SIMULADOR DE UN ROBOT DE 6GL INTEGRANDO MATLAB, SOLIDWORKS **Y LABVIEW**

Vázquez Cuevas Ignacio Javier¹, López Corella José Alejandro², Hernández Ruiz Sergio Iván³, Herrera Velarde Maribel⁴, Ramírez Valenzuela Julio Cesar⁵, Pérez Arce Beatrice⁶

Tecnológico Nacional de México / I.T. de Nogales, Metal Mecánica Ingeniería Mecatrónica, Ingeniería Eléctrica y Electrónica, Av. Tecnológico 911, Nogales, Sonora, México +52(631) 311 1881. Ext 1129. *jose.lc@nogales.tecnm.mx*

RESUMEN.

En el presente trabajo se demuestra el desarrollo de un simulador de Robot de 6 GDL utilizando LabView, para la solución de la cinemática directa se utilizó el Algoritmo de Denavit-Hartenberg, en el modelado de la cinemática inversa se utilizó Desacoplo Cinemático para encontrar la solución de posición y orientación, se comprobó el modelado matemático en Matlab utilizando la herramienta GUI (graphical user interface), posteriormente se desarrolló el simulador del robot en LabView con objetos 3D exportados de SolidWorks, de esta manera se pudo comparar el funcionamiento correcto del simulador en LabView en la cual es una interfaz mucho más versátil y amigable para la creación de un prototipo físico del robot como futuro desarrollo ya que la simulación da la certeza del funcionamiento de la cinemática como proceso previo para la aplicación física del Robot de 6GDL.

Palabras Clave: Robot 6 GDL, LabView, Matlab, SolidWorks

ABSTRACT.

In the present work the development of a 6 DOF Robot simulator using LabView is demonstrated, for the solution of the direct kinematics the Denavit - Hartenberg Algorithm was used, in the modeling of the inverse kinematics Kinematic Decoupling was used to find the solution of position and orientation, the mathematical modeling was checked in Matlab using the GUI (graphical user interface) tool, later the robot simulator was developed in LabView with 3D objects exported from SolidWorks, in this way it was possible to compare the correct operation of the simulator in LabView in which it is a much more versatile and friendly interface for the creation of a physical prototype of the robot as a future development since the simulation gives us the certainty of the operation of the kinematics as a previous process for the physical application of the 6GDL Robot.

Keywords: Robot 6 GDL, LabView, LabView, Matlab, SolidWorks

1. INTRODUCCIÓN

Existen muchas herramientas o software para el desarrollo de simuladores de robots en 3D, se han creados simuladores utilizando en WOLFRAM MATHEMATICA [1], también se han desarrollado robots con el TOOLBOX ROBOTICS del software MATLAB [2] en la cual facilita la aplicación de los modelos matemáticos creando una interfaz interactiva, también se han desarrollado simuladores con otros lenguajes de programación como C# [3] utilizando librerías de OpenGL, también se han creado simuladores de robots utilizando SolidWorks con una herramienta llamada SIMMECHANICS en la cual emula los movimientos del robot integrando Matlab y SolidWorks.

En el presente trabajo se muestra el procedimiento del desarrollo del robot de 6 GDL integrando diferentes tecnologías facilitando el diseño del simulador en 3D, como se observa en la figura 1 primero se desarrolla los modelados matemáticos de cinemática directa e inversa con el software Matlab con el objetivo de simular los modelos matemáticos del robot y validarlos, posteriormente se diseñó el robot en SolidWorks, se tiene que crear sub ensamblajes de cada parte móvil del robot, esto con el fin de crear el simulador usando estas partes móviles, y para finalizar se crea el simulador con Labview, en este software se integran las modelos de las partes móviles creadas en SolidWorks y estas se manipulan con los modelos matemáticos creados en Matlab.



Figura 1.- Integración de software

2. MATLAB

Con el software de Matlab se comprobaron los modelos matemáticos de la cinemática directa e inversa, se realizaron funciones para poder facilitar el desarrollo de la simulación en LabView utilizando el complemento MathScript.

Las funciones que se crearon son:

- MatrizDH (tetha, d, a, alfa) •
- Directa6GDL(q,d,a) •
- Inversa6GDL (t, codo, muñeca)

Matriz Denavit – Hartenberg

La función "MatrizDH" crea la matriz homogénea de 4X4 según el algoritmo de Denavit-Hartenverg en la tabla 4 se pude ver el código de esta función. [4] [5] [6]

Función: MatrizDH (tetha, d, a, alfa) d: es la traslación por el eje Z a: es la traslación por el eje X alfa: es la rotación por el eje X. **tetha**: es el ángulo de rotación por el eje z.

Function dh=MatrizDH (tetha, d, a, alpha)
dh=[cos(tetha), -cos(alpha)*sin(tetha),
sin(alpha)*sin(tetha), a*cos(tetha); sin(tetha),
cos(alpha)*cos(tetha), -sin(alpha)*cos(tetha),
a*sin(tetha); 0, sin(alpha), cos(alpha), d;0 0 0 1];
end

Tabla 1 código de la función MatrizDH

Cinemática Directa

Se creó la función Directa6GDL para realiza los parámetros de las articulaciones del Robot 6 GDL como se muestra en la tabla 2 y figura 2. [4] [5] [6]



Figura 2. Configuración robot 6 GDL

No	Articulación	θİ	Li	di	αί
1	H_0^1	θ_1	ι_1	0	$-\frac{\pi}{2}$
2	H_1^2	θ_2	0	a2	0
3	H_2^3	$ heta_3$	0	0	$\frac{\pi}{2}$
4	H_3^4	$ heta_4$	ι_4	0	$-\frac{\pi}{2}$
5	H_4^5	θ_5	0	0	$\frac{\pi}{2}$
6	H_5^6	θ_6	l ₆	0	0

Tabla 2.- eslabones del robot 6 GDL

$H_0^6 = H_0^1 X H_1^2 X H_2^3 X H_3^4 X H_4^5 X H_5^6$

La función "Directa6GDL" devuelve el resultado de la cinemática directa, esta función devuelve dos resultados, el primero es el resultado de la matriz de transformación del robot de 6 GDL, el segundo resultado regresa los puntos para simular el robot. En la tabla 3 se puede ver el código de esta función.

Función : Directa6GDL(q,d,a) q: Ángulos de rotación de cada grado de libertad. d: tamaño de los eslabones. a :traslación de beta function [H06,puntos] = Directa6GDL(q,d,a) Parámetros Denavit-Hartenberg del robot tetha = q; aph = [-pi/2 0 pi/2 -pi/2 pi/2 0]; Matrices de transformación homogénea entre sistemas de coordenadas consecutivos H01 = MatrizDH(tetha(1), d(1), a(1), aph (1)); H12 = MatrizDH(tetha(2), d(2), a(2), aph(2));H23 = MatrizDH(tetha(3), d(3), a(3), aph(3));H34 = MatrizDH(tetha(4), d(4), a(4), aph(4));H45 = MatrizDH(tetha(5), d(5),a(5),aph (5)); H56 = MatrizDH(tetha(6), d(6),a(6),aph (6)); Matrices de transformación sistema al del primer correspondiente H06 = H01 + H12 + H23 + H34 + H45 + H56;% Vector de posición (x, y, z) de cada sistema de coordenadas x0 = 0; y0 = 0; z0 = 0;x1 = H01(1,4); y1 = H01(2,4); z1 = H01(3,4); $x^{2} = HO2(1, 4); \quad y^{2} = HO2(2, 4); \quad z^{2} = HO2(3, 4);$ = HO3(1,4); y3 = HO3(2,4); z3 = HO3(3,4); x3 = H04(1,4); y4 = H04(2,4); z4 = H04(3,4); x4 = H05(1,4); y5 = H05(2,4); z5 = H05(3,4); x5 x6 = H06(1,4); y6 = H06(2,4); z6 = H06(3,4);% Se dibuja el robot x = [x0 x1 x2 x3 x4 x5 x6];y = [y0 y1 y2 y3 y4 y5 y6]; z = [z0 z1 z2 z3 z4 z5 z6];puntos=[x;y;z]; end

Tabla 3 código de la función Directa6GDL

Cinemática Inversa

El desacoplo cinemático consiste en dividir el sistema en dos problemas (posición y orientación). Para ello, dada una posición y orientación final deseadas, establece las coordenadas del punto de corte de los 3 últimos ejes (muñeca del robot) calculándose los valores de las tres primeras variables articulares ($\theta_1, \theta_2, \theta_3$) que consiguen posicionar este punto. [5] [4]

Se realiza el desacoplo cinemático restando la posición del robot con tamaño de la muñeca por el vector unitario de la orientación del efector final $(l_4 \vec{a}_6)$



Figura 3.-Desacoplo cinemático

Con \vec{p}_3 que representa la posición del desacoplo cinemático se resuelve el problema de esta posición cálculo de las 3 primeras articulaciones $\theta_1, \theta_2, \theta_3$ en esta ocasión se utilizó métodos geométricos [6]. Conocido los 3 primeros ángulos $\theta_1, \theta_2 y \theta_3$ se resuelve la cinemática directa para los 3 primeros eslabones obteniéndose la matriz H_0^3 , necesaria para la revolución de las ultimas 3 articulaciones. Para resolver el problema de la orientación se han conseguido que las 3 últimas articulaciones $(\theta_4, \theta_5 y \theta_6)$ con los criterios siguientes:

1.- Establecer la articulación θ_4 de forma tal que una rotación respecto de la rotación θ_5 alineara el eje de movimiento de la articulación θ_6 con el sector de aproximación dado (a).

2.- la articulación θ_5 alineara el eje de movimiento de la articulación θ_6 con el vector de aproximación.

3.- Fijar la articulación θ_6 para alinear el vector de orientación

La función "Inversa6GDL" te regresa los valores de las articulaciones según la matriz de transformación T la cual contiene la matriz de orientación 3X3 y el vector de posición 3X1.La solución de la cinemática inversa es por el método geométrico también se utiliza el desacoplamiento cinemático para resolver la orientación. En la tabla 4 se puede observar el código de esta función

 t: Matriz de transformación homogénea 4x4 d: tamaño de las articulaciones a: distancias en beta codo: tipo de codo muñeca: tipo de muñeca Function q = Inversa6GDL(T, d, a, codo, muneca)
d: tamaño de las articulaciones a: distancias en beta codo: tipo de codo muñeca: tipo de muñeca Function q = Inversa6GDL(T, d, a, codo, muneca)
a: distancias en beta codo: tipo de codo muñeca: tipo de muñeca Function q = Inversa6GDL(T,d,a,codo,muneca)
codo: tipo de codo muñeca: tipo de muñeca Function q = Inversa6GDL(T,d,a,codo,muneca)
muñeca: tipo de muñeca Function q = Inversa6GDL(T,d,a,codo,muneca)
Function q = Inversa6GDL(T,d,a,codo,muneca)
Function q = Inversa6GDL(T, d, a, codo, muneca)
<pre>% Posicion de la mano del manipulador multipulador</pre>
p = T(1:3,4) - d(6) * T(1:3,3);
% Solucion de la primera articulacion: ql
$R = sqrt(p(1)^{2}+p(2)^{2});$
sql=p(2)/R;
$cq_{1}=p(1)/R;$
<pre>q1 = ataliz(Sq1,Cq1); % Solución do la socurda articulación: q2</pre>
$r = \operatorname{sart}(\mathbb{P}^{2}+(n/3)-d(1))^{2})$
r = 3qrc(R 2 + (p(3) - n(1)) / r;
calfa = R/r .
cheta = $(r^{2}+a(2))^{2}-d(4)^{2}/(2*r*a(2))$:
$sbeta = sgrt(1-cbeta^{2});$
if $codo == -1$ % Codo abajo
<pre>sg2 = salfa*cbeta+sbeta*calfa;</pre>
cg2 = calfa*cbeta-salfa*sbeta;
else % Codo arriba
<pre>sq2 = salfa*cbeta-sbeta*calfa;</pre>
cq2 = calfa*cbeta+salfa*sbeta;
end
q2 = atan2(sq2,cq2);
% Solución de la tercera articulación: q3
cbeta=(a(2)^2+d(4)^2-r^2)/(2*a(2)*d(4));
<pre>sbeta=sqrt(1-cbeta^2);</pre>
<pre>beta=atan2(sbeta, cbeta);</pre>
if codo == 1 % Codo arriba
q3 = 3*pi/2-beta;
else % Codo abajo
$q_3 = beta - p_1/2;$
ena 8 Gelución de la cuenta enticulación, e4
<pre>% Solucion de la cuarta articulacion: q4 % transformación 002</pre>
σ transformation AUS $= MatrigDU(\alpha 1 - d(1) - a(1) - alpha(1))$
H12 = MatrizDH(q2, d(2), a(2), alpha(2)):

H23 = MatrizDH(q3, d(3), a(3), alpha(3)); H03 = H01 * H12 * H23; x3 = H03(1:3,1);y3 = H03(1:3,2);z3 = H03(1:3,3);z4 = cross(z3,T(1:3,3)); % Determinación orientación omega aux = dot(T(1:3,2),z4); if aux ~= 0 omega = aux; else aux=dot(T(1:3,1),z4); if aux ~=0 omega=aux; else omega=0; end end M = muneca*sign(omega); sq4 = -M*dot(z4, x3);cq4 = M*dot(z4, y3);q4 = atan2(sq4, cq4);% Solución de la quinta articulación: q5 z5 = T(1:3,3);H34 = MatrizDH(q4, d(4), a(4), alpha(4)); H04 = H03 * H34; x4 = H04(1:3,1);y4 = H04(1:3,2);sq5 = dot(T(1:3,3), x4);cq5 = -dot(T(1:3,3), y4);q5 = atan2(sq5, cq5);% Solución de la sexta articulación: q6 v6 = T(1:3,2);H45 = MatrizDH(q5, d(5), a(5), alpha(5)); H05 = H04 * H45; y5 = H05(1:3,2);sq6 = dot(T(1:3,1), y5);cq6 = dot(T(1:3,2), y5);q6 = atan2(sq6, cq6);q = [q1 q2 q3 q4 q5 q6]'; end

Tabla 4 código de la función Inversa6GDL

3 DISEÑO EN SOLIDWORKS

Se utilizó el software de SolidWorks para el diseño del Robot, se hicieron sub-ensamblajes de las articulaciones con la finalizar de crear el ensamblaje completo del Robot y de esta manera se pueda crear una simulación del diseño mecánico usando las herramientas del SolidWorks. En la tabla 5 se puede observar los sub-ensamblajes que se crearon.

Nombre	Pieza	Descripción
Base1	Ē	Base fija del robot
Base2		Base giratoria que representa el movimiento de la cadera del robot
Brazo1	() · · ·	Primer eslabón que re presenta el movimiento del hombro del robot
Brazo2	•	Segundo eslabón que representa el movimiento del codo del robot.

Brazo3		Pieza giratoria que representa el movimiento del antebrazo del robot
Миñеса		Pieza que representa el movimiento de la muñeca
Gripper	Mar North	Actuador final que rota sobre su eje

Tabla 5.- piezas del robot en SolidWorks

En la figura 4 se puede ver el ensamblaje final del Robot de 6 GDL.



4 SIMULADOR EN LABVIEW.

El software Labview tiene como ventaja la versatilidad de integración con otros softwares y de la misma manera es muy fácil la implementación con hardware usando tarjetas de adquisición de datos o microcontroladores. Por esta razón se utilizó este programa para desarrollar el simulador del robot y posteriormente unirlo con el prototipo físico.

3.4.1 Graficar en Labview el robot en 3D

Se utilizó las herramientas de "3D picture control" para graficar el robot en un ambiente de 3D, también se utilizaron otras herramientas para crear la SubVI "Robot 3D" que facilita la programación del mimo, en la tabla 6 se explica las funciones utilizadas para realizar la SubVI.

Nombre de VI	Figura	Descripción
Robot 3D	ROBOT 3D	Función que grafica el robot en 3D
First call?	٨	Indica que una subVI o sección de un diagrama de bloques se ejecuta por primera vez
Path	B	Devuelve la ruta del archivo de la VI actual
Strip Path	2 <mark>0 ₽8</mark>	Devuelve el último nombre del componente de la ruta de acceso y el camino despojado que conduce a ese componente.
Build Path	<mark>≫</mark> ∎⊷ ∰⊷	Crea una nueva ruta añadiendo un nombre a una ruta existente.
Load VRML File VI	wei	Carga un modelo de objeto tridimensional guardado en formato Virtual Reality Modeling Language (VRML) para mostrar en una escena 3D. Los archivos VRML aparecen con una

		extension de .wrl.	
Create Object VI		Crea un nuevo objeto 3D para mostrar en la escena 3D.	
Invoke Node (Add Object)	Preventer SceneObject Preventer Pre	Agrega el objeto a la escena	
Setup Camera Method	Setup Camera Setup Camera Camera Postion Target Up Direction	Especifica la posición de la cámara con respecto a la escena.	
3D Picture Control	3D Picture	Venta de visualización de la escena de 3D	
Find Object VI	0	De la escena 3D busca el objeto con el nombre de referencia y devuelve el objeto en la escena con un nombre coincidente.	
Clear Transformation VI	* ⊲ √	Borra las transformaciones aplicadas previamente a un objeto en una escena 3D	
Degree to radians	degrees to radians (x/180.0)*pi	Cambia el ángulo de grados a radianes	
Rotate Object VI		Gira un objeto de una escena 3D por los valor del eje y el ángulo. Este VI realiza una rotació relativa de la posición actual del objeto.	
Translate Object VI		Traslada un objeto de una escena 3D por el vector que se especifique. Este VI realiza una traslación relativa de la posición actual del objeto.	

Tabla 6 Funciones para crear la SubvI "Robot 3D"

El programa de la SubVI Robot 3D se encuentra dentro una estructura case, la cual tiene dos estados cuando es verdadero o cuando es falso, se utilizó la función llamada "First call?" como se explica en la tabla 6 esta función es verdadera cuando se ejecuta por primera vez. Cuando se ejecuta por primera vez la función Robot 3D se abren todos los archivos con formato "WRL" creados en SolidWorks y posteriormente se hace el ensamblaje esta sección del programa se puede ver en la figura 5.

En la Figura 6 se puede ver el programa de la estructura case cuando es falso en esta sección de programa se utiliza para crear las rotaciones del robot usando los controladores numéricos de los 6GDL. En la figura 7 se observa la gráfica del robot en 3D



Figura 6 estructura case falso de la SubVI Robot 3D



Figura 7 grafica del robot en 3D

SubVI de incrementos para trayectoria lineal.

La SubVI incrementos vista en la figura 8, es utilizada crear una trayectoria lineal con un numero de n pasos, esta función tiene como parámetro de entrada las 6 articulaciones así como la posición a la que se desea mover y también el número de pasos para llegar a esta posición y da como resultado los incrementos de posición X, Y,Z y orientación R,P,Y. también da como resultado la posición actual del robot y la Matriz de transformación Homogénea



Figura 9 Código de la SibVI incrementos

SubVI Cinemática inversa

Se creó una SubVI de la Cinemática Inversa vista en la figura10, esta SubVI, llama las funciones creadas en Matlab explicadas en el punto 3.3. El código de la figura 11 tiene como parámetros de entrada la Matriz de Transformación Homogénea y los incrementos para llegar a la posición deseada, esta función da como resultado las articulaciones, la matriz homogénea y la posición de<u>l robot</u>.



Figura 10 SubVI Inversa



Figura 11 Código de la SubVI inversa

SubVI Movimientos.

Se creó una función que realiza los dos movimientos del robot World y Tool, la función tiene como parámetro de entrada la velocidad de movimiento, la Matriz Homogénea, el tipo de movimiento, la posición u orientación a mover y el incremento para llegar al punto; se utilizó las funciones creadas en Matlab para realizar el movimiento. Esta función da como resultado la nueva posición y orientación así como la matriz Homogénea, en la figura 12 se puede observar el código de esta función.



Figura 12 Código de la SubVI Movimiento

5 RESULTADOS

En la figura 13 se muestra la interfaz del robot, este diseño se creó en el panel frontal de LabView. Con la interfaz el usuario

puede interactuar con el simulador del Robot con sus controladores e indicadores, los indicadores más importantes del simulador son las Gráfica en 3D y la Gráfica en hilo, estos visualizadores muestran la articulación del robot, en la Gráfica en 3D se visualiza el robot con las piezas creadas en SolidWorks. También tiene como indicadores la posición actual del Robot y los pasos para la secuencia.

Los controladores más importantes son las perillas que simula el movimiento en Joint, los botones que simulan los movimientos en World yTool entre otros.



Figura 13 Interfaz del robot.

6 CONCLUSIONES

Con este proyecto se pudo comprobar la aplicación del LabView para el desarrollo del robot de 6GDL utilizando las piezas creadas en SolidWorks. Cabe destacar que, al utilizar desacoplamiento cinemático simplifico la solución de la cinemática inversa para encontrar el valor de las articulaciones de una coordenada especificada, sin embargo, es importante saber que existen también otros métodos como lo son los métodos numéricos, inteligencia artificial así como algebra de cuaterniones, y que conllevan a los mismos resultados al buscar solución, por esta razón se propone para futuras investigaciones y aplicaciones del desarrollo del Robots, se aplicará otros métodos anteriormente mencionados.

Se pudo comprobar el uso de un teach pendant (controlador deposiciones de articulaciones y traslación de herramienta) virtual a través del software de análisis experimental LabView, el cual permitió manipular al robot articulado de 6 GDL de manera directa en cada una de sus articulaciones, o inversa trasladando la punta de la herramienta desde un punto inicial hasta el solicitado, resolviendo la trayectoria del movimiento. El uso de SolidWorks fue de mucha utilidad para la creación de los archivos con formato "WRL", que básicamente este formato es fácil de utilizar para graficar el simulador con objetos 3D

7 REFERENCIAS

- [1] J. E. P. Ruiz, Modelación Cinematico del robot CRS A465 Utilizando el algebra de quaterniones, Mexico D.F.: Universidad Nacional Autónoma de México, 2006.
- [2] J. S. M. Erazo, SIMULACIÓN DE UN ROBOT CON 6 GRADOS DE LIBERTAD UTILIZANDO TOOLBOX ROBOTICS DEL SOFTWARE MATLAB, ESCUELA POLITÉCNICA NACIONAL, 2020.
- [3] Ignacio Vázquez Cuevas, «Modelación y Diseño de un Simulador de un Robot Paralelo Manejado por un Controlador Manual Didáctico,» Congreso Nacional de Mecatrónica, vol. 8, pp. 329-324, 2009.
- [4] A. Barrientos, Fundamentos de Robótica, Madrid España: McGraw- HILL, 1997.
- [5] J. J. Graic, Robótica, Mexico DF: PEARSON, 2006.
- [6] F. R. CORTÉS, MATLAB APLICADO A ROBÓTICA Y MECATRÓNICA, MEXICO DF: ALFAOMEGA, 2012.
- [7] R. G. Andrade, «Manufactura de prototipo de robot de 6 GDL,» Número Especial de la Revista Aristas: Investigacion Básica y Aplicada, vol. 6, nº 12, pp. 175-181, 2018.
- [8] W. S. B. Escobedo, «Modelado y simulación de un manipulador de seis grados de libertad para aplicaciones industriales,» *Pueblo Cont.*, vol. 24, nº 1, pp. 15-27, 2013.
- [9] A. Hossian, «El problema cinemático en manipuladores robóticos industriales un abordaje en solucion mediante redes neuronales artificiales,» VI Jornadas liberoamericanas de Ingeniería del Software e Ingenieria del conocimiento, pp. 427-433.
- [10] M. V. G. ORAMAS, MODELACIÓN Y ANÁLISIS DE LA CINEMÁTICA DIRECTA E INVERSA DEL MANIPULADOR STANFORD DE SEIS GRADOS DE LIBERTAD, ESCUELA POLITÉCNICA NACIONAL, 2014.
- [11] J. V. Tovar, «Simulación de trayectorias del robot LR MATE 200iB mediante Matlab,» *CIINDET*, pp. 614-620, 2010.
- [12] A. R. Ramos, SIMULACIÓN DE UN BRAZO ROBÓTICO EN SIMMECHANICS, Madrid: Universidad Carlos III de madrid, 2015.