



Lámpsakos

E-ISSN: 2145-4086

lampsakos@amigo.edu.co

Fundación Universitaria Luis Amigó

Colombia

Guzmán, Jaime A.; Gutiérrez, Andrés F.; Meza, Juan D.
Estudio y Modelado de un Brazo Robótico Lego de tres grados de libertad
Lámpsakos, núm. 6, julio-diciembre, 2011, pp. 5-11
Fundación Universitaria Luis Amigó
Medellín, Colombia

Disponible en: <https://www.redalyc.org/articulo.oa?id=613965338002>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Study and modeling of a Lego robotic arm with three degrees of freedom

Estudio y Modelado de un Brazo Robótico Lego de tres grados de libertad

Jaime A. Guzmán; Andrés F. Gutiérrez; Juan D. Meza

*Grupo de Investigación SINTELWEB, Escuela de Sistemas Facultad de Minas,
Universidad Nacional de Colombia, sede Medellín*

{jaguzman, afgutierrez, judmezago}@unal.edu.co

(Artículo de INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA. Recibido el 22/10/2011. Aprobado el 06/12/2011)

Abstract

This article presents the study and modeling of a Lego robotic arm with three degrees of freedom in association with their respective models of direct and inverse kinematics with an approach to dynamic analysis to generate trajectories of this robot. The equations governing the motion are deduced from the mathematical models proper to the kinematics chains and are implemented in the Java programming language and executed by the robot through the Java-LeJOS extension.

Keywords: Robotics, Forward Kinematics, Inverse Kinematics, trajectory generation, Lego Robot.

Resumen

En este artículo se presenta el estudio y modelado de un brazo robótico lego de tres grados de libertad en asocio con sus respectivos modelos de su cinemática directa e inversa junto con una aproximación al análisis dinámico para la generación de trayectorias de este robot. Las ecuaciones que rigen el movimiento son deducidas de los modelos matemáticos propios a las cadenas cinemáticas y son implementadas en el lenguaje de programación Java y ejecutadas por el robot a través de la extensión Java –LeJOS.

Palabras clave: Robótica, Cinemática Directa, Cinemática Inversa, Generación de trayectorias, Robot Lego.

I. INTRODUCCIÓN

El uso de robots, junto con los sistemas de diseño y sistemas de fabricación asistidos por computadora (CAD) y (CAM) respectivamente, son la última tendencia en automatización de los diferentes procesos requeridos en la industria actual, como la soldadura, pinturas de aerosol, transporte, molienda de materiales, moldeado en la industria plástica, máquinas-herramientas, y otras más.[1]

Llevar estos objetivos a cabo requiere de la correcta ejecución de los movimientos del robot y esto se logra haciendo uso de los conceptos que regulan la cinemática, dinámica y el control de movimientos del robot [2].

Actualmente están al alcance una variedad de aplicaciones que soportan las tareas de diseño, construcción,

simulación, comunicación y programación de robots. En este contexto se reconocen herramientas tales como: Matlab-Simulink [3], Microsoft Robotics Developer Studio [4], Artificial Robotics [5], Google Sketchup [6], Al igual que kits para construir rápidamente prototipos de robots tales como el Lego Mindstorm 1.0 [8], Lego Mindstorm Education [9] y otros tipos de robots como el Pioneer P3 – DX [10] o Handy Cricket [11] entre otros; sin embargo son escasas las aplicaciones que hacen uso de uno de los lenguajes de programación más conocidos como Java [7].

En este artículo se pretende como primer objetivo mostrar cómo se realizó el diseño e implementación de un brazo robótico haciendo uso del Kit Lego Mindstorm NXT 2.0. [8].

Como Segundo objetivo se pretende mostrar cómo se dio solución a los dos problemas fundamentales en un

brazo robótico como son, el problema cinemático y el de trayectorias. El primero es solucionado por medio del uso del algoritmo de Denavit Hartenberg (D-H) [14]. Este algoritmo constituye en sí la forma más robusta de darle solución a ambos problemas y se caracteriza por el uso de matrices de transformaciones homogéneas para pasar de sistemas coordenados desde la base del brazo hasta su efector final. El segundo es abordado usando el método "rest-to-rest"[2] para coordinar el movimiento de las dos articulaciones en función de tiempo.

Como tercer objetivo se pretende detallar la implementación del programa de control del brazo robótico construido basado en los conceptos planteados en el objetivo anterior, describiendo el algoritmo básico del mismo al igual que algunos aspectos de diseño de su interfaz de usuario.

Este documento está organizado de la siguiente manera: en la sección II Diseño y construcción del brazo robótico, se presenta el proceso de selección de materiales y ensamblaje de los mismos, en la sección III Manejo de la cinemática y trayectorias de movimientos, se exponen los modelos matemáticos utilizados para darle solución al problema cinemático y de trayectorias, en la sección IV Implementación, se muestra un fragmento del código y la interfaz grafica de la aplicación creada, en la sección V Resultados obtenidos, se expone como fueron los valores reales vs los valores teóricos para el problema cinemático y los valores para un problema de trayectoria lineal específico.

Por ultimo en las secciones VI, VII, VIII y IX se presentan las Conclusiones, Trabajos Futuros, Agradecimientos y Referencias respectivamente.

II. DISEÑO Y CONSTRUCCIÓN DEL BRAZO ROBÓTICO

En cuanto al diseño de un brazo robótico, independientemente de su funcionalidad proyectada, se requieren ciertos conocimientos con respecto a la estructura como tal de cada uno de estos, por ejemplo la fortaleza estructural que deben tener, el equilibrio y estabilidad de cada una de sus componentes y la distribución de estas [12]. En particular, para diseñar un brazo robótico utilizando las piezas de un kit Lego Mindstorm NXT, se debe tener en cuenta el peso y en especial longitud de las extremidades que conforman el brazo [13].

Dada la disponibilidad de un kit Lego Mindstorm NXT 2.0, luego de analizar las facilidades y restricciones que ofrece este kit, se tomó la decisión de construir un brazo robótico con tres grados de libertad similar a un robot de tipo SCARA. Para ello se tomó como referencia dos modelos existentes en la literatura como son el desarrollado por Dave Parker [17], y el desarrollado por Yoriyama Yamamoto [18].

En la etapa de diseño de este robot, hizo uso de la herramienta Google Sketchup [6], con la cual se realizó un modelo tridimensional del brazo robótico que permitió identificar los componentes reales a ser utilizados en el robot a construir, al igual que permitió tener un modelo computacional del mismo desde cualquier perspectiva del brazo, facilitando de este modo su construcción real (ver Fig 1).

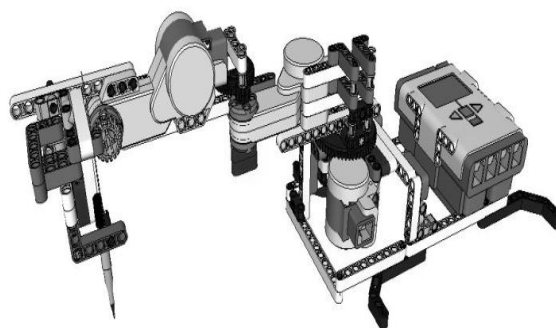


Fig. 1. Modelo 3D del brazo robótico diseñado con Google Sketchup.

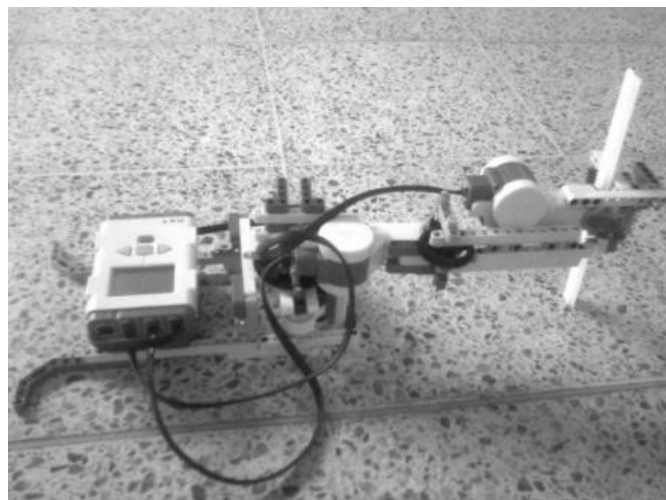


Fig. 2. Imagen real, del diseño final del brazo robótico.

En el diseño del robot se tuvieron en cuenta dos consideraciones básicas: **(i)** con el fin de asegurar una estructura adecuadamente rígida y estable, las extremidades que conforman el brazo se diseñaron lo más corta posible y **(ii)** Con el fin de disminuir al máximo posible la inclinación del efector final con respecto a la base, se buscó mantener los motores encargados del movimiento de cada componente lo más cercanos posible a la base del robot, buscando con esto disminuir la fuerza que la estructura deba soportar debido al peso de estos.

III. MANEJO DE LA CINEMATICA Y TRAYECTORIAS DE MOVIMIENTOS

Inicialmente se analizará la cinemática enfocada hacia el estudio de brazos robóticos como pilar de todo el proyecto, estableciendo en primera instancia la descripción analítica del movimiento espacial del brazo robótico, basándose en las propiedades físicas del modelo.

En el siguiente bosquejo están expuestos los ejes coordenados correspondientes a cada articulación en los cuales está basada la extracción de las matrices homogéneas de transformación.

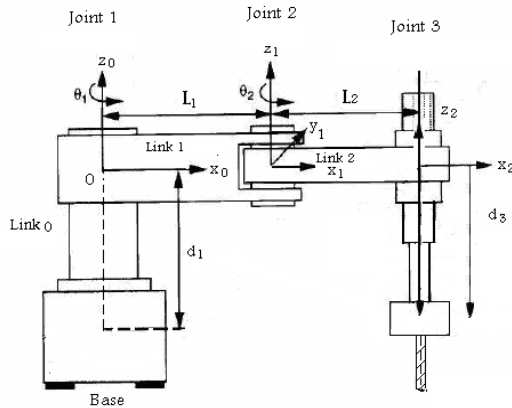


Fig. 3. (Tomada de: Kinematic Modeling and Simulation of a SCARA Robot by Using Solid Dynamics and Verification by MATLAB/Simulink)

El primer paso consiste en establecer los parámetros D-H. Que se representan por convención en una tabla. Dicha tabla proporciona la información necesaria, para llevar a cabo posteriormente los cálculos cinemáticos correspondientes a cada grado de libertad del brazo robótico, es decir, que por cada grado de libertad deberá existir una tabla como la siguiente:

TABLA 1: Matriz D-H: Matriz con los parámetros correspondientes al algoritmo de Denavit Hartenberg.

i	θ_i	D_i	a_i	α_i
1	θ_1	0	L_1	0
2	θ_2	0	L_2	0
3	0	d_3	0	0

Luego con estos parámetros se procede a formular la matriz de transformación para cada una de las articulaciones utilizando la convención del algoritmo de D-H.

$$A_1 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & L_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 & 0 & L_1 \sin \theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$A_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & L_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & L_2 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

A. Solución al problema cinemático directo

La cinemática directa se preocupa por conocer la posición del efector final del brazo después de ejecutar algunos movimientos es sus articulaciones, una vez obtenidas las matrices de transformación, la solución al problema se reduce a multiplicar las matrices y reemplazar los valores en la matriz resultante.

$$A = A_1 * A_2 * A_3 \quad (4)$$

Para facilidades de visualización se expresara:

$\cos \theta_1$ Como c_1 , $\cos \theta_2$ como c_2 , $\sin \theta_1$ como s_1 , $\sin \theta_2$ como s_2 .

$$A = \begin{bmatrix} c_1 c_2 - s_1 s_2 & -c_1 s_2 - c_2 s_1 & 0 & L_1 c_1 + L_2 c_1 c_2 - L_2 s_1 s_2 \\ c_1 s_2 + c_2 s_1 & c_1 c_2 - s_1 s_2 & 0 & L_1 s_1 + L_2 c_1 s_2 + L_2 c_2 s_1 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Para el propósito del proyecto se enfoca la atención en la posición del efector final en el plano (X,Y) formado en el piso, estas coordenadas las brinda la matriz anterior en las posiciones $A_{1,4}$ y $A_{2,4}$.

Por lo cual se puede concluir que respecto al eje coordenado del origen, las coordenadas (X,Y) son las siguientes:

$$X = L1 \cos\theta_1 + L2 \cos\theta_1 \cos\theta_2 - L2 \sin\theta_1 \sin\theta_2 \quad (6)$$

$$Y = L1 \sin\theta_1 + L2 \cos\theta_1 \sin\theta_2 + L2 \cos\theta_2 \sin\theta_1 \quad (7)$$

Pasando del modelo lógico al modelo físico reemplazamos los valores de las constantes L1 y L2 por las correspondientes longitudes en el brazo robótico construido.

Con lo que se obtiene las siguientes expresiones:

$$X = 9,5 \cos\theta_1 + 11,8 \cos\theta_1 \cos\theta_2 - 11,8 \sin\theta_1 \sin\theta_2 \quad (8)$$

$$Y = 9,5 \sin\theta_1 + 11,8 \cos\theta_1 \sin\theta_2 + 11,8 \cos\theta_2 \sin\theta_1 \quad (9)$$

Donde (X,Y) son medidos en centímetros.

B. Solución al problema cinemático inverso

Expresando el resultado obtenido en la cinemática directa en forma vectorial se obtiene el siguiente sistema de ecuaciones.

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} L1 \cos\theta_1 + L2 \cos\theta_1 \cos\theta_2 - L2 \sin\theta_1 \sin\theta_2 \\ L1 \sin\theta_1 + L2 \cos\theta_1 \sin\theta_2 + L2 \cos\theta_2 \sin\theta_1 \end{bmatrix} \quad (10)$$

Utilizando identidades trigonométricas se reescribe como:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} L1 \cos\theta_1 + L2 \cos(\theta_1 + \theta_2) \\ L1 \sin\theta_1 + L2 \sin(\theta_1 + \theta_2) \end{bmatrix} \quad (11)$$

Dado que X e Y son conocidos y L1 y L2 son constantes se procede a despejar θ_1 y θ_2 .

Elevando al cuadrado a ambos lados:

$$X^2 + Y^2 = L1^2 + L2^2 + 2 L1 L2 \cos\theta_2 \quad (12)$$

$$\cos\theta_2 = \frac{X^2 + Y^2 - L1^2 - L2^2}{2 L1 L2} \quad (13)$$

$$\theta_2 = \cos^{-1} \frac{X^2 + Y^2 - L1^2 - L2^2}{2 L1 L2} \quad (14)$$

Dada la inexactitud de las funciones \sin^{-1} y \cos^{-1} se utiliza la fórmula del Angulo mitad:

$$\tan^2 \frac{\theta}{2} = \frac{1 - \cos\theta}{1 + \cos\theta} \quad (15)$$

Y por medio de la función atan2 se encuentra el valor de θ_2

$$\theta_2 = \pm 2 \operatorname{atan2} \sqrt{\frac{(L1+L2) - (X^2 + Y^2)}{(X^2 + Y^2) - (L1-L2)}} \quad (16)$$

Nótese que la solución anterior puede adoptar un valor positivo y uno negativo, esto se debe a que el problema cinemático inverso no tiene solución única, para el caso de un brazo de dos articulaciones las soluciones se llaman "codo arriba" y "codo abajo" como se muestra en la figura [X].

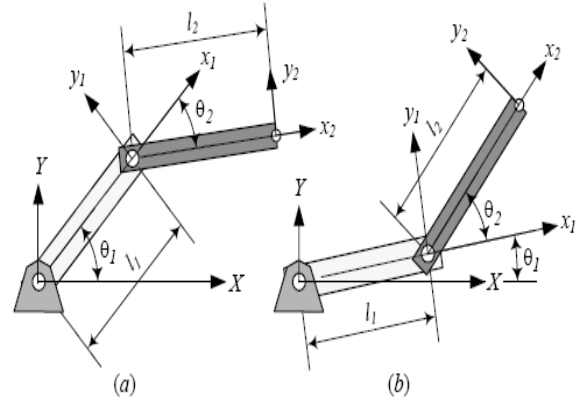


Fig. 4. Tomada de Theory of Applied Robotics[2] pag. 332

Una vez conocido el valor de θ_2 es posible encontrar una expresión para θ_1 geométricamente.

$$\theta_1 = \operatorname{atan2} \sqrt{\frac{Y}{X}} + \operatorname{atan2} \sqrt{\frac{L2 \sin\theta_2}{L1 + L2 \cos\theta_2}} \quad (17)$$

C. Trayectorias

Planear una trayectoria para el efector final del brazo robótico requiere de una coordinación de las dos articulaciones, para lo cual se usa el tiempo como parámetro.

Si la trayectoria es una línea recta esta estará definida por un par de puntos, en el plano cartesiano, estos delimitarán donde comenzara el movimiento y donde terminará.

Utilizando el método "rest-to-rest" se generan las ecuaciones para la posición en X e Y en función del tiempo de la forma:

$$X(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (18)$$

$$Y(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (19)$$

Donde los coeficientes a_0 , a_1 , a_2 y a_3 toman su valor partiendo de la coordenada en X del punto inicial p_0 y el punto final p_f para (18) y las coordenadas en Y en los mismos puntos para (19).

Tomando como tiempo inicial $t_0 = 0$ y t_f el tiempo requerido para realizar la trayectoria, las expresiones son las siguientes:

$$a_0 = q_0 \quad (20)$$

$$a_1 = q'_0 \quad (21)$$

$$a_2 = \frac{3(q_f - q_0) - (2q'_0 + q'_f)t_f}{t_f^2} \quad (22)$$

$$a_3 = \frac{-2(q_f - q_0) - (q'_0 + q'_f)t_f}{t_f^2} \quad (23)$$

Una vez conocidas la posiciones en función del tiempo, se procede a hallar la rotación de cada variable articular θ_1 y θ_2 en función del tiempo, utilizando las ecuaciones de la cinemática inversa.

Para ello basta con reemplazar (18) y (19) en (16) y (17). Hecho esto se encuentra la expresión para $\theta_1(t)$ y $\theta_2(t)$ que representen el ángulo de cada articulación para cualquier instante durante la trayectoria.

Evaluando estos resultados en el modelo físico se requirió además conocer la velocidad de la articulación para cada instante de tiempo por lo cual se derivó las ecuaciones $\theta_1(t)$ y $\theta_2(t)$ obteniendo $\theta_1'(t)$ y $\theta_2'(t)$.

IV. IMPLEMENTACIÓN

En este segmento de código se expone como el programa provee las dos posibles soluciones al problema cinemático inverso recibiendo como entrada las especificaciones físicas del robot y las coordenadas del punto al cual se desea llegar.

```
public static double[] inversa(Robot R, double px, double py)
{
    double a[] = new double[4];
    double l1 = R.getL1();
    double l2 = R.getL2();

    double c2 = (Math.pow(px, 2) + Math.pow(py, 2) - Math.pow(l1, 2) - Math.pow(l2, 2)) / (2 * l1 * l2);
    double s2 = Math.sqrt(1 - Math.pow(c2, 2));
    double s22 = -Math.sqrt(1 - Math.pow(c2, 2));
    double t2 = Math.atan2(s2, c2);

    double t4 = Math.atan2(s22, c2);

    double t1 = Math.atan2(py, px) - Math.atan2((l2 * s2), l1 + (l2 * c2));
    double t3 = Math.atan2(py, px) - Math.atan2((l2 * s22), l1 + (l2 * c2));

    a[0] = Math.toDegrees(t1);
    a[1] = Math.toDegrees(t2);
    a[2] = Math.toDegrees(t3);
    a[3] = Math.toDegrees(t4);

    return a;
}
```

Fig. 5. Segmento de código para la Cinemática Inversa creada para la aplicación

Para fines de evaluación y veracidad se puede hacer uso de la cinemática directa y evaluar la reciprocidad esperada en la cinemática inversa, Aplicando una rotación para valores específicos se obtiene una posición del efector final tal como se aprecia en la figura 6.

Fig.6. Segmento para la Cinemática Directa de la interfaz gráfica creada para la aplicación

Reemplazando dichos valores en los campos que requiere la cinemática inversa se obtienen los siguientes resultados.

Fig. 7. Segmento para la Cinemática Inversa de la interfaz gráfica creada para la aplicación

La aplicación retorna las dos soluciones para el punto (x,y) especificado, correspondientes a los casos de “codo arriba” y “codo abajo”. El error que se observa en la interfaz está dada por la conversión de radianes a grados que debe ejecutar la máquina virtual de Java [19].

V. RESULTADOS OBTENIDOS

Para probar la efectividad del procedimiento se realizaron algunas pruebas para diferentes ángulos de rotación en ambas articulaciones. Los resultados se exponen a continuación:

TABLA 2: Pruebas de ángulos de rotación

i	$\theta 1$	$\theta 2$	X Robot (cm)	Y Robot (cm)	X Teórico (cm)	Y Teórico (cm)
1	0	45	20.1	9.7	17.84	8.34
2	30	30	15.4	16.3	14.13	14.96
3	45	0	15.2	15.2	15.06	15.06
4	90	-30	7.7	17.7	5.90	19.72

Las causas más directas de los errores obtenidos en los resultados físicos, con respecto a lo arrojado por el computador, se deben a la forma y estructura del brazo robótico, dado la debilidad estructural anteriormente mencionada, el brazo presenta cierta inclinación del efector final con respecto a la base del mismo, adicionalmente, en el diseño final del brazo no se presenta una linealidad ideal de todas sus partes, como en el análisis cinemático.

TABLA 3: Diferencias entre los ángulos de rotación

i	Diferencia X	Diferencia Y
1	2.26	1.36
2	1.27	1.34
3	0.14	0.14
4	1.80	-2.02

Suponga que el brazo efectuará una trayectoria en línea recta de un punto p_0 al punto p_f .

$$p_0 = (10.5998, 21.2064)$$

Donde los ángulos son: $\theta 1(t) = 45$ y $\theta 2(t) = 30$
 $p_f = (-12.9903, 17.0000)$

Donde los ángulos son: $\theta 1(t) = 90$ y $\theta 2(t) = 60$

Los resultados de realizar el análisis expuesto anteriormente se pueden apreciar en las gráficas del cambio de la posición y de la velocidad a través del tiempo, donde se estableció que la trayectoria debe ser efectuada en 10 segundos y las velocidades al iniciar y finalizar serán cero.

En la Figura 8 se exponen los ángulos para las variables articulares $\theta 1$ y $\theta 2$ durante el transcurso del tiempo fijado. Se puede observar que los valores para los tiempos $t = 0$ y $t = 10$, son los que se establecieron inicialmente con una leve desviación en el ángulo $\theta 1$, el cual requiere de más cálculos y por lo tanto presenta más error que el ángulo $\theta 2$, el cual es mucho más preciso.

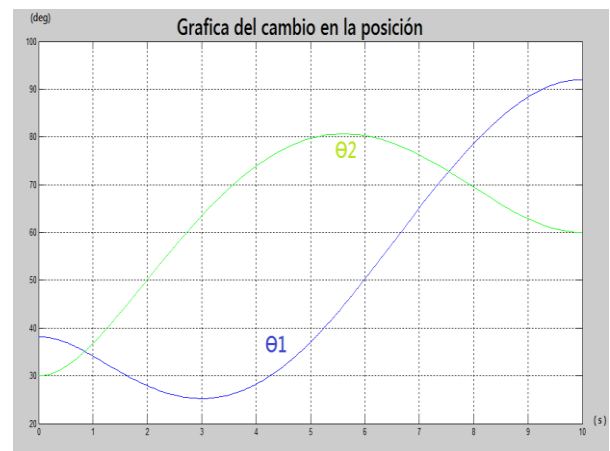


Fig. 8. Gráfica de $\theta 1(t)$ y $\theta 2(t)$.

El brazo robótico debe tener dentro de su entorno de trabajo los ángulos expuestos en la Fig. 7, pero llevar el efector final, punto por punto; durante el proceso generaría una trayectoria por tramos y no una trayectoria lineal como se busca. Para suplir esta necesidad y conociendo la facilidad que proporcionan los motores del brazo, que nos permite establecer la velocidad de los motores, se derivaron las ecuaciones $\theta 1(t)$ y $\theta 2(t)$, de esta forma se consigue que los motores estén siempre en movimiento, variando su velocidad a través del tiempo por medio de $\theta 1'(t)$ y $\theta 2'(t)$ y la trayectoria descrita sea la esperada.

VI. CONCLUSIONES

Las piezas de las cuales está constituido el robot, entre ellas los motores, uniones, engranajes son una excelente herramienta para la robótica en general, pero presentan muchas dificultades al momento de ensamblar un brazo robótico firme y estable, algunos componentes por fuera de los kit lego MINDSTORM y Lego MINDSTORM Education

suplen esta necesidad, con nuevas piezas que transmiten el movimiento de un sentido al otro y reposan el peso sobre bases sólidas y móviles.

En la sección de las trayectorias se expuso la metodología para encontrar la posición y velocidad de cada articulación en función del tiempo, el tamaño de dichas ecuaciones es considerable y requiere realizar cálculos con funciones trigonométricas que constituyen en si un costo computacional y una aproximación para ciertos valores, motivo por el cual los datos nos son siempre exactos.

El número de grados de libertad de un brazo robótico, define su capacidad y agilidad para cumplir ciertas tareas, sin embargo también aumenta en gran cantidad la complejidad de los cálculos cinemáticos necesarios para hallar la posición del efector en un tiempo cualquiera, un brazo de 3 grados de libertad es suficiente para realizar una tarea como describir trayectorias.

VII. TRABAJOS FUTUROS

En posteriores trabajos se pretende enfatizar en las trayectorias, no solo lineales sino trayectorias más genéricas e irregulares, que le permitan al robot llevar a cabo tareas con limitación de espacios.

Reducir el error en las medidas con un rediseño o estabilización del robot físico.

VIII. AGRADECIMIENTOS

Este trabajo es apoyado por el proyecto "Programa de Fortalecimiento a Grupos de Investigación y Creación Artística de la Universidad Nacional de Colombia", mediante el cual se apoyan semilleros de investigación de la sede. Y desarrollado al interior del grupo de investigación SINTELWEB.

IX. REFERENCIAS

- [1] M. Edwardsa "Robots in industry: An overview" Department of Engineering Production, PO Box 363, University of Birmingham, Birmingham B15 2TT

- [2] Jazar, Reza, "Theory of Applied Robotics Kinematics, Dynamics, and Control" 2nd ed., 2010, XXIII, 883 p. 200 illus.
- [3] Matlab-simulink [En Línea] Disponible en: <http://www.mathworks.com/products/simulink>
- [4] Microsoft Robotics Developer Studio [En Línea] Disponible en: <http://www.microsoft.com/robotics/>
- [5] Artificial Robotics [En Línea] Disponible en: <http://artificialrobotics.com/>
- [6] Google Sketchup [En Línea] Disponible en: <http://sketchup.google.com/intl/es/>
- [7] Java Developer Studio [En Línea] Disponible en: <http://www.oracle.com/technetwork/java/index.html>
- [8] Lego Mindstorm Nxt 2.0 y 1.0 [En Línea] Disponible en: <http://mindstorms.lego.com/en-us/Default.aspx>
- [9] Lego Education [En Línea] Disponible en: <http://www.legoeducation.us/store/detail.aspx?ID=1263&bhcp=1>
- [10] Pioneer P3 - DX [En Línea] Disponible en: <http://www.mobilerobots.com/researchrobots/researchrobots/pioneerp3dx.aspx>
- [11] Handy Cricket [En Línea] Disponible en: <http://www.handyboard.com/cricket/>
- [12] D. G. Lee, N. P. Suh "Axiomatic Design and Fabrication of Composite Structures: Applications in Robots, Machine Tools, and Automobiles", pp. 732. Oxford University Press, Nov 2005.
- [13] L. Valk, "The Lego Mindstorms NXT 2.0 Discovery Book. A Beginner's Guide to Building and Programming Robots" May 2010, 336 pp.
- [14] D. Hartenberg, J. Hartenberg, "A Kinematic Notation For Lower Pair Mechanics Based On Marices", 1995 pp. 215-221.
- [15] Java LeJOS [En Línea] Disponible en: <http://lejos.sourceforge.net>
- [16] TurnTable [En Línea] Disponible en: <https://www.legoeducation.us/store/detail.aspx?ID=1323&bhcp=1>
- [17] Robot Arm [En Línea] Disponible en: http://www.nxtprograms.com/robot_arm/steps.html
- [18] NXT Scara (Two-Link Planar Robot Arm) Controller Design [En Línea] Disponible en: <http://www.mathworks.com/matlabcentral/fileexchange/22126>
- [19] H. M. Deitel y P. J. Deitel, "Java How to Program" 2012, pp.1536