

Implementación de un control fuzzy para el control cinemático directo en un robot manipulador

Implementation of a fuzzy control for the direct kinematic control of a robot manipulator

D.A. Tibaduiza¹ I. Amaya² S. Rodríguez² N. Mejía² M. Flórez²

Recibido 1 de junio de 2011, aceptado 9 de diciembre de 2011

Received: June 1, 2011 Accepted: December 9, 2011

RESUMEN

En este artículo se muestra el desarrollo e implementación de la lógica difusa como herramienta de control de posición para cada una de las articulaciones de un robot tipo PUMA. Se hace una descripción general del robot y se muestra el cálculo del volumen de trabajo, el cual es usado para la fuzzificación en el desarrollo del controlador. Finalmente es mostrado el desarrollo y la simulación del controlador usando la toolbox fuzzy de Matlab, así como la descripción de una implementación realizada en un PLC.

Palabras clave: Control cinemático, fuzzy, robótica, manipuladores, PLC.

ABSTRACT

In this article, the development and implementation of a fuzzy logic system as position control tool of each one of the joints in a PUMA robot is shown. A general description, which include general descriptions about the robot as workspace and therefore the development of the strategy of control with the definition of the rules in the fuzzification process is also included. Finally are shown the development and simulation of the controller using the fuzzy control toolbox of Matlab and the description of a implementation in a PLC.

Keywords: Kinematic control, fuzzy, robotic, robotic arms, PLC.

INTRODUCCIÓN

La robótica ha experimentado en los últimos años un gran auge y se encuentra cada vez más presente en los medios universitarios e industriales. Desde que en 1978 General Motor Studios creara el robot PUMA (Manipulador Universal Programable para Tareas de Ensamblaje) se han desarrollado nuevos robots y entornos de programación que incluyen la sinergia de nuevos modelos del robot, su sensorica asociada y el entorno de trabajo para permitir la interacción del robot en ambientes dinámicos. Aquellos primeros robots como el PUMA hoy en día se han consolidado como

modelos estándar y, aunque no pueden rivalizar con los robots actuales utilizados con fines comerciales, poseen ciertas características que los hacen muy atractivos desde el punto de vista didáctico. Es interesante por tanto trabajar con estos modelos en el desarrollo de controladores que permitan probar las diferentes estrategias de control desde el punto de vista cinemático y dinámico. Técnicas como el control fuzzy proveen una poderosa herramienta para resolver problemas con ambientes impredecibles [5]. Los controladores fuzzy pueden desempeñarse mejor que un controlador PID lineal puesto que estos no poseen características no lineales [6-7, 9].

¹ Departamento de matemática aplicada III. Universidad Politécnica de Cataluña. Barcelona, España. Carrer Comte d'Urgell 187. C.P: 08234. E-mail: diego.tibaduiza@upc.edu

² Facultad de Ingeniería Mecatrónica. Universidad Autónoma de Bucaramanga, Bucaramanga, Colombia. E-mail: iamaya2@gmail.com, silviaf.rodriguez@gmail.com, dairo100@gmail.com, mariomk13@gmail.com

MARCO CONCEPTUAL

Clasificación de los robots

Un robot puede ser clasificado atendiendo a diferentes criterios o características [1]. Esta clasificación suele depender de la aplicación o la tarea a desempeñar. De una manera bastante general se podrían definir dos grandes grupos: robots manipuladores y robots móviles. Los primeros son esencialmente brazos articulados o dicho de otra manera son cadenas cinemáticas abiertas formadas por varios eslabones, mientras que los segundos corresponden a robots cuyo desplazamiento no está restringido a la máxima longitud de sus articulaciones; entre ellos los más comunes son los vehículos con ruedas [3, 10-11].

Existen diferentes subclasificaciones y combinaciones que han terminado en nuevas generaciones de robots en los que se involucran técnicas de inteligencia artificial para dotarlos de cierta autonomía. En el caso de los robots manipuladores es posible clasificarlos de acuerdo a su configuración como: Cartesiano, Cilíndrico, Polar o Esférico, articular, SCARA y paralelo.

Modelado cinemático de un manipulador

En robótica existen dos opciones a la hora de mover un robot [2]. Estas opciones normalmente son conocidas como modelo cinemático del manipulador. La primera de ellas (conocida como modelo cinemático directo) hace referencia a obtener las coordenadas del efector final cuando se conoce la posición de cada una de las articulaciones (sin importar si estas son prismáticas o rotacionales). El segundo caso ocurre cuando se conoce el punto y la orientación en el espacio a la que se quiere hacer llegar el efector final, pero se desconocen las coordenadas de cada articulación y por tanto se requiere calcularlas para hacer llegar el robot a esta posición; esta estrategia se conoce como cinemática inversa.

Para el cálculo cinemático se aprovechan algunas herramientas matemáticas ampliamente conocidas en el mundo de la robótica industrial como lo son las matrices homogéneas, las que representan la posición y orientación de un elemento en el espacio. Específicamente en cinemática se usan además herramientas como los screws y los parámetros de Denavit-Hartenberg para tener un conocimiento físico del robot y de sus movimientos.

Lógica fuzzy

El control difuso es una técnica que se encuentra agrupada bajo el esquema de los sistemas expertos, y su popularidad se basa en la dificultad (o imposibilidad) de modelar algunos sistemas a través de ecuaciones matemáticas. Adicionalmente, su utilidad radica en que se puede aprovechar el conocimiento de un experto [13] sin necesidad de modelar matemáticamente un proceso, lo que lo hace atractivo en procesos de gran complejidad.

La lógica fuzzy o lógica borrosa es una extensión de la lógica clásica diseñada para permitir el razonamiento sobre conceptos imprecisos [9]. Entre sus aplicaciones es posible encontrar el desarrollo de sistemas expertos difusos para reconocimiento de patrones y clasificación, fuzzy clustering, control fuzzy, monitoreo, diagnóstico, optimización, entre otros [4].

A través de los años se han generado importantes aportes al desarrollo de la lógica difusa, como el propuesto por Looney en 1998, quien señala que los algoritmos difusos tradicionales no son capaces de predecir comportamientos futuros, por lo que propone un modelo que utiliza redes de petri para simular, a través de un algoritmo difuso, una red neuronal, en el sentido de aprendizaje y ejecución de lógica predictiva [13]. Asimismo, se han desarrollado aplicaciones que evidencian su aplicabilidad a diferentes campos. Estas incluyen un híbrido de control difuso y red neuronal, propuesto en 1995 por Daxwanger y Schmidt, para imitar el conocimiento de un experto y hacer que un robot coordine el parqueo de un vehículo [14]. De forma similar, Bentalba, Hajjaji y Rachid proponen, en 1997, un sistema fuzzy para estabilizar la respuesta de dos sistemas lineales y lograr controlar un robot móvil [15]. Continuando en el campo de la robótica móvil, se encuentran el trabajo de Yang y el de Raimondi. El primero de ellos, 2004, investiga sobre un robot no-holonómico y formula un conjunto de reglas difusas basado en el método directo de Lyapunov. Como resultado, obtienen el ángulo de giro deseado y un seguimiento satisfactorio de la trayectoria propuesta [17]. Por su parte, los segundos proponen utilizar, en 2005, el método de Lyapunov y el lema de Barbalat como alternativa para un problema de seguimiento de trayectoria donde tanto la cinemática como la dinámica del robot son desconocidas [18].

En robots PUMA se tienen igualmente algunas referencias de diseños de controladores para el movimiento de las articulaciones del robot; en algunos casos estas combinan el uso de otras técnicas como los algoritmos genéticos para obtener la definición de los elementos del controlador fuzzy [16].

En 2008 Mazhari y Kumar [19] comparan técnicas heurísticas de búsqueda global, algoritmos genéticos, optimización por enjambre de partículas, recocido simulado, búsqueda generalizada de patrones, algoritmos genéticos híbridos con Nelder-Mead y búsqueda generalizada de patrones para sintonizar un controlador PID-Fuzzy para un robot PUMA 560. Este mismo año Alavandar y Nigam presentaron el uso de un control fuzzy PD+I para controlar las articulaciones de un robot PUMA [20]. La prueba de los algoritmos desarrollados fue aplicada mediante simulación numérica usando el modelo dinámico del robot.

Piltan, Sulaiman, Gavahian, Soltani y Roosta en 2011 aplicaron control fuzzy deslizante a tres grados de libertad de un robot PUMA. El controlador fuzzy propuesto es planteado como una estructura en paralelo de una superficie deslizante P+D y otra P+I+D [21].

También existen aplicaciones sobre otros tipos de robots, pero debido a limitaciones de espacio solo se resalta el de Zheng-jie [22] en el campo del control de vehículos aéreos y el de Lin para robots bípedos [23]. El primero de ellos logró, en 2010, diseñar un modo de control combinado entre el modo de control aerodinámico y el modo de control de aceleración lateral directa, mejorando la velocidad de respuesta y reduciendo el nivel de sobrecarga en los cambios. Finalmente, y también en 2010, Lin, Chang, Lyu, Wang y Li [23] aplicaron la lógica difusa en el ajuste de la estabilización de un robot bípedo, logrando desplazamientos en un plano horizontal y en el ascenso de escalones, suprimiendo la inestabilidad inherente al sistema y reduciendo la duración de oscilaciones transitorias. En este trabajo se aplica la metodología difusa para diseñar un controlador de posición de un robot tipo PUMA, buscando que ejecute movimientos más estables entre los puntos de su recorrido. Dado que el robot disponible no estaba diseñado para fines industriales, se definió una carga máxima de 200 g para evitar dañar la estructura de acrílico.

A nivel local se han realizado algunos trabajos visibles que incluyen el diseño de manipuladores a los cuales se les aplica el diseño de controladores de tipo Proporcional [3, 24-25].

Conjuntos fuzzy

Los conjuntos fuzzy son una generalización de los conjuntos clásicos en los cuales a diferencia de estos últimos se tienen valores de pertenencia entre 0 y 1; por ejemplo, en teoría clásica si se definiera como conjunto a personas altas (mayores de 1,90 m), todos los individuos que estuvieran en este conjunto tendrían valor de pertenencia igual 1 y los que no serían cero.

En los conjuntos fuzzy, puesto que este es un concepto impreciso, es posible definir valores que van desde 0 hasta 1, pudiéndose decir, por ejemplo, que alguien que mide 1,85 pertenece al conjunto de los altos con un valor de 0,8.

Funciones de pertinencia

En teoría es posible usar cualquier función que se encuentre en el intervalo entre 0 y 1, aunque se han adoptado de manera clásica en la mayoría de los trabajos las funciones de tipo trapezoidal, triangular, gaussiana, sigmoidea y singleton, esta última hace referencia a que un único valor de pertenencia es igual a 1 en un solo punto y el resto es igual a cero. Ver Figura 1.

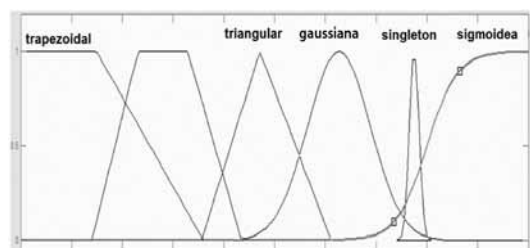


Figura 1. Funciones de pertinencia.

Variables lingüísticas

Una variable lingüística es aquella cuyos valores son conjuntos difusos [9]. Por ejemplo, la temperatura, velocidad, etc. Cada una de estas variables lingüísticas a su vez puede tomar diferentes valores como muy pequeña, pequeña, media, alta, muy alta. Estos diferentes valores de la variable son llamados términos lingüísticos.

Operaciones con conjuntos fuzzy

Las operaciones que se pueden realizar en conjuntos fuzzy son extensiones de las que se realizan en la teoría clásica [9]. Las tres operaciones fundamentales son la unión, intersección y el complemento, ver Tabla 1.

Tabla 1. Operaciones básicas usadas en la lógica fuzzy.

Complemento: $\mu_{A^c}(x) = 1 - \mu_A(x)$	(1)
Unión: $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$	(2)
Intersección: $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$	(3)

Generación de trayectorias

La generación de trayectorias puede describirse, de manera general, como el conjunto de técnicas que permiten llevar un robot de una posición inicial a una final y en la que pueden existir o no obstáculos. Existen diferentes técnicas de generación de trayectorias en robots manipuladores, las cuales, en algunos casos, tienen en cuenta tanto la cinemática como la dinámica de los robots.

DESCRIPCIÓN DEL ROBOT

El robot usado para la implementación de este trabajo es un robot angular de seis grados de libertad, conocido como manipulador tipo PUMA. Este robot está compuesto por seis motores, cuatro (4) de corriente continua (DC) y dos (2) servomotores. Se disponen en el sistema de tal forma que los DC trabajan en el accionamiento de las articulaciones de la cadera, el hombro, el codo y el efector, mientras que los servomotores se encuentran dispuestos paralelamente y son los responsables del movimiento de la muñeca (ver Figura 2).

Para verificar el desempeño del robot y poder conocer el consumo de corriente de los actuadores DC para las cargas definidas, se realizó una serie de pruebas, las cuales se muestran en la Tabla 2. La carga utilizada fue de 0,2 kilogramos.



Figura 2. Robot PUMA.

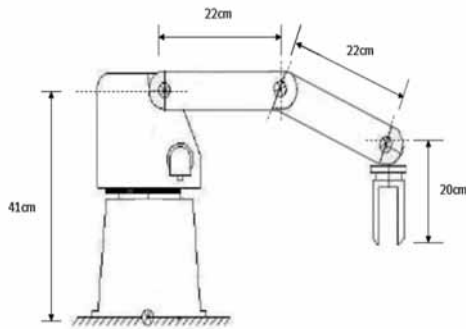
Tabla 2. Respuesta de los motores.

	Sin Carga	Con Carga
Cadera		
Rotando	0,1 A	0,1 A
Hombro		
Subiendo	0,3 A	0,35 A
Bajando	0,16 A	0,14 A
Codo		
Subiendo	0,35 A	0,4 A
Bajando	0,28 A	0,26 A
Efector		
Abriendo y Cerrando	0,075 A	0,075 A

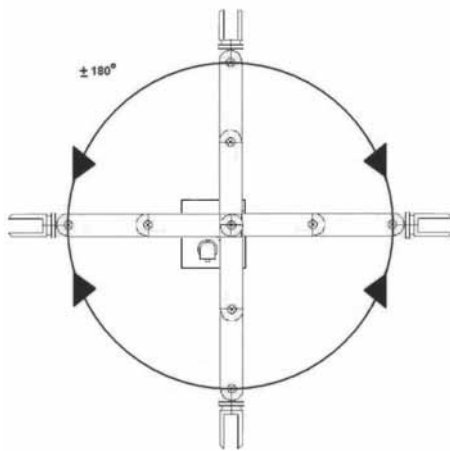
En el caso de la muñeca, dependiendo del servomotor activado se pueden realizar diferentes movimientos, tales como rotaciones en diferentes sentidos, así como desplazamientos positivos y negativos. Durante el proyecto resumido en este documento se restringió el movimiento de este grado de libertad a una rotación respecto al eje Y, para lo cual se utilizó el 15% de la señal de control. Por lo anterior, si un servomotor recibía una señal con un ciclo activo del 14%, el otro debía recibir una con el 1%.

Volumen de trabajo

El volumen de trabajo para el cual el manipulador está diseñado se describe por la trayectoria que enmarca el movimiento de su brazo, así como por el barrido de 360° que está en capacidad de realizar gracias a su articulación de cadera. En la Figura 3a se muestran las medidas del robot y en la Figura 3b el alcance máximo visto desde un plano lateral.



a.



b.

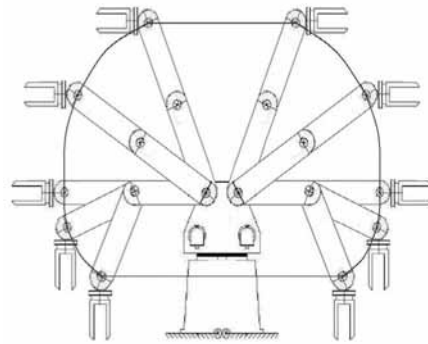
Figura 3. Medidas generales del robot.

Visto desde la parte superior, es posible calcular el área de trabajo del robot utilizando la elongación máxima del brazo (0,64 m). Dicha área es mostrada en la Figura 4a. En la Figura 4b se tiene una vista lateral del área de trabajo del robot.

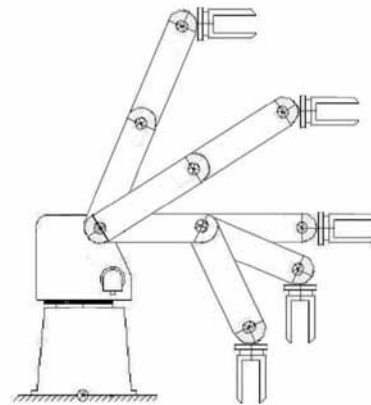
Como a lo largo de esta aplicación el recorrido del brazo se limitó a 180°, el volumen de trabajo se ve disminuido. Sin embargo, es necesario conocer su valor. Para ello se toma la ecuación del volumen de una esfera (4).

$$V = \frac{4 \times \pi \times r^3}{3} \quad (4)$$

Donde r corresponde al radio de la esfera que se obtiene con el brazo estirado y V al volumen. Utilizando los datos del manipulador se obtiene que $V = 1,098 \text{ m}^3$.



a.



b.

Figura 4. Vistas del área de trabajo.

La zona que no cubre el manipulador está representada por un cilindro, como se muestra en la Figura 5:

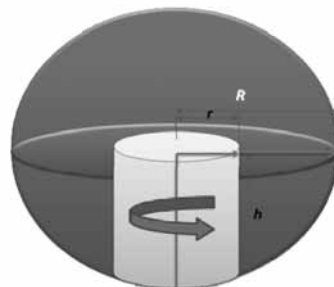


Figura 5. Zona muerta.

Para el cálculo del volumen del cilindro se toman nuevamente los parámetros físicos propios del robot, con lo que se obtiene:

$$V = \pi \times r^2 \times h \quad (5)$$

$$V = 0,064m^3 \quad (6)$$

Por lo que al efectuar la diferencia entre los dos se obtiene un volumen de trabajo equivalente a $1,034 m^3$.

Es necesario además incluir una limitación física adicional del robot debido al movimiento de la cadera; dada la restricción en la rotación de la cadera a 180° el volumen real de trabajo (limitado) corresponde a la mitad del valor obtenido anteriormente ($0,517 m^3$), lo que se ve reflejado en la Figura 6.

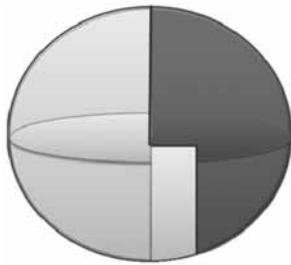


Figura 6. Volumen final.

ESTRATEGIA DE CONTROL

Este sistema consiste en un controlador fuzzy de posición, el cual fue diseñado en Matlab[®] usando la toolbox fuzzy[®] y posteriormente implementado en un PLC.

El robot dispone de seis sensores electromecánicos, uno en cada articulación, que permiten conocer la posición actual del robot y sirven de elemento de realimentación para el sistema de control. La Figura 7 muestra el diagrama de bloques general para cada articulación; como se puede observar en la figura la variable de control es el error en la posición que se obtiene de la resta entre la señal de consigna y la lectura actual obtenida de cada sensor.

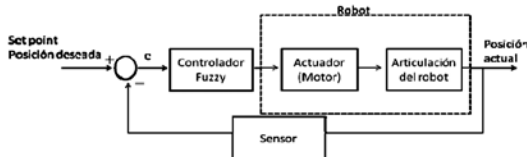


Figura 7. Diagrama de bloques del sistema de control.

El robot cuenta además con cuatro finales de carrera; dos para la articulación del hombro y dos

más para la del codo. Solo uno de ellos (indicador de 0° en articulación del hombro) se encuentra normalmente cerrado.

Descripción del controlador

Para el diseño y la simulación inicial del controlador se usó la toolbox Fuzzy Controller de Matlab. Este diseño consta de un controlador con una entrada y una salida como se muestra en la Figura 8. La entrada es la señal de error que se produce entre el set point en la posición y la medida en las articulaciones del robot y la salida es la señal de actuación a enviar al motor.

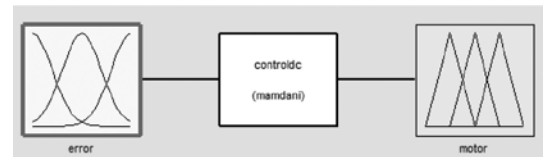


Figura 8. Esquema general del sistema fuzzy.

Fuzzificación

La fuzzificación consiste en la descripción del comportamiento del robot en lenguaje fuzzy, para lo cual es necesario definir los términos y variables lingüísticas así como las reglas que asocian la entrada con la salida del sistema.

Para cada una de las variables propuestas se plantean seis funciones de membresía en el rango de valores entre 1.000 y -1.000 . (Figura 9 y Figura 10). Las funciones de membresía para la entrada son: GrandeNeg, MedNeg, PeqNeg, PeqPos, MedPos, Grande Pos y para la salida son: RapidoNeg, MedNeg, LentoNeg, LentoPos, MedPos, RapidoPos.

La selección de los rangos así como la forma de las funciones fueron ajustadas de acuerdo a los resultados obtenidos.

Para crear la base de conocimiento se definen las reglas del controlador relacionando la entrada y la salida con reglas del tipo IF, THEN. El controlador diseñado consta de seis reglas que son el producto de la observación del comportamiento del robot y son las encargadas de actuar sobre los motores DC. Ver Figura 11.

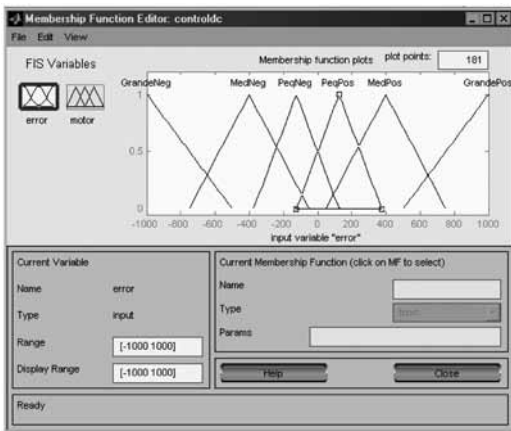


Figura 9. Función de entrada.

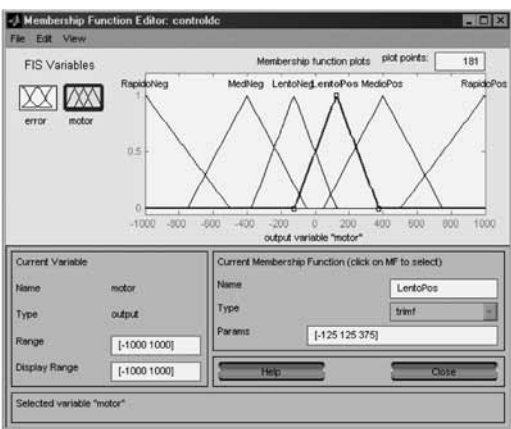


Figura 10. Funciones de salida.

1. If (error is PeqPos) then (motor is LentoPos) (1)
2. If (error is MedPos) then (motor is MedioPos) (1)
3. If (error is GrandePos) then (motor is RapidoPos) (1)
4. If (error is GrandeNeg) then (motor is RapidoNeg) (1)
5. If (error is MedNeg) then (motor is MedNeg) (1)
6. If (error is PeqNeg) then (motor is LentoNeg) (1)

Figura 11. Reglas del controlador.

En cuanto respecta al método de defuzzificación, se utilizó el centroide puesto que sus resultados eran mejores que el uso de otro tipo de métodos incluidos en la toolbox de Matlab, además dada su simplicidad y puesto que la idea es llevarlo a un PLC su implementación resulta un poco más simple.

Para verificar el comportamiento del controlador se analizó el visor de reglas, realizando variaciones en la variable de entrada (“error”), con el objeto de verificar la respuesta de su salida (“motor”), como se puede observar en la Figura 12. En esta se puede

apreciar la representación gráfica de las seis reglas y los siete conjuntos de salida propuestos así como el rango definido para cada una de estas.

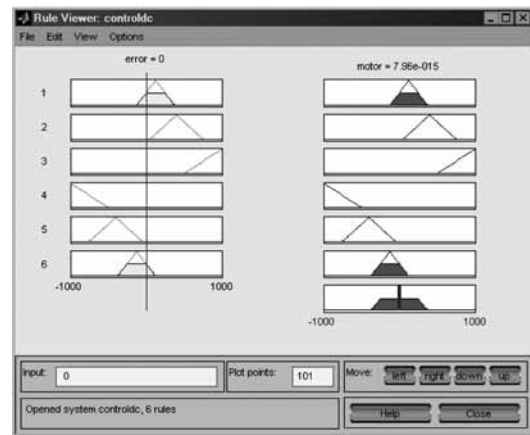


Figura 12. Respuesta del controlador ante una determinada señal de error.

Otra representación del comportamiento del controlador se puede apreciar en la Figura 13, donde se muestra una gráfica espacial de las reglas que, como se puede deducir, tiende a una línea recta, indicando que existe una relación casi directa entre la entrada y la salida, lo que hace que su comportamiento tienda a ser como el de un controlador proporcional.

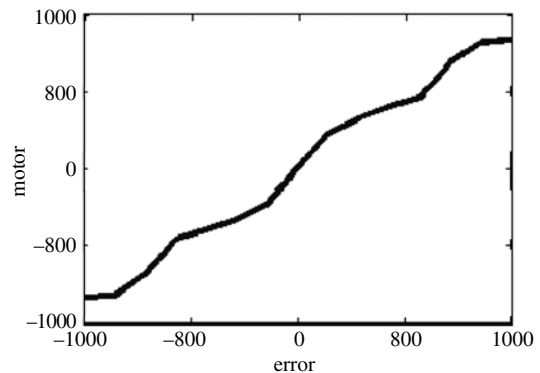


Figura 13. Respuesta vs error.

Desarrollo del controlador fuzzy

La implementación de un controlador fuzzy en un PLC puede llegar a convertirse en una tarea tediosa si no se cuenta con módulos especializados. Puesto que para este trabajo no se contaba con estos módulos fue necesario implementar todas las etapas del controlador.

Para ilustrar cómo se realiza esta implementación de un sistema tipo Mandani se empezará por mostrar un ejemplo sencillo en el cual se poseen dos reglas para el controlador. En la Figura 14 pueden observarse los valores de pertenencia para cada regla.

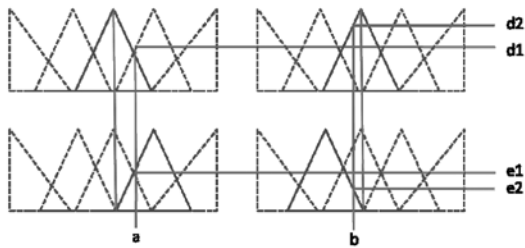


Figura 14. Funciones de pertenencia.

Para cada regla se obtiene la razón de pertenencia (valor entre cero y uno) referente a los valores utilizados (a y b en este caso) y se escoge la menor de ellas (d y e respectivamente). Para el caso del actual manipulador, solo existe una variable de entrada (error), por lo que el valor obtenido en este punto será el equivalente a la razón de pertenencia para cada una de las reglas, como se observa en la Figura 15.

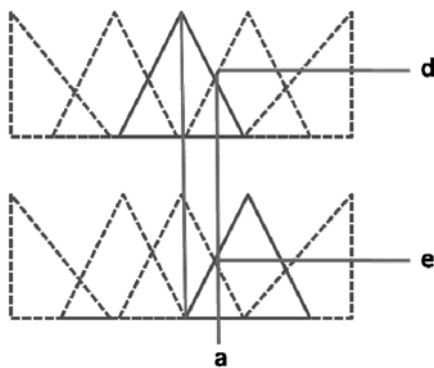


Figura 15. Funciones de pertenencia para el controlador fuzzy con una sola entrada.

Posteriormente, se calcula el promedio ponderado entre las razones de pertenencia obtenidas y el centro de la función correspondiente a la salida de cada una de las reglas (c1 y c2 para este ejemplo) como en (7). Es decir:

$$S = \frac{c1 * d + c2 * e}{d + e} \quad (7)$$

El promedio obtenido corresponde al valor de salida del controlador. En el caso del manipulador se deben tomar todas las reglas y los valores de sus centros.

Implementación en el PLC

De acuerdo a las características propias de este robot, se realizó el acondicionamiento de las señales y el desarrollo del controlador. La Figura 16 muestra el algoritmo general del proceso ejecutado por el controlador para calcular la respuesta del sistema, es decir, la acción sobre los motores.

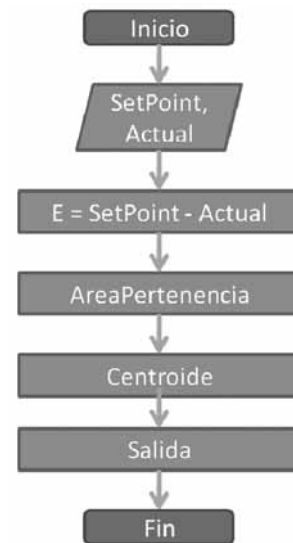


Figura 16. Diagrama general del algoritmo implementado.

Este algoritmo puede ser descrito en los siguientes pasos:

- Leer el desplazamiento deseado para la primera articulación (set point), así como el recorrido realizado en esta dirección (valor actual).
- Calcular el error instantáneo existente.
- Utilizar las reglas del controlador fuzzy para determinar el grado de pertenencia a cada una de las funciones de membresía.
- Calcular los centroides de las áreas, así como su promedio, obteniendo un valor de salida para el actuador.

Este proceso es repetido hasta que se alcanza el set point, controlando un máximo de dos articulaciones en paralelo y exceptuando los servomotores.

El acondicionamiento de las señales de entrada y salida se implementó a través del diseño de circuitos electrónicos de acondicionamiento básico. La Figura 17 muestra el diagrama de bloques de conexión hacia los motores.



Figura 17. Acople a motor DC.

Por otra parte, la Figura 18 muestra el diagrama de bloques del acondicionamiento de la señal de conexión a los servomotores.



Figura 18. Acople a servomotor.

Como protección adicional, se incluyó un circuito de final de carrera para detectar un punto crítico en el movimiento de alguna articulación.

Las pruebas realizadas para verificar la estrategia implementada incluían pruebas de repetitividad, de las cuales se pudo comprobar la respuesta de cada uno de los motores, obteniéndose sobrepasos máximos menores al 10%.

CONCLUSIONES

Se implementó un controlador fuzzy diseñado y simulado en Matlab con una entrada (error del motor) y una salida (ancho de pulso). De esta forma se logró obtener movimientos con una respuesta más suave, lo que repercute en menores picos de corriente (< 10% de sobrepico máximo) y por ende en un menor consumo de potencia, lo que genera un menor costo de utilización del manipulador.

Al implementar la secuencia de trabajo del manipulador se diseñó un conjunto de funciones y bloques de funciones encargados de aplicar el control fuzzy a un motor específico. El programa

principal fue diseñado utilizando el lenguaje KOP y se encarga de tomar cada uno de los movimientos y enviarlos al controlador para su ejecución.

El diseño de un controlador fuzzy puede ser dividido en varias etapas de cálculos geométricos sencillos. Con esto se logra simplificar en gran medida la complejidad del controlador y se facilita la detección y corrección de errores.

Es conveniente diseñar un conjunto de funciones de membresía simétrico ya que permite obtener una señal de valor cero cuando, a través de las entradas y combinaciones de reglas, se desea estabilizar el sistema en un punto.

Debido al restringido y bajo número de entradas/salidas de la CPU del PLC utilizado, se diseñó un circuito electrónico utilizando multiplexores y demultiplexores para distribuir la señal de activación de los motores y poder leer los pulsos enviados por los encoders.

Adicionalmente, se incorporaron medidas de seguridad como búferes y optoacopladores para proteger los encoders y el PLC. Por último, se incorporó una etapa de activación de potencia utilizando Puentes H dependiente de la señal direccionada por los demultiplexores, lo que indica que, dependiendo de la aplicación, se pueden ampliar las capacidades de entrada/salida sin necesidad de adquirir un costoso módulo.

AGRADECIMIENTOS

Los autores agradecen a la Universidad Autónoma de Bucaramanga, lugar donde se realizó el montaje y la simulación del diseño presentado.

REFERENCIAS

- [1] A. Barrientos, L. Peñín, C. Balaguer y R. Aracil. "Fundamentos de Robótica". Mc Graw Hill. Segunda edición. 2007.
- [2] A. Ollero Baturone. "Robótica, Manipuladores y robots móviles". Alfaomega. Marcombo, 2001.
- [3] D. Tibaduiza, N. Chio, J. Grosso y M. Anaya. "Parámetros Biomecánicos de un Exoesqueleto Mecatrónico de Miembro Superior". 5th IEEE Colombian Workshop on Robotics and Automation. 2009.

- [4] D. Tibaduiza y C. Daza. "Control Fuzzy de Velocidad para una Transmisión Hidrostática". *Revista Facultad de Ingeniería Universidad de Antioquia*. No. 44 pp. 137-143. Junio 2008
- [5] S. Alvandar and M.J. Nigam. "Adaptative Neuro-Fuzzy Inference based control of six DOF robot manipulator". *Journal of Engineering and Technology Review*. Vol. 1, pp. 106-111. 2008.
- [6] H.R. Berenji. "Fuzzy logic controllers". In R.R. Yager and L.A. Zadeh, Eds. *An Introduction to Fuzzy Logic Application in Intelligent System*. Boston, MA: Kluwer. 1992.
- [7] E.H. Mandani. "Application of fuzzy algorithm for control of simple dynamic plant". *Proc. Inst. Elect. Eng.* Vol. 121, pp.1585-1588. 1974.
- [8] Ch.S. Kim, W.H.Seo, S.H. Han and O. Khatib. "Fuzzy Logic Control of a robot manipulator based on visual servoing". *Proc. ISIE*. Pusan, Korea. 2001.
- [9] F. Escolano, M. Cazorla, M. Alfonso, O. Colomina y M. Lozano. "Inteligencia Artificial Modelos, Técnicas y Areas de Aplicación". Thomson Editores Spain. 2001. ISBN:84-9732-183-9.
- [10] D.A. Tibaduiza y M. Anaya. "Campos de Potencial Aplicados al Planeamiento de Caminos de Robots Móviles". *Revista de Ingeniería*. Vol. 1, N° 3, pp. 23-34. Enero 2007.
- [11] D.A. Tibaduiza. "Planeamiento de Trayectorias de Robots Móviles". Tesis para optar al grado de Maestría. Universidad Industrial de Santander. 2005.
- [12] B. Gaines. "From fuzzy logic to expert systems". *Information Sciences*. Vol. 36, pp. 5-16. August, 1985.
- [13] C.G. Looney. "Fuzzy Petri Nets for Rule-Based Decision making". *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*. Vol. 18, pp. 178-183. 1988.
- [14] W.A. Daxwanger and G.K. Schmidt. "Skill based visual parking control using neural and fuzzy networks". *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, pp. 1659-1664. 1995.
- [15] S. Bentalba, A. El Hajjaji and A. Rachid. "Fuzzy control of a mobile robot: a new approach". *Proceedings of the 1997 IEEE International Conference on Control Applications*, pp. 69-72. 1997.
- [16] P. Dassanayake, K. Watanabe, K. Kiguchi and K. Izumi. "Fuzzy behavior-based motion planning for the PUMA robot". *Proceedings of Intelligent robots and systems*. Vol. 3, pp. 1912-1917. 2000.
- [17] S.X. Yang, H. Li, M.Q.-H. Meng and P.X. Liu. "An Embedded Fuzzy Controller for a Behavior-Based Mobile Robot With Guaranteed Performance". *IEEE Transactions on Fuzzy Systems*. Vol. 12, pp. 436-446. August, 2004.
- [18] F.M. Raimondi, M. Melluso and L.S. Ciancimino. "A New Kinematic and Dynamic Direct Adaptive Fuzzy Control of Constrained Mobile Wheeled Vehicles". *2005 IEEE Conference on Emerging Technologies and Factory Automation*. Vol. 2, pp. 181-188. 2005.
- [19] S. Mazhari and S. Kumar. "Heuristic Search Algorithms for Tuning PUMA 560 Fuzzy PID Controller". *International Journal of Electrical and Computer Engineering*. Vol. 3. 2008.
- [20] S. Alvandar and N.F. Nigam. "Fuzzy PD+ I control of a six DOF robot manipulator". *Industrial Robot, An International Journal*. Vol. 35, Issue 2, p. 8. 2008.
- [21] F. Piltan, N. Sulaiman, A. Gavahian, S. Soltani and S. Roosta. "Design mathematical tunable gain PID-like sliding mode fuzzy controller with minimum rule base". *International Journal of Robotics and Automation (IJRA)*. Vol. 2, Issue 3. 2011.
- [22] W. Zheng-jie, L. Yang-bo, L. Jing and S. Wei. "Modeling and Combined Control of Aerial Vehicle Based on Aerodynamic Force and Lateral Thrust". *The 2010 International Conference on Modelling, Identification and Control (ICMIC)*, pp. 106-111. 2010.
- [23] J. Lin, J. Chang, S.M. Lyu, S.W. Wang and Y.W. Lin, "Locomotion control of a biped robot for stair-climbing by fuzzy stabilization tuning approach". *2010 IEEE International Conference on Control Applications*, pp. 1590-1595. September, 2010.

- [24] A. Gorki, J. Pérez and M. Suarez. “Prototipo de un robot tipo PUMA de seis grados de libertad”. Trabajo de grado. Universidad Industrial de Santander. 2005.
- [25] M. Romero y J. García. “Diseño, construcción y programación de un manipulador SCARA”. Trabajo de grado. Universidad Autónoma de Bucaramanga. 2005.