

# DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL BASADO EN REDES NEURONALES ARTIFICIALES PARA EL BRAZO ROBÓTICO MOVEMASTER

DESIGN AND IMPLEMENTATION OF AN ARTIFICIAL NEURAL BASED CONTROL SYSTEM NETWORK FOR A MOVEMASTER ROBOTIC ARM

**Recibido:** 15 de octubre del 2009

**Aprobado:** 10 de diciembre del 2009

CAMILO ANDRÉS GUTIÉRREZ RAMÍREZ\*  
DIEGO FELIPE BELTRÁN GARCÍA\*\*  
JAIRO ORLANDO MONTOYA GÓMEZ\*\*\*

## Resumen

Este proyecto se dirige a encontrar un desplazamiento óptimo en el cual el robot MoveMaster, de cinco grados de libertad, pueda mover sus eslabones y determinar una trayectoria apropiada para alcanzar un punto final. Para lograr este objetivo se implementó un sistema inteligente basado en una red neuronal que almacena todas las posibles rotaciones del manipulador para llegar a un punto deseado, determinando el movimiento óptimo, sin que el robot colisione en su trayectoria.

Fue desarrollado un modelo matemático para obtener las ecuaciones que identificarán el manipulador, y serán usadas para el entrenamiento de la red neuronal y el control del sistema.

**Palabras clave:** cinemática, control inteligente, red neuronal, espacio de trabajo.

## Abstract

The project aims at finding an optimum displacement level with which, the five degree of freedom, MoveMaster robot may move its joints when determining an appropriate trajectory to reach a specific point. To achieve this end we implemented a neural network based smart system that stores all the possible rotations done by the handler needed to reach a specific point, determining optimum movement while avoiding collision with the robot.

A mathematical model was developed to obtain the equations that would identify the handler, and which will be used to train the neural network and the system's control.

**Keywords:** kinematics, intelligent control, neural network, workspace.

\* Estudiante Facultad de Ingeniería de Diseño y Automatización Electrónica, Universidad de La Salle, correo electrónico: kmo273@hotmail.com

\*\* Estudiante Facultad de Ingeniería de Diseño y Automatización Electrónica, Universidad de La Salle, correo electrónico: dfinib@gmail.com

\*\*\* Msc, Ingeniero Mecánico, profesor Facultad de Ingeniería de Diseño y Automatización Electrónica, Universidad de La Salle, correo electrónico: jamontoya@unisalle.edu.co

## Introducción

Debido al gran avance de la tecnología y la informática en los últimos tiempos, se han creado nuevas técnicas y metodologías para controlar los procesos de manufactura y los elementos que la conforman como máquinas, brazos manipuladores, control de calidad, con el fin de buscar una integración óptima y precisa de las operaciones que permita la automatización de los equipos (Richard, 1989).

Los brazos robóticos, como dispositivos manipuladores de herramientas o materiales, desempeñan un papel importante dentro de los procesos de manufactura (Barrientos, 1997), por lo que se hace necesario que cuenten con un mecanismo de control adecuado que permita optimizar tanto el desplazamiento como el posicionamiento del manipulador dentro de un área de trabajo definida.

Debido a que los métodos clásicos de control no contienen ciertas características de los sistemas inteligentes, en especial, el uso de valores cualitativos, difusos, inciertos, de información incompleta, heurística, intuitiva y de aprendizaje a través de la experiencia (Hilera, 1995), las técnicas de control inteligente se muestran atractivas para la utilización sobre plantas complejas como robots manipuladores, en donde existen grandes dificultades para un correcto modelado y posterior control.

Como el Robot MoveMaster es un manipulador con las características de una planta variable y no lineal, se optó por utilizar redes neuronales para su control, y con ello evitar el modelamiento dinámico del manipulador, el cual posee un alto nivel de complejidad.

Con este proyecto se está adecuando el robot MoveMaster con una herramienta de control inteligente, la cual optimizará en gran medida los movimientos del manipulador para la realización de las funciones específicas dentro del CIM, dando una pauta para el comienzo de las aplicaciones de manufactura en el laboratorio de CAD-CAM basado en sistemas inteligentes.

## Modelamiento matemático e implementación en Matlab

El modelamiento matemático es fundamentalmente la representación de las características cinemáticas del manipulador expresadas en valores numéricos,

que serán los parámetros para el entrenamiento de la red neuronal y el control del sistema.

## Cinemática del robot

### Parámetros Denavit-Hartenberg

Teniendo en cuenta las características del manipulador, sus dimensiones y el rango de movimiento, se halla la matriz homogénea que modelará analíticamente la trayectoria del robot (Barrientos, 1997). La matriz resultante se muestra en la tabla 1.

Tabla 1. Matriz resultante del algoritmo dh

	$\theta$	$D_i$	$A$	$\infty$
1	$q_1$	$l_1$	0	90
2	$q_2+90$	0	$l_2$	0
3	$q_3$	0	$l_3$	0
4	$q_4$	0	$l_4$	90
5	$q_5$	$l_5$	0	0

Fuente: los autores

Para relacionar cada uno de los eslabones con la base del manipulador, se generan las matrices de traslación y rotación representadas por las filas de la tabla 1, produciendo de esta forma una matriz general (ver tabla 2) que relacione el origen con el extremo superior del manipulador, de acuerdo con la configuración de ángulos de cada eslabón.

Tabla 2. Parámetros de la relación entre la base y el eslabón final

	$n_1$	$o_1$	$a_1$	$p_x$
PDH05	$n_2$	$o_2$	$a_2$	$p_y$
	$n_3$	$o_3$	$a_3$	$p_z$
	0	0	0	1

Fuente: los autores

Donde los parámetros  $n$ ,  $o$ ,  $a$  y  $p$  representan ecuaciones que caracterizan la relación que existe entre la base del manipulador y el efector final. En la figura 1 se muestra el volumen de trabajo obtenido de la cinemática del robot.

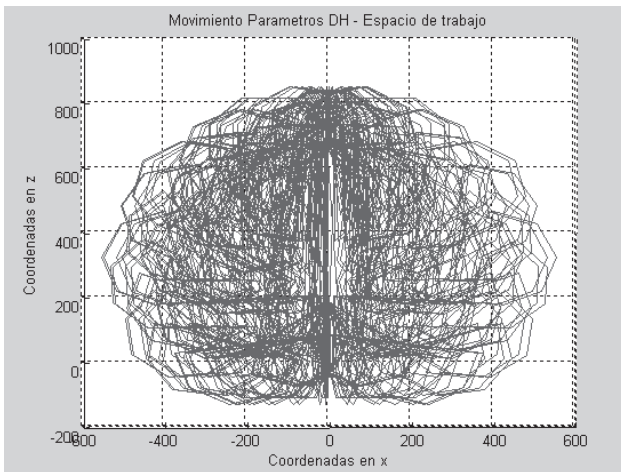


Figura 1. Espacio de trabajo aproximado del manipulador (ejes x - z)

**Matriz jacobiana**

Para construir la matriz jacobiana característica, se opera sobre las columnas 3 y 4 de la matriz homogénea del Denavit-Hartenberg, que relaciona cada eslabón con la base. Con esto se obtiene uno de los parámetros requeridos para determinar la razón de cambio de cada uno de los eslabones con respecto a un punto de referencia fijo. La matriz resultante es de 6 x 5, cuyos valores son funciones algebraicas de senos y cosenos que relacionan las dimensiones de los eslabones con los ángulos de giro.

**Planeación de trayectorias.**

Por medio de las características físicas del manipulador se obtienen los parámetros requeridos para la planeación de trayectorias de cada eslabón (ecuaciones 1 y 2).

Con la planeación de trayectorias obtenemos las funciones (ecuaciones 3,4 y 5) para representar en el tiempo el movimiento de los ángulos (figura 2), sus velocidades (figura 3) y su correspondiente aceleración (figura 4) (Barrientos, 1997).

$$\tau = \frac{V}{ac} \quad (1)$$

$$T = \left( \frac{s * (ang - 0)}{V} \right) + \tau \quad (2)$$

Donde:

V= Velocidad máxima permitida.

ac = Aceleración máxima permitida.

ang= Ángulo actual

s= Signo de la diferencia entre ang-0

$\tau$  y T= Instantes específicos de tiempo

De acuerdo con el valor de  $t$  se obtienen tres funciones diferentes que representan la trayectoria que une dos puntos, dividiéndolos en tres tramos consecutivos.

$$q(t) = \begin{cases} \left( \frac{(s * ac)}{2} \right) t^2 & \text{para } t \leq \tau \\ \left( \frac{-s * V^2}{2ac} \right) + sVt & \text{para } \tau < t \text{ y } t \leq T \\ \left( ang + \left( s * \left( \frac{-ac * T^2}{2} + acTt - \left( \frac{act^2}{2} \right) \right) \right) \right) & \text{para } (T - \tau) < t \text{ y } t \leq T \end{cases} \quad (3)$$

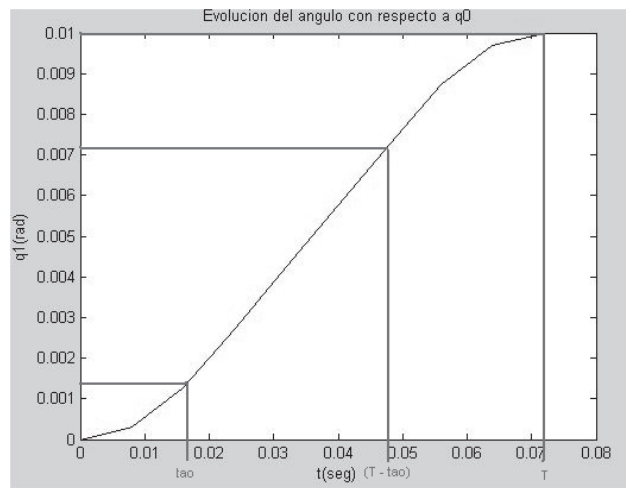


Figura 2. Evolución del ángulo con respecto a q0

Donde:

q0= Valor del ángulo en la posición home

q1= Valor del ángulo al cual el eslabón debe moverse

$$\frac{dq}{dt} = \begin{cases} sact & \text{para } t \leq \tau \\ sV & \text{para } \tau < t \text{ y } t \leq T \\ s * (acT - act) & \text{para } (T - \tau) < t \text{ y } t \leq T \end{cases} \quad (4)$$

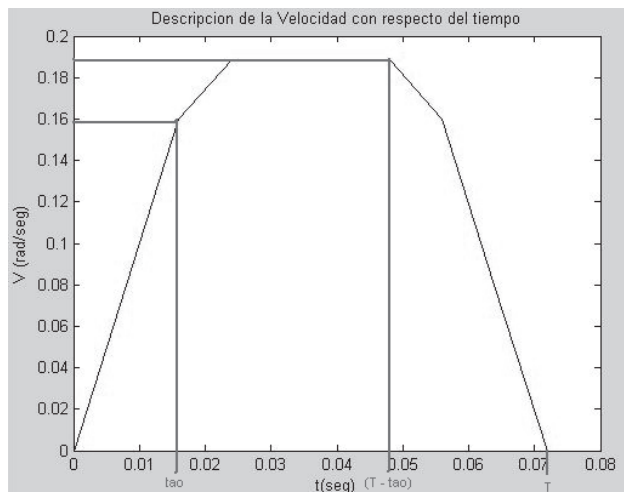


Figura 3. Velocidad en función del tiempo por cada eslabón

$$\frac{d^2q}{dt^2} \begin{cases} s * a & \text{para } t \leq \tau \\ 0 & \text{para } \tau < t \text{ y } t \leq (T - \tau) \\ -s * ac & \text{para } (T - \tau) < t \text{ y } t \leq T \end{cases} \quad (5)$$

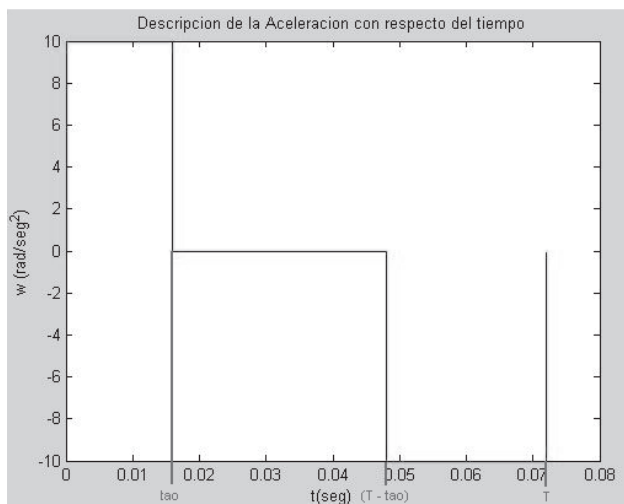


Figura 4. Aceleración en función del tiempo por cada eslabón

En la figuras 3 y 4 se observa que en el primer instante de tiempo ( $t \leq \tau$ ) la aceleración es positiva debido a que la velocidad aumenta. En el segundo instante de tiempo ( $\tau < t \text{ y } t \leq (T - \tau)$ ) la aceleración es cero ya que la velocidad permanece constante. Por último ( $(T - \tau) < t \text{ y } t \leq T$ ), la aceleración es negativa debido a la disminución de la velocidad en el último tramo del movimiento.

Se obtiene la magnitud de la velocidad de llegada para cada eslabón ( $q_1, q_2, q_3, q_4, q_5$ ), calculando

el área bajo la curva de la gráfica de la aceleración (ecuaciones 6 y 7).

$$\dot{q} = \int_0^T \omega dt \quad (6)$$

$$\dot{q} = |(sac * (\tau))| + |(0 * (T - \tau)t)| + |(-sac * (T))| \quad (7)$$

Se realiza el producto cruz con la matriz jacobiana para hallar las velocidades lineales y angulares del extremo del manipulador. El resultado es un vector, para lo cual se halla la correspondiente magnitud, que será la razón de cambio que determinará cuál de todas las combinaciones de ángulos llegará más rápido a un punto final.

### Modelamiento de los objetos presentes en el espacio de trabajo

Debido a que el robot MoveMaster está ubicado en las instalaciones del laboratorio de CAD/CAM, al evaluar una trayectoria para alcanzar la coordenada final el sistema de control debe tener información respecto al entorno en el cual va a realizar el movimiento, con el fin de evitar colisiones contra los objetos ubicados en el área de trabajo

Cada uno de los objetos del espacio de trabajo se expresa por medio de uno o varios cubos. De acuerdo con esto, se toman las coordenadas de cada una de las caras de todos los objetos presentes en el espacio de trabajo del manipulador en CAD/CAM (figura 5), y se crea una matriz que relaciona cada una de ellas. En la matriz se trabajan seis variables que definen el objeto con respecto a un sistema de coordenadas fijo:  $X_{\min}, X_{\max}, Y_{\min}, Y_{\max}, Z_{\min}, Z_{\max}$ .

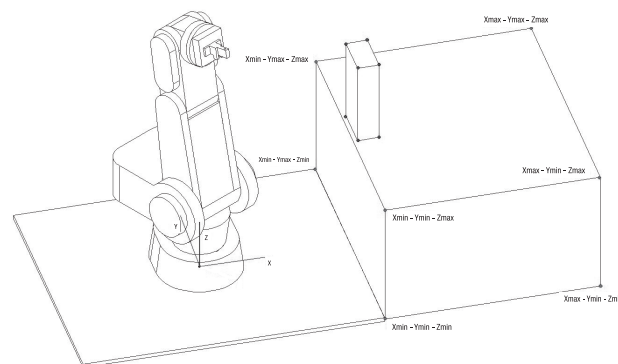


Figura 5. Representación en coordenadas de los objetos en el espacio de trabajo Fuente: los autores.

### Diseño y entrenamiento de la red neuronal

El objetivo de la red neuronal artificial implementada es determinar las coordenadas a las cuales el manipulador debe desplazarse en caso de que en la trayectoria escogida en un principio se encuentre con un obstáculo. Se crea una red neuronal artificial del tipo Back Propagation, que en su neurona de entrada recibe la información del objeto con el cual el manipulador se estrella y en sus tres neuronas de salida las coordenadas a las cuales se debe dirigir para evitar el objeto. En la figura 6 se representa el esquema de la red neuronal.

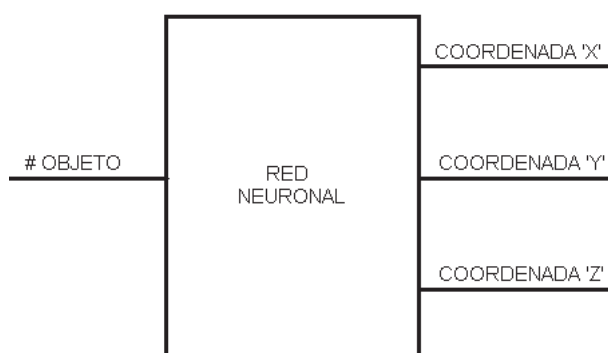


Figura 6. Esquema red neuronal

Fuente: los autores

La red tiene una neurona de entrada, 20 en las tres capas ocultas y tres en la capa de salida.

Se utiliza un entrenamiento de retro propagación de gradiente conjugado (TRAINCGP), que se determina por medios heurísticos, al evaluar tiempo y efectividad con respecto a los otros métodos, obteniendo un error de entrenamiento de  $1 \times 10^{-8}$ .

### Control del sistema

El sistema de control toma los datos ingresados por el usuario y verifica que se encuentren dentro del espacio de trabajo del manipulador. Posterior a esta verificación, y dependiendo de el manipulador ha sido configurado desde la interfaz en Java para trabajar en un espacio libre de objetos o en CAD/CAM, el sistema de control trabaja de dos formas diferentes.

En el primer caso, en el cual no hay objetos en el espacio de trabajo del manipulador, el sistema de control evalúa todas las trayectorias que puede tomar

el robot para llegar al punto final, escogiendo sólo una, tomando como parámetro la velocidad con la cual llegan a la coordenada ingresada, según los cálculos descritos con anterioridad.

En el segundo caso, cuando el sistema se configura para trabajar con objetos en el espacio de trabajo, el control evalúa si el manipulador al realizar el desplazamiento para llegar al punto final colisiona con algún objeto. De ser así, evalúa otras posibilidades determinadas por el orden como se mueva cada articulación (seis posibilidades). Si el robot continúa estrellándose o el punto alcanzado está muy alejado de la coordenada ingresada por el usuario, se llama la red neuronal para que establezca una coordenada alternativa a la cual el robot debe moverse, para comenzar nuevamente con la evaluación desde este nuevo punto de referencia hasta el punto final (figura 7).

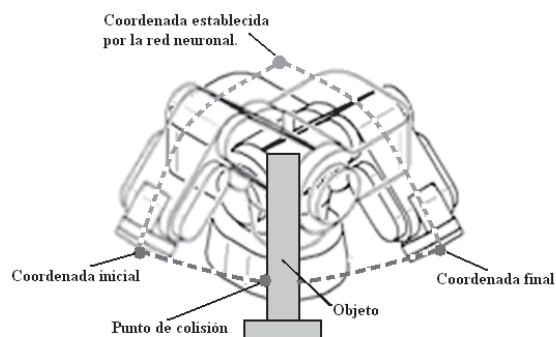


Figura 7. Trayectoria establecida por la red neuronal

Fuente: los autores

Como resultado se obtienen cuatro u ocho valores correspondientes a los ángulos de giro de cada articulación, definidos por el sistema de control cuando hace o no uso de la red neuronal para evitar un objeto.

### Diseño del software en Java

La plataforma que se utilizó para el desarrollo del software es Netbeans IDE 6.1, que integra las características de editor de código, compilador, depurador y constructor de interfaces gráficas, convirtiéndolo en una buena opción para la implementación del programa de simulación y control del manipulador (Deitel, 2004).

Por medio de la interfaz gráfica (figura 8), el usuario puede observar la configuración del robot en tres planos diferentes (X-Y, X-Z y Y-Z), ingresar los datos de las coordenadas finales a las cuales se quiere que el manipulador acceda, configurar los parámetros de comunicación y observar los ángulos obtenidos posteriormente a la evaluación de la red neuronal, que corresponden a los grados que debe girar cada eslabón para alcanzar la coordenada indicada.

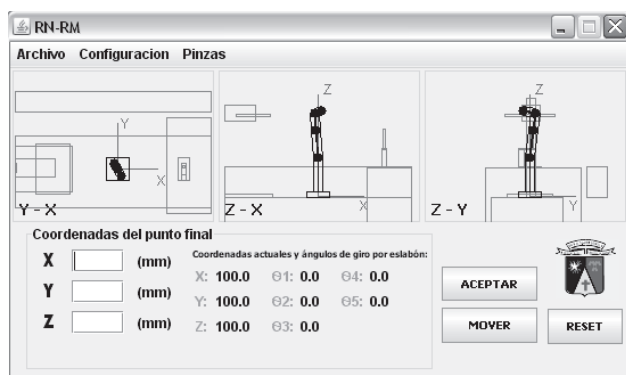


Figura 8. Interfaz gráfica realizada en Java  
 Fuente: los autores

## Conclusiones

El sistema de control logró determinar la trayectoria adecuada para que el manipulador alcanzara el punto deseado, obteniendo un porcentaje de falla muy pequeño (aproximadamente 5%, según las pruebas realizadas), fundamentado en la proximidad al punto final o la colisión con un objeto.

El error es mayor cuando la coordenada final está cercana a la banda transportadora o al módulo de ensamble hidráulico del laboratorio, debido a que son los únicos objetos que intervienen en la trayectoria del manipulador al mover los eslabones de forma vertical.

La fragmentación realizada al giro de cada eslabón influye en la colisión del manipulador, ya que los puntos no son evaluados en la totalidad de la trayectoria, lo que ocasiona que algunas coordenadas de un objeto no sean tenidas en cuenta para la evaluación.

Al realizar el modelamiento cinemático del manipulador se obtiene un espacio de trabajo que comprende 800.000 datos correspondientes al conjunto de coordenadas obtenidas del

fraccionamiento de giro de cada eslabón, logrando una representación adecuada del volumen de trabajo.

Con el sistema de control se evalúan diferentes trayectorias que puede seguir el manipulador, de acuerdo con la coordenada final ingresada por el usuario, y en las cuales se toman como parámetros la velocidad de llegada o los objetos presentes en el espacio de trabajo. En el primer caso, sin objetos en el área de trabajo, la respuesta del sistema de control toma de 30 a 40 segundos. En el segundo caso, teniendo en cuenta los objetos presentes en el espacio de trabajo, el sistema de control demora de 1,5 a 2 minutos en obtener una respuesta.

Debido a que el algoritmo de Denavit-Hartenberg representa cada eslabón como un eje y no como un objeto sólido, no se tiene en cuenta el grosor de cada articulación, por lo cual se genera un error al momento de comparar los puntos del manipulador con las coordenadas de los objetos presentes en el espacio de trabajo. Para esto se aumenta el tamaño de los objetos con el fin de compensar la distancia que hay desde el eje central hasta la superficie de cada eslabón, pero en algunas trayectorias aun se refleja este inconveniente.

El sistema de control implementado evalúa diferentes trayectorias que dependen del orden con el que se muevan los eslabones. En total son 24 posibilidades de desplazamiento, dentro de las cuales se escogen sólo 6 para disminuir el tiempo de procesamiento del programa. Estas combinaciones son seleccionadas de acuerdo con las pruebas realizadas durante el desarrollo del proyecto, con lo que se determinó que el porcentaje de colisiones disminuye cuando el manipulador se levanta para evitar un objeto.

Por medio del software desarrollado en Java se logra mostrar al usuario gráfica y numéricamente la respuesta del sistema de control. Asimismo, se brinda la posibilidad de configurar características importantes del manipulador como velocidad de desplazamiento, configuración del puerto serial, apertura y cierre de pinzas, y reinicio del manipulador.

Por las características del manipulador y la lógica de programación implementada, no es posible enviar la instrucción que permite ubicar el robot en la coordenada exacta, ya que se perdería la referencia en los ángulos y los próximos movimientos tendrían un error en su evaluación.

### Referencias

- Barrientos, A., Peñín, L., Balaguer C. y Aracil, R. (1997), *Fundamentos de robótica*, McGraw-Hill, Madrid.
- Deitel, H. M. (2004), *Cómo programar en Java*, México, Pearson.
- Dorf, R. C. (1989), *Sistemas modernos de control. Teoría y práctica*, Addison Wesley.
- Hilera, J. R. y Martínez, V. (1995), *Redes neuronales artificiales. Fundamentos, modelos y aplicaciones*, Madrid, RA-MA.