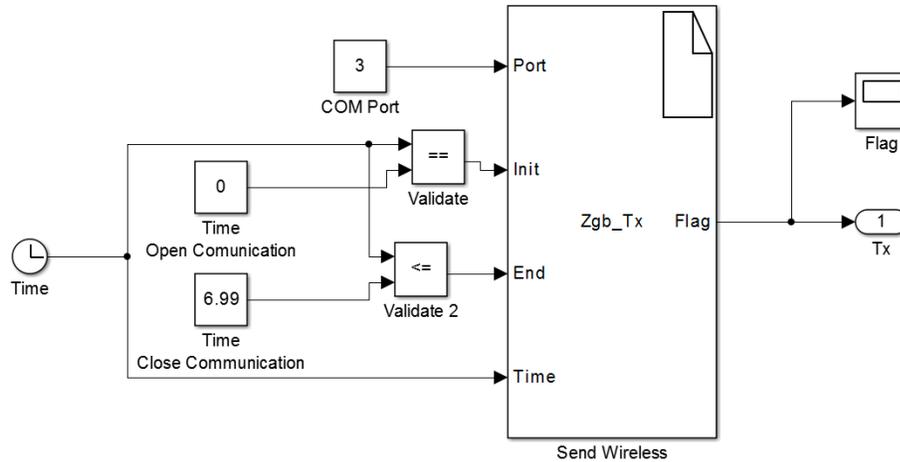


Fig. 7. Subsistema “Send Data Wireless” en Simulink.

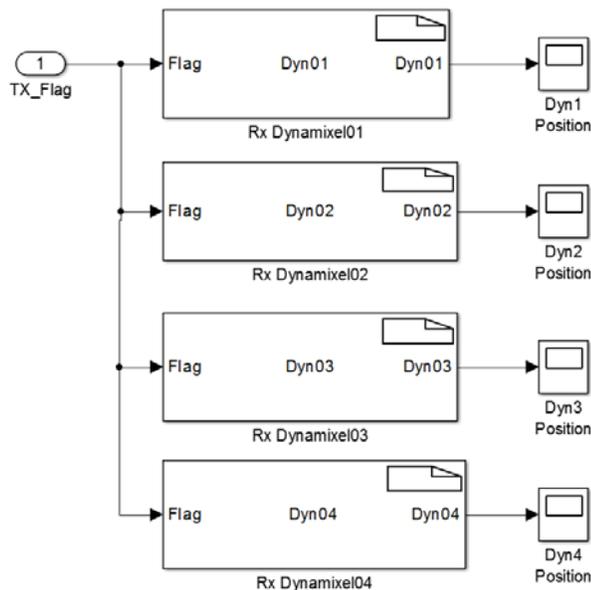
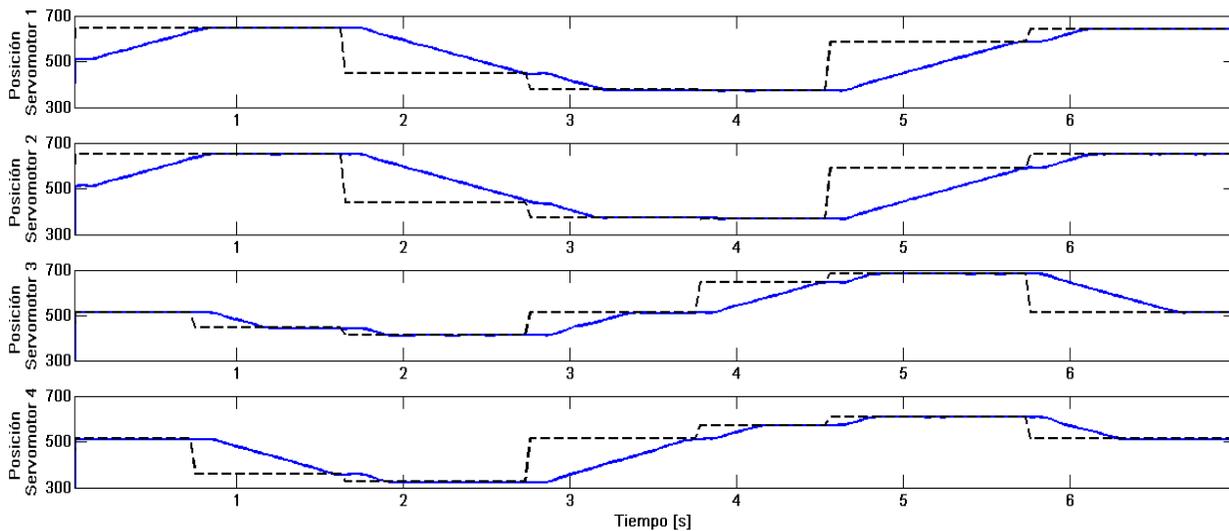


Fig. 8. Subsistema “Receive Data Wireless” en Simulink.

### 3.2 Resultados experimentales

En Fig. 9 se presentan los resultados experimentales correspondientes a las posiciones medidas de cada servomotor y obtenidas mediante la retroalimentación inalámbrica en Simulink de Matlab. Obsérvese un aparente retraso en el momento en que la referencia (línea punteada) es establecida y la reacción de cada servomotor para seguir dicha trayectoria; esto es debido al tiempo que le toma al dato llegar una vez que ha sido enviado inalámbricamente. Nótese entonces que es posible aproximar numéricamente, al menos de manera experimental, los retardos generados por la comunicación inalámbrica.



**Fig. 9. Posiciones de las articulaciones del droide caminante: línea punteada para posiciones deseadas y línea sólida para posiciones medidas.**

### 4. Conclusiones y trabajo futuro

En el presente trabajo se propuso un esquema inalámbrico de adquisición de datos para un sistema robótico bípedo con cuatro articulaciones. Esta plataforma permite la implementación de tareas de movimiento en el robot y puede ser también utilizada en el

ámbito educativo y como una plataforma experimental para apoyar desarrollos teóricos dentro del área de la robótica. Este esquema puede extenderse para ser aplicado en el robot de 18 articulaciones que permite armar el sistema robótico Bioloid. Asimismo, también es posible validar modelos cinemáticos directos e indirectos para robots en las diversas configuraciones disponibles (Véase por ejemplo los desarrollos presentados en [8, 9]).

Es importante señalar que una de las limitaciones propias del sistema al incrementar el número de articulaciones del sistema robótico corresponde a la capacidad de la computadora en donde se ejecuta la aplicación. Una característica generada a partir de la naturaleza inalámbrica de la comunicación corresponde a la necesidad de establecer mecanismos de validación de datos; esto es, no hay garantía de que el receptor reciba los datos en el mismo orden en el que fueron enviados por el transmisor. En el experimento realizado para la tarea de caminata se implementó la concatenación de información adicional para poder determinar cuál servomotor corresponde al valor de posición recibido.

Algunos aspectos a ser abordados como trabajo futuro corresponden a la necesidad de realizar un análisis de los retrasos generados por la implementación de una adquisición inalámbrica de datos y la forma en que influye en el desempeño de la aplicación. De igual forma, el trabajo en progreso es con la finalidad de implementar mecanismos para tolerancia a fallos y la obtención de modelos dinámicos para la implementación de algoritmos de control.

## **5. Referencias**

- [1] Humanoid Robots (IEEE Spectrum), <http://spectrum.ieee.org/robotics/humanoids>.
- [2] G. Galeano, Programación de Sistemas Embebidos en C. 1ª Ed. 2009. Alfaomega. México.

- [3] Robotis. Bioloid Premium Robotic Kit. [http://www.robotis.com/xe/BIOLOID\\_Premium\\_en](http://www.robotis.com/xe/BIOLOID_Premium_en). Consultado en Septiembre del 2014.
- [4] The Mathworks Inc. Supported and Compatible Compilers – Release 2012a. <http://www.mathworks.com/support/compilers/R2012a/index.html>. Consultado en Septiembre 2014.
- [5] The Mathworks Inc. Real-Time Execution in Normal Mode. <http://www.mathworks.com/help/rtwin/ug/simulink-real-time-normal-mode.html>. Consultado en Septiembre 2014.
- [6] The Mathworks Inc. Real-Time Execution in External Mode. <http://www.mathworks.com/help/rtwin/ug/simulink-external-mode.html>. Consultado en Septiembre del 2014.
- [7] Robotis. E-manual. <http://support.robotis.com/en/>. Consultado en Septiembre del 2014.
- [8] M. Akhtaruzzaman, A.A. Shafie. "Geometrical analysis on BILOID humanoid system standing on single leg", *Mechatronics (ICOM)*, 2011 4th International Conference On , pp. 1 – 5. doi: 10.1109/ICOM.2011.5937193
- [9] J.V. Nuñez, A. Briseño, D.A. Rodriguez, J.M. Ibarra, V.M. Rodriguez, "Explicit Analytic Solution for Inverse Kinematics of Bioloid Humanoid Robot", *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS)*, 2012 Brazilian, pp. 33 – 38. doi: 10.1109/SBR-LARS.2012.62.
- [10] The Mathworks Inc. Basic C MEX S-function. <http://www.mathworks.com/help/simulink/sfg/example-of-a-basic-c-mex-s-function.html>. Consultado en Septiembre del 2014.

## 6. Autores

José Luis Monay Arredondo obtuvo el grado de Ingeniero en Electrónica en el Instituto Tecnológico de Ensenada. Actualmente es estudiante de la Maestría en Ciencias de la Ingeniería del programa de Posgrado del Instituto Tecnológico de Ensenada.

Iliana Marlen Meza Sánchez es profesor investigador de la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Ensenada.