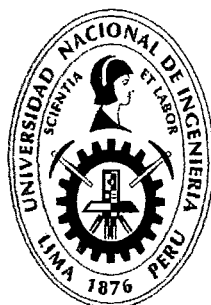


**UNIVERSIDAD NACIONAL DE INGENIERÍA**  
**FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**



**“DISEÑO Y SIMULACIÓN DE UN SISTEMA DE CONTROL NO LINEAL  
MULTIVARIABLE POR LÓGICA DIFUSA APLICADO A UN  
MANIPULADOR ROBÓTICO TRANSLACIONAL DE 2DOF”**

**TESIS**

**PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN CIENCIAS  
CON MENCIÓN EN AUTOMÁTICA E INSTRUMENTACIÓN**

**ELABORADO POR**

**JACOB ASTOCONDOR VILLAR**

**ASESOR**

**M.Sc. ING. RAUL NICANOR BENITES SARAVIA**

**LIMA - PERÚ**

**2012**

**Digitalizado por:**

**Consortio Digital del  
Conocimiento MebLatam,  
Hemisferio y Dalse**

A mis hijos:  
Jacob, y Lizbhet

## **Agradecimiento**

Quisiero expresar mi agradecimiento el apoyo total que recibí de mis profesores y personal administrativo de la Sección de Postgrado de la FIEE-UNI, y en particular a mi asesore al MSc. Raúl Benites Saravia por sus enseñanzas y sugerencias para este trabajo de tesis. Mi agradecimiento al Instituto de Investigación de la FIEE por su apoyo en el desarrollo de esta tesis

UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad de Ingeniería Eléctrica y Electrónica

DISEÑO y SIMULACIÓN DE UN SISTEMA DE CONTROL NO LINEAL  
MULTIVARIABLE POR LÓGICA DIFUSA APLICADO A UN MANIPULADOR  
ROBÓTICO TRANSLACIONAL DE 2DOF

TESIS

Para optar el grado de Maestro en Ciencias

Mención: Automática e Instrumentación

Presentada por:

JACOB ASTOCONDOR VILLAR

Lima - Perú

**Extracto**

La presente tesis trata sobre el control de trayectoria de un manipulador robótico translacional multivariable de 2DOF (Two-Degrees-of-Freedoms) que consta de un móvil accionado por una polea y un eslabón articulado en el CG (centro de gravedad) de dicho móvil. Este proceso será controlado mediante la técnica de control fuzzy. La acción de control está orientada a controlar el movimiento translacional del móvil y el movimiento angular del brazo que es libre de girar en ambas direcciones. Las metas impuestas en la tesis son: diseño, modelado, y simulación del sistema, controlado con la ley de control fuzzy. Los sistemas convencionales de control son diseñados tradicionalmente usando modelos matemáticos de sistemas físicos para posteriormente aplicar técnicas de diseño para obtener controladores apropiados para el sistema. Sin embargo, en la realidad, el modelo y sus parámetros son con frecuencia desconocidos. Esto se debe a cambios en el ambiente de trabajo, dinámicas no modeladas y la presencia de no linealidades e incertidumbres que son difícil de tratar con los controladores convencionales los cuales no siempre son capaces de aprender o de adaptarse a nuevas situaciones. A continuación se plantea este problema y la solución del mismo utilizando la técnica de lógica difusa. Actualmente las técnicas avanzadas de control basadas en modelos, como son el control predictivo, el control

por modos deslizantes, el control adaptable, entre otras, están siendo combinadas con técnicas de control basadas en los sistemas difusos.

Universidad Nacional de Ingeniería  
Facultad de Ingeniería Eléctrica y Electrónica

Design and Implementation of a System of Control Not Lineal Multivariable for  
Fuzzy Logic Applied a Translational Robotic Manipulator of 2DOF

THESIS

Requirement for the degree of Master in Science  
Major: Automatic and Instrumentation

by:

JACOB ASTOCONDOR VILLAR

Lima - Perú

**Abstract**

The present thesis deals with on the control of fuzzy of a multivariable translational robotic manipulator of 2DOF (Two-Degrees-of-Freedoms) that consists of a motive worked for a pulley and a link articulated in the CG (center of gravity) of this motive. Such a process will be controlled by means of the technique of control fuzzy. The control action is guided to control the movement translational of the motive and the one angular movement of the arm that is free of rotating in both address. The goals imposed in the thesis are: modelling, disign and simulation of the system with the law of control fuzzy. This control law uses the discontinuous non lineal feedback in a given surface that it belongs to the space of the system.

# Tabla de Contenido

## Capítulo 1.

<b>Introducción</b>	1
1.1. Formulación del Problema . . . . .	1
1.2. Perspectiva Histórica . . . . .	3
1.3. Organización de la Tesis . . . . .	4
1.4. Objetivos generales . . . . .	4

## Capítulo 2.

<b>Lógica Difusa y Control Difuso</b>	6
2.1. Introducción . . . . .	6
2.2. Teoría de Conjuntos difusos . . . . .	7
2.2.1. Variables lingüísticas . . . . .	10
2.2.2. Proposiciones difusas . . . . .	12
2.2.3. Reglas de Inferencia o Razonamiento . . . . .	14
2.3. Control Difuso . . . . .	15
2.4. Fuzzyficación . . . . .	17
2.5. Defuzzyficación . . . . .	17
2.5.1. Tipos de Controladores Difusos . . . . .	18

## Capítulo 3.

<b>Modelado de la Planta</b>	21
3.1. Descripción del Sistema . . . . .	21
3.2. Modelo de Lagrange del Proceso BRTM . . . . .	22
3.2.1. Ecuaciones de Energía para el Carro . . . . .	22
3.2.2. Ecuaciones de Energía para el Brazo . . . . .	23
3.3. Ecuaciones de Lagrange del Proceso BRTM . . . . .	24
3.3.1. Modelo del Servomotor DC $M_1$ - Polea . . . . .	26
3.3.2. Modelo del Servomotor DC $M_2$ - Brazo . . . . .	28
3.3.3. Modelo Lagrangiano . . . . .	30
3.4. Respuesta de la Planta al Escalón Unitario . . . . .	31

## Capítulo 4.

<b>Diseño del Control Fuzzy Tiempo Continuo</b>	36
4.1. Introducción . . . . .	36
4.2. Diseño del controlador . . . . .	36
4.2.1. Controlador difuso de posición . . . . .	41
4.2.2. Controlador de posición angular ( brazo) . . . . .	47

4.2.3. Compensador de gravedad . . . . .	54
<b>Capítulo 5.</b>	
<b>Diseño del control difuso discreto</b>	56
5.1. Discretización del sistema continuo . . . . .	56
5.2. Control difuso de posición lineal del carro y de posición angular del brazo . . . . .	58
5.2.1. Evaluación del controlador fuzzy discreto . . . . .	61
<b>Capítulo 6.</b>	
<b>Resultados</b>	63
6.1. Resultados parciales . . . . .	63
6.1.1. Resultados del sistema de control difuso en tiempo continuo en lazo cerrado . . . . .	63
6.1.2. Resultados del sistema de control difuso en tiempo discreto en lazo cerrado . . . . .	65
6.1.3. Señales de los sensores para el desplazamiento lineal y angular del Brazo robótico . . . . .	68
6.2. Resultados Finales . . . . .	68
<b>Capítulo 7.</b>	
<b>Discusión de Resultados</b>	72
7.1. Contratación de hipótesis con los resultados . . . . .	72
7.2. Contratación de resultados con otros estudios similares . . . . .	72
7.2.1. Experimento1 . . . . .	73
7.2.2. Experimento2- Sistema con control fuzzy . . . . .	74
7.2.3. Experimento3-Sistema con control fuzzy con condiciones ini- ciales diferente de cero . . . . .	77
<b>Capítulo 8.</b>	
<b>Conclusiones y Recomendaciones</b>	79
8.1. Conclusiones . . . . .	79
<b>Bibliografía.</b>	82
<b>Apéndice A.</b>	
<b>Identificación de Parámetros</b>	84
A.1. Identificación de Parámetros del Servomotor . . . . .	84
A.2. Curva de la Velocidad para la Identificación . . . . .	85
A.3. Coeficiente de Fricción del Carro . . . . .	88
A.4. Coeficiente de Fricción Viscosa del Brazo . . . . .	90
A.5. Estimación de los parámetros físicos de un motor de corriente directa usando el algoritmo de Steiglitz- Mcbride Considerando la Inductan- cia $L_a$ . . . . .	95
A.5.1. Estimación de los parámetros del motor de corriente directa. .	95
<b>Apéndice B.</b>	
<b>Programas en MATLAB</b>	99



B.1. Listado de Programas . . . . .	99
B.1.1. Simulación de la planta en lazo abierto . . . . .	99
B.1.2. Simulación del sistema de control en tiempo continuo . . . . .	101
B.1.3. Simulación del sistema de control en tiempo discreto . . . . .	106
B.1.4. Simulación de los sensores de los motores1 y motor2 . . . . .	112
<b>Apéndice C.</b>	
<b>Programas en C</b>	116
C.1. Listado de Programas . . . . .	116

# Capítulo 1

## Introducción

Muchas aplicaciones de manipuladores robóticos sólo tratan el problema del control de posición; sin embargo, existen aplicaciones relacionadas con el seguimiento de trayectorias. Las estrategias de control de posición en general se emplean para trasladar la mano del manipulador de un lugar a otro. Las estrategias de control de trayectoria también se emplean para el traslado de la mano, pero este traslado se realiza cumpliendo ciertas especificaciones de diseño (por ejemplo, el tiempo de establecimiento del sistema de posicionamiento, o la precisión), las cuales no son fáciles de conseguir con las estrategias de control de posición (empleando por ejemplo controladores proporcionales derivativos).

En este trabajo de tesis el problema a resolver es diseñar e simular un sistema de control por lógica difusa, para controlar el seguimiento de trayectorias arbitrarias mediante un manipulador robótico translacional de dos grados de libertad. Las variables a controlar son las trayectorias del carro y del brazo del manipulador. El sistema de control debe ser capaz de hacer que las variables controladas sigan la evolución de las trayectorias de referencia arbitrarias con mínimo sobreimpulso, mínimo tiempo de estabilización y error en estado estable nulo.

El control difuso está fundamentado en la teoría de conjuntos difusos y en la lógica difusa, que permite convertir una estrategia de control lingüística, expresado en el formato de reglas difusas "si - entonces", en una estrategia de control automático. El diseño de esta estrategia de control generalmente se aborda desde una perspectiva de ensayo y error con alto grado de heurística. En esta tesis, se diseña un controlador, que interprete en forma intuitiva (y numéricamente exacta) el estado de ciertas variables, y en base a ello deduzca en forma lógica una actuación que permite llevar la variable al estado deseado.

### 1.1. Formulación del Problema

El problema que se estudia en esta tesis es el control simultáneo de la posición angular del brazo y la posición de translación del carro, siendo resuelto por medio

de la técnica de control por lógica difusa. Para esto se requiere que el manipulador robótico realice el seguimiento de trayectorias de referencia arbitrarias empleando la fuerza de control adecuada. Por consiguiente, el proceso a controlar posee dos entradas y dos salidas, es decir, es un proceso MIMO (Multiple-Input-Multiple-Output).

El manipulador robótico translacional de 2DOF, es un proceso multivariable que posee un eslabón (el brazo del robot) articulado al eje del motor DC. Este motor está montado sobre un carro que desarrolla su movimiento de translación por la acción de dos poleas. Una de las poleas está acoplada al eje de un segundo motor DC. Este último motor va a gobernar el movimiento del carro.

El sistema manipulador translacional de 2DOF, diseñado tomando como modelo al sistema de péndulo invertido. La finalidad de diseñar el sistema es para implementar técnicas de control avanzado, en nuestro caso la implementación de un controlador por lógica difusa.

Fue necesario modelar la dinámica de la planta mediante las ecuaciones de energía de Lagrange. El modelo requiere de la identificación de los parámetros del servomotor DC cuyos resultados experimentales son presentados en el apéndice A.

la simulación del proceso se realiza haciendo uso de matlab tanto para el sistema en tiempo continuo y tiempo discreto.

Se propone la simulación del proceso consignado el uso de un procesador, de elevada potencia PIC18F4550 que se encarga de manejar los datos de entradas digitales y dos salidas generando la señal de PWM, las que a su vez se requiere de un amplificador de potencia (circuito H), cuyo diseño tiene por finalidad manejar una lógica de control evitando que las entradas simultáneas procedentes del generador de *PWM* sean de igual amplitud, siendo dañinas para el funcionamiento de los servomotores DC.

## 1.2. Perspectiva Histórica

En 1965 Lofti A. Zadeh escribió su celebre artículo Fuzzy Sets [52] en el que menciona que "muy comúnmente, las clases de objetos encontradas en el mundo físico real no tienen definido de forma precisa su criterio de pertenencia". Esta misma ambigüedad crece en el caso de un número, tal como el 10 con relación a la clase de todos los números reales que son mucho más grandes que 1. Podemos observar con claridad que "las clases de todos los números reales mucho más grandes que 1", o "la clase de las mujeres altas", o "la clase de los hombres jóvenes", no constituyen clases o conjuntos en el sentido matemático usual. Sin embargo, los datos restantes, imprecisamente llamados clases, desempeñan un papel importante en el pensamiento humano, particularmente en el dominio de reconocimiento de patrones, comunicación de información y abstracción. Zadeh estableció en su artículo las bases principales del lenguaje difuso, definiendo el universo de discurso, como el área dónde se atribuyen cualidades o se califican determinadas variables, y llamando a un tipo de conjuntos que encierran una propiedad específica, funciones de pertenencia. A partir de los primeros resultados presentados por Zadeh y las primeras aplicaciones en los procesos de control realizadas por Mamdani, el control difuso ha mostrado ser una aproximación exitosa para los sistemas no lineales complejos. Desde que las teorías de control clásicas se han combinado dentro de los sistemas difusos, ha surgido un gran número de análisis de estabilidad y resultados de síntesis en el control.

El problema de control de movimiento para los robots, ha llamado la atención durante la última década, en donde los robots jugaban un papel importante. Actualmente se ha visto incrementada su utilización, debido a la flexibilidad de interactuar en mundos relativamente complejos. Las aplicaciones en robótica han ido mejorando desde tareas limitadas como: aplicaciones a baja velocidad, tareas de baja precisión hasta operaciones de alta velocidad.

Los manipuladores robóticos presentan un sistema complejo de articulaciones que interactúan con una dinámica no lineal. A medida que la velocidad de los manipuladores se incrementa la cantidad de interacciones entre sus articulaciones complican el problema de control.

Las estrategias de control son limitadas cuando se utilizan técnicas de control convencionales siendo confiables cuando presentan un funcionamiento a baja velocidad, dónde la interacción de la dinámica es mínima y el modelo puede ser eficazmente linealizado y desacoplado. Sin embargo, las aplicaciones futuras involucran altas velocidades, y para mejorar la eficiencia y rango de operación del manipulador, se debe de considerar métodos de control más sofisticados, siendo uno de ellos el método de control por lógica difusa

### **1.3. Organización de la Tesis**

La tesis está organizada como sigue:

El Capítulo 1 trata sobre la introducción de la tesis el capítulo 2 introduce los conceptos básicos de Lógica Difusa y Control difuso.

El capítulo 3 comienza con la descripción del modelado de la planta no lineal y se plantea las ecuaciones dinámicas mediante el desarrollo de ecuaciones de Lagrange en términos de las energías cinética, potencial y disipativa. Se simula la respuesta de la planta debido a un escalón unitario, usando los parámetros de los servomotores DC.

El capítulo 4 formula el problema de diseño del controlador para la planta no lineal en tiempo continuo

El capítulo 5 formula el problema de diseño del controlador para la planta no lineal en tiempo discreto.

El capítulo 6 y capítulo 7 Se muestran los resultados y discusión de los resultados. El capítulo 8 conclusiones y recomendaciones. Finalmente la Bibliografía

### **1.4. Objetivos generales**

El propósito de esta tesis es desarrollar un sistema de control por lógica difusa que resuelva el problema de control propuesto en la sección anterior. Los objetivos generales de esta investigación son:

- Diseñar una ley de control por lógica difusa que resuelva el problema de control propuesto.
- Obtener como respuesta del sistema controlado, un tiempo de establecimiento

corto, un mínimo sobreimpulso, un error en estado estable muy cercano a cero.

- Aplicar la ley de control al sistema y verificar los resultados.

## Capítulo 2

# Lógica Difusa y Control Difuso

Se presentan los conceptos básicos que describen a la lógica difusa y a los sistemas de control difuso.

### 2.1. Introducción

El desarrollo de controladores difusos ha emergido como una de las más prosperas y fructíferas aplicaciones de la teoría de conjuntos difusos. Esta teoría fue postulada por Lotfi Zadeh hace ya más de dos décadas como una generalización de los conjuntos clásicos. Ya en los trabajos seminales, introduce la idea de formular el problema de control mediante el uso de reglas expresadas con representaciones lingüísticas. Las experiencias diarias de la vida real nos dan muchos ejemplos donde se confirma como el entendimiento, el pensamiento y la habilidad humana pueden resolver eficientemente el problema de control para una gran variedad de sistemas sin hacer uso de los sofisticados algoritmos de la teoría de control. Es quizás por eso, que buscando emular las capacidades humanas de control, los primeros trabajos en control difuso fueron desarrollados desde el punto de vista de la inteligencia artificial y no bajo la tutela de la teoría de control. Mamdani y sus colegas son los primeros en aplicar la lógica difusa al control de procesos, particularmente, en un sistema de control para una máquina de vapor. Desde entonces, el número de aplicaciones de los controladores difusos ha ido en aumento, no sólo para el control de procesos, sino prácticamente en todos los campos de la ingeniería. Desde el punto de vista de las aplicaciones pueden diferenciarse dos periodos. En el primero, los esfuerzos se concentraron en experimentos de nivel laboratorio y en el diseño de prototipos, siendo relativamente raros los desarrollos de nivel industrial.

Durante el segundo periodo, iniciado a finales de los años ochenta, han surgido un gran número de aplicaciones comerciales en lo que se conoce como electrónica de consumo (lavadoras, secadoras, cámaras de video, etc.) así como en aplicaciones industriales de todo tipo aunque con menor auge que en las comerciales. De manera

muy notoria ha sido en Japón donde el control difuso ha sido impulsado con mayor fuerza. En este país la lógica difusa se ha convertido en palabra de uso común en respuesta al éxito obtenido al aplicarse y comercializarse en todo tipo de aparatos electrodomésticos e incluso en algunos proyectos industriales.

Los controladores difusos tienen potencialmente un gran número de ventajas cuando las especificaciones del control requieren robustez, adaptabilidad y flexibilidad debido a perturbaciones del entorno o a efectos no modelables de la dinámica del sistema. Los controladores difusos son básicamente controladores basados en reglas que discretizan el espacio de operación continuo, no lineal y multidimensional en clases discretas. Utilizando lógica multivaluada, las variables difusas de las reglas preservan la información cuantitativa en los correspondientes valores de clase y el valor de pertenencia a esa clase. El comportamiento del controlador difuso es optimizado mediante una máquina de estados finitos en términos de las clases discretas y finalmente usa la información de la función de pertenencia para hacer una interpolación suave entre los puntos vecinos del espacio de operación continuo.

No es fácil responder a la pregunta de cuando aplicar, o no, la tecnología de control difuso. Por un lado es una tecnología que actualmente se encuentra en expansión tanto en sus bases teóricas como en sus aplicaciones, no conociéndose muy bien sus límites, por el otro, el número de aplicaciones es aún insuficiente para poder generalizar los resultados alcanzados. Con el objetivo de intentar responder a esta pregunta, en los apartados siguientes expondremos los principales beneficios y desventajas que actualmente exhibe la tecnología de control difuso.

En este capítulo se introducen algunos de los fundamentos básicos la lógica difusa, así como, el control difuso y los términos de fuzzyficación y defuzzyficación. También se habla de la aproximación de funciones, esto es, de la utilización de la lógica difusa y más específicamente del control difuso en la aproximación de funciones.

## **2.2. Teoría de Conjuntos difusos**

Un conjunto clásico es una colección de objetos de cualquier clase. El concepto de un conjunto se ha vuelto una de las nociones más fundamentales de la Matemática. La denominada Teoría de Conjuntos se fundó por el matemático alemán George



Cantor (1845-1918) [5]. En la teoría de conjuntos la noción de *conjunto* y *elemento* es primitiva. Ellos no se definen por lo que se refiere a otros conceptos.

Un conjunto se especifica totalmente por los elementos que contiene. Permitiendo  $A$  ser un conjunto,  $x \in A$  significa que  $x$  es un elemento del conjunto  $A$  y  $x \notin A$  significa que  $x$  no pertenece al conjunto  $A$

**Nota1.** En la Teoría de la Lógica Difusa los conjuntos no difusos son llamados conjuntos **crisp** y sus elementos o valores son llamados de igual manera valores **crisp**, y de esta manera se hace referencia a dichos conjuntos clásicos. Para diferenciar a los conjuntos clásicos de los conjuntos difusos se coloca una tilde sobre la letra que indica el nombre del conjunto difuso.

Una manera de definir un conjunto es enumerar sus elementos y otra es usar un predicado  $P(x)$ , esto significa que cada elemento  $x$  del conjunto tiene una propiedad  $P$ . Una tercera manera es la definición de un conjunto  $A$  usando su función característica  $\mu_a$ , que permite definir  $A$  en el dominio  $X$ , el cual es un conjunto clásico con los elementos  $1$  y  $0$ .

**Definición 1:**  $\mu_a \rightarrow \{0, 1\}$  es una función característica del conjunto  $A$  si y solo si para toda  $x$

$$\mu_a(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

La función característica de un conjunto difuso asigna un valor entre 0 y 1 a cada uno de los elementos del conjunto universal (conjunto que contiene a todos los elementos del discurso, es un término relativo). Esta función puede generalizarse tal que los valores asignados a los elementos del conjunto de discurso (llamado también universo local, el cual es el marco de referencia dentro del cual se está trabajando) se encuentre dentro de un rango específico e indiquen el número de miembros de estos elementos en el conjunto de discurso. Los valores grandes denotan los grados más altos de pertenencia del conjunto. Tal función se llama función de pertenencia (FP), y el conjunto definido por ésta, "conjunto difuso".

El rango normalmente usado de valores de funciones de pertenencia es el intervalo  $[0, 1]$ . En este caso, cada función de pertenencia mapea los elementos del conjunto universal  $U$ , dentro los números reales  $[0, 1]$ . El conjunto que se define con base en

una función de pertenencia se le da el nombre de conjunto difuso.

**Definición 2** La función de pertenencia  $\mu_{\tilde{A}}$  de un conjunto difuso  $\tilde{A}$  es una función

$$\mu_{\tilde{A}} : U[0, 1].$$

Así, cada elemento  $x$  de  $U$  tiene un grado pertenencia  $\mu_{\tilde{A}}(x) \in [0, 1]$  esta completamente determinada por el conjunto de duplas

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in U\} \quad (2.1)$$

La notación comúnmente empleada en la literatura para denotar las funciones de pertenencia es la mostrada en la definición anterior.

Según la notación, el símbolo del conjunto difuso  $\tilde{A}$  es distinguido por su función de pertenencia  $\mu_{\tilde{A}}$ .

Así, cada elemento  $x \in U$  tiene un grado de pertenencia  $\mu_{\tilde{A}} : U \rightarrow [0, 1]$ .  $\tilde{A}$  es completamente determinado por el conjunto de duplas de la ecuación (2.1). El grado de pertenencia se cuantifica mediante la función de pertenencia  $\mu_{\tilde{A}}$ . como sigue:

Si  $\mu_{\tilde{A}}$  indica la función de pertenencia de  $x$  al conjunto  $\tilde{A}$ , entonces:

$\mu_{\tilde{A}}(X)$  está entre 0 y 1.

Si  $\mu_{\tilde{A}}(x) = 1$ ,  $x$  pertenece totalmente a  $\tilde{A}$

Si  $\mu_{\tilde{A}}(x) = 0$ ,  $x$  no pertenece a  $\tilde{A}$

Existen diferentes formas de una función de pertenencia. Las más importantes se presentan en la figura 2,1

Aquí solo se describirá la función de pertenencia del tipo **gaussiana**. Una función de pertenencia gaussiana se especifica con dos parámetros  $c, \rho$ :

- $c$ , Representa el centro de la función de pertenencia.
- $\rho$ , Determina su anchura.

$$gauss(x, c, \rho) = e^{-\frac{1}{2}\left(\frac{x-c}{\rho}\right)^2} \quad (2.2)$$

**Ejemplo .1** Supóngase que alguien quiere describir la función  $\sin(x)$ , cuando la variable  $x$  se encuentra cercana a  $\pi$ . Utilizando un conjunto difuso con los elementos  $(x, \mu_{\tilde{C}_\pi}(x))$ , se proponen las siguientes duplas

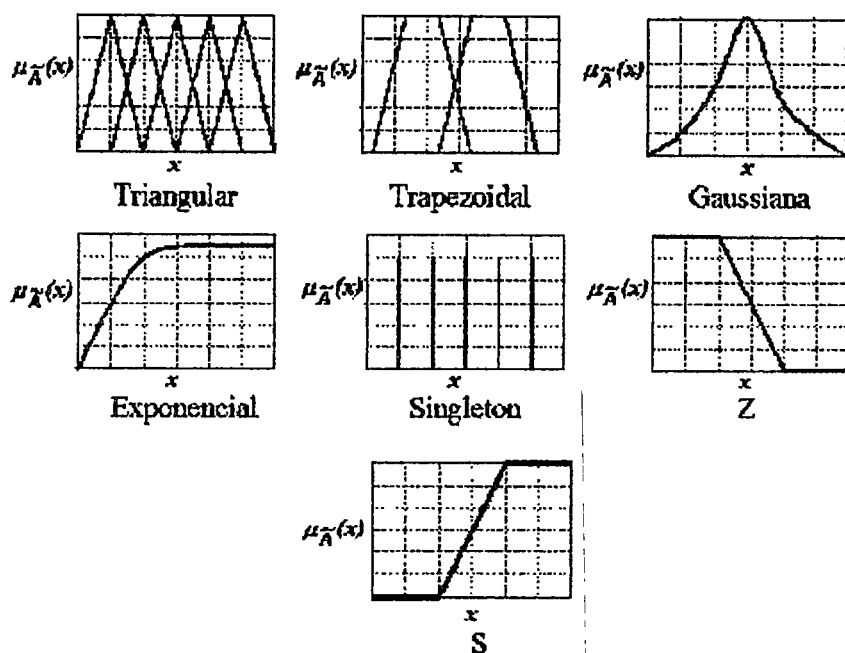


Figura 2.1: Funciones de pertenencia más comunes.

$$\{\tilde{C}_\pi = \{(0, 0,072), (\frac{\pi}{2}, 0,291), (\pi, 1), (3\pi/2, 0,291), (2\pi, 0,072)\}\} \quad (2.3)$$

Es decir, el punto 0 tiene un grado de pertenencia de 0.072, el punto  $\frac{\pi}{2}$  de 0.0291, el punto  $\pi$  de 1 etc. Para este ejemplo se utilizó una función de pertenencia (*FP*) del tipo gaussinana (con centro en  $\pi$  y de anchura 1) la cual es obtenida utilizando la ecuación (2.2), y se ilustra en la figura 2.2. Si el valor de  $x$  es menor a cero o mayor a  $2\pi$ , entonces, el grado de pertenencia es cero ya que no esta dentro del rango de la variable que se quiere describir, es decir, no esta cerca de  $\pi$ .

### 2.2.1. Variables lingüísticas

La unidad de representación fundamental del conocimiento en el razonamiento aproximado es la noción de variable lingstica. Una variable lingstica es aquella en la cual sus valores son palabras o sentencias en un lenguaje natural o artificial. Por ejemplo, la edad es una variable lingüística si sus valores son lingsticos más que numéricos, es decir, joven, no tan joven, muy joven, bastante joven, viejo, no muy viejo, etc., más que valores numéricos como 21 años, 22 años, 23, etc.

En el razonamiento aproximado es usual tener la siguiente estructura asociada con

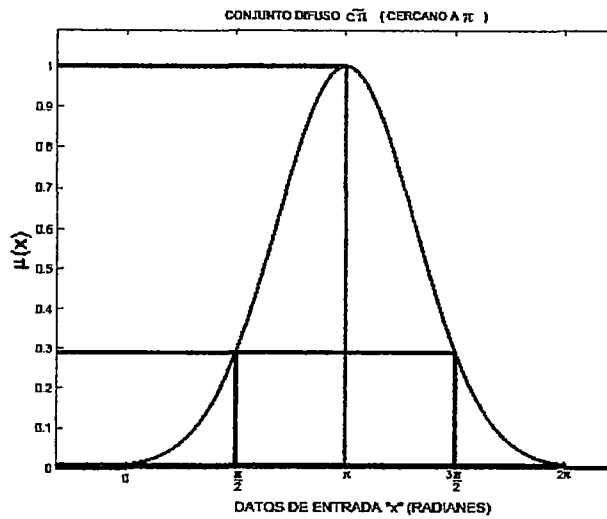


Figura 2.2: Funciones de pertenencia para una función seno.

la noción de variable lingüística.

$$\langle X; L_X; D_X; M_X \rangle \tag{2.4}$$

Donde :

- $X$  Denota la variable lingüística, por ejemplo, la edad, temperatura, altura, etc.
- $L_X$  Es el conjunto de los valores lingüísticos que puede tomar la variable  $X$ . (Un valor lingüístico denota un símbolo para una propiedad en particular de  $X$ ).
- $D_X$  es el dominio sobre el cual la variable lingüística  $X$  toma sus valores cuantitativos.
- $M_X$  es una función semántica que proporciona una interpretación del valor lingüístico en términos de los elementos cuantitativos de  $D_X$ , es decir

$$M_X : L_X \rightarrow \widetilde{L}_X \tag{2.5}$$

Donde :

$\widetilde{L}_X$  es un conjunto difuso definido sobre  $D_X$

**Ejemplo .2** Siguiendo el ejemplo (1) se define la variable lingüística  $X$  que puede tomar los siguientes valores lingüísticos:

$$L_X = \{\text{cercano a cero}(CC), \text{cercano a } \frac{\pi}{2}(C\frac{\pi}{2}), \text{cercano a } \frac{3\pi}{2}(C\frac{3\pi}{2}), \text{cercano a } 2\pi(C2\pi)\} \quad (2.6)$$

El dominio  $D_X$  sobre la variable lingüística  $X$  toma sus valores en el intervalo  $[0, 2\pi]$ .

## 2.2.2. Proposiciones difusas

El razonamiento aproximado se usa para representar y razonar con conocimiento expresado en "Proposiciones Atómicas", las cuales son expresadas en un lenguaje de forma natural.

La información puede venir dada en forma de sentencias o proposiciones atómicas de la forma:

$$"X \text{ es } A"$$

donde  $X$  es el nombre de un objeto (atributo, hecho...,) y  $A$  es el valor que toma este objeto.

Las Proposiciones Atómicas pueden tomar valores de verdad dentro de un conjunto definido de valores posibles. Esto implica la existencia de distintas lógicas, clasificadas por el número de valores de verdad posibles: lógica bivaluada (two-valued logic), trivaluada (threevalued logic), ..., multivaluada (many-valued logic). Las proposiciones con atributos imprecisos conllevan al uso de lógica multivaluada o lógica difusa (fuzzy logic).

La diferencia fundamental entre las proposiciones clásicas y las proposiciones difusas están en el rango de sus valores de verdad. Mientras que las proposiciones clásicas corresponden a la lógica bivaluada (verdad o falsedad), la verdad o falsedad de proposiciones difusas son una cuestión de grado. Asumiendo que esta verdad y falsedad son expresados por los valores 1 y 0, respectivamente, el grado de verdad de cada proposición difusa se expresa por un número en el intervalo  $[0, 1]$ .

Con base en la notación de proposiciones difusas atómicas y conectores lingüístico tales como "y", "o", "no", y "si – entonces", se pueden formar proposiciones difusas complejas.

- X es A y X es B
- X es A o X es B
- X no es A
- (X es A y X no es B) o X es C
- Si X es A entonces X es B, etc

Y el significado de esas proposiciones difusas complejas es proporcionado por la interpretación de los conectores *y*, *o* y *no*, como conjunción, disyunción y negación, respectivamente.

**Conjunción.** Sea *p* y *q* dos proposiciones difusas atómicas; *p* : "X es A y *q* : "X es B" donde  $\tilde{A}$  y  $\tilde{B}$  son conjuntos difusos definidos sobre el mismo universo en discurso *U*. Entonces la conjunción  $\cap$  es definida como sigue.

Simbolico  $\rightarrow$  significado

$$X \text{ es } A \rightarrow \mu_{\tilde{A}}$$

$$X \text{ es } B \rightarrow \mu_{\tilde{B}}$$

$$X \text{ es } A \cap B \rightarrow \mu_{\tilde{A} \cap \tilde{B}}$$

El significado de la conclusión  $X \text{ es } A \cap B$  es proporcionado por  $\mu_{\tilde{A} \cap \tilde{B}}$  la cual es definida por una T-norma . Por ejemplo, si *p* = "La presión no es muy alta" y *q* = "La presión no es muy baja ", entonces la conclusión producida por la regla puede ser, "La presión no es muy alta y no es tan bajaz su significado puede ser proporcionado por

$$\mu_{(\text{no es muy alta}) \cap (\text{no baja})}$$

**Disyunción.-** La disyunción  $\cup$  está dada por.

Simbolico  $\rightarrow$  significado

$$X \text{ es } A \rightarrow \mu_{\tilde{A}}$$

$$X \text{ es } B \rightarrow \mu_{\tilde{B}}$$

$$X \text{ es } A \cup B \rightarrow \mu_{\bar{A} \cup \bar{B}}$$

El significado de la conclusión  $X$  es  $A \cup B$  es proporcionado por  $\mu_{\bar{A} \cup \bar{B}}$  la cual es definida por una norma  $S$

**Negación.** La negación  $X$  no es  $A$  de una **proposición difusa**, que es dada por "X es A", es proporcionado por

Simbolo  $\rightarrow$  Significado

$$X \text{ es } A \rightarrow \mu_{\bar{A}}$$

$$X \text{ es } A' \rightarrow \mu_{\bar{A}'}$$

Donde  $A'$  es el complemento de  $A$ , la negación puede ser definida por cualquier norma .

### 2.2.3. Reglas de Inferencia o Razonamiento

Las reglas son un modo de representar estrategias o técnicas apropiadas cuando el conocimiento proviene de la experiencia o de la intuición (careciendo de demostración matemática o física) que usan conectores lingüísticos **SI ENTONCES (IF THEN)**.

SI < antecedente ó condición < ENTONCES < consecuente ó conclusión >

$$\text{SI } x \text{ es } \tilde{A} \text{ ENTONCES } y \text{ es } \tilde{B}$$

Es decir, lo que se busca es una relación difusa  $\tilde{R}$  cuya función de pertenencia exprese el grado de verdad de "SI  $x$  es  $A$  ENTONCES  $y$  es  $B$ ".

A esta acción se la denomina "*implicar*", y hay muchas formas de hacerlo que más adelante se explicarán. Implicar es el paso previo a "*inferir*", que consiste en extraer una conclusión a partir de la relación generada por la implicación y de un conjunto difuso de entrada  $X$  es  $A = \tilde{A}$ . Dicha conclusión es un nuevo conjunto difuso  $y$  es  $B = \tilde{B}$ .

En otras palabras, el enunciado (2.6) se interpreta como una relación difusa definida sobre  $X * Y$  (donde  $*$  es un producto cartesiano) en tres etapas:

1.-Antecedente, parte que viene después del *SI* y antes del *ENTONCES*  $x$  es  $A$  se interpreta con el conjunto difuso

$$\tilde{A} = \sum_{x \in X} \{\mu_{\tilde{A}}(x)/x \quad (2.7)$$

donde el operador /"denota una dupla, y los elemento a sumarse son los grados de pertenencia; el significado de este operador es el mismo para las dos etapas siguientes.

2.- Consecuente, parte que viene después del *ENTONCES* "y es B" se interpreta con el conjunto difuso

$$\tilde{B} = \sum_{y \in Y} \{\mu_{\tilde{B}}(y)/y \quad (2.8)$$

3.- Condicionante *ENTONCES* se interpreta a partir de la relacin difusa.

$$\tilde{R} = \sum_{(x,y) \in X} \{\mu_{\tilde{R}}(x,y)/x,y \quad (2.9)$$

con  $\mu_{\tilde{R}}(x,y) = \mu_{\tilde{A}}(x) * \mu_{\tilde{B}}(y)$  donde \* es un operador de implicación difusa. A este proceso se le denomina *inferencia*

## 2.3. Control Difuso

Un sistema de control básico se muestra en la figura 2.3. El proceso (o planta) es el objeto a ser controlado, la señal de control es  $u(t)$ , la salida es  $y(t)$ , y la entrada de referencia es  $r(t)$ . La teoría de control convencional proporciona numerosos métodos para la construcción de controladores para sistemas dinámicos. En general,

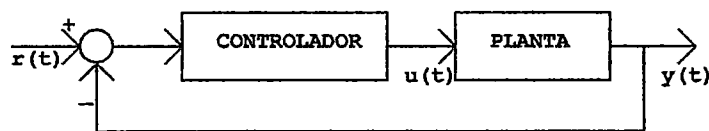


Figura 2.3: Sistema de control

los controladores difusos son sistemas expertos que emplean el conocimiento como fundamento, expresado en términos de reglas difusas y de un proceso de inferencia para resolver un problema de control dado.

Los controladores difusos a diferencia de los controladores clásicos, son capaces de utilizar el conocimiento obtenido de operadores humanos. El conocimiento de un operador humano puede ser utilizado como una alternativa para un modelo preciso



de un proceso controlado. Mientras este conocimiento es complicado de expresar en términos precisos, una descripción lingüística imprecisa de la forma en que se realiza el control puede usualmente ser articulada por el operador con relativa facilidad. Esta descripción lingüística consiste de un conjunto de reglas que hacen uso de proposiciones difusas

En general, un controlador difuso está formado por cuatro módulos: una base de reglas difusas, un proceso de inferencia de reglas, un módulo denominado fuzzyfización y uno denominado defuzzyfización. Las interconexiones entre estos módulos y el proceso principal son mostradas en la figura 2.4.

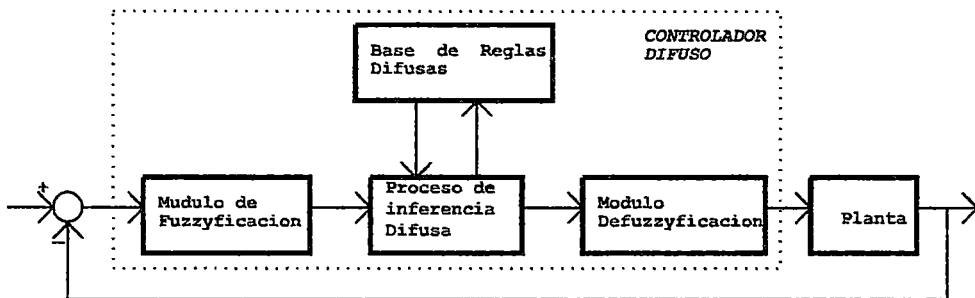


Figura 2.4: Esquema General de un controlador difuso

Un controlador difuso opera mediante el ciclo repetitivo considerando los siguientes cuatro pasos:

1. Las mediciones de todas las variables que representan condiciones relevantes del proceso controlado son realizadas.
2. Las mediciones son convertidas en conjuntos difusos apropiados para expresar mediciones insuficientes.
3. Este paso es llamado fuzzyfización. Las mediciones fuzzyficadas son usadas por el proceso de inferencia para evaluar las reglas de control almacenadas en las base de reglas difusas. El resultado de esta evaluación es un conjunto difuso (o varios conjuntos difusos) definido sobre el universo de posibles acciones.
4. El conjunto difuso encontrado en el paso 3 es convertido en el último paso del ciclo, es decir, se convierte el conjunto difuso en un solo valor (crisp) o

en un vector de valores. Esta conversión es llamada defuzzyficación. Los valores defuzzyficados representan acciones de control tomadas por el controlador difuso.

## 2.4. Fuzzyficación

La fuzzyficación (codificación) traduce a valores difusos los valores de las variables de entrada (del controlador). La función de la fuzzyficación más simple es la tipo "singleton", es decir, el valor crisp será el único valor del conjunto difuso (ver figura 2.1) [4]:

**Singleton** : Asociación entre un valor crisp  $x$  y un conjunto difuso  $\tilde{A}$ , es decir el conjunto difuso solo tendrá un solo grado de pertenencia.

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 & \text{si } x = x' \\ 0 & \text{si } x \neq x' \end{cases}$$

donde:  $x'$  es el valor actual de entrada.

**No Singleton** : el grado de pertenencia va disminuyendo según se separa del punto  $x$ , para la función de pertenencia del tipo gaussiana:

$$\mu_{\tilde{A}}(x') = \left( e^{-\frac{(x'-x)^2}{(\rho)^2}} \right)$$

donde:  $x'$  es el valor actual de entrada.

**Ventajas de la fuzzyficación:**

- Permite minimizar posibles cambios pequeños en las variables de entrada. Por ejemplo, si medimos 25°C de temperatura, ésta puede variar mientras se efecta la inferencia, por lo que puede ser ms exacto considerar que la temperatura es "aproximadamente 25°C".
- Permite minimizar los posibles errores al tomar los datos.
- Permite suavizar el comportamiento del sistema ampliando el rango de influencia de la variable.

## 2.5. Defuzzyficación

La defuzzyficación convierte los valores difusos de las variables de salida en valores concretos dentro del universo de discurso correspondiente. Es decir genera una

acción no difusa a partir de la acción difusa resultante del sistema de inferencia.

Existen diversos métodos de defuzzyficación, a continuación se presentan algunos:  
Asociación entre un conjunto difuso  $\tilde{A}$  y un valor crisp  $y = (y_1, y_2, \dots, y_n, \dots, y_N)^T$ .  
donde  $y_n$  es el valor actual de la entrada.

$M$  es el número de reglas.

$\mu_{A_n^m}(y_n)$  es el grado de pertenencia de valor de entrada actual para la  $m$ ésima regla.

El operador " ." indica el producto escalar.

- Máximo

$$y = \sup_{y \in Y} (\mu_{\tilde{A}}(y))$$

- Centro de media

$$y = \frac{\sum_{m=1}^M y_n \mu_{\tilde{A}_n^m}(y_n)}{\sum_{m=1}^M \mu_{\tilde{A}_n^m}(y_n)}$$

- Centro de media modificado

$$y = \frac{\sum_{m=1}^M y_n \frac{\mu_{\tilde{A}_n^m}(y_n)}{\delta_n}}{\sum_{m=1}^M \frac{\mu_{\tilde{A}_n^m}(y_n)}{\delta_n}}$$

donde  $\delta$  es la media de los datos de entrada, la cual esta dada por

$$\delta_n = \frac{y_1 + y_2 + \dots + y_n + \dots + y_N}{N} \quad (2.10)$$

### 2.5.1. Tipos de Controladores Difusos

Los controladores difusos se pueden clasificar en dos tipos:

- 1.- Controlador tipo Mamdani.
- 2.- Controlador tipo Takagi-Sugeno.

La principal diferencia entre estos controladores radica en la consecuencia de las reglas. Para el controlador tipo Mamdani esta consecuencia es un conjunto difuso y para el tipo Takagi-Sugeno es una función lineal de las entradas.

En este trabajo se utilizará el controlador tipo Takagi-Sugeno con función lineal constante como consecuencia de sus reglas.

### Controlador Difuso del tipo Takagi – Sugeno

Siguiendo la estructura del controlador difuso que se muestra en la figura 2.4, se explica de manera breve el desarrollo del controlador del tipo Takagi-Sugeno.

- 1.- Fuzzyficación de los datos de entrada, es decir convertirlos en conjuntos difusos.
- 2.- Obtención una base reglas. El controlador de Takagi-Sugeno tiene la particularidad de que las reglas que definen su funcionamiento son de la forma siguiente.

$$R^1 : Si(x_1 es L_{x_1}^1), \dots, y(x_n es L_{x_n}^1), y, \dots, y(x_N es L_{x_N}^1) ENTONCES u_1 = f_1(x_1 + x_2, \dots, x_N)$$

$$R^m : Si(x_1 es L_{x_1}^m), \dots, y(x_n es L_{x_n}^m), y, \dots, y(x_N es L_{x_N}^m) ENTONCES u_m = f_m(x_1 + x_2, \dots, x_N)$$

$$R^M : Si(x_1 es L_{x_1}^M), \dots, y(x_n es L_{x_n}^M), y, \dots, y(x_N es L_{x_N}^M) ENTONCES u_M = f_M(x_1 + x_2, \dots, x_N)$$

Donde.

$R^{(M)}$  es la m-ésima regla,

M es el número de reglas,

X es el conjunto de las variables lingüísticas, la cual es descrito por

$$X = (x_1, x_2, \dots, x_n, \dots, x_N)^T$$

N es el número de elementos del conjunto X,

X es el valor lingüístico que toman las variables lingüísticas.

- 3.- Obtención de la inferencia de la base de reglas, es decir, extracción de la conclusión de cada una de las reglas.

Con estas reglas se realiza primero para algunos valores de entradas dadas del proceso o del sistema  $x_1^*, \dots, x_n^*, \dots, x_N^*$ , sus grados de pertenencia  $\mu_{L_{x_n}^m}^m(x_n^*)$ , son calculados y llamados antecedentes de reglas o pesos de reglas, se obtiene el mínimo (Implicación del mínimo) de los N grados de pertenencia para la regla m y se denota como  $\mu_{u_m}$ . Posteriormente, los valores  $\mu_m$  son computados como  $\mu_m^* = f_m(x_1^*, x_2^*, \dots, x_N^*)$ . Para cada una de las reglas  $R^{(m)}$  se obtiene la dupla  $(\mu_{u_m}, u_m^*)$ .

- 4.- Finalmente, la salida combinada de todas las reglas es la suma de los pesos normalizados de todos los pares, es decir se utiliza el proceso de defuzzyficación centro

de media para obtener la salida del controlador difuso Takagi-Sugeno

$$u = \frac{\sum_{m=1}^M \mu_{u_m} x_n^* \cdot u_n^*}{\sum_{m=1}^M \mu_{u_m} (x_n)^*} \quad (2.11)$$

## Capítulo 3

# Modelado de la Planta

Para desarrollar adecuadamente la técnica de lógica difusa en el sistema, como ya se ha mencionado es un sistema que por su naturaleza es no lineal, y algo complejo al momento de obtener su modelo matemático. A continuación obtendremos un modelo general que, a partir del esquema mostrado en la figura 3.1, usando la ecuación dinámica de Lagrange, que nos va a proporcionar la exactitud del modelo.

### 3.1. Descripción del Sistema

El sistema del Brazo Robótico Translacional Multivariable de 2DOF que en adelante lo llamaremos (BRTM de 2DOF) es mostrado en la figura 3.1.  $M_1$  es un servomotor de DC con un mecanismo de reducción y un encoder óptico articulado a una polea de radio  $R_p$ . Esta polea usa un cable para transmitir la fuerza  $F_1$  que maneja el movimiento corridizo de un carro de masa  $M_c$  montado a un par de rieles a lo largo del eje de x.  $M_2$  también es un servomotor DC con un encoder óptico empleado para manejar el movimiento rotatorio del brazo (libre de girar en ambas direcciones) del BRTM alrededor de un punto del pivote; se asume que tal punto es el CG (centro de gravedad) del carro.

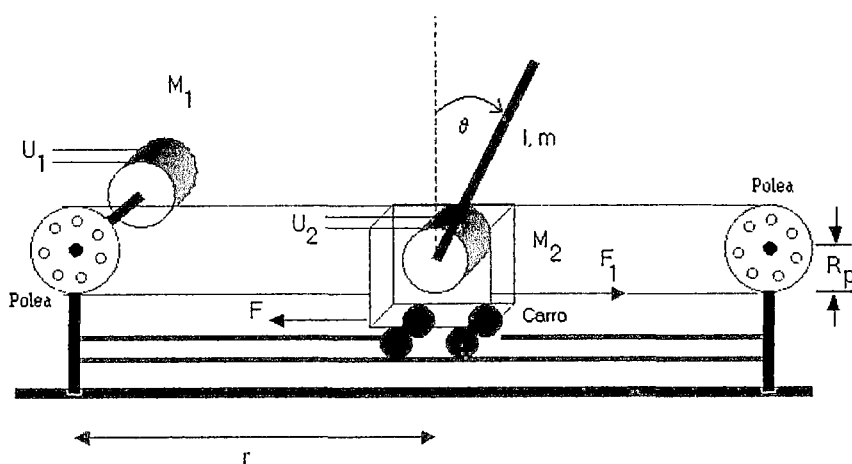


Figura 3.1: Manipulador Robótico Translacional Multivariable (MRTM).

En la figura 3.1,  $\theta$  es la posición angular del brazo de longitud  $l$  y masa  $m$ ,  $r$  es la posición del carro,  $F$  es la fuerza friccional opuesta al movimiento del carro.

El proceso de lazo cerrado para ser controlado representa un proceso con dos entradas; los voltajes de control  $K_{A1}u_1$  y  $K_{A2}u_2$  aplicados a los terminales de armadura de  $M_1$  y  $M_2$  respectivamente, y dos salidas:  $r$  y  $\theta$ . Los parámetros  $K_{A1}$  y  $K_{A2}$  son las ganancias de amplificación.

## 3.2. Modelo de Lagrange del Proceso BRTM

Antes de generalizar la ecuación dinámica de Lagrange, se debe de construir las condiciones de energía necesarias para cada componente del sistema. El propósito de este modelo matemático es desarrollar y probar una ley de control que pueda hacer que la trayectoria del carro y la trayectoria del brazo sigan trayectorias de referencia arbitrarias. Nosotros asumimos que las dos componentes de la salida  $y(r, \theta)$  son medidas y que la fuerza  $u(u_1, u_2)$  es la entrada de control.

Nuestro primer objetivo es encontrar la relación dinámica entre  $u(u_1, u_2)$  y  $y(r, \theta)$ . Para esto podemos usar las ecuaciones de Lagrange. Para expresar la energía en este sistema, introducimos las variables de la ecuación de Lagrange. El principio de Lagrange nos permite escribir en forma directa las ecuaciones de energía cinética  $T$ , energía potencial  $U$  y energía disipada  $D$ . En otras palabras, una vez que modelamos  $T$ ,  $U$  y  $D$ , tendremos las ecuaciones dinámicas del sistema.

### 3.2.1. Ecuaciones de Energía para el Carro

El carro es restringido para un movimiento en la dirección horizontal, cuyas ecuaciones dinámicas comprenden el desarrollo de las energías cinética ( $T_1$ ), potencial ( $U_1$ ) y de disipación ( $D_1$ ) dadas por.

$$T_1 = \frac{1}{2}M_c\dot{r}^2 \quad (3.1)$$

$$U_1 = 0 \quad (3.2)$$

$$D_1 = \frac{1}{2}F_c\dot{r}^2 \quad (3.3)$$

### 3.2.2. Ecuaciones de Energía para el Brazo

Sea la energía cinética ( $T_2$ ) del brazo dada por.

$$T_2 = \frac{1}{2}J\dot{\theta}^2 + \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) \quad (3.4)$$

donde  $J$  es el momento de inercia del brazo, de masa de  $m$  de la figura 3.1.

El desarrollo de la ecuación (3.4) se explica con ayuda de la figura 3.2 que muestra las coordenadas  $(x,y)$  en función de los parámetros del sistema del brazo robótico.

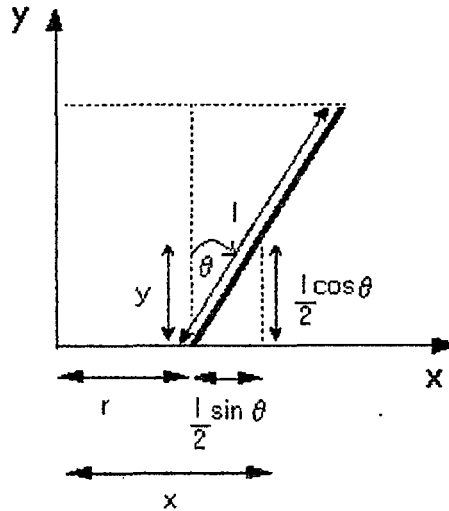


Figura 3.2: Subsistema del brazo robótico.

$$x = r + \frac{l}{2}\sin\theta \quad (3.5)$$

$$y = \frac{l}{2}\cos\theta \quad (3.6)$$

derivando las ecuaciones (3.5) y (3.6).

$$\dot{x} = \dot{r} + \frac{l}{2}\cos(\theta)\dot{\theta} \quad (3.7)$$

$$\dot{y} = -\frac{l}{2}\sin(\theta)\dot{\theta} \quad (3.8)$$

elevando al cuadrado.

$$\dot{x}^2 = \dot{r}^2 + r l \cos(\theta)\dot{\theta} + \frac{l^2}{4}\cos^2(\theta)\dot{\theta}^2 \quad (3.9)$$

$$\dot{y}^2 = \frac{l^2}{4}\sin^2(\theta)\dot{\theta}^2 \quad (3.10)$$



sumando las ecuaciones (3.9) y (3.10).

$$\dot{x}^2 + \dot{y}^2 = \dot{r}^2 + r\dot{l}\cos(\theta)\dot{\theta} + \frac{l^2}{4}\dot{\theta}^2 \quad (3.11)$$

reemplazando (3.11) en (3.4) tenemos la energía cinética.

$$T_2 = \frac{1}{2}J\dot{\theta}^2 + \frac{1}{2}m(\dot{r}^2 + r\dot{l}\cos(\theta)\dot{\theta} + \frac{l^2}{4}\dot{\theta}^2) \quad (3.12)$$

La energía potencial y la energía de disipación del brazo, es dada por las ecuaciones.

$$U_2 = \frac{mgl}{2}\cos\theta \quad (3.13)$$

$$D_2 = \frac{1}{2}C\dot{\theta}^2 \quad (3.14)$$

### 3.3. Ecuaciones de Lagrange del Proceso BRTM

La ecuación de Lagrange generalizada es.

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + \frac{\partial D}{\partial \dot{q}} = F_q \quad (3.15)$$

donde,  $L$  es el conjunto de energías del sistema ( $L = T - U$ ),  $D$  es la disipación de energía externa del sistema,  $q$  es la coordenada de variable independiente del sistema,  $\dot{q}$  es la derivada de variable independiente del sistema y  $F_q$  es la fuerza externa que actúa en la dirección de  $q$ .

Para el sistema BRTM de 2DOF, los términos de las energías individuales componen en su conjunto la totalidad energética del sistema.

Energía cinética del sistema

$$T = T_1 + T_2 = \frac{1}{2}M_c\dot{r}^2 + \frac{1}{2}J\dot{\theta}^2 + \frac{1}{2}m(\dot{r}^2 + r\dot{l}\cos(\theta)\dot{\theta} + \frac{l^2}{4}\dot{\theta}^2)$$

Energía potencial del sistema

$$U = U_1 + U_2 = \frac{mgl}{2}\cos\theta$$

Energía disipada del sistema

$$D = D_1 + D_2 = \frac{1}{2}F_c\dot{r}^2 + \frac{1}{2}C\dot{\theta}^2$$

Podemos generalizar la ecuación de Lagrange para el vector  $q$  en función de coordenadas generales  $(r, \theta)$  del proceso MRTM de 2DOF es formulado de acuerdo con (3.15).

$$\frac{\partial}{\partial t} \left( \frac{\partial T}{\partial \dot{r}} \right) - \frac{\partial T}{\partial r} + \frac{\partial U}{\partial r} + \frac{\partial D}{\partial \dot{r}} = F_1 \quad (3.16)$$

$$\frac{\partial}{\partial t} \left( \frac{\partial T}{\partial \dot{\theta}_1} \right) - \frac{\partial T}{\partial \theta_1} + \frac{\partial U}{\partial \theta_1} + \frac{\partial D}{\partial \dot{\theta}_1} = T_L \quad (3.17)$$

Para el planteo de las ecuaciones de Lagrange, se observa que el término de energía potencial  $U$  no es una función del tiempo derivativo de  $r$ . Las ecuaciones dinámicas del Lagrange serán desarrolladas para las variables en donde interviene el sistema.

Desarrollamos las derivadas parciales que indica la ecuación (3.16) para determinar la fuerza  $F_1$ .

$$\frac{\partial T}{\partial \dot{r}} = (M_c + m)\dot{r} + \frac{ml}{2}\cos(\theta)\dot{\theta} \quad (3.18)$$

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{r}} \right) = (M_c + m)\ddot{r} - \frac{ml}{2}\sin(\theta)\dot{\theta}^2 + \frac{ml}{2}\cos(\theta)\ddot{\theta} \quad (3.19)$$

$$\frac{\partial T}{\partial r} = 0; \quad \frac{\partial U}{\partial r} = 0; \quad \frac{\partial D}{\partial \dot{r}} = F_c\dot{r} \quad (3.20)$$

Obtenemos la primera ecuación parcial del sistema para  $F_1$ .

$$F_1 = (M_c + m)\ddot{r} - \frac{ml}{2}\sin(\theta)\dot{\theta}^2 + \frac{ml}{2}\cos(\theta)\ddot{\theta} + F_c\dot{r} \quad (3.21)$$

Desarrollamos las derivadas parciales que indica la ecuación (3.17) para determinar el torque  $T_L$ .

$$\frac{\partial T}{\partial \dot{\theta}} = J\dot{\theta}_1 + \frac{ml}{2}\cos(\theta)\dot{r} + \frac{ml^2}{4}\dot{\theta} \quad (3.22)$$

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\theta}} \right) = J\ddot{\theta} + \frac{ml}{2}\cos(\theta)\ddot{r} - \frac{ml}{2}\sin(\theta)\dot{\theta}\dot{r} + \frac{ml^2}{4}\ddot{\theta} \quad (3.23)$$

$$\frac{\partial T}{\partial \theta} = -\frac{ml}{2}\sin(\theta)\dot{r}; \quad \frac{\partial U}{\partial \theta} = -\frac{ml}{2}g\sin(\theta); \quad \frac{\partial D}{\partial \dot{\theta}_1} = C\dot{\theta} \quad (3.24)$$

Obtenemos la segunda ecuación parcial del sistema para  $T_L$ .

$$T_L = \frac{ml}{2} \cos(\theta) \ddot{r} + \left(\frac{ml^2}{4} + J\right) \ddot{\theta} + C\dot{\theta} - \frac{mgl}{2} \sin(\theta) \quad (3.25)$$

El modelo Lagrangiano puede ser derivado usando las ecuaciones (3.21) y (3.25). Sin embargo si nosotros estamos interesados en tener como fuerzas de control a los voltajes de armadura  $u_1$  y  $u_2$  en lugar de  $F_1$  y  $T_L$ , respectivamente, para esto es necesario modelar los subsistemas eléctricos del MRTM de 2DOF.

### 3.3.1. Modelo del Servomotor DC $M_1$ - Polea

La figura 3.3 muestra el servomotor de DC  $M_1$  que es articulado a una polea. Despreciando la inductancia de la armadura  $L_a$ , el voltaje de entrada  $K_A u_1$  aplicado a la armadura se da por.

$$R_a I_{a1} + V_{b1} = K_A u_1 \quad (3.26)$$

El voltaje de la fuerza electromotriz es proporcional a la velocidad de motor.

$$V_{b1} = K_b \dot{\theta}_{m1} \quad (3.27)$$

La ecuación del torque  $T_{m1}$  del servomotor es dado por.

$$T_{m1} = J_m \ddot{\theta}_{m1} + B_m \dot{\theta}_{m1} + T_{g1} \quad (3.28)$$

El torque  $T_{g2}$  requiere la transmisión de la polea, y el torque en la polea es directamente proporcional a la fuerza  $F_1$ , luego formulamos.

$$T_{g2} = nT_{g1} = (J_g + J_p) \ddot{\theta}_L + (B_g + B_p) \dot{\theta}_L + F_1 R_p \quad (3.29)$$

donde  $J_m$ ,  $J_g$  y  $J_p$  son los momentos de inercia de la armadura, mecanismo de reducción de engranaje y polea, respectivamente, mientras  $B_m$ ,  $B_g$  y  $B_p$  son las constantes de fricción de la armadura, mecanismo de reducción de engranaje y polea respectivamente.

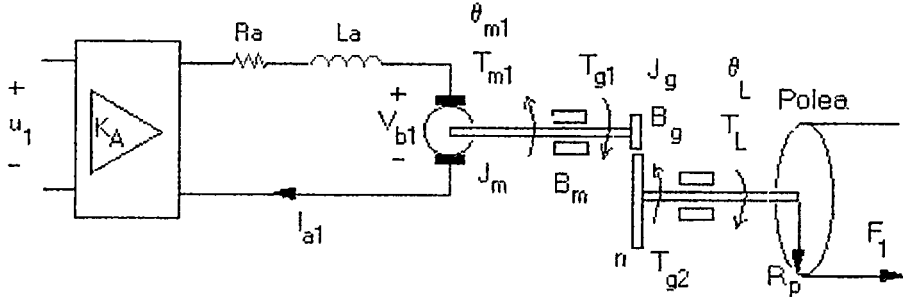


Figura 3.3: Subsistema motor - polea.

También se ha usado la relación para la conservación de energía de engranajes  $T_{g1}\dot{\theta}_{m1} = T_{g2}\dot{\theta}_L$  con  $(n\theta_L = \theta_{m1})$ , de este modo podemos expresar (3.28) como.

$$T_{m1} = J_m n \ddot{\theta}_L + B_m n \dot{\theta}_L + T_{g1} \quad (3.30)$$

despejando  $T_{g1}$  de (3.29) y reemplazamos en (3.30), luego agrupando términos tenemos.

$$nT_{m1} = J_{eq1} \ddot{\theta}_L + B_{eq1} \dot{\theta}_L + F_1 R_p \quad (3.31)$$

donde.

$$J_{eq1} = n^2 J_m + J_g + J_p$$

$$B_{eq1} = n^2 B_m + B_g + B_p$$

el torque del servomotor  $T_{m1}$  es proporcional a la corriente  $I_{a1}$ .

$$T_{m1} = K_m I_{a1} \quad (3.32)$$

reemplazando (3.31) en (3.32), luego despejamos  $I_{a1}$ .

$$I_{a1} = \frac{J_{eq1}}{nK_m} \ddot{\theta}_L + \frac{B_{eq1}}{nK_m} \dot{\theta}_L + \frac{R_p F_1}{nK_m} \quad (3.33)$$

usando las ecuaciones (3.27), (3.33) en (3.26), luego despejamos  $F_1$ .

$$F_1 = \frac{nK_m K_A}{R_a R_p} u_1 - \frac{J_{eq1}}{R_p} \ddot{\theta}_L - \left( \frac{B_{eq1}}{R_p} + \frac{n^2 K_m K_b}{R_a R_p} \right) \dot{\theta}_L \quad (3.34)$$

El movimiento rotacional de la polea puede ser transformado como un movimiento translacional del carro usando la siguiente relación (ver figura 3.3).

$$\theta_L = \frac{r}{R_p} \quad (3.35)$$

derivando (3.35) y sustituyendo en (3.34), obtenemos la ecuación general del subsistema del motor.

$$F_1 = \frac{nK_m K_A}{R_a R_p} u_1 - \frac{J_{eq1}}{R_p^2} \ddot{r} - \left( \frac{B_{eq1}}{R_p^2} + \frac{n^2 K_m K_b}{R_a R_p^2} \right) \dot{r} \quad (3.36)$$

igualando (3.36) con (3.21), se obtiene la primera ecuación dinámica de Lagrange del sistema.

$$u_1 = \frac{R_a R_p}{n K_m K_A} \left\{ \left( M_c + m + \frac{J_{eq1}}{R_p^2} \right) \ddot{r} + \left( F_c + \frac{B_{eq1}}{R_p^2} + \frac{n^2 K_m K_b}{R_a R_p^2} \right) \dot{r} + \frac{ml}{2} \cos(\theta) \ddot{\theta} + \dots - \frac{ml}{2} \sin(\theta) \dot{\theta}^2 \right\} \quad (3.37)$$

### 3.3.2. Modelo del Servomotor DC $M_2$ - Brazo

La figura 3.4 muestra el servomotor de DC  $M_2$  que articula a un punto del pivote localizado en el CG del carro. Teniendo en cuenta que  $M_1$  y  $M_2$  tiene los mismos parámetros y despreciando la inductancia de la armadura  $L_a$ .

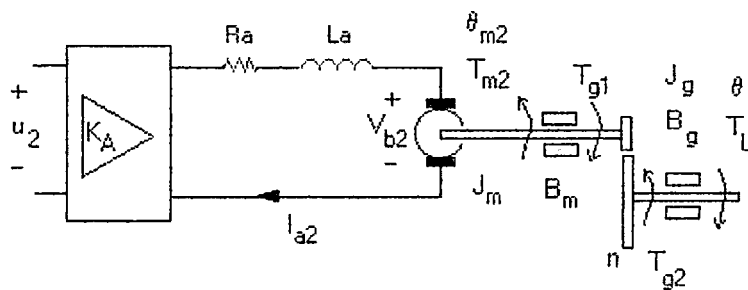


Figura 3.4: Subsistema del torque del brazo.

El voltaje de entrada  $K_A u_2$  aplicado a la armadura se da por.

$$R_a I_{a2} + V_{b2} = K_A u_2 \quad (3.38)$$

donde.

$$V_{b2} = K_b \dot{\theta}_{m2} \quad (3.39)$$

La relación de desplazamiento angular del eje del motor y eje de carga es

$$\theta_{m2} = n\theta \quad (3.40)$$

La ecuación de torque  $T_{m2}$  del servomotor (ver figura 3.4) será

$$T_{m2} = J_m n \ddot{\theta} + B_m n \dot{\theta} + T_{g1} \quad (3.41)$$

El torque  $T_{g2}$  requiere manejar el brazo que se expresa por

$$T_{g2} = nT_{g1} = J_g \ddot{\theta} + B_g \dot{\theta} + T_L \quad (3.42)$$

reemplazando (3.42) en (3.41), siendo  $T_L$  el torque de carga se obtiene

$$nT_{m2} = J_{eq2} \ddot{\theta} + B_{eq2} \dot{\theta} + T_L \quad (3.43)$$

donde.

$$J_{eq2} = n^2 J_m + J_g$$

$$B_{eq2} = n^2 B_m + B_g$$

El torque  $T_{m2}$  del servomotor es proporcional a la corriente  $I_{a2}$ .

$$T_{m2} = K_m I_{a2} \quad (3.44)$$

sustituyendo (3.40), (3.39) en (3.38) y despejando  $I_{a2}$ .

$$I_{a2} = \frac{K_A}{R_a} u_2 - \frac{nK_b}{R_a} \dot{\theta} \quad (3.45)$$

sustituyendo (3.45), (3.44) en (3.43) y despejando  $T_L$ .

$$T_L = -J_{eq2} \ddot{\theta} - (B_{eq2} + \frac{n^2 K_b K_m}{R_a}) \dot{\theta} + \frac{nK_m K_A}{R_a} u_2 \quad (3.46)$$

igualando (3.46) con (3.25), se obtiene la segunda ecuación dinámica de Lagrange del sistema.

$$u_2 = \frac{R_a}{nK_m K_A} \left\{ \frac{ml}{2} \cos(\theta) \dot{r} + \left( \frac{ml^2}{4} + J + J_{eq2} \right) \ddot{\theta} + (C + B_{eq2} + \frac{n^2 K_m K_b}{R_a}) \dot{\theta} + \dots - \frac{m l g}{2} \sin(\theta) \right\} \quad (3.47)$$

### 3.3.3. Modelo Lagrangiano

La ecuación (3.21) con (3.37) y (3.25) con (3.47) conforman el proceso Lagrangiano del BRTM de 2DOF que toma la forma.

$$u_1 = m_{11} \ddot{r} + p_{11} \dot{r} + m_{12} \cos \theta \ddot{\theta} + p_{12} \sin \theta \dot{\theta}^2 \quad (3.48)$$

$$u_2 = m_{22} \ddot{\theta} + p_{22} \dot{\theta} + m_{21} \cos \theta \dot{r} + d_{21} \sin \theta \quad (3.49)$$

Las ecuaciones (3.48) y (3.49) pueden ser transformadas en su representación de la matriz como sigue.

$$M(x) \ddot{x} + P(x, \dot{x}) \dot{x} + d(x) = u \quad (3.50)$$

donde  $x$  y  $u$  son los estados.

$$x = \begin{bmatrix} r \\ \theta \end{bmatrix} \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

las matrices  $M$ ,  $P$  y  $Q$  del modelo del sistema son.

$$M = \begin{bmatrix} m_{11} & m_{12} \cos \theta \\ m_{21} \cos \theta & m_{22} \end{bmatrix} \quad P = \begin{bmatrix} p_{11} & p_{12} \dot{\theta} \sin \theta \\ 0 & p_{22} \end{bmatrix} \quad Q = \begin{bmatrix} 0 \\ d_{21} \sin \theta \end{bmatrix}$$

La tabla 3.1 muestra los coeficientes de las matrices  $M$ ,  $P$  y  $Q$  del sistema, mientras que la tabla 3.2 muestra los parámetros del servomotor DC. El método de identificación de los parámetros del motor y la obtención de las constantes de fricción del carro  $F_c$  y del brazo  $C$  se explican en el apéndice A. Se han considerado las siguientes equivalencias para la momento de inercia y la fricción viscosa  $J_{eq} = J_{eq1} = J_{eq2}$  y  $B_{eq} = B_{eq1} = B_{eq2}$ ; debido a que reflejan valores similares, tal como lo demuestran los resultados experimentales obtenidos en la identificación de los parámetros para los dos servomotores DC.

Tabla 3.1: Coeficientes de las matrices  $M$ ,  $P$  y  $Q$ .

---


$$\begin{aligned}
 m_{11} &= \left(\frac{R_a R_p}{n K_A K_m}\right) (M_c + m + \frac{J_{eq}}{R_p^2}) & p_{11} &= \left(\frac{R_a R_p}{n K_A K_m}\right) (F_c + \frac{B_{eq}}{R_p^2} + \frac{n^2 K_m K_b}{R_a R_p^2}) \\
 m_{12} &= \left(\frac{R_a R_p}{n K_A K_m}\right) \frac{ml}{2} & p_{12} &= -\left(\frac{R_a R_p}{n K_A K_m}\right) \frac{ml}{2} \\
 m_{21} &= \left(\frac{R_a}{n K_A K_m}\right) \frac{ml}{2} & p_{21} &= 0 \\
 m_{22} &= \left(\frac{R_a}{n K_A K_m}\right) \left(\frac{ml^2}{4} + J + J_{eq}\right) & p_{22} &= \left(\frac{R_a}{n K_A K_m}\right) (C + B_{eq} + \frac{n^2 K_m K_b}{R_a}) \\
 d_{12} &= 0 & d_{21} &= -\left(\frac{R_a}{n K_A K_m}\right) \frac{mlg}{2}
 \end{aligned}$$


---

Tabla 3.2: Parámetros estimados del sistema.

Símbolo	Descripción	Valor	Unidades
$M_c$	masa del carro + masa de la polea	1.6492	$Kg$
$m$	masa del brazo	0.0519	$Kg$
$l$	longitud del brazo	0.30	$m$
$J$	momento de inercia del brazo robótico	$2,38e^{-3}$	$Kgm^2$
$J_{eq}$	momento de inercia equivalente	$3.0551 \times 10^{-4}$	$Kgm^2$
$B_{eq}$	fricción viscosa equivalente	$5.3848 \times 10^{-4}$	$Nm/rad/s$
$C$	coeficiente de fricción viscosa brazo	$9,5e^{-4}$	$Kgm^2/seg$
$F_c$	coeficiente de fricción del carro	2.81	$Kg/seg$
$g$	aceleración de la gravedad	9.81	$mseg^{-2}$
$K_b$	constante de fuerza contraelectromotriz	0.0565	$V/rad/s$
$K_m$	constante de torque de motor	0.0421	$Nm/A$
$R_p$	radio de la polea	0.048	$m$
$R_a$	resistencia de armadura	5.3	$\Omega$
$n$	factor de reducción	12.5	
$K_A$	constante de ganancia de amplificación	5.5	

### 3.4. Respuesta de la Planta al Escalón Unitario

En la figura 3.5 muestra el diagrama de bloques del proceso BRTM de 2DOF en lazo abierto, se muestra la planta con sus entradas y salidas.

Las entradas del sistema nos representa los voltaje aplicado a cada motor DC y se simboliza como  $u_1$  y  $u_2$ ; las salidas en este caso son la posición lineal ( $r$ ) del carro y las posición angular del segundo motor  $\theta$



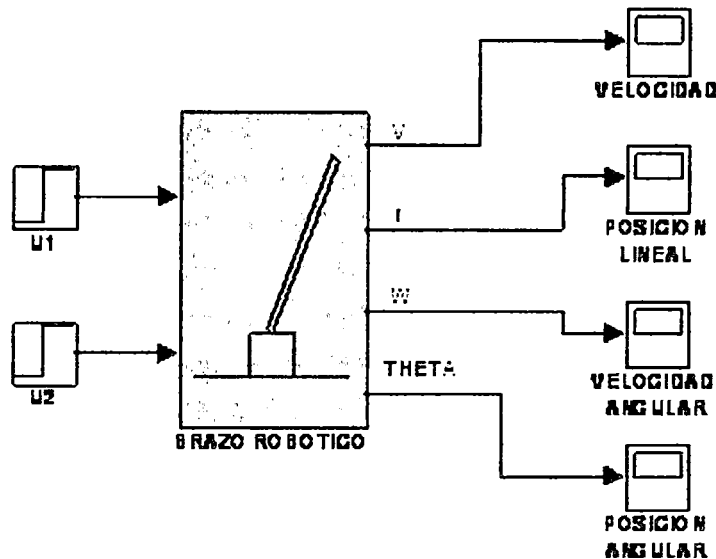


Figura 3.5: Planta no lineal a controlar con inductancia de valor cero.

Para poder observar una respuesta, se omitirá el efecto de la zona muerta, de modo que la señal constante de 1 voltio produzca un efecto en el sistema. De incluirse el efecto de la zona muerta, el sistema será insensible a esta entrada, como en efecto es.

Los valores de los parámetros considerados para esta simulación son aquellos obtenidos a través de la documentación, mediciones y experimentos mencionados hasta este punto. en la tabla 3.2 se tiene los valores estimados.

Las gráficas de las variables simuladas son :

Gráfica figura 3.6: Es la respuesta del sistema para la posición lineal  $[r]$  y posición angular  $\theta$ .

Gráfica figura 3.7: Es la velocidad lineal del carro dado en  $m/s$ .

Gráfica figura 3.8: Es la aceleración lineal del carro en  $m/s^2$ .

Gráfica figura 3.9: Velocidad angular del brazo ó varilla

Gráfica figura 3.10: Aceleración angular del brazo ó varilla

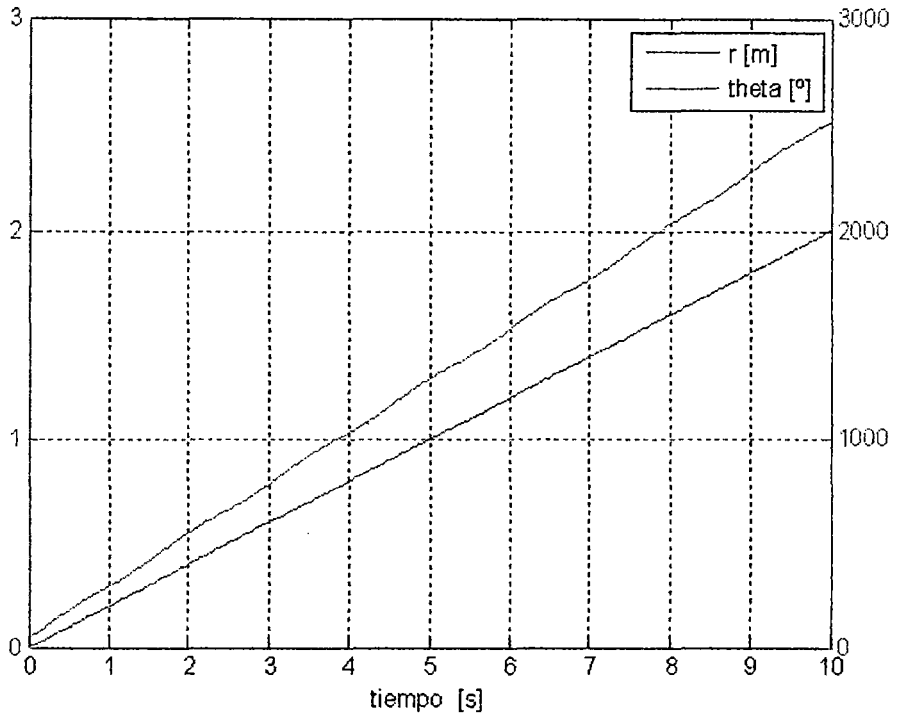


Figura 3.6: Respuesta de la posición lineal  $r$  y angular  $\theta$  debido a un escalón.

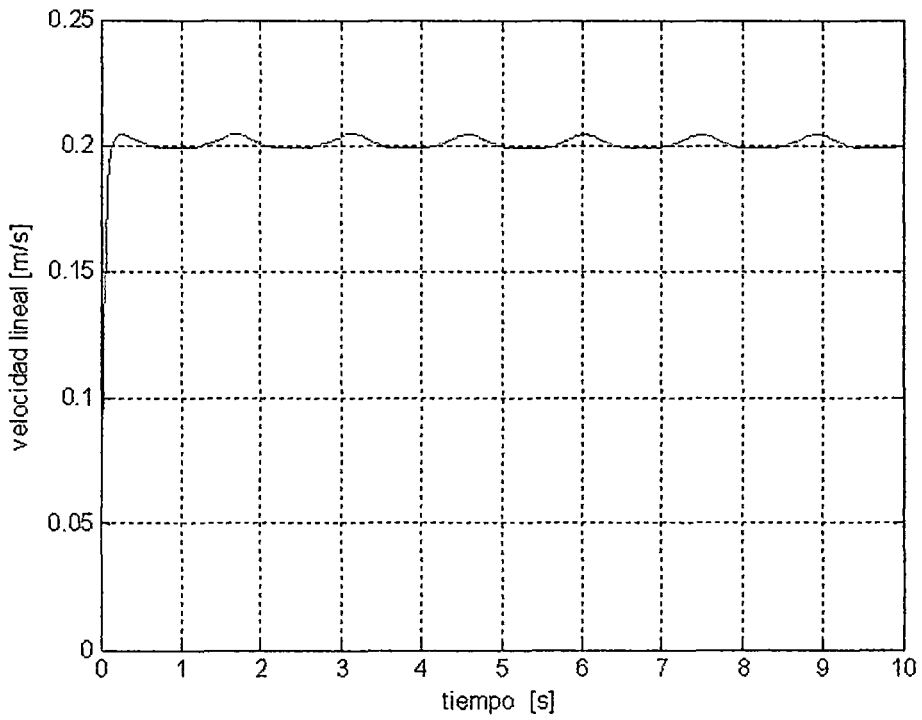


Figura 3.7: Respuesta de la velocidad lineal

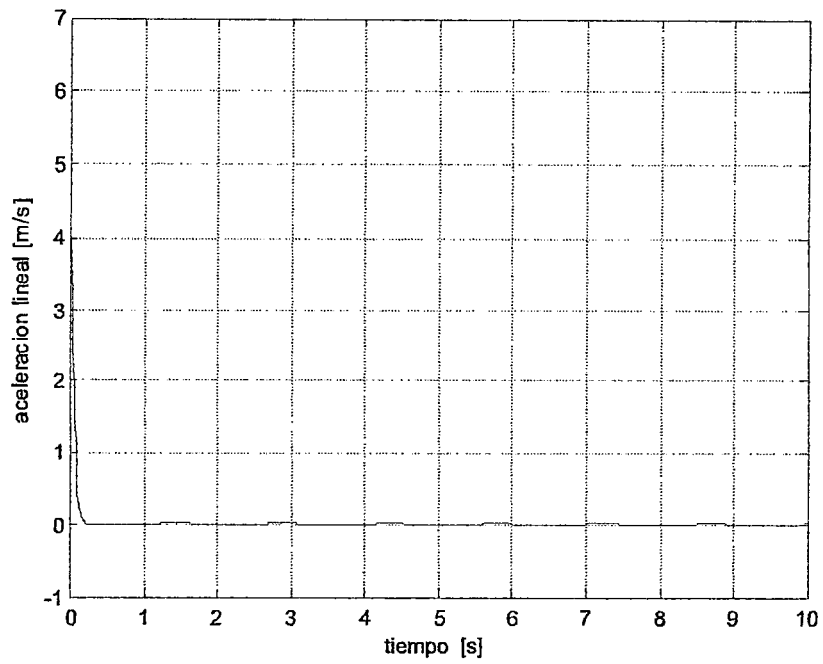


Figura 3.8: Respuesta de la aceleración lineal

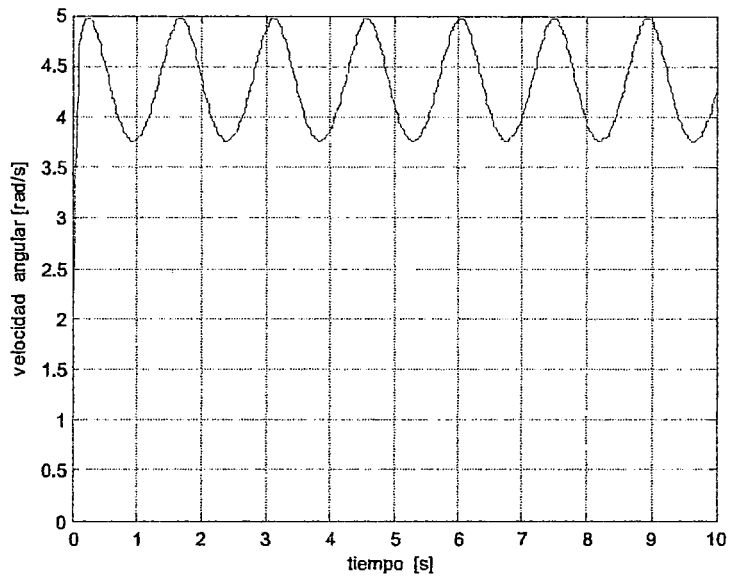


Figura 3.9: Respuesta de la velocidad angular

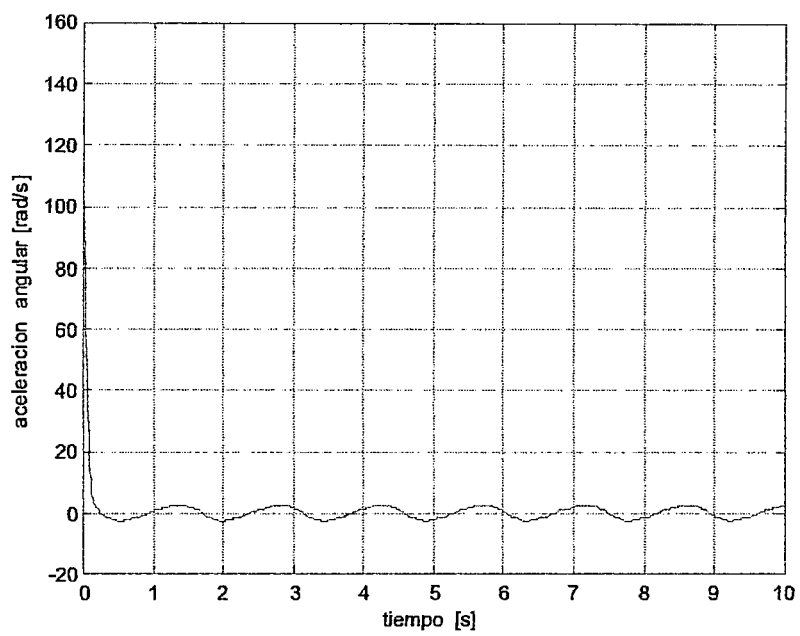


Figura 3.10: Respuesta de la Aceleración angular

## Capítulo 4

# Diseño del Control Fuzzy Tiempo Continuo

### 4.1. Introducción

En esta investigación se considerará admisible un error de posicionamiento angular de la varilla de  $\pm 0,3$  grados sexagesimales. El error admisible en la posición del carro se considerará como  $\pm 4mm.$ , para el sistema de brazo robótico. Asimismo, el sobreimpulso admisible será equivalente a  $\pm 5$  por ciento del recorrido, y el tiempo de establecimiento no deberá ser mayor a 6 segundos, que corresponden a 4 segundos para la aproximación a la posición final y 3 segundos para el posicionamiento fino y estabilización.

### 4.2. Diseño del controlador

El controlador basado en lógica difusa que se propone consiste de dos sistemas de inferencia en paralelo, cuyas salidas son independientes. Siendo el primer FIS un seguidor de la trayectoria generada especialmente para este FLC, controla la posición del carro ( posición lineal), mientras que el segundo FIS controla la posición del ángulo de la varilla . En la figura 4.1 se muestra el diagrama de bloques de este FLC. Se propone el procedimiento ha seguir:

**Primero:** Generar una trayectoria eficiente para el movimiento del carro

**segundo :** Se diseñarán los dos controladores considerando la robustez ante la variación de parámetros.

**tercero:** Se parte de la fuente para el diseño de los FLC: el conocimiento de ingeniería del diseñador

**Cuarto:** El controlador inicial es luego mejorado a través de la simulacion y experimentación.

**Consideraciones del procedimiento del diseño del controlador Fuzzy**

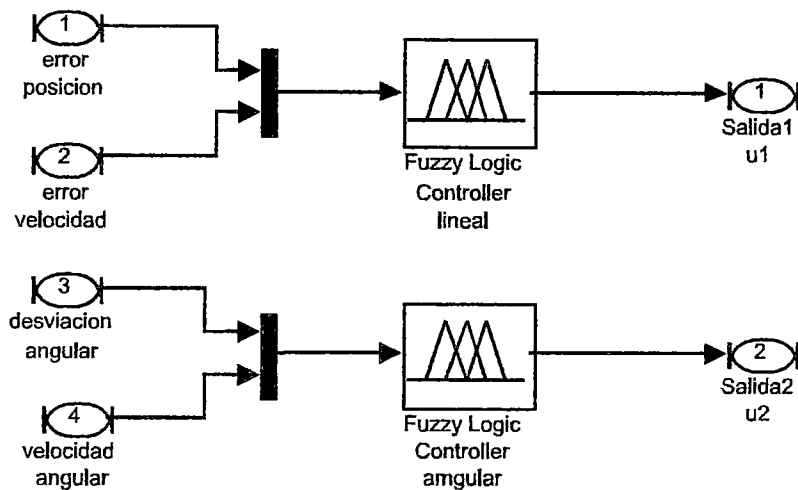


Figura 4.1: Diagrama del controlador

No existe un procedimiento estándar, pero es posible definir ciertos pasos de trabajo para llegar a un buen diseño del CBLD. La metodología que se propone sigue los lineamientos generales. Como se verá, en cada uno de ellos no existe una solución única, y su diseño se asienta en conocimiento heurístico sobre el proceso y pruebas de ensayo-error hasta conseguir una respuesta adecuada a nuestras especificaciones previas, en un proceso de refinamiento que la mayoría de las veces requiere una nueva definición en el criterio de selección usado en estos pasos. Esta "sintonización" del controlador puede hacerse también en forma automática, por ejemplo con algoritmos genéticos, o redes neuronales, de modo tal de encontrar un controlador óptimo. Incluso esta optimización puede hacerse en línea, una vez que el controlador está en funcionamiento normal en planta, aunque deben contemplarse especialmente los aspectos de estabilidad. Sin embargo, este tipo especial de CBLD denominados adaptivos, será objeto de otro trabajo.

La metodología propuesta se resume en los siguientes pasos:

1. Selección de las variables y del Universo de Discurso
2. Selección de la estrategia de emborronamiento
  - 2.1 Particionamiento de los espacios de entrada y salida
  - 2.2 Elección de las funciones de pertenencia
  - 2.3 Discretización/normalización de los Universos de Discurso
  - 2.4 Completitud

3. Construcción de la base de reglas
  - 3.1 Elección de las variables de entrada y las de control
  - 3.2 Elección de las reglas de control
4. Selección de lógica de toma de decisiones
  - 4.1 Definición del tipo de implicancia a utilizar
  - 4.2 Interpretación de los conectivos AND y ALSO
  - 4.3 Definición del operador de composición
  - 4.4 Mecanismo de inferencia
5. Selección de la estrategia de desemborronamiento

## Controlador Modelo difuso TAKAGI-SUGENO

Para nuestro caso se consideró el controlador modelo difuso TAKAGI- SUGENO, este modelo que se ha convertido en un método de razonamiento lógico muy conocido y utilizado, debido a sus numerosas aplicaciones en sistemas difusos, en controles difusos y en ingeniería difusa. La mayor ventaja del modelo Takagi-Sugeno es que es un modelo simple y puede ser utilizado en muchas aplicaciones prácticas para el modelado de sistemas complejos.

Este modelo se caracteriza por relaciones basadas en reglas difusas, donde las premisas de cada regla representan subespacios difusos y las consecuencias son una relación lineal de entrada - salida (*TakagiSugeno1995*). Las variables de entrada en las premisas de cada regla son relacionadas por operadores "Y", y las variables de salida en función de las variables de estado en general una función lineal ver figura 4.2

Las reglas de base de conocimiento de un sistema Sugeno es diferente a las de sistemas Mandami pues el consecuente de estas reglas ya no es una etiqueta lingüística si no es una función de entrada que tenga el sistema en un momento dado esto se ilustra a continuación.

*IF La entrada es baja THEN La salida  $Y = F(\text{entrada})$*

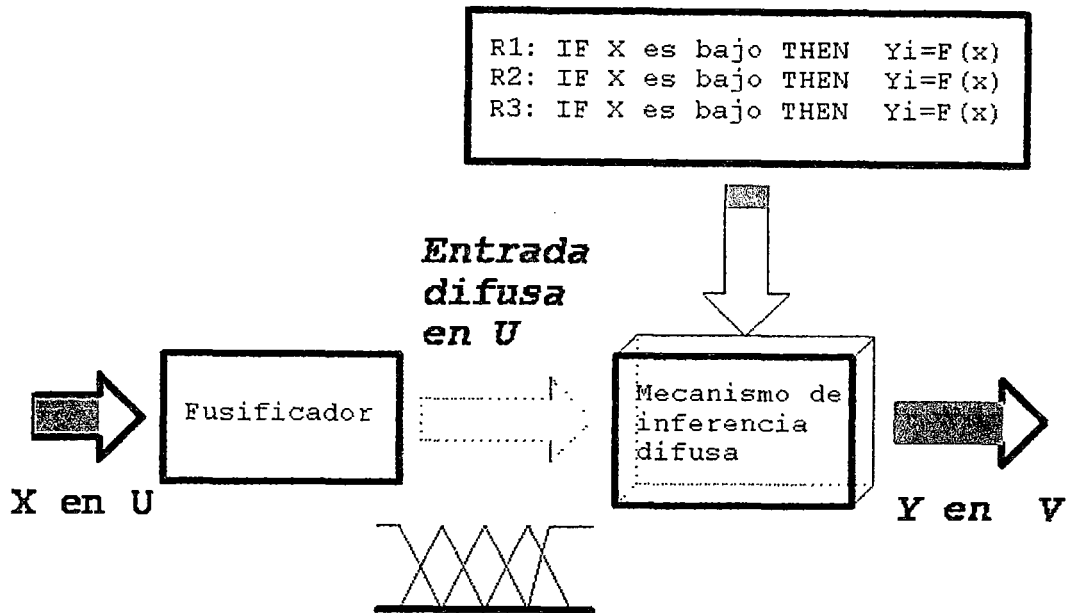


Figura 4.2: Procesamiento general sistema difuso Takagi-Sugeno

En los sistemas difusos tipo Sugeno, los valores que arrojan los consecuentes de las diferentes reglas que se han activado en un momento determinado ya son valores numéricos, por lo que no se necesita una etapa de defuzzificación. En la figura 4.3 se presentan el modelo *Takagi – Sugeno* de dos regla difusas , con dos variables de entrada y una de salida.

Debido a una dependencia lineal en el sistema de variables de entrada , este modelo es ideal para actuar como supervisor interpolador de múltiples controladores lineales que aplican en diferentes condiciones de operaciones de un sistema dinámico no lineal.

En consecuencia para el sistema a controlar denominado Manipulador robótico Traslacional multivariable , se consideró el controlador FLC tipo *Takagi – Sugeno* doble sistema de inferencia en paralelo defuzzificación de orden 0. De acuerdo con una investigación previa [22] se ha determinado que una defuzzificación de orden nulo es suficiente para un buen desempeño del sistema de control, además es más simple de proccsar numéricamente en el DSP o cualquier otro proccsador , lo que permite tener un menor tiempo de cálculo.

Mediante un proceso de simulaciones y experimentación se determinará posteriormente el valor de las constantes de valores de la ganancias de adaptación , que da



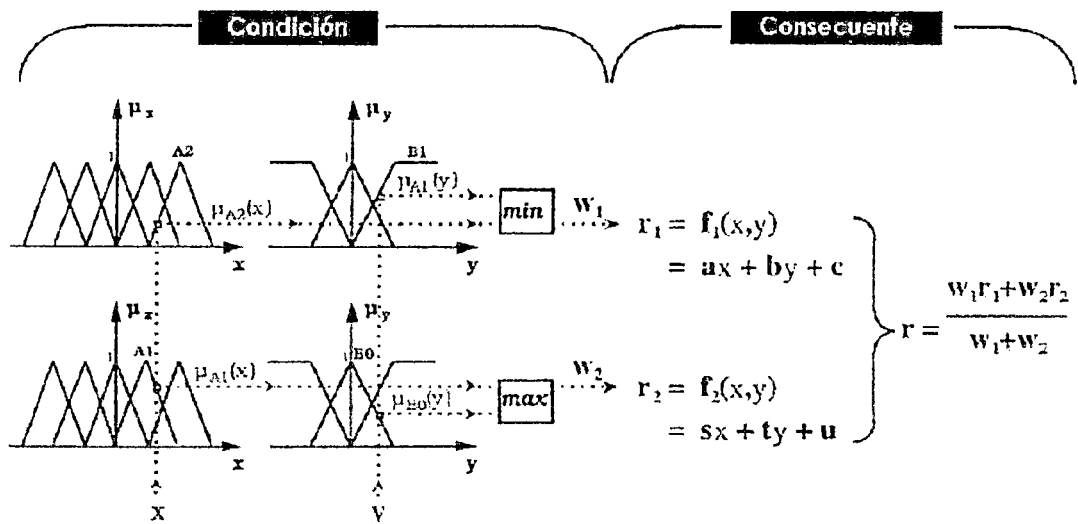


Figura 4.3: Modelo Takagi Sugeno

un factor de escala a las salidas del controlador.

Teniendo dos variables a controlar como son : La posición del carro  $r$  y la posición angular  $\theta$  de la varilla.

El sistema de control en tiempo continuo se puede expresar mediante el diagrama de bloques de la figura 4.4 en el cual se muestra las diferentes etapas a tomar en cuenta.

### Estructura de los lazos de control sobre las variables.

Los controladores basados en Lógica Difusa ofrecen un razonamiento para el control basado en un Set Point Difuso de cada variable que controlan. Este Set Point es inherente y con lleva a dos estructuras de control sobre las variables del sistema. Así, podemos establecer

- Control de posición del carro:  $r$  con set-point constante ó variable
- Control de posición angular :  $\theta$  ángulo de la posición de la varilla
- Compensador de gravedad :  $V_g$

Dichas Estructuras de control se aprecian mejor en el esquema de la Figura.4.4, en donde se ha incluido el modelo del MRTM-2DOF dentro del diagrama de bloques con fines de simulación.

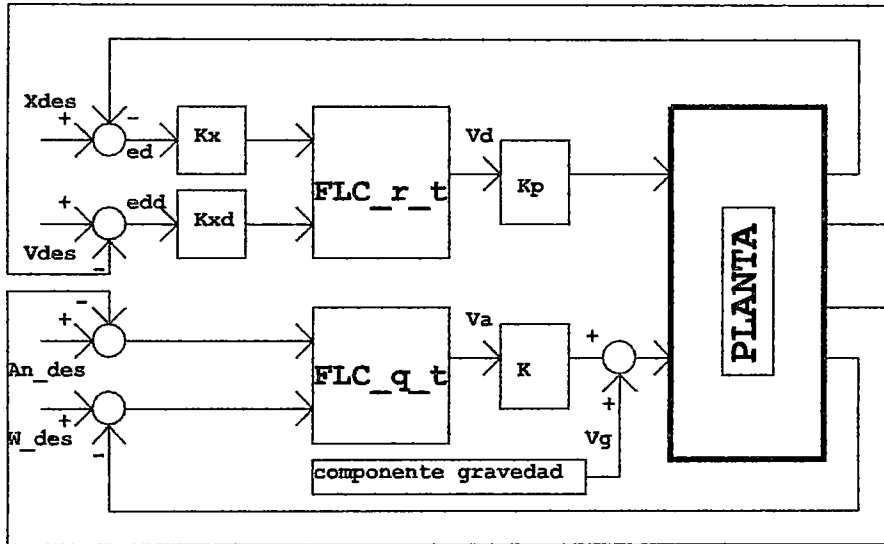
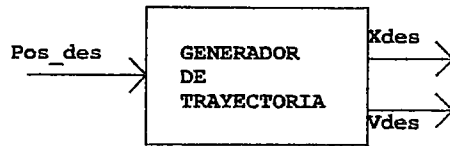


Figura 4.4: Sistema de control en tiempo continuo

### 4.2.1. Controlador difuso de posición

La estructura del controlador de posición del carro se muestra en la figura 4.5.

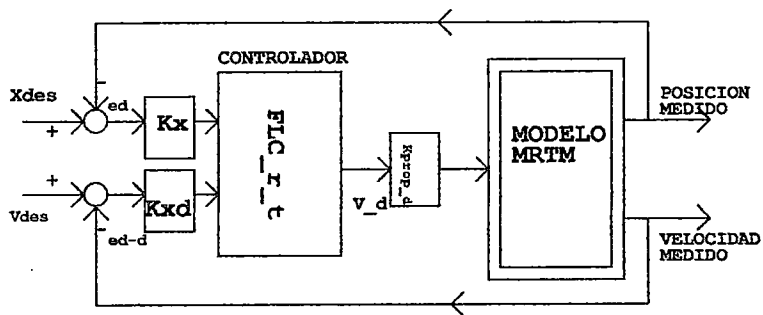


Figura 4.5: Sistema de control de posición

Donde:

$e_d$  = señal error de posición

$ed_d$  = señal error de la derivada ó señal error de velocidad.

$K_x$  = constante de adaptación de entrada antes de la fuzzificación de la posición para darle el nivel adecuado al error de posición.

$K_{xv}$  = constante de adaptación de entrada antes de la fuzzificación de la velocidad para darle el nivel adecuado al error de velocidad.

$K_{prop-d}$  = constante proporcional de la salida para la posición para dar el nivel de tensión adecuado al motor de posicionamiento del carro.

$V_d$  = señal de control, como voltaje de salida para el motor  $M_1$

### **Elección del Universo de Discurso de las Variables de Entrada.**

Es usual escoger universos de discurso estandarizados, p.e. [-1,+1] y luego aplicar factores de escala para cada variable según la planta a controlar .

1. El error de posición denominado **distance** en el programa
2. La Velocidad denominado **distance<sub>v</sub>** en el programa

### **Sistema de Inferencia Difusa del Tipo Takagi-Sugeno para el Control de la Posición**

Para el Sistema de Inferencia Difusa correspondiente al control de la posición se han considerado las funciones de membresía Gaussianas que se muestran en las figuras 4.6 y 4.7 que corresponden a la posición y a la velocidad, respectivamente.

#### **Variable de entrada distancia: distance**

El rango de esta variable es de (-15, 15). Esta variable va a tomar 5 términos lingüísticos, ya que son suficientes para caracterizar esta variable de entrada distancia . Las funciones de pertenencia son gaussianas respectivamente. Los términos lingüísticos utilizados tienen la siguiente matriz

#### **Entrada matriz de parametros de la MSFS de distance**

distance		
-15.0	-0.400	1
0.11	-0.250	0
0.11	0.000	0
0.11	0.250	0
15	0.350	1]

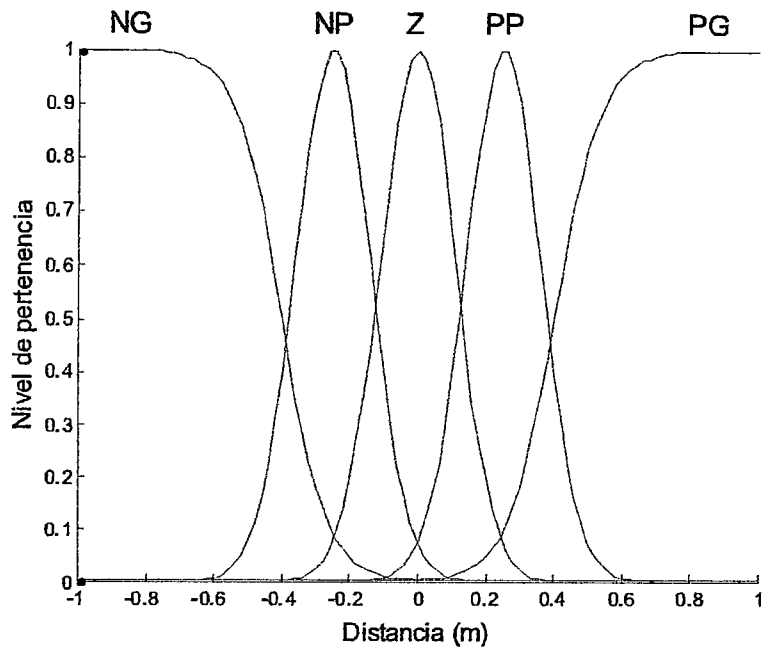


Figura 4.6: Funciones de membresía para la posición lineal

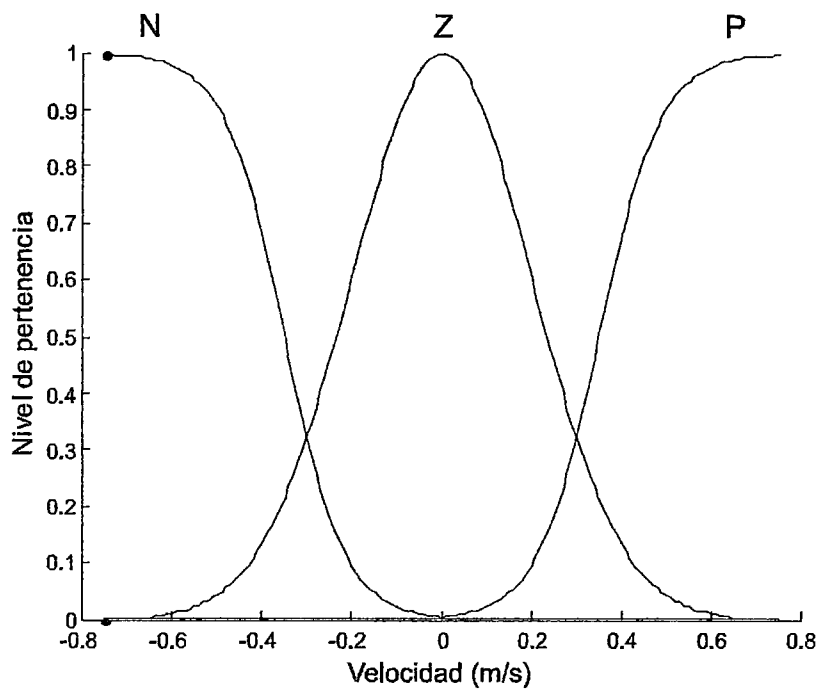


Figura 4.7: Funciones de membresía para la Velocidad lineal

**Funciones de membresía para la velocidad lineal del carro**

**Variable de entrada Velocidad :distance-d**

La variable de entrada distance-d es la derivada de la distancia o sea la velocidad; comprendido en el rango de [-15,15], tomando tres términos lingüísticos , suficiente para caracterizarla, considerando la siguiente matriz

**Entrada matriz de parámetros de la MSFS de distancia-d**

distance-d		
-15.0	-0.350	1
0.200	0.000	0
15	0.350	1]

### Base de reglas del controlador de posición Lineal del carro

La base de reglas para el sistema de inferencia difusa para el control de la posición del carro se ilustra en la figura.4.8 y corresponde al modelo de inferencia de Mamdani [15], que busca el equilibrio de las acciones de control.

		ERROR DE VELOCIDAD		
		N	Z	P
ERROR DE POSICION	NG	NG	Z	NG
	NM	NM	Z	NM
	NP	NP	Z	PP
	Z	Z	Z	Z
	PP	PP	Z	PP
	PM	PM	Z	PM
	PG	PG	Z	PG

Figura 4.8: Base de reglas del controlador de posición del carro o cabezal

Para definir las reglas se ha tenido en cuenta que:

- Un error negativo significa que se obtiene una salida mayor que la referencia, por lo que se debe disminuir la tensión de actuación a la planta.
- Un error positivo significa que se obtiene una salida menor que la referencia, por lo que se debe aumentar la tensión de actuación a la planta.
- Si la derivada del error es del mismo signo que el error, significa que el error está aumentando.
- Si la derivada del error es de signo contrario al error, significa que el error está disminuyendo.
- Si la derivada del error es cero, significa que el error permanece constante.
- Si el error es negativo grande (NG) ó positivo grande (PG) la derivada del error da igual, ya que hay que aplicar la mínima tensión o la máxima tensión respectivamente.

### **Elección del Universo de Discurso de las Variables de Salida. Votaje de Actuación**

Lo que se pretende con la fusificación de los términos lingüísticos de la variable de salida no es lo mismo que con las variables de entrada. Para la variable de salida lo que se pretende es que una vez truncado un término lingüístico, se sepa a qué valor de la variable de salida pertenece ese truncamiento y se pueda construir el consecuente, para posteriormente, mediante el método de defusificación elegido, obtener el valor concreto de salida.

A la hora de definir las funciones de pertenencia hay que tener en cuenta que la mínima salida defusificada que vamos a tener va a ser el centro del término lingüístico situado más a la izquierda, y la máxima salida va a ser el centro del término lingüístico situado más a la derecha.

Posteriormente, un mecanismo de inferencia, tal como la regla composicional de Zadeh, se usa para determinar la acción de control. El paso final involucra la aplicación de un método de defuzzificación.

### **Defuzificación**

La defuzificación que se propone, dentro del esquema de los FIS tipo Takagi- Sugeno, es la de orden 0. Esta defuzificación genera, como salida del FIS, el promedio ponderado de los niveles de pertenencia obtenidos a través del proceso de inferencia y los valores crisp correspondientes a las funciones de membresía de salida  $C_i$ , que son implicadas por la base de reglas. En la figura 4.9 se muestran las funciones de membresía de salida.

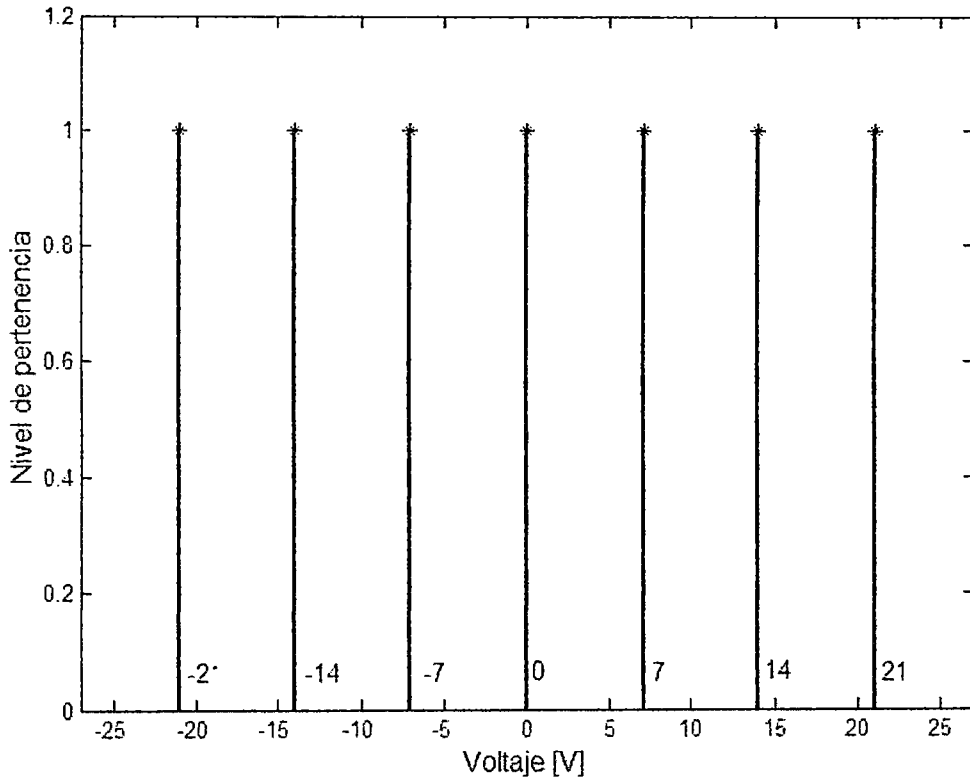


Figura 4.9: Funciones de membresía  $C_i$  para la defuzificación.

La ecuación 4.1 expresa el cálculo de la salida  $u_x$  del FIS de seguimiento, obtenida a partir del promedio ponderado o defuzzificación de orden 0. NC es el número de funciones de pertenencia que describen al conjunto difuso de salida "Voltaje", conformado por las funciones  $C_i$ , y  $\mu_{C_i}$  representa al nivel de pertenencia asociado con cada  $C_i$ .

$$Salida = \frac{\sum_{i=1}^{NC} \mu_{C_i} \cdot z_{C_i}}{\sum_{i=1}^{NC} \mu_{C_i}} \quad (4.1)$$

## 4.2.2. Controlador de posición angular ( brazo)

El esquema general del controlador de posición angular del brazo del sistema BRTM se muestra en la figura 4.10 con la nomenclatura que utilizaremos.

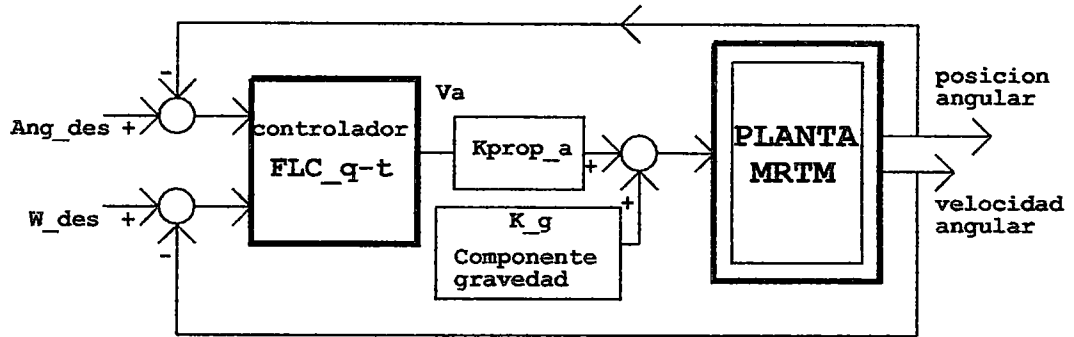


Figura 4.10: Sistema de control de posición angular del brazo del BRTM

donde:

- $ang_{des}$  = ángulo deseado ó referencia.
- $W_{des}$  = velocidad angular deseada.
- $K_g$  = componente de gravedad del brazo (varilla) del BRTM.
- $K_{prop_a}$  = constante proporcional de salida para la señal de control del ángulo.

### Diseño del controlador de posición angular del brazo del BRTM

En primer lugar se identifican las variables de entrada y salida del controlador. La información de la que se dispone es el valor de salida de la planta a controlar, que mediante comparación con la referencia introducida, proporciona el error existente y la derivada del error (velocidad y sentido del cambio del error). Estas serán las entradas del controlador difuso. La salida del controlador será la tensión que se debe aplicar a la planta en función de las entradas, para llegar al objetivo. Para el cálculo del error, se resta la referencia menos la salida de la planta.

$$e = ang_{des} - ang_{controlado} \quad (4.2)$$

donde:

$e$  = error

$ang_{des}$  = ángulo deseado,



$ang_{controlado}$ =ángulo controlado

El cálculo de la derivada del error se realiza a partir de dos muestras consecutivas del error.

$$d_e = \frac{e(actual) - e(anterior)}{Tiempo\ de\ muestreo} \quad (4.3)$$

donde:

$d_e$ = derivada del error

### Fusificación

Conocidas las variables de entrada y salida hay que elaborar el número de términos lingüísticos que van a representar a cada variable y qué tipo de función de pertenencia va a emplear cada uno.

### Sistema de Inferencia Difusa del Tipo Takagi-Sugeno para el control de posición angular

Para el Sistema de Inferencia Difusa para el control de posición angular de la varilla o brazo se han considerado las funciones de membresía Gaussianas que se muestra en la figura 4.11. correspondientes al ángulo en grados sexagesimal

#### Variable de entrada de error de posición angular (e)

Esta variable va a tomar 5 términos lingüísticos, ya que son suficientes para caracterizar esta variable de entrada, Los términos lingüísticos utilizados tienen el siguiente significado.

error de posicion			
NL	[-4.0	-2.00	1
NS	0.8	-0.50	0
Z	0.8	0.0	0
PS	0.8	0.50	0
PL	4	2.0	1]

#### Variable de entrada derivada del error de posición angular ó velocidad angular (angle-d)

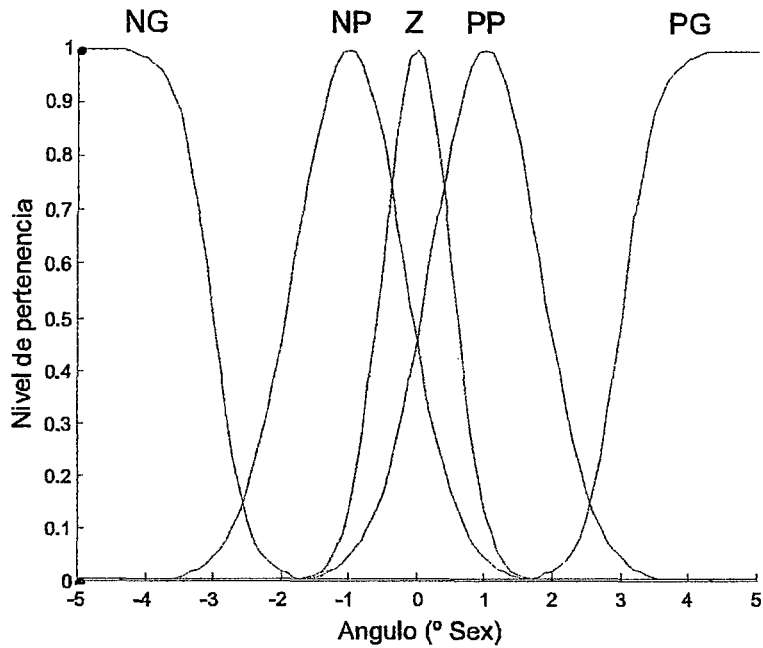


Figura 4.11: Funciones de membresía para desviación angular

Para el Sistema de Inferencia Difusa para el control de la velocidad angular de la varilla ó brazo se han considerado las funciones de membresía Gaussianas tal como se muestra en al figura 4.12.

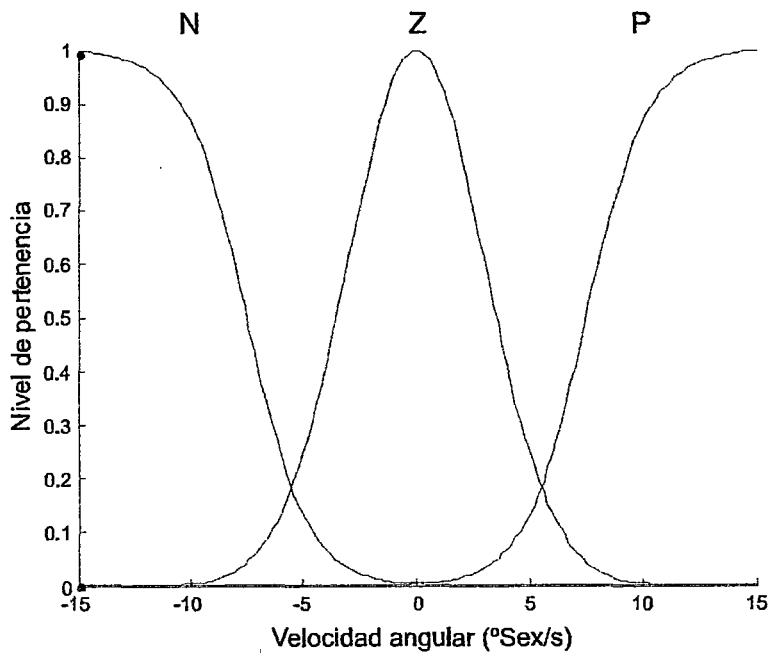


Figura 4.12: Funciones de membresía para la velocidad angular

Esta variable va a tomar 3 términos lingüísticos, suficientes para caracterizarla. Los términos lingüísticos utilizados tienen el siguiente significado y parámetros:

derivada del error de posición ( angle-d)			
N[	-0.75	-0.75	1
Z	3.0	0.0	0
P]	0.75	0.75	1]

#### Base de reglas del controlador de posición angular del brazo ó varilla

Así como se procedió para el control de posición del carro, para el control de la posición del brazo ó varilla se emplea una base de reglas de tipo Mamdani, que se ilustra en la tabla 4.2 es similar al del control de posición lineal

		ERROR DE VELOCIDAD		
		N	Z	P
ERROR DE POSICION	NG	NG	Z	NG
	NM	NM	Z	NM
	NP	NP	Z	PP
	Z	Z	Z	Z
	PP	PP	Z	PP
	PM	PM	Z	PM
	PG	PG	Z	PG

Figura 4.13: Base de reglas del controlador de posición angular de la varilla

## Variable de salida (voltaje<sub>q</sub>)

El rango de esta variable es de (-21,21). Lo que se pretende con la fusificación de los términos lingüísticos de la variable de salida no es lo mismo que con las variables de entrada. Para la variable de salida lo que se pretende es que una vez truncado un término lingüístico, se sepa a qué valor de la variable de salida pertenece ese truncamiento y se pueda construir el consecuente, para posteriormente, mediante el método de defusificación elegido, obtener el valor concreto de salida. Para las pruebas de simulación se ha considerado un vector de voltajes tales:

Vector de voltajes: <i>Voltaje<sub>q</sub></i>							
MSFs = [	1	2	3	4	5	6	7]
<i>Votaje<sub>q</sub></i> = [	-21	-14	-7	0	7	14	21]

## Relaciones Matemáticas

Se considera las siguientes relaciones matemáticas para la simulación del sistema.

## Generador de trayectoria

Para obtener la posición deseada del desplazamiento lineal del carro en eje "x" se genera una trayectoria tipo rampa para ello se considera la ecuación.

$$xdes(actual) = xdes(anterior) + vdes(anterior) * Ts.$$

- $xdes$  = Posición  $x$  deseada ;
- $vdes$  = velocidad deseada ;
- $T_s$  = tiempo de muestreo virtual.

## Relaciones matemáticas del controlador

Se considera las siguientes relaciones:

1.  $tt$  = tiempo del controlador coincidente con el de la simulación del modelo .
2.  $pp$  = posición particular en el eje  $x$
3.  $vp = (pp - pp_{ant}) / t_s$ ; velocidad para posición particular  $x$

4.  $ascx = X(i,4) * 57.2958$  ,posición angular para una posición  $x$  particular.

5.  $wsex = X(i,5) * 57.2958$  ;velocidad angular para una posición  $x$  particular

### Calculo de las señales de error

Las señales de error para la posición  $x$  y para la posición angular del brazo  $\theta$  se expresa:

1.  $ed = Kx * (xdes - pp)$ , error de posición en  $x$ .
2.  $ed-d = Kxd * (vdes - Vp)$ , error de velocidad en  $x$
3.  $eq = (ang - des) - asex$ , error de posición angular  $\theta$

Donde:

$Kx$  = constante de adaptación de error de posición lineal.

$Kxd$  = constante de adaptación de velocidad.

### Evaluación de controlador fuzzy

- **Evaluación del control fuzzy para la posición lineal**

Para implementar y evaluar el controlador fuzzy se ha realizado los cálculos para obtener la señal de control como voltaje para el motor de posición lineal denominado  $V_d$  ( voltios) y para el motor de posición angular denominado  $V_a$  ( voltios).

Para obtener la señal de control de posición  $x$  denominado  $V_d$  se ha considerado los siguientes parámetros.

1.  $ed$  = error de posición
2.  $ed_d$  = error de velocidad
3.  $distance$  = función de membresía de posición
4.  $distance_d$  = función de membresía para velocidad
5.  $BR_r$  = base de reglas de posición lineal
6.  $Voltaje_r$  = base de reglas para la salida como voltaje

La función que realiza la evaluación es:

$$v_d = FLC_t(e_d, ed_d, distance, distance_d, RB_r, voltaje_r)$$

- Evaluación del control fuzzy para la posición angular

Para obtener la señal de control de posición angular  $\theta$  denominado  $V_q$  se considera los siguientes parámetros.

1.  $e_q$  = error de posición angular
2.  $w_{sex}$  = error de velocidad angular
3.  $angle$  = función de membresía de posición angular
4.  $angle_d$  = función de membresía para velocidad angular
5.  $BR_q$  = base de reglas de posición angular
6.  $Voltaje_q$  = base de reglas para la salida como voltaje de posición angular

La función que realiza la evaluación es:

$$v_q = FLC - t(e_q, -wsex, angle, angle_d, RB_q, voltaje_q)$$

### Adaptación de los niveles de tensión de control

En cuanto a la señal de control para la posición lineal y angular se debe realizar una adaptación para darle el nivel adecuado, luego se debe considerar las constantes de adaptación adecuada para cada caso:

\*Adaptación de la señal de control de posición  $u_1$

$$u_1 = (K_{prop-d}) * (V - d);$$

$K_{prop-d}$  = contante de adaptación de posición de salida.

\*Adaptación de la Señal de control de posición angular  $u_2$ .

La ecuación matemática para adaptar la señal de control de posición angular se considera en la ecuación (4.4)

$$u_2 = (K_{prop-a}) * (V_q) - K_g * \sin(\theta) \quad (4.4)$$

donde:

$K_g$  = es la constante de componente de gravedad de la varilla acoplado al eje del motor, está dado por la ecuación(4.5)

$$K_g = \frac{l * m * g * ra}{2 * Km} \quad (4.5)$$

### 4.2.3. Compensador de gravedad

Considerando el gráfico de la figura 4.14 se indica que el peso de varilla hace que por gravedad tienda girar en sentido horario, por lo que se debe considerar un voltaje en sentido contrario para compensar esta desviación, para lo cual se tienen la siguiente relaciones:

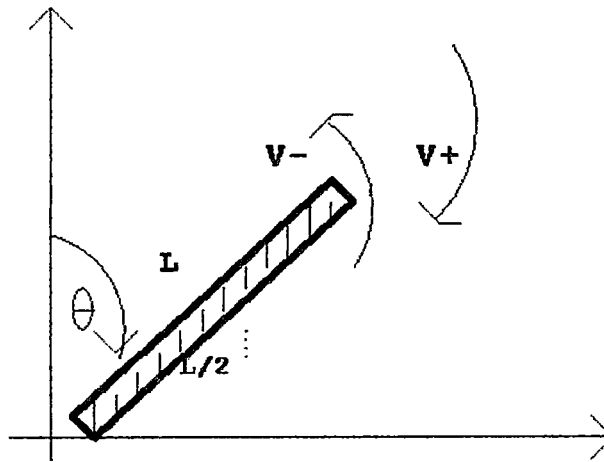


Figura 4.14: Compensación por gravedad del brazo

$$T_g = K_{vg} * V_g \quad (4.6)$$

$$T_g = (L/2) * m * g * \sin(\theta). \quad (4.7)$$

El torque por gravedad también se puede escribir

$$Tg = i * Km. \quad (4.8)$$

$$Vg = (Tg * Ra) / Km \quad (4.9)$$

Luego la ecuación del voltaje de compensación se puede escribir considerando la figura 4.14

$$V_g = (L/2) * m * g * \text{sen}\theta * ia / Km. \quad (4.10)$$

$$K_g = (L/2) * m * g * ia / Km. \quad (4.11)$$

La expresión anterior se puede escribir como

$$Vg = Kg * \text{sen}\theta \quad (4.12)$$

$$(4.13)$$

donde Kg=constante de compensación de gravedad y Reemplazando los valores se tiene que  $Vg = -0,139$



## Capítulo 5

# Diseño del control difuso discreto

Esta sección presenta la discretización del sistema de control en tiempo discreto de acuerdo a la estructura mostrada en la figura 5.1

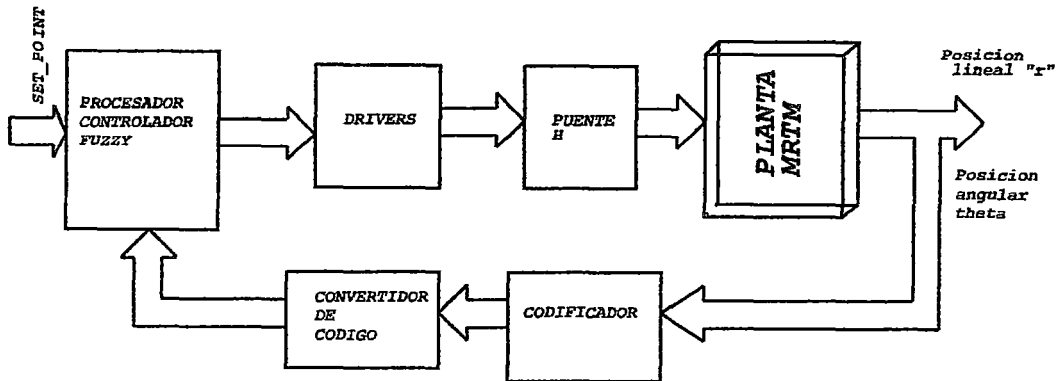


Figura 5.1: Sistema de control en tiempo discreto

### 5.1. Discretizacion del sistema continuo

En primer lugar se explican las consideraciones tomadas en la discretización de los sensores y en la elección del período de muestreo.

En segundo lugar la simulación del controlador discreto consigna el uso de de dos períodos de muestreo, uno para la planta y otro para el controlador.

El período de muestreo de la planta es de 4 ms y el del controlador se escoge finalmente como 30Hz, tras evaluar las simulaciones del sistema en tiempo continuo. Como referencia, en el apéndice se anexan los programas en MATLAB usados para estas simulaciones.

#### Discretización de los sensores Sensores

Los sensores empleados son de naturaleza discreta. Los encoders permiten contar un número entero de fracciones de revolución, de este modo, al obtener directamente la posición a través del conteo de pulsos, se computa un valor que es proporcional

al número de pulsos de la señal en cuadratura.

La doble señal en cuadratura de los éncoders a emplear permite definir el desplazamiento angular no sólo en magnitud, sino también en sentido doble.

El sensado tiene el decodificador de cuadratura que se encarga de leer los datos del encoder óptico para emitir salidas *UP/DOWN* luego conectar al puerto de entradas digitales de un procesador . El codificador como la mayoría de los casos, incluye dos pistas de marcas o ranuras en cuadratura (canales A y B), cada una asociada a una fuente de luz y un fotodetector independientes. Los canales A y B producen dos ondas cuadradas desfasadas en 90 grados entre sí que se utilizan para determinar la dirección del movimiento. Por ejemplo si el canal B adelanta al canal A, el eje estará girando en el sentido antihorário y viceversa.

Cada motor debe disponer de un sensor para la velocidad y posición ,luego para la simulación se considera las siguientes relaciones matemáticas:

#### **Selección sensor motor1: Sensado de Posición lineal**

**número de pulsos del encoder1** Para medir la posición lineal del carro. en el programa se considera el sensor 512 pulsos por vuelta denominado.

$$npulse\_encoder1=512.$$

**Multiplicador QEP** se considera  $qeq =2$ .

**Diámetro de la polea** Se considera una polea denominado  $dpolea$  en el programa, con radio de  $0.084$   $dpolea=0.084*2$ .

**Número de pulsos considerando factores** luego se plantea la relación

$$m2pulse=npulse\_encoder1 * qeq * 19,745/(pi * dpolea)4.$$

#### **Selección Sensor motor2 : posición angular**

- Número de pulsos del encoder2 para la posición denominado en el programa  $npulse\_motor2=512$ . se considera 512 pulsos por vuelta.
- Multiplicador del tacómetro para el sensor2 de posición angular en el programa se asigna  $ls714=4$ .
- Número de pulsos considerando factores:  
 $gsex2pulse=npulse\_encoder2*ls7184/360$ .

## Muestreo de las señales sensadas para la simulación

Se considera la frecuencia de muestreo  $f_{sc}$  ó  $tsc=1/fsc$  las relaciones de velocidad del motor 1 y del motor 2 serán.

- $V2pulse= npulse\_encoder1*qe q*5.9/(pi*polea)*tsc.$
- $wsex2pulse=npulse\_encoder2*ls7184/360*tsc;$

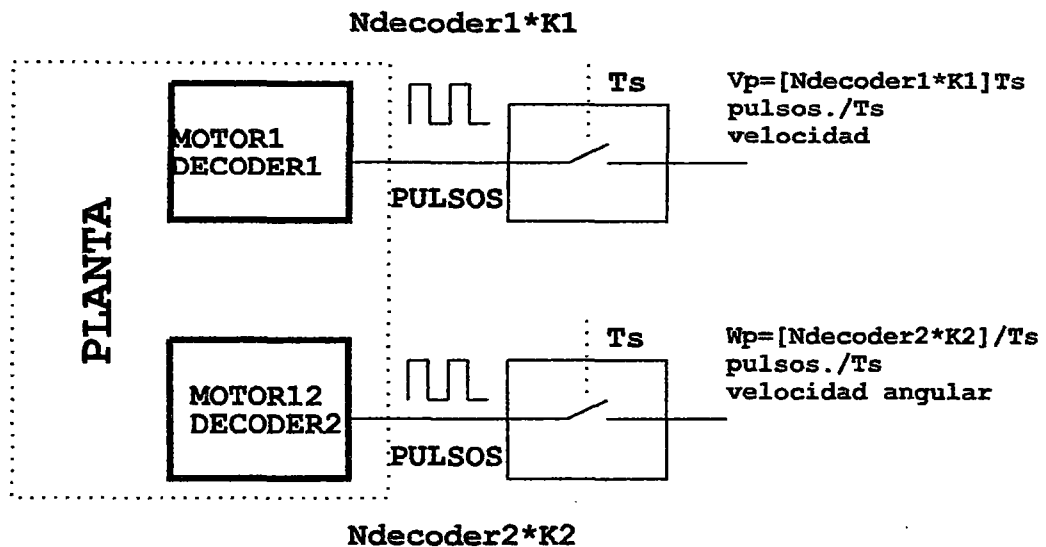


Figura 5.2: Sensores de motores 1 y 2

## 5.2. Control difuso de posición lineal del carro y de posición angular del brazo

En esta etapa se convierte todas las variables de magnitudes físicas a número de pulsos (discretizar) luego se tiene :

- $x(k)$  posición lineal del carro actual.
- $x(k - 1)$  posición anterior del carro
- $v(k) = x(k) - x(k - 1)/ts$
- $\theta(k)$  posición angular del brazo o de la varilla
- $w(k)$  velocidad angular

## Relación de las señales de error en tiempo discreto

Las ecuaciones que se tienen para determinar las señales de error del sistema que son señales de entrada al controlador difuso se expresan:

1.  $e_{d(k)} = Kx^*(x_{des}(k) - pp(k))$ , error de posición en  $x$ .
2.  $x_{des(k)} = m2pulse * x_{des}$ .
3.  $e_{d-d(k)} = Kxd^*(v_{des}(k) - Vp(k))$ , error de velocidad lineal en  $x$ .
4.  $v_{des(k)} = v_{des} * v2pulse$ .
5.  $e_q(k) = \theta_{des}(k) - \theta(k)$ , error de posición angular  $\theta$ .

En la figura 5.3 se muestra las señales de entrada y salida del controlador difuso discreto.

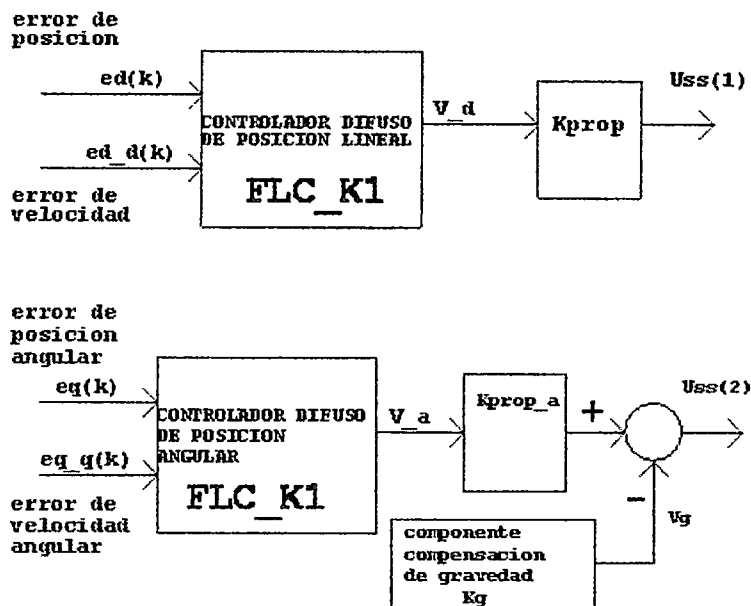


Figura 5.3: Señales del Controlador logico difuso entrada y salida

Funciones de membresía del controlador difuso de posición lineal del carro y posición angular del brazo

Considerando las funciones de membresía del sistema de control continuo, para el sistema discreto se tiene que realizar la transformación de la funciones de membresía

del sistema continuo al universo de números enteros.

### Constantes de transformación

Las constantes de transformación para la posición angular y lineal son:

1. secuencia de pulsos del motor1 para posición lineal denominado:  $m2pulse$
2. Secuencia de pulsos del motor2 para la posición angular denominado :  $gsex2pulse$ .
3. Velocidad de pulsos para posición lineal denominado:  $V2pulse = m2pulse * Tsc$ .
4. Velocidad de pulsos para la posición angular denominado:  $Wsex2pulse = gsex2pulse * tsc$ .

Parámetros para calcular las constantes son:

1. Número de pulsos del encoder del motor1 denominado  $npulse\_encoder1 = 512$ .
2. Número de pulsos del encoder del motor2 denominado  $npulse\_encoder2 = 512$ .
3. Diámetro de la polea de la faja :  $dpolea = 0.048 * 2$ .
4. Relación de engranaje para la posición  $n1 = 19.745$ .

### Tamaños de las funciones de membresía

Se tiene que determinar el tamaño de las funciones de membresía de las entradas del controlador difuso de posición lineal del carro y de la posición angular del brazo de la varilla, luego podemos indicar lo siguiente:

1. Tamaño de la función de membresía de posición :  
 $sizep = size(distance), sisep = sisep(1, 1)$ .
2. Tamaño de la función de membresía de velocidad :  
 $sizev = size(distance\_d), sisev = sisev(1, 1)$ .
3. Tamaño de la función de membresía de posición angular:  
 $sizea = siz(angle), sisea = sisea(1, 1)$ .
4. Tamaño de la función de membresía de velocidad angular:  
 $sizew = size(angle\_d); sizew = sizew(1, 1)$ .

Para discretizar las funciones de membresía se debe conservar la última columna de cada función de membresía ( $FM$ ) luego se tiene.

- $distance_{pulsos}(:,3) = distance(:,3)$ .
- $distance\_d\_pulsos(:,3) = distance\_d(:,3)$ .
- $angle\_pulsos(:,3) = angle(:,3)$
- $angle\_d\_pulsos(:,3) = angle\_d(:,3)$

Luego debe transformarse el offset ("C") de cada función de membresía (MF) entonces debe realizarse las siguientes relaciones.

- $distance\_pulsos(:,2) = distance(:,2) * m2pulse$ .
- $distance\_d\_pulsos(:,2) = distance\_d(:,2) * v2pulse$
- $angle\_pulsos(:,2) = angle(:,2) * gsex2pulse$
- $angle\_d\_pulsos(:,2) = angle\_d(:,2) * wsex2pulse$

Luego debe transformarse las ganancias de cada función de membresía mediante las siguientes relaciones.

- $distance\_pulsos(K,1) = distance(k,1) * m2pulse * 10000 / m2pulse$ , donde k varia de 1 a 5.
- $distance\_d\_pulsos(k,1) = distance\_d(k,1) * 10000 / v2pulse$ , donde K varia de 0 a 5
- $angle\_pulsos(k,1) = angle(k,1) * 10000 / gsex2pulse$ , donde k varia de 0 a 5
- $angle\_d\_pulsos(k,1) = angle\_d(:,2) * 10000 / wsex2pulse$ , k varia de 0 a 5

### 5.2.1. Evaluación del controlador fuzzy discreto

Las señales de salida del controlador fuzzy para la posición lineal del carro y de posición angular de la varilla se obtiene considerando las relaciones siguientes.

- $V\_d = FLC\_k1(ed(i,1), 0 * ed\_d(i,1), distance\_pulsos, distance\_d\_pulsos, RB\_r, Voltaje\_r)$

Donde los parámetros para evaluar son.

1.  $e_{d(i,1)}$  es error de posición discreto lineal .
2.  $e_{d-d(i,1)}$  = error de velocidad lineal

3.  $distance_{pulsos}$  = función de membresía de posición lineal
4.  $distance_{d\_pulsos}$  = función de membresía de velocidad lineal.
5.  $RB_r$  = bases de reglas para posición lineal del carro
6.  $Voltaje_r$  = bases de reglas como voltaje de salida para posición lineal

Donde la  $V_d$  es el voltaje de salida del controlador fuzzy y el voltaje par actuar sobre el proceso esta dado por:

- $u1 = Kprop * V_d$ , voltaje para controlar el motor 1

Pra calcular el voltaje de salida del controlador fuzzy para la posición del brazo esta dado por la relación.

- $V_a = FLC_k1(eq(i,1), 0, angle_{pulsos}, angle_{d\_pulsos}, RB_q, Voltaje_q)$

los parametros para evaluar son:

1.  $e_{q(i,1)}$  es el error de posición angular discreto .
2.  $e_{q-d(i,1)}$  = error de velocidad angular = 0
3.  $angle_{pulsos}$  = función de membresía de posición angular
4.  $angle_{d\_pulsos}$  = función de membresía de velocidad angular.
5.  $RB_q$  = bases de reglas para posición angular del brazo
6.  $Voltaje_q$  = función de membresía del voltaje de salida para posición angular

La adaptación de los niveles de tensión al motor 2 tiene la siguiente relacion

- $u2 = 0,5 * V_a - 0,1398 * sen(\theta)$  para controlar el motor 2
- $ug$  = voltaje de compensacion de gravedad

## Capítulo 6

# Resultados

### 6.1. Resultados parciales

#### 6.1.1. Resultados del sistema de control difuso en tiempo continuo en lazo cerrado

Mediante el programa en Matlab, denominado `mrtm_FLC_c.m`, nos permite obtener la respuesta mostrada en la figura 6.1. el programa `mrtm_FLC_c.m` requiere de las siguientes funciones:

- `FLC_t.m` : Que a su vez necesita de `MF_t.m`
- `f_mrtm2dof`: función que determina la aceleración para la posición y ángulo

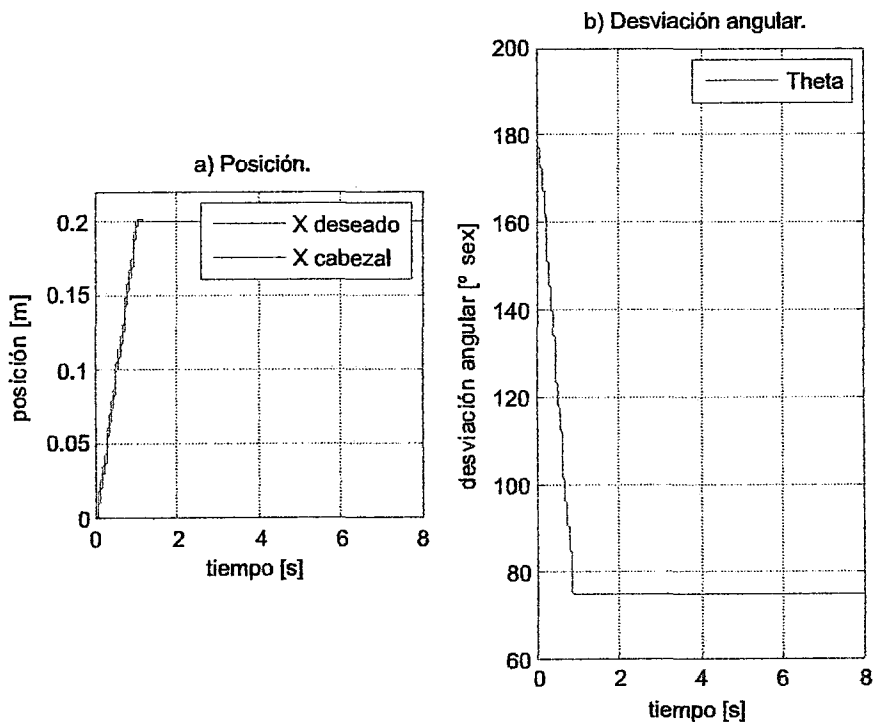


Figura 6.1: Posición lineal y angular del brazo robótico en lazo cerrado de tiempo continuo

La evaluación se realizó considerando los siguientes valores de referencia:



- $X_{deseado} = 0,2m$
- $\theta_{deseado} = 75\text{grados}$

El programa también nos muestra la señal de control de posición lineal del carro y la señal de control para la posición angular del brazo. (ver figura 6.2).

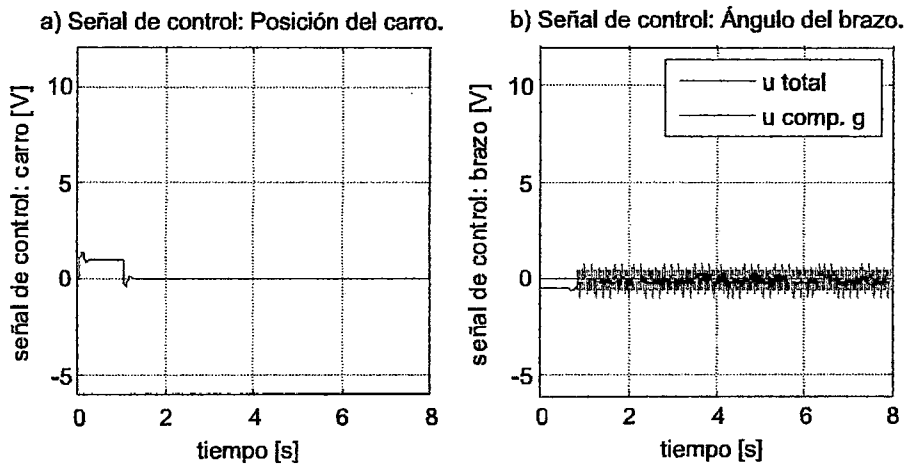


Figura 6.2: Señales de control de posición lineal y angular del brazo robótico

El programa incluye la simulación en forma animada del sistema tal como se muestra en la figura 6.3

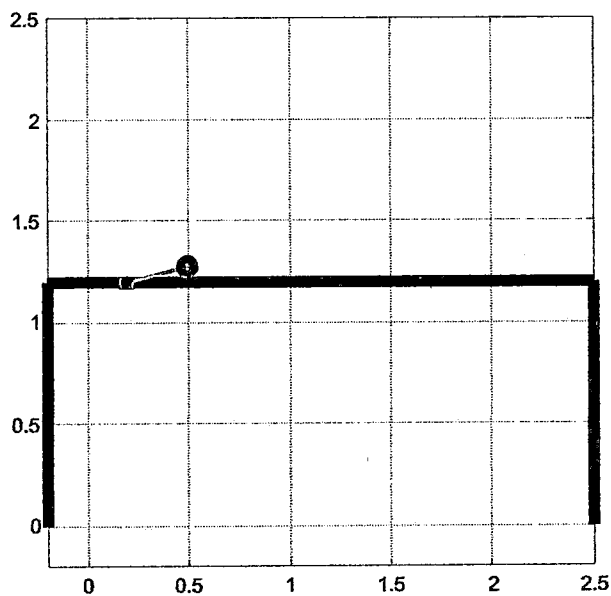


Figura 6.3: simulación del sistema en forma animada

## 6.1.2. Resultados del sistema de control difuso en tiempo discreto en lazo cerrado

Mediante el programa en Matlab, denominado `mrtm_FLC_d3.m`, se obtiene la respuesta del sistema y se muestra en la figura 6.4

El programa `mrtm_FLC_d3.m` requiere de las siguientes funciones:

- Función `FLC_k1.m`: Que a su vez necesita de la función `:MF_k.m`
- Función `f_mrtm2dof`: Que determina la aceleración para la posición y ángulo

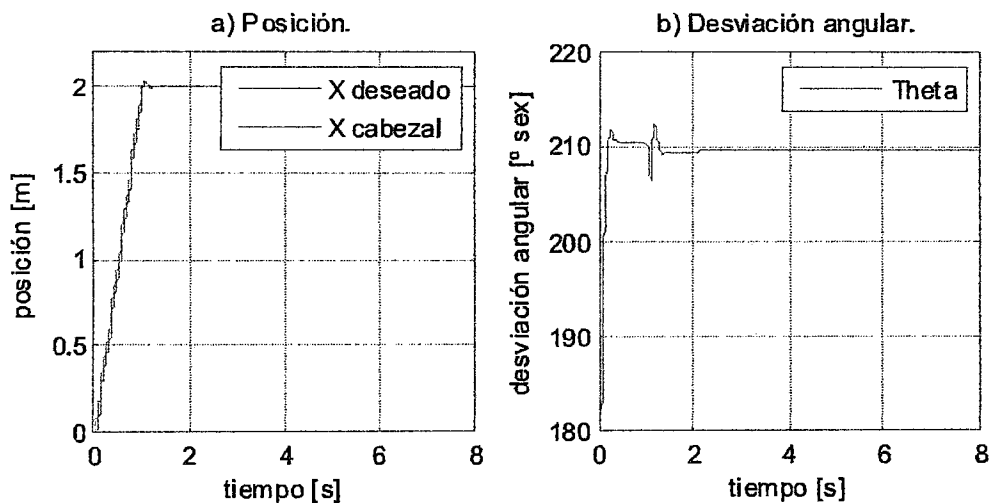


Figura 6.4: Posición lineal y angular del brazo robótico en lazo cerrado tiempo Discreto

La evaluación se realizó considerando los valores de referencia:

- $X_{deseado} = 2m$
- $\theta_{deseado} = 210\text{grados}$

El programa también nos muestra las señales de control de posición lineal del carro y brazo (ver figura 6.5)

En la figura 6.6 se muestra también la animación de la posición del brazo robótico para los valores de referencia  $X_{des}=2$  m y  $\theta = 210$  grados sexg.

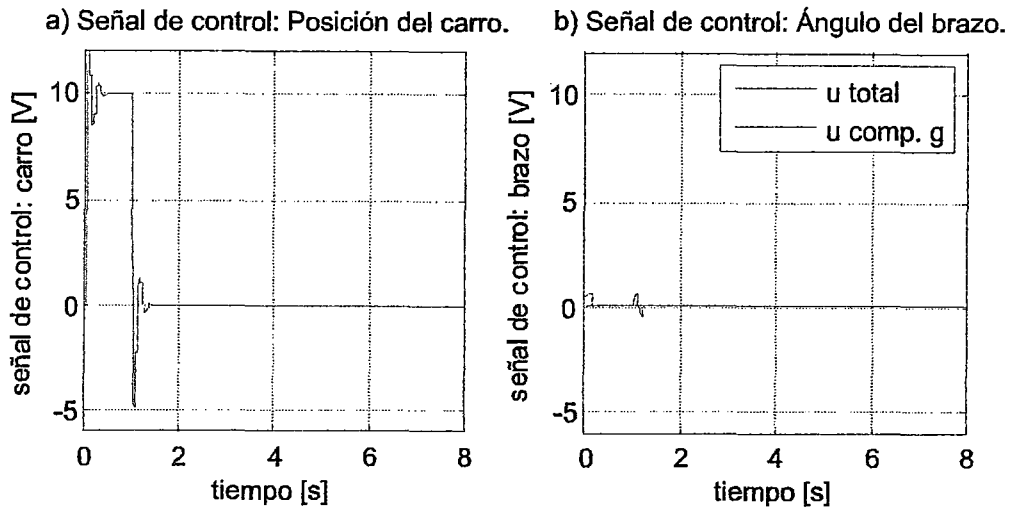


Figura 6.5: Señal de control de posición lineal y desviación angular del brazo robótico para el sistema discreto

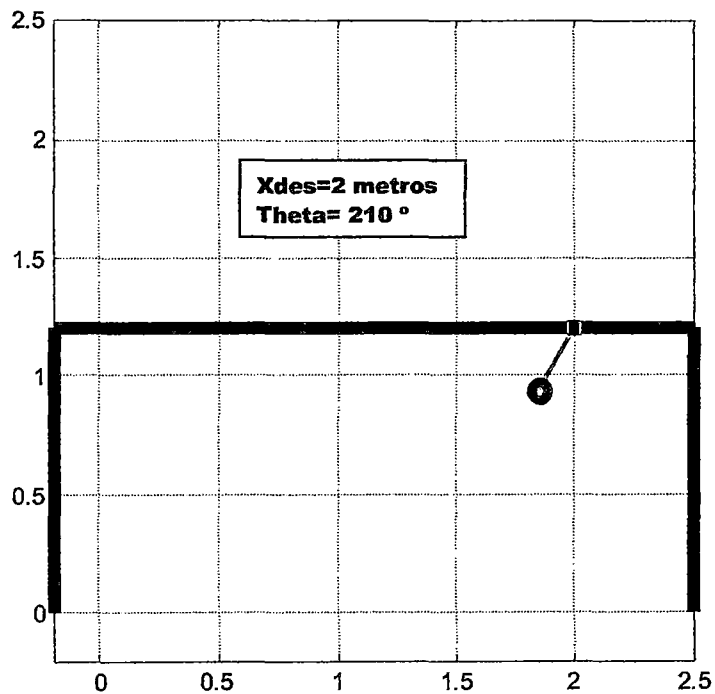


Figura 6.6: Posición lineal y angular el sistema Brazo Robótico en forma animada tiempo discreto

## Experimento: Sistema Discreto con Señal de Referencia Senoidal

Las trayectorias deseadas senoidales para el carro y para el brazo son:

- Posición deseada del carro : Amplitud de la posición deseada = 2.0 m

- Posición deseada del brazo : Amplitud de la posición angular deseada =45 grados
- Periodo de Prueba del experimento = 8 seg.
- Frecuencia de muestreo del controlador = 100 Hz

La figura 6.7 nos muestra la salida controlada realizando un seguimiento a la señal deseada del sistema para la posición del carro y posición angular del brazo.

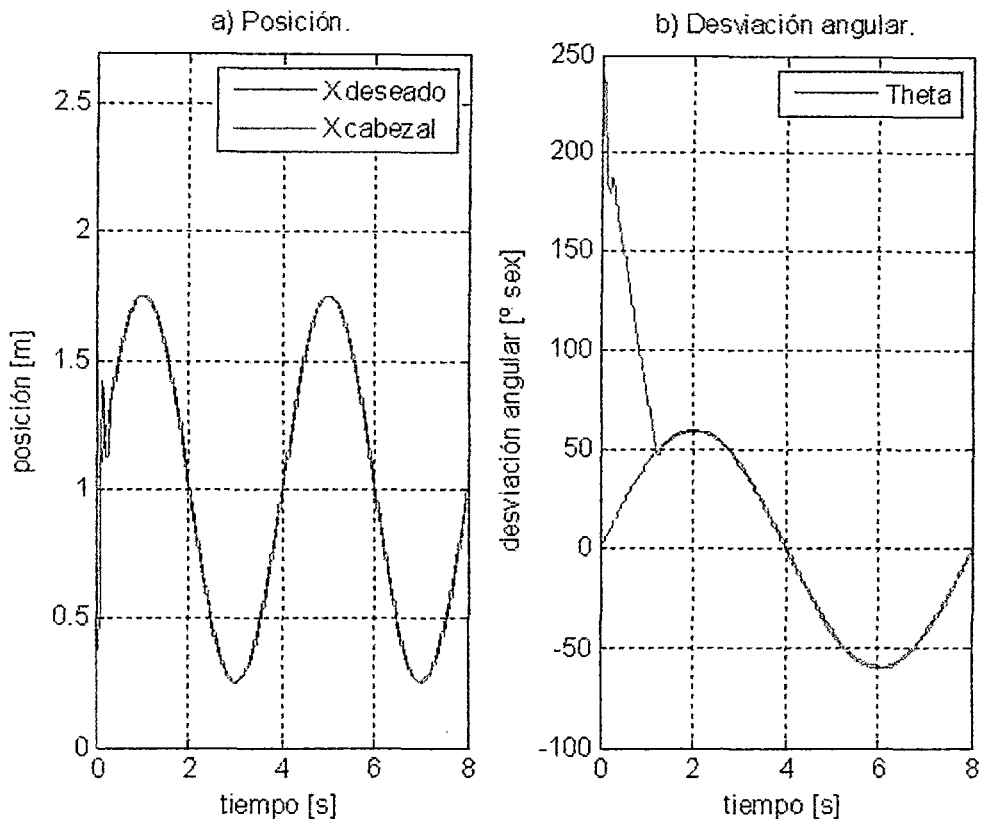


Figura 6.7: Trayectoria controlada para el carro y el brazo

La figura 6.7 muestra la ley de control del sistema. La oscilación autosostenida de la ley de control es abrupta en un instante de tiempo corto.

Se observa el gran esfuerzo que hacen los servomotores para que el sistema sea controlado, es decir los ejes están conmutando en ambos sentidos a una frecuencia de 40Hz haciendo que el sistema sea controlado. La amplitud de las oscilaciones de la señal de control para el ángulo del brazo es más pequeño

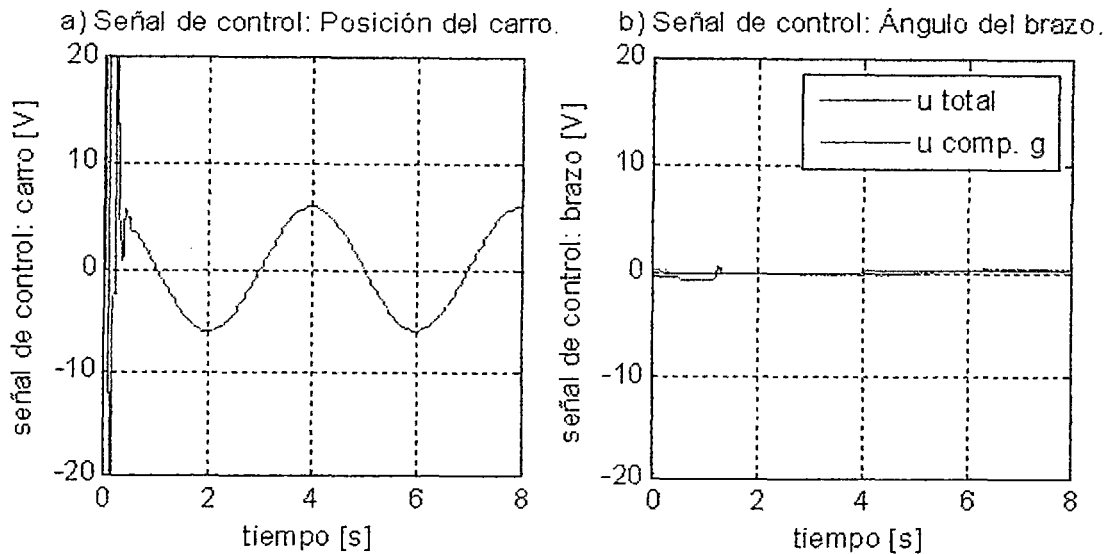


Figura 6.8: Señales de control para el carro y para el brazo.

### 6.1.3. Señales de los sensores para el desplazamiento lineal y angular del Brazo robótico

Para obtener las señales de los sensores de los motores para el desplazamiento lineal y angular del brazo se implemento el programa denominado : **mrmtm2dof-discrete-sensors.m**,este programa nos permite obtener las señales de los sensores en forma discreta.

Para el control de la posición lineal del carro y posición angular del brazo se consideran referencias de tipo senoidal tal como se muestran en al figura 6.8.

En la figura 6.9 se muestra las señales discretas de los sensores de posición lineal y angular para la frecuencia de muestreo  $f_{sc} = 50HZ$ .

En la figura 6.10 se muestra las señales discretas de los sensores de posición lineal y angular para la frecuencia de muestreo  $f_{sc} = 150HZ$ .

## 6.2. Resultados Finales

De los resultados parciales podemos concluir que nos permite evaluar el desempeño del sistema en lazo abierto y lazo cerrado en tiempo continuo y discreto considerando la estrategia de control fuzzy.

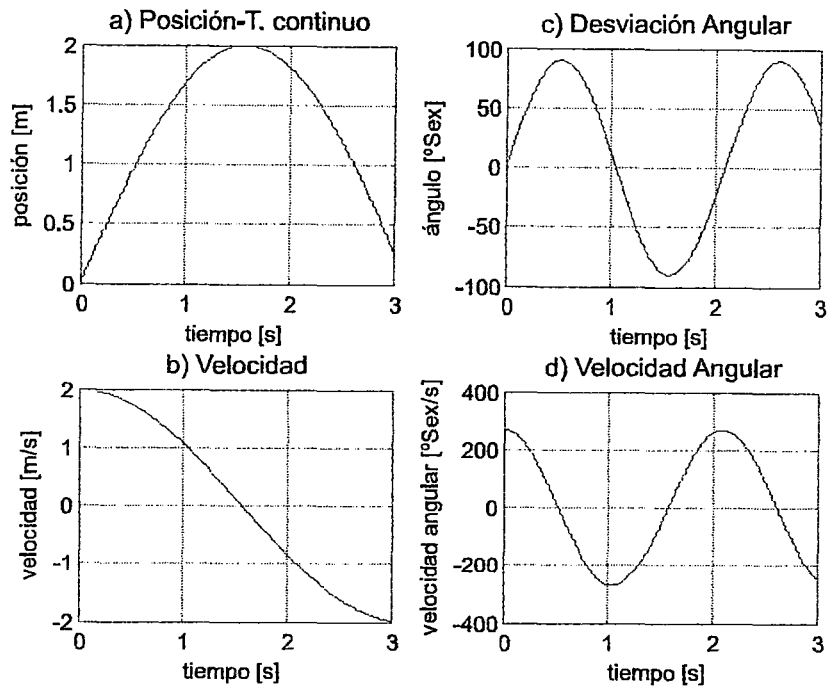


Figura 6.9: Señales de los sensores en tiempo continuo para posición lineal y angular del brazo

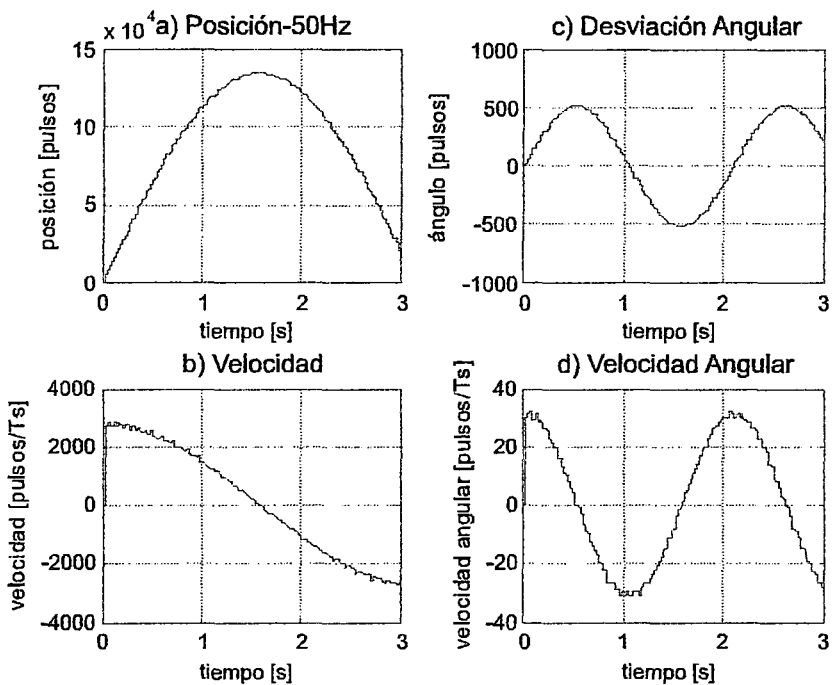


Figura 6.10: Señales de los sensores en tiempo discreto

En la figura 6.12 se muestra la evaluación del sistema en lazo abierto; en la figura se observa que la variable desplazamiento lineal y angular crece indefinidamente, el

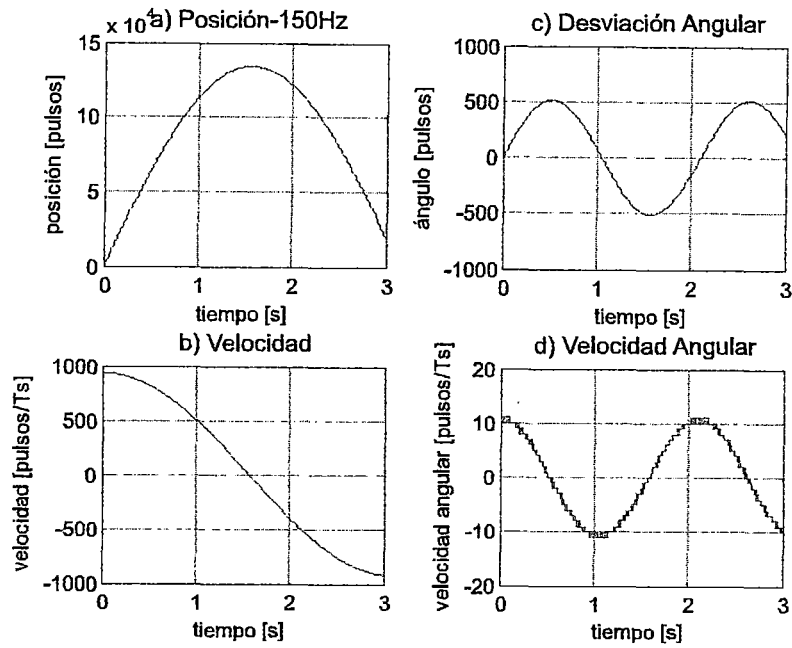


Figura 6.11: Señales de los sensores en tiempo discreto

sistema es inestable.

En la figura 6.1 y figura 6.4 el sistema de control es de lazo cerrado en tiempo continuo y discreto las variables desplazamiento lineal y angular del brazo siguen a las respectivas referencias, con error prácticamente cero y el tiempo de respuesta transitoria aproximadamente a 1 seg.

Con respecto al comportamiento de las señales de control para la posición lineal del carro y angular del brazo se muestran en las figuras 6.2 y 6.5 las señales de control son inicialmente grande por ser el error grande y luego tiende a cero cuando el error disminuye a cero ó cuando la variable controlada y deseada son iguales.

**El tiempo transitorio:**  $T_s = 0,95s$

La frecuencia de muestreo al aumentar, se observa que el sistema se comporta mejor para una frecuencia  $F_{sc} = 100Hz$ . con respecto a la frecuencia de 50Hz.

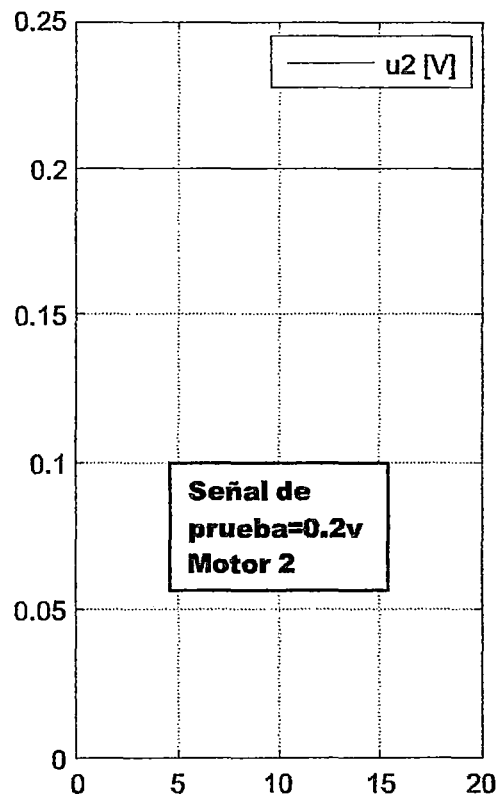
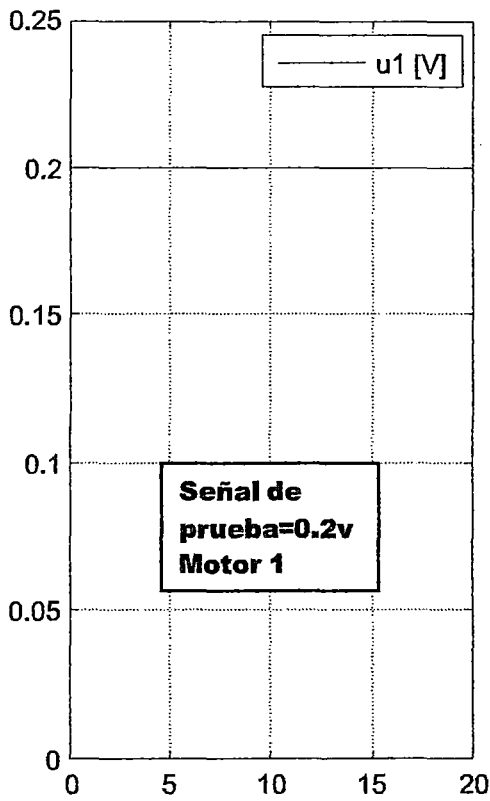
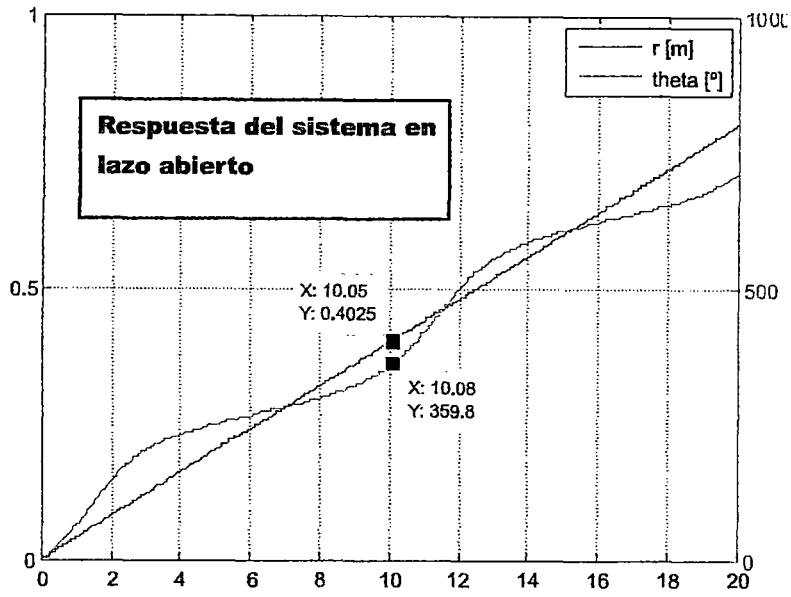


Figura 6.12: sistema de control en lazo abierto



## Capítulo 7

# Discusion de Resultados

### 7.1. Contrastación de hipótesis con los resultados

De los resultados mostrados en el cuadro 4.1, se puede observar que la técnica de Control Fuzzy es el adecuado para esta aplicación, ya que se cumple con los requerimientos de funcionamiento, con un sobrepico nulo, y un tiempo de establecimiento 0.95seg, cumpliendose con lo previsto. En tal sentido, se ha cumplido con la hipótesis planteada en el trabajo, el cual paso a anotar:

”La aplicación de estrategias de control fuzzy permitirán controlar la posición lineal y angular del Brazo Robótico translacional de segundo orden denominado BRT2DOF a niveles de referencia deseados, y error estacionario nulo en tiempo continuo y tiempo discreto.

### 7.2. Contrastación de resultados con otros estudios similares

En esta sección mostramos las simulaciones del sistema de control por modo deslizante para el sistema BRT2DOF tomado de referencia [5] y las simulaciones del sistema de control por lógica difusa. En la primera parte se encuentra el desarrollo de las simulaciones utilizando la ley de control original, es decir, sin aplicar el método de supresión de chattering.

Todas las simulaciones del sistema controlado son respuesta a una referencia arbitraria constante, con el propósito de realizar una mejor comparación de los resultados de las simulaciones. El desarrollo de las simulaciones se estructuran en tres experimentos:

## 7.2.1. Experimento1

### I.-Sistema de control por modo deslizante para el sistema BRT2DOF

Las condiciones iniciales son  $x_1 = 0$  m para el carro y  $x_2 = 0$  rad para la posición angular del brazo, estas condiciones serán utilizadas para los demás experimentos.

Las trayectorias constantes deseadas para el carro y para el brazo son  $x_{d1} = 0,1$ m y  $x_{d2} = \pi/4$ rad respectivamente. El tiempo utilizado en la prueba es de 50 seg.

La figura 7.2.1 muestra las salidas controladas para la posición del carro y posición angular brazo, luego de un tiempo de 4.5 seg y 2.5 seg respectivamente. Se observa que en la posición del carro y posición del brazo no existe sobrepaso.

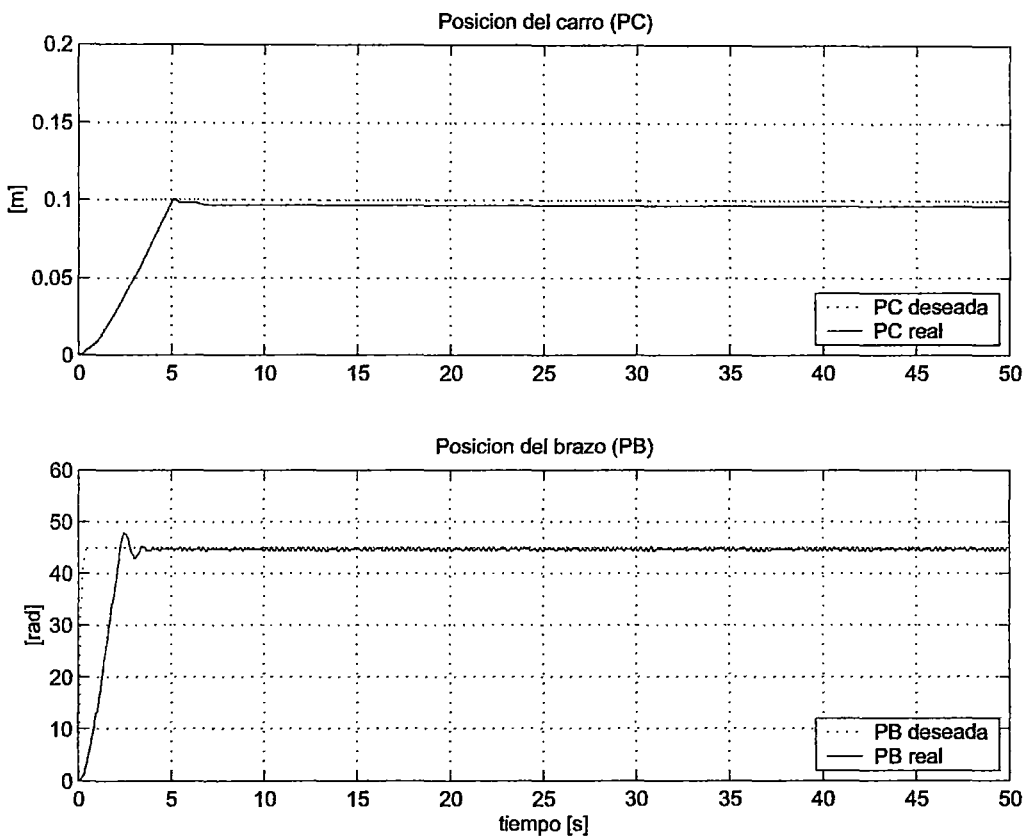


Figura 7.1: Control de posición para el carro y el brazo.

La figura 7.2.1 muestra la respuesta del sistema para la velocidad del carro y velocidad angular del brazo, en ambos casos la velocidad decae al valor cero en un tiempo breve de 4.5 seg y 2.5 seg respectivamente. La oscilación que esta presente a medida que el sistema se controla se debe a que los motores están ejerciendo fuerza de control a medida que el sistema se estabiliza hasta encontrar la posición deseada.

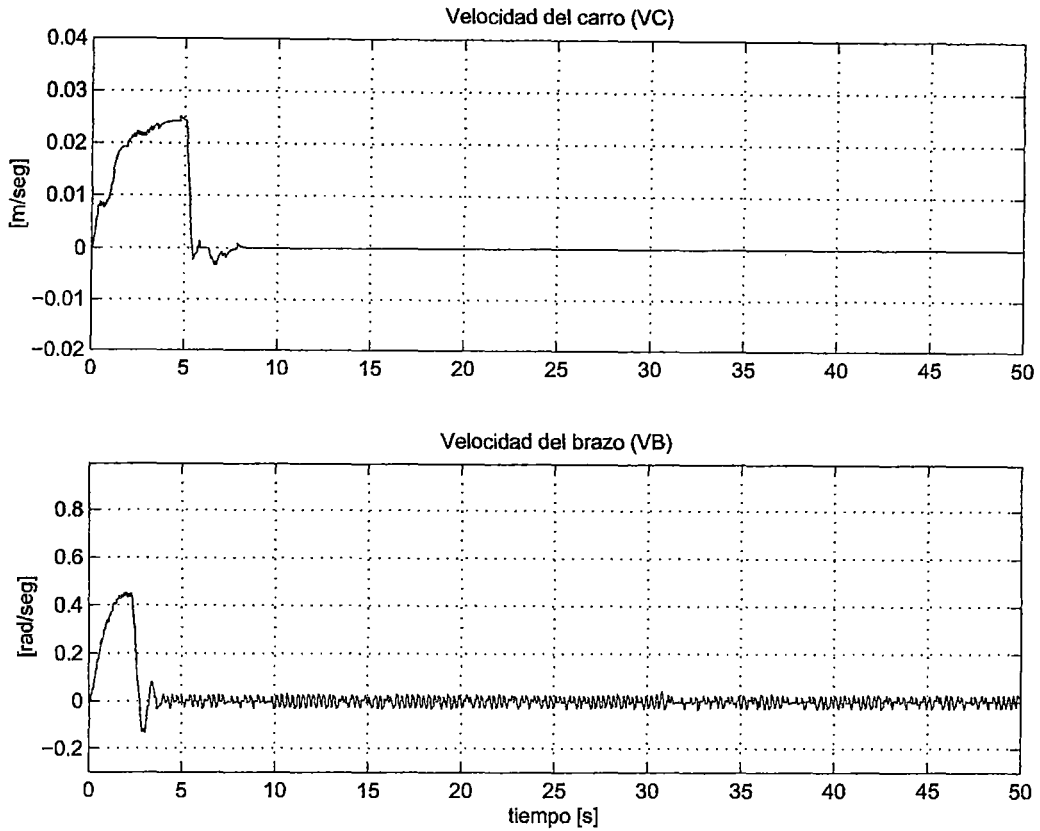


Figura 7.2: Curva de la velocidad para en carro y velocidad del brazo.

## 7.2.2. Experimento2- Sistema con control fuzzy

### II.-Sistema de control fuzzy para el sistema BRT2DOF

Las trayectorias constantes deseadas para el carro y para el brazo son  $x_{d1} = 0,1m$  y  $x_{d2} = \pi/4rad$  respectivamente. El tiempo utilizado en la prueba es de 8 seg. La figura 7.3 muestra las salidas controladas para la posición del carro y posición angular, y se muestran también las velocidades del carro y el brazo luego de un tiempo 8 seg respectivamente Observando las respuestas de ambos sistemas se puede concluir que el sistema con control fuzzy para la posición del carro es  $t_s = 0.95$  seg , para el sistema de control con ley modo deslizante es  $t_s = 5$  seg. luego podemos concluir que el sistema con control difuso es más rápido

Las respuesta del sistema para la posición angular , el sistema con control fuzzy es mas rápido; el error en estado estacionado del sistema con control modo deslizante es mayor.

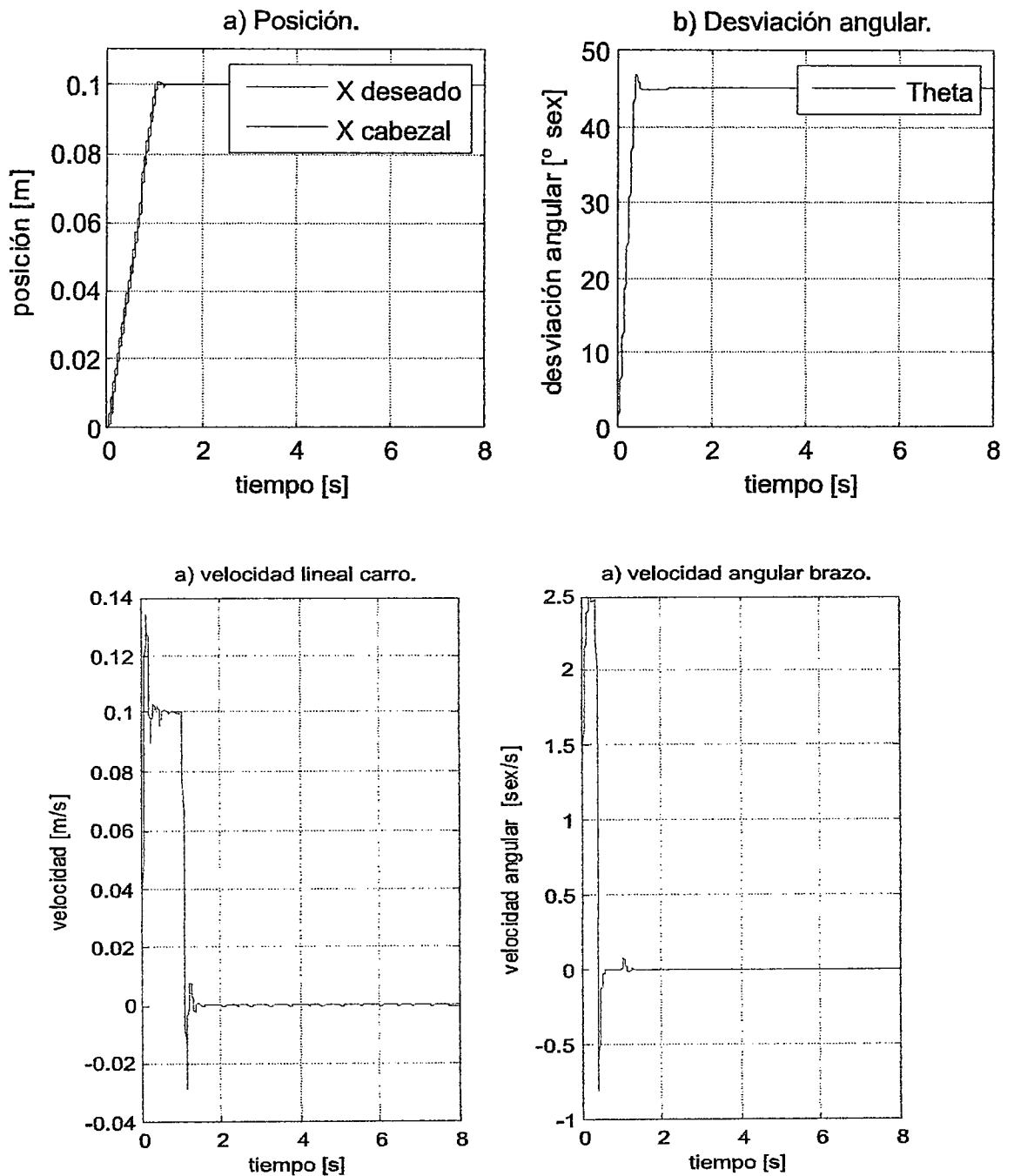


Figura 7.3: Curva de la posición lineal del carro angular del brazo y sus velocidades del BRT2DOF.

Luego se indica que el sistema con control fuzzy es más eficiente por las consideraciones indicadas

### Señales de control y aceleración del carro y del brazo

en la figura 7.4 se muestran las señales de control para la posición lineal y angular, en la misma figura 7.4 se presentan las aceleraciones para el carro y brazo del sistema, considerando las condiciones  $X_{des}=0.1m$  y la Posición deseada  $\theta=\pi/4$  con condiciones nulas

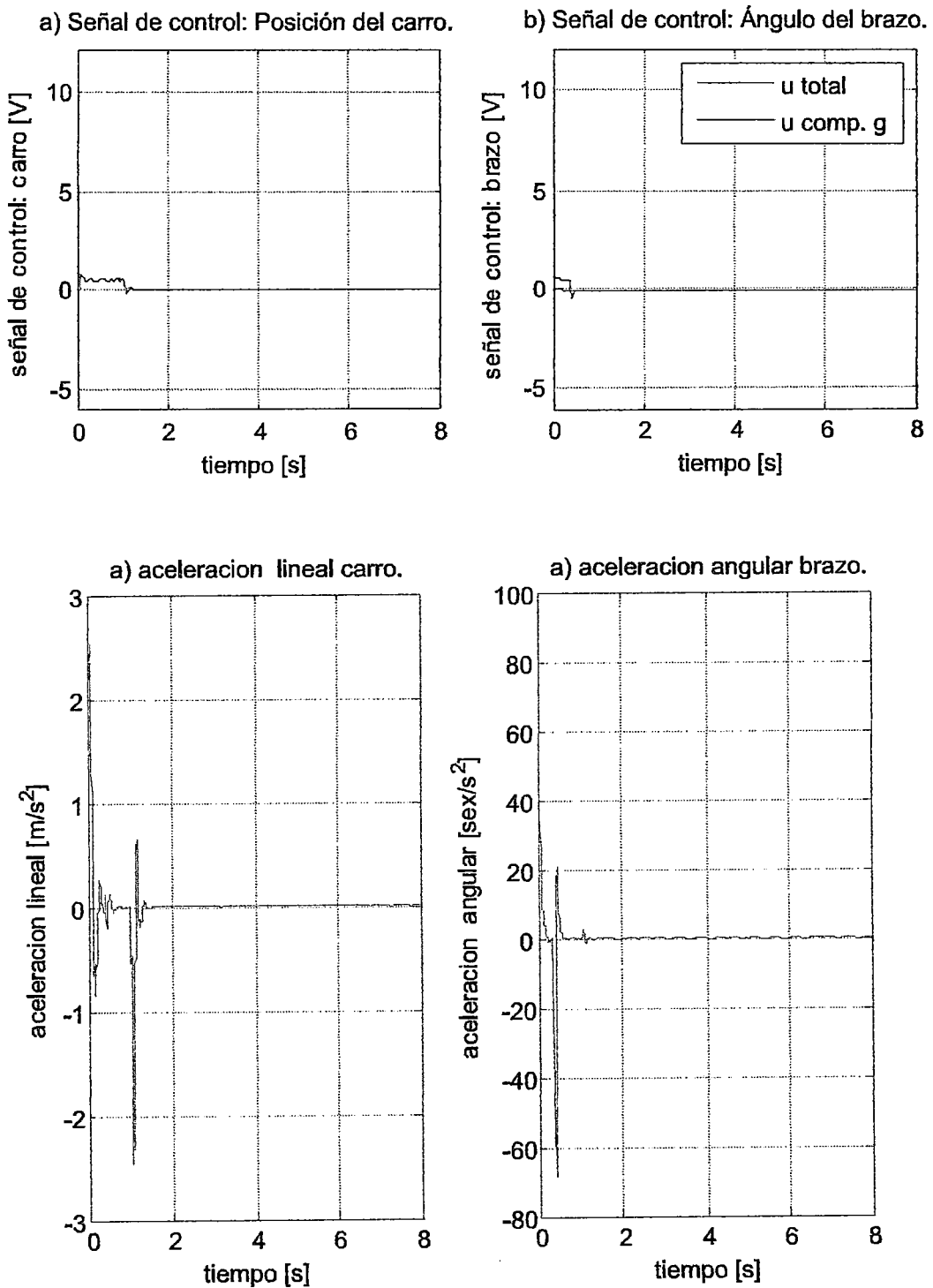


Figura 7.4: Curva de la señales de control lineal del carro, angular del brazo y sus aceleraciones del BRT2DOF.

### 7.2.3. Experimento3-Sistema con control fuzzy con condiciones iniciales diferente de cero

Señales de posición lineal del carro, angular del brazo y las señales de las velocidades de posición lineal y angular

Las condiciones iniciales para realizar el experimento es el siguiente:

- Posición lineal :  $X_1 = 0$  mts
- posición angular:  $X_2 = 2\pi$

Las trayectorias constantes para las posiciones deseadas son:

- Posición lineal deseada :  $X_{d1} = 0.1$  mts
- posición angular:  $X_{d2} = \pi/4$
- Tiempo de simulación considerado:  $simtime = 8$  seg

Analizando los gráficos de la figura 7.5 se determina que:

1. La posición lineal del carro se establece en 0,101 metros en un tiempo  $t_{ss} = 1.01$  seg teniendo un error de 0.001 m.
2. La posición angular del Brazo se establece en  $45,14^\circ$ . en un tiempo  $t_{ss} = 1,21$  seg. Estableciéndose un error de  $0,14^\circ$ .
3. La velocidad del carro inicialmente es alto con un sobrepico hasta 0.1339 m/s ,luego se mantiene constante en  $v = 0.1001$  m/s en un tiempo de 0.7 seg, después se hace  $v = 0.0004755$  m/s para un tiempo  $t_{ss} = 1.425$  seg
4. La velocidad angular del brazo durante el tiempo transitorio inicialmente es negativo hasta obtener un sobreimpulso de  $-2.466$  sex/seg durante un tiempo de 0.165 seg ,luego la velocidad angular cae a 0.0046 sex/seg, en un tiempo de  $t_{ss} = 1.28$  seg

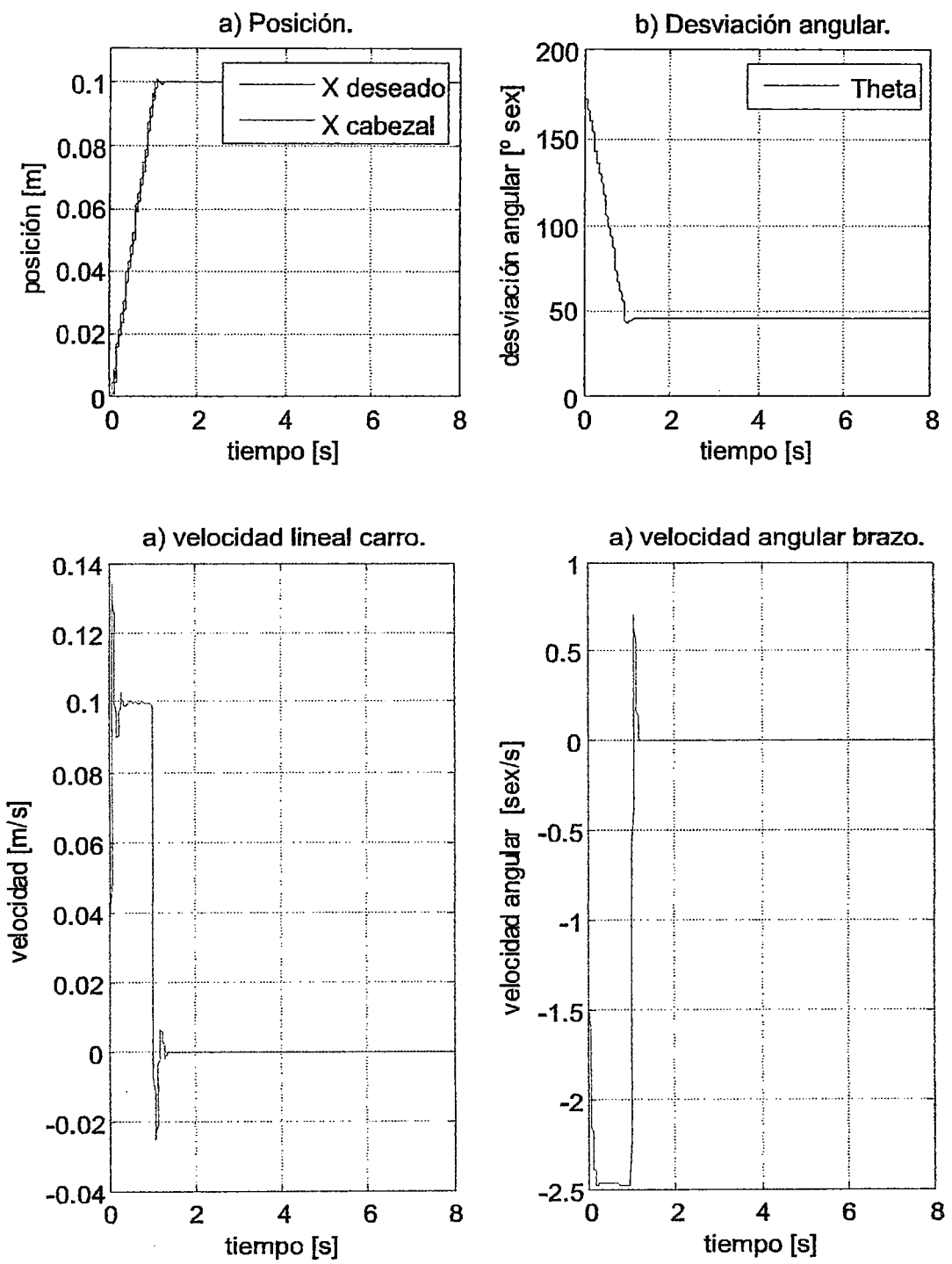


Figura 7.5: Curva de la señales de posición lineal del carro , angular del brazo y velocidades del BRT2DOF.

## Capítulo 8

# Conclusiones y Recomendaciones

### 8.1. Conclusiones

Concluimos que en esta tesis se presentan los conceptos fundamentales de la lógica difusa, luego se realiza el modelo matemático del sistema a controlar, se realiza las simulaciones en lazo abierto y lazo cerrado para tiempo continuo y luego para discreto considerando diferentes entradas de referencia tipo escalón,

Las conclusiones de la tesis son:

1. De los resultados de simulación se puede observar que el controlador lógico difuso diseñado, presenta un buen desempeño, como se puede apreciar en la figura 4.1 La repuesta del sistema de lazo cerrado tiempo continuo , a los cambios de referencia, son rápidos; así mismo para el sistema de control difuso discreto ,tal como se observa en al figura 4.4 la respuesta completa no presenta error en estado estable.
2. El controlador por lógica difusa diseñado resuelve el problema de control propuesto, obteniendo para todas las simulaciones, mostradas en el capítulo 4, desempeños satisfactorios con sobreimpulsos nulos , tiempos de establecimiento aproximados de 1 seg: y errores en estado estable muy cercanos a cero.
3. Como pruebas complementarias de simulación del controlador difuso se aplicaron otras señales de prueba como son las señales senoidal, las cuales realizan un buen seguimiento de la señal de referencia, aunque se presenta un ligero error en estado estacionario.
4. Incorpora lenguaje común al diseño
5. No requiere un modelo riguroso de la planta.
6. Es fácil realizar modificaciones al controlador.
7. Puede llegar a ser lento debido al proceso de muestreo.



8. Depende de la aplicación.

## RECOMENDACIONES

Se recomienda para un trabajo futuro la implementación del sistema BRTM considerando la figura 7.4.

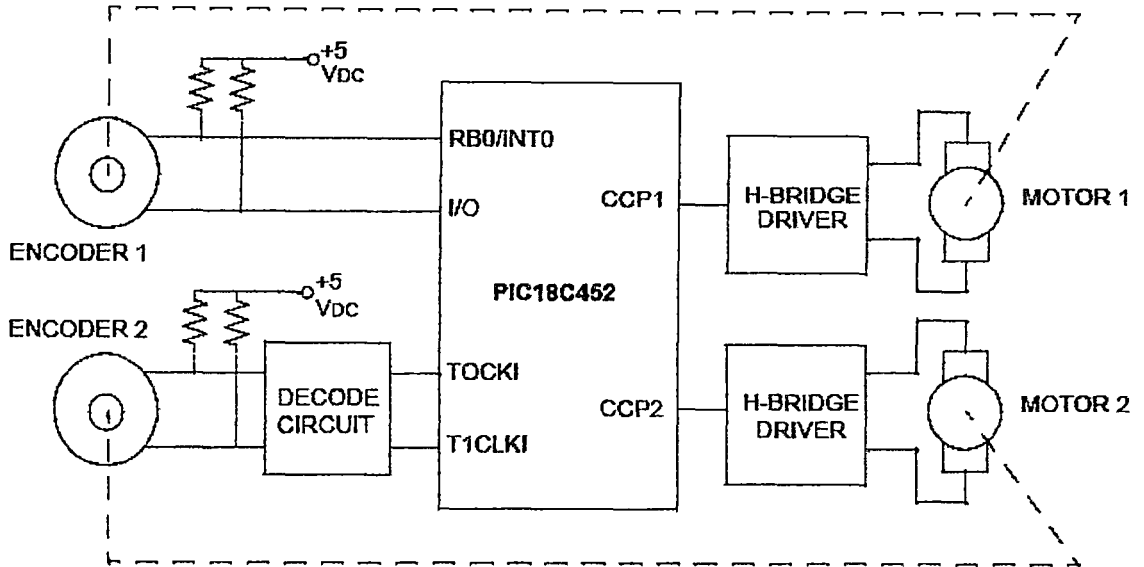


Figura 8.1: Diagramas de bloques de la implementación del sistema.

Donde los componentes físicos está constituido por dos servomotores (modelos iguales) con encoder óptico incremental de 1250 cuentas por vuelta con salida a dos canales en cuadratura.

Los circuitos actuadores se implementa con dos tarjetas similares basados en el drive L6302, este driver integra un puente H completo realizado en Multipower-BCD, tecnología que combina transistores de potencia aislados DMOS con CMOS y Bipolar circuitos en el mismo chip. Mediante el uso de tecnología mixta que ha sido posible optimizar los circuitos de la lógica y la etapa de potencia a lograr el mejor rendimiento posible. los transistores DMOS de salida puede operar con voltajes de hasta 42V y 5 amperios, de manera eficiente a altas frecuencias de conmutación de velocidades. Todas las entradas lógicas son TTL, CMOS y MC compatibles. Cada canal (medio puente) de el dispositivo es controlado por una entrada lógica por separado, mientras que los controles comunes permite a ambos canales. ver figura 7.5.

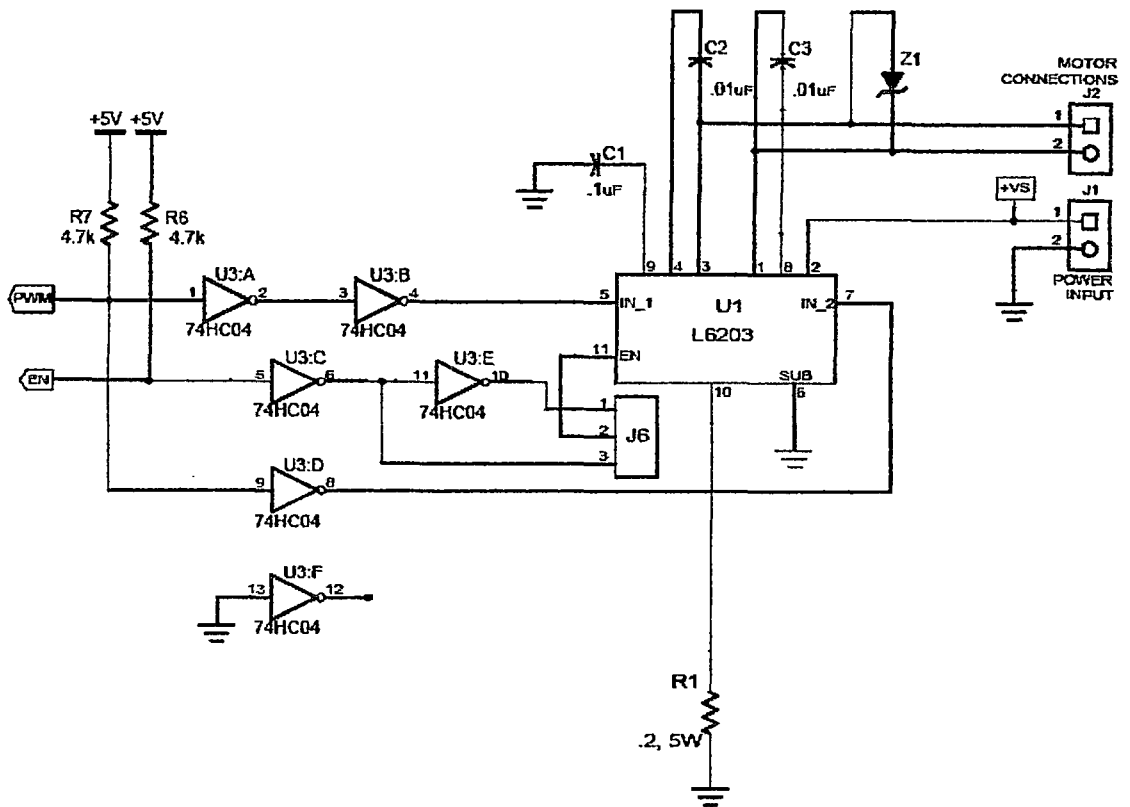


Figura 8.2: CIRCUITO DEL ELEMENTO ACTUADOR.

En particular, este sistema BRTM se recomienda implementarlo tanto en el PIC18F4550 y PIC16F877 por el costo y también porque disponen de los recursos necesarios MCU para realizar la aplicación del servomotor. Con una comprensión de las funciones de servomotor, se puede comenzar con el diseño que se muestra aquí y poner en práctica basado en el PIC16F4550 que se adapte a sus necesidades. El MCU PICmicro puede realizar las siguientes funciones en el aplicación del BRT2DOF , tales como:

- Interfaz de usuario de control
- Medición de la posición del motor
- Cálculo del perfil de movimiento
- Cálculo de la señal de error y la compensación controlador algoritmo ( control difuso)
- Generación de la señal de accionamiento del motor

# Bibliografía

- [1] Acosta ,Gerardo.. **Taller de Controlador Basado en Lgica Difusa..** *Universidad Nacional del Centro de la Provincia de Buenos Aires. Publicación Independiente, Noviembre 2001.*
- [2] *Angulo Usategui, Jose María, Aritza Etxebarria Ruiz, Ignacio Angulo Martinez y Ivaán Trueba Parra.* **dsPIC Diseño Práctico de Aplicaciones.** McGRAW-HILL/INTERAMERICANA S.A.U. España -Madrid, 2006.
- [3] Benajamin C. Kuo. **Sistemas de Control Digital.** *Prentice Hall Hispanoamericana S.A. México, New York ,1997.*
- [4] *Bonifacio Martin del Rio y Alfredo Sanz Molina.* **Redes Neuronales y Sistemas Difusos. 2a. edicion ampliada y actualizada.** AlfaOmega grupo editor S.A. México, 2002.
- [5] Garcia Breijo, Eduardo . **Compilador C CCs y Simulador PROTEUS para Microcontroladores PIC.** , *AlfaOmega grupo editor S.A.de C.V. México, 2008.*
- [6] *Lofti A. Zadeh.* **Fuzzy sets Department of electrical Engineering research laboratory.** University of California, Berkeley California, 1965.
- [7] Lofti A. Zadeh. **Fuzzy sets.** *Information and Control, Vol 8, No.3 pp.338-353, june 1965.*
- [8] *Ogata, Katsuhiko.* **Ingeniería de Control Moderna.** Prentice Hall, Hispanoamericana S.A. México, New York ,1990.
- [9] Ogata, Katsuhiko. **Sistemas de Control en Tiemp Discreto.** *Compañía Editorial Continental S.A. México ,1999.*
- [10] *Ollero Bayurone, Aníbal.* **Control Por computadora Descripcion interna y diseño óptimo.** Alfaomega S.A. de C.V. S.A. México ,1991.

- [11] Passino M.K., Yurkovich S. **Fuzzy Control**,. *Addison-Wesley, USA,1998.*
- [12] Reynoso Oscar ,José María Sebastian , Zuñiga y Rafael Aracil Santoja y Fernando Torres Medina . **Control De Sistemas Discretos**. McGRAW-HILL/INTERAMERICANA S.A.U. España -Madrid, 2004.
- [13] Rojas Moreno, Arturo. **Control Avanzado-Diseño y Aplicaciones en Tiempo Real**. *Universidad Nacional de Ingeniería ,Publicación Independiente,Lima- Perú, 2001.*
- [14] M. Takagi, T. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, pages 15:116132, 1985.
- [15] S. Mamdani, E.H. Assilian. **An experiment in linguistic synthesis with a fuzzy logic controller**. *International Journal of Man-Machine Studies*, pages 7(1):113, 1975.
- [16] Mohammad Jamshidi,. **Fuzzy logic and control Software and hardware applications**,. Ed. Prentice Hall, Estados Unidos,1994.
- [17] Ying, Hao. **Fuzzy Control and Modeling. Analytical Foundations and Applications**.. *IEEE Press. 2000*

## Apéndice A

# Identificación de Parámetros

Por identificación se entiende la formulación de un modelo que permita representar al sistema con todas sus propiedades y características relevantes. En nuestro caso determinaremos los parámetros de los servomotores para el sistema BRTM de 2DOF, estos parámetros hallados, nos permitirán validar los resultados experimentales. En la sección desarrollaremos el modelo matemático del sistema para realizar la identificación, basándonos en el análisis del servomotor DC expuesto en la sección anterior. Un modelo matemático es un conjunto de ecuaciones que envuelven las variables del sistema que proporcionan una representación óptima al sistema. En general, si realizamos un experimento en que aplicamos ciertos valores a las entradas del sistema, obtendremos ciertas respuestas en sus salidas; de ese modo dando los mismos valores que en el experimento a las variables independientes de nuestro modelo, podemos esperar que después de resolver el conjunto de ecuaciones la respuesta del modelo deberá tener respuestas similares a aquellas salidas del sistema físico.

### A.1. Identificación de Parámetros del Servomotor

Para identificar los parámetros de un servomotor DC (sin carga) es necesario tener la edición de un programa en C para la generación de un archivo de datos de la velocidad del servomotor en lazo abierto. Los archivos usados para la experiencia en tiempo real son:

- *adqs.cpp*, contiene el programa principal para adquisición de la data de la velocidad. Este programa es desarrollado en lenguaje C.
- *adq\_d1.cpp*, contiene el programa de la configuración de la tarjeta LABPC+ que trabaja con la librería *Nidaqbc.lib*.
- *adq\_d1.h*, contiene las funciones para el manejo de la tarjeta LABPC+.
- *Nidaqbc.lib* (contiene las librerías de funciones para el manejo de la tarjeta LABPC+).

La creación del proyecto se realiza por medio del archivo *adqs.prj*, incluye los archivos: {*adqs.cpp*, *adq1.cpp* y *Nidaqbc.lib*}.

La ejecución del archivo *prj* (ejecución en tiempo real), generan el archivo de datos *adqs24.dat* para el servomotor. El archivo de datos es utilizado para obtener la curva de velocidad que se muestra en la figura A.1. El uso de un método de identificación para determinar los parámetros del servomotor nos conduce a hallar la resistencia de armadura  $R_a$ , constante de torque de servomotor  $K_m$ , constante de fuerza contraelectromotriz  $K_b$ , fricción viscosa  $B$  y momento de inercia  $J$ .

## A.2. Curva de la Velocidad para la Identificación

Usaremos el método gráfico de aproximación de la respuesta del proceso a un sistema de primer orden, cuando es despreciada la inductancia de armadura  $L_a$ . Los servomotores que utilizaremos son de fabricación Matsushita modelo GMX 6MPO13A con encoder incremental, disponemos de la información (ver tabla A.1) del servomotor Matsushita con encoder incremental. Para obtener los datos necesarios como son la amplitud  $K_{med}$  y a constante de tiempo  $T_{med}$  de la curva de velocidad, trazamos una recta tangente en el punto de inflexión, tal como se muestra en la figura A.1, posteriormente trazamos las perpendiculares hacia los ejes coordenados que cortaran a estos en dos puntos, cuyos valores son:  $K_{med} = 9,1$  y  $T_{med} = 0,35$ .

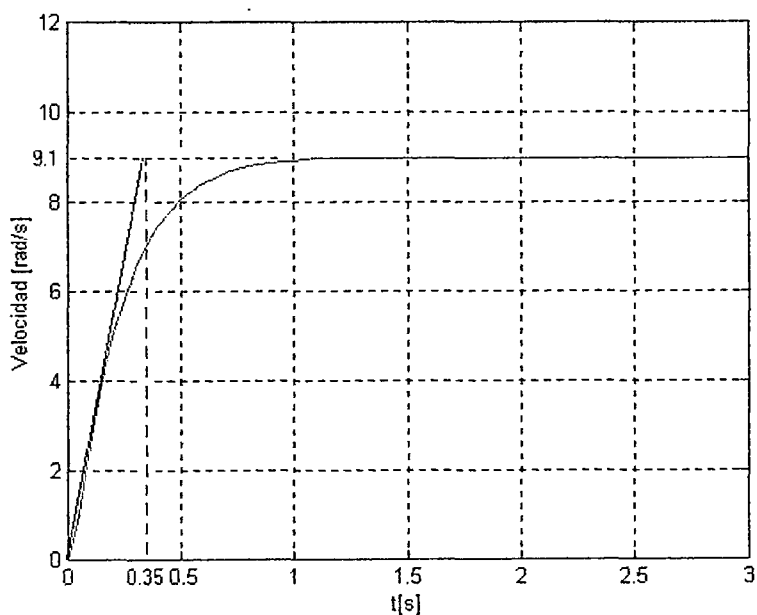


Figura A.1: Curva S.

Disponemos de la información del servomotor Matsushita Modelo GMX-6MPO13A con encoder incremental; Alimentación 24VDC/0.2A; Velocidad sin carga 310RPM; Reducción 12.5:1. La tabla A.1 muestra los valores que corresponden a las pruebas de laboratorio a 24VDC.

Tabla A.1: Especificaciones del servomotor DC.

Velocidad ( $W$ ) [rad/seg]	Torque ( $nT_m$ ) [Nm]	Corriente [A]
32.46	N.L.	0.2
29.84	0.2118	0.5
27.22	0.5507	1.0
24.39	0.8826	1.5
21.46	1.2710	2.0
19.37	1.5534	2.5
15.70	1.8712	3.0
10.47	2.0336	3.5
6.28	2.1466	4.0
STALL	2.3725	4.5

Siguiendo el mismo análisis realizado para los subsistemas de los modelos de los servomotores  $M_1$  y  $M_2$ , y tratándose de servomotores de la mismo modelo de fabricación, se realizó la experiencia en tiempo real para el servomotor  $M_2$  que está acoplado al carro, es decir, los parámetros para el servomotor  $M_2$  serán los mismos para el servomotor  $M_1$ .

Usualmente un modelo lineal de segundo orden para un servomotor DC se expresa mediante la función de transferencia de la velocidad de un servomotor DC dada por.

$$\frac{W(s)}{V_a(s)} = \frac{K_m}{J_{eq}L_a s^2 + (L_a B_{eq} + R_a J_{eq})s + (R_a B_{eq} + K_m K_b)} \quad (A.1)$$

Si omitimos la inductancia, es decir  $L_a = 0$ , entonces (A.1) se convierte es una función de transferencia de primer orden de la forma.

$$\frac{W(s)}{V_a(s)} = \frac{K_m}{R_a J_{eq} s + (R_a B_{eq} + K_m K_b)} = \frac{K_{med}}{T_{med} s + 1} \quad (A.2)$$

donde.

$$K_{med} = \frac{K_m}{R_a B_{eq} + K_m K_b} \quad (A.3)$$

y

$$T_{med} = \frac{R_a J_{eq}}{R_a B_{eq} + K_m K_b} \quad (A.4)$$

La resistencia  $R_a$  se obtiene de la siguiente relación.

$$R_a = \frac{V}{I_{aSTALL}} = \frac{24V}{4,5A} = 5,3\Omega \quad (A.5)$$

La constante contraelectromotriz  $K_b$  se obtienen de acuerdo con la tabla A.1 y considerando la inductancia  $L_a = 0$ .

$$K_b = \frac{V_a - R_a I_a}{nW} = \frac{24V - 5,3\Omega \times 0,2A}{12,5 \times 32,46 rad/s} = 0,0565V/rad/s \quad (A.6)$$

La constante de torque  $K_m$  se obtienen de acuerdo con la tabla A.1;

$$nT_{m|STALL} = 2,3725Nm$$

El factor de reducción es 12,5, entonces.

$$T_{m|STALL} = 0,1898Nm$$

La constante del motor se determina a partir de la relación.

$$K_m = \frac{T_{mSTALL}}{I_{aSTALL}} = \frac{0,1898Nm}{4,5A} = 0,0421Nm/A \quad (A.7)$$

Reemplazamos los valores  $K_{med} = 9,1$  y  $T_{med} = 0,35$  que fuerón obtenidos a partir de las pruebas de laboratorio para la velocidad en lazo abierto, utilizando la tarjeta de adquisición de datos LABPC+, usando los resultados de  $R_a$  y  $K_m$  de las ecuaciones (A.5) y (A.7) y dividiendo (A.4) de (A.3) obtenemos la expresión para  $J_{eq}$ .

$$J_{eq} = \frac{T_{med} K_m}{K_{med} R_a} = \frac{0,35 \times 0,0421}{9,1 \times 5,3} = 3,0551 \times 10^{-4} Kgm^2 \quad (A.8)$$

La constante de fricción viscosa obtenemos reemplazando los valores de  $K_{med}$ ,  $K_m$  y  $R_a$  en la ecuación (A.3), luego despejamos  $B_{eq}$ .

$$B_{eq} = \frac{K_m/K_{med} - K_m^2}{R_a} = \frac{0,0421/9,1 - 0,0421^2}{5,3} = 5,3848 \times 10^{-4} Nm/rad/s \quad (A.9)$$



Finalmente se puede determinar la ganancia del amplificador  $K_A$ , por medición directa de la tensión en los terminales de armadura del servomotor (salida del amplificador), y considerando la tensión de entrada  $u$  de 1.4 voltios.

$$K_A = \frac{V_{aMEDIDO}}{u} = \frac{9,1V}{1,4V} = 5,5 \quad (A.10)$$

### A.3. Coeficiente de Fricción del Carro

El concepto de fricción estática puede entenderse mejor considerando el diagrama mostrado en figura A.2. La figura A.2 representa el carro de peso  $W$  que descansan en un plano horizontal (el peso  $W$  considera los pesos del carro, servomotor y brazo). Una fuerza  $F_1$  que tiende a producir el movimiento hacia la derecha. El movimiento se opone a una fuerza  $F$  conocida como fuerza de fricción del carro.  $F$  es siempre paralela al plano y opuesta al movimiento. El peso es soportado por la fuerza normal  $N$  (el peso de  $N$  es siempre perpendicular a la superficie de contacto). Todas las fuerzas son mostradas mediante vectores simples y el objetivo es el equilibrio.

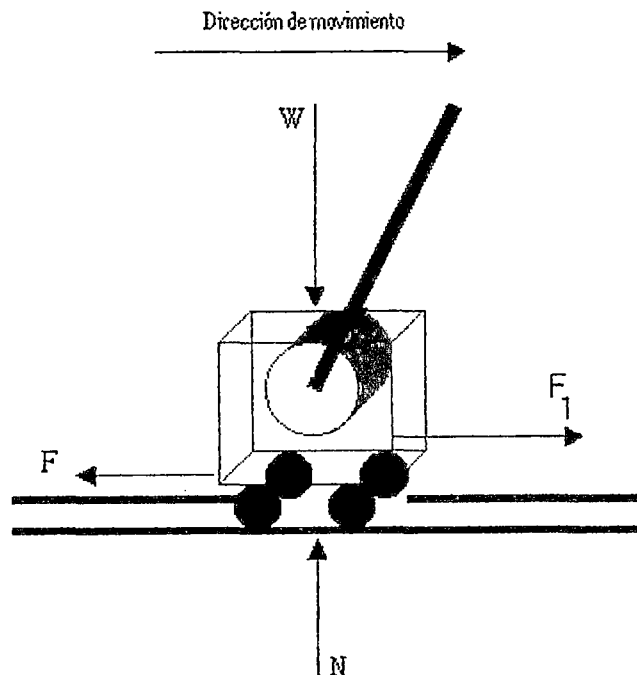


Figura A.2: Dirección de movimiento.

$F$  es una fuerza pasiva, es decir, aumenta  $F_1$  hasta un máximo  $F_m$  conocido como la fuerza friccional máxima.  $F_m$  es determinado por el peso del objeto y la

naturaleza de las superficies en el contacto. Si  $F_1$  es mayor que  $F_m$ , el cuerpo se moverá.  $F_m$  es igual a la fuerza mínima, mientras se omiten las fuerzas inerciales, necesarias para que el objeto empiece a moverse.

## Fricción Deslizante

La fricción deslizante es la fuerza requerida para mantener un movimiento relativo entre dos cuerpos. La figura A.2 puede ser usada para representar el bloque deslizante con una velocidad constante en el plano horizontal. A velocidades bajas, la fricción es independiente de la velocidad deslizante. A velocidades más altas, la fricción disminuye como el aumento de velocidad. La fricción deslizante normalmente es menor que la fricción estática; por consiguiente, los coeficientes de fricción deslizante son normalmente más pequeños que los coeficientes de fricción estática.

## Valor Experimental del Coeficiente de Fricción

El coeficiente de fricción es determinado experimentalmente y puede variar dependiendo de las superficies de contacto y las condiciones particulares bajo la cual se han medido. La tabla A.2 es considerada como una guía general [?].

Sean las ecuaciones de movimiento independientes.

$$\begin{aligned} F_1 &= F_m \\ F_m &= fN \end{aligned} \tag{A.11}$$

la ecuación en el equilibrio.

$$\sum F_y = 0 = N - W \tag{A.12}$$

la ecuación de la fuerza friccional máxima queda expresada en función de la constante de fricción  $F_c$  y la velocidad  $v$ .

$$vF_c = fN \tag{A.13}$$

luego, la constante de fricción queda expresada.

$$F_c = \frac{fN}{v} \tag{A.14}$$

Los valores de  $W$  y  $v$  se asignan de acuerdo a las mediciones realizadas experimentalmente, tal es el caso del peso  $W = M_c + m = 1,7011Kg$  (ver tabla 2.2) y para calcular la velocidad, tenemos la longitud que recorre el carro de  $0,9m$  (longitud máxima de la riel) que recorre en un tiempo de  $2,8seg$ , de este modo tenemos la velocidad promedio de  $0,32m/seg$ . Para el coeficiente de fricción, se considera el caso de material metal sobre metal con una condición seca y para un tipo de fricción deslizando, el valor asignado es  $f = 0,054$  (ver tabla A.2). Con estos datos reemplazamos en (A.14) para obtener el coeficiente de fricción del carro  $F_c$ .

$$F_c = \frac{0,054(1,7011)(9,8)}{0,32} = 2,81Kg/seg$$

Tabla A.2: Coeficiente de fricción  $f$ .

Material	Condición	Estática	Deslizante
Madera sobre madera	Seco	0.30-0.60	0.20-0.50
Madera sobre madera	Mojado	0.40-0.70	0.20-0.40
Madera sobre metal	Seco	0.20-0.60	0.30-0.50
Madera sobre metal	Mojado	0.50-0.70	0.20-0.40
Metal sobre metal	Seco	0.20-1.10	0.03-0.50
Metal sobre metal	Engrasado	0.10-0.50	0.03-0.30
Cuero sobre metal	Seco	0.30-0.50	0.30-0.50
Piedra sobre concreto	Seco	0.60-1.00	
Acero sobre hielo	Seco	0.02-0.03	0.01-0.02

## A.4. Coeficiente de Fricción Viscosa del Brazo

La obtención del coeficiente de fricción viscosa del brazo, tiene que ver con el modelamiento y compensación de la fricción de Coulomb. La fricción de Coulomb es el torque requerido para mantener al servomotor en movimiento y es una fricción que se opone al movimiento. Se ha comprobado experimentalmente que la fricción de Coulomb puede ser considerada constante. El efecto de la fricción de Coulomb es modelado como un torque constante proporcional al signo de la velocidad, es decir.

$$T_{fc} = \begin{cases} -T_c & \dot{q} < 0 \\ T_c & \dot{q} > 0 \end{cases} \quad (A.15)$$

donde  $T_{fc}$  es el torque de fricción de Coulomb,  $\dot{q}$  es la velocidad del eje del servomotor.

El modelo discreto para el servomotor DC ( $L = 0$ ) considerando la fricción de Coulomb se deducen a partir de las siguientes ecuaciones.

$$T_m = J_{eq}\dot{w} + B_{eq}w + f^*(nw) \quad (\text{A.16})$$

$$T_m = K_m I_a \quad (\text{A.17})$$

$$I_a = \frac{V_b K_A}{R_a} - \frac{n K_b w}{R_a} \quad (\text{A.18})$$

reemplazamos (A.18) en (A.17), luego igualamos con (A.16).

$$J_{eq}\dot{w} = \frac{V_b K_A K_m}{R_a} - (B_{eq} + \frac{n K_b w}{R_a}) - f^*(nw) \quad (\text{A.19})$$

Las señales en tiempo discreto surgen si el sistema involucra la operación de muestreo de señales continuas. sumimos a  $w_k$  como la velocidad angular muestreada en el  $k$ -ésimo instante de tiempo.

Usamos la aproximación de Euler  $T_s \dot{w} = w_{k+1} - w_k$  para discretizar la ecuación (A.19), luego obtenemos.

$$w_{k+1} = [1 - \frac{T_s}{J_{eq}} (B_{eq} + \frac{n K_m K_b}{R_a})] w_k + \frac{T_s K_m K_A}{J_{eq} R_a} [V_{bk} - \frac{R f^*(nw_k)}{K_m K_A}] \quad (\text{A.20})$$

$f^*(nw_k)$  modela el torque de fricción de Coulomb; podemos indicar que la aproximación es dada por  $f^*(nw_k) = V_{zm} \text{sgn}(nw)$  ( $V_{zm}$  es la mitad del ancho de la zona muerta que será determinada más adelante). Si la velocidad es distinta de cero, el torque tomará dos valores constantes dependiendo del sentido de giro del servomotor.

La ecuación (A.20) podemos expresar como.

$$w_{k+1} = [1 - \frac{T_s}{J_{eq}} (B_{eq} + \frac{n K_m K_b}{R_a})] w_k + \frac{T_s K_m K_A}{J_{eq} R_a} [V_{bk} - \frac{RC \text{sgn}(nw_k)}{K_m K_A}] \quad (\text{A.21})$$

El diagrama de bloques de la figura A.3 muestra al modelo del servomotor discreto. El diagrama involucra la compensación de la fricción de Coulomb determinada por el parámetro  $G_3$ , que será objeto de análisis para determinar el coeficiente de fricción viscosa  $C$ .

Los parámetros para el sistema sin compensar se describen en la tabla A.3, donde  $G_i$ ,  $i = 1, 2, 3, 4$ .

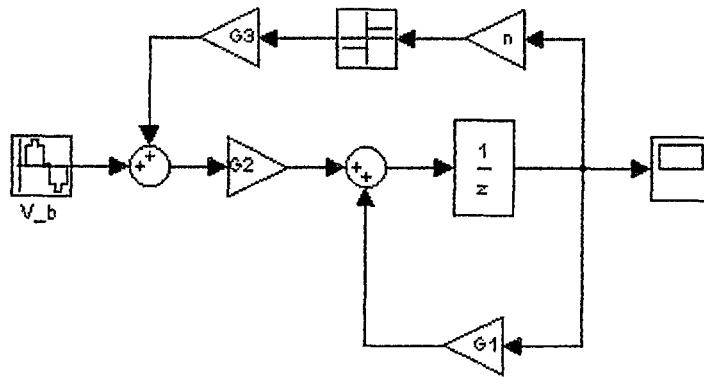


Figura A.3: Modelo discreto del servomotor DC.

Tabla A.3: Parámetros del sistema a simular.

Parámetro	Descripción	Valor
$G_1$	$1 - (T_s/J_{eq})(B_{eq} + nK_mK_b/R_a)$	0.9799
$G_2$	$K_mK_A T_s/(R_a J_{eq})$	0.1430
$G_3$	$-C R_a/(K_m K_A)$	-0.2174
$G_4$	$n$	12.5
$T_s$		0.001

Las simulaciones han sido desarrolladas en SIMULINK, cuya entrada  $V_b$  es una entrada senoidal de amplitud 0,5V y frecuencia de 3rad/seg. El resultado de la simulación para la velocidad angular  $w$  es mostrada en la figura A.4

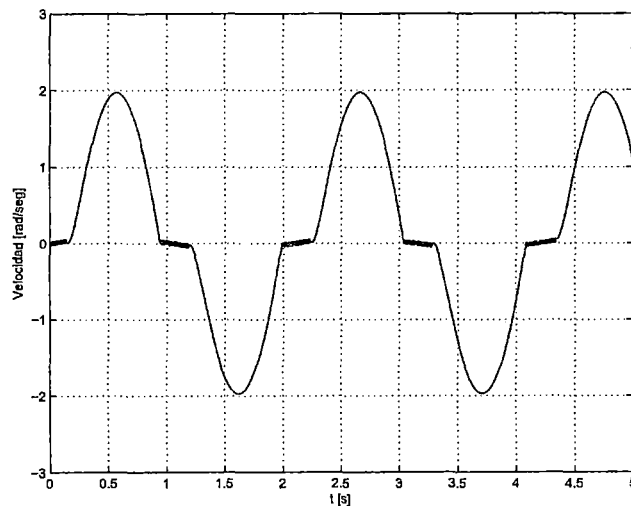


Figura A.4: Simulación del modelo simplificado  $L = 0$ .

La gráfica de la velocidad angular en estado estacionario del servomotor DC versus el voltaje aplicado al amplificador de potencia, se muestra en la figura A.5, nos permite apreciar una zona muerta a la velocidad cero. Esta zona muerta es debida a la fricción estática del servomotor. Note que el efecto no lineal en las cercanías de la zona muerta.

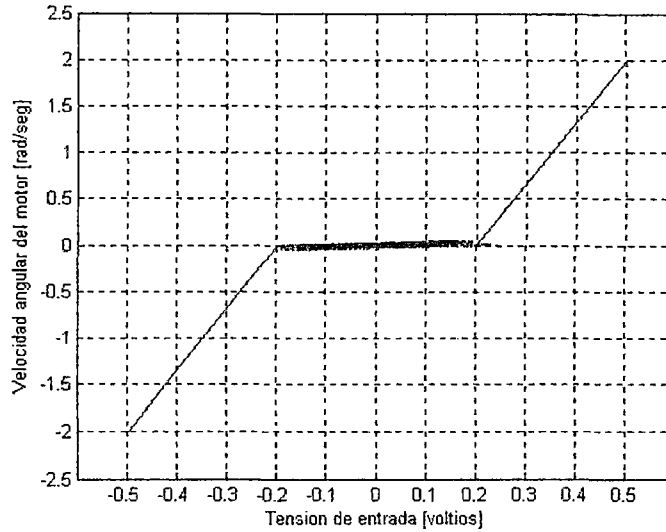


Figura A.5: Tensión de salida versus tensión de entrada.

Para linealizar la ecuación (A.19) se debe de adicionar una tensión de entrada a la entrada del amplificador tal que compense la fricción de Coulomb cuando la velocidad del servomotor  $w_M \geq w_{th}$  ( $w_{th}$  velocidad umbral). Un esquema general es el que presenta la compensación de las fricciones estática y de Coulomb.

$$u = v + V_c \quad (\text{A.22})$$

siendo  $u$  la ley de control,  $v$  el voltaje de compensación de la fricción estática y  $V_c$  el voltaje de compensación de Coulomb. El objetivo del primer término (dependiente de la dirección del voltaje de armadura) es sacar al servomotor de la zona donde la fricción estática tiene efecto. El objetivo del segundo término es compensar el torque de fricción de Coulomb presente cuando el servomotor está en movimiento a velocidades mayores al umbral  $w_{th}$ .

El voltaje de compensación de Coulomb es dado por la relación.

$$V_c = \frac{R_a C}{K_m K_A} \text{sgn}(V_b) \quad \text{si } w_M \geq w_{th} \quad (\text{A.23})$$

El coeficiente de fricción viscosa  $C$  es pequeño para servomotores DC, siendo difícil su identificación en presencia de ruido. Usamos el método de prueba y error para el cálculo de  $V_c$ , según la figura A.5, el valor inicial de  $V_{zm}$  es 0.2, este valor fue disminuido hasta encontrar el valor apropiado de  $V_{zm} = 0,0217$ .

Despejamos el coeficiente de fricción  $C$  de la ecuación (A.23) para reemplazar los parámetros previamente identificados, como son:  $R_a = 5,3\Omega$ ,  $K_A = 5,5$  y  $K_m = 0,0421Nm/A$ , luego.

$$C = \frac{V_{zm}K_mK_A}{R_a} = \frac{0,0217(0,0421)(5,5)}{5,3} = 9,5e^{-4}Kg\,m^2/seg \quad (A.24)$$

El valor  $V_{zm} = 0,0217$  que conduce a compensar la fricción de Coulomb resultando el coeficiente de fricción de valor  $C = 9,5e^{-4}$ . Los efectos de compensación son mostrados en la figura A.6.

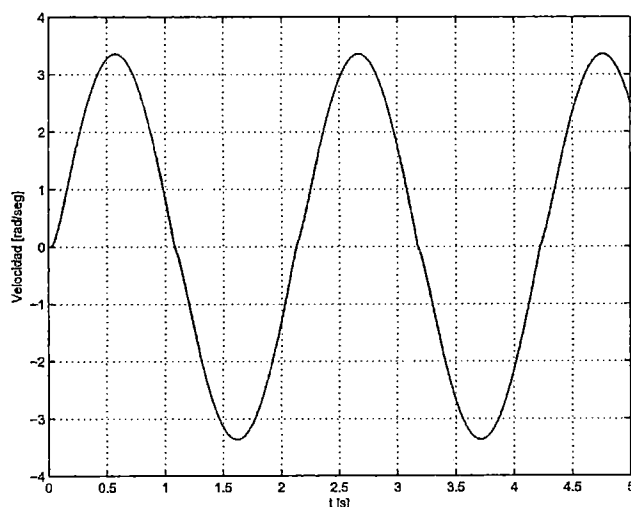


Figura A.6: Experimento de la compensación de la fricción no lineal.

## A.5. Estimación de los parámetros físicos de un motor de corriente directa usando el algoritmo de Steiglitz- Mcbride Considerando la Inductancia $L_a$

El método de Steiglitz-Mcbride es una técnica muy popular para la identificación de sistemas lineales descritos por ecuaciones de diferencias, es esencialmente un procedimiento fuera de línea, y es atractivo particularmente por ser un esquema de iteraciones simples. Esta técnica iterativa consiste en identificar un sistema lineal usando muestras de su entrada y salida en la presencia de ruido, aplicando mínimos cuadrados entre la respuesta del sistema real y la respuesta del sistema estimado. En este apéndice se muestran el resultado de la estimación del motor de corriente directa de los cuales se conocen sus parámetros aplicando el algoritmo de Steiglitz-Mcbride.

### A.5.1. Estimación de los parámetros del motor de corriente directa.

Motor de corriente directa, del cuál los parámetros reales fueron obtenidos mediante un método experimental tal como se expresa en el apéndice A1. El fabricante del motor es Matsushita Modelo GMX-6MPO13A con encoder incremental; Alimentación 24VDC/0.2A y sus parámetros se muestran en la tabla A.4.

Tabla A.4: Parámetros del Motor.

Símbolo	Descripción	Valor	Unidades
$V_a$	Voltaje de armadura	24	$V$
$R_a$	resistencia de armadura	5.3	$\Omega$
$L_a$	Inductancia de armadura	0.0061	$H$
$K$	constante de Par	0.0565	$V/rad/s$
$J$	momento de inercia	$2,38e^{-3}$	$Kgm^2$
$B$	fricción viscosa	$0.3307 \times 10^{-4}$	$Nm/rad/s$
$n$	factor de reducción	12.5	

Para la estimación de los parámetros se usan muestras de las señales de entrada y salida del sistema bajo estudio, en este caso dichas muestras fueron tomadas de la



simulación de los modelos reales en SIMULINK , tal como se muestra en la figura A.7.

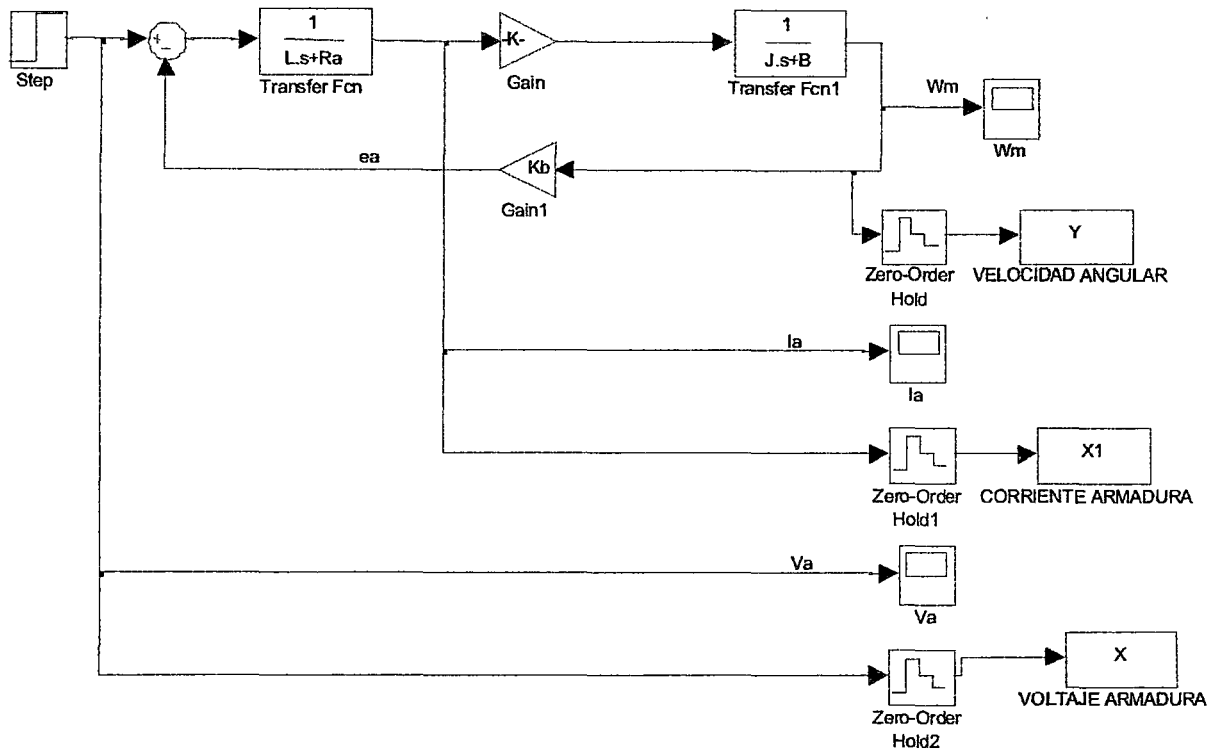


Figura A.7: Obtención de las muestras de señales para la estimación.

Con el arreglo mostrado en la figura A.7. se obtuvieron 3001 muestras con un periodo de muestreo de 0.01 seg. La razón para el tiempo de muestreo es que los fenómenos transitorios para este sistema que ayudan al proceso de estimación suceden un tiempo mucho mayor a 0.01 seg. La señal de prueba usada como voltaje de armadura corresponde a un escalón de 24 V. La razón por la cuál se escoge el escalón es que dicha señal presenta un cambio súbito en el arranque del motor, con este cambio subitito de tensión es posible obtener información dinámica que con otro tipo de señal no se podria. Por ejemplo si se aplicar una señal senoidal de 60 hz, no sera útil porque el motor no podria seguir los cambios de dicha señal y la filtraria, observándose como resultado que el motor no se mueva. En la mayoría de las referencias consultadas para la estimación de parámetros se opta por una señal aleatoria con un alto contenido espectral, y la razón ah fue que el modelo por estimar siempre fue un filtro y estos dependen grandemente de la frecuencia.

Tomando en cuenta lo anteriormente expuesto se muestra en la tabla A.5. la comparación de los parámetros obtenidos contra los reales. Estos valores estimados se obtuvieron aplicando el programa denominado  $SMDC_{F.m}$ .

Tabla A.5: Comparacion de valores reales con los valores estimados.

Parametro	Valor Real	Valor Estimado	Error de variación
$K$	0.0565	0.0612	1.02
$J$	0.00238	0.020	13.89
$B$	0.0033	0.0034	5
$L_a$	0.0061	0.0062	1.58
$R_a$	5.3	5.32	0.36

Los programas de simulación para la estimación del motor DC se listan .

```
%Programa 1. SMIIR.m
%Estimacin del filtro IIR mediante el algoritmo de Steiglitz-McBride
[NUMF,DENF] = STMCB(y,x,3,3);
t=[0.1626 0.0098 0.0366 0.0366];
s=[1 -0.001 0 -0.82];
```

```
Programa 2. SMIIR2.m
%Estimacin del filtro IIR mediante el algoritmo de Steiglitz-McBride
[NUMF,DENF] = STMCB(y,x,4,4)
t=[0.8 -1.46 1.46 -0.72 0.17]
s=[1 -0.36 0.91 0.03 0.16]
```

El programa Principal para realizar la estimacion de los parametros del servomotor se listan en el programa 3 denominado  $SMDC_{F.m}$

```
Programa 3. SMDC_F.m
%Estimacin del los parametros del motor de corriente directa
%usando el algoritmo de Steiglitz-McBride
% Vectores de muestras:
% x(voltaje de armadura)
% y(velocidad angular)
% x1(corriente de armadura)
warning off MATLAB:toeplitz:DiagonalConflict;
[NUMF1,DENF1] = STMCB(y,x1,0,1); %Obtiene la estimacion del modelo w/i
NUMd1=NUMF1; %Define el polinomio discreto del numerador NUMd1
DEND1=DENF1; %Define el polinomio discreto del denominador DEND1
[NUMc1,DENc1] = D2CM(NUMd1,DEND1,0.01,'zoh'); %Obtiene la funcion de
%transferencia en s para w/i
warning off MATLAB:toeplitz:DiagonalConflict;
[NUMF2,DENF2] = STMCB(x1,x,2,2); %Obtiene la estimacion del modelo i/v
NUMd2=NUMF2; %Define el polinomio discreto del numerador NUMd2
DEND2=DENF2; %Define el polinomio discreto del denominador DEND2
%Obtiene la funcion de transferencia en s para i/v
[NUMc2,DENc2] = D2CM(NUMd2,DEND2,0.01,'zoh');
```

```

%Obtiene el tamaño del vector muestras x(voltaje de armadura)
[m,n]=size(x);
%Obtiene el tamaño del vector muestras y(velocidad angular)
[o,p]=size(y);
%Obtiene el tamaño del vector muestras x1(corriente de armadura)
[s,t]=size(x1);
Kte=x(m,n)/y(o,p) %Obtención de Kte
%Define los coeficientes estimados con los coeficientes del modelo en MATLAB
a1e=DENC1(2);
a2e=NUMc1(2);
a3e=NUMc2(3);
a4e=DENC2(2);
a5e=NUMc2(2);
a6e=DENC2(3);
Je=Kte/a2e %Obtención de J(estimado)
Be=a1e*Je %Obtención de b(estimado)
Le=1/a5e %Obtención de Le(estimado)
Re=(a4e-(be/Je))*Le %Obtención de Re(estimado)

```

## Apéndice B

# Programas en MATLAB

## B.1. Listado de Programas

### B.1.1. Simulación de la planta en lazo abierto

Los programas de simulación están escritos en código MATLAB. Se presentan el programa llamado *openlooptestmr2dof.m* que nos permite obtener la respuesta del sistema

El programa *openlooptestmr2dof.m* es utilizado en las simulaciones realizadas en la sección 3. La característica principal del programa es el uso de la función *f\_mrtm2dof.m* para obtener las respuesta de posición del carro y la posición angular de la varilla brazo El código del programa de control se lista a continuación.

```
%-----programa openlooptest_mrtm2dof.m-----
clear all; close all; ts=0.01; simtime=0:ts:10;
nsamples=size(simtime);nsamples=nsamples(1,2);

x(1,:)=[0 0 0 pi/4 0 0];
u1=1.0; u2=0.0;

timecontrol=0;u=0;hh=1;v_d=0;v_a=0;inted=0;pp=0;ap=0;ppant=0;apant=0;
for i=1:nsamples
    tt=(i-1)*ts;
%-----
    u1ss(i,1)=u1;
    u2ss(i,1)=u2;

%%%%%%SIMULACION DE LA RESPUESTA DEL SISTEMA%%%%%%
    xx=x(i,1);
    v=x(i,2);
    a=x(i,3);
    theta=x(i,4);
    w=x(i,5);
    ag=x(i,6);
%%%%%%INICIO DE LA Rutina DE RESPUESTA DEL MODELO%%%%%%
    h=ts;
    k1x=h*v;
    k1t=h*w;
    [m1xx,m1tx]=f_mrtm2dof(xx,v,a,theta,w,ag,u1,u2);
    m1x=h*m1xx;
    m1t=h*m1tx;
```

```

k2x=h*(v+m1x);
k2t=h*(w+m1t);
[m2x,m2t]=f_mrtm2dof(xx+k1x,v+m1x,a,theta+k1t,w+m1t,ag,u1,u2);
m2x=h*m2x;
m2t=h*m2t;

xx = xx + 0.5*(k1x+k2x);
theta = theta + 0.5*(k1t+k2t);

v = v + 0.5*(m1x+m2x);
w = w + 0.5*(m1t+m2t);

a = m1xx; %aceleracion de posicion o desplazamiento lineal
ag = m1tx; % aceleracion angular

x(i+1,:)=[xx v a theta w ag];
%%%%%%%%%
end figure(1)
plotyy(simtime,[x(1:nsamples,1)],simtime,[x(1:nsamples,4)*57.2958]);
grid legend('r [m]','theta [ ]'); figure(2) subplot(121)
plot(simtime,[u1ss(1:nsamples,1)]); grid legend('u1 [V]');
axis([0,simtime(1,nsamples),0,u1*1.25]); subplot(122)
plot(simtime,[u2ss(1:nsamples,1)]); grid legend('u2 [V]');
axis([0,simtime(1,nsamples),0,u2*1.25]);

%%%%%%%%%%%%%% ANIMACION%%%%%%%%%%%%%%
sim=1;

%Datos para animacion
hei=1; le=0.3; offset=0.2; ddx=0.03; xc(1)=0; yc(1)=hei; xe(1)=0;
ye(1)=hei-le;

xc = x(:,1); yc = hei*ones(nsamples+1,1); xe = x(:,1) +
le*sin(x(:,4)); ye = hei + le*cos(x(:,4));

if sim==1
figure(4)
for i=1:10:nsamples
pause(0.005);
plot([-offset,2.5],[hei,hei],'-k',[-offset,-offset],[0,hei],
'-k',[2.5,2.5],[hei,0],'-k','linewidth',5);
grid;
hold on;
plot([xc(i),xe(i)],[yc(i),ye(i)],'-b','linewidth',3);
plot([xc(i)-ddx,xc(i)+ddx],[hei,hei],'-hy','linewidth',6);
axis([-offset,2.5,-offset,2.5]);
axis square
hold off;
end
end

-> function f_mrtm2dof
[xpp,thetapp]=f_mrtm2dof(x,xp,xpp,theta,thetapp,thetapp,u1,u2);
%=====
% Multivariable Translational Robot Manipulator Dynamic Model
% -----
% This function returns the values for the accelerations

```

```

% of the position and angle, respectively. This function
% is usually a part of a main program to simulate the
% dynamic behavior of the system.
%
% [xpp,thetapp]=f_mrtm2dof(x, xp, xpp, theta, thetap, thetapp, u1, u2)
%
% where:
%         thetapp(k)=ag;
%         thetap(k) =w;
%         theta(k)  =theta;
%         xpp(k)    =a;
%         xp(k)     =v;
%         x(k)      =x;
%=====
%
Mc=1.6492;      %Kg
m=0.0519;      %Kg
l=0.3;         %m
J=2.38e-3;     %Kgm^2
Jeq=3.0551e-4; %Kgm^2
Beq=5.3848e-2; %Nm/rad/s
C=9.5e-4;      %Kgm^2/s
Fc=2.81;       %Kg/s
g=9.81;        %m/s^2
Rp=0.048;      %m

%%%%%%%%%Constante motor DC%%%%%%%%%
Ra=5.3; Ka=5.5; n=12.5; Km=0.0421; Kb=0.0565;

%%%%%%%%% M, P & Q Quoeeficients del modelo del sistema%%%%%%%%%

m_11=((Ra*Rp)/(n*Ka*Km))*(Mc+m+(Jeq/Rp^2));
m_12=((Ra*Rp)/(n*Ka*Km))*m*l/2; m_21=(Ra/(n*Ka*Km))*m*l/2;
m_22=(Ra/(n*Ka*Km))*((m*l^2)/4+J+Jeq);
p_11=((Ra*Rp)/(n*Ka*Km))*(Fc+(Beq/Rp^2)+((n^2*Km*Kb)/(Ra*Rp^2)));
p_12=-((Ra*Rp)/(n*Ka*Km))*m*l/2; p_21=0;
p_22=(Ra/(n*Ka*Km))*(C+Beq+((n^2*Km*Kb)/Ra)); d_12=0;
d_21=-((Ra/(n*Ka*Km))*m*l*g/2;

%-----
% Mathematical model using the Euler-Lagrange Formulation
%-----
xpp1=(u1-p_11*xp-m_12*cos(theta)*thetapp-p_12*sin(theta)*thetap^2)/m_11;
thetapp=(u2-p_22*thetap-m_21*cos(theta)*xpp-d_21*sin(theta))/m_22;
xpp=xpp1;

```

## B.1.2. Simulación del sistema de control en tiempo continuo

Se presenta los programas y subprogramas ( funciones ) para realizar la simulación del sistema de control BRTM ; el progrAma principal se denomina:  $mrtm_{FLC_c.m}$

```

%Programa de simulacin en tiempo continuo
%Sistema: Manipulador Rob\otico Traslacional Multivariable
%Controlador: FLC
%         Defuzzificacin de orden 0.
%Funciones necesarias:

```

```

% FLC_t.m : Que a su vez necesita de: MF_t.m
% f_mrtm2dof

clear all; close all;

ts=0.005;          %Tiempo de muestreo virtual para el modelo
simtime=0:ts:8;    %Duracin de la simulacin
nsamples=size(simtime);nsamples=nsamples(1,2); %Nmero de muestras
Kprop_d=14;        %Const. proporcional de salida para la posicin
Kprop_a=0.25;      %Const. proporcional de salida para el ngulo
K_g=0.1398;        %Constante de la compensacin de gravedad

KX=0.60; KXd=0.30; %Ganancias anteriores a la fuzzificacin

x(1,:)=[0 0 0 pi 0 0]; %Estados iniciales
xdes(1,1)=0;          %Posicin inicial deseada
pos_des=input('Ingrese la posicin deseada = '); %Pos. deseada final
ang_des=input('Ingrese el ngulo deseado = '); %Angulo deseado
T_rise=1;             %Tiempo de la rampa
Vrise=(pos_des/T_rise);

%CONTROLADOR DE POSICIN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Funciones de Pertenencia
%Entradas:
%-----
%distance=Matriz de parmetros de las MSFs de Distance
distance=[ -15.00 -0.400 1; % [ a c sig;
           0.11 -0.250 0; % sigma c gauss;
           0.11 0.000 0;
           0.11 0.250 0;
           15.00 0.400 1];

%Distance=Matriz de parmetros de las MSFs de Distance
distance_d=[ -15.000 -0.350 1;
             0.200 0.000 0;
             15.000 0.350 1];

%Salida:
%-----
%voltage=vector de voltajes
%MSFs      1 2 3 4 5 6 7
voltage_r=[-21 -14 -7 0 7 14 21];

%Base de Reglas

%distance-d
%NL NM NS Z PS PM PL
RB_r=...
[1 2 3; %NL
 2 3 4; %NS
 3 4 5; %Z      distance
 4 5 6; %PS
 5 6 7]; %PL

%CONTROLADOR DEL BRAZO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Funciones de Pertenencia
%Entradas:

```

```

%-----
%angle=Matriz de parmetros de las MSFs de Angle
angle=[-4.0 -2.0 1;
       0.8 -0.5 0;
       0.8 0.0 0;
       0.8 0.5 0;
       4.0 2.0 1];

%angle_d=Matriz de parmetros de las MSFs de Angle_d
angle_d=[-0.75 -7.5 1;
         3.00 0.0 0;
         0.75 7.5 1];

%Salida:
%-----
%voltage=vector de voltajes
%MSFs      1  2  3  4  5  6  7
voltage_q=[-12 -4.5 -2 0 2 4.5 12];

%Base de Reglas

%angle-d
%NL Z PL
RB_q=...
[1 2 3 ; %NL
 2 3 4 ; %NS
 3 4 5 ; %Z      angle
 4 5 6 ; %PS
 5 6 7 ]; %PL

timecontrol=0;u=0;hh=0;v_d=0;v_a=0;inted=0;pp=0;asex=0;ppant=0;asexant=0;
for i=1:nsamples
    tt=(i-1)*ts;

%%%%%%%%%%GENERADOR DE TRAYECTORIA%%
if (tt<T_rise);
    vdes(i,1)=Vrise;
    xdes(i+1,1)=xdes(i,1)+vdes(i,1)*ts;    %Se genera una rampa
else
    vdes(i,1)=0;
    xdes(i+1)=pos_des;
end

%%%%%%%%% CONTROLADOR %%%
timecontrol=tt;    %Tiempo del controlador coincidente con el de la
hh=hh+1;          %simulacin del modelo, para el caso continuo.
ppant=pp;
pp=(x(i,1));
vp=(pp-ppant)/ts;

asexant=asex;    %Registro de variables para uso posterior
asex=(x(i,4)*57.2958);
wsex=(x(i,5)*57.2958);wphh(hh,1)=wsex;

ed(i,1)=KX*(xdes(i,1)-pp);    %Clculo de las seales de error.
ed_d(i,1)=KXd*(vdes(i,1)-vp);
eq(i,1)=ang_des-asex;

%Evaluacin Fuzzy:
v_d=FLC_t(ed(i,1),ed_d(i,1),distance,distance_d,RB_r,voltage_r);

```



```

v_a=FLC_t(eq(i,1),-wsex,angle,angle_d,RB_q,voltage_q);

xhh(hh,1)=x(i,1);
thetahh(hh,1)=x(i,4);
thh(hh,1)=tt;
%end
v_dmat(i,1)=v_d; v_amat(i,1)=v_a; %intedmat(i,1)=inted;
%----- deadzone -----
%dzone=4.1111;
%if u<-dzone
% u=u+dzone;
%elseif u>dzone
% u=u-dzone;
%else
% u=0;
%end
%-----
uss(i,1)=Kprop_d*v_d;
uss(i,2)=Kprop_a*v_a-K_g*sin(x(i,4));
usg(i,1)=-K_g*sin(x(i,4));
u1=uss(i,1);%Kprop*v_d;
u2=uss(i,2);%0.1;%0*0.01*v_a+0.0519*0.15*9.81*sin(x(i,4))*0.0002;
%%%%%%%% SIMULACION DE LA RESPUESTA DEL SISTEMA
xx=x(i,1);
v=x(i,2);
a=x(i,3);
theta=x(i,4);
w=x(i,5);
ag=x(i,6);
%%%%%%%% INICIO DE LA RUTINA DE RESPUESTA DEL MODELO
%%%%%%%% RUNGE KUTTA DE 2DO. ORDEN
h=ts;

k1x=h*v;
k1t=h*w;
[m1xx,m1tx]=f_mrtm2dof(xx,v,a,theta,w,ag,u1,u2);
m1x=h*m1xx;
m1t=h*m1tx;

k2x=h*(v+m1x);
k2t=h*(w+m1t);
[m2x,m2t]=f_mrtm2dof(xx+k1x,v+m1x,a,theta+k1t,w+m1t,ag,u1,u2);
m2x=h*m2x;
m2t=h*m2t;

xx = xx + 0.5*(k1x+k2x);
theta = theta + 0.5*(k1t+k2t);

v = v + 0.5*(m1x+m2x);
w = w + 0.5*(m1t+m2t);

a = m1xx;
ag = m1tx;

x(i+1,:)= [xx v a theta w ag];
%%%%%%%%
end

```

```

figure(1) subplot(121)
plot(simtime,xdes(1:nsamples,1),'r',simtime,x(1:nsamples,1),'b',...
'LineWidth',1); grid
legend('X deseado','X cabezal'); axis([0,8,0,pos_des*1.1]);axis
square; xlabel('tiempo [s]');ylabel('posicin [m]'); title('a)
Posicin. '); subplot(122) plot(simtime,[x(1:nsamples,4)*57.2958],...
'LineWidth',1); grid
legend('Theta');
%axis([0,8,-10,10]);axis square;
xlabel('tiempo [s]');ylabel('desviacin angular [ sex]'); title('b)
Desviacin angular. ');

xxxx=1; if xxxx==1 figure(2) subplot(121)
plot(simtime,uss(:,1),'r','LineWidth',1); grid
%legend('u total','u seguimiento');
xlabel('tiempo [s]');ylabel('seal de control: carro [V]');
axis([0,8,-6,12]);axis square; title('a) Seal de control: Posicin
del carro. '); subplot(122)
plot(simtime,uss(:,2),'r',simtime,usg(:,1),'b','LineWidth',1); grid
legend('u total','u comp. g'); axis([0,8,-6,12]);axis square;
xlabel('tiempo [s]');ylabel('seal de control: brazo [V]');
title('b) Seal de control: ngulo del brazo. '); end

%figure(3);
%plot(thh,[xhh*10 thetahh*57.2958 u_track u_damp]); grid;
%legend('X x10','theta (deg)','u_tracking','u_damping');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ANIMACION %%%%%%%%%
if pos_des<0
    sim=2;
else
    sim=1;
end
%sim=3;
%Datos para animacion
hei=1.2; le=0.3; offset=0.2; ddx=0.03; xc(1)=0; yc(1)=hei; xe(1)=0;
ye(1)=hei-le;

xc = x(:,1); yc = hei*ones(nsamples+1,1); xe = x(:,1) +
le*sin(x(:,4)); ye = hei + le*cos(x(:,4));

if sim==1
    figure(4)
    for i=1:10:nsamples
        pause(0.005);
        plot([-offset,2.5],[hei,hei'],'-k',[-offset,-offset],[0,hei'],'-k',
[2.5,2.5],[hei,0'],'-k','linewidth',5);
        grid;
        hold on;
        plot([xc(i),xe(i)],[yc(i),ye(i)],'-g','linewidth',2);
        plot([xc(i)-ddx,xc(i)+ddx],[hei,hei'],'-r','linewidth',6);
        plot(xe(i),ye(i),'ob','linewidth',4);
        axis([-offset,2.5,-offset,2.5]); axis square;
        hold off;
    end
end

if sim==2
    figure(4)

```

```

for i=1:10:nsamples
    pause(0.005);
    plot([offset,-2.5],[hei,hei],'-k',[offset,offset],[0,hei],'-k',
        [-2.5,-2.5],[hei,0],'-k','linewidth',5);
    grid;
    hold on;
    plot([xc(i),xe(i)],[yc(i),ye(i)],'-g','linewidth',2);
    plot([xc(i)-ddx,xc(i)+ddx],[hei,hei],'-r','linewidth',6);
    plot(xe(i),ye(i),'ob','linewidth',4);
    axis([-2.5,offset,-offset,2.5]); axis square;
    hold off;
end
end

```

### B.1.3. Simulación del sistema de control en tiempo discreto

Se presenta los programas y subprogramas ( funciones ) para realizar la simulación del sistema de control BRTM ; el programa principal se denomina:  $mrtm_{FLC_d}3.m$

```

%Programa de simulacin en tiempo discreto
%Sistema: Manipulador Robtico Traslacional Multivariable
%Controlador: FLC
%          Defuzzificacin de orden 0.
%Funciones necesarias:
%  FLC_k1.m : Que a su vez necesita de: MF_k.m
%  f_mrtm2dof

clear all; close all;

fsc=100;          %Frecuencia de muestreo del controlador
tsc=1/fsc;       %Perodo de muestreo del controlador
ts=0.005;        %Perodo de muestreo para la simulacin
simtime=0:ts:8;  %Tiempo de prueba
nsamples=size(simtime);nsamples=nsamples(1,2);
Kprop=14;        %Constante proporcional (posicin)

KX=0.60; KXd=0.30; %Ganancias para la posicin
                % y velocidad lineal

x(1,:)=[0 0 0 pi 0 0]; %Estados iniciales
xdes(1,1)=0;          %Inicializacin de la posicin deseada.
longpend=0.3; Tpend=2.0061*sqrt(longpend); %Parmetros para
                %formar la trayectoria.
pos_des=input('Ingrese la posicin deseada = ');
ang_des=input('Ingrese el ngulo deseado = ');
Vmax=(pos_des/Tpend)*8/13; %Parmetro para la trayectoria.
T_rise=1; Vrise=(pos_des/T_rise);

%CONTROLADOR DE POSICIN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Funciones de Pertenencia

%Entradas:
%-----
%distance=Matriz de parametros de las MFs de posicin
distance=[ -15.00 -0.400 1; % [ a   c sig;

```

```

0.11 -0.250 0; % sigma c gauss;
0.11 0.000 0;
0.11 0.250 0;
15.00 0.400 1];

%Distance=Matriz de parmetros de las MSFs de Distance
distance_d=[ -15.000 -0.350 1;
             0.200 0.000 0;
             15.000 0.350 1];

%Salida:
%-----
%voltage=vector de voltajes
%MSFs      1  2  3 4 5 6 7
voltage_r=[-21 -14 -7 0 7 14 21]; %Valores singleton

%Base de Reglas

%velocidad lineal
% N Z P
RB_r=...
[1 2 3; %NL
 2 3 4; %NS
 3 4 5; %Z      posicion
 4 5 6; %PS
 5 6 7]; %PL

%CONTROLADOR DEL BRAZO

%%%%%%%%%%%%%%
%Funciones de Pertenencia

%Entradas:
%-----
%angle=Matriz de parmetros de las MFs de Angle
angle=[-4.0 -2.0 1;
       0.8 -0.5 0;
       0.8 0.0 0;
       0.8 0.5 0;
       4.0 2.0 1];
sizerangle=5; mbflabels_q=strvcat('NG','NP','Z','PP','PG');

%angle_d=Matriz de parmetros de las MSFs de Angle_d
angle_d=[-0.75 -7.5 1;
         3.00 0.0 0;
         0.75 7.5 1];
sizerangle_d=3;

%Salida:
%-----
%voltage=vector de voltajes
%MSFs      1  2  3 4 5 6 7
voltage_q=[-12 -4.5 -2 0 2 4.5 12];

%Base de Reglas

%angle-d

```

```

% N Z P
RB_q=...
[1 2 3 ; %NL
 2 3 4 ; %NS
 3 4 5 ; %Z angle
 4 5 6 ; %PS
 5 6 7 ]; %PL

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%TRANSFORMACION DE LAS FUNCIONES DE MEMBRESIA AL
%UNIVERSO DE NUMEROS ENTEROS

%%Constantes de transformacin
npulse_encoder1=512; npulse_encoder2=512; %Parmetros para
qep=2; ls7184=4; dpolea=0.048*2; n1=19.745; %calcular las constantes

m2pulse=npulse_encoder1*qep*n1/(pi*dpolea);
gsex2pulse=npulse_encoder2*ls7184/360; v2pulse=m2pulse*tsc;
wsex2pulse=gsex2pulse*tsc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Tamaos de las MFs
sizep=size(distance); sizep=sizep(1,1); sizev=size(distance_d);
sizev=sizev(1,1); sizea=size(angle); sizea=sizea(1,1);
sizew=size(angle_d); sizew=sizew(1,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Se conserva la ltima columna de cada MF.
distance_pulsos(:,3)=distance(:,3);
distance_d_pulsos(:,3)=distance_d(:,3);
angle_pulsos(:,3)=angle(:,3); angle_d_pulsos(:,3)=angle_d(:,3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Se transforma el offset ("c") de cada MF.
distance_pulsos(:,2)=distance(:,2).*m2pulse;
distance_d_pulsos(:,2)=distance_d(:,2).*v2pulse;
angle_pulsos(:,2)=angle(:,2).*gsex2pulse;
angle_d_pulsos(:,2)=angle_d(:,2).*wsex2pulse;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Se transforma la ganancia en cada MF.
for k=1:sizep
    if distance(k,3)==1
        distance_pulsos(k,1)=distance(k,1)*10000/m2pulse;
    else
        distance_pulsos(k,1)=10000/(distance(k,1))/m2pulse;
    end
end

for k=1:sizev
    if distance_d(k,3)==1
        distance_d_pulsos(k,1)=distance_d(k,1)*10000/v2pulse;
    else
        distance_d_pulsos(k,1)=10000/(distance_d(k,1))/v2pulse;
    end
end

for k=1:sizea
    if angle(k,3)==1
        angle_pulsos(k,1)=angle(k,1)*10000/gsex2pulse;

```

```

else
    angle_pulsos(k,1)=10000/(angle(k,1))/gsex2pulse;
end
end

for k=1:sizeof
    if angle_d(k,3)==1
        angle_d_pulsos(k,1)=angle_d(k,1)*10000/wsex2pulse;
    else
        angle_d_pulsos(k,1)=10000/(angle_d(k,1))/wsex2pulse;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% INICIO DE LA SIMULACION %%%%%%%%%
timecontrol=0;u=0;hh=0;v_d=0;v_a=0;inted=0;pp=0;ases=0;ppant=0;asesant=0;ap=0;apant=0;
for i=1:nsamples
    tt=(i-1)*ts; %tt=valor del tiempo en segundos.

    %%%%%%%%% GENERADOR DE TRAYECTORIA
    if (tt<T_rise);
        vdes(i,1)=Vrise;
        xdes(i+1,1)=xdes(i,1)+vdes(i,1)*ts;
    else
        vdes(i,1)=0;
        xdes(i+1,1)=pos_des;
    end

    %%%%%%%%% CONTROLADOR
    controlar=floor((tt-timecontrol)/tsc);
    if controlar==1; %Detecta el momento en que el controlador debe
        timecontrol=tt; %muestrear las entradas y actualizar la seal
        hh=hh+1; % de control.
        controltime(hh)=tt;

        %EN ADELANTE TODAS LAS VARIABLES SE TRANSFORMAN DE MAGNITUDES FSICAS
        %A NNERO DE PULSOS
        ppant=pp; %Se guarda el valor de la posicin
        pp=floor(m2pulse*x(i,1)); %prdda de info. debida a la discretizacin
        vp=(pp-ppant);

        apant=ap;
        ap=floor(gsex2pulse*x(i,4)*57.2958);%prdda de info. debida a la discretizacin
        wp=0*(ap-apant);
        wphh(hh,1)=wp;

        ed(i,1)=KX*(xdes(i,1)*m2pulse-pp);
        ed_d(i,1)=KXd*(vdes(i,1)*v2pulse-vp);
        eq(i,1)=ang_des*gsex2pulse-ap;

        v_d=FLC_k1(ed(i,1),0*ed_d(i,1),distance_pulsos,distance_d_pulsos,RB_r,voltage_r);
        v_a=FLC_k1(eq(i,1),0,angle_pulsos,angle_d_pulsos,RB_q,voltage_q);

        xhh(hh,1)=x(i,1);
        thetahh(hh,1)=x(i,4);

```

```

        thh(hh,1)=tt;
    end
    v_dmat(i,1)=v_d; v_amat(i,1)=v_a; %intedmat(i,1)=inted;
    %----- deadzone -----
    %dzone=4.1111;
    %if u<-dzone
    %   u=u+dzone;
    %elseif u>dzone
    %   u=u-dzone;
    %else
    %   u=0;
    %end
    %-----
    uss(i,1)=Kprop*v_d;
    uss(i,2)=0.5*v_a-0.1398*sin(x(i,4));
    usg(i,1)=-0.1398*sin(x(i,4));
    u1=uss(i,1);
    u2=uss(i,2);

    %%%%%%%%%% SIMULACION DE LA RESPUESTA DEL SISTEMA
    xx=x(i,1);
    v=x(i,2);
    a=x(i,3);
    theta=x(i,4);
    w=x(i,5);
    ag=x(i,6);

% INICIO DE LA RUTINA DE RESPUESTA DEL MODELO

%RUNGE KUTTA DE 2DO. ORDEN
    h=ts;
    k1x=h*v;
    k1t=h*w;
    [m1xx,m1tx]=f_mrtm2dof(xx,v,a,theta,w,ag,u1,u2);
    m1x=h*m1xx;
    m1t=h*m1tx;

    k2x=h*(v+m1x);
    k2t=h*(w+m1t);
    [m2x,m2t]=f_mrtm2dof(xx+k1x,v+m1x,a,theta+k1t,w+m1t,ag,u1,u2);
    m2x=h*m2x;
    m2t=h*m2t;

    xx = xx + 0.5*(k1x+k2x);
    theta = theta + 0.5*(k1t+k2t);

    v = v + 0.5*(m1x+m2x);
    w = w + 0.5*(m1t+m2t);

    a = m1xx;
    ag = m1tx;

    x(i+1,:)= [xx v a theta w ag];
    %%%%%%%%%%
end

```

```

figure(1) subplot(121)
plot(simtime,xdes(1:nsamples,1),'r',simtime,x(1:nsamples,1),'b',...
'LineWidth',1); grid
legend('X deseado','X cabezal'); axis([0,8,0,pos_des*1.1]);axis
square; xlabel('tiempo [s]');ylabel('posicin [m]'); title('a
Posicin. '); subplot(122) plot(simtime,[x(1:nsamples,4)*57.2958],...
'LineWidth',1); grid
legend('Theta');
%axis([0,8,-10,10]);axis square;
xlabel('tiempo [s]');ylabel('desviacin angular [ sex]'); title('b
Desviacin angular. ');

xxxx=1; if xxxx==1 figure(2) subplot(121)
plot(simtime,uss(:,1),'r','LineWidth',1); grid
%legend('u total','u seguimiento');
xlabel('tiempo [s]');ylabel('seal de control: carro [V]');
axis([0,8,-6,12]);axis square; title('a Seal de control: Posicin
del carro. '); subplot(122)
plot(simtime,uss(:,2),'r',simtime,usg(:,1),'b','LineWidth',1); grid
legend('u total','u comp. g'); axis([0,8,-6,12]);axis square;
xlabel('tiempo [s]');ylabel('seal de control: brazo [V]');
title('b Seal de control: ngulo del brazo. '); end

%figure(3);
%plot(thh,[xhh*10 thtaih*57.2958 u_track u_damp]); grid;
%legend('X x10','theta (deg)','u_tracking','u_damping');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ANIMACION
if pos_des<0
sim=2;
else
sim=1;
end
%sim=3;
%Datos para animacion
hei=1.2; le=longpend; offset=0.2; ddx=0.03; xc(1)=0; yc(1)=hei;
xe(1)=0; ye(1)=hei-le;

xc = x(:,1); yc = hei*ones(nsamples+1,1); xe = x(:,1) +
le*sin(x(:,4)); ye = hei + le*cos(x(:,4));

if sim==1
figure(4)
for i=1:10:nsamples
pause(0.005);
plot([-offset,2.5],[hei,hei'],'-k',[-offset,-offset],[0,hei'],'-k',
[2.5,2.5],[hei,0'],'-k','linewidth',5);
grid;
hold on;
plot([xc(i),xe(i)],[yc(i),ye(i)],'-g','linewidth',2);
plot([xc(i)-ddx,xc(i)+ddx],[hei,hei'],'-r','linewidth',6);
plot(xe(i),ye(i),'ob','linewidth',4);
axis([-offset,2.5,-offset,2.5]); axis square;
hold off;
end
end

if sim==2
figure(4)

```



```

for i=1:10:nsamples
    pause(0.005);
    plot([offset,-2.5],[hei,hei],'-k',[offset,offset],[0,hei],'-k',
        [-2.5,-2.5],[hei,0],'-k','linewidth',5);
    grid;
    hold on;
    plot([xc(i),xe(i)],[yc(i),ye(i)],'-g','linewidth',2);
    plot([xc(i)-ddx,xc(i)+ddx],[hei,hei],'-r','linewidth',6);
    plot(xe(i),ye(i),'ob','linewidth',4);
    axis([-2.5,offset,-offset,2.5]); axis square;
    hold off;
end
end

```

### B.1.4. Simulación de los sensores de los motores1 y motor2

Los sensores empleados son de naturaleza discreta. Los encoders permiten contar un número entero de fracciones de revolución, de este modo, al obtener directamente la posición a través del conteo de pulsos, se computa un valor que es proporcional al número de pulsos de la señal en cuadratura. La doble señal en cuadratura de los encoders empleados permite definir el desplazamiento angular no sólo en magnitud, sino también en sentido. En Matlab, se ha empleado la siguiente secuencia para discretizar el ángulo medido por los encoders en función de la resolución:

El programa desarrollado en matlab es el archivo **mrtm2dof-discrete -sensores.m**

```

clear all; close all;
%Programa de simulación en tiempo discreto para sensores del motor1 y motor2
%Sistema: Manipulador Robótico Translacional Multivariable
%de 2 grados de libertad.
%Controlador: FLC
%          Defuzzificación de orden 0.

npulse_encoder1=512;%numero de pulsos del encoder para la posición
qep=2;%multiplicador del QEP
npulse_encoder2=512;
ls7184=4;%multiplicador del t2
dpolea=0.048*2;
m2pulse=npulse_encoder1*qep*19.745/(pi*dpolea);
gsex2pulse=npulse_encoder2*ls7184/360;

ts=0.001;
simtime=0:ts:3;
nsamples=size(simtime);nsamples=nsamples(1,2);
x(:,1)=2*sin(1*simtime');
x(:,2)=2*cos(1*simtime');
x(:,4)=90*sin(3*simtime');
x(:,5)=270*cos(3*simtime');

fsc=150;
tsc=1/fsc;

```

```

v2pulse=npulse_encoder1*qep*5.9/(pi*dpolea)*tsc;
wsex2pulse=npulse_encoder2*ls7184/360*tsc;

timecontrol=0;hh=0;ppac=0;apac=0;ppant=0;apant=0;

for i=1:nsamples
    tt=(i-1)*ts;
    controlar=floor((tt-timecontrol)/tsc);
if controlar==1;
    timecontrol=tt;
    hh=hh+1;
    controltime(hh)=tt;
    ppant=ppac;
    ppac=floor(m2pulse*x(i,1));
    apant=apac;
    apac=floor(gsex2pulse*x(i,4));
end
pp(i)=ppac;
vp(i)=(ppac-ppant);
ap(i)=apac;
wp(i)=(apac-apant);

end

figure(1)
subplot(221)
plot(simtime,x(:,1),'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('posicin [m]');
title('a) Posicin-T. continuo','FontSize',11);
subplot(223)
plot(simtime,x(:,2),'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('velocidad [m/s]');
title('b) Velocidad','FontSize',11);
subplot(222)
plot(simtime,x(:,4),'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('ngulo [Sex]');
title('c) Desviacin Angular','FontSize',11);
subplot(224)
plot(simtime,x(:,5),'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('velocidad angular [Sex/s]');
title('d) Velocidad Angular','FontSize',11);

figure(2)
subplot(221)
plot(simtime,pp,'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('posicin [pulsos]');
title('a) Posicin-150Hz','FontSize',11);
subplot(223)
plot(simtime,vp,'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('velocidad [pulsos/Ts]');
title('b) Velocidad','FontSize',11);
subplot(222)
plot(simtime,ap,'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('ngulo [pulsos]');
title('c) Desviacin Angular','FontSize',11);
subplot(224)
plot(simtime,wp,'k','LineWidth',1);grid;

```

```

xlabel('tiempo [s]');ylabel('velocidad angular [pulsos/Ts]');
title('d) Velocidad Angular','FontSize',11);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fsc=100;
tsc=1/fsc;
ts=0.001;
v2pulse=npulse_encoder1*qep*5.9/(pi*dpolea)*tsc;
wsex2pulse=npulse_encoder2*ls7184/360*tsc;

timecontrol=0;hh=0;ppac=0;apac=0;ppant=0;apant=0;

for i=1:nsamples
    tt=(i-1)*ts;
    controlar=floor((tt-timecontrol)/tsc);
    if controlar==1;
        timecontrol=tt;
        hh=hh+1;
        controltime(hh)=tt;
        ppant=ppac;
        ppac=floor(m2pulse*x(i,1));
        apant=apac;
        apac=floor(gsex2pulse*x(i,4));
    end
    pp(i)=ppac;
    vp(i)=(ppac-ppant);
    ap(i)=apac;
    wp(i)=(apac-apant);

end

figure(3)
subplot(221)
plot(simtime,pp,'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('posicin [pulsos]');
title('a) Posicin-100Hz','FontSize',11);
subplot(223)
plot(simtime,vp,'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('velocidad [pulsos/Ts]');
title('b) Velocidad','FontSize',11);
subplot(222)
plot(simtime,ap,'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('ngulo [pulsos]');
title('c) Desviacin Angular','FontSize',11);
subplot(224)
plot(simtime,wp,'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('velocidad angular [pulsos/Ts]');
title('d) Velocidad Angular','FontSize',11);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fsc=50;
tsc=1/fsc;
ts=0.001;
v2pulse=npulse_encoder1*qep*5.9/(pi*dpolea)*tsc;
wsex2pulse=npulse_encoder2*ls7184/360*tsc;

timecontrol=0;hh=0;ppac=0;apac=0;ppant=0;apant=0;

```

```

for i=1:nsamples
    tt=(i-1)*ts;
    controlar=floor((tt-timecontrol)/tsc);
if controlar==1;
    timecontrol=tt;
    hh=hh+1;
    controltime(hh)=tt;
    ppant=ppac;
    ppac=floor(m2pulse*x(i,1));
    apant=apac;
    apac=floor(gsex2pulse*x(i,4));
end
pp(i)=ppac;
vp(i)=(ppac-ppant);
ap(i)=apac;
wp(i)=(apac-apant);

end

figure(4)
subplot(221)
plot(simtime,pp,'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('posicin [pulsos]');
title('a) Posicin-50Hz','FontSize',11);
subplot(223)
plot(simtime,vp,'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('velocidad [pulsos/Ts]');
title('b) Velocidad','FontSize',11);
subplot(222)
plot(simtime,ap,'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('ngulo [pulsos]');
title('c) Desviacin Angular','FontSize',11);
subplot(224)
plot(simtime,wp,'k','LineWidth',1);grid;
xlabel('tiempo [s]');ylabel('velocidad angular [pulsos/Ts]');
title('d) Velocidad Angular','FontSize',11);

```

# Apéndice C

## Programas en C

### C.1. Listado de Programas

Los programas para la adquisición de la data de la velocidad del servomotor DC están estructurados por los siguientes programas:

programa principal *adqs.cpp* que se encarga de la adquisición de los datos de la posición y velocidad debido a una entrada de referencia escalón. Este programa es utilizado para la identificación de los parámetros del servomotor DC que se expone en el Apéndice A. El código del programa se lista a continuación.

⇒ *adqs.cpp*

```
#include "adq_d1.h"
#include <stdio.h>
#include <math.h>
#include <assert.h>
#include <alloc.h>
#include <conio.h>
#include sensor 1
const float ESCALA= 2.*M_PI/(100.*12.5);
/* Declaración de las funciones propias del programa */
void Inicializa_Sistema();
void Mide_Variables(float *yk, float *vel, float t);
float Aplica_Control(float u, float ykp);
/* Declaracin de las variables globales */
float N, y, yaut, T, t, tsimul, Fs;
int p0, pk;
float VcFC, VcFE, Wth, Ofst;
/* Funciones*/
float Aplica_Control(float u, float ykp)
{
    float v;
    if(fabs(ykp)<=Wth)
    {
        if(u<0.) v = -VcFE;
        if(u>0.) v = VcFE;
    }
    else
    {
        if(u<0.) v = -VcFC;
        if(u>0.) v = VcFC;
    }
    if(u+v<-1.4) u= -1.4;
```

```

    else if(u+v>1.4) u= 1.4;
    EnviarVoltaje(u+v+Ofst);
    return u;
}
void Inicializa_Sistema()
{
    VcFE = .2; VcFC = .15;
    Wth = .1; Ofst = 2.17; //servosystem
    Fs = 100.; T = 1./Fs;
    tsimul = 20.;
    ConfigurarHardware(Fs);
    EnviarVoltaje(Ofst);
    p0 = LeerPosicion(sensor);
/* p0 = LeerPosicion() */
}
void Mide_Variables(float *y, float *vel)
{
/* Ahora lee posicion */
    pk = LeerPosicion(sensor);
    *y = yant + ((float)(abs(pk-p0)>8000?0:pk-p0))*ESCALA;
    p0 = pk;
/* Y la velocidad angular */
/* vel = 0.*/
    *vel = (5.*(*y-yant)+*vel)/(1.+5.*T);
/* vel= (*y-yant)/(T);*/
/* Actualizamos los estados*/
    yant=*y;
}
void main()
{
    int ant=0, act=0, k, n;
    FILE *out;
    float *yb, *ub, *velb, vel, y, u;
    float yant, w, f;
    Inicializa_Sistema();
    clrscr();
    assert(tsimul<120);
    N = tsimul/T;
    n = (int)N;
    t = vel= 0.;
/* resetContadores();*/
    assert(NULL! =(ub= (float *)calloc(n/5,sizeof(float))));
    assert(NULL! =(yb= (float *)calloc(n/5,sizeof(float))));
    assert(NULL! =(velb= (float *)calloc(n/5,sizeof(float))));
    for(k=0; k<n;)
    {
        act=NivelClock();
        if ((ant==0)&&(act==1))
        {
            Mide_Variables(&y, &vel);
/* Seal de Control */
            f = 0.8;
/* w = 2*M_PI*f;*/
/* u = 0.5*sin(w*t);*/
            u = 1;

```

```

    u = Aplica_Control(u, vel);
    if(!(k%5))
    {
        yb[k/5]=y*180./M_PI; ub[k/5]=u; velb[k/5]=vel;
    }
    /* ResetContadores();*/
    k++;
    t+ = T;
    gotoxy(15,5); printf("*****");
    gotoxy(18,7); printf("Identificacion del Motor D.C.");
    gotoxy(15,9); printf("*****");
    gotoxy(21,11);printf("tiempo : t= %+5.3f",t);
    gotoxy(21,17);printf("señal de control : u= %+5.3f",u);
    gotoxy(21,19);printf("posición angular : y= %+5.3f",y*180./M_PI);
    gotoxy(21,21);printf("velocidad angular : vel= %+5.3f",vel);
    gotoxy(15,25);printf("*****");
    if(kbhit()) if(getch()==27) break;
    /* ResetContadores();*/
    }
    ant=act;
    }
    EnviarVoltaje(Ofst);
    out=fopen("adqs24.dat","wt");
    for (int i=0; i<n/5; i++)
    fprintf(out,"%10f%10f%10f%10f\n",i*5*T, yb[i], ub[i], velb[i]);
    delete yb, ub, velb;
    fclose(out);
}

```

El programa *adq\_d1.cpp* es el encargado de configurar la tarjeta LABPC+ en forma análoga al programa *Slid\_pen.cpp*. El código del programa es listado a continuación.

```

⇒ adq_d1.cpp
#include "adq_d1.h"
#include "nidaq.h"
#include <dos.h>
#include DAC1 0x266
#include PORTA 0x270
#include PORTB 0x271
#include PORTC 0x272
#includec void ConfigurarHardware (float frcc)
{
    int periodoTimer, deviceCode;
    periodoTimer = (int)(2000000/frcc);
    USE_LAB();
    Init_DA_Brds(1,&deviceCode);
    ICTR_Setup(1,0,3,periodoTimer,1);
    outportb(0x273, 0x9A);
}
void EnviarVoltaje (float u)
{

```

```

    unsigned salidaDigital;
    if(u<0.) u=0.8;
    if(u>3.7) u=3.7;
    salidaDigital = (unsigned) (409.6*u+0.5);
    outport(DAC1,salidaDigital);
}
unsigned int LeerPosicion(int sensor)
{
    unsigned char byteH,byteL;
    if(sensor) outportb(PORTC,inportb(PORTC)&251);
    else outportb(PORTC,inportb(PORTC)|4);
    byteL = inportb(PORTA);
    byteH = inportb(PORTB);
    return (256*byteH+byteL);
}
int NivelClock()
{
    unsigned char bytePoll;
    bytePoll = inportb (PORTC);
    return (bytePoll>127);
}
void ResetContadores()
{
    outportb(PORTC,inportb(PORTC)|8);
    for(int i=0;i<10000;i++);
    outportb(PORTC,inportb(PORTC)&(255-8));
}
}

```

El programa *adq\_d1.h* contiene el manejo de las funciones de la tarjeta LABPC+ para el manejo de los datos de entrada/dsalida. El código del programa se lista a continuación.

```

⇒ adq_d1.h
#ifndef _ADQ_D1_H
#define _ADQ_D1_H
    void ConfigurarHardware (float);
    void EnviarVoltaje (float);
    unsigned int LeerPosicion(int);
    int NivelClock ();
#endif

```