

CONTROL POSITION SYSTEM FOR A PLANE SURFACE BASED ON FUZZY LOGIC AND ASSEMBLER LANGUAGE

SISTEMA DE CONTROL DE POSICION PARA UNA SUPERFICIE PLANA BASADO EN LOGICA DIFUSA Y LENGUAJE ENSAMBLADOR

MSc. Wilson Javier Pérez Holguín, Ing. María Luisa Pinto Salamanca

Universidad Pedagógica y Tecnológica de Colombia

Grupo de Investigación en Robótica y Automatización Industrial "GIRA"
Escuela de Ingeniería Electrónica, Facultad Seccional Sogamoso, Colombia.

Telefax: (8) 7-705450 Ext. 222, Calle 4S No 15-134 Sogamoso

E-mail: wjperezh@yahoo.es, pintosml@gmail.com

Abstract: In this paper the results of the analysis, design, simulation and implementation of a control strategy for the position of an object on a plane surface with two grades of freedom are presented. The control strategy mixes fuzzy logic implemented on C++ and a rules base developed in assembler language. They were carried out simulations of the SLD (Fuzzy Logical System) in Unfuzzy 1.2 and in Matlab 6.5. Also, a specially designed table was built for the verification of the operation of the real system.

Resumen: En este artículo se presentan los resultados del análisis, diseño, simulación e implementación de una estrategia de control para la posición de un objeto sobre una superficie plana con dos grados de libertad. La estrategia de control mezcla lógica difusa implementada sobre C++ y una base de reglas desarrollada en lenguaje ensamblador. Se realizaron simulaciones del SLD (Sistema Lógico Difuso) en Unfuzzy 1.2 y en Matlab 6.5. También, se construyó una mesa especialmente diseñada para la verificación del funcionamiento del sistema real.

Keyword: Assembler language, C++ Programming, Fuzzy Logic.

1. INTRODUCCION

En muchos sistemas se requiere controlar la posición de un objeto dentro de una zona determinada. Algunos robots de servicio, por ejemplo, deben transportar materiales sobre una superficie de carga, que deben mantenerse en una posición de referencia durante todo el trayecto, aunque el robot atraviese terrenos irregulares.

En términos generales éste podría considerarse como un problema de péndulo invertido, el cual tiene múltiples aplicaciones en sistemas robóticos y de automatización.

Como una alternativa de solución para este tipo de problemas, se desarrolló una estrategia de control que mezcla la lógica difusa y una base de reglas adicional, para controlar el funcionamiento de dos motores que trabajan como actuadores principales en la corrección de la posición de una mesa, sobre la que se dispone un objeto que debe mantenerse en una posición de referencia determinada, a pesar de ser sometido a diferentes perturbaciones.

2. DESCRIPCION DEL SISTEMA

Se construyó una mesa apoyada sobre una rotula central que permitiera movimientos giratorios de una superficie plana (lamina rectangular) en dos dimensiones, (Ver Figura 1).

En dos puntos de la mesa (Ver Figura 2) se dispusieron tornillos sin fin acoplados a la misma mediante pequeñas rótulas. Cada tornillo a su vez está conectado al eje de un motor para conseguir desplazamientos verticales, y de esta forma obtener un movimiento bidimensional de la superficie plana.

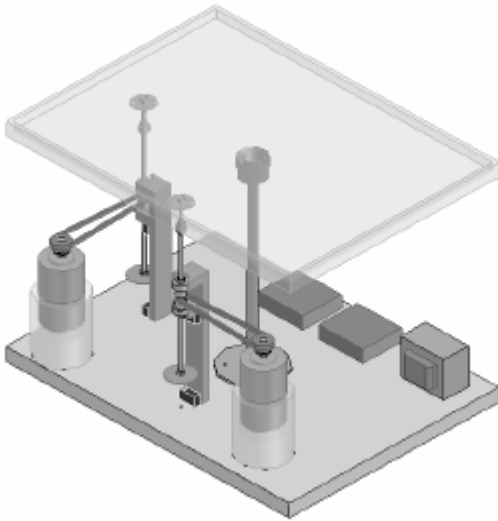


Fig. 1: Vista general del sistema

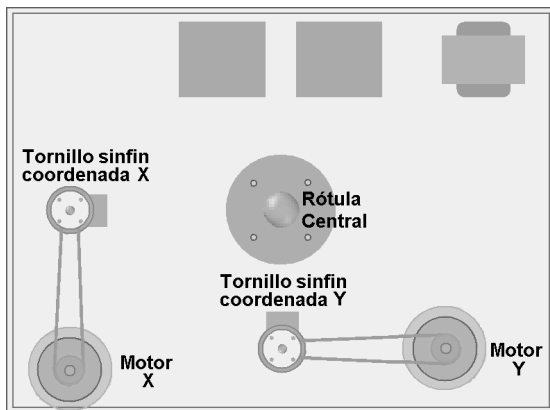


Fig. 2: Ubicación de los tornillo sin fin y los motores

Se utilizaron motores DC de 35V a 3A, gobernados por un sistema Push-pull con transistores Darlington que permite manejar su sentido de giro, conectados a un sistema electrónico que actúa como interfaz con un PC.

El objeto ubicado sobre la mesa, es un *mouse* inalámbrico-óptico, que además sirve como sensor para el sistema de control por entregar una medida de su posición directamente en coordenadas x-y. Dadas sus características ópticas, fue necesario adecuar la superficie para permitir su desplazamiento.

Se emplearon fines de curso para indicar cuando ocurre el máximo desplazamiento de los tornillos y así proteger los motores. En la base de la mesa se ubicaron las tarjetas de control y la fuente de alimentación del sistema.

3. ESTRATEGIA DE CONTROL EMPLEADA

La estrategia de control mezcla lógica difusa implementada sobre C++ y una base de reglas desarrollada en lenguaje ensamblador.

La lógica difusa es una rama de la inteligencia artificial que le permite a un computador analizar información del mundo real en una escala entre falso y verdadero. Cuando se carece de algoritmos que dictan cómo responde un sistema ante ciertas entradas, la lógica difusa puede controlar o describir un sistema usando reglas de sentido común que se refieren a cantidades indefinidas.

Los sistemas lógicos difusos usan la experiencia de las personas que normalmente interactúan con el sistema que se desea controlar (expertos) para formar un conjunto de reglas difusas. El valor medido en el sistema es convertido en un número difuso por medio de un difusor. Se definen los grados de pertenencia de éste número a varios conjuntos difusos de entrada. Mediante un motor de inferencia se establece el grado de pertenencia a un grupo de conjuntos difusos de salida. A través de un congresor se establece el valor real de salida que actuará como señal de control para la planta. En la figura 3 se observa la estructura general de un SLD.

En este caso, se ha implementado un SLD para determinar el tiempo de energización de cada uno de los motores de acuerdo con la distancia a la que se encuentre el objeto de su posición de referencia. Adicionalmente, se implementó un algoritmo en lenguaje ensamblador que permite el manejo de la interface de entrada/salida del sistema y la determinación del cuadrante en el que se encuentra el objeto a controlar.

La señal de retroalimentación, correspondiente a la coordenada x-y reportada por el *mouse*. Esta información ingresa al sistema mediante el uso de las instrucciones de llamada de funciones de la interrupción 33H, cuando la interfaz de usuario se realiza exclusivamente en lenguaje ensamblador, o mediante la edición del evento *OnMouseMove* del Formulario cuando la interfaz se hace en lenguaje C, sobre Builder C++ 6.0.

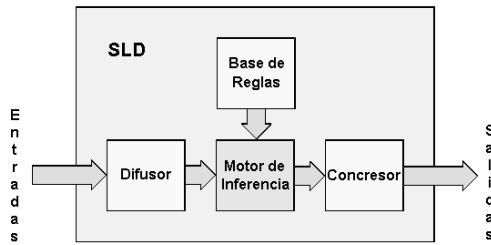


Fig. 3: Estructura general de un SLD

A partir de los datos correspondientes a la posición del *mouse* se identifica la zona en la que se encuentra. De esta forma se puede establecer si está en la región considerada como centro, cerca o lejos, y el cuadrante donde está ubicado. En la figura 4 se ilustra las zonas de trabajo definidas, sobre la mesa de prueba –superficie plana-, durante el desarrollo del proyecto.

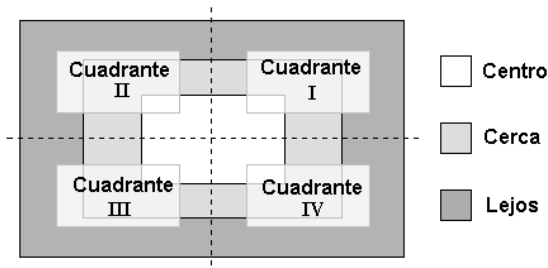


Fig. 4: Definición de regiones de trabajo

3.1 Sistema de Lógica Difusa

El SLD genera dos salidas que son empleadas para determinar el tiempo de encendido de cada uno de los motores, de acuerdo con la información de la posición en la que se encuentre el *mouse*.

Se definieron dos variables de entrada correspondientes a la posición x (denominada CoordenadaX) y a la posición y (denominada CoordenadaY), y dos de salida que indican el tiempo de energización de cada motor (Designadas en el SLD como EnergizaMotor X y EnergizaMotor Y).

Los conjuntos de entrada son mostrados en la figura 5. Se emplearon cinco conjuntos tipo triángulo que representan a las zonas LejosNegativo, CercaNegativo, Centro, CercaPositivo y LejosPositivo.

Los conjuntos de salida que simbolizan la energización de los motores son Poco, Medio y Alto. Estos también son tipo triángulo y se observan en la figura 6.

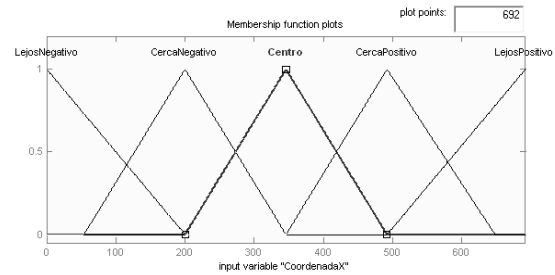


Fig. 5: Conjuntos de entrada para el SLD.

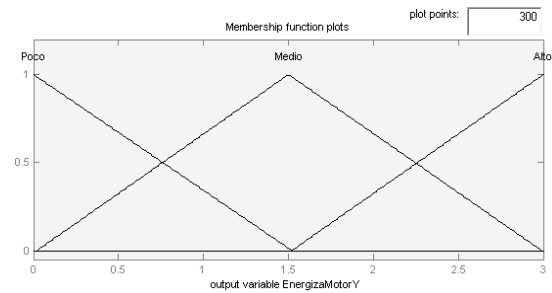


Fig. 6: Conjuntos de salida para el SLD

La superficie difusa muestra la relación entre las entradas, las salidas y la base de reglas definidas en el SLD (Ver figura 7).

Obsérvese como la superficie difusa correspondiente a la energización del motor X esta desfasada 90° respecto a la del motor Y, ambas mantienen la misma forma de “V” que indica una convergencia total del sistema hacia la zona centro.

El Sistema de Lógica Difusa fue desarrollado con los siguientes parámetros: inferencia tipo *Mandani*, método AND *Mínimo*, método OR *Máximo*, fusificador *Singleton*, defusificador *Centroide*, implicación *Mínimo*, agregación *Máximo* y una base de 25 reglas (Ver Tabla 1).

Tabla 1: Tiempo de energización de los motores

Motor X Motor Y		Coordenada Y				
		Lejos Negativo	Cerca Negativo	Centro	Cerca Positivo	Lejos Positivo
Coordenada X	Lejos Negativo	Alto	Alto	Alto	Alto	Alto
	Cerca Negativo	Medio	Medio	Medio	Medio	Medio
	Centro	Poco	Poco	Poco	Poco	Poco
	Cerca Positivo	Medio	Medio	Medio	Medio	Medio
	Lejos Positivo	Alto	Alto	Alto	Alto	Alto

La simulación, análisis y sintonización del sistema se hizo mediante la herramienta Fuzzy del Matlab 6.5, y el controlador en C++ se generó empleando Unfuzzy 1.2. Por último, en Builder C++ se vinculó el código de C++ obtenido con Unfuzzy con la interface usuario y el código assembler que comanda los motores. El código resultante es demasiado extenso, razón por la cual no se incluye en este artículo.

3.2 Algoritmo de Control en Ensamblador

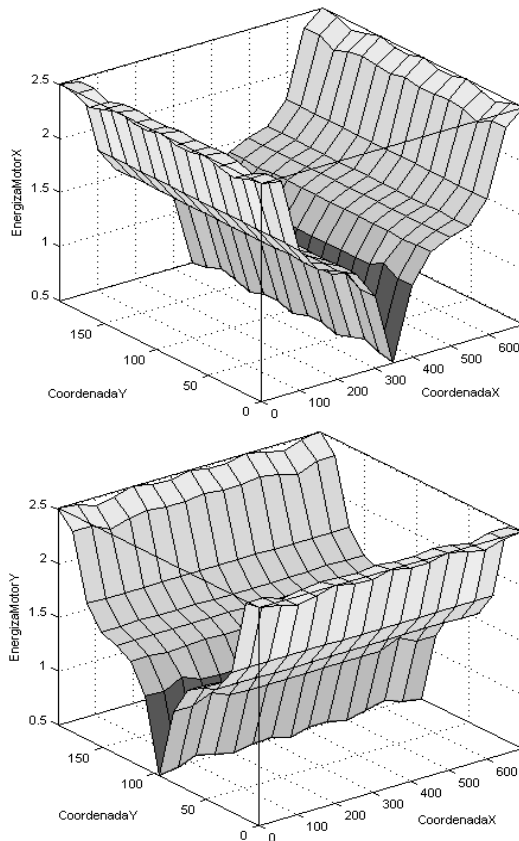


Fig. 7: Base de Reglas y Superficies Difusas

El algoritmo en ensamblador establece la secuencia de encendido de los motores y su sentido de giro. Esto se logra a partir de la coordenada (x,y) sensada a través del *mouse*, la cual permite determinar en que cuadrante se encuentra y, por lo tanto, cuales motores deben activarse. Ya que aquí solo se genera la orden de activación para cada motor y el sentido de giro, es el algoritmo difuso quien determina durante cuanto tiempo debe permanecer cada uno en este estado.

Como circuito de interfaz electrónica entre la planta (mesa) y el computador, se empleó un circuito PPI (interfaz periférica programable) configurado mediante el lenguaje ensamblador. A través del puerto B de la PPI en modo salida se manejaron tres señales para cada uno de los motores, correspondientes a energización y sentido de giro (encendido o apagado y derecha o izquierda).

La PPI está conectada al puerto paralelo del computador LPT1, empleando la dirección 0378H para datos y la 037AH para control. Debido a que en este puerto no existen líneas de control de lectura y escritura (\overline{RD} y \overline{WR}), las mismas deben generarse junto con las líneas A0 y A1 (selectores de registros) dentro del programa y ser enviadas a través del puerto 037AH. Debe prestarse atención al estado lógico de estas líneas, dado que algunas de ellas son de lógica negada ($\overline{C3}$, $C2$, $\overline{C1}$, $\overline{C0}$).

La figura 8 muestra la ventana de la interfaz de usuario diseñada en Builder C++. Sobre el *mouse* se observa la lectura de posición x-y y en la izquierda de la ventana se muestra el dato de tiempo de energización de los motores necesario para llevar el mouse a la zona centro. La energización del motor se encuentra en un intervalo de tiempo entre 0s y 3s. Este último corresponde al tiempo que demora el tornillo en realizar su máximo desplazamiento. La ventana de interfaz también incluye botones para inicializar y detener el control, generar un listado de datos e inicializar la PPI.

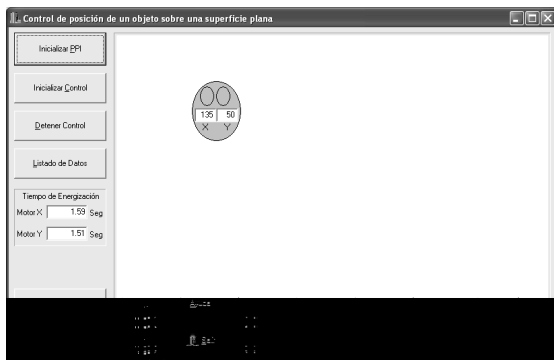


Fig. 8: Ventana de Interfase con el Usuario

6. CONCLUSIONES

Este proyecto demuestra la viabilidad de combinar estrategias de control de naturaleza diferente como lo son un Sistema de Lógica Difusa SLD, con un programa en lenguaje ensamblador. El primero contiene la dinámica del modelo del sistema y el segundo establece la secuencia de encendido de los motores y su sentido de giro.

El diseño de interfaces de usuario con Builder C++ y lenguaje ensamblador resulta altamente eficiente para el manejo de dispositivos entrada salida en bajo nivel, ya que evita el uso de librerías especiales de interfaz I/O diseñadas para C++. La idea es obtener un ambiente gráfico amigable y atractivo para el usuario, el cual se desarrolla de forma fácil y rápida en Builder C++, y confiar el manejo de señales de bajo nivel al lenguaje ensamblador, que se desenvuelve de manera excelente para este tipo de aplicaciones.

Teniendo en cuenta que la señal de activación de cada motor es ON-OFF, fue necesario definir una señal de control que considerara la posición inicial del cursor en la mesa y las acciones a realizar de acuerdo con la nueva posición a alcanzar. Esta señal de control se relaciona con el tiempo de aplicación de una señal de activación a cada motor, y la especificación de su sentido de giro, mediante el uso de la PPI.

El tipo de difusor, conector, método de inferencia, implicación, AND, OR, agregación y el grado de certeza que se tiene sobre cada regla en el SLD, determinan el correcto funcionamiento del sistema de control, y su escogencia está estrechamente relacionada con las características del sistema a controlar.

El número y tipo de conjuntos de entrada y salida, debe escogerse de forma adecuada para que se modele en ellos la dinámica propia del sistema que se desea controlar. Los resultados son significativamente diferentes si se emplean otros tipos de conjuntos o si se distribuyen de forma diferente a la planteada.

REFERENCIAS

- Delgado, A. *Inteligencia Artificial y mini-robots*. Bogotá: ECOE Ediciones, 1998. ISBN: 958-648-155-7.
- Corzo, J. *La Lógica Difusa*. <http://personales.ya.com/casanchi/mat/difusa01.htm> 13 de Junio de 2006.
- Galindo G. J. *Lógica Difusa y Aplicaciones: Conjuntos y Sistemas Difusos*. Departamento de lenguajes y Ciencias de la Computación. Universidad de Malaga. <http://www.lcc.uma.es/~ppgg/FSS/FSS1.pdf>. 10 de Junio de 2006.
- Jang, J.-S.R., Sun, C.-T. and Mizutani, E. *Neuro – Fuzzy an Soft computing*. Prentice Hall. ISBN: 0-13-261066-3
- Pérez H. G. *Sistemas de Lógica Difusa*. Universidad Nacional de Colombia, Sede Bogotá. 2001.