

Diseño y Análisis de Controladores para un robot de 3GDL con capacidad de dibujo

J. Albites, M. Rimachi y J. Paucar . MSc. Ing. N. Anchayhua

Abstract

El presente trabajo muestra el diseño de controladores para un brazo robótico de 3 grados de libertad (GDL). En el presente trabajo se desarrolla un análisis de 3 tipos de controladores, controladores clásicos tipo PID, controladores por Torque y finalmente un controlador Fuzzy basado en 9 reglas.

Términos clave

Controladores, PID, Torque Computado, Lógica Fuzzy

CONTENIDO

I Introducción	2
II Modelo del Robot	2
III Controladores	5
3.1 Control PID	5
3.2 Control por Torque Computado	5
3.3 Control Fuzzy para Sintonización PID.....	6
IV Simulación	7
V Resultados 3	
5.1 Control PID	7
5.2 Control por Torque Computado	9
5.3 Control Fuzzy para Sintonización PID.....	10
VI Observaciones	11
VII Conclusiones	11
VIII Recomendaciones	11
IX Referencias	12

I. INTRODUCTION

Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas.

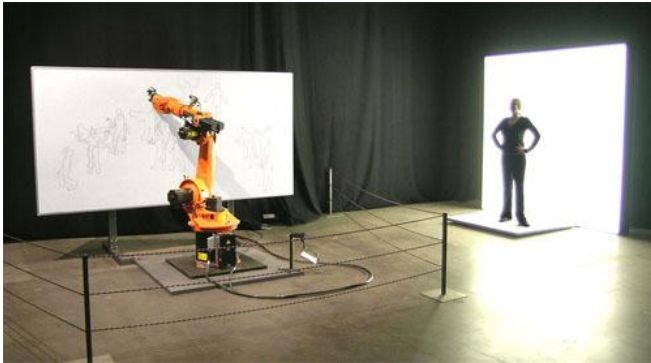


Fig. 1 Ejemplo de robot dibujante

Un robot se compone principalmente de tres partes: parte mecánica (forma y tamaño de la carcasa), parte electrónica (compuesta por el circuito de control y la sensoria del robot) y la parte de potencia (control de motores del robot). La carcasa constituye la parte mecánica, la forma de esta es uno de los factores determinantes en el éxito del desarrollo de una determinada tarea. La parte electrónica esta constituida por el circuito de control que a su vez contiene un microcontrolador el cual se programa empleando un determinado algoritmo, en dicho algoritmo radica todas las decisiones que va a efectuar el robot ante determinados casos dados de acuerdo al problema.

II. MODELO DEL ROBOT

La estructura a utilizar se basa en un robot PUMA de 3grados de libertad (GDL) rotacionales, de modo que el modelo dinámico del robot resulte más sencillo y manipulable para efectuar el control y se evite redundancia en la medida de lo posible.

Para el diseño del robot se utilizó Solidworks como se puede apreciar en la figura 2.

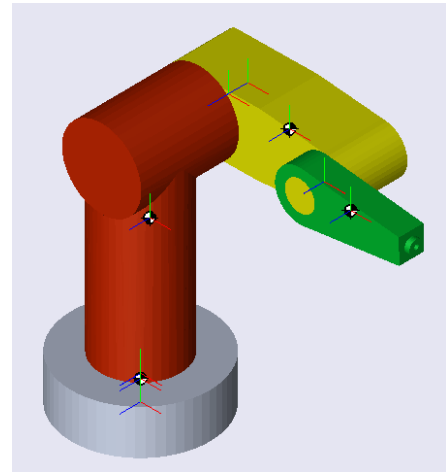


Fig. 2 Diseño CAD Robot 3GDL

IIA. MODELO DE LA CINEMÁTICA

Mediante las matrices de transformación Denavit Hertember, obtenemos las expresiones de la cinemática Directa e Inversa. En la siguiente imagen se muestra los ejes utilizados en dicho cálculo:

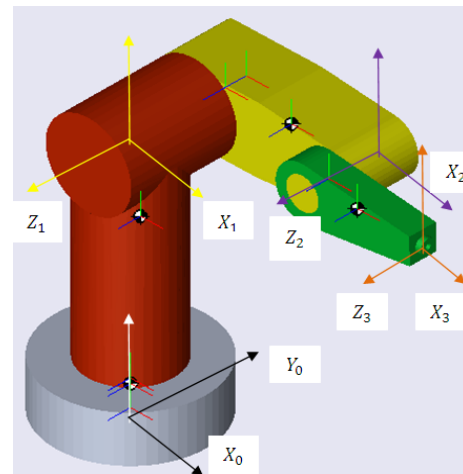


Fig.3 Cinemática Directa DH

$$X = l_2 \cos(Q_1) \cos(Q_2) - l_3 \sin(Q_1) + l_4 \cos(Q_1) \cos(Q_2 + Q_3) + l_5 \sin(Q_1)$$

$$Y = l_2 \sin(Q_1) \cos(Q_2) + l_3 \cos(Q_1) + l_4 \sin(Q_1) \cos(Q_2 + Q_3) - l_5 \cos(Q_1)$$

$$Z = l_1 + l_2 \sin(Q_2) + l_4 \sin(Q_2 + Q_3)$$

Cinemática Inversa

La obtención de la Cinemática Inversa, se realizó tomando como referencia, los cálculos realizados al manipulador de 3 GDL que se encuentra en el texto expuesto en el anexo.

Una mejor opción es hallar dicha dinámica, desde un aspecto geométrico.

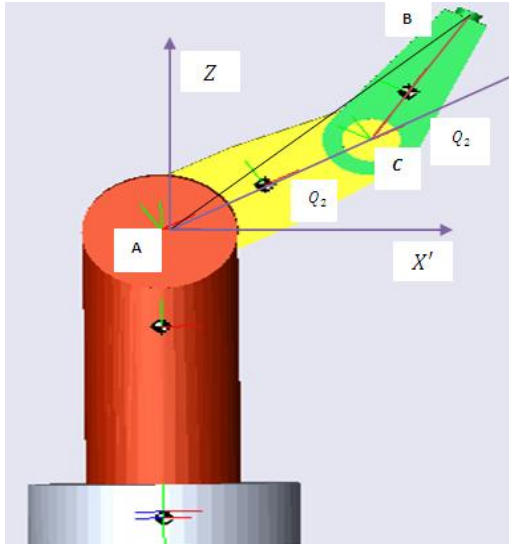


Fig. 4 Lateral Derecho

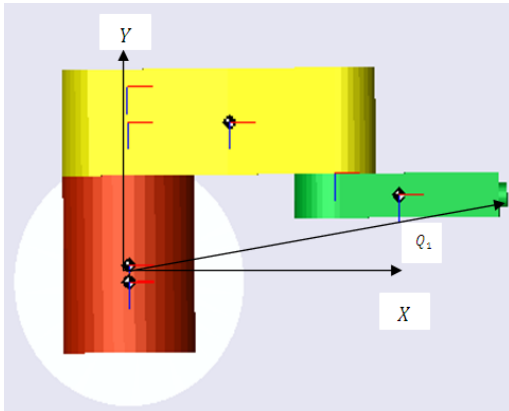


Fig. 5 Vista Superior

La cinemática Inversa de nuestro robot-Puma tiene la siguiente forma:

$$Q_3 = \tan^{-1} \left(\frac{\sqrt{1-D^2}}{D} \right)$$

$$Q_2 = \tan^{-1} \left(\frac{Z - l_1}{\sqrt{X^2 + Y^2 - (l_3 - l_5)^2}} \right) - \tan^{-1} \left(\frac{\sqrt{1-d^2}}{d} \right)$$

$$Q_1 = \tan^{-1} \left(\frac{Y}{X} \right) - \tan^{-1}(e);$$

En donde las variables D, d, e son expresiones que dependen de la geometría del diseño y son usadas como términos auxiliares de cálculo.

Se expone una prueba de los cálculos realizados en la comprobación de la dinámica.

```
>> CDpuma([pi/4, pi/6, pi/6])
ans =
    0.0860
    0.1390
    0.3496
>> CIpuma([0.086 0.139, 0.3496])
ans =
    0.7853
    0.5234
    0.5243
```

Probando en la simulación del robot, ingresando una posición determinada en los bloques de simulink.

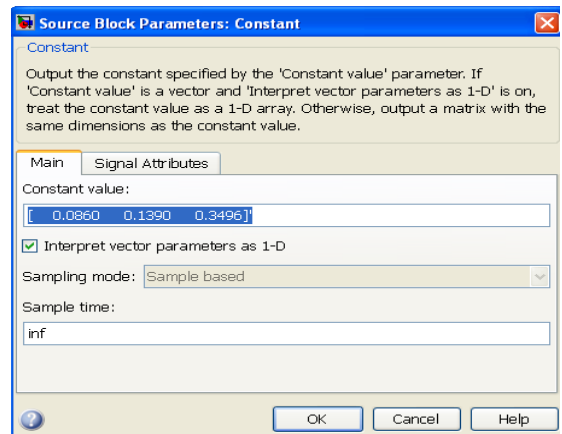


Fig. 6 Introducción de posición referencial

En la figura 7 se muestra la respuesta en posición del robot; con el cual se verifica la cinemática Directa, así como, la Inversa.

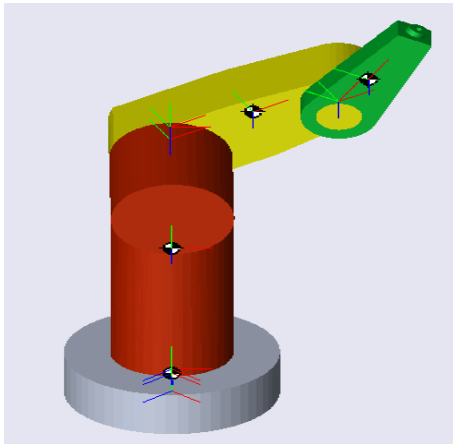


Fig. 7 Posición robot de acuerdo a la referencia

En las siguientes imágenes se observa los resultados del efector final, en las 3 posiciones X, Y, Z.

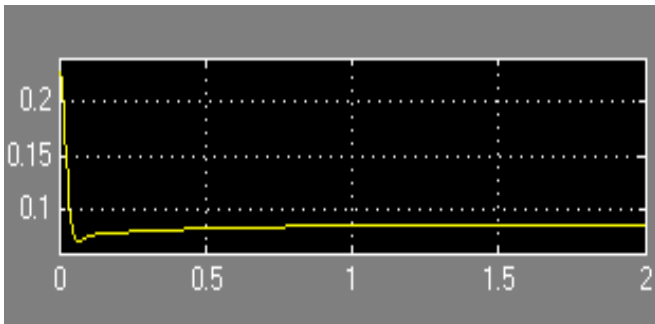


Fig. 8 Posición final del robot en el eje X.

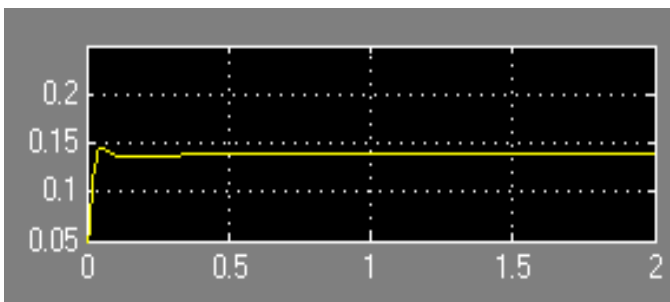


Fig.9 Posición final del robot en el eje Y.

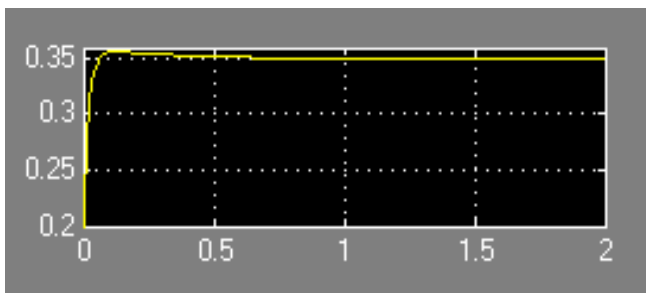


Fig.10 Posición final del robot en el eje Z.

IIB. MODELO DE LA DINÁMICA

La estructura a utilizar se basa en un robot PUMA de 3grados de libertad (GDL) rotacionales, de

El modelo del robot se obtuvo de 2 maneras, utilizando la aplicación Simechanics del Matlab, y a través del modelo matemático de Lagrange. Asimismo en fases posteriores se analizó el método de Euler y el Algoritmo Luh-Walker para el control por Torque Calculado como se verá posteriormente.

2.1 Modelo del Robot utilizando Simechanics

Se obtuvo un modelo a través de la herramienta de cálculo MATLAB, trabajando sobre el diseño mecánico hecho mediante Solidworks, obteniendo el siguiente diagrama, el cual considera cada una de las juntas del robot (véase Fig. B).

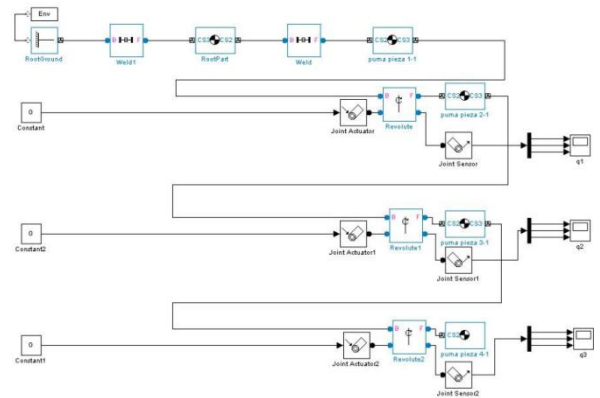


Fig. 11 Modelo Robot obtenido con SimMechanics

Este modelo, pese a diferir del modelo hecho únicamente con Simulink en ciertas no linealidades, como por ejemplo la fricción, nos ha de proporcionar resultados semejantes para verificar la consistencia del modelo realizado.

2.2 Modelo matemático

Para obtener la dinámica del robot se aplicó el Método de Lagrange. Partimos directamente de la ecuación general que relaciona el torque de control con el movimiento del robot.

$$\tau = H(q) * \ddot{q} + c(q, \dot{q}) + G(q) + b(\dot{q}) \dots (1)$$

Donde:

τ : torque control actuadores

$H(q)$: matriz inercia robot

$c(q, \dot{q})$: fuerza centrífuga y coriolis

$G(q)$: fuerza gravedad

$b(\dot{q})$: fuerza rozamiento, en nuestro caso se depreciará.

Los cálculos de las matrices H, c y G, fueron obtenidos de acuerdo a procedimientos generales explicados en diversos libros de robótica. A continuación se presenta el esquema en simulink de la ecuación presentada arriba.

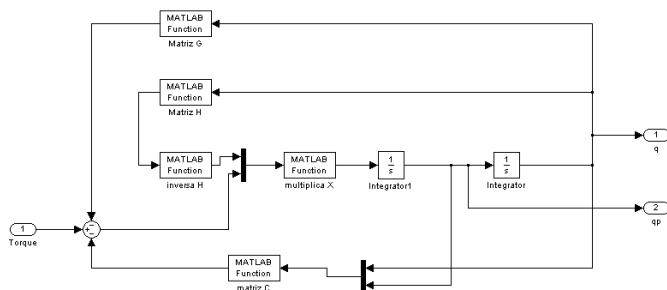


Fig. 12 Modelo robot en Matlab - Simulink

III. CONTROLADORES

La elección del controlador para un robot es una decisión crucial en el funcionamiento de éste y el éxito que el robot pueda alcanzar en una determinada tarea dependerá básicamente de la eficiencia y eficacia que tenga el controlador.

A continuación se realizará el análisis de 3 algoritmos de control; un controlador PID, un control por Torque Calculado y un Controlador Fuzzy para sintonización PID.

3.1 Control PID

Se probaron diversos diseños de control de este tipo, partiendo desde sólo un control proporcional, hasta el que se muestra a continuación que es un PID con compensación de gravedad.

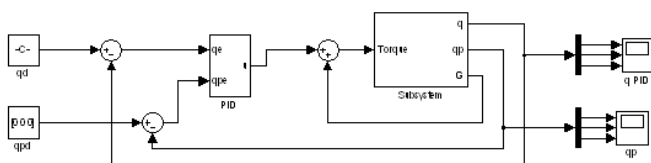


Fig.13 Diagrama Simulink Control PID

El diagrama sigue el siguiente algoritmo de control:

$$\tau = k_p \tilde{q}_e + k_d \dot{\tilde{q}}_e + k_i \int \tilde{q}_e + G(q) \dots (2)$$

La ecuación arriba mostrada es la típica ecuación de un PID, las constantes se sintonizaron utilizando base teórica también encontrada en diversos libros especificados en la bibliografía. Se debe notar que adicionalmente se ha realizado una compensación por gravedad, esto es necesario para evitar cambios bruscos en el estado transitorio sobre todo al inicio por efecto de la gravedad. Más adelante en los resultados se verá la diferencia entre un control normal y uno con compensación de gravedad.

Control de Posición:

Como se puede observar, el modelo anterior solo controla el robot pero en base al espacio Q, en la práctica lo que realmente se necesita es la control directa de la posición a través del espacio XYZ, para ello se necesitará hacer conversiones de un espacio a otro, por lo que hará uso de la cinemática directa e inversa del robot así como del jacobiano directo y inverso del robot. A continuación se presenta el diagrama en simulink utilizado para el control de posición de nuestro robot.

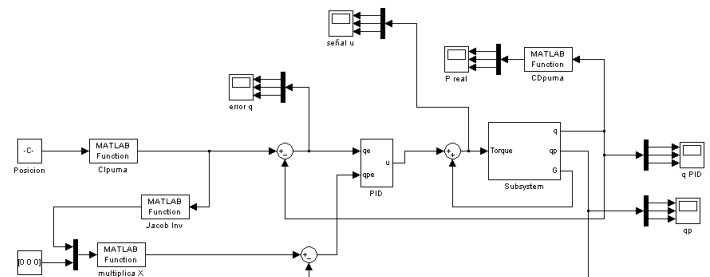


Fig. 14 Diagrama Control de Posición del Robot

3.2 Control por Torque Calculado

Este tipo de control tiene como principal característica la estimación de las propiedades dinámicas del robot, de modo que se intenta compensar el efecto de cada una con un control más preciso; el presente trabajo utiliza el algoritmo de Luh Walker Paul, el cual consiste en una serie de cálculos vectoriales cinemáticos a partir de los datos de posición y velocidad tanto lineales como angulares del elemento base (en este caso para el manipulador todos son

cero), desde allí se calculan siguiendo acorde a la configuración del robot y las posiciones y velocidades articulares las posiciones y velocidades tanto lineales como angulares de los siguientes elementos, calculando también los mismos valores para los centros de masa de cada eslabón del manipulador (véase Fig. 15).

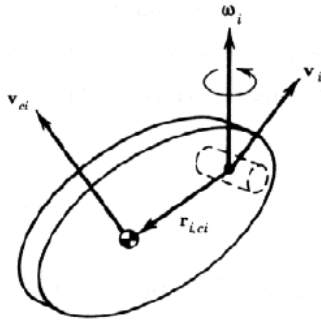


Fig. 15 Representación genérica de las velocidades lineales y angulares en un eslabón

Con estos datos se hace después un cálculo “hacia atrás” partiendo de los datos de fuerza y torque aplicados al efector final (para efectos de cálculo en las simulaciones asumimos que son cero) y a partir de las propiedades geométricas y físicas (masa y momento de inercia) se calculan las fuerzas y torques en cada junta hasta dar con el elemento cero habiendo calculado ya todos los anteriores (se emplea las ecuaciones 3 y 4); el último paso de este algoritmo es el cálculo de los torques que se aplican en cada uno de los actuadores a partir de los valores calculados de fuerza y torque en cada junta.

$$f_{i-1,i} = f_{i,i+1} - m_i g + m_i a_{ci} \dots (3)$$

$$N_{i-1,i} = N_{i,i+1} - r_{i,ci} \times f_{i,i+1} + r_{i-1,ci} \times f_{i-1,i} + I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i) \dots (4)$$

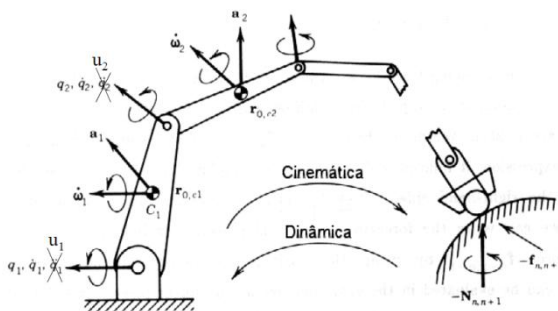


Fig. 16 Algoritmo Luh – Walker para torque calculado

3.3 Control Fuzzy para sintonización PID

La lógica difusa es una herramienta de la inteligencia artificial que en los años recientes viene siendo utilizada en diversos sistemas de control, pero veamos brevemente en que se basa este tipo de control.

La lógica difusa inicia con el concepto de conjunto Fuzzy. A diferencia de un conjunto clásico, en un conjunto Fuzzy no todos sus elementos están totalmente definidos, es decir hay una cierta probabilidad manejada por una función de membresía que indicará en qué medida una variable pertenecerá a un conjunto o no.

Pero si este concepto parece demasiado superficial y difícil de entender, veámoslo con un ejemplo muy conocido, si queremos listar el conjunto dado por los días que corresponden a fines de semana, muchos diremos que sábado y domingo pertenecerán a este conjunto, pero quizá algunos consideraremos que viernes también pertenece a este conjunto, es aquí cuando se define un conjunto Fuzzy y es la clave del concepto anterior.

De forma general un sistema de control con lógica difusa que presenta las siguientes partes:

- Interface de fuzzyficación: convierte los valores numéricos de entrada (números reales) en conjuntos difusos.
- Aplicación de las Operaciones difusas: en esta fase se analizarán y ejecutarán las reglas de producción.
- Aplicando Método Implicación: Con la operación ya obtenida en el paso anterior se truncará la función de membresía de salida, generalmente triangular.
- Acoplado Salidas: Debido a que las decisiones son tomadas por una serie de reglas, las salidas de cada regla deben ser sumadas para luego poder obtener una decisión final.
- Interface de Defuzzificación: En esta fase final se transformará las variables lingüísticas de la salida en valores numéricos. El método más conocido es el método del cálculo del centroide de la curva final obtenida del penúltimo paso.

En la figura 18, se muestra un esquema general obtenido de la ayuda del Matlab con todos los pasos antes descritos.

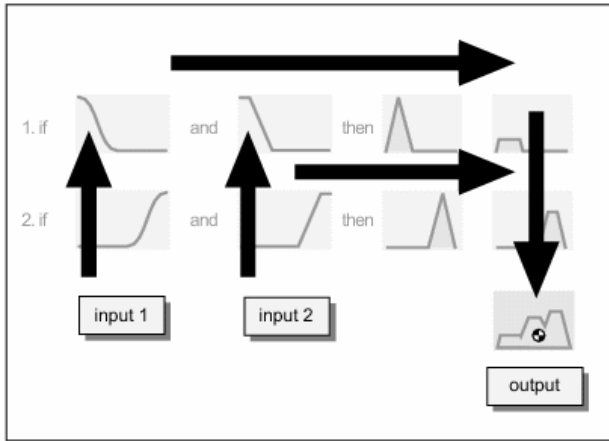


Fig.17 Diagrama de un Sistema Inferencia Fuzzy (FIS)

Se puede encontrar información más detallada sobre este control en la bibliografía.

Tenemos que decir que el control Fuzzy generalmente viene acompañado de otro tipo de función que ayudará a obtener una optimización de las reglas para el control.

En este informe, sólo se aplicarán reglas Fuzzy básicas, hechas por nuestro conocimiento en la sintonización de sistemas PID, con estas reglas podremos obtener las constantes del control PID dependiendo del error de nuestra planta.

IV. SIMULACIÓN

Para la simulación se desarrollaron modelos en Simulink para cada tipo de control utilizado.

Se ha implementado para este caso en simulación un controlador PID junto con un bloque para el “torque calculado”.

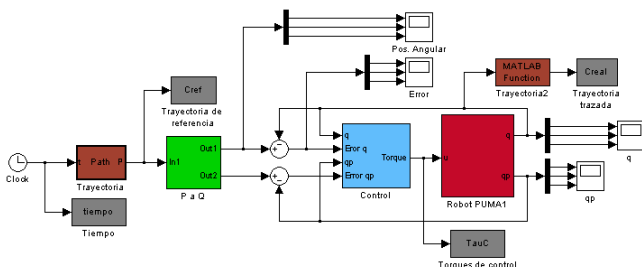


Fig. 18 Control por torque calculado utilizando un controlador PID

En la Figura 19 se muestra el diagrama en Simulink para el Controlador Fuzzy. Se diseñó el controlador sólo para la última articulación, este porque esta

articulación es la que está en contacto directo con el efector final.

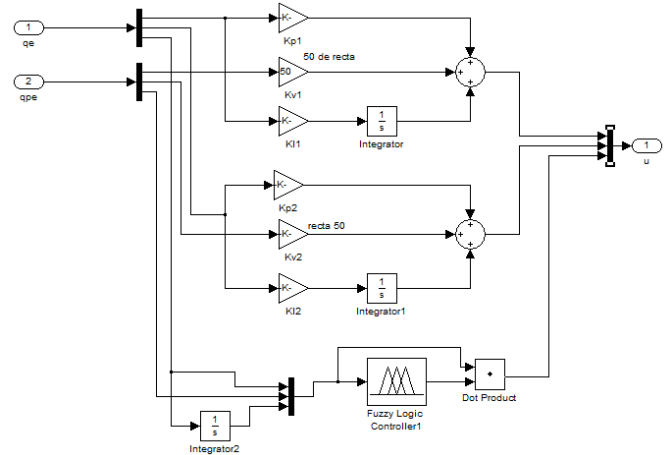


Fig.19 Control Fuzzy para Sintonización PID

V. RESULTADOS

A continuación se presentan resultados obtenidos al realizar los diversos tipos de control en el manipulador.

5.1 Control PID

Se van a mostrar en los valores del error con cada coordenada q el torque de control aplicado a cada junta y la posición en XYZ, la eficacia de los resultados se basa en los valores asignados a los controladores Kp, Kv y Ki:

Se ha probado por ejemplo la posición siguiente:

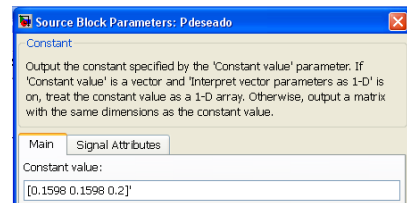


Fig. 20 Posición referencial en XYZ

Las curva de posición de “x”, “y” y “z” respecto del tiempo son como siguen:

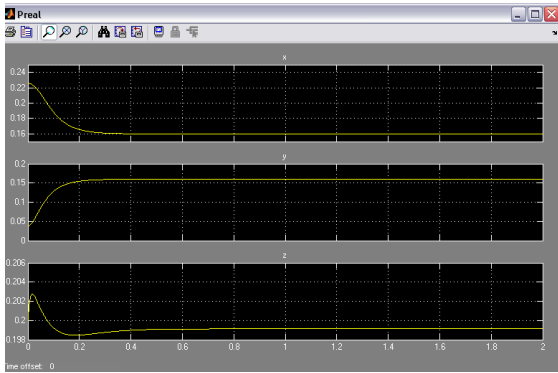


Fig. 21 Posición final del robot

Los valores tienden a los deseados como se aprecia en la figura, manteniéndolo en el valor correspondiente (de manera estable).

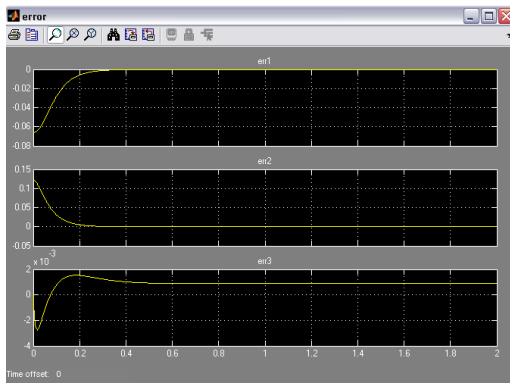


Fig. 22 Errores en XYZ.

En los errores de posición evidentemente se tiene el mismo efecto; sin embargo, esta gráfica nos muestra con más detalle la efectividad del control implementado en la simulación. Como se podrá ver, el error en q_1 y q_2 se establece en 0,

mientras que el tercero se establece en un valor constante muy pequeño (orden de las milésimas).

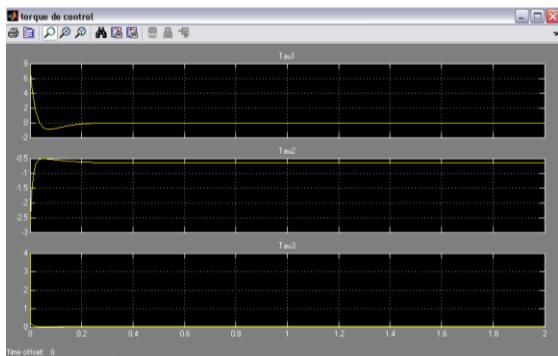


Fig. 23 Posiciones en espacio articular

El torque de control se queda establecido en un valor constante, en el caso de q_1 es 0 dado que no es influyente el valor de Tau_1 para mantener al robot en la posición deseada, en el caso de q_2 se establece en un valor de torque constante para ese fin, mientras que q_3 atraviesa por un pico grande para vencer la inercia al iniciar el movimiento y luego se reduce a un valor que bordea al 0.

Comparamos los resultados obtenidos con el modelo de control PID en el espacio Q, es decir sólo nos preocuparemos de los valores de las coordenadas articulares q_1 , q_2 y q_3 .

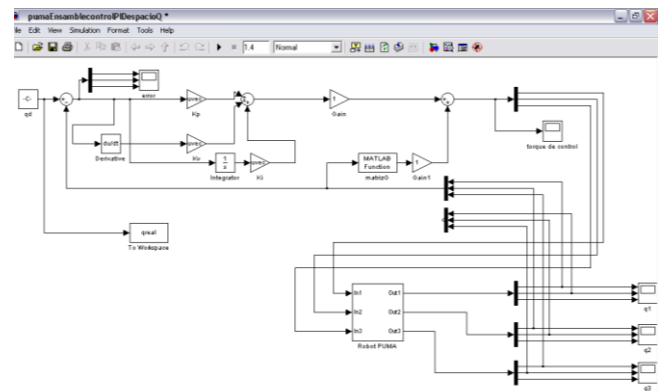


Fig. 24 Diagrama Simulink PID.

Para este modelo también se aplicaron los valores adecuados para los controladores K_p , K_v y K_i

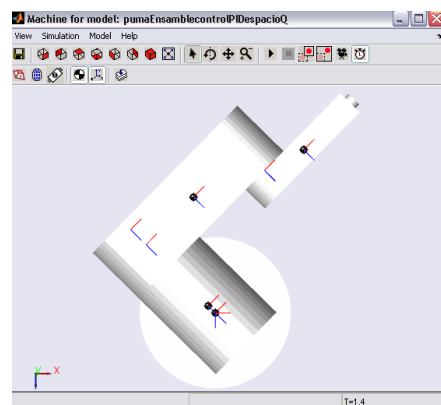


Fig. 25 Animación robot con Simechanics-Simulink

Los errores de posición son los que se muestran, como se podrá ver, con el error de posición 3 ocurre el mismo efecto que con el modelo anterior:

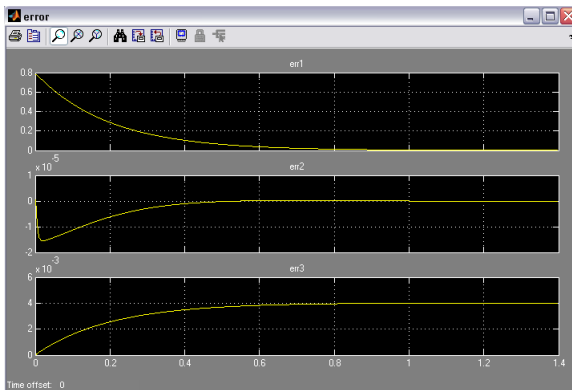


Fig. 26 Errores de posición en los 3 ejes.

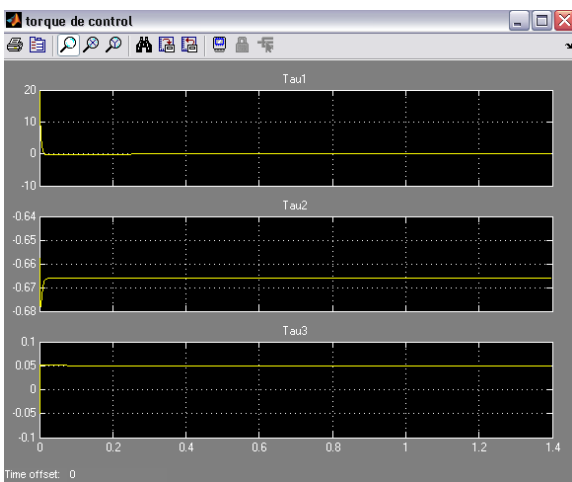


Fig. 27 Torque de control a la entrada del robot.

Esta gráfica 8 muestra los mismos valores de torque de control en estado estable que el modelo anterior lo que nos permite verificar dichos resultados, particularmente el Tau3 se muestra más a detalle.

5.2 Control por torque Calculado

Se ha efectuado pruebas del controlador asignándole tareas según el objetivo de este trabajo, es decir, dado que el manipulador realizará trazos o escritura sobre una superficie, se le ha designado para las pruebas una curva seno ambas sobre un plano paralelo al plano YZ (el robot está situado encima del plano X-Y con el eje del elemento "1" perpendicular a él, véase Fig. 2), para dicha curva el tiempo de simulación ha sido de 10 segundos, obtuvimos las siguiente trayectorias (la trayectoria de referencia es la curva

roja y la trayectoria real trazada corresponde a la curva azul:

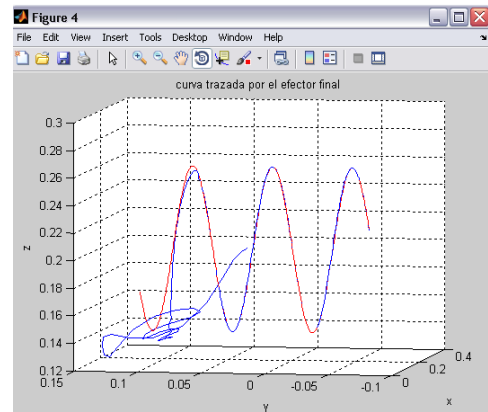


Fig. 28 Seguimiento de la trayectoria (curva seno)

Se muestran para este caso los errores de posición (cartesiana):

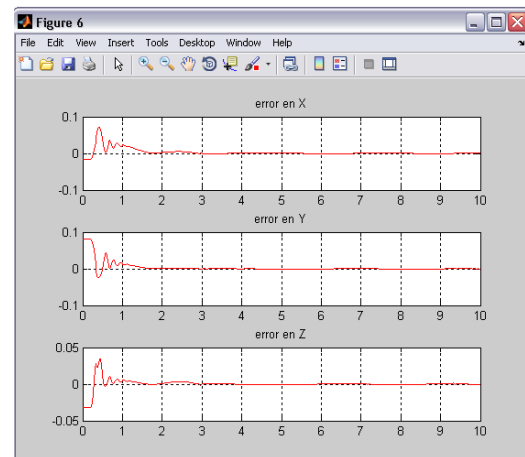


Fig. 29 Errores de posición en los ejes cartesianos

Se muestra que los errores en los 3 ejes coordenados se establecen en cero en un tiempo corto (no más de 2 segundos), pasando antes por una etapa transitoria en la cual se presenta una fluctuación levemente apreciable, esto se da debido a que la posición inicial del manipulador está a una cierta distancia del primer punto de la trayectoria de referencia, luego de ubicarse dentro de la trayectoria asignada, la sigue con el mejor esfuerzo (según el desempeño determinado por los parámetros del controlador).

De entre los resultados también graficamos los torques de control aplicados a cada actuador, se presenta una situación similar a los errores cartesianos de posición:

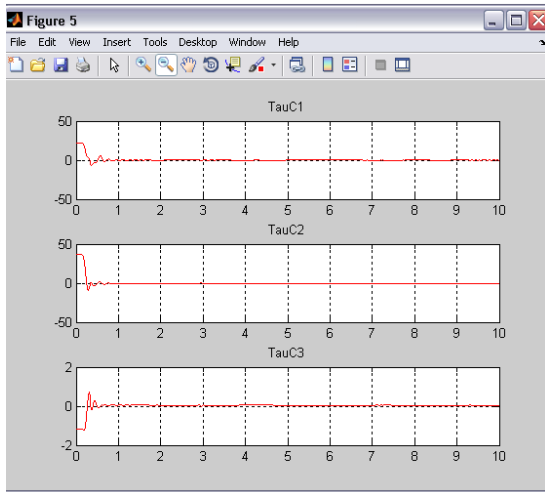


Fig. 30 Torques de control aplicados a cada articulación

Éstos torques de control también presentan una etapa transitoria, la diferenciación con el caso de los errores de posición es que cada torque comienza con un valor relativamente alto respecto de los demás valores a lo largo del tiempo (véase los torques de control “TauC2” y “TauC3” de la Fig. 1), la explicación reside básicamente en la 1era ley de Newton, para que toda la estructura se mueva y la posición del efector final se incorpore a la trayectoria tiene que impulsarse cada actuador con la fuerza necesaria para sacar al manipulador de su estado de reposo y esto es lo que da lugar, dado que se trata de grados de libertad rotacionales, a “torques de arranque” los cuales inician el movimiento y luego se atenúan estableciéndose en una cierta magnitud, la mínima necesaria para que la estructura se mantenga donde le corresponda estar (siguiendo la trayectoria deseada).

En la etapa de control del manipulador se ha tenido que ajustar los parámetros K_p , K_i y K_v para cada articulación hasta que finalmente se obtiene una combinación adecuada (no óptima) y lo suficientemente efectiva para que el robot trace la circunferencia lo mejor posible. Esto también puede lograrse y de modo más eficiente utilizando los principios de la inteligencia artificial, es decir, mediante un conjunto de criterios o “reglas” que se establecen previo conocimiento de los efectos de un cierto rango de valores de los parámetros del controlador PID sobre el error de posición, determinan en todo momento los valores óptimos de dichos parámetros en base a ellas.

5.3 Control Fuzzy Sintonización PID

En la fig. 31 se muestran en primer lugar las reglas obtenidas para el control Fuzzy, como se mencionó

antes, lo ideal sería tener un algoritmo de optimización para estas reglas, pero como base dejamos estas reglas para la sintonización PID.

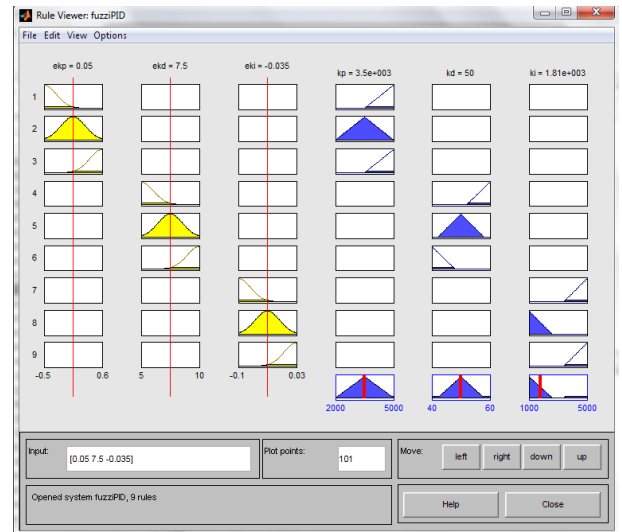


Fig.31 Reglas Fuzzy para el sistema de control

Los resultados obtenidos de la simulación con estas reglas se muestran a continuación:

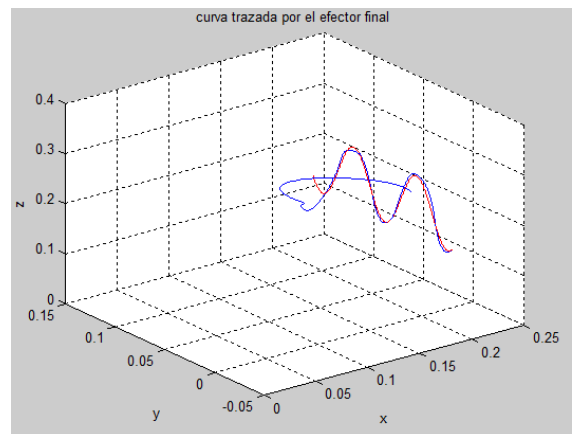


Fig. 32 Curva trazada por el efector final.

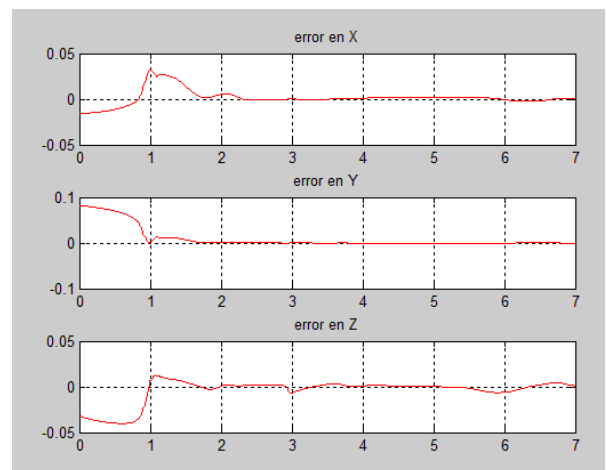


Fig. 33 Errores obtenidos en el efector final.

En la Fig. 33 la curva de color roja muestra la trayectoria de referencia, mientras que la curva azul muestra la trayectoria real trazada por el efector final del robot. Se puede observar que debido a las condiciones iniciales, el robot tarda un poco en alcanzar la referencia, pero una vez alcanzada no tiene problemas en seguirla realizando.

VI. OBSERVACIONES

- Una forma para verificar la dinámica del robot es quitar la actuación de torques, es decir la caída por efecto de la gravedad.
- Se utilizaron solo 9 reglas fuzzy, y se observó que el procesamiento era más tedioso, y por ende la simulación mucho más lenta.
- Para la obtención de los rangos de las variables de entrada en las reglas fuzzy, se utilizaron valores ya conocidos de los errores obtenidos con simulaciones anteriores.
- El control Fuzzy se diseñó no pensado en reemplazar al PID, sino como una alternativa para sintonizar las constantes PID, y de esta forma generar constantes K_p , K_d y K_i ; que variarán dependiendo las condiciones del robot.

VII. CONCLUSIONES

- El sistema no puede ser controlado de forma sencilla dado que la dinámica no lo permite, e porque la matriz de inercias se encuentra bastante acoplada, resultando no ser de diagonal dominante, lo cual nos indica que el efecto de un eslabón afecta a los otros.
- Como conclusión de lo anterior, al intentar controlar uno de los eslabones, esto influirá en los otros, es aquí donde radica la importancia del controlador para cada articulación.
- Aún para el robot analizado de tan sólo 3 GDL, el cálculo y sintonización de las constantes K_p , K_d , K_i para cada articulación resultó demasiado tedioso, debido a esto es que se busca optimizar el ajuste de dichos parámetros utilizando métodos alternativos.

- Este tipo de control resulta muy efectivo dado que a través de la estimación de los torques a partir de las propiedades del robot así como su configuración evitamos errores no deseados (se estiman estas propiedades para que «lo real alcance la referencia»).
- A pesar de esto, por el hecho de aplicarse un controlador PID existe un «conflicto» entre la adecuada elección de los parámetros del mismo y el error permisible así como también se toma en cuenta la respuesta transitoria del robot.
- El control Fuzzy, optimizó la señal de control, pues al variar las constantes del PID en medida del incremento o disminución del error, optimizamos la señal de control u , pues no siempre se tuvo que tener unas constantes altas cuando no se requería.
- La simulación resultó lenta debido a la incorporación del toolbox Fuzzy, y esto debido al alto requerimiento de procesamiento para el análisis de reglas y fuzzificación y defuzzificación de las variables.

VIII. RECOMENDACIONES

- Si se desea sólo usar un controlador PID, se recomienda utilizar un algoritmo optimizador para el cálculo de las constantes K_p , K_i y K_d ; para así evitar la tarea tediosa e ineficiente de la obtención de estos parámetros.
- Se debe siempre tener en cuenta la relación de masa-longitud para cada elemento-eslabón y así no exigir demasiado torque a los eslabones de la base.
- Se debe analizar siempre la señal de control obtenida a la salida del controlador, pues de ser muy alta, a pesar de controlar bien, no servirá debido a que la exigencia para los actuadores será muy alta.
- Se puede y debe optimizar el controlador Fuzzy, combinándolos con otros algoritmos, formando así controladores Neuro-Fuzzy por ejemplo.

IX. REFERENCIAS

- [1] John H. Holland, "Adaptation in Natural and Artificial Systems", 1975
- [2] Katsuhiko Ogata, "Ingeniería de Control Moderna", Pearson, 2003, 4ta edición
- [3] Ahmed A. Shabana, "Computational Dynamics", Editorial John Wiley
- [4] Bezdec, J.C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
- [5] Dubois, D. and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, 1980.
- [6] Sugeno, M., *Industrial applications of fuzzy control*, Elsevier Science Pub. Co., 1985.
- [7] Jang, J.-S. R. and C.-T. Sun, "Neuro-fuzzy modeling and control, *Proceedings of the IEEE*, March 1995.
- [8] Mathworks, "Fuzzy Logic Toolbox de Matlab"
- [9] *Robot Modeling and Control* - Mark W. Spong, Seth Hutchinson, and M. Vidyasagar – 1era Ed.-pags. 101-107