

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# **Diseño e implementación de una estrategia de control cinemática para robot hexápodo orientado a la evasión de obstáculos**

Santiago Lotero Muñoz

Julián Arango Correa

Ingeniería Mecatrónica

Juan Sebastián Botero

**INSTITUTO TECNOLÓGICO METROPOLITANO**

**5 de Agosto de 2016**

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# RESUMEN

---

Una plataforma de desarrollo robótico puede ser pensada y utilizada como una herramienta para la investigación y experimentación, estas son usadas para la realización de pruebas, modelado y validación de un sistema de control que se desee implementar para su manejo; además de la examinación de algoritmos de navegación autónomos. En esta carrera por buscar tecnologías que nos permitan emular cada vez más el funcionamiento y comportamiento de la naturaleza se han creado plataformas que nos permiten copiar los movimientos de muchos animales, entre ellos los insectos, los cuales a través del tiempo han desarrollado diversas formas de locomoción, algunas de estas son la inclusión de 2 o más extremidades; por la facilidad de movimiento una de las más reproducidas en la robótica es la que involucra 6 extremidades, las cuales se han convertido en plataformas de movimiento hexápodos todoterreno.

El presente proyecto consiste en el desarrollo e implementación de una estrategia de control del robot hexápodo Scorpion de la compañía Robotis (serie Bioloids Premium) para la evasión de obstáculos de una manera eficaz y precisa mejorando el actual algoritmo de movimiento del robot. El robot Scorpion posee 18 servomotores digitales, los cuales cuentan con diversas ventajas con respecto a los servomotores convencionales, entre ellas se encuentra la posibilidad de tener una mejor precisión y un mayor par de torsión, además de un controlador CM-530 el cual integra la batería y requiere un bajo consumo de corriente para su funcionamiento, el resto de la arquitectura consiste en una carcasa plástica que sostiene los servomotores.

*Palabras clave:* Hexápodo, Algoritmo de navegación, Robótica, servomotores.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# RECONOCIMIENTOS

---

Agradecemos a nuestras familias y amigos por el constante apoyo durante nuestro proceso académico, a los diferentes profesores que nos brindaron sus enseñanzas para lograr el conocimiento adquirido, a nuestros compañeros que de una u otra manera ayudaron en nuestro proceso académico, a Juan Sebastián Botero por su apoyo incondicional y asesoría en el desarrollo de este proyecto además del ITM por facilitar los recursos para la realización de este trabajo y proporcionar los espacios para el crecimiento como personas y profesionales.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# ACRÓNIMOS

---

IEEE	Instituto de Ingenieros Eléctricos y Electrónicos.
CW	Clock Wise (sentido horario)
CCW	Counter Clock Wise (Sentido anti horario)
ISO	Organización Internacional para la estandarización
PETMAN	Protection Ensemble Test Mannequin
DARPA	Agencia de investigación de proyectos de defensa avanzados

# TABLA DE CONTENIDO

TABLA DE FIGURAS.....	6
1. INTRODUCCIÓN.....	7
1.1 Generalidades.....	7
1.2 Justificación.....	8
1.3 Objetivos.....	9
1.3.1 Objetivo General.....	9
1.3.2 Objetivos Específicos.....	9
1.4 Organización de la tesis.....	9
2. MARCO TEÓRICO.....	11
2.1 Historia breve de los robots con extremidades.....	12
2.1.1 Trabajos similares.....	14
2.2 Conceptos Introdutorios.....	14
2.2.1 Robots.....	14
2.2.2 Dispositivo Robótico.....	14
2.3 Sistema mecánico.....	14
2.3.1 Elementos de fijación.....	15
2.4 Actuadores eléctricos.....	15
2.5 Cinemática del robot.....	15
2.5.1 Matrices de rotación.....	15
2.5.2 Transformadas homogéneas.....	16
2.5.3 Estándar Denavit Hartenberg.....	16
3. METODOLOGÍA.....	19
4. RESULTADOS Y DISCUSIONES.....	30
4.1 Simulación cinemática y de movimiento del robot hexápodo.....	30
4.2 Desarrollo del algoritmo para la estrategia de control para la evasión de obstáculos.....	33
5 CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS.....	41
5.1 Conclusiones.....	41
5.2 Recomendaciones y trabajos futuros.....	41

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Referencias.....42

Apéndices.....44

## TABLA DE FIGURAS

Figura 1. Diagrama del sistema robótico.....	11
Figura 2. Denavit Hartenberg.....	16
Figura 3. Coordenadas del robot.....	17
Figura 4. Simulación movimiento hacia adelante MATLAB.....	20
Figura 5. Entorno gráfico y simulación de movimiento en R+ Motion.....	20
Figura 6. Simulación de los datos sacados del software R+ Motion mostradas en las tablas anteriores en MATLAB.....	25
Figura 7. Movimientos obtenidos por el algoritmo de movimiento.....	27
Figura 8. Dirección y valores de torque según su dirección.....	28
Figura 9. Programación de la cinemática usando Matlab.....	30
Figura 10. Ángulos de movimientos obtenidos del software R+ Motion 2.0.....	31
Figura 11. Comparación de las simulaciones realizadas en Matlab y R+ Motion 2.0.....	32
Figura 12. Gráfica posición (azul). Grafica torque (rojo).....	33
Figura 13. Ingreso de ángulos de movimiento para cada una de las extremidades del robot hexápodo Scorpion.....	34
Figura 14. Conversión y transmisión de los datos para la marcha del robot.....	34
Figura 15. Algoritmo de marcha implementado en el robot.....	35
Figura 16. Dirección CCW del servomotor para valores de torque obtenidos entre 0 a 1023.....	36
Figura 17. Dirección CW del servomotor para valores de torque obtenidos entre 1024 a 2048.....	37
Figura 18. Funciones para la lectura de torque y posición en los servomotores.....	38
Figura 19. Condicionales para la modificación en los ángulos de movimiento del robot Scorpion para evasión de obstáculos.....	38
Figura 20. Pruebas realizadas del algoritmo de control para la evasión de obstáculos.....	39
Figura 21. Servomotor alarmado por mala conexión entre las extremidades.....	40

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# 1. INTRODUCCIÓN

---

## 1.1 Generalidades

Con la necesidad de avanzar en el campo de la robótica móvil, los robots con extremidades han surgido como alternativa a los autómatas con ruedas debido al mejoramiento, versatilidad y acceso a terrenos no estructurados facilitando su exploración. Esto se debe a que al desarrollar robots con esta estructura se tiene en cuenta la capacidad de que permanezca en equilibrio cuando se encuentra una perturbación (estabilidad estática) y la elaboración de una estrategia de movimiento para lograr una coordinación en sus extremidades que permita su desplazamiento (estabilidad dinámica). Además los autómatas con extremidades pueden resolver diferentes problemas de locomoción que tienen dichos robots con ruedas en terrenos irregulares debido a que estos autómatas al tener extremidades pueden lograr movimientos coordinados y estructurados con el fin de evadir obstáculos y adaptarse a estos terrenos fácilmente.

Hasta ahora los robots con extremidades han sido controlados mediante supervisión humana, donde se controla la velocidad y dirección del autómata, mientras que un ordenador simplifica el desplazamiento de las patas con el fin de lograr el movimiento deseado, pero este modo de operación es adecuado para terrenos libres de agujeros o grandes obstáculos (Ozguner, Tsai & MGhee, 1984).

Para la solucionar la difícil navegación en los terrenos irregulares se han desarrollado técnicas de reconocimiento de imágenes, censado de variables asociadas al control cinemático y dinámico del robot, algoritmos y estrategias de navegación que permitan un buen desempeño del hexápodo a la hora de implementar una trayectoria. Un ejemplo de esto es la estrategia basada en la tolerancia a fallas donde se analiza una articulación que no se pueda mover y se encuentre bloqueada en un lugar para saber la estabilidad del hexápodo, debido a que estos autómatas tienen redundancia en el movimiento de sus articulaciones según el número de patas que este tenga, permitiendo una estabilidad dinámica y estática (Yang, 2005). Esta estrategia no reduce la estabilidad en el movimiento del autómata, pero limita el espacio de trabajo de la pierna que falla.

Otro ejemplo de algoritmos y estrategias de navegación son los de generación de marcha libre que posibilita al robot elegir los estados más estables y crear un patrón de marcha continua con una



	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

estabilidad grande (Erden & Leblebicioglu, 2008). Para mejorar la marcha libre se puede implementar en esta un aprendizaje de refuerzo donde se puede aplicar en tiempo real para una locomoción estable con diferentes velocidades. Para este tipo de estrategias de navegación se han realizado aplicaciones e investigaciones como el uso del aprendizaje por refuerzo para el desarrollo de movimientos de pro-tracción y retracción adecuados (Erden & Leblebicioglu, 2008). Kirchner lo aplicó en un robot hexápodo en tiempo real con el fin de realizar los movimientos elementales para una sola extremidad. Además se han generado estrategias de marcha libre mediante programación fuera de línea creando un enfoque difuso lo cual pretende lograr la trayectoria optima y la generación de la marcha simultáneamente, donde los controladores de lógica difusa cambian las longitudes de carrera al manejar el giro del robot y así evitar obstáculos al desplazarse (Erden & Leblebicioglu, 2008).

## **1.2 Justificación**

El avance de la robótica mundialmente nos ha permitido realizar trabajos peligrosos para las personas con ayuda de estas máquinas, entre ellas se encuentra el desminado, la búsqueda de explosivos en zonas de guerra, exploración tanto terrestre como sub acuática, e incluso viajes de reconocimiento de terreno en otros planetas. Las diversas arquitecturas que suelen ser usadas para estas aplicaciones van desde drones, pasando por humanoides y robots antropomórficos hasta plataformas que usan ruedas, sin embargo estas arquitecturas no son útiles para ciertos terrenos, que por sus características escabrosas, no nos permiten acercamientos terrestres o por características de vegetación no nos permiten un reconocimiento aéreo, es en este momento en el que se incluye el uso de plataformas que poseen extremidades para sortear estos terrenos debido a su facilidad para escalar y para caminar sobre terrenos irregulares.

El proyecto que se plantea consiste en mejorar la estrategia y el algoritmo de control actual de movimiento del robot hexápodo Scorpion, implementando opciones que permitan al robot decidir las mejores opciones que posee el encontrarse un obstáculo en su desplazamiento, ya que el actual algoritmo funciona predeterminadamente sin considerar los cambios del entorno como por ejemplo escalones, el algoritmo actual no toma en cuenta ninguno de los valores de los sensores que posee el robot con lo cual no le permite realizar estas decisiones.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### **1.3 Objetivos**

#### **1.3.1 Objetivo general**

Diseñar e implementar una estrategia de control cinemática para la evasión de obstáculos en un robot hexápodo.

#### **1.3.2 Objetivos específicos**

- Construir el modelo cinemático de la arquitectura de robot hexápodo.
- Desarrollar diferentes estrategias de control para el movimiento de un robot hexápodo.
- Implementar la estrategia de control y evaluar la eficiencia de la estrategia seleccionada.

### **1.4 Organización de la tesis**

En el capítulo 1 se hace una introducción al trabajo a realizar donde se describen unas generalidades para tener un conocimiento previo del tema a tratar y desarrollar en toda la tesis, al igual que el planteamiento del problema y la justificación de este lo cual muestra cómo se abordara y se solucionara el problema que se quiere abordar por último en este capítulo se presenta los objetivos propuestos en el informe.

En el capítulo 2 se expone la teoría y el estado del arte en que se basa esta tesis, donde en primer lugar se habla un poco de la historia de los diferentes robots con extremidades que se han desarrollado a través del paso de los años, luego se tendrá los conceptos introductorios los cuales presentan de forma resumida las características del robot como lo es el sistema mecánico, electrónico, elementos de fijación y actuadores. Finalizando este capítulo se habla de los conceptos de cinemática para los cálculos de trayectoria en los robots la cual se muestra las matrices de rotación, matrices de transformación homogéneas y como utilizar los parámetros Denavit Hartenberg para la realización de estos cálculos.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En el capítulo 3 se explica detalladamente como fue realizada la tesis, donde se presenta el paso a paso de cada una de las pruebas y simulaciones necesarias que se fueron desarrollando hasta llegar al cumplimiento de los objetivos inicialmente planteados.

En el capítulo 4 se presenta los resultados obtenidos en la metodología implementada, donde mostrara cada resultado logrado por cada una de las pruebas y simulaciones realizadas en el capítulo anterior para así dar una solución al problema y los objetivos sugeridos en el trabajo.

En el capítulo 5 se muestra las conclusiones que deberán responder a los objetivos propuestos en la tesis lo cual debe incluir las fortalezas, limitaciones y restricciones que se tuvieron mediante el desarrollo del estudio elaborado. También en este capítulo se describen las recomendaciones las cuales permiten saber las mejoras que se pueden hacer para el trabajo que fueron descubiertas en el transcurso del desarrollo del proyecto. Por último se habla de los trabajos futuros que pueden ser realizadas con el estudio ejecutado para la implementación en otras aplicaciones o mejorar el desarrollo de este.

## 2. MARCO TEÓRICO

Este proyecto busca desarrollar un algoritmo que le permita al robot Scorpion realizar evasión de obstáculos teniendo en cuenta los valores de torque obtenidos por los sensores que poseen los motores, para de esta manera lograr pasar por algunos objetos sin necesidad de rodearlos. La cual se planteara a través del mapa conceptual de la figura 1.

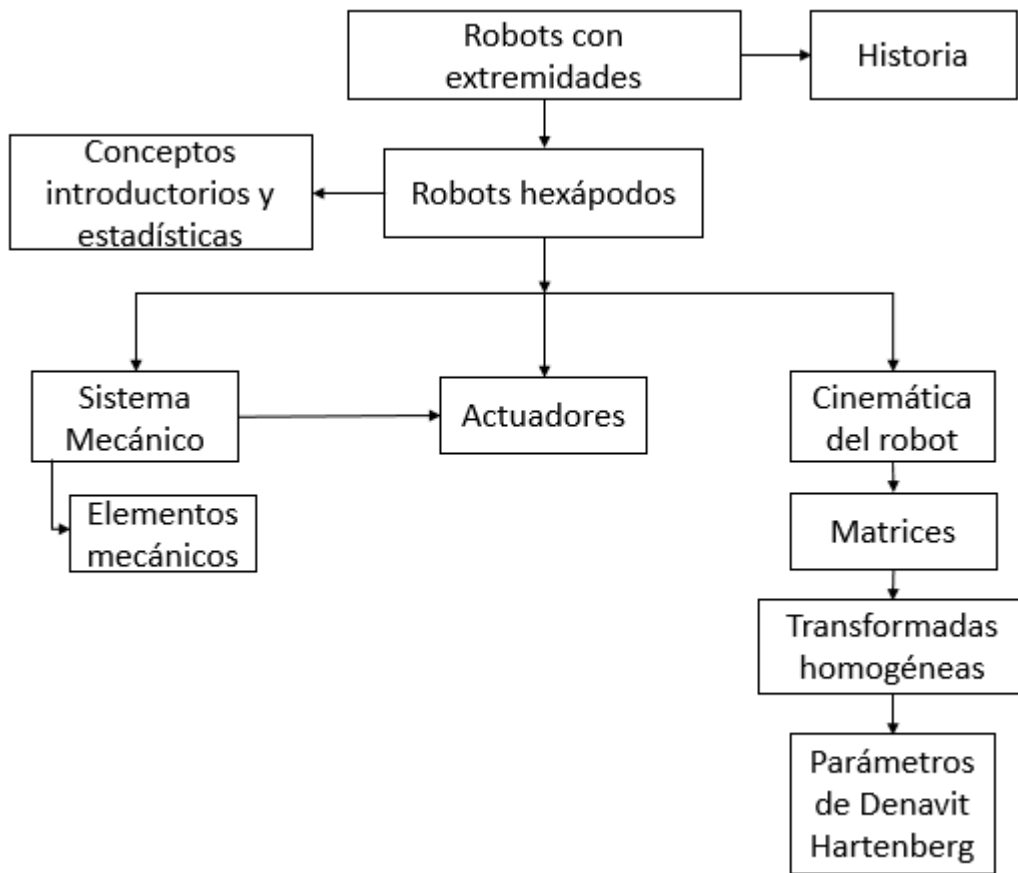


Figura 1-Diagrama del sistema robótico.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 2.1 Historia breve de los robots con extremidades

Los robots con Extremidades son alternativas viables a los robots con ruedas, debido a que muchas veces es muy complicado que dichos robots puedan acceder a algunos entornos de terrenos escabrosos. El uso de las extremidades permite evadir algunos obstáculos realizando contactos discretos sobre ellos y escalándolos en lugar de rodearlos, gracias a algunas estructuras mecánicas interconectadas que hacen las veces de patas o piernas en el esqueleto del robot; este tipo de configuraciones hacen posible que el paso por terrenos irregulares sea más fácil, sin embargo el uso de estas extremidades presenta algunas dificultades, por ejemplo a la hora de analizar su cinemática y dinámica se pueden generar algunos fenómenos de complejo modelamiento matemático, lo cual dificulta el diseño de estrategias de control efectivas que permitan un movimiento óptimo y eficiente en cualquier tipo de terreno . Los robots desarrollados para intentar imitar la anatomía de los animales son llamados, robots biomiméticos, estos usualmente sirven no solo como medio para analizar los movimientos de las diversas especies que se replican , si no como alternativas altamente eficientes en labores de investigación sobre terrenos que para la robótica convencional son difíciles de alcanzar, en el ámbito de los robots con extremidades, los modelos más comunes suelen ser insectos, aunque se han visto también imitaciones otros animales como perros o mulas, debido a que su simplicidad ósea nos suele dar pistas sobre como orientar los diseños para que estos tengan una mayor adaptabilidad. Los estudios de desarrollo de máquinas que puedan emular comportamientos o características de personas o animales se remontan a el ingeniero griego Ctesibius hacia el año 270 A.C quien aplico conocimientos de hidráulica y neumática para desarrollar un reloj con figuras móviles (Silva & Tenreiro Machado, 2007); Leonardo da Vinci (1452-1519 D.C) fue otro ingeniero que estuvo interesado en la mecánica que desarrollaron los griegos, apoyado en esto inicio estudios de anatomía y escultura adicionando el estudio de los metales con lo cual obtendría los conocimientos necesarios para construir un robot de la época, fue este quien realizo la implementación de la locomoción con patas la cual hizo a través de un caballero que movía el cuello, caminaba (un poco) y blandía su arma. Más tarde hacia el año de 1850 matemático ruso Pafnuti Chebyshev (1821-1894) desarrollo un modelo basado en vinculación cinemática para mover un cuerpo cambiando sus puntos de apoyo (M. H. Raibert, 1990).Eadweard Muybridge fue unos de los primeros científicos en documentar sus estudios sobre la locomoción de los animales incluidos los humanos pero especialmente los caballos y publico sus estudios en la Scientific American Journal de 1878 (Silva & Tenreiro Machado, 2007).

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En la década de los 60 se presentaron diversos avances en el campo de la robótica que involucraban desde el modelamiento, Joseph Edward Shingley fue uno de los que más estudio el diseño de mecanismos, la compañía Space General Corporación desarrollo 2 vehículos con extremidades con el fin de usarlos en la exploración lunar, uno con 8 extremidades y el otro con 6, pero no fue sino hasta el final de la década la compañía General Electric creo un vehículo de automoción cuadrúpedo que pretendía poner a la venta aunque la idea no prospero. Ya en la década de los 90 la electrónica logro hacer que los componentes fueran más pequeños y permitió el desarrollo de robots más pequeños y avanzados.

El siglo XXI Boston Dynamics ha estado a la vanguardia en el ámbito de la robótica, algunos de sus robots con piernas han sido unos de los más avanzados en el medio, en el año 2005 presentaron el BigDog, que es un robot de cuatro patas diseñado inicialmente para el transporte de provisiones militares y heridos, originalmente pesaba 110Kg media 90cm de largo y 75cm de alto, sin embargo en 2008 se mostró la nueva generación que pesa alrededor de 109Kg, mide 1.1m de largo y 1m de alto y puede cargar aproximadamente 154Kg de peso, las piernas que usa este robot para desplazarse tiene sensores que miden fuerza y la posición de las piernas del robot en todo momento (M. Raibert, Blankespoor, Nelson, & Playter, 2008).

El Robot RHex de la compañía Boston Dynamics es otro modelo de robot con extremidades, este es un robot de configuración hexápoda cuyas piernas son curvas, el primer modelo de este robot fue presentado en el año 1999, hasta la fecha ha sufrido varias modificaciones, el ultimo que fue presentado en 2012 tenía un peso de alrededor de 14 Kg, pero tenía la capacidad de atravesar terrenos complicados e incluso subir escaleras.

Otro robot inspirado en la naturaleza es el Atlas, este robot está pensado para ser un humanoide, de la compañía Boston Dynamics tiene un peso de 150Kg y altura de 1.8m, este fue presentado en el año 2013 y es el sucesor de un robot llamado PETMAN, que tenía características similares, Atlas está equipado con sistemas de visión artificial, este robot participa en el DARPA Robotics Challenge donde se prueban las habilidades de los robots en diversas tareas, en el año 2015 quedo en segundo puesto, detrás del robot DRC-HUBO Coreano, actualmente se encuentran desarrollando una nueva generación, se presentó en el mes de febrero del año 2016 que lo muestra operando tanto fuera en espacios interiores como en exteriores, la nueva versión tiene un peso de 82Kg y una altura de 1.75m.

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### **2.1.1 Trabajos similares**

(D. Campbell, 2003) Nos muestra un controlador para el robot hexápodo RHex, este controlador busca lograr que el robot pueda subir escaleras del mundo real dependiendo de la inclinación de las escaleras con un control de lazo abierto para dicho proceso. (Lambrinos, Möller, Labhart, Pfeifer, & Wehner., 2000) Muestra como mediante complejos algoritmos matemáticos podemos simular algunas estrategias de navegación de los insectos, como el uso de hormonas por parte de las hormigas, lo cual se puede simular con algún agente sintético. (Franz & Mallot, 2000) nos presenta el estudio del movimiento de robots biomiméticos basado en diversos métodos de navegación.

## **2.2 Conceptos Introductorios**

### **2.2.1 Robots**

Según la Organización Internacional para la estandarización (ISO por sus siglas en inglés) Un robot es un mecanismo que posee 2 o más ejes de movimiento, que puede ser reprogramado y que puede ser autónomo, esto significa que puede realizar tareas sin intervención humana. (“ISO 8373: 2012 (ES), robots y dispositivos robóticos - Vocabulario,” n.d.)

La ISO también define los tipos de robots, los cuales son:

Robot Industrial: Es un manipulador reprogramable de múltiples usos que posee 3 o más ejes.

Robot de servicio: Es un manipulador que realiza aplicaciones útiles para el ser humano sin ser aplicaciones industriales. Estos pueden ser personales o profesionales.

Robot móvil: Es capaz de moverse bajo su propio control.

### **2.2.2 Dispositivo Robótico**

Un dispositivo robótico es un mecanismo que cumple con las características de un robot industrial pero o carece de un número de ejes programables, o de autonomía.

## **2.3 Sistema mecánico**

Un sistema mecánico son un conjunto de elementos o eslabones que pueden transmitir o convertir un tipo de energía en otra, en este caso convertir energía eléctrica en mecánica, de tal manera que los

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

eslabones que están unidos entre sí mediante servomotores trabajen como un sistema robótico capaz de realizar movimiento de desplazamiento del sistema en un entorno.

### 2.3.1 Elementos de fijación

**Tornillos:** Es un elemento de fijación roscado empleado a nivel doméstico e industrial, usualmente usado para mantener unidas 2 piezas y tiene diversos tamaños y tipos de roscado los cuales se pueden ajustar mejor a las aplicaciones.

**Tuercas:** Es una pieza con un orificio, este orificio esta roscado para permitir que un tornillo se ajuste, esto sirve para crear uniones desmontables, las tuercas siempre deben tener las mismas características que los tornillos con los que se acopla, eso significa tener el mismo diámetro y el mismo tipo de roscado.

### 2.4 Actuadores eléctricos

**Servomotor:** Los servomotores o Servos como son llamados, son motores que tienen la capacidad de ubicarse en una posición dentro de su rango de operación y mantenerse estable, esto lo logra debido a que posee un sistema de control con un sensor de posición. Los servos además de tener la capacidad de ser controlados en su posición también pueden ser controlados en velocidad (Tdrobotica, 2014).

### 2.5 Cinemática del robot

La cinemática en robótica se dedica al análisis y a la solución de problemas que se derivan del posicionamiento de los elementos de un manipulador o de un sistema robótico que tenga extremidades (Angulo, 2000).

#### 2.5.1 Matrices de rotación

Las matrices de rotación se emplean para ubicar la orientación de un sistema coordinado respecto a otro, para esto, se utilizan las coordenadas del segundo sistema empleando vectores unitarios en la dirección de los ejes del primer sistema coordinado (Torres, Pomares, Gil, Puente, & Aracil, 2002).



### 2.5.2 Transformadas homogéneas

La transformación homogénea permite por medio de matrices resolver diferentes problemas gráficos (Cortes, 2012). La matriz homogénea tiene dimensiones 4 X 4 se compone de cuatro sub-Matrices:

$$T = \begin{bmatrix} Rotacional & Traslacional \\ Perspectiva & Escalado \end{bmatrix} = \begin{bmatrix} 3 & \\ 3X3 & X \\ 1 & \\ 1X3 & 1X1 \end{bmatrix} \quad (1)$$

Debido a que en robótica se trabaja con posiciones y orientación reales en el espacio tridimensional (Cortés, 2008), la perspectiva y el escalado no tienen importancia, por lo que la matriz quedara así:

$$T = \begin{bmatrix} Rotación & Traslación \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & \\ 3X3 & X \\ 1 & \\ 1X3 & 1X1 \end{bmatrix} \quad (2)$$

### 2.5.3 Estándar Denavit Hartenberg

Un robot que se compone de n+1 eslabones unidos de manera consecutiva por n articulaciones de un grado de libertad cada 1 (Cortes, 2012), este método se utiliza para describir la posición de los eslabones en el espacio gracias a determinados parámetros de forma simple.

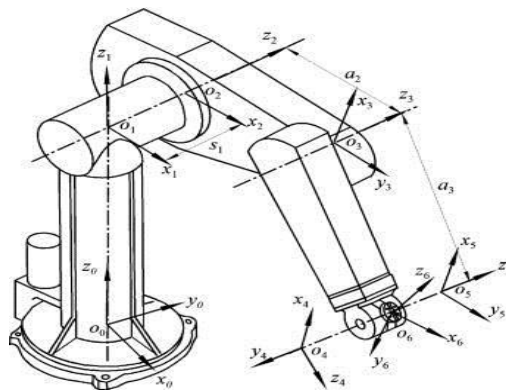


Figura 2- Denavit Hartenberg.

Tomado de:  
<http://mechanicaldesign.asmedigitalcollection.asme.org/data/Journals/JMDEDB/27745/015301j.2.jpeg>

El estándar Denavit Hartenberg es un procedimiento que permite describir cinemáticamente una cadena articulada donde cada articulación tiene solo un grado de libertad.

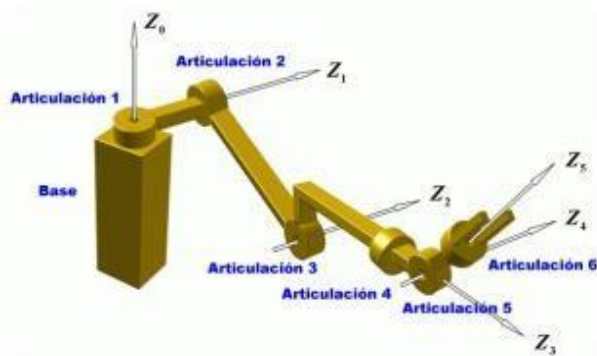


Figura 3- Coordenadas del robot.

Tomado de:

<http://mechanicaldesign.asmedigitalcollection.asme.org/data/Journals/JMDEDB/27745/015301j.2.jpeg>

(Torres, Pomares, Gil, Puente, & Aracil, 2002) Muestra cómo resolver el problema cinemático con este estándar indicando y definiendo cada variable necesaria para llegar a la matriz de transformación final.

Con esto, Logramos obtener la descripción del desplazamiento y/o rotación de cada articulación respecto a su articulación anterior.

	Tipo de articulación	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	Prismática	0	$d_1$	0	0
2	Rotacional	90	$q_2$	$l_2$	0
3	Rotacional	$q_3$	0	$l_3$	0
4	Rotacional	$q_4 + 90$	0	0	90
5	Rotacional	$q_5 - 90$	$l_5$	0	-90
6	Rotacional	$q_6$	0	0	90
7	Rotacional	$q_7$	$l_7$	0	0

Tabla 1– Parámetros Denavit Hartenberg para el robot hexápodo Corpiño.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Luego de obtener estos datos se procede a realizar la transformación homogénea de cada articulación y para concretar el modelo cinemático, se realiza la transformada homogénea total para lograr describir posición y orientación del actuador final.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

### 3. METODOLOGÍA

El control de movimiento que se encontraba implementado en el robot hexápodo Scorpion realizaba trayectorias de desplazamiento cinemático en cada una de sus extremidades siendo poco óptimo al ejecutar una estrategia de movimiento en un entorno determinado (superficie con algún tipo de obstáculos). Por esta razón se generó la idea de desarrollar una estrategia de control y movimiento en el autómata para la evasión de obstáculos; para la elaboración del algoritmo de movimiento se debe tener en cuenta el torque de cada servomotor y los grados que se debe mover cada una de sus extremidades para permitir que el hexápodo logre un óptimo movimiento en el momento de evadir un obstáculo.

Para la simulación del movimiento cinemático del robot se utilizó el software MATLAB y R+ Motion. En el toolbox de robótica de Peter Corke para MATLAB se hicieron estudios y simulaciones de la cinemática directa, cinemática inversa y generación de trayectorias para cada una de las patas del hexápodo para poder observar e identificar los parámetros y el desplazamiento realizados por la estrategia de movimiento que se desea desarrollar e implementar en el autómata. En primer lugar se hizo una primera simulación de una trayectoria simple, donde se pudiera observar el movimiento del robot hacia adelante y atrás (figura 4), además de la identificación de parámetros como ángulos y posición de movimiento de cada extremidad simulada para la modificación de la trayectoria simulada.

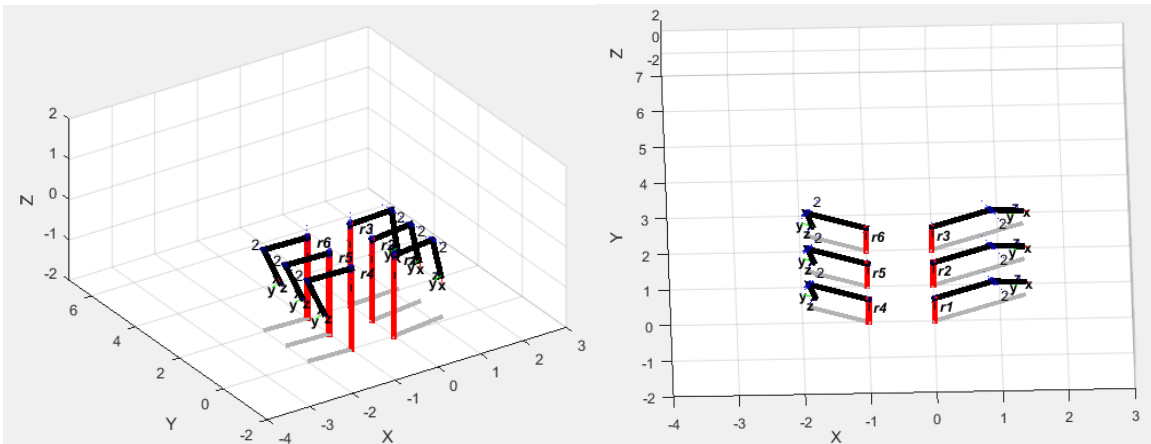


Figura 4 – Simulación movimiento hacia adelante MATLAB

En el software R+ Motion se tiene una interfaz gráfica (figura y) que facilita la simulación, observación e identificación de la cinemática y movimiento del robot hexápodo Scorpion, así mismo otorga información sobre algunos parámetros del robot como los grados que desplazaron los servomotores y la velocidad; además de poder cargar al autómata trayectorias y movimientos hechos en este software efectuando una conexión online con el robot.

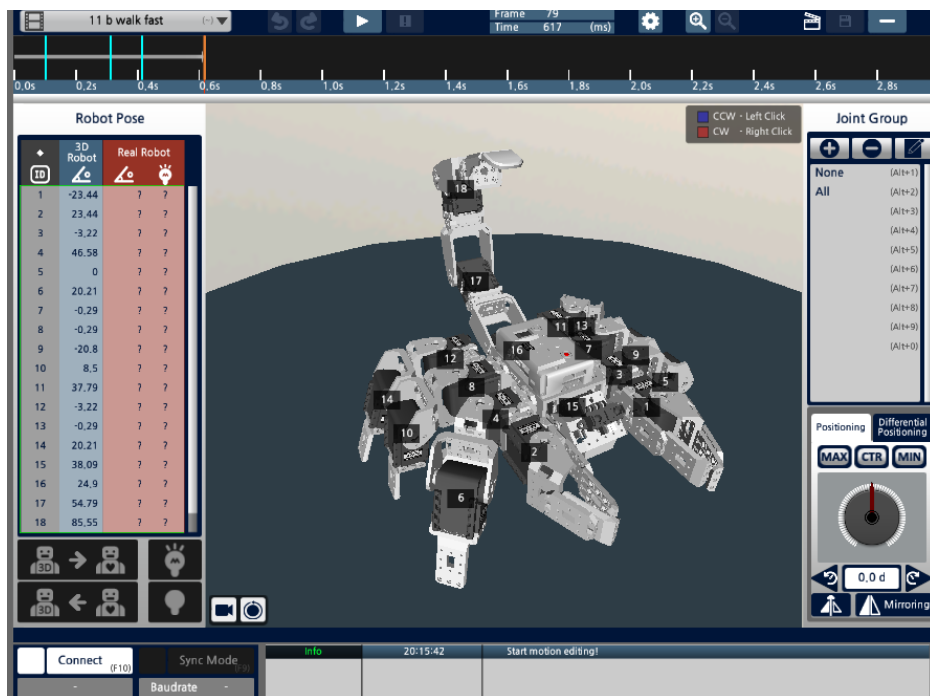


Figura 5 – Entorno gráfico y simulación de movimiento en R+ Motion

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Para obtener los datos de los parámetros hablados anteriormente se hace mediante un archivo MTNX, en el cual se puede determinar vectores de dieciocho posiciones con valores en ángulos que representan el movimiento de cada uno de los servomotores que constituyen el robot hexápodo. Los datos arrojados por R+ Motion fueron organizados en diferentes tablas dependiendo de la trayectoria simulada; estos datos fueron llevados a MATLAB para comprobar cada una de las trayectorias ejecutadas en programa R+ Motion y así tener seguridad y confiabilidad a la hora de la implementación física de la estrategia de movimiento y control que se quiere desarrollar para no causar daños al robot.

CAMINAR NORMAL ADELANTE				
	PASOS (ángulos)			
Motores	1	2	3	4
1	-23.44	-23.44	-23.44	-23.44
2	23.44	23.44	23.44	23.44
3	-3.22	-3.22	-46.88	-46.88
4	46.58	46.58	2.93	2.93
5	0	-20.51	-20.51	0
6	20.21	-0.29	-0.29	20.21
7	-0.29	-0.29	0	0
8	-0.29	-0.29	0	0
9	-20.8	-9.08	-8.79	-26.37
10	8.5	26.07	20.51	8.79
11	37.79	37.79	2.93	2.93
12	-3.22	-3.22	-38.09	-38.09
13	-0.29	-20.8	-20.51	0
14	20.21	-0.29	0	20.51
15	38.09	38.09	38.09	38.09
16	24.9	24.9	24.9	24.9
17	54.79	54.79	54.79	54.79
18	85.75	85.55	85.55	85.55

*Tabla 2– Movimiento en grados de las extremidades para marcha hacia adelante*

CAMINAR NORMAL ATRAS				
	PASOS (ángulos)			
Motores	1	2	3	4
1	-23.44	-23.44	-23.44	-23.44
2	23.44	23.44	23.44	23.44
3	-46.88	-46.88	-3.22	-3.22
4	2.93	2.93	46.58	46.58
5	0	-20.51	-20.51	0
6	20.21	-0.29	-0.29	20.21
7	0	0	-0.29	-0.29
8	0	0	-0.29	-0.29
9	-26.37	-8.79	-9.08	-20.8
10	8.79	20.51	26.07	8.5
11	2.93	2.93	37.79	37.79
12	-38.09	-38.09	-3.22	-3.22
13	0	-20.51	-20.8	-0.29
14	20.51	38.09	-0.29	20.21
15	38.09	24.9	38.09	38.09
16	24.9	54.79	24.9	24.9
17	54.79	54.79	54.79	54.79
18	85.55	85.55	85.55	85.55

*Tabla 3– Movimiento en grados de las extremidades para marcha hacia atrás*

MOVIMIENTO IZQUIERDA (Motores 3-5-7-9-11-13)					
	PASOS (ángulos)				
Motores	1	2	3	4	5
1	-23.44	-23.44	-23.44	-23.44	-23.44
2	23.44	23.44	23.44	23.44	23.44
3	-29.3	-44.24	-43.95	-43.95	0
4	0	-0.29	0	0	0
5	-20.51	-20.8	0	0	0
6	0	-0.29	0	0	0
7	0	-0.29	0	0	0
8	0	-0.29	0	0	0
9	0	-0.29	0	-20.51	-20.51
10	0	-0.29	0	0	0
11	29.3	-0.29	0	0	29.3
12	-29.3	-29.59	-29.3	-29.3	-29.3
13	-20.51	-20.8	0	0	0
14	0	-0.29	0	0	20.51
15	38.09	38.09	38.09	38.09	38.09
16	24.9	24.9	24.9	24.9	24.9
17	54.79	54.79	54.79	54.79	54.79
18	85.55	85.55	85.55	85.55	85.55

Tabla 4– Movimiento en grados de las extremidades para girar a la izquierda de los motores 3-5-7-9-11-13

MOVIMIENTO IZQUIERDA (Motores4-6-8-10-12-14)					
	PASOS(angulos)				
Motores	1	2	3	4	5
1	-23.44	-23.44	-23.44	-23.44	-23.44
2	23.44	23.44	23.44	23.44	23.44
3	0	0	0	0	0
4	0	43.95	49.95	43.95	29.3
5	0	0	0	0	0
6	0	0	0	20.51	20.51
7	0	0	0	0	0
8	0	0	0	0	0
9	-20.51	0	0	0	0
10	0	20.51	0	0	0
11	29.3	29.3	29.3	29.3	29.3
12	-29.3	0	0	0	-29.3
13	0	0	0	0	0
14	0	0	0	20.51	20.51
15	38.09	38.09	38.09	38.09	38.09
16	24.9	24.9	24.9	24.9	24.9
17	54.79	54.79	54.79	54.79	54.79
18	85.55	85.55	85.55	85.55	85.55

Tabla 5– Movimiento en grados de las extremidades para girar a la izquierda de los motores 2-4-6-8-10-12-14

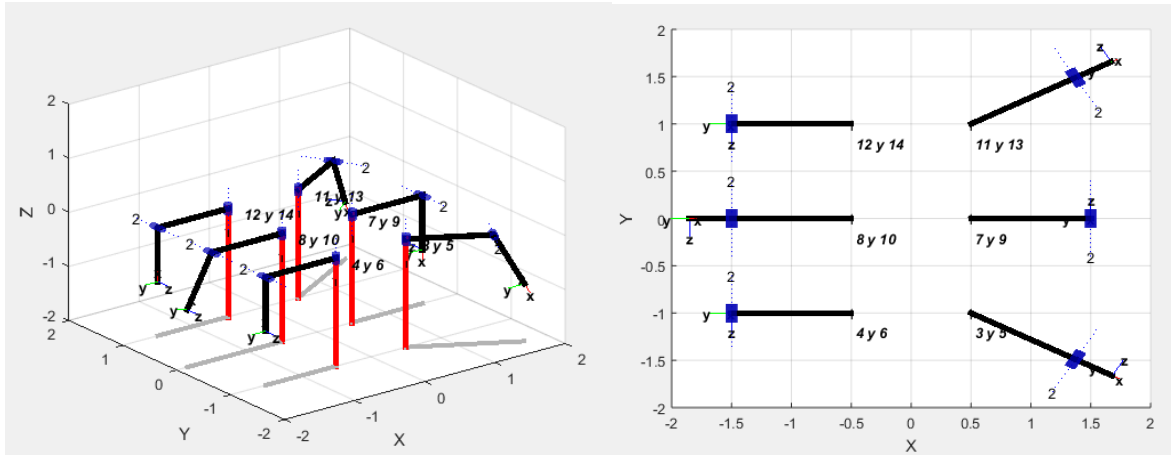


MOVIMIENTO DERECHA (Motores 3-5-7-9-11-13)					
	PASOS (ángulos)				
Motores	1	2	3	4	5
1	-23.44	-23.44	-23.44	-23.44	-23.44
2	23.44	23.44	23.44	23.44	23.44
3	0	0	0	0	0
4	29.3	43.95	43.95	43.95	0
5	0	0	0	0	0
6	20.51	20.51	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	20.51	20.51
11	29.3	29.3	29.3	29.3	29.3
12	-29.3	0	0	0	-29.3
13	0	0	0	0	0
14	20.51	20.51	0	0	0
15	38.09	38.09	38.09	38.09	38.09
16	24.9	24.9	24.9	24.9	24.9
17	54.79	54.79	54.79	54.79	54.79
18	85.55	85.55	85.55	85.55	85.55

Tabla 6– Movimiento en grados de las extremidades para girar a la derecha de los motores 3-5-7-9-11-13

MOVIMIENTO DERECHA (Motores4-6-8-10-12-14)					
	PASOS (ángulos)				
Motores	1	2	3	4	5
1	-23.44	-23.44	-23.44	-23.44	-23.44
2	23.44	23.44	23.44	23.44	23.44
3	0	-43.95	-43.95	-43.95	-29.3
4	0	0	0	0	0
5	0	0	0	-20.51	-20.51
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	-20.51	-20.51	0	0	0
10	0	0	0	0	0
11	29.3	0	0	0	29.3
12	0	-29.3	-29.3	-29.3	-29.3
13	0	0	0	0	-20.51
14	20.51	0	0	0	0
15	38.09	38.09	38.09	38.09	38.09
16	24.9	24.9	24.9	24.9	24.9
17	54.79	54.79	54.79	54.79	54.79
18	85.55	85.55	85.55	85.55	85.55

Tabla 7– Movimiento en grados de las extremidades para girar a la derecha de los motores 2-4-6-8-10-12-14



*Figura 6 – Simulación de los datos sacados del software R+ Motion mostradas en las tablas anteriores en MATLAB*

Par el desarrollo del algoritmo de movimiento se utilizó la herramienta OpenCM9.04, el cual es una tarjeta de desarrollo la cual su lenguaje de programación está basada en C; para la realización de la estrategia de control se debe tener en cuenta la posición máxima y mínima en ángulos que pueden alcanzar los servomotores AX-12A del autómeta (0 a 300 grados), de igual forma saber que el convertidor análogo digital (ADC) que contiene la OpenCM es de 10 bits, la cual posibilitará mediante la ecuación z relacionar y elaborar una equivalencia entre los grados necesarios para el movimiento de las extremidades con un número binario entre los valores de 0 a 1024 que es lo que la tarjeta de desarrollo procesa y así transmitir y recibir los datos entregados por cada uno de los servomotores.

$$conversión = \frac{(grados) \times (1024)}{300} \quad (3)$$

En la siguiente tabla se podrá observar la relación entre los grados con el valor de 0 a 1024 que le corresponde.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

GRADOS	CONVERSIÓN ADC (0-1024)
0	0
20	68
45	153
60	205
90	307
110	375
135	460
150	512
180	614
200	682
240	819
270	921
300	1024

*Tabla 8– Valores de grados con su respectivo equivalente entre valores de 0 a 1024*

A la ecuación 3 se tuvo que modificar añadiéndole la corrección del offset ya que la posición cero del robot hexápodo al momento de realizar un movimiento con la OpenCM no coincidía con lo simulado en MATLAB y en R+ Motion, reformando lo anterior la nueva ecuación obtenida es:

$$conversión = \frac{(grados+150)x(1024)}{300} \quad (4)$$

Antes de elaborar el algoritmo que se implementara en el robot hexápodo se hicieron pruebas con un solo servomotor, donde se mandaron datos de movimiento a este para que se desplazara los grados transmitidos y verificar que la ecuación 4 si funciona

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

correctamente, además de observar que al ingresar los valores de la tabla x4 se ejecutan los movimientos correctamente y estos si coinciden con lo simulado en los diferentes software para así tener un estándar en los datos tanto en la parte simulada como en la parte física.

Luego se realizó un algoritmo de movimiento el cual permitiría observar las diferentes marchas que puede realizar el robot según las tablas de movimiento obtenidas por las simulaciones, además de verificar que en cada uno de los pasos programados al robot hexápodo para que este realice un tipo de marcha y que este coincida con los valores de las tablas (figura 7), estos valores son mostrados mediante la consola virtual del software ROBOTIS\_OpenCM para ser comparados con los datos arrojados por las simulaciones realizadas.

```

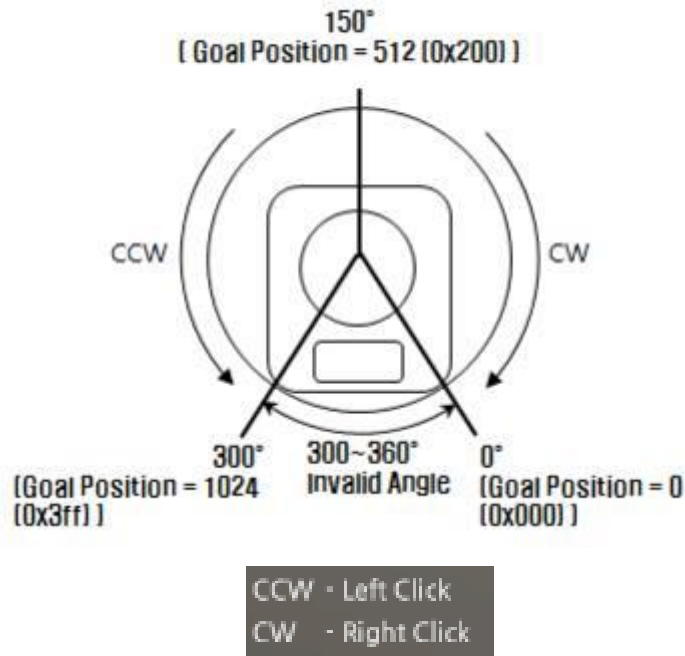
WFM3[] = { -3.22, -3.22, -46.88, -46.88 };
WFM4[] = { 46.58, 46.58, 2.93, 2.93 };
WFM5[] = { 0, -20.51, -20.51, 0 };
WFM6[] = { 20.21, 0, 0, 20.21 };
WFM7[] = { -0.29, -0.29, -0.29, -0.29 };
WFM8[] = { -0.29, -0.29, -0.29, -0.29 };
WFM9[] = { -20.80, -9.08, -8.79, -26.37 };
WFM10[] = { 8.50, 26.07, 20.51, 8.79 };
WFM11[] = { 37.79, 37.79, 2.93, 2.93 };
WFM12[] = { -3.22, -3.22, -38.09, -38.09 };
WFM13[] = { -0.29, -20.8, -20.51, 0 };
WFM14[] = { 20.21, -0.29, 0, 20.51 };

```

*Figura 7 –Movimientos obtenidos por el algoritmo de movimiento*

Después se elaboró pruebas nuevamente con un solo servomotor, esta vez para establecer la medición de torque y así poder identificar como esta variable es entregada por cada uno de los servomotores que componen el autómata. La variable sensada de torque por los servomotores son dadas en valores de 0 a 1024 o de 1025 a 2048, estos valores depende de la dirección hacia donde es ejercida la fuerza al motor para la generación de torque en este (figura 8).

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



*Figura 8– Dirección y valores de torque según su dirección*

Al terminar las pruebas anteriormente realizadas, se procederá con el desarrollo del algoritmo de movimiento necesario para la estrategia de control para la evasión de obstáculos que se implementara en el robot hexápodo Scorpion. Para esto se lee la variable torque de los servomotores 3 y 4; cuando el autómatas se encuentre con un obstáculo genera un aumento en los valores entregados de toque en estos motores y así conseguir evaluar estos valores mediante un condicional que permita por medio de un rango saber si es un obstáculo con lo que el hexápodo se encontró en su marcha normal, por ende el autómatas levantara más cada una de sus extremidades aumentando los ángulos que tiene que desplazarse cada servomotor mostradas en las tablas de marcha (tabla 2) hasta que permita al robot Scorpion superar el obstáculo encontrado. La ecuación que proporciona el aumento de los grados en cada extremidad depende de donde es ejercida la fuerza para la generación de torque, estas son:

$$conversión = \frac{(grados+180)x(1024)}{300} \quad (5)$$

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

$$conversion = \frac{(grados+120)x(1024)}{300} \quad (6)$$

La ecuación 5 es para la dirección CW del servomotor (valores de torque entre 1024 a 2047) y la ecuación 6 para la dirección CCW del servomotor (valores de torque entre 0 a 1023).

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 4. RESULTADOS Y DISCUSIÓN

### 4.1 Simulación cinemática y de movimiento del robot hexápodo

Se dispuso del software Matlab para realizar y verificar el modelo cinemático del robot Scorpion en cada una de sus extremidades utilizando el toolbox de robótica de Peter Coorke, para esto se empleó las siguientes funciones:

```

%posicion inicial del robot (X,Y,Z)
R1.base = transl(0,0,0);
R2.base = transl(0,1,0);
R3.base = transl(0,2,0);
R4.base = transl(-1,0,0);
R5.base = transl(-1,1,0);
R6.base = transl(-1,2,0);
%-----
np = 10;
[Tr1R] = lin4([.8,0.4,-.5],[0.8,-0.4,-1],np);
[Tr2R] = lin4([.8,-0.4,-1],[0.8,0.4,-1.5],np);
[Tr3R] = lin4([.8,0.4,-1.5],[0.8,0.4,-1.5],np);
|
%-----
[q11, q12] = Cinem_inv(R1,Tr1R(:,1),Tr1R(:,2),Tr1R(:,3));
[q21, q22] = Cinem_inv(R1,Tr2R(:,1),Tr2R(:,2),Tr2R(:,3));
[q31, q32] = Cinem_inv(R1,Tr3R(:,1),Tr3R(:,2),Tr3R(:,3));

```

*Figura 9– programación de cinemática usando Matlab*

Al ejecutar la función de la figura9 calculara la cinemática inversa de cada una de las patas del robot desde el punto inicial hasta el punto final al que quiere ser llevado el robot para luego seguir con el análisis de movimiento que debe realizar el hexápodo para realizar una marcha. Para esto se apoyó en el software R+ Motion 2.0 debido a que este permite una simulación en tiempo real de los movimientos del autómeta, además de tener diferentes marchas que puede elaborar el robot y así tener información de los grados que se debe mover cada uno de los servomotores para realizar movimientos para el desplazamiento del autómeta.

Paso 1		Paso 2		Paso 3		Paso 4	
ID	3D Robot	ID	3D Robot	ID	3D Robot	ID	3D Robot
1	-23.44	1	-23.44	1	-23.44	1	-23.44
2	23.44	2	23.44	2	23.44	2	23.44
3	-46.88	3	-46.88	3	-3.22	3	-3.22
4	2.93	4	2.93	4	46.58	4	46.58
5	0	5	-20.51	5	-20.51	5	0
6	20.21	6	-0.29	6	-0.29	6	20.21
7	0	7	0	7	-0.29	7	-0.29
8	0	8	0	8	-0.29	8	-0.29
9	-26.37	9	-8.79	9	-9.08	9	-20.8
10	8.79	10	20.51	10	26.07	10	8.5
11	2.93	11	2.93	11	37.79	11	37.79
12	-38.09	12	-38.09	12	-3.22	12	-3.22
13	0	13	-20.51	13	-20.8	13	-0.29
14	20.51	14	0	14	-0.29	14	20.21
15	38.09	15	38.09	15	38.09	15	38.09
16	24.9	16	24.9	16	24.9	16	24.9
17	54.79	17	54.79	17	54.79	17	54.79
18	85.55	18	85.55	18	85.55	18	85.55

Figura 10– Ángulos de movimientos obtenidos del software R+ Motion 2.0

Con los ángulos obtenidos para cada una de las marchas como se muestra en la figura 10, se hicieron unas tablas mostradas anteriormente (Tabla 2, tabla 3, tabla 4, tabla 5) y así ingresar esto a Matlab y obtener el movimiento que el robot debería realizar mediante los cálculos cinemáticos que se ejecutan en el programa hecho en Matlab para poder ser comparados con los resultados adquiridos en R+ Motion, logrando que estos dos sean iguales para tener un estándar en las simulaciones en cada uno de los software como se muestra en la figura 11.



```
qr1i = [0 3.14/2];
qr1f = [-0.5114 1.21];
qr2i = [0 3.14/2];
qr2f = [0 3.14/2];
qr3i = [0 3.14/2];
qr3f = [0.511 1.22];
qr4i = [0 3.14/2];
qr4f = [0 3.14/2];
qr5i = [0 3.14/2];
qr5f = [0 3.14/2-0.3579];
qr6i = [0 3.14/2];
qr6f = [0 3.14/2];
```

```
-----
t = [0:.05:2];
qr1 = jtraj(qr1i, qr1f, t);
qr2 = jtraj(qr2i, qr2f, t);
qr3 = jtraj(qr3i, qr3f, t);
qr4 = jtraj(qr4i, qr4f, t);
qr5 = jtraj(qr5i, qr5f, t);
qr6 = jtraj(qr6i, qr6f, t);
```

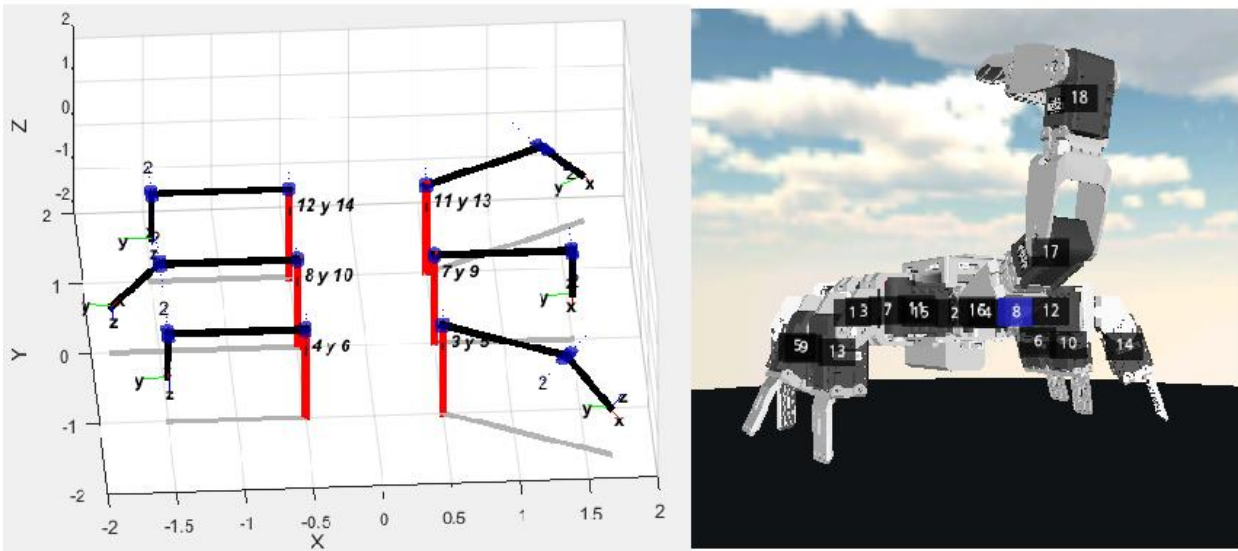
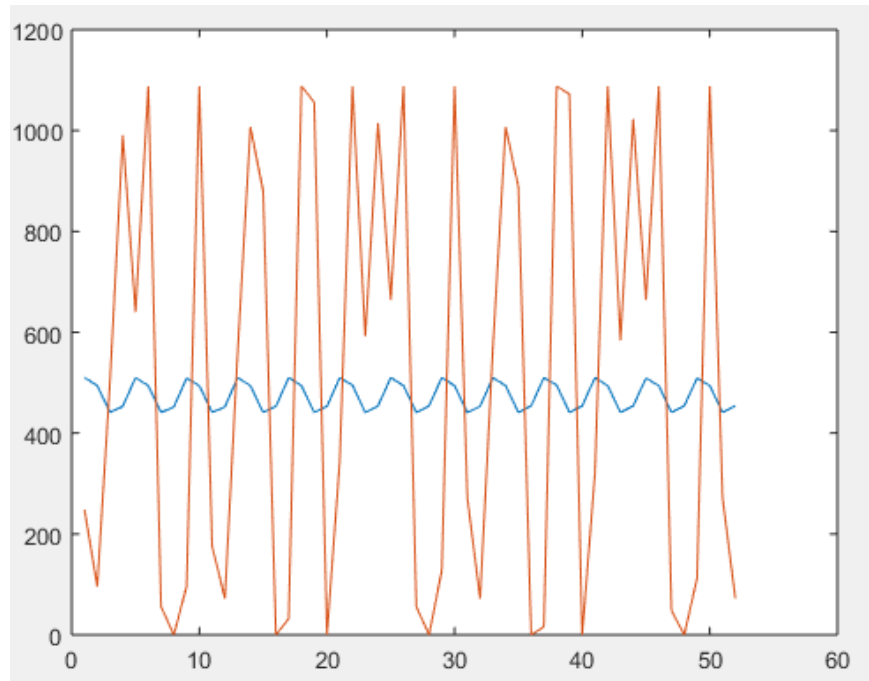


Figura 11– Comparación de las simulaciones realizadas en Matlab y R+ Motion 2.0

Por último se hicieron pruebas físicas con el robot para obtener y analizar una base de datos de posición y toque y así graficarlas en Matlab (Figura 12) para observar si estos datos eran constantes cada vez que el hexápodo realizaba un ciclo de su marcha, donde se pudo percibir que la posición

y el torque en los servomotores mantienen un patrón definido permitiendo saber que el algoritmo de control a desarrollar cumplirá con su función.



*Figura 12– Gráfica posición (azul). Grafica torque (rojo)*

#### **4.2 Desarrollo del algoritmo para la estrategia de control para la evasión de obstáculos.**

Antes de realizar el algoritmo para la evasión de obstáculos, se hizo una programación para el movimiento del robot según los datos obtenidos en las simulaciones elaboradas y así comprobar de forma física que la marcha ejecutada por el autómata si sea la que se esperaba según las simulaciones. Para esto se utilizó la tarjeta de desarrollo OpenCM9.04 y el software ROBOTIS OpenCM donde se ingresó los ángulos de marcha obtenidos (Figura13) para cada una de las extremidades del hexápodo y que realizara estos movimientos continuamente mediante las funciones de conversión de ángulos de la figura 14.

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

double WFM3[] = { -3.22, -3.22, -46.88, -46.88};
double WFM4[] = { 46.58, 46.58, 2.93, 2.93};
double WFM5[] = { 0, -20.51, -20.51, 0};
double WFM6[] = { 20.21, 0, 0, 20.21 };
double WFM7[] = { -0.29, -0.29, -0.29, -0.29};
double WFM8[] = { -0.29, -0.29, -0.29, -0.29};
double WFM9[] = { -20.80, -9.08, -8.79, -26.37};
double WFM10[] = { 8.50, 26.07, 20.51, 8.79};
double WFM11[] = { 37.79, 37.79, 2.93, 2.93};
double WFM12[] = { -3.22, -3.22, -38.09, -38.09};
double WFM13[] = { -0.29, -20.8, -20.51, 0};
double WFM14[] = { 20.21, -0.29, 0, 20.51};

```

Figura 13– Ingreso de ángulos de movimiento para cada una de las extremidades del robot

*hexápodo Scorpion*

```

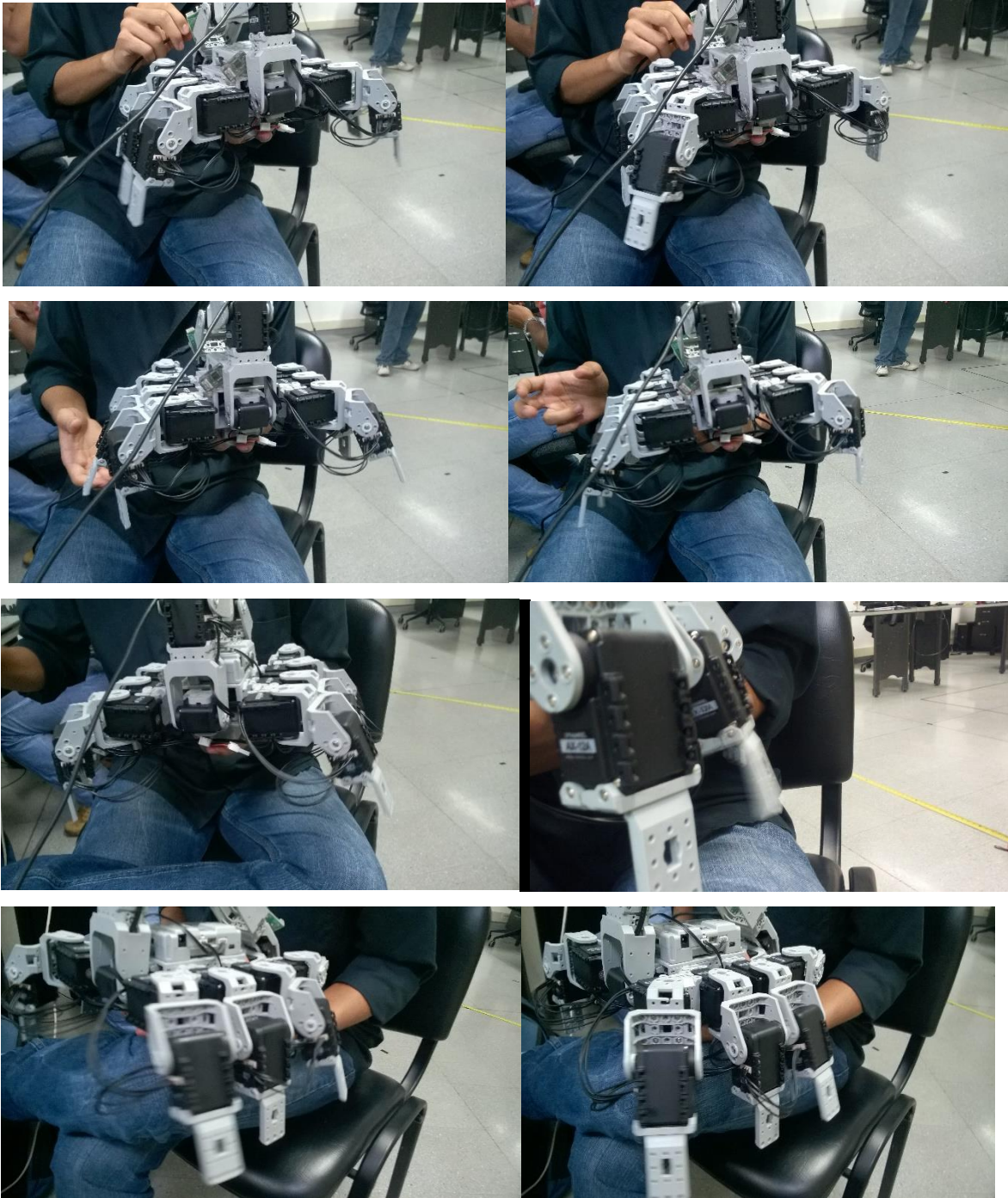
//CONVERTIR A BINARIO
WFM3c= (((WFM3[i]+150)*1024)/300);
WFM4c= (((WFM4[i]+150)*1024)/300);
WFM5c= (((WFM5[i]+150)*1024)/300);
WFM6c= (((WFM6[i]+150)*1024)/300);
WFM7c= (((WFM7[i]+150)*1024)/300);
WFM8c= (((WFM8[i]+150)*1024)/300);
WFM9c= (((WFM9[i]+150)*1024)/300);
WFM10c= (((WFM10[i]+150)*1024)/300);
WFM11c= (((WFM11[i]+150)*1024)/300);
WFM12c= (((WFM12[i]+150)*1024)/300);
WFM13c= (((WFM13[i]+150)*1024)/300);
WFM14c= (((WFM14[i]+150)*1024)/300);

//ESCRIBIR EN LAS PATAS DEL ROBOT
Dx1.writeWord(ID_NUM3, 30, WFM3c);
Dx1.writeWord(ID_NUM4, 30, WFM4c);
Dx1.writeWord(ID_NUM5, 30, WFM5c);
Dx1.writeWord(ID_NUM6, 30, WFM6c);
Dx1.writeWord(ID_NUM7, 30, WFM7c);
Dx1.writeWord(ID_NUM8, 30, WFM8c);
Dx1.writeWord(ID_NUM9, 30, WFM9c);
Dx1.writeWord(ID_NUM10, 30, WFM10c);
Dx1.writeWord(ID_NUM11, 30, WFM11c);
Dx1.writeWord(ID_NUM12, 30, WFM12c);
Dx1.writeWord(ID_NUM13, 30, WFM13c);
Dx1.writeWord(ID_NUM14, 30, WFM14c);

```

Figura 14– Conversión y transmisión de los datos para la marcha del robot

Luego de cargar el programa al robot se evidencio que los movimientos realizados por este coincidían con lo obtenido en las simulaciones como se evidencia en la figura 15.



*Figura 15–Algoritmo de marcha implementado en el robot*

Para el algoritmo de evasión de obstáculos se debe tener leer la variable torque que se puede obtener de los servomotores para así establecer condiciones para que el autómatas realice los movimientos necesarios para que pueda evadirlos. Para esto se debe tener en cuenta hacia donde es ejercida la fuerza en los servomotores de las patas del autómatas, ya que el aumento de la fuerza puede aumentar el torque en cada una de las extremidades y así el hexápodo poder saber si se ha encontrado en su trayectoria un obstáculo y así evadirlo. Para identificar la lectura del torque en los servomotores que componen el robot Scorpion se cogió un servomotor y fue sometido a diferentes fuerzas para saber los valores de torque entregados por estos servomotores que estos valores dependen de la dirección de donde es ejercida la fuerza (ver figura 8). Las pruebas realizadas se muestran a continuación en las figuras 16 y 17.



COM7	COM7	COM7
torq = 0	torq = 352	torq = 512
torq = 0	torq = 352	torq = 512
torq = 0	torq = 352	torq = 512
torq = 0	torq = 352	torq = 512
torq = 0	torq = 352	torq = 512
torq = 0	torq = 352	torq = 512

Figura 16- Dirección CCW del servomotor para valores de torque obtenidos entre 0 a 1023.



COM7	COM7	COM7
torq = 0	torq = 1472	torq = 1576
torq = 0	torq = 1472	torq = 1576
torq = 0	torq = 1472	torq = 1568
torq = 0	torq = 1472	torq = 1568
torq = 0	torq = 1472	torq = 1568
torq = 0	torq = 1472	torq = 1568

Figura 17- Dirección CW del servomotor para valores de torque obtenidos entre 1024 a 2048.

Luego de tener establecido los valores de torque posibles que puede detectar las extremidades del robot hexápodo al encontrarse con un obstáculo se persiguió al desarrollo y programación del algoritmo para que el autómatas logre guardar los valores de torque obtenidos como se muestra en la figura 18 y con estos evadir los obstáculos que se interpongan en su marcha normal; para esto se hicieron condicionales que permitan saber si el autómatas se encontró un elemento que no permita la movilidad del robot para así el robot pueda tomar la decisión de realizar un movimiento que permita evadirlo o esquivarlo modificando los ángulos de movimiento de cada una de sus extremidades (figura 19).

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

WFM5c=WFM5c1=Dx1.getPosition(5);
torq5=Dx1.getLoad(5);
WFM6c1=Dx1.getPosition(6);
WFM6c=Dx1.getPosition(6);
torq6=Dx1.getLoad(6);
WFM9c=WFM9c1=Dx1.getPosition(9);
torq9=Dx1.getLoad(9);
WFM10c=Dx1.getPosition(10);
WFM10c1=Dx1.getPosition(10);
torq10=Dx1.getLoad(10);
WFM13c=Dx1.getPosition(13);
torq13=WFM13c1=Dx1.getLoad(13);
WFM14c=Dx1.getPosition(14);
torq14=Dx1.getLoad(14);

```

*Figura 18-Funciones para la lectura de torque y posición en los servomotores*

```

if(torq4 > 1750)
WFM6c=((WFM6[i]+180)*1024)/300);
if(WFM6c1 > 610)
WFM10c=((WFM10[i]+180)*1024)/300);
if(WFM10c1 > 610)
WFM14c=((WFM14[i]+180)*1024)/300);
if(torq3 > 1750)
WFM5c=((WFM5[i]+120)*1024)/300);
if(WFM5c1 > 610)
WFM9c=((WFM9[i]+120)*1024)/300);
if(WFM9c1 > 610)
WFM13c=((WFM13[i]+120)*1024)/300);

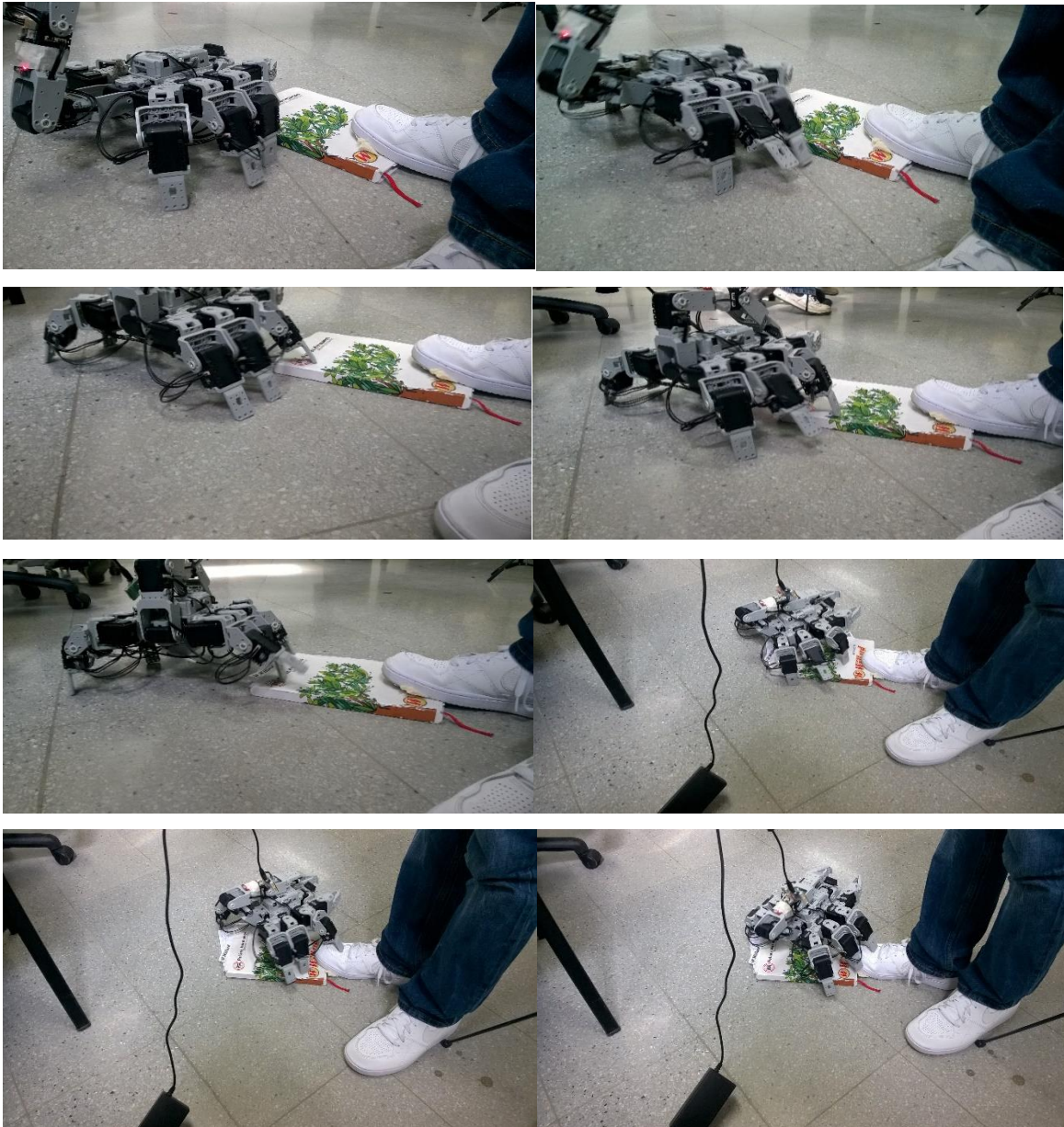
Dx1.writeWord(ID_NUM3, 30, WFM3c);
Dx1.writeWord(ID_NUM4, 30, WFM4c);
Dx1.writeWord(ID_NUM5, 30, WFM5c);
Dx1.writeWord(ID_NUM6, 30, WFM6c);
Dx1.writeWord(7, 30, 512);
Dx1.writeWord(8, 30, 512);
Dx1.writeWord(ID_NUM9, 30, WFM9c);
Dx1.writeWord(ID_NUM10, 30, WFM10c);
Dx1.writeWord(ID_NUM11, 30, WFM11c);
Dx1.writeWord(ID_NUM12, 30, WFM12c);
Dx1.writeWord(ID_NUM13, 30, WFM13c);
Dx1.writeWord(ID_NUM14, 30, WFM14c);

```

*Figura 19-Condicionales para la modificación en los ángulos de movimiento del robot Scorpion para evasión de obstáculos*

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Por ultimo este algoritmo fue implementado en el robot hexápodo para verificar la funcionalidad de este como se muestra en la figura 20 y saber si cumple con el objetivo de evadir algún elemento que intervenga en su marcha. En las siguientes imágenes se observa las pruebas realizadas que se hicieron con el autómata.



*Figura 20-Pruebas realizadas del algoritmo de control para la evasión de obstáculos*



	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Hay que tener en cuenta que en el momento de realizar las pruebas físicas en el robot hexápodo Scorpion se debe verificar que ninguno de los servomotores que lo componen este alarmado debido a la mala conexión entre los cables que conectan cada extremidad, mala programación o falta de transmisión y recepción de datos entre el controlador y los servomotores; ya que esto causa que alguna extremidad deje de funcionar dificultando así que el algoritmo desarrollado e implementado no funcione eficientemente, esto se detecta fácilmente ya que los servomotores encienden un led de color rojo cuando uno de estos errores aparece como se muestra en la figura 21 .



*Figura 21-Servomotor alarmado por mala conexión entre las extremidades*

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## 5. CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

---

### 5.1 Conclusiones

- Se desarrolló el algoritmo conforme se había especificado, realizando las pruebas necesarias para garantizar su correcto funcionamiento y presentando tanto sus virtudes como sus limitaciones.
- Gracias al desarrollo del modelo cinemático de la arquitectura se redujo la dificultad al entender el movimiento específico del robot y facilitó la creación del algoritmo.
- El robot puede ser usado en aplicaciones de reconocimiento de terrenos gracias al algoritmo de control creado, que también puede ser intervenido y entendido fácilmente debido a su lenguaje de programación.
- En el trabajo futuro se esclarece la proyección del trabajo realizado, en el sentido de otras aplicaciones, mayores desarrollos, otras consideraciones, etc. Es dejar en evidencia otras oportunidades en el área estudiada.
- Durante todo el transcurso del proyecto hubo inconvenientes con la comunicación física de los motores debido a algunos problemas de hardware con los cables.

### 5.2 Recomendaciones y trabajo futuro

- Como trabajo futuro se puede mejorar el algoritmo de control mediante la inclusión de cámaras y sensores de proximidad para hacer el movimiento más fluido y más preciso.
- El uso de un controlador de más capacidad de procesamiento puede permitir una mejora notable en los tiempos de reacción del robot depurando mayor cantidad de instrucciones del algoritmo en menor tiempo.
- Se puede integrar el uso de las pinzas y de la cola del robot que no fueron usadas en este caso como apoyo para quitar algunos obstáculos o ser de apoyo a la hora de sortear otros.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

## REFERENCIAS

---

- Angulo, J. M. (2000). *Robótica Practica*. Paraninfo
- Cortes, F. R. (2012). *Robotica, Control de robots manipuladores*. Mexico: Alfaomega.
- Cortés, J. (2008). *La representación Denavit-Hartenberg*. Obtenido de [http://personal.us.es/jcortes/Material/Material\\_archivos/Articulos%20PDF/RepresentDH.pdf](http://personal.us.es/jcortes/Material/Material_archivos/Articulos%20PDF/RepresentDH.pdf)
- D. Campbell. (2003). Stair Descent in the Simple Hexapod &#8217;RHex&#8217;. *In IEEE Int. Conf. on Robotics and Automation*, 1380–1385.
- Erden, M.S. and Leblebicioğlu, K., 2008. “Free gait generation with reinforcement learning for a six-legged robot.” *Robotics and Autonomous Systems*, 56: 199-212, DOI:10.1016/j.robot.2007.08.001.
- Franz, M. O., & Mallot, H. A. (2000). Biomimetic robot navigation. *Robotics and autonomous Systems.*, 30, 133–153.
- ISO 8373: 2012 (ES), robots y dispositivos robóticos - Vocabulario. (n.d.). Retrieved June 1, 2016, from <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>
- Lambrinos, D., Möller, R., Labhart, T., Pfeifer, R., & Wehner., R. (2000). A Mobile Robot Employing Strategies For Navigation. *Robotics and Autonomous Systems*, 30(February), 39–64.
- Raibert, M., Blankespoor, K., Nelson, G., & Playter, R. (2008). BigDog , the Rough-Terrain Quaduped Robot. *Control*, 1–5.
- Raibert, M. H. (1990). Legged robots. *Artificial Intelligence at MIT: Expanding Frontiers*, Vol.2(6), 499–514.
- Silva, M. F., & Tenreiro Machado, J. a. (2007). A Historical Perspective of Legged Robots. *Journal of Vibration and Control*, 13(9-10), 1447–1486. <http://doi.org/10.1177/1077546307078276>
- Tdrobotica. (2014). *Tdrobotica.co*. Obtenido de <http://tienda.tdrobotica.co/categoria/81>

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Torres, F., Pomares, J., Gil, P., Puente, S. T., & Aracil, R. (2002). *Automatica y Robotica - Robots y sistemas sensoriales*. Madrid: Prentice Hall.

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

# APÉNDICE

---

## APÉNDICE A

```

clc;
clear all;
close all;
%-----<
%Links de las patas de la izquierda
a1= Link ([0,0,1,-pi/2,0]);
a2= Link ([0,0,1,0,0]);
%Links de las patas de la derecha
a3= Link ([0,0,-1,pi/2,0]);
a4= Link ([0,0,-1,0,0]);
%-----
%nombres de las extremidades
R1=SerialLink ([a1 a2], 'name', 'r1');
R2=SerialLink ([a1 a2], 'name', 'r2');
R3=SerialLink ([a1 a2], 'name', 'r3');
R4=SerialLink ([a3 a4], 'name', 'r4');
R5=SerialLink ([a3 a4], 'name', 'r5');
R6=SerialLink ([a3 a4], 'name', 'r6');
%-----
%posicion inicial del robot (X,Y,Z)
R1.base = transl(0,0,0);
R2.base = transl(0,1,0);
R3.base = transl(0,2,0);
R4.base = transl(-1,0,0);
R5.base = transl(-1,1,0);
R6.base = transl(-1,2,0);
%-----
np = 10;
[Tr1R] = lin4([.8,0.4,-.5],[0.8,-0.4,-1],np);
[Tr2R] = lin4([.8,-0.4,-1],[0.8,0.4,-1.5],np);
[Tr3R] = lin4([.8,0.4,-1.5],[0.8,0.4,-1.5],np);
%-----
[q11, q12] = Cinem_inv(R1,Tr1R(:,1),Tr1R(:,2),Tr1R(:,3));
[q21, q22] = Cinem_inv(R1,Tr2R(:,1),Tr2R(:,2),Tr2R(:,3));
[q31, q32] = Cinem_inv(R1,Tr3R(:,1),Tr3R(:,2),Tr3R(:,3));
%-----
R1.plot([0 0]);
axis([-4 3 -2 7 -2 2]);
hold on;
R2.plot([0 pi/2]);
R3.plot([0 pi/2]);
R4.plot([0 pi/2]);

```

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

R5.plot([0 pi/2]);
R6.plot([0 pi/2]);
%-----
for i=1:np
    R1.plot([q11(i) q12(i)])
    R2.plot([q11(i) q12(i)])
    R3.plot([q11(i) q12(i)])
    R4.plot([q21(i) q22(i)])
    R5.plot([q21(i) q22(i)])
    R6.plot([q21(i) q22(i)])
end
%-----
for i=1:np
    R1.plot([q21(i) q22(i)])
    R2.plot([q21(i) q22(i)])
    R3.plot([q21(i) q22(i)])
    R4.plot([q11(i) q12(i)])
    R5.plot([q11(i) q12(i)])
    R6.plot([q11(i) q12(i)])
end

```

## APÉNDICE B

```

clc;

%-----<
%Links de las patas de la izquierda
a1= Link ([0,0,1,-pi/2,0]);
a2= Link ([0,0,1,0,0]);
%Links de las patas de la derecha
a3= Link ([0,0,-1,pi/2,0]);
a4= Link ([0,0,-1,0,0]);
%-----
R1=SerialLink ([a1 a2], 'name', '3 y 5');
R2=SerialLink ([a1 a2], 'name', '7 y 9');
R3=SerialLink ([a1 a2], 'name', '11 y 13');
R4=SerialLink ([a3 a4], 'name', '4 y 6');
R5=SerialLink ([a3 a4], 'name', '8 y 10');
R6=SerialLink ([a3 a4], 'name', '12 y 14');
%-----
R1.base = transl(.5,-1,0);
R2.base = transl(.5,0,0);
R3.base = transl(.5,1,0);
R4.base = transl(-0.5,-1,0);
R5.base = transl(-0.5,0,0);
R6.base = transl(-0.5,1,0);
%----- Resta cuando sale
%----- Suma cuando la pata se dirige al
%robot
qr1i = [0 3.14/2];

```

 Institución Universitaria	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```
qr1f = [-0.5114 1.21];
qr2i = [0 3.14/2];
qr2f = [0 3.14/2];
qr3i = [0 3.14/2];
qr3f = [0.511 1.22];
qr4i = [0 3.14/2];
qr4f = [0 3.14/2];
qr5i = [0 3.14/2];
qr5f = [0 3.14/2-0.3579];
qr6i = [0 3.14/2];
qr6f = [0 3.14/2];
```

%-----

```
t = [0:.05:2];
qr1 = jtraj(qr1i, qr1f, t)
qr2 = jtraj(qr2i, qr2f, t)
qr3 = jtraj(qr3i, qr3f, t)
qr4 = jtraj(qr4i, qr4f, t)
qr5 = jtraj(qr5i, qr5f, t)
qr6 = jtraj(qr6i, qr6f, t)
for i=t
R1.plot(qr1)
hold on
R2.plot(qr2)
hold on
R3.plot(qr3)
hold on
R4.plot(qr4)
hold on
R5.plot(qr5)
hold on
R6.plot(qr6)
hold on
end
```

## APÉNDICE C

```
#define DXL_BUS_SERIAL1 1 //Dynamixel on Serial1(USART1) <-OpenCM9.04
#define DXL_BUS_SERIAL2 2 //Dynamixel on Serial2(USART2) <-LN101,BT210
#define DXL_BUS_SERIAL3 3 //Dynamixel on Serial3(USART3) <-OpenCM 485EXP
//Definir ID motores
#define ID_NUM3 3
#define ID_NUM4 4
#define ID_NUM5 5
```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

#define ID_NUM6 6
#define ID_NUM7 7
#define ID_NUM8 8
#define ID_NUM9 9
#define ID_NUM10 10
#define ID_NUM11 11
#define ID_NUM12 12
#define ID_NUM13 13
#define ID_NUM14 14

Dynamixel Dxl(DXL_BUS_SERIAL1);
//int i=1;
int flag=0;
unsigned long Time;
//VARIABLES DONDE SE ALMACENAN LAS CONVERSIONES
double
WFM3c,WFM4c,WFM5c,WFM6c,WFM7c,WFM8c,WFM9c,WFM10c,WFM11c,WFM12c,WFM13c,WFM14c;
double
WBM3c,WBM4c,WBM5c,WBM6c,WFB7c,WFB8c,WFB9c,WBM10c,WBM11c,WBM12c,WBM13c,WBM14c;
int torq3,torq4,torq5,torq6,torq7,torq8,torq9,torq10,torq11,torq12,torq13,torq14;
//vectores de tiempo
int forw[]={200,100,200,100};

//VECTORES DE MOVIMIENTO
//WALK FORWARD (Walk Forward Motor #)

double WFM3[]={ -3.22, -3.22, -46.88, -46.88};
double WFM4[]={ 46.58, 46.58, 2.93, 2.93};

```



 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

double WFM5[]= { 0, -20.51, -20.51, 0};
double WFM6[]= { 20.21, 0, 0, 20.21 };
double WFM7[]= { -0.29, -0.29, -0.29, -0.29};
double WFM8[]= { -0.29, -0.29, -0.29, -0.29};
double WFM9[]= { -20.80, -9.08, -8.79, -26.37};
double WFM10[]={ 8.50, 26.07, 20.51, 8.79};
double WFM11[]={ 37.79, 37.79, 2.93, 2.93};
double WFM12[]={ -3.22, -3.22, -38.09, -38.09};
double WFM13[]={ -0.29, -20.8, -20.51, 0};
double WFM14[]={ 20.21, -0.29, 0, 20.51};

```

```

void setup() {
// put your setup code here, to run once:
Dxl.begin(3);// Dynamixel 2.0 Baudrate -> 0: 9600, 1: 57600, 2: 115200, 3: 1Mbps
Dxl.jointMode(ID_NUM3); //jointMode (motor3) is to use position mode
Dxl.jointMode(ID_NUM4); //jointMode (motor4) is to use position mode
Dxl.jointMode(5); //jointMode (motor5) is to use position mode
Dxl.jointMode(ID_NUM6); //jointMode (motor6) is to use position mode
Dxl.jointMode(ID_NUM7); //jointMode (motor7) is to use position mode
Dxl.jointMode(ID_NUM8); //jointMode (motor8) is to use position mode
Dxl.jointMode(ID_NUM9); //jointMode (motor9) is to use position mode
Dxl.jointMode(ID_NUM10); //jointMode(motor10) is to use position mode
Dxl.jointMode(ID_NUM11); //jointMode(motor11) is to use position mode
Dxl.jointMode(ID_NUM12); //jointMode(motor12) is to use position mode
Dxl.jointMode(ID_NUM13); //jointMode(motor13) is to use position mode
Dxl.jointMode(ID_NUM14); //jointMode(motor14) is to use position mode

}
int i=0;

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

void loop() {
  // put your main code here, to run repeatedly:

  //if (flag==1){
  for (i=0;i<4;i++){

    //Time = millis();

    //CONVERTIR A BINARIO
    WFM3c= (((WFM3[i]+150)*1024)/300);
    WFM4c= (((WFM4[i]+150)*1024)/300);
    WFM5c= (((WFM5[i]+150)*1024)/300);
    WFM6c= (((WFM6[i]+150)*1024)/300);
    WFM7c= (((WFM7[i]+150)*1024)/300);
    WFM8c= (((WFM8[i]+150)*1024)/300);
    WFM9c= (((WFM9[i]+150)*1024)/300);
    WFM10c=(((WFM10[i]+150)*1024)/300);
    WFM11c=(((WFM11[i]+150)*1024)/300);
    WFM12c=(((WFM12[i]+150)*1024)/300);
    WFM13c=(((WFM13[i]+150)*1024)/300);
    WFM14c=(((WFM14[i]+150)*1024)/300);

    //ESCRIBIR EN LAS PATAS DEL ROBOT
    Dxl.writeWord(ID_NUM3, 30, WFM3c);
    Dxl.writeWord(ID_NUM4, 30, WFM4c);
    Dxl.writeWord(ID_NUM5, 30, WFM5c);
    Dxl.writeWord(ID_NUM6, 30, WFM6c);
    Dxl.writeWord(ID_NUM7, 30, WFM7c);
    Dxl.writeWord(ID_NUM8, 30, WFM8c);

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

Dxl.writeWord(ID_NUM9, 30, WFM9c);
Dxl.writeWord(ID_NUM10, 30, WFM10c);
Dxl.writeWord(ID_NUM11, 30, WFM11c);
Dxl.writeWord(ID_NUM12, 30, WFM12c);
Dxl.writeWord(ID_NUM13, 30, WFM13c);
Dxl.writeWord(ID_NUM14, 30, WFM14c);

//delay de espera
delay(forw[i]);
torq3=Dxl.getLoad(3);
torq4=Dxl.getLoad(4);
torq5=Dxl.getLoad(5);
torq6=Dxl.getLoad(6);
torq7=Dxl.getLoad(7);
torq8=Dxl.getLoad(8);
torq9=Dxl.getLoad(9);
torq10=Dxl.getLoad(10);
torq11=Dxl.getLoad(11);
torq12=Dxl.getLoad(12);
torq13=Dxl.getLoad(13);
torq14=Dxl.getLoad(14);
    }
}

```

## APÉNDICE D

```

#define DXL_BUS_SERIAL1 1 //Dynamixel on Serial1(USART1) <-OpenCM9.04
#define DXL_BUS_SERIAL2 2 //Dynamixel on Serial2(USART2) <-LN101,BT210

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```
#define DXL_BUS_SERIAL3 3 //Dynamixel on Serial3(USART3) <-OpenCM 485EXP
```

```
#define GOAL_POSITION 30
```

```
#define ID_NUM 18
```

```
#define MOVING 30
```

```
#define XL_MOVING 49
```

```
Dynamixel Dxl(DXL_BUS_SERIAL1);
```

```
int Movingxd ;
```

```
int goalPosition = 0;
```

```
void setup() {
```

```
    // Initialize the dynamixel bus:
```

```
    // Dynamixel 2.0 Baudrate -> 0: 9600, 1: 57600, 2: 115200, 3: 1Mbps
```

```
    Dxl.begin(3);
```

```
    // Dxl.wheelMode(ID_NUM);
```

```
    Dxl.jointMode(ID_NUM); //jointMode() is to use position mode
```

```
}
```

```
void loop()
```

```
{
```

```
    // isMoving = Dxl.readWord(ID_NUM, MOVING);
```

```
    delay(2000);
```

```
    Dxl.writeWord(ID_NUM, 30, 512);
```

```
    delay(2000);
```

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

Dxl.writeWord(ID_NUM, 30, 312); //ID 1 dynamixel moves to position 1
delay(60);
Movingxd = Dxl.readWord(ID_NUM, 40);
// Movingxd = Dxl.getLoad(ID_NUM);
SerialUSB.print("torq = ");
SerialUSB.println(Movingxd);
delay(2000);
Dxl.writeWord(ID_NUM, 30, 512);
delay(2000);

```

```

Dxl.writeWord(ID_NUM, 30, 712); //ID 1 dynamixel moves to position 1
delay(60);
Movingxd = Dxl.getLoad(ID_NUM);
SerialUSB.print("torq = ");
SerialUSB.println(Movingxd);
delay(1000);
Dxl.writeWord(ID_NUM, 30, 512);

```

```

// delay(2000);
// Dxl.goalPosition(ID_NUM, 1023); //ID 1 dynamixel moves to position 1023
// delay(2000);
}

```

## APÉNDICE E

```

#define DXL_BUS_SERIAL1 1 //Dynamixel on Serial1(USART1) <-OpenCM9.04
#define DXL_BUS_SERIAL2 2 //Dynamixel on Serial2(USART2) <-LN101,BT210
#define DXL_BUS_SERIAL3 3 //Dynamixel on Serial3(USART3) <-OpenCM 485EXP

```

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```
//Definir ID motores
```

```
#define ID_NUM3 3
```

```
#define ID_NUM4 4
```

```
#define ID_NUM5 5
```

```
#define ID_NUM6 6
```

```
#define ID_NUM7 7
```

```
#define ID_NUM8 8
```

```
#define ID_NUM9 9
```

```
#define ID_NUM10 10
```

```
#define ID_NUM11 11
```

```
#define ID_NUM12 12
```

```
#define ID_NUM13 13
```

```
#define ID_NUM14 14
```

```
Dynamixel Dxl(DXL_BUS_SERIAL1);
```

```
int i=0;
```

```
int y=0;
```

```
//VARIABLES DONDE SE ALMACENAN LAS CONVERSIONES
```

```
double
```

```
WFM3c,WFM4c,WFM5c,WFM6c,WFM7c,WFM8c,WFM9c,WFM10c,WFM11c,WFM12c,WFM13c,WFM14c,WFM6c1,WFM10c1,WFM9c1,WFM13c1,WFM5c1;
```

```
M13c,WFM14c,WFM6c1,WFM10c1,WFM9c1,WFM13c1,WFM5c1;
```

```
int torq3,torq4,torq5,torq6,torq7,torq8,torq9,torq10,torq11,torq12,torq13,torq14;
```

```
//vectores de tiempo
```

```
int forw[]={200,100,200,100};
```

```
//VECTORES DE MOVIMIENTO
```

```
//WALK FORWARD (Walk Forward Motor #)
```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

double WFM3[] = { -3.22, -3.22, -46.88, -46.88};
double WFM4[] = { 46.58, 46.58, 2.93, 2.93};
double WFM5[] = { 0, -20.51, -20.51, 0};
double WFM6[] = { 20.21, 0, 0, 20.21 };
double WFM9[] = { -20.80, -9.08, -8.79, -26.37};
double WFM10[] = { 8.50, 26.07, 20.51, 8.79};
double WFM11[] = { 37.79, 37.79, 2.93, 2.93};
double WFM12[] = { -3.22, -3.22, -38.09, -38.09};
double WFM13[] = { -0.29, -20.8, -20.51, 0};
double WFM14[] = { 20.21, -0.29, 0, 20.51};

```

```
void setup() {
```

```
  // put your setup code here, to run once:
```

```
  Dxl.begin(3); // Dynamixel 2.0 Baudrate -> 0: 9600, 1: 57600, 2: 115200, 3: 1Mbps
```

```
  Dxl.jointMode(ID_NUM3); //jointMode (motor3) is to use position mode
```

```
  Dxl.jointMode(ID_NUM4); //jointMode (motor4) is to use position mode
```

```
  Dxl.jointMode(ID_NUM5); //jointMode (motor5) is to use position mode
```

```
  Dxl.jointMode(ID_NUM6); //jointMode (motor6) is to use position mode
```

```
  Dxl.jointMode(ID_NUM7); //jointMode (motor9) is to use position mode
```

```
  Dxl.jointMode(ID_NUM8); //jointMode (motor6) is to use position mode
```

```
  Dxl.jointMode(ID_NUM9); //jointMode (motor9) is to use position mode
```

```
  Dxl.jointMode(ID_NUM10); //jointMode(motor10) is to use position mode
```

```
  Dxl.jointMode(ID_NUM11); //jointMode(motor11) is to use position mode
```

```
  Dxl.jointMode(ID_NUM12); //jointMode(motor12) is to use position mode
```

```
  Dxl.jointMode(ID_NUM13); //jointMode(motor13) is to use position mode
```

```
  Dxl.jointMode(ID_NUM14); //jointMode(motor14) is to use position mode
```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```
}

```

```
void loop() {

```

```
// put your main code here, to run repeatedly:

```

```
for (i=0;i<4;i++){

```

```
//CONVERTIR A BINARIO

```

```
WFM3c= (((WFM3[i]+150)*1024)/300);

```

```
WFM4c= (((WFM4[i]+150)*1024)/300);

```

```
WFM5c= (((WFM5[i]+150)*1024)/300);

```

```
WFM6c= (((WFM6[i]+150)*1024)/300);

```

```
WFM9c= (((WFM9[i]+150)*1024)/300);

```

```
WFM10c=(((WFM10[i]+150)*1024)/300);

```

```
WFM11c=(((WFM11[i]+150)*1024)/300);

```

```
WFM12c=(((WFM12[i]+150)*1024)/300);

```

```
WFM13c=(((WFM13[i]+150)*1024)/300);

```

```
WFM14c=(((WFM14[i]+150)*1024)/300);

```

```
// if(torq3 > 1750)

```

```
// WFM5c=(((WFM5[i]+120)*1024)/300);

```

```
if(torq4 > 1750)

```

```
WFM6c=(((WFM6[i]+180)*1024)/300);

```

```
if(WFM6c1 > 610)

```

```
WFM10c=(((WFM10[i]+180)*1024)/300);

```

```
if(WFM10c1 > 610)

```

```
WFM14c=(((WFM14[i]+180)*1024)/300);

```

```
if(torq3 > 1750)

```

```
WFM5c=(((WFM5[i]+120)*1024)/300);

```



 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

if(WFM5c1 > 610)
WFM9c=((WFM9[i]+120)*1024)/300);
if(WFM9c1 > 610)
WFM13c=((WFM13[i]+120)*1024)/300);

```

```
//ESCRIBIR EN LAS PATAS DEL ROBOT
```

```

Dxl.writeWord(ID_NUM3, 30, WFM3c);
Dxl.writeWord(ID_NUM4, 30, WFM4c);
Dxl.writeWord(ID_NUM5, 30, WFM5c);
Dxl.writeWord(ID_NUM6, 30, WFM6c);
Dxl.writeWord(7, 30, 512);
Dxl.writeWord(8, 30, 512);
Dxl.writeWord(ID_NUM9, 30, WFM9c);
Dxl.writeWord(ID_NUM10, 30, WFM10c);
Dxl.writeWord(ID_NUM11, 30, WFM11c);
Dxl.writeWord(ID_NUM12, 30, WFM12c);
Dxl.writeWord(ID_NUM13, 30, WFM13c);
Dxl.writeWord(ID_NUM14, 30, WFM14c);

```

```

delay(40);
torq3=Dxl.getLoad(3);
torq4=Dxl.getLoad(4);

```

```

WFM5c=WFM5c1=Dxl.getPosition(5);
torq5=Dxl.getLoad(5);
WFM6c1=Dxl.getPosition(6);
WFM6c=Dxl.getPosition(6);
torq6=Dxl.getLoad(6);
WFM9c=WFM9c1=Dxl.getPosition(9);

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```
torq9=Dxl.getLoad(9);  
WFM10c=Dxl.getPosition(10);  
WFM10c1=Dxl.getPosition(10);  
torq10=Dxl.getLoad(10);  
WFM13c=Dxl.getPosition(13);  
torq13=WFM13c1=Dxl.getLoad(13);  
WFM14c=Dxl.getPosition(14);  
torq14=Dxl.getLoad(14);  
SerialUSB.println(torq6);
```

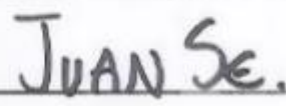
```
//delay de espera
```

```
delay(forw[i]);
```

```
}//llave cierre for
```

```
}//llve cierre loop
```

	<b>INFORME FINAL DE TRABAJO DE GRADO</b>	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FIRMA ESTUDIANTES	 <hr/>  <hr/>
FIRMA ASESOR	 <hr/>
	FECHA ENTREGA: _____

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____
RECHAZADO___      ACEPTADO___      ACEPTADO CON MODIFICACIONES___
ACTA NO. _____
FECHA ENTREGA: _____

FIRMA CONSEJO DE FACULTAD _____
ACTA NO. _____
FECHA ENTREGA: _____