

**UNIVERSIDAD NACIONAL DEL CALLAO
ESCUELA DE POSGRADO**

**UNIDAD DE POSGRADO DE LA FACULTAD DE INGENIERÍA
ELÉCTRICA Y ELECTRÓNICA**



TESIS

**“MODELADO Y SINTONIZACIÓN DE UN CONTROLADOR PID DE
UN ROBOT TIPO PUMA DE TRES GRADOS DE LIBERTAD
UTILIZANDO LÓGICA DIFUSA”**

**TESIS PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN
CIENCIAS DE LA ELECTRÓNICA CON MENCIÓN: CONTROL Y
AUTOMATIZACIÓN**

AUTOR: ELÍAS JOSUÉ ALBA MEJÍA

Callao, 2019

PERÚ

MIEMBROS DEL JURADO

DR. SANTIAGO LÍNDER, RUBIÑOS JIMÉNEZ : PRESIDENTE

Mg. CARLOS ALBERTO, HUAYLLASCO MONTALVA : SECRETARIO

M.sc. RICARDO RAUL, RODRIGUEZ BUSTINZA : MIEMBRO

DR. NOÉ MANUAL JESÚS, CHÁVEZ TEMOCHE : MIEMBRO

M.sc. JACOB ASTOCONDOR VILLAR : ASESOR

N° DE LIBRO : 01

ACTA FOLIO N° : 080

FECHA DE APROBACIÓN : Enero 22 del 2019

RESOLUCIÓN DIRECTORAL : 004-2019-DUP-FIEE

DEDICATORIA

Esta tesis la consagro a los Docentes de mi alma mater quiénes supieron guiarme por el camino doctrinario de la investigación científica, de igual manera, por darme fuerza para seguir adelante y no desvanecer sin perder la dignidad ni declinar en el intento de la solución de los problemas objetos de investigación.

Dedico este trabajo a las personas que siempre confiaron en mí, me apoyaron y alentaron en los momentos difíciles, y continuar el camino, mis hijos e esposa gracias por su amor y comprensión, los quiero mucho.

AGRADECIMIENTO

Agradezco infinitamente a todas las personas que de una forma u otra han aportado un granito de arena a todos los profesores que durante la carrera me guiaron y aconsejaron por el camino correcto, haciendo que me esforzara al máximo, gracias por sus enseñanzas la realización de esta tesis a los docentes de Maestría de la Universidad Nacional.

A mi Madre, que me enseñó todo en la vida y siempre me apoyó en los momentos más difíciles de mi infancia, gracias por todo, lo que soy te lo debo a ti.

A mi esposa, que ha estado firme a mi lado en todo momento y ha soportado mis momentos difíciles en concluir mis estudios.

A TODOS MUCHAS GRACIAS.

El Autor

ÍNDICE

CARÁTULA	i
PÁGINA DE RESPETO	ii
HOJA DE REFERENCIA DEL JURADO Y APROBACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE	1
ÍNDICE DE TABLAS	4
INDICE DE FIGURAS	5-6
RESUMEN	7
ABSTRACT	8
I. PLANTEAMIENTO DE LA INVESTIGACIÓN	10
1.1 Identificación del Problema	10
1.2 Formulación del Problema	14
1.3 Objetivos de la Investigación	14
1.3.1 Objetivo General	14
1.3.2 Objetivos Específicos	14
1.4 Justificación	15
1.5 Limitaciones	15
II. MARCO TEÓRICO	16
2.1 Antecedentes	16
2.2 Modelamiento Dinámico	17
2.3 Modelo de la Planta	18
2.4 Descripción del Sistema	18
2.5 El Modelo Dinámico	20
2.5.1 El Procedimiento Denavit-Hartengerg	21
2.5.2 El Algoritmo de Lagrange-Euler	24
2.6 Aplicación y Programación del Modelo	26

2.6.1	Cálculos	26
2.6.2	Programación del Algoritmo Lagrange-Euler	31
2.6.3	El Modelo de los Actuadores	31
2.6.4	Modelo en el Espacio Estado	36
2.7	Respuesta en Lazo Abierto	39
2.8	Definición de Términos	42
III.	VARIABLES E HIPÓTESIS	43
3.1	Definición de Variables	43
3.2	Operacionalización de Variables	43
3.3	Hipótesis General e Hipótesis Específicas	43
3.3.1	Hipótesis General	43
3.3.2	Hipótesis Específicas	44
IV.	METODOLOGÍA	45
4.1	Lógica Difusa	45
4.1.1	Conjunto Difuso	46
4.1.2	Funciones de Pertenencia	48
4.1.3	Operaciones con Conjuntos Difusos	50
4.1.4	Variables Lingüísticas	52
4.2	Sistema de Inferencia Difusa	53
4.2.1	Implicaciones Difusas	55
4.2.2	Principio de Extensión	56
4.2.3	Inferencia Difusa de Mamdani	58
4.2.4	Estructura del FIS de Mamdani	60
4.3	Sistemas de Control Difuso	61
4.3.1	Diagrama Básico de un Controlador Difuso	64
4.3.2	Edición de Reglas	65
4.4	Estructura de Control Difuso para el Robot	67
4.4.1	Controlador Difuso Puro (FPID)	67
4.5	Población y Muestra	68
4.6	Técnicas e Instrumentos de Recolección de Datos	68

4.7	Procedimientos de Recolección de Datos	68
4.8	Procedimiento Estadístico y Análisis de Datos	68
V.	RESULTADOS	69
5.1	Simulaciones Control Difuso tipo PID	69
VI.	DISCUSIÓN DE RESULTADOS	73
6.1	Simulaciones de la Cinemática Directa	73
6.2	Simulaciones del Control Difuso tipo PID	78
6.3	Técnicas de Sintonizado	81
6.4	Simulaciones del Control Difuso Tipo PID	84
6.4.1	Inferencia Difusa Articulación 1 – 2	85
6.4.2	Inferencia Difusa Articulación 3	88
VII.	CONCLUSIONES Y RECOMENDACIONES	90
7.1	Conclusiones	90
7.2	Recomendaciones	91
VIII.	BIBLIOGRAFIA	92
ANEXOS		
ANEXO 1	Códigos MATLAB	93
ANEXO 2	Matriz de Consistencia	107

ÍNDICE DE TABLAS

Tabla N ^a 2.1	Valores de los parámetros DH	27
Tabla N ^a 2.2	Parámetros de la Planta No-lineal	108
Tabla N ^a 4.1	Implicaciones difusas más usadas en control	56
Tabla N ^a 4.2	Posibles estructuras elementales de los controladores PID-difusos	63
Tabla N ^a 6.1	Valores de los parámetros del controlador PID	82

INDICE DE FIGURAS

Figura N ^a 2.1	Esquema del Robot Manipulador Esférico de 3GL	18
Figura N ^a 2.2	Esquema con las medidas del Manipulador	19
Figura N ^a 2.3	Sistema de referencias del Manipulador	20
Figura N ^a 2.4	Momento de inercia del disco respecto al sistema de referencia1 (S_1)	28
Figura N ^a 2.5	Momento de inercia del prisma respecto al sistema de referencia 1 (S_1)	29
Figura N ^a 2.6	Servomotor DC1 de la base del Manipulador	33
Figura N ^a 2.7	Servomotor DC2 de la primera varilla del Manipulador	34
Figura N ^a 2.8	Servomotor DC3 de la segunda varilla del Manipulador	35
Figura N ^a 2.9	Respuesta del manipulador a una trayectoria	40
Figura N ^a 2.10	Respuesta del manipulador a una trayectoria senoidal	41
Figura N ^a 2.11	Respuesta de salida de los actuadores	41
Figura N ^a 4.1	Definición de la temperatura calurosa según la lógica clásica (izquierda) y según la lógica difusa (derecha)	46
Figura N ^a 4.2	Conjunto difuso “A” definido dentro del universo de discurso “X”	48
Figura N ^a 4.3	Tipos de funciones de pertenencia	49
Figura N ^a 4.4	Operaciones difusas	51
Figura N ^a 4.5	Variable lingüística velocidad	53
Figura N ^a 4.6	Sistema de inferencia difusa	54
Figura N ^a 4.7	Mapeo de entrada-salida para la función parábola	58
Figura N ^a 4.8	Estructura del FIS de Mamdani	61
Figura N ^a 4.9	Sistema de control en lazo cerrado	62
Figura N ^a 4.10	Diagrama de bloques de un sistema controlado por un sistema FLC	65
Figura N ^a 4.11	Tabla de regla base	66
Figura N ^a 4.12	Conjuntos difusos definidos para las entradas y salida del Sistema FLC	66

Figura N ^a 4.13	Estructura del FLC con un PID paralelo	67
Figura N ^a 4.14	Estructura del FLC tipo PID	68
Figura N ^a 5.1	Sistema de control difuso primera articulación	69
Figura N ^a 5.2	Control de posición articulación 1	69
Figura N ^a 5.3	Torque ejercido en la articulación 1	70
Figura N ^a 5.4	Control de posición articulación 2	70
Figura N ^a 5.5	Torque ejercido en la articulación 2	71
Figura N ^a 5.6	Control de posición articulación 3	71
Figura N ^a 5.7	Torque ejercido en la articulación 3	72
Figura N ^a 6.1	Robot de 3GDL	75
Figura N ^a 6.2	Robot de 3GDL y sus posturas	77
Figura N ^a 6.3	Diagrama de bloques del Motor DC	78
Figura N ^a 6.4	Diagrama de bloques del Motor DC	79
Figura N ^a 6.5	Parámetros características del motor DC	79
Figura N ^a 6.6	Diagrama de bloques del control PID del robot de 3GDL en desacoplo	81
Figura N ^a 6.7	Parámetros de sintonía del robot 3GDL	84
Figura N ^a 6.8	Sistema de control difuso tipo PID del robot 3GDL	85
Figura N ^a 6.9	Funciones de pertenencia de entrada	86
Figura N ^a 6.10	Funciones de pertenencia de salida	86
Figura N ^a 6.11	Ventana de visualizador de reglas base	87
Figura N ^a 6.12	Funciones de pertenencia de entrada articulación 3	88
Figura N ^a 6.13	Funciones de pertenencia de salida	88
Figura N ^a 6.14	Ventana de visualizador de reglas base articulación 3	89

RESUMEN

La presente tesis trata sobre el control no lineal de sistemas multivariables, denominados MIMO (Múltiple-Input-Múltiple-Output), para sistemas manipuladores robóticos, es un campo de la Ingeniería de Control, que ha logrado su desarrollo gracias a sus variadas aplicaciones industriales y una diversidad de métodos de análisis y síntesis de los sistemas de control. Debemos de destacar, que todos los sistemas físicos del mundo real a controlar son generalmente no lineales por naturaleza y pueden ser descritos por ecuaciones diferenciales.

Los procesos multivariables, se caracterizan por presentar un comportamiento dinámico muy complejo, sin embargo, se implementan con las técnicas de control no lineal, los cuales, en los últimos años, han alcanzado un elevado nivel de aceptación industrial en control de procesos.

La técnica a emplear en el presente trabajo es la de control difuso tipo PID y se basa en modelo, es una de las técnicas de control inteligente más potentes y goza de mucha aceptación tanto en las aplicaciones industriales como en el campo académico. Por esto la técnica es una candidata natural y por ello, la presente tesis plantea un algoritmo de control basado en reglas usando la inferencia difusa de E. Mamdani.

Los controladores difusos presentan diversas topologías o estructuras de diagramas de bloques debido a que se desea es buscar la consistencia la ley de control y encontrar la mejor performance del sistema de control. Uno de los objetivos del control es obtener la regla base que es el conocimiento del sistema de control el cual posee 5 partes importantes como son: **el fuzificador, la operación difusa, la implicación, la agregación y la defuzificador.**

La aplicación para la técnica propuesta y que será explicada en esta tesis es un manipulador de tres grados de libertad (3GDL) que es un proceso inherentemente no lineal, para el cual se diseñará un controlador difuso tipo PID de 3 entradas y 3 salidas, de tal manera que permita al robot seguir en forma arbitraria las referencias, donde las entradas al sistema, son los voltajes de armadura a los servomotores DC

usados en las articulaciones y las salidas q_1 , q_2 y q_3 son las posiciones angulares del manipulador.

ABSTRAC

The present thesis deals with the non-linear control of multivariable systems, called MIMO (Multiple-Input-Multiple-Output), for robotic manipulator systems. It is a field of Control Engineering that has achieved its development thanks to its varied industrial applications and a variety of methods of analysis and synthesis of control systems. It should be noted that all real-world physical systems to be controlled are generally non-linear in nature and can be described by differential equations.

The multivariate processes are characterized by a very complex dynamic behavior, however, they are implemented with nonlinear control techniques, which, in recent years, have reached a high level of industrial acceptance in process control.

The technique to be used in the present work is the diffuse control type PID and is based on model, it is one of the most powerful intelligent control techniques and enjoys great acceptance both in industrial applications and in the academic field. This is why the technique is a natural candidate and therefore, the present thesis presents a rule-based control algorithm using the diffuse inference of E. Mamdani. The fuzzy controllers present various topologies or structures of block diagrams because it is desired to look for the consistency of the control law and find the best performance of the control system. One of the objectives of the control is to obtain the basic rule that is the knowledge of the control system which has five important parts such as the fuzzifier, the diffuse operation, the implication, the aggregation and the defuzzification.

EN LA PRESENTE TESIS, SE HA ORGANIZADO EL TRABAJO EN CINCO CAPÍTULOS DE LA SIGUIENTE FORMA:

I. Presenta el planteamiento inicial del problema, identificando y formulando el problema, los objetivos generales y el alcance del trabajo, se presenta el enunciado del problema, es decir las variables que serán controladas; así mismo las partes que componen el sistema.

II. Presenta el marco teórico en la que se indican antecedentes del problema, el estudio del modelado de la planta no lineal y se plantea las ecuaciones dinámicas mediante el desarrollo del algoritmo Lagrange-Euler en términos de energía cinética, potencial y disipativa.

III. Las variables e hipótesis que indican en presente tesis, así como la operacionalización de las variables y la formulación de la hipótesis.

IV. Presenta la metodología del trabajo, haciendo una descripción general de la técnica de control difuso tipo PID, en el cual se presenta el fundamento teórico de dicha técnica de control para procesos MISO (múltiple entrada y simple salida).

V. Se obtiene los resultados realizando mediante el programa simulink.

VI. Se ilustra los resultados obtenidos de las simulaciones hechas en MATLAB. Las simulaciones han sido realizadas para el control de consignas arbitrarias.

VII. Esta las conclusiones y recomendaciones.

VIII. Finalmente se presenta la bibliografía y los anexos

El anexo contiene los códigos del algoritmo de Lagrange-Euler y el algoritmo de control difuso tipo PID, escritos en código MATLAB y la matriz de consistencia

I.- PLANTEAMIENTO DEL PROBLEMA DE INVESTIGACION

1.1. Identificación del Problema

La Robótica es un área de investigación que se está desarrollando dentro de la Unidad de Posgrado de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional del Callao. Todo sistema que se desea controlar es necesario conocerlo, es decir, establecer las condiciones de su funcionamiento y así proponer una estrategia para modificar su accionar. Los robots manipuladores son sistemas no lineales ya que presentan de forma natural fricción, tiempos muertos o histéresis. Los manipuladores son una cadena cinemática despejada formado por un conjunto de eslabones que interrelacionan entre sí mediante articulaciones

Definición de Grados de Libertad de un Robot

Desde los cursos de Ingeniería mecánica, para el orden de la ubicación de puntos en el espacio, uno necesita especificar tres coordenadas, tal que las coordenadas x , y , z , estén a lo largo de ejes Cartesianos. Las tres coordenadas son necesarias y suficientes para definir la ubicación de un punto. También estas tres coordenadas pueden expresarse en términos de diferentes sistemas coordenados, estos son siempre necesarios. Sin embargo, no es posible tener dos o cuatro coordenadas, entonces dos es inadecuado para ubicar un punto en el espacio y cuatro es posible en tres dimensiones. Similarmente podemos considerar un mecanismo de tres dimensiones que contenga tres grados de libertad sin considerar el espacio de trabajo del mecanismo.

Juntas del Robot

Los robots tienen diferentes tipos de juntas, como la lineal, rotacional, siringa o esférica. Aunque las juntas esféricas son muy comunes en algunos sistemas, debido a que ellos tienen múltiples posiciones de grado de libertad, y por lo tanto, son difíciles de controlar, las juntas esféricas no son comunes en robótica, excepto en investigación. La mayoría de los robots tienen una unión (prismática) lineal o una articulación rotacional (revoluta).

Coordenadas del Robot

Las configuraciones de robot generalmente siguen a los marcos de coordenada con los que son definidos, como se muestra en la Figura 2. Las juntas prismáticas son denotadas por **P**, las articulaciones de revoluta son denotadas por **R**, y las articulaciones esféricas son denotadas por **S**. Las configuraciones de robot son especificadas por una sucesión de P, R, o S. Por ejemplo, un robot con tres juntas prismáticas y tres revolutas son especificados por 3P3R.

Las siguientes configuraciones son comunes para colocar la mano del robot.

Cartesiana Rectangular (3P)

Estos robots están hechos de tres uniones lineales que colocan al efector final, que es seguida generalmente por una junta revoluta adicional que orienta al efector de final.

Cilíndrica (R2P)

Los robots de coordenada cilíndricas tienen dos juntas prismáticas y una junta revoluta para posicionar la parte, adiciona la junta para orientar la parte.

Esférica (2RP)

Estos robots siguen las coordenadas esféricas del sistema, tienen una junta prismática y dos juntas revolutas para posicionar la parte, adiciona una junta revoluta para la orientación.

Brazo Robótico SCARA

Los robots SCARA tienen dos juntas revoluta y son paralelos y permiten que el robot se mueva en un plano horizontal, adicionalmente tiene una junta prismática para movimiento vertical.

El Robot Poli articulado

En este grupo de los robots de muy diversa forma de configuración se trata de brazos de robots Sedentarios que están configurados para mover su elementos terminales (pinzas de sujeción, herramientas, elementos de soldadura en un determinado lugar de trabajo y actuar sobre objetos con un plano de simetría vertical.

Es esta última de los manipuladores poli articulados la base para el modelado de sintonización de un robot puma de tres grados de libertad.

En los últimos años se ha observado una tendencia hacia el desarrollo de sistemas manipuladores robóticos académicos como son los robots Kawasaki, Mitsubishi de carga 3Kg. Estos robots de manufactura alemana y japonesa vienen con un controlador integrado y un software propietario siendo al usuario realizar algunas pruebas de algoritmo de control no lineal. En ese sentido, es una limitación implementar un algoritmo o un trabajo de reprogramación, para desarrollar nuevas tareas, sea lo menor posible. Los robots industriales son componentes importantes dentro de las celdas de manufactura flexible. Estos realizan diversas tareas tales como soldadura, posicionamiento de cargas, pintura, etc. Para realizar estas tareas eficientemente y con alta precisión muchos enfoques de control moderno han sido propuestos en los últimos años tales como control difuso, control no lineal por modos deslizantes, control adaptativo no lineal y el control predictivo no lineal entre otros.

Uno de los controladores ampliamente difundidos es el PID (Proporcional-Integral-Derivativo). Sin embargo, para propósitos de manufactura flexible el control PID no es el más adecuado debido a que un cambio en la tarea necesita nuevamente sintonizarlo (seleccionar otras ganancias proporcional, derivativa e integral).

Además, robots seriales con muchos componentes vinculados presentan una dinámica altamente no lineal y el fuerte acoplamiento entre sus articulaciones dificulta la tarea de sintonización del PID. Por consiguiente, las estrategias de control inteligente que hacen uso de la dinámica no lineal del sistema se perfilan como la mejor alternativa para alcanzar un mejor desempeño en sistemas de

manufactura flexible. La estrategia que trataremos en esta tesis es, control difuso de tipo PID, es una de las técnicas de control más potentes y goza de mucha aceptación tanto en las aplicaciones industriales como en el campo académico. Los principios de este controlador son resumidos a continuación:

1. Conocimiento basado en reglas que puede ser descrita en forma tabular y resuelta mediante el plano fase y análisis de la respuesta del sistema de control a lazo cerrado.
2. Desarrollo del algoritmo de inferencia difusa previo valor del error y cambio de error, entradas que serán procesadas por el fuzificador, luego se efectuara la operación difusa, posteriormente la implicación de Mamdani, y en la salida la agregación y finalmente la defuzificador que calcula el centro de gravedad (COA= Center Of Área) para que este valor excite a la planta y pueda realizar la tarea de control.

Todos los sistemas reales que existen en el mundo son no lineales. Sin embargo, la propiedad intrínseca que poseen hace posible de que se aproximen a ser lineales bajo ciertas condiciones. Desde el punto de vista matemático las no linealidades pueden ser continuas y discontinuas, estas últimas (como la histéresis y la fricción) no pueden ser aproximadas localmente por funciones lineales. Sin embargo, hay determinados rangos de operación del sistema en donde estas no linealidades pueden aproximarse a funciones lineales. Para que ello suceda es preferible de que las no linealidades no sean fuertes (cambios bruscos de las pendientes en la curva de sus funciones que las describen). Los rangos de operación mencionados vienen definidos por aquel conjunto de valores en las variables de estado que ocasiona que el sistema no tienda a cambiar el valor de estos mismos en el tiempo. Mediante el uso de formulaciones matemáticas o restricciones físicas podrá ser posible aproximar razonablemente el sistema no lineal a uno lineal en dicho rango de operación.

1.2. Formulación del Problema

El sistema a controlar es un manipulador robótico de 3 grados de libertad, que está compuesto de 3 eslabones conectados en serie por articulaciones rotativas, cada una de ellas conducida por un motor reductor DC con encoder óptico incremental. Los eslabones y acoplamientos del robot han sido considerados como rígidos. El problema que se estudia en la presente tesis es el control de seguimiento a una trayectoria angular para el manipulador. Esto se ha conseguido haciendo uso de la técnica de control difuso tipo PID. Las entradas y salidas del robot vienen dados por el voltaje entregado a cada motor reductor DC y por el ángulo que gira una articulación, respectivamente.

1.3. Objetivo de la Investigación

1.3.1. Objetivo general

Mantener y evaluar la performance del controlador difuso tipo PID de un robot PUMA de 3GDL. El efector final del manipulador robótico en la posición y orientación requeridas actuando sobre los servomotores y los parámetros de las articulaciones que hay entre cada uno de los eslabones que componen el manipulador robótico mediante una tensión $u(t)$ (señal de control) aplicada a cada servomotor, de ese modo los estudios de simulación demostraran que la técnica de control difuso tipo PID es una candidata idónea para que el sistema de control actúen en forma simultánea sobre las articulaciones y pueden hacer que las salidas sigan eficientemente a trayectorias de referencia arbitraria, con mínimo sobre impulso y error en estado estacionario nulo.

1.3.2. Objetivos Específicos:

- a.- Elaborar el algoritmo difuso de tipo PID utilizando la inferencia difusa de Mamdani.
- b.- Análisis de la cinemática directa, cinemática inversa y dinámica del robot PUMA de 3GDL.

- c.- Simular el controlador difuso tipo PID del robot PUMA de tres grados de libertad, con la finalidad de verificar la performance ante consignas arbitrarias.

1.4. Justificación

La ejecución del presente trabajo, es un aporte más a otros trabajos que se desarrollan en la Universidad Nacional del Callao, Unidad de Postgrado de la Facultad de Ingeniería Eléctrica y Electrónica, tiene la finalidad de realizar el desarrollo de algoritmos computacionales los mismos que pueden ser probados en prototipos (manipuladores esféricos), y en un futuro estos puedan ser parte de un desarrollo tecnológico en beneficio de aplicaciones industriales en el Perú.

Para demostrar las técnicas expuestas se ha escogido un tema de robótica que es una tecnología con más auge en la actualidad y con mayor impacto en procesos de manufactura, ya que se puede lograr mayor precisión en el control de trayectorias. De esta manera se ha escogido como aplicación del control de un robot de 3 GDL adecuado para mostrar el funcionamiento del control difuso tipo PID.

1.5. Limitaciones

No existe mucha información referente al uso de técnicas de control difuso, más aún cuando se refieren a diferentes tipos de estructuras que estos controladores pueden manejar como es el que se está usando para aplicaciones en robótica. Por otro lado, las inferencias que trata la teoría de control difuso como son de Mamdani y de Sugeno tienen sus particularidades desde el punto de vista de abordar los conceptos de cómo se emplea la lógica difusa en ambos casos, encontrando en ellas también temas como modelamiento basados en lógica difusa.

II.- MARCO TEORICO

2.1 Antecedentes

Los robots industriales son ampliamente usados para realizar varias tareas tales como, ensamblaje, soldadura o pintado. Para realizar estas tareas de manera precisa y eficiente, muchos enfoques de control han sido propuestos. Entre estos tenemos al simple Control Proporcional Derivativo con Compensación de Gravedad, propuesto por Spong M. W y Vidyasagar M. en 1989. Posteriormente, el Control por Linealización Exacta por Realimentación de Estado, ampliamente investigado por Slotine-Li, 1991, Isidori, 1995 y Khalil, 1996, permite mejorar el comportamiento de la variable controlada. La técnica de control predictivo fue introducida por Richalet en 1978 con su Control Predictivo Heurístico Basado en Modelo (Model Predictive Heuristic Control, MPHHC), más tarde conocido como Control Algorítmico Basado en Modelo (Model Algorithmic Control, MAC), por otro lado, de forma independiente: Cutler y Ramaker, ingenieros de la compañía Shell, presentaron en 1980 su Control con Matriz Dinámica (Dynamic Matrix Control, DMC). Ambos algoritmos utilizan explícitamente el modelo dinámico del proceso a controlar, esto se obtuvo de la respuesta al impulso para el primer caso y de la respuesta al escalón para el segundo, para predecir el efecto de las futuras señales de control en las variables a controlar.

La formulación del control predictivo está íntimamente ligada al problema de control óptimo y a la programación lineal. El concepto de horizonte deslizante, es una de las ideas centrales del control predictivo y fue propuesto por Propoi en 1963. El control predictivo para sistemas lineales ha recibido considerable atención en la última década debido a su robustez con respecto a un modelo con incertidumbre. En consecuencia, podemos encontrar muchos de estos algoritmos implementados en el campo industrial. Sin embargo, muchos sistemas son inherentemente no lineales. Por tanto, el uso de modelos lineales para describir la dinámica de un proceso resulta inadecuado. Ante este problema se han realizado muchos esfuerzos con el objetivo de extender la estrategia de control predictivo a sistemas no lineales, Michalska H. y Mayne D. en 1993.

La desventaja para lograr este propósito es la alta carga computacional que se requiere para solucionar el problema de optimización en cada instante de muestreo, por ello la aplicación de esta técnica al control de robots (dinámica rápida) suena muy remota. Para vencer la carga computacional, muchos algoritmos han sido propuestos Ping L. en 1995, Singh S. M. en 1995, Souroukh M.-Kravaris C. en 1996 y Chen W. en 2003, los cuales han tenido un objetivo común, derivar una ley de control fuera de línea, es decir, que no sea calculada en cada instante de muestreo. Desde el nacimiento de la lógica difusa en 1965 a partir de la publicación del artículo “Fuzzy sets”¹ escrito por el ingeniero y matemático Lotfi Zadeh, los sistemas de control han adoptado por el uso de las técnicas de control inteligente siguiendo patrones de razonamiento semejantes al pensamiento del hombre. Sin embargo, el campo de desarrollo de esta área de la inteligencia artificial ha cobrado más notoriedad en los sistemas de control, dando así origen a los sistemas de control difuso. Estos controladores inteligentes fueron aplicados en su mayoría a diversos sistemas como electrodomésticos, procesos industriales, algunas relacionadas en el área de medicina y en el ámbito espacial.

2.2 Modelamiento Dinámico

Como trabajo previo al robot de 3GDL tenemos los trabajos realizados por: Ing. Cortés Gómez Celina, Ing. Cortés Gómez Rutilo Omar Ing. Pérez Treviño Moisés, Ing. Monzón Guzmán Guillermo Dr. Mauricio C. Méndez Canseco, de la Universidad Anáhuac del Sur. México, D.F., titulado Diseño y Construcción de un Brazo Manipulador con Tres Grados de Libertad.

Basados en las referencias dadas, este trabajo presenta el modelado análisis y control de un manipulador plano subactuado con tres grados de libertad. El manipulador está formado por dos juntas rotacionales, que permite al conjunto seguir trayectorias definidas en el plano. El modelo matemático del sistema se obtiene a través de las ecuaciones de Lagrange-Euler. Se comprueba que el sistema tiene una dinámica interna estable, lo que permitirá diseñar una ley de control capaz de seguir trayectorias y atenuar perturbaciones.

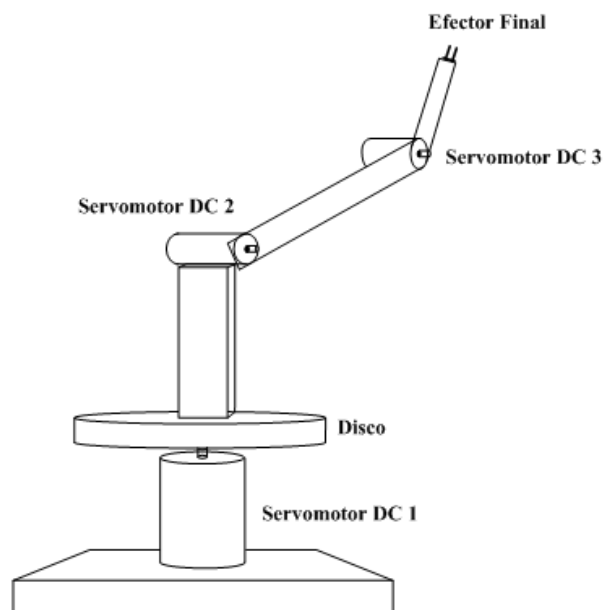
2.3 Modelado de la Planta

El modelado de la planta consiste en determinar la descripción matemática (usualmente un conjunto de ecuaciones diferenciales) del proceso a ser controlado, que en nuestro caso es el robot manipulador esférico. Tal modelado describe la dinámica del proceso y tiene por objetivo conocer la relación entre el movimiento del robot y las fuerzas implicadas en el mismo. Para determinar las ecuaciones del modelo usaremos el algoritmo de Lagrange-Euler.

2.4 Descripción del Sistema

El sistema robot de 3GDL es mostrado en la Figura 2.1.

Figura 2.1
Esquema del Robot Manipulador Esférico de 3GL



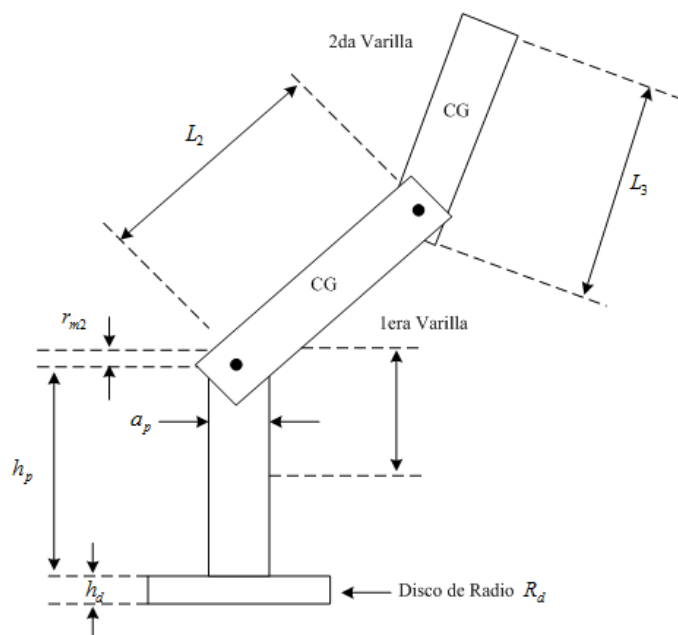
Fuente propia del Autor

A continuación, se describe los tres grados de libertad usados en el proceso:

1. Primer Grado de Libertad: Está compuesto por un servomotor DC1 de 24 VDC con encoder óptico incorporado, y una base conformado por un disco y un prisma de aluminio; de tal manera que en la parte inferior del disco de radio R_d y altura h_d , irá acoplado con una bocina de bronce dicho servomotor en posición vertical y sobre la parte superior del disco irá un prisma de sección cuadrada a_p , y de altura h_p , a su vez en la parte superior de dicho prisma irá el otro servomotor en posición horizontal que corresponde al segundo grado de libertad. En la Figura 2.2 se muestra una vista del esquema con las medidas del SRM de 3GL.

Figura 2.2

Esquema con las medidas del Manipulador



Fuente propia del Autor

2. Segundo Grado de Libertad: compuesto por un servomotor DC2 a 24 VDC con encoder óptico incorporado, y una varilla rígida de aluminio, de tal manera que en el eje de dicho servomotor, en posición horizontal, irá acoplado con una bocina de bronce dicha varilla de longitud L_2 , en el extremo de dicha varilla irá el otro servomotor en posición horizontal que corresponde al tercer grado de libertad.

3. Tercer Grado de Libertad: compuesto por otro servomotor DC3 a 24 VDC con encoder óptico incorporado, y una varilla rígida de aluminio, de tal manera que en el eje de dicho servomotor, en posición horizontal, irá acoplado con una bocina de bronce dicha varilla de longitud L_3 , en el extremo de dicha varilla podrá ir un efector final

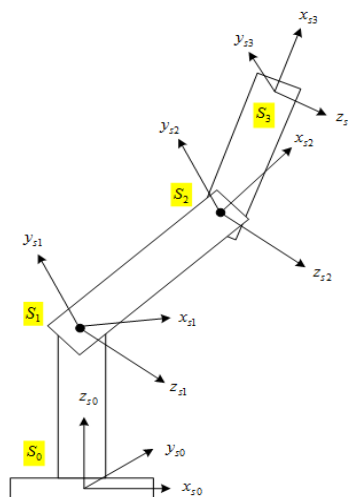
2.5. El Modelo Dinámico

Para determinar la ecuación dinámica del manipulador existen varios métodos tales como el método de Newton-Euler, el método de Lagrange-Euler, entre otros. Para el manipulador en estudio usaremos el método de Lagrange-Euler que emplea la fórmula Lagrangiana conjuntamente con la representación de Denavit-Hartenberg (D-H).

En la Figura 2.3 se muestra los sistemas de referencias usados para la representación D-H.

Figura 2.3

Sistemas de referencias del Manipulador



Fuente propia del Autor

S_0 : Sistema de referencia base (eje de motor 1)

S_1 : Sistema de referencia 1 (eje de motor 2)

S_2 : Sistema de referencia 2 (eje de motor 3)

S_3 : Sistema de referencia 3

q_1, q_2, q_3 : Ángulos de las articulaciones

2.5.1 El Procedimiento Denavit-Hartenberg

Un manipulador robótico consiste de una secuencia de cuerpos rígidos (los eslabones) articulados por juntas rotacionales o prismáticas. Cada par de articulación-eslabón representa un grado de libertad (GL). Un sistema de coordenadas ortonormales $(x_{s_i}, y_{s_i}, z_{s_i})$ puede ser establecido usando la regla de la mano derecha para cada articulación i , donde $i = 1, \dots, n$. Así un manipulador con n GL posee n pares de articulaciones. El último sistema de coordenadas $(x_{s_n}, y_{s_n}, z_{s_n})$ constituye el sistema de coordenadas del efector final, el cual es usualmente articulado al último eslabón. El manipulador es unido a una base de referencia 0. El sistema de coordenada base es definido como $(x_{s_0}, y_{s_0}, z_{s_0})$. Las coordenadas de la base son también las coordenadas inerciales del manipulador. El sistema de coordenadas base aumentado es definido como: $(x_{s_0}, y_{s_0}, z_{s_0}, 0)$.

El procedimiento D-H origina una matriz de transformación homogénea que representa el sistema inercial del manipulador. Para describir tal matriz, requerimos establecer el sistema de coordenadas $(x_{s_i}, y_{s_i}, z_{s_i})$ del manipulador en $i = 0, \dots, n$, para determinar los parámetros de las articulaciones y eslabones con el fin de desarrollar la matriz de transferencia homogénea.

Estableciendo el Sistema de Coordenadas D-H

Para establecer el sistema de coordenadas D-H $(x_{s_i}, y_{s_i}, z_{s_i})$ $i = 0, \dots, n$, se usa las siguientes reglas:

1. El sistema de coordenadas base $(x_{s_0}, y_{s_0}, z_{s_0})$ se determina usando la regla de la mano derecha. El eje z_{s_0} se escoge libremente y está localizado a lo largo del eje del movimiento del primer eslabón.
2. El eje $z_{s_{i-1}}$ es alineado con el eje del movimiento (giratorio o prismático) de la i – ésima articulación.
3. El origen del sistema de coordenadas s_i – ésima está en la intersección del eje z_{s_i} y $z_{s_{i-1}}$ o en la intersección de la normal común entre los ejes de dos articulaciones consecutivas z_{s_i} y $z_{s_{i-1}}$ y el eje z_{s_i} .
4. El eje x_{s_i} es perpendicular al eje $z_{s_{i-1}}$.
5. Aplicar la regla de la mano derecha en la articulación i – ésima para determinar y_{s_i} .
6. Generalmente, la articulación n – ésima es una articulación giratoria. El sistema de coordenadas $(x_{s_n}, y_{s_n}, z_{s_n})$, puede estar en cualquier parte del efector final con tal de que el eje z_{s_n} esté a lo largo de la dirección del eje $z_{s_{i-1}}$ y apuntando hacia afuera del manipulador. El eje x_{s_n} es perpendicular a ambos ejes: z_{s_n} y $z_{s_{n-1}}$

Parámetros D-H

Los cuatro parámetros geométricos del manipulador $(\theta_i, d_{DH_i}, a_{DH_i}$ y $\alpha_i)$, se asocian con cada par articulación-eslabón y son requeridos para la descripción de una articulación giratoria o prismática. Después de establecer el sistema de coordenadas del manipulador tales parámetros pueden ser determinados de la siguiente manera:

1. θ_i , es el ángulo de rotación del eje $x_{s_{i-1}}$ con el eje x_{s_i} en el plano del eje $z_{s_{i-1}}$. Este es variable si la articulación es giratoria.
2. d_{DH_i} , es la distancia del origen del sistema de coordenadas $s_{(i-1)}$ – ésima a lo largo del eje $z_{s_{i-1}}$, hasta la intersección del eje $z_{s_{i-1}}$ con el eje x_{s_i} . Esta distancia d_{DH_i} es variable si la articulación i es prismática.

3. a_{DH_i} , es la distancia a lo largo del eje x_{s_i} , que va desde la intersección del eje $z_{s_{i-1}}$ con el eje x_{s_i} hasta el origen del sistema de coordenadas s_i – *ésima* en el caso de articulaciones giratorias.
4. α_i , es el ángulo de separación entre los ejes $z_{s_{i-1}}$ y z_{s_i} medido en un plano perpendicular al eje x_{s_i} , utilizando la regla de la mano derecha.

Matriz de Transformación Homogénea D-H

La matriz de transformación homogénea D-H caracteriza cada sistema de coordenadas del par articulación-eslabón con respecto al sistema de coordenadas previo. Por consiguiente un punto r_{s_i} del sistema de coordenadas s_i – *ésimo* se puede expresar como $(r_{s_{i-1}})$ en el sistema de coordenadas $s_{(i-1)}$ – *ésimo* ejecutando sucesivamente las siguientes operaciones:

1. $T(z_s, \theta_i)$: Rotación de un ángulo θ_i alrededor del eje $z_{s_{i-1}}$, para alinear el eje $x_{s_{i-1}}$ con el eje x_{s_i} .
2. $T(z_s, d_{DH_i})$: Translación a lo largo del eje $z_{s_{i-1}}$ de una distancia d_{DH_i} , hasta hacer coincidir el eje $z_{s_{i-1}}$ con el eje x_{s_i} .
3. $T(x_s, a_{DH_i})$: Translación a lo largo del eje x_{s_i} una distancia a_{DH_i} , que va desde la intersección del eje $z_{s_{i-1}}$ y el eje x_{s_i} al origen del sistema de coordenadas s_i – *ésimo*.
4. $T(x_s, \alpha_i)$: Rotación de un ángulo α_i , del eje $z_{s_{i-1}}$ con el eje z_{s_i} alrededor del eje x_{s_i} .

El producto de estas cuatro operaciones básicas produce la matriz de transformación homogénea A_i^{i-1} , del eslabón i con respecto al eslabón $i - 1$ o articulación i con respecto a la articulación $i - 1$. Por lo tanto:

$$A_i^{i-1} = T(z, \theta_i) T(0,0, d_{DH_i}) T(a_{DH_i}, 0,0) T(x, \alpha_i) =$$

$$\begin{bmatrix} c(\theta_i) & -s(\theta_i) & 0 & 0 \\ s(\theta_i) & c(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_{DH_i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_{DH_i} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} c(\theta_i) & -c(\alpha_i)s(\theta_i) & s(\alpha_i)s(\theta_i) & a_{DH_i}c(\theta_i) \\ s(\theta_i) & c(\alpha_i)c(\theta_i) & -s(\alpha_i)c(\theta_i) & a_{DH_i}s(\theta_i) \\ 0 & s(\alpha_i) & c(\alpha_i) & d_{DH_i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Donde c y s , son las funciones \cos y \sin respectivamente. La siguiente matriz de transformación homogénea:

$$T_i^0 = \prod_{j=1}^i A_j^{j-1} = A_1^0 A_2^1 \dots A_i^{i-1} \quad i = 1, 2, \dots, n \quad (2.2)$$

Determina la localización del sistema de coordenada (s_i) –ésima con respecto al sistema de coordenada base

2.5.2. El Algoritmo de Lagrange-Euler

El procedimiento de Lagrange-Euler requiere ejecutar los siguientes pasos:

1. Asignar a cada par articulación-eslabón del manipulador un sistema de coordenadas D-H.
2. Obtener las matrices de transformación homogénea A_i^{i-1} para $i = 1, \dots, n$.
3. Obtener las matrices U_{ij} que es el efecto del movimiento de la articulación j en todos los puntos del eslabón i :

$$U_{ij} = \frac{\partial A_i^0}{\partial q_j} = \begin{cases} A_{j-1}^0 Q_j A_i^{j-1} & \text{si } j \leq i \\ 0 & \text{si } j > i \end{cases} \quad (2.3)$$

Donde A_{j-1}^0 relaciona al sistema de coordenadas $(j-1)$ –ésimo con el sistema de coordenadas base, y Q_j toma los siguientes valores:

$$Q_j = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{si la articulación es giratoria}$$

$$Q_j = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{si la articulación es prismática}$$

4. Los efectos de interacción entre las articulaciones son expresadas por las siguientes matrices U_{ijk} :

$$U_{ijk} = \frac{\partial U_{ij}}{\partial q_k} = \frac{\partial}{\partial q_k} \left(\frac{\partial A_i^0}{\partial q_j} \right) = \begin{cases} A_{j-1}^0 Q_j A_{k-1}^{j-1} Q_k A_i^{k-1} & \text{si } i \geq k \geq j \\ A_{k-1}^0 Q_k A_{j-1}^{k-1} Q_j A_i^{j-1} & \text{si } i \geq j \geq k \\ 0 & \text{si } k > i \text{ ó } j > i \end{cases} \quad (2.4)$$

5. Obtener el tensor de inercia J_i del eslabón i – ésimo mediante:

$$J_i = \begin{bmatrix} \frac{-I_{xx_{s_i}} + I_{yy_{s_i}} + I_{zz_{s_i}}}{2} & -I_{x_{s_i}y_{s_i}} & -I_{x_{s_i}z_{s_i}} & m_i \bar{x}_{s_i} \\ -I_{x_{s_i}y_{s_i}} & \frac{-I_{xx_{s_i}} + I_{yy_{s_i}} + I_{zz_{s_i}}}{2} & -I_{y_{s_i}z_{s_i}} & m_i \bar{y}_{s_i} \\ -I_{x_{s_i}z_{s_i}} & -I_{y_{s_i}z_{s_i}} & \frac{-I_{xx_{s_i}} + I_{yy_{s_i}} + I_{zz_{s_i}}}{2} & m_i \bar{z}_{s_i} \\ m_i \bar{x}_{s_i} & m_i \bar{y}_{s_i} & m_i \bar{z}_{s_i} & m_i \end{bmatrix} \quad (2.5)$$

Dónde: $(\bar{x}_{s_i}, \bar{y}_{s_i}, \bar{z}_{s_i})$ es el centro de gravedad (CG) del eslabón i con respecto al sistema de coordenadas s_i – ésimo, $I_{xx_{s_i}}, I_{yy_{s_i}}, I_{zz_{s_i}}$ son los momentos de inercia con respecto al sistema de coordenadas $(x_{s_i}, y_{s_i}, z_{s_i})$, m_i es la masa del cuerpo i , $\bar{x}_{s_i}, \bar{y}_{s_i}, \bar{z}_{s_i}$ son las distancias del CG del cuerpo i al sistema de coordenadas $(x_{s_i}, y_{s_i}, z_{s_i})$, e $I_{x_{s_i}y_{s_i}}, I_{y_{s_i}z_{s_i}}, I_{x_{s_i}z_{s_i}}$ son los productos correspondientes del momento de inercia.

6. Obtener la matriz H de inercia simétrica de dimensión $n \times n$, la cual está relacionada con la aceleración y cuyos elementos se expresan como:

$$H_{ik} = \sum_{j=\max(i,k)}^n \text{Tr}(U_{jk} J_j U_{ji}^T) \quad i, k = 1, 2 \dots n \quad (2.6)$$

Donde, Tr es la traza y n es el número de grados de libertad.

7. Obtener los términos h_{ikm} definidos por:

$$h_{ikm} = \sum_{j=\max(i,k,m)}^n \text{Tr}(U_{jkm} J_j U_{ji}^T) \quad i, k, m = 1, 2 \dots n \quad (2.7)$$

8. Obtener el vector no lineal fuerza centrífuga y Coriolis de orden $n \times 1$ cuyos elementos C_i son dados por:

$$C_i = \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \dot{q}_k \dot{q}_m \quad (2.8)$$

9. Obtener el vector no lineal de fuerza de gravedad d de orden $n \times 1$ cuyos elementos son:

$$d_i = \sum_{j=1}^n (-m_j U_{ji} r_{s_j}^{-j}) \quad i = 1, 2 \dots n \quad (2.9)$$

Donde, $g = [g_{x_{s_0}} \ g_{y_{s_0}} \ g_{z_{s_0}} \ 0]$ es el vector gravedad fila expresado en el sistema de coordenadas base y $\bar{r}_{s_j}^j = [\bar{x}_{s_j} \ \bar{y}_{s_j} \ \bar{z}_{s_j} \ 1]^T$ (el elemento 1 es un factor de escala) es el vector centro de masa del eslabón j y expresado en el sistema de coordenadas s_j -ésimo.

Finalmente, la ecuación dinámica del manipulador, con el método Lagrange-Euler toma la forma:

$$T_i = H_{ik} \ddot{q}_i + C_i + d_i \quad (2.10)$$

Donde, T_i es el vector de fuerzas y torques aplicado a cada coordenada generalizada q_i .

2.6. Aplicación y Programación del Modelo

La aplicación del trabajo es el robot manipulador esférico de tres grados de libertad. Para determinar la ecuación dinámica de dicho manipulador se aplica el algoritmo de LaGrange-Euler.

A continuación, se presenta los siguientes cálculos que usaremos para la programación con matemática simbólica del Toolbox de MATLAB.

2.6.1. Cálculos

Cálculo de los Parámetros D-H

Primeramente, se fijan los sistemas de referencia en las articulaciones como se muestra en la Figura 2.3, posteriormente se determina los parámetros D-H, los cuales se muestran en la Tabla 2.1:

Tabla 2.1

Valores de los parámetros DH

Juntura	θ_i	d_{DH_i}	a_{DH_i}	α_i
1	q_1	$h_p + h_d + r_{m2}$	0	90^0
2	q_2	0	L_2	0^0
3	q_3	0	L_3	0^0

Cálculo de las Matrices de Transformación Homogénea

Reemplazando los valores de la Tabla 2.1 en la ecuación (2.1), obtenemos las matrices de transformación homogénea A_1^0 , A_2^1 y A_3^2 :

A_1^0 : Posición y orientación del sistema de referencia 1 (s_1) de la base con respecto al sistema de referencia base (s_0):

$$A_1^0 = \begin{bmatrix} \cos(q_1) & 0 & \sin(q_1) & 0 \\ \sin(q_1) & 0 & -\cos(q_1) & 0 \\ 0 & 1 & 0 & h_p + h_d + r_{m2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A_2^1 : Posición y orientación del sistema de referencia 2 (s_2) de la primera varilla con respecto al sistema de referencia 1 (s_1):

$$A_2^1 = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & L_2 \cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & L_2 \sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A_3^2 : Posición y orientación del sistema de referencia 3 (s_3) de la segunda varilla con respecto al sistema de referencia 2 (s_2):

$$A_3^2 = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & L_3 \cos(q_3) \\ \sin(q_3) & \cos(q_3) & 0 & L_3 \sin(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A_2^0 : Posición y orientación del sistema de referencia 2 (s_2) de la primera varilla con respecto al sistema de referencia base (s_0):

$$A_2^0 = A_1^0 A_2^1$$

A_3^0 : Posición y orientación del sistema de referencia 3 (s_3) de la segunda varilla con respecto al sistema de referencia base (s_0):

$$A_3^0 = A_1^0 A_2^1 A_3^2$$

A_2^0 y A_3^0 se determina mediante matemática simbólica de MATLAB, y el programa se muestra en el apéndice A.

Cálculo del Momento de Inercia del Disco

El momento de inercia del disco respecto al sistema de referencia 1 (s_1) $x_{s_1}, y_{s_1}, z_{s_1}$, (ver Figura 2.4), está dado por:

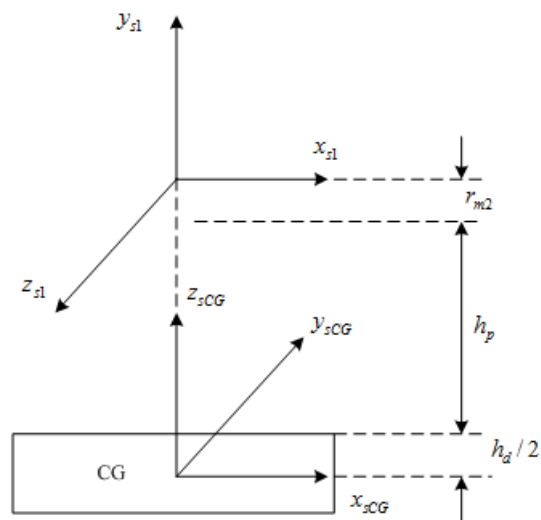
$$\begin{aligned} I_{x_{s_1d}} &= 0 \\ I_{y_{s_1d}} &= \frac{m_d R_d^2}{2} \\ I_{z_{s_1d}} &= 0 \end{aligned}$$

Y los productos de los momentos de inercia son:

$$I_{x_{s_1d}y_{s_1d}} = I_{x_{s_1d}z_{s_1d}} = I_{y_{s_1d}z_{s_1d}} = 0$$

Figura 2.4

Momento de inercia del disco respecto al sistema de referencia 1 (s_1).



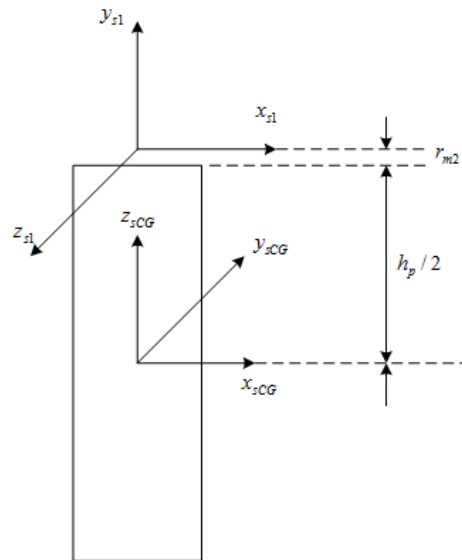
Fuente propia del Autor

Cálculo del momento de inercia del prisma

El momento de inercia del prisma respecto al sistema de referencia 1 (s_1) $x_{s_1}, y_{s_1}, z_{s_1}$, (ver Figura 2.5), está dado por:

Figura 2.5

Momento de inercia del prisma respecto al sistema de referencia 1 (s_1)



Fuente propia del Autor

$$I_{x_{s_1p}} = 0$$

$$I_{y_{s_1p}} = \frac{m_p a_p^2}{6}$$

$$I_{z_{s_1p}} = 0$$

Y los productos de los momentos de inercia son:

$$I_{x_{s_1d}y_{s_1d}} = I_{x_{s_1d}z_{s_1d}} = I_{y_{s_1d}z_{s_1d}} = 0$$

Cálculo del Momento de Inercia de la Base (Disco más Prisma)

El momento de inercia de la base es la suma de los momentos de inercia del disco más el prisma, y está dado por:

$$I_{x_{s_1}} = 0$$

$$I_{y_{s_1}} = \frac{m_p a_p^2}{6} + \frac{m_d R_d^2}{2}$$

$$I_{z_{s_1}} = 0$$

Cálculo de la Matriz de la Base J_1

De la ecuación (2.5) se obtiene la matriz de pseudoinercia de la base J_1 :

$$J_1 = \begin{bmatrix} I_1 & 0 & 0 & 0 \\ 0 & -I_1 & 0 & -m_1 L_{y_{s_1}} \\ 0 & 0 & I_1 & 0 \\ 0 & -m_1 L_{y_{s_1}} & 0 & m_1 \end{bmatrix} \quad (2.11)$$

Donde, $L_{y_{s_1}}$ es la longitud del eje de referencia 1 (s_1) al *C.G.* de la base, m_1 es igual a $m_d + m_p + m_{servo\ de\ 1} + m_{accesorios}$ y la inercia viene dada por: $I_1 = \frac{I_{y_{s_1}}}{2}$

Cálculo de la Matriz de Pseudoinercia de la Primera Varilla J_2

El momento de inercia de la primera varilla I_2 , se obtiene de [3], [6] $I_{xx_{s_2}} = 0$, y $I_2 = I_{yy_{s_2}} = I_{zz_{s_2}} = m_2 L_2^2 / 3$. Reemplazando estos valores en la ecuación (2.5), obtenemos la matriz de pseudoinercia J_2 :

$$J_2 = \begin{bmatrix} I_2 & 0 & 0 & -m_2 L_{x_{s_2}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -m_2 L_{x_{s_2}} & 0 & 0 & m_2 \end{bmatrix} \quad (2.12)$$

Donde, $L_{x_{s_2}}$ es la longitud del eje de referencia 1 (s_1) al *C.G.* de la primera varilla y m_2 es igual a $m_{brazo} + m_{objeto}$.

Cálculo de la Matriz de Pseudoinercia de la Segunda Varilla J_3 .

El momento de inercia de la segunda varilla I_3 , se obtiene de [3], [6]: $I_{xx_{s_3}} = 0$ y $I_3 = I_{yy_{s_3}} = I_{zz_{s_3}} = m_3 L_3^2 / 3$. Reemplazando estos valores en la ecuación (3.5), obtenemos la matriz de pseudoinercia J_3 :

$$J_3 = \begin{bmatrix} I_3 & 0 & 0 & -m_3 L_{x_{s_3}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -m_3 L_{x_{s_3}} & 0 & 0 & m_3 \end{bmatrix} \quad (2.13)$$

Donde, $L_{x_{s_3}}$ es la longitud del eje de referencia 2 (s_2) al C. G. de la segunda varilla y m_3 es igual a $m_{brazo} + m_{objeto}$.

2.6.2. Programación del Algoritmo Lagrange-Euler

El programa de cálculo de la ecuación dinámica del robot manipulador esférico de tres grados de libertad (3 GDL) se determina aplicando el Toolbox de Matemática Simbólica de MATLAB, [9] y se presenta en el apéndice A. La ecuación dinámica obtenida para los tres grados de libertad es:

- Primer grado de libertad:

$$T_1 = H_{11}\ddot{q}_1 + C_1 \quad (2.14)$$

- Segundo grado de libertad:

$$T_2 = H_{22}\ddot{q}_2 + H_{23}\ddot{q}_3 + C_2 + d_2 \quad (2.15)$$

- Tercer grado de libertad:

$$T_3 = H_{32}\ddot{q}_2 + H_{33}\ddot{q}_3 + C_3 + d_3 \quad (2.16)$$

Que son los elementos del torque T_i , dado en la ecuación (2.10).

2.6.3. El Modelo de los Actuadores

1. Modelado del Actuador 1 de la Base del Manipulador

En la Figura 2.6, se muestra el servomotor DC1, correspondiente a la base.

El voltaje de entrada $K_{A1}u_1$ aplicado a la armadura está dado por:

$$V_{A1} = K_{A1}u_1 = L_{a1}\dot{I}_{a1} + R_{a1}I_{a1} + V_{b1} \quad (2.17)$$

La fuerza electromotriz del servomotor DC es:

$$V_{b1} = K_{b1}\omega_{m1} = K_{b1}n_1\dot{q}_1 \quad (2.18)$$

En el cual n_1 es la relación de piñones. El torque T_{m1} del servomotor está dado por:

$$T_{m1} = K_{m1}I_{a1} = J_{m1}n_1\ddot{q}_1 + B_{m1}n_1\dot{q}_1 + T_{g1} \quad (2.19)$$

Por lo tanto la ecuación del torque T_{g2} requerido por el actuador para la base tiene la forma:

$$T_{g2} = n_1T_{g1} = J_{g1}\ddot{q}_1 + B_{g1}\dot{q}_1 + T_1 + n_1T_{c1} \quad (2.20)$$

Donde T_1 es el torque de la carga y T_{c1} es el torque de fricción de Coulomb.

Reemplazando (2.20) en (2.19) resulta:

$$n_1K_{m1}I_{a1} = J_{eq1}\ddot{q}_1 + B_{eq1}\dot{q}_1 + T_1 + n_1T_{c1} \quad (2.21)$$

Dónde:

$$J_{eq1} = n_1^2J_{m1} + J_{g1}$$

$$B_{eq1} = n_1^2B_{m1} + B_{g1}$$

$$T_1 = H_{11}\ddot{q}_1 + C_1$$

$$H_{11} = a_1 + a_2\cos^2(q_2) + a_3\cos^2(q_2 + q_3) + 2a_4\cos(q_2)\cos(q_2 + q_3) + a_5\cos^2(q_2)$$

$$C_1 = -a_2\sin(2q_2)\dot{q}_1\dot{q}_2 - 2(a_3\sin(q_2 + q_3)\cos(q_2 + q_3) - a_4\sin(q_2 + q_3)\cos(q_2))\dot{q}_1\dot{q}_3$$

$$a_1 = 2I_1$$

$$a_2 = I_2 + 2m_2L_2L_{cs2} - m_2L_2^2$$

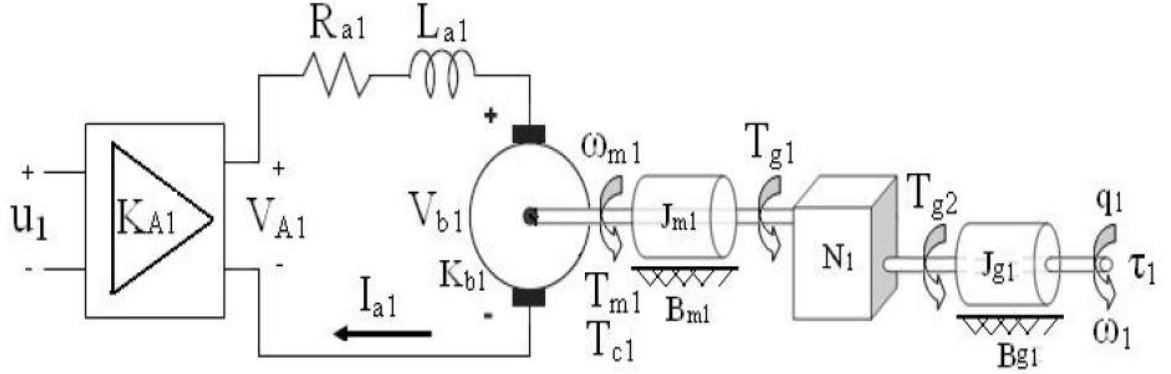
$$a_3 = I_3 + 2m_3L_3L_{cs3} - m_3L_3^2$$

$$a_4 = m_3L_2L_{cs3}$$

$$a_5 = m_3L_2^2$$

Figura 2.6

Servomotor DC1 de la base del Manipulador



Fuente propia del Autor

Despreciando la inductancia de armadura o sea L_{a1} , obtenemos de (2.17) y (2.18)

$$K_{A1}u_1 = R_{a1}I_{a1} + K_{b1}n_1\dot{q}_1 \quad (2.22)$$

Despejando I_{a1} en (2.21) y reemplazando en (2.22), obtenemos la ley de control para la primera entrada:

$$u_1 = R_{T1}(J_{eq1}\ddot{q}_1 + B_{eq1}\dot{q}_1 + T_1) + N_{T1}\dot{q}_1 + n_1R_{T1}T_{c1} \quad (2.23)$$

Dónde:

$$R_{T1} = \frac{R_{a1}}{n_1K_{m1}K_{A1}}$$

$$N_{T1} = \frac{n_1K_{b1}}{K_{A1}}$$

$$T_{c1} = C_{c1}\text{sign}(\dot{q}_1)$$

2. Modelado del Actuador 2 de la Primera Varilla del Manipulador

En la Figura 2.7, se muestra el servomotor DC2 correspondiente a la primera varilla. En la misma Figura se muestra el circuito eléctrico del servomotor DC2, controlado también por la armadura. Dicho servomotor se encuentra ubicado en la parte superior del prisma. Para obtener la ley de control para la segunda entrada se

procede de la misma manera que para el primer servomotor, obteniendo de esta manera:

$$u_2 = R_{T2}(J_{eq2}\ddot{q}_2 + B_{eq2}\dot{q}_2 + T_2) + N_{T2}\dot{q}_2 + n_2R_{T2}T_{c2} \quad (2.24)$$

Dónde:

$$T_{c2} = C_{c1}sign(\dot{q}_2)$$

$$J_{eq2} = n_2^2J_{m2} + J_{g2}$$

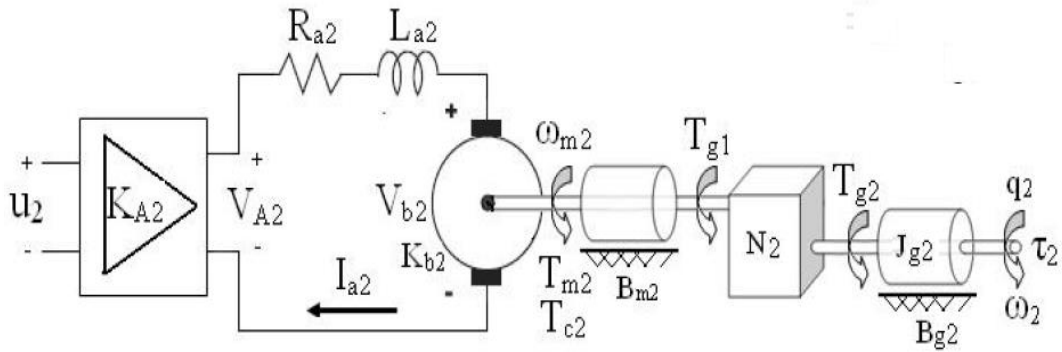
$$B_{eq2} = n_2^2B_{m2} + B_{g2}$$

$$R_{T2} = \frac{R_{a2}}{n_2K_{m2}K_{A2}}$$

$$N_{T2} = \frac{n_2K_{b1}}{K_{A2}}$$

Figura 2.7

Servomotor DC2 de la primera varilla del Manipulador



Fuente propia del Autor

$$T_2 = H_{22}\ddot{q}_2 + H_{23}\ddot{q}_3 + C_2 + d_2$$

$$H_{22} = a_2 + a_3 + 2a_4\cos(q_3) + a_5$$

$$H_{23} = a_3 + a_4\cos(q_3)$$

$$C_2 = a_2\sin(q_2)\cos(q_2)\dot{q}_1^2 - 2a_4\sin(q_3)\dot{q}_2\dot{q}_3 - a_4\sin(q_3)\dot{q}_3^2$$

$$d_2 = a_6\cos(q_2) + a_7\cos(q_2 + q_3)$$

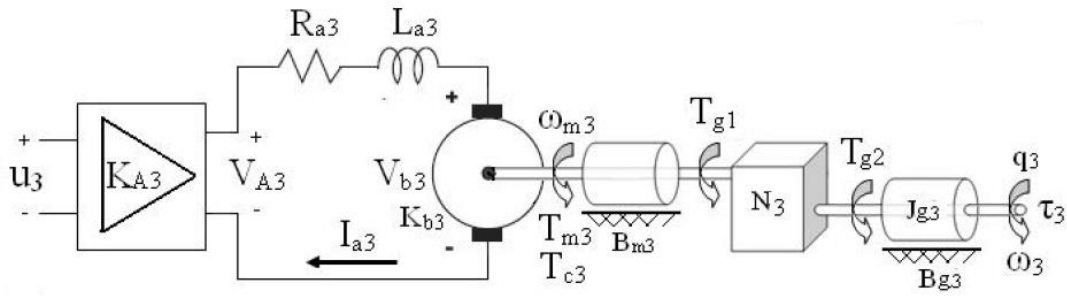
$$a_6 = m_2gL_{cs_2} + m_3gL_2$$

$$a_7 = m_3gL_{cs_3}$$

3. Modelado del Actuador 3 de la Segunda Varilla del Manipulador

En la Figura 2.8, se muestra el servomotor DC3 correspondiente a la segunda varilla. En la misma Figura se muestra el circuito eléctrico del servomotor DC3, controlado también por la armadura. Dicho servomotor se encuentra ubicado en extremo de la primera varilla. Para obtener la ley de control para la tercera entrada se procede de la misma manera que para el primer servomotor, obteniendo de esta manera:

Figura 2.8: Servomotor DC3 de la segunda varilla del Manipulador.



Fuente propia del Autor

Obtenemos la señal de control para el servomotor DC3.

$$u_3 = R_{T3}(J_{eq3}\ddot{q}_3 + B_{eq3}\dot{q}_3 + T_3) + N_{T3}\dot{q}_3 + n_3R_{T3}T_{c3} \quad (2.25)$$

Dónde:

$$T_{c3} = C_{c3}\text{sign}(\dot{q}_3)$$

$$J_{eq3} = n_3^2J_{m3} + J_{g3}$$

$$B_{eq3} = n_3^2B_{m3} + B_{g3}$$

$$R_{T3} = \frac{R_{a3}}{n_3K_{m3}K_{A3}}$$

$$N_{T3} = \frac{n_3K_{b3}}{K_{A3}}$$

$$T_3 = H_{32}\ddot{q}_2 + H_{33}\ddot{q}_3 + C_3 + d_3$$

$$H_{32} = a_3 + a_4\cos(q_3)$$

$$H_{33} = a_3$$

$$C_3 = (a_3\sin(q_2 + q_3)\cos(q_2 + q_3) + a_4\sin(q_2 + q_3)\cos(q_2))\dot{q}_1^2 + a_4\sin(q_3)\dot{q}_2^2$$

$$d_3 = a_7\cos(q_2 + q_3)$$

2.6.4. Modelo en el Espacio Estado

La ecuación dinámica del manipulador obtenida en las ecuaciones (2.14), (2.15) y (2.16), se representa en el espacio de estados mediante la siguiente asignación de variables:

$$x_1 = q_1 \quad (2.26)$$

$$x_2 = q_2 \quad (2.27)$$

$$x_3 = q_3 \quad (2.28)$$

$$x_4 = \dot{q}_1 \quad (2.29)$$

$$x_5 = \dot{q}_2 \quad (2.30)$$

$$x_6 = \dot{q}_3 \quad (2.31)$$

Dónde:

x_1 = Posición angular de la base

x_2 = Posición angular de la primera varilla

x_3 = Posición angular de la segunda varilla

x_4 = Velocidad angular de la base

x_5 = Velocidad angular de la primera varilla

x_6 = Velocidad angular de la segunda varilla

La ecuación no lineal para sistemas multivariables es:

$$\dot{x} = f(x) + G(x)u \quad (2.32)$$

Su representación en forma matricial:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ f_4(x) \\ f_5(x) \\ f_6(x) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ g_{41}(x) & 0 & 0 \\ 0 & g_{52}(x) & g_{53}(x) \\ 0 & g_{62}(x) & g_{63}(x) \end{bmatrix} \begin{bmatrix} u_1(x) \\ u_2(x) \\ u_3(x) \end{bmatrix} \quad (2.33)$$

Reemplazando las ecuaciones (2.26) y (2.29) en la ecuación (2.23) se obtiene u_1 en variables de estado:

$$u_1 = R_{T1}(J_{eq1}\dot{x}_4 + B_{eq1}x_4 + T_1) + N_{T1}x_4 + C_{c1}n_1R_{T1}sign(x_4) \quad (2.34)$$

Así mismo, reemplazando las ecuaciones (2.27) y (2.30) en la ecuación (2.24) se obtiene u_2 en variables de estado:

$$u_2 = R_{T2}(J_{eq2}\dot{x}_5 + B_{eq2}x_5 + T_2) + N_{T2}x_5 + C_{c1}n_2R_{T2}sign(x_5) \quad (2.35)$$

De la misma manera, reemplazando las ecuaciones (2.28) y (2.31) en la ecuación (2.25) se obtiene u_3 en variables de estado:

$$u_3 = R_{T3}(J_{eq3}\dot{x}_6 + B_{eq3}x_6 + T_3) + N_{T3}x_6 + C_{c3}n_3R_{T3}sign(x_6) \quad (2.36)$$

La ecuación dinámica del manipulador obtenida en las ecuaciones (2.14), (2.15) y (2.16), también se representa en variables de estado:

$$T_1 = H_{11}\dot{x}_4 + C_1 \quad (2.37)$$

$$T_2 = H_{22}\dot{x}_5 + H_{23}\dot{x}_6 + C_2 + d_2 \quad (2.38)$$

$$T_3 = H_{32}\dot{x}_5 + H_{33}\dot{x}_6 + C_3 + d_3 \quad (2.39)$$

Posteriormente, se reemplaza la ecuación (2.37) en (2.34) y se obtiene la primera entrada u_1 :

$$u_1 = R_{T1}(J_{eq1}\dot{x}_4 + B_{eq1}x_4 + H_{11}\dot{x}_4 + C_1) + N_{T1}x_4 + C_{c1}n_1R_{T1}sign(x_4) \quad (2.40)$$

Luego, se reemplaza la ecuación (2.38) en (2.35) y se obtiene la segunda entrada u_2 :

$$u_2 = R_{T2}(J_{eq2}\dot{x}_5 + B_{eq2}x_5 + H_{22}\dot{x}_5 + H_{23}\dot{x}_6 + C_2 + d_2) + N_{T2}x_5 + C_{c1}n_2R_{T2}sign(x_5) \quad (2.41)$$

Finalmente, se reemplaza la ecuación (2.39) en (3.36) y se obtiene la tercera entrada u_3 :

$$u_3 = R_{T3}(J_{eq3}\dot{x}_6 + B_{eq3}x_6 + H_{32}\dot{x}_5 + H_{33}\dot{x}_6 + C_3 + d_3) + N_{T3}x_6 + C_{c3}n_3R_{T3}sign(x_6) \quad (2.42)$$

Así mismo de (2.40) se obtiene:

$$\dot{x}_4 = \frac{u_1 - R_{T1}B_{eq1}x_4 - N_{T1}x_4 - R_{T1}C_1 - C_{c1}n_1R_{T1}sign(x_4)}{R_{T1}(J_{eq1} + H_{11})} \quad (2.43)$$

de (2.41) y (2.42) se obtiene \dot{x}_5 y \dot{x}_6 :

$$\dot{x}_5 = \frac{R_{T3}(J_{eq3} + H_{33})u_2 - R_{T2}H_{23}u_3 - R_{T3}(J_{eq3} + H_{33})Q_2 + R_{T2}H_{23}Q_3}{Det} \quad (2.44)$$

$$\dot{x}_6 = \frac{-R_{T3}H_{32}u_2 + R_{T2}(J_{eq2} + H_{22})u_3 + R_{T3}H_{32}Q_2 - R_{T2}(J_{eq2} + H_{22})Q_3}{Det} \quad (2.45)$$

Dónde:

$$Q_2 = (R_{T2}B_{eq2} + N_{T2} \pm n_2^2 R_{T2}C_{c1})x_5 + R_{T2}(C_2 + d_2)$$

$$Q_3 = (R_{T3}B_{eq3} + N_{T3} \pm n_3^2 R_{T3}C_{c3})x_6 + R_{T3}(C_3 + d_3)$$

$$Det = R_{T2}R_{T3}(J_{eq2} + H_{22})(J_{eq3} + H_{33}) - R_{T2}R_{T3}H_{23}^2$$

Finalmente reemplazando las ecuaciones (2.29), (2.30), (2.31), (2.43), (2.44) y (2.45), en la ecuación (2.33) se obtiene:

$$f_1(x) = x_4$$

$$f_2(x) = x_5$$

$$f_3(x) = x_6$$

$$f_4(x) = \frac{-(R_{T1}B_{eq1} + N_{T1} \pm C_{c1}n_1^2 R_{T1})x_4 - R_{T1}C_1}{R_{T1}(J_{eq1} + H_{11})}$$

$$f_5(x) = \frac{-R_{T3}(J_{eq3} + H_{33})Q_2 + R_{T2}H_{23}Q_3}{Det}$$

$$f_6 = \frac{R_{T3}H_{32}Q_2 - R_{T2}(J_{eq2} + H_{22})Q_3}{Det}$$

$$g_{41}(x) = \frac{1}{R_{T1}(J_{eq1} + H_{11})}$$

$$g_{52}(x) = \frac{R_{T3}(J_{eq3} + H_{33})}{Det}$$

$$g_{53} = \frac{-R_{T2}H_{23}}{Det}$$

$$g_{62}(x) = \frac{-R_{T2}H_{32}}{Det}$$

$$g_{63}(x) = \frac{R_{T2}(J_{eq2} + H_{22})}{Det}$$

La ecuación salida viene representada por:

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \quad (2.46)$$

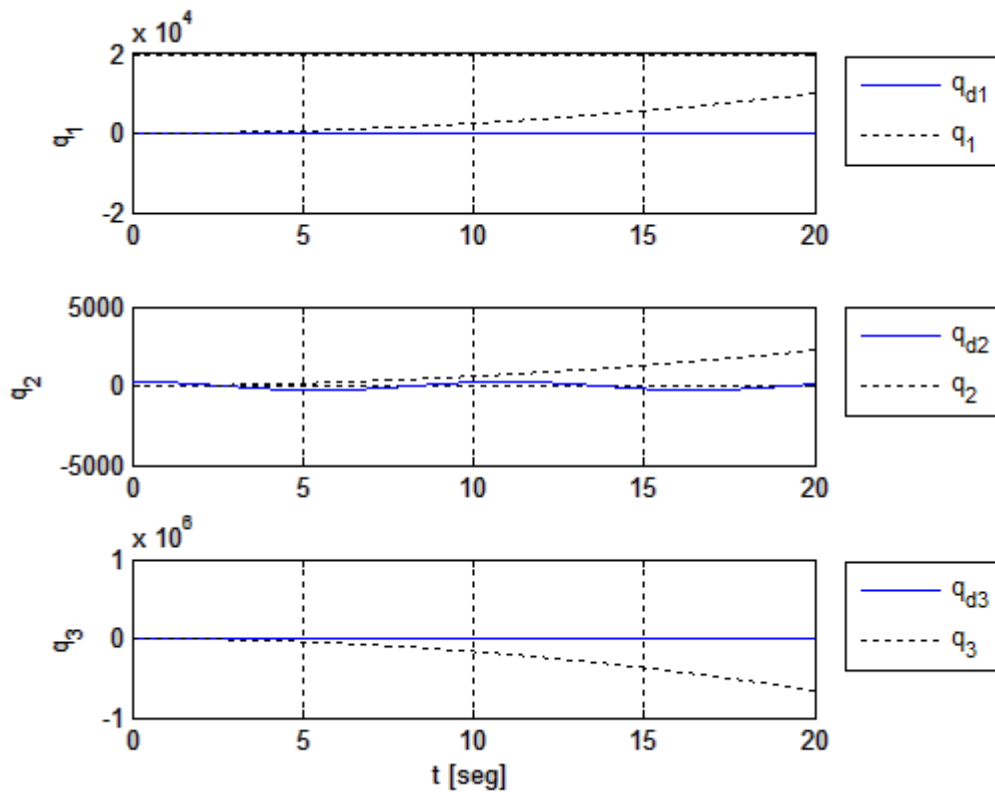
2.7. Respuesta en Lazo Abierto

La simulación del sistema manipulador de 3GL en lazo abierto, contribuye a analizar la estabilidad es más fácil de lograr puesto que no constituye un problema importante. En cambio en los sistemas de lazo cerrado, la estabilidad si es un problema importante, por su tendencia a sobre corregir errores que pueden producir oscilaciones de amplitud constante o variable. Hay que puntualizar que para sistemas cuyas entradas son conocidas previamente y en los que no hay la presencia de perturbaciones, es recomendable utilizar el control de lazo abierto. Los sistemas de control de lazo cerrado tienen ventajas solamente si se presentan perturbaciones no previsible o variaciones de componentes del sistema.

Respuesta en Lazo abierto del manipulador robótico a una trayectoria constante

La figura 2.9 describe el giro angular del brazo descrito por la variable q_i donde $i = 1,2,3$. Se presenta una entrada escalón unitario aplicada en tiempo cero. Las simulaciones se han llevado a cabo para las tres articulaciones por columna, en donde en orden de recuadro de arriba hacia abajo podemos ver la representación de la variable posición q , velocidad angular \dot{q} y aceleración angular \ddot{q} .

Figura 2.9 Respuesta del manipulador a una trayectoria.

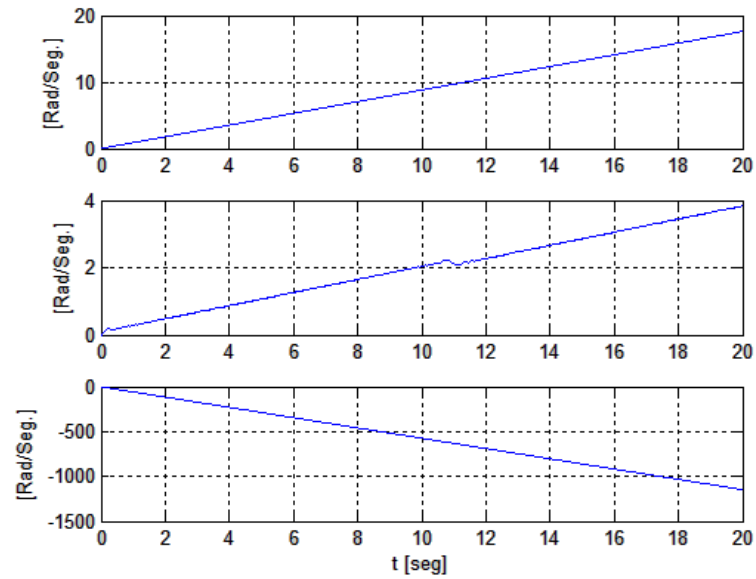


Fuente Autoría propia

Respuesta en Lazo abierto del manipulador robótico a una trayectoria senoidal

La figura 2.10 describe la velocidad angular de las articulaciones, en este caso se ha sometido al sistema a una entrada senoidal. Debemos destacar que las representaciones de las velocidades del manipulador 3GL vienen dado por la velocidad angular de la base, $x_4 = \dot{q}_1$ la velocidad angular de la primera varilla, $x_5 = \dot{q}_2$ y la velocidad angular de la segunda varilla $x_6 = \dot{q}_3$.

Figura 2.10: Respuesta del manipulador a una trayectoria senoidal

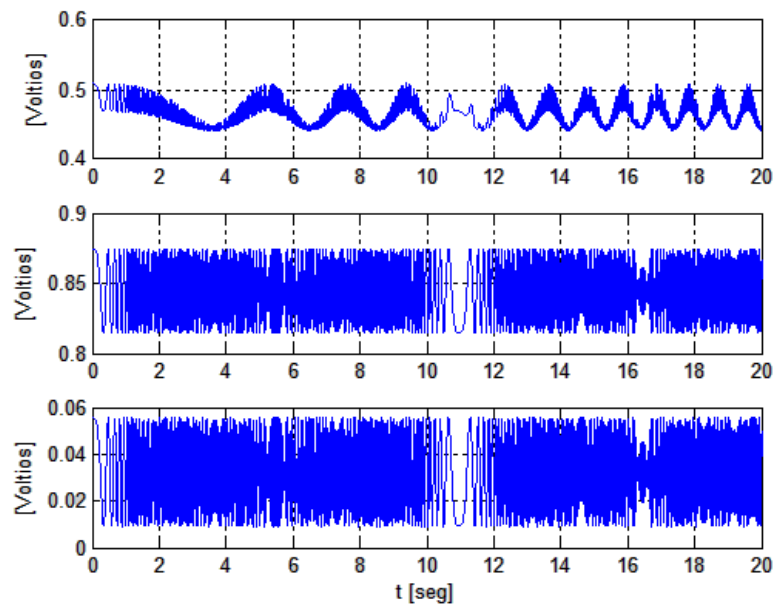


Fuente Autoría propia

Respuesta en la salida del actuador

La figura 2.11 describe la salida de los actuadores sin que estén controlados. Observamos que hay mucha oscilación auto sostenida cuando se le da una entrada senoidal al sistema de control.

Figura 2.11: Respuesta a la salida de los actuadores.



Fuente Autoría propia.

2.8 Definición de Términos

Algoritmo de control Abstracción de lógica de un sistema distribuido, se habla de un conjunto de procesos y de líneas de comunicación virtuales, con la finalidad de realizar una acción de control

Articulaciones Las articulaciones permiten el movimiento relativo entre los sucesivos eslabones

FIS	Sistema de Inferencia Difusa
FLC	(Fuzzy Logic Control) lógica de control difuso
FPID	Controlador Difuso Puro
Gdl	Grados de Libertad
DH	Denavit-Hartenberg
DMC	Control con Matriz Dinámica (Dynamic Matrix Control,).
MAC	Modelo Algorítmico Control
MIMO	Múltiple Input Múltiplo Output (Múltiples Entradas Múltiples Salidas)
MISO	Múltiple Input Single Output (Múltiples Entradas Simple Salida)
PD	Proporcional Derivativo
PID	Proporcional Integral Derivativo
PUMA	Programmable Universal Manipulator for Assembly (Manipulador Universal para Ensamble)
MPHC	Modelo (Model Predictive Heuristic Control)

III.- VARIABLES E HIPÓTESIS

3.1. Definición de Variables

Las variables del presente Tesis de investigación son:

- a. Variable dependiente → Sistema de control de robot tipo PUMA
- b. Variable independiente: Y → Lógica difusa

3.2. Operacionalización de las Variables

Para demostrar y comprobar la hipótesis, la operacionalizamos, obteniéndose las variables y los indicadores que a continuación se indican:

a.- Variable X → Sistema de control del robot tipo PUMA

Indicadores:

- Kp del controlador del robot tipo PUMA → X1
- Ti del controlador del robot tipo PUMA → X2
- Td del controlador del robot tipo PUMA → X3

b.- Variable Y → Lógica difusa

Indicadores:

- Conjunto difusos → Y1
- Función de pertenencia → Y2
- Reglas heurísticas → Y3

3.3. Formulación de la Hipótesis

3.3.1. Hipótesis General:

En función de las interrogantes planteadas del problema, así como de los objetivos generales y específicos que se persigue, el siguiente trabajo de tesis plantea la siguiente hipótesis:

¿Es posible realizar el diseño del controlador difuso tipo PID de un Robot Tipo PUMA de 3 grados libertad?

3.3.2. Hipótesis Específicas:

- a.- Es posible la elaboración del algoritmo de inferencia difusa de Mamdani.
- b.- Es posible la sintonización del controlador PID del robot tipo PUMA de tres grados de libertad utilizando lógica difusa.
- c.- Es posible la simulación del controlador del robot PUMA de tres grados de libertad, con la finalidad de verificar la performance de control difuso tipo PID.

IV.- METODOLOGIA

Control Difuso PID

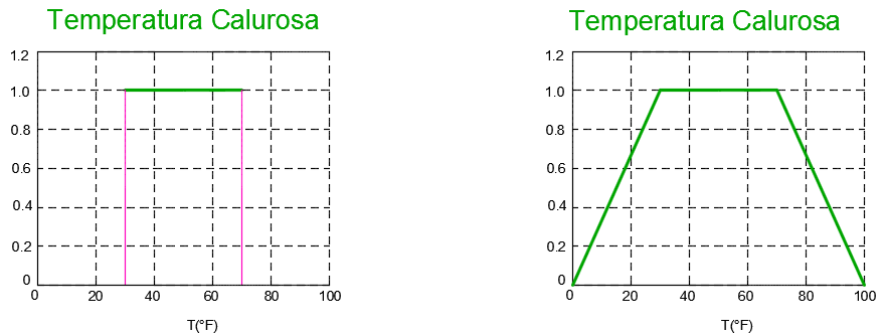
4.1. Lógica Difusa

La lógica difusa es una rama de la inteligencia artificial así como una extensión de la lógica tradicional (Booleana), que permite a un computador analizar la información del mundo real en una escala entre lo “verdadero” y lo “falso”. Este concepto fue introducido en 1965 como una generalización de la lógica clásica por Lotfi Zadeh para la revista “Information and control”. La lógica difusa está asociada con la manera en que las personas perciben su entorno, pues a diferencia de la lógica tradicional que emplea conceptos absolutos para referirse a la realidad, la lógica difusa define los conceptos en grados variables de pertenencia utilizando patrones de razonamiento similares a los del pensamiento humano.

Por ejemplo, el definir la temperatura calurosa de una habitación puede ser definido por la lógica clásica dentro de un intervalo de $[30\ 70]$ °F, por lo que una temperatura de 29.9999 °F se puede considerar como frío o 70.1111 °F se puede considerar como muy caluroso; es decir el concepto de temperatura no tiene límites naturales definidos. Por otro lado, la lógica difusa, empleando la interpretación humana, puede establecer una transición gradual entre lo “caluroso” y “no caluroso”, otorgando un cierto grado de pertenencia en términos de un conjunto difuso a esta temperatura calurosa representándola en una escala multievaluada entre lo verdadero y lo falso o en el intervalo de 0 a 1 (Ver Figura 2.1).

Figura 4.1

Definición de la temperatura calurosa según la Lógica clásica (izquierda) y según la lógica difusa (derecha).



Fuente propia del Autor

La lógica difusa puede aplicarse a diversas áreas siendo una de ellas en los sistemas de control. Estos controladores inteligentes tienen la ventaja de poder estabilizar sistemas no lineales y trabajar en procesos complejos permitiendo llevar a cabo estrategias de control usados por operadores humanos, es decir usan las reglas del experto humano de la forma IF-THEN.

4.1.1 Conjunto difuso.

Un conjunto difuso se define como una colección de elementos que existen dentro de un universo donde cada uno de sus elementos puede o no pertenecer a este conjunto en cierto grado. Este grado de pertenencia es definido mediante una función llamada función de pertenencia. A diferencia de los conjuntos clásicos, donde sus elementos pertenecen o no a dicho conjunto y no se definen situaciones intermedias, en los conjuntos difusos un elemento puede pertenecer parcialmente a más de un conjunto.

La función de pertenencia asociada a los conjuntos clásicos establece un grado de pertenencia de 0 o 1 en sus elementos, mientras que en los conjuntos difusos pueden tener cualquier valor entre 0 y 1. Además, en los conjuntos difusos un elemento puede pertenecer parcialmente a un conjunto difuso y simultáneamente puede

pertenecer parcialmente al complemento de este conjunto difuso, lo cual no es posible en los conjuntos clásicos.

Un conjunto difuso en universo de discurso U es establecido por una función de pertenencia $\mu_A(x)$ que puede tomar todos los valores reales comprendidos en un rango de 0 a 1. El universo de discurso es el rango de valores en los cuales se encuentra definido dicho conjunto difuso y viene delimitado por los valores físicos de la variable que lo describe. Un conjunto difuso A en U puede ser definido por un conjunto de pares ordenados de elementos x de la siguiente manera:

$$A = \{(x, \mu_A(x)) / x \in U\} \quad (4.1)$$

Donde $\mu_A(x)$ es el grado de pertenencia en el universo de discurso U . Si el universo de discurso U es continuo, el conjunto difuso puede denotarse por:

$$A = \int_U (\mu_A(x)/x) \quad (4.2)$$

Donde el símbolo de integración no denota la operación del mismo, sino la colección de todos los puntos $x \in U$, las cuales tienen asociado una función de pertenencia $\mu_A(x)$. Si el universo de discurso U es discreto, el conjunto difuso A se escribe según la siguiente notación:

$$A = \sum_i \mu_A(x)/x \quad (4.3)$$

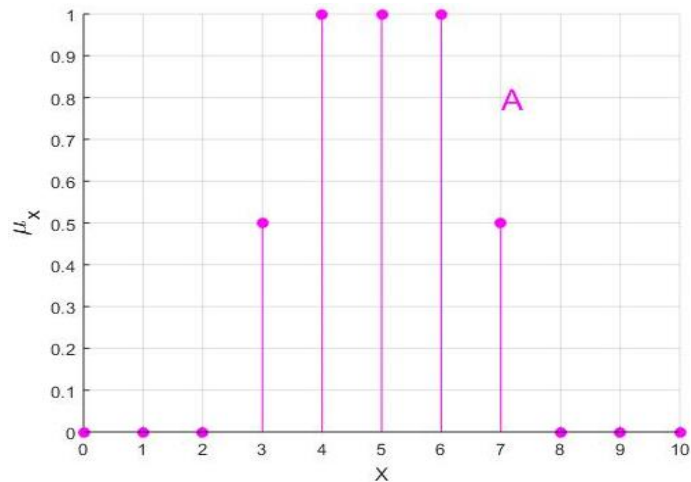
En dicha notación el símbolo de sumatoria no representa la operación de la misma, sino la colección de todos los puntos $x \in U$, las cuales tienen asociado una función de pertenencia $\mu_A(x)$. Además la expresión $\mu_A(x)/x$ representa a que el conjunto difuso A contiene exactamente un elemento x (parcial) con grado de pertenencia $\mu_A(x)$. Un ejemplo de cómo está relacionada la notación de un conjunto difuso con su respectiva gráfica, puede ser definiendo el conjunto difuso discreto A en el universo de discurso $X = [0 10]$ como:

$$A = \frac{0}{0} + \frac{0}{1} + \frac{0}{2} + \frac{0.5}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{0.5}{7} + \frac{0}{8} + \frac{0}{9} + \frac{0}{10} \quad (4.4)$$

En la notación anterior del conjunto difuso A se expresa la función de pertenencia de cada elemento perteneciente al universo de discurso cuyos pares ordenados representan el grafico de la Figura 4.2

Figura 4.2

Conjunto difuso “A” definido dentro del universo de discurso “X”.



Fuente propia del Autor

4.1.2 Funciones de pertenencia.

Las funciones de pertenencia o membresía representan el grado de pertenencia de un elemento a un conjunto difuso, el cual tiene un valor en un rango de 0 a 1. El empleo de una función de pertenencia dependerá del contexto de su aplicación. Por lo general, es de preferencia utilizar funciones de pertenencia simples, debido a la facilidad de computación, simplifican muchos cálculos y no pierden la exactitud; pues precisamente definen al conjunto difuso asociado (definen un concepto difuso). Existen muchas funciones de pertenencia, de las cuales las más comunes son las de tipo triangular, trapezoidal, gaussianas y sigmoideal definidas por las ecuaciones 4.5, 4.6, 4.7 y 4.8 respectivamente y son graficadas en la Figura 4.3.

$$\text{triangulo}(x: a, b, c) = \begin{cases} 0 & x < a \\ (x - a)/(b - a) & a \leq x < b \\ (c - x)/(c - b) & b \leq x < c \\ 0 & x \geq c \end{cases} \quad (4.5)$$

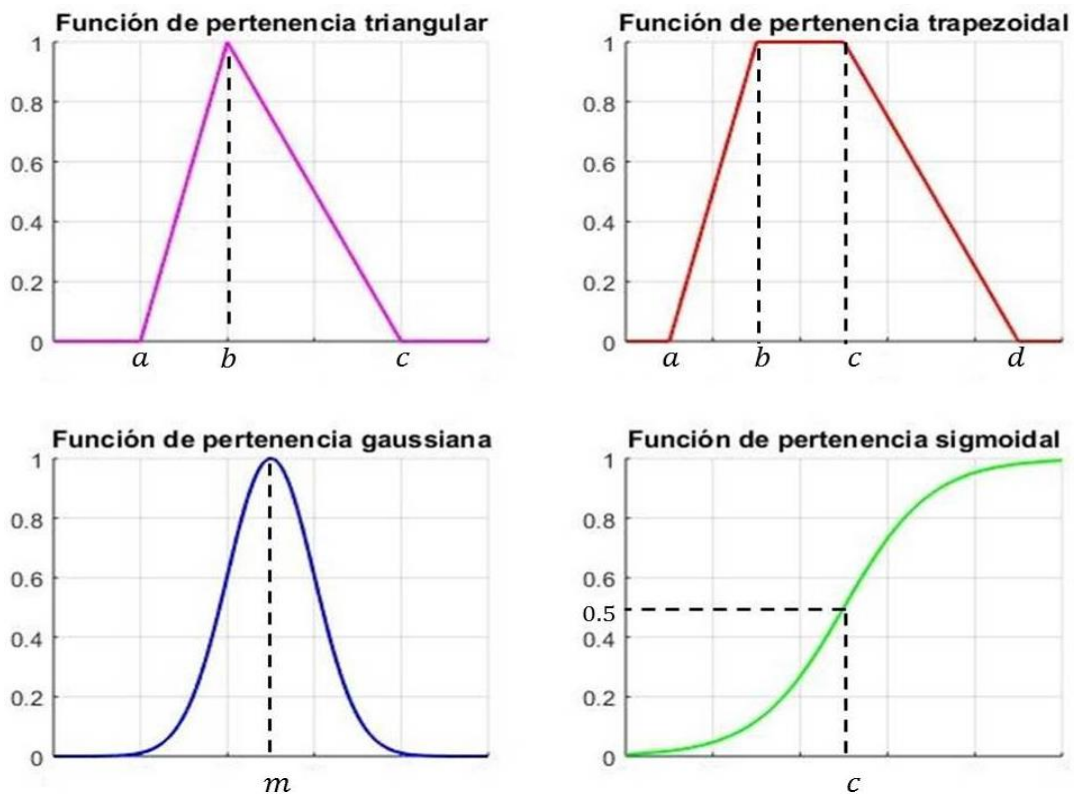
$$\text{trapecio}(x: a, b, c, d) = \begin{cases} 0 & x < a \\ (x - a)/(b - a) & a \leq x < b \\ 1 & b \leq x < c \\ (d - x)/(d - c) & c \leq x < d \\ 0 & x > d \end{cases} \quad (4.6)$$

$$\text{gaussiana}(x: m, \sigma) = e^{-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2} \quad (4.7)$$

$$\text{sigmoidal}(x: a, c) = \frac{1}{1 + e^{-a(x-c)}} \quad (4.8)$$

Figura 4.3

Tipos de funciones de pertenencia



Fuente propia del Autor

4.1.3 Operaciones con conjuntos difusos.

De forma similar que las operaciones en los conjuntos clásicos, en los conjuntos difusos existen las mismas operaciones de igualdad, contención, complemento, unión e intersección de conjuntos. Sin embargo, a causa de estas nuevas características definidas en su campo, la formulación de estas operaciones es diferente. Para dos conjuntos difusos A y B definidos en un mismo universo de discurso U, las operaciones mencionadas anteriormente pueden definirse de la siguiente manera:

- **Igualdad:** Los conjuntos difusos A y B serán igual si y solo si:

$$\mu_A(x) = \mu_B(x) \quad \forall x \in U \quad (4.9)$$

- **Contención:** Se dice que B contiene a A ($A \subset B$), si y solo si:

$$\mu_A(x) \leq \mu_B(x) \quad \forall x \in U \quad (4.10)$$

- **Complemento:** Se denota al complemento del conjunto A como \bar{A} definido también en el universo de discurso U, cuya función de pertenencia es definida por:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \Rightarrow \mu_A(x) \quad (4.11)$$

- **Unión:** La unión o disyunción de dos conjuntos difusos A y B definidos en U, es otro conjunto difuso definido en el mismo universo de discurso cuya función de pertenencia es dada por:

$$\mu_{A \cup B}(x) = \text{máx}[\mu_A(x), \mu_B(x)] \quad (4.12)$$

- **Intersección:** La intersección o conjunción de dos conjuntos difusos A y B definidos en U, es otro conjunto difuso definido en el mismo universo de discurso cuya función de pertenencia es dada por:

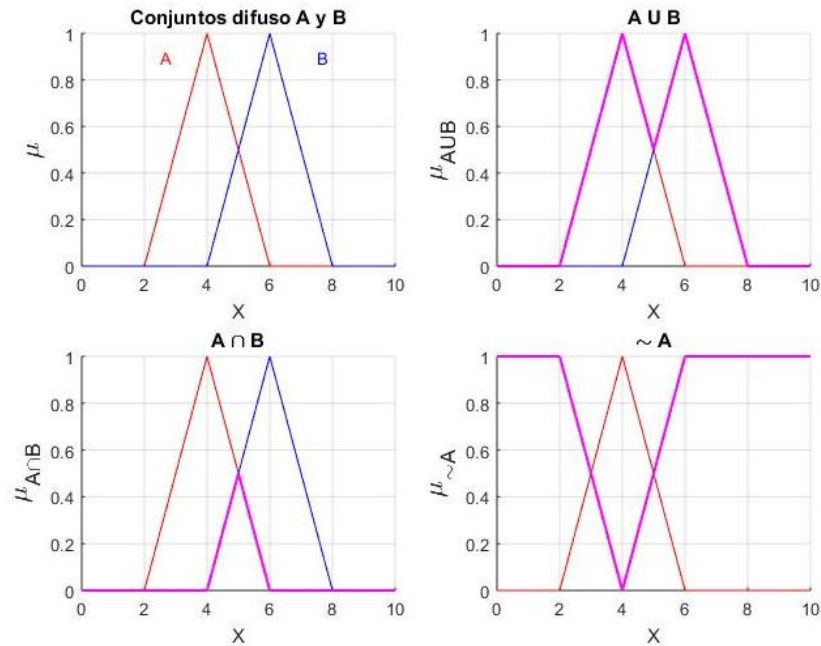
$$\mu_{A \cap B}(x) = \text{min}[\mu_A(x), \mu_B(x)] \quad (4.13)$$

Gráficamente se puede observar las tres últimas operaciones en la Figura 4.4, donde para dos conjuntos difusos de tipo triangular A y B definidos en el universo de

discurso [0 10] se realizan dichas operaciones obteniendo los conjuntos difusos remarcados.

Figura 4.4

Operaciones difusas



Fuente propia del Autor

El uso del operador “máx.” para la operación de la unión se justifica de manera intuitiva. La unión de dos conjuntos difusos A y B es aquel conjunto más pequeño que puede contener a ambos conjuntos, de tal manera de que si C es cualquier conjunto difuso que contiene a A y B, también debe contener a la unión de A y B. Bajo la premisa de (4.12), $A \cup B$ contiene a ambos conjuntos difusos A y B debido a que $\text{máx}[\mu_A(x), \mu_B(x)] \geq \mu_A(x)$ y $\text{máx}[\mu_A(x), \mu_B(x)] \geq \mu_B(x)$. Además, si C contiene a A y B, entonces se cumple por definición que $\mu_C(x) \geq \mu_A(x)$ y $\mu_C(x) \geq \mu_B(x)$. Finalmente debido a que C contiene a la unión de A y B, se cumple que $\mu_C(x) \geq \text{máx}[\mu_A(x), \mu_B(x)] = \mu_{A \cup B}(x)$ lo que ocasiona que $A \cup B$ sea definida por (4.12) que es un conjunto difuso pequeño que contiene a A y B. El operador “min” para la intersección se puede demostrar de la misma manera.

4.1.4 Variables lingüísticas.

Una variable lingüística se define como una variable cuyos valores son palabras o sentencias (no son números) para describir determinado estado de un fenómeno u objeto. Las variables lingüísticas son importantes porque son los elementos fundamentales de la representación del conocimiento humano. Estas variables lingüísticas necesitan un nombre y un valor lingüístico sobre un universo de discurso definido, las cuales darán lugar a sentencias generadas por reglas semánticas posteriormente. En la experiencia diaria, se usan palabras para describir variables, por ejemplo al decir “la velocidad del carro es alta” se utiliza la palabra alta, para describir la velocidad actual, lo cual quiere decir que la variable velocidad actual toma la palabra “alta” como su valor, el cual puede tomar algún número por ejemplo 80 km/h. Si dicha variable toma números como sus valores, se tiene una descripción bien formulada matemáticamente, sin embargo, si dicha variable toma palabras como sus valores no se tiene una descripción formal matemáticamente. Es por ello que si una variable puede tomar palabras en lenguaje natural como sus valores, esta es llamada variable lingüística.

Según **Zadeh** (1975) una variable lingüística viene definida por: $VL \Leftrightarrow (X, T, U, M)$, donde:

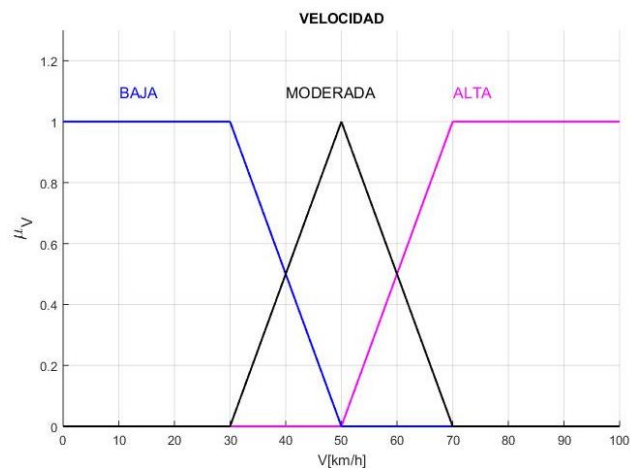
- X es el nombre de la variable lingüística, X puede ser la velocidad de un carro.
- T es el conjunto de valores lingüísticos que X puede tomar, por ejemplo, para la velocidad de un carro $T = \{\text{Bajo, Moderado, Alto}\}$.
- U es el dominio físico actual en la cual la variable lingüística X toma valores numéricos, para el ejemplo para la velocidad sería: $U = [0, V_{max}]$.
- M es una regla semántica que relaciona cada valor lingüístico en T con un conjunto difuso en U. Por ejemplo, M=si la velocidad es Alta entonces Aplicar el freno.

Para el ejemplo de un automóvil que viaja a lo largo de una trayectoria recta, este puede ir en una velocidad definida dentro del universo de discurso de $[0\ 100]$ km/h. Además, se define al conjunto difuso velocidad como $T = \{\text{Baja, Moderada, Alta}\}$

y al conjunto de las variables lingüísticas X como “Velocidad Baja alrededor de 30 km/h”, “Velocidad Moderada alrededor de 50 km/h” y “Velocidad Alta alrededor de 70 km/h”. Las reglas semánticas M son asociadas mediante implicaciones difusas analizando el grado de certeza. De esta manera, la variable lingüística velocidad puede ser graficada en la Figura 4.5.

Figura 4.5

Variable lingüística velocidad



Fuente propia del Autor

4.2 Sistemas de inferencia difusa

Un sistema se define tradicionalmente como un conjunto de elementos en donde cada uno de ellos se relaciona entre sí para realizar una función determinada. Los sistemas difusos son aquellos, que bajo la definición anterior, están fundamentados en el conocimiento de un experto humano. La parte fundamental de un sistema difuso consiste de reglas difusas de la siguiente forma:

IF antecedente THEN consecuente

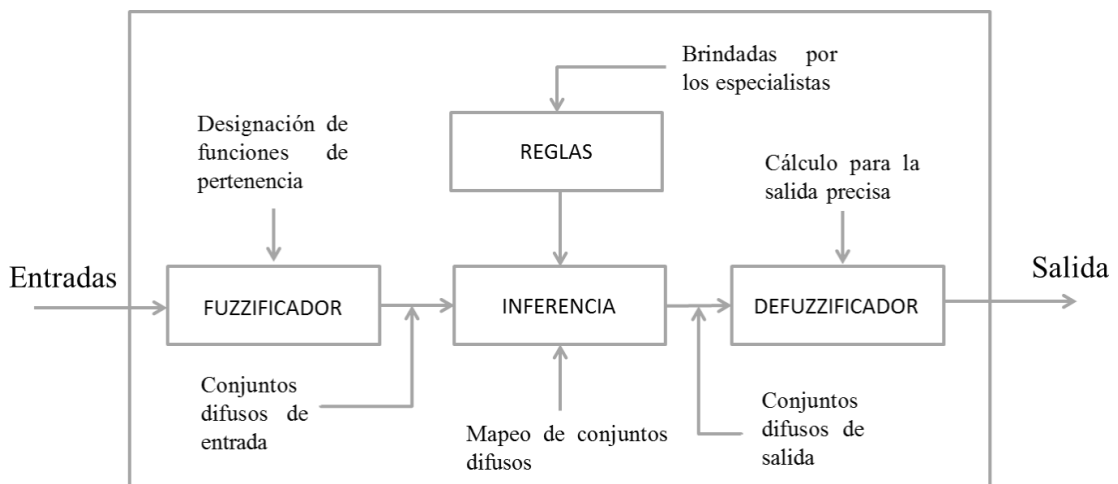
La primera parte de la regla denominada antecedente o premisa contiene una o más condiciones relacionadas a las entradas del sistema, mientras que la segunda parte

denominada consecuente es la acción o función de salida que ejecutará el sistema. Estas reglas forman lo que se denomina base de reglas del sistema, donde tanto el antecedente como el consecuente tienen asociado por cada variable lingüística un conjunto difuso que describe un determinado estado para dicha variable. Al utilizar la regla difusa del caso de la velocidad de un automóvil, esta puede ser “IF la velocidad del auto es alta, THEN aplicar menos fuerza al acelerador” donde las variables lingüísticas son la velocidad del auto y la fuerza del acelerador y los conjuntos difusos asociados a ellos son “alta” y “menos” respectivamente. Este conjunto de reglas difusas de la forma IF-THEN de expertos humanos en base a la experiencia y conocimiento, es necesario para el diseño de los sistemas de control difusos.

Estos sistemas fundamentalmente están constituidos por tres etapas las cuales son presentadas en la Figura 4.6.

Figura 4.6

Sistema de inferencia difusa.



Fuente propia del Autor

La entrada a estos sistemas generalmente se considera no difusa debido a que suelen provenir de mediciones de una determinada variable. Es por ello que se realiza la

conversión de esas entradas a representaciones en conjuntos difusos. A este proceso se le denomina Fuzzificación y es aquí donde se le asigna a un grado de pertenencia a las reglas difusas para las diferentes situaciones que se puedan dar en el funcionamiento de dicho sistema.

En la siguiente etapa, se establece el conjunto de reglas difusas como la relación que existen entre las entradas y salidas, las cuales son propuestas por el experto en base a su conocimiento y experiencia según la aplicación. Finalmente, obtenido el conjunto difuso de salida resultante proveniente de todas las reglas establecidas, se realiza la interpretación de la información calculando las salidas precisas mediante la etapa de defuzzificación.

4.2.1 Implicaciones difusas.

Las reglas difusas denotan el conocimiento y la experiencia del experto, sin embargo, la relación que existe entre las variables de entrada y las variables salida calculadas (en términos difusos) es llevada gracias a la implicación difusa. De manera clásica para dos proposiciones crisp, $p = \text{“x está en A”}$, $q = \text{“x está en B”}$, la implicación $p \rightarrow q$, que es equivalente a $\neg(p \wedge \neg q)$, puede interpretarse como:

$$\tau(p \rightarrow q) = \begin{cases} 1, & \tau(p) \leq \tau(q) \\ 0, & \text{otro caso} \end{cases} \quad (4.14)$$

La notación de (4.4) define la implicación clásica para conjuntos tradicionales. Para conjuntos difusos existe una gran cantidad de implicaciones, pero hay 6 que son las más conocidas empleadas en ciertas aplicaciones, siendo las de Mamdani y la de Larsen las que generan mejores resultados en el campo de control. Dichas implicaciones son presentadas en la Tabla 4.1. La implicación difusa resultante para dos conjuntos difusos A y B viene dado por la siguiente notación:

$$(A \rightarrow B)_{(u,v)} = \mu_A(u) \rightarrow \mu_B(v) \quad (4.15)$$

Tabla 4.1

Implicaciones difusas más usadas en control.

Nombre	Definición
Zadeh	$x \rightarrow y = \max\{1 - x, \min(x, y)\}$
Lukasewicz	$x \rightarrow y = \min\{1, 1 - x + y\}$
Mamdani	$x \rightarrow y = \min\{x, y\}$
Larsen	$x \rightarrow y = x * y$
Standart Strict	$x \rightarrow y = \begin{cases} 1 & x \leq y \\ 0 & \text{otro caso} \end{cases}$
Godel	$x \rightarrow y = \begin{cases} 1 & x \leq y \\ y & \text{otro caso} \end{cases}$

La implicación usada en la presente tesis es la de Mamdani la cual permite modelar la relación causal entre las variables difusas tomando el mínimo valor verdadero de los predicados difusos. Esta implicación viene dada por la notación:

$$\mu_A(u) \rightarrow \mu_B(v) = \min\{\mu_A(u), \mu_B(v)\} \quad (4.16)$$

A diferencia de las otras implicaciones de la tabla 4.1, no es una extensión correcta de ellas pues $0 \rightarrow 0$ dará como resultado 0. No obstante, en los sistemas inteligentes basados en el conocimiento, el objetivo comúnmente no está interesado en reglas en las cuales la parte del antecedente es falsa.

4.2.2 Principio de Extensión.

El principio de extensión brinda un mecanismo básico para extender el dominio de una función de los puntos crisp en U (expresiones matemáticas de conjuntos exactos) al dominio difuso en U. Este principio generaliza la idea de un mapeo punto a punto de una función en conjuntos tradicionales $y=f(x)$ a un mapeo entre conjuntos difusos. Matemáticamente existe $f:U \rightarrow V$ la cual es una función de mapeo de un conjunto crisp U a un conjunto crisp V. Sea un conjunto difuso A

definido en U y se desea determinar un conjunto difuso B definido en V , tal que $B = f(A)$. Si f es un mapeo uno a uno, entonces se define:

$$\mu_B(y) = \mu_A[f^{-1}(y)], \quad y \in V \quad (4.17)$$

Para la formalización de este concepto, se tiene una función f con n argumentos con trazo de puntos en $U_1 \times U_2 \times \dots \times U_n$ a un punto en V . Además, sea A un subconjunto difuso de $U_1 \times U_2 \times \dots \times U_n$. Entonces el principio de extensión indica que la imagen del subconjunto A bajo la función f es un subconjunto difuso de V con la función de pertenencia:

$$\mu_{f(A)}(y) = \bigoplus_{f(x_1, x_2, \dots, x_n) = y} \mu_A(x_1, x_2, \dots, x_n) \quad (4.18)$$

La función de pertenencia también se puede entender como la distribución de posibilidad de todos los argumentos. No obstante, los argumentos difusos de la función pueden tener cada uno una distribución de posibilidad ($x_1 \rightarrow A_1, x_2 \rightarrow A_2, \dots, x_n \rightarrow A_n$). Bajo lo anterior, la ecuación de $\mu_{f(A)}(y)$ necesita ser modificada para que la expresión $\mu_A(x_1, x_2, \dots, x_n)$ en el lado derecho de la ecuación sea reemplazada por la conjunción difusa de $\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n)$.

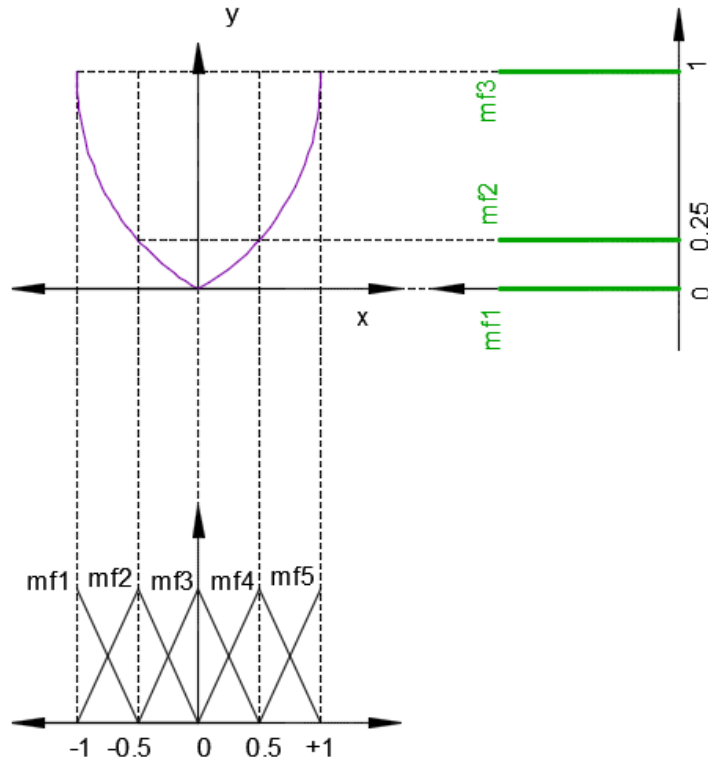
$$\mu_{f(A)}(y) = \bigoplus_{f(x_1, x_2, \dots, x_n) = y} \mu_{A_1}(x_1) \oplus \mu_{A_2}(x_2) \oplus \dots \oplus \mu_{A_n}(x_n) \quad (4.19)$$

Se puede obtener un gráfico referencial que permita entender el mapeo de conjuntos difusos y la formulación de la regla base aplicando el principio de extensión aplicando el principio de extensión al siguiente ejemplo:

Sea $x \in [-1, +1]$ el universo de discurso del conjunto A y sea la función $y = f(x) = x^2$, tal que los puntos de este conjunto difuso elegido son $x = [-1, -0.5, 0, 0.5, +1]$. Los puntos de y respectivos a x son $y = [1, 0.25, 0, 0.25, 1]$. El gráfico del mapeo entre la entrada y salida usando la función mencionada se muestra en la Figura 4.7.

Figura 4.7

Mapeo de entrada-salida para la función parábola.



Fuente propia del Autor

4.2.3 Inferencia difusa de Mamdani.

Existen diversos modelos de sistemas difusos, en los cuales los antecedentes de las reglas que lo conforman están formados por conjuntos difusos, y la diferencia entre estos modelos se origina en la formulación del “consecuente” de la regla base. El modelo que emplea la inferencia de Mamdani, dicha inferencia se da mediante proposiciones difusas, mientras que la inferencia del modelo de Sugeno, se da por interpolación.

El modelo del sistema difuso empleado en la presente tesis es el modelo de Mamdani, el cual se caracteriza por usar conjuntos difusos en los antecedentes y consecuentes de las reglas difusas (Ambos son mapeados mediante conjuntos difusos). Una regla típica en el modelo de Mamdani posee la forma IF-THEN y tienen la siguiente forma:

$$\text{if } x \text{ is } A \text{ then } y \text{ is } B \quad (4.20)$$

Por otra parte, si la regla base tiene dos entradas y una salida tendrá la siguiente forma:

$$\text{if } x \text{ is } A \text{ and/or } y \text{ is } B \text{ then } z \text{ is } C \quad (4.21)$$

La formulación del algoritmo difuso mediante la inferencia de Mamdani es de mucho interés ya que el procedimiento definirá el diseño del controlador difuso. Considerando la regla base para dos entradas y una salida (Sistema MISO: múltiples entradas, una sola salida) como la denotada en (4.21), el procedimiento para ejecutar el algoritmo difuso viene dado por las siguientes instrucciones:

- **Fructificador:** Convierte los valores numéricos dados por $(x, y) = (x_0, y_0)$ de las escalas numéricas de los ejes x e y en valores de grados de pertenencia definidos por:

$$(\mu_x, \mu_y) = (\mu_{Ai}(x_0), \mu_{Bi}(y_0)) \quad (4.22)$$

- **Operación difusa en el antecedente:** Después de obtener el grado del fuzzificador, se procede a realizar la operación difusa del conector lógico AND o PROD, los cuales son los más usados y vienen dados por la expresión (2.23) ó (2.24)

$$s_i(x_0) = \min(\mu_{Ai}(x_0), \mu_{Bi}(y_0)) \quad (4.23)$$

$$s_i(x_0) = \mu_{Ai}(x_0) * \mu_{Bi}(y_0) \quad (4.24)$$

De manera análoga, si el antecedente emplea el operador OR, la operación difusa sería (4.25) ó (4.26):

$$s_i(x_0) = \max(\mu_{Ai}(x_0), \mu_{Bi}(y_0)) \quad (4.25)$$

$$s_i(x_0) = \mu_{Ai}(x_0) + \mu_{Bi}(y_0) - \mu_{Ai}(x_0) * \mu_{Bi}(y_0) \quad (4.26)$$

- **Implicación de cada regla:** Se aplica la implicación de Mamdani, en donde cada regla contribuirá al consecuente difuso mediante el operador mín (4.27) o el producto de la expresión (4.28), la cual tiene por argumentos al disparo

proveniente de la operación difusa (s_i) y la función de pertenencia de salida ($\mu_{C_i}(z)$).

$$D_i(z) = \min(s_i(x_0), \mu_{C_i}(z)) \quad (4.27)$$

$$D_i(z) = s_i(x_0) * \mu_{C_i}(z) \quad (4.28)$$

El consecuente resultante de cada regla modifica la función de pertenencia $C_i(z)$.

- **Agregación:** Este paso se denomina también implicación del consecuente de cada regla, la cual es una combinación de todas las reglas evaluadas de forma individual. Este paso presenta los métodos de operación: máx o sum.

$$C_{agg}(z) = \max(D_i(z)) \quad (4.29)$$

$$C_{agg}(z) = \text{sum}(D_i(z)) \quad (4.30)$$

- **Defuzzificador:** El último paso consiste en la traducción de la función $C_{agg}(z)$ en un valor crisp de salida z_0 . Se puede utilizar cualquier método, entre los cuales los más usados son el del Centroide, Bisector, Medios de Máximos, etc. Para la presente tesis se usará el método del Centroide, el cual es el más usado y viene dado por la siguiente expresión:

Cálculo del Centroide en forma continua:

$$z_0 = \frac{\int z * C_{agg}(z) dz}{\int C_{agg}(z) dz} \quad (4.31)$$

Cálculo del Centroide en forma discreta:

$$z_0 = \frac{\sum z_i * C_{agg}(z_i)}{\sum C_{agg}(z_i)} \quad (4.32)$$

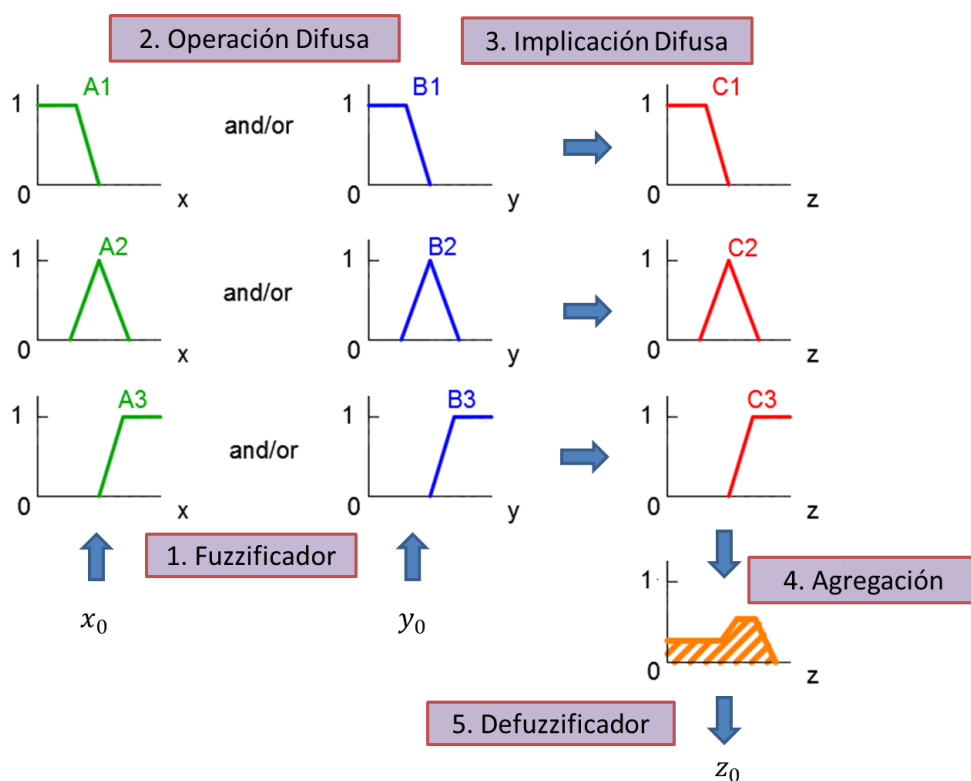
4.2.4 Estructura del FIS de Mamdani.

El modelo de Mamdani usa conjuntos difusos tanto en el antecedente como en el consecuente de la regla base. En este modelo la salida de la etapa de la inferencia es representada por un conjunto difuso, el cual es el resultado obtenido de la agregación de las salidas inferidas por cada regla, que posteriormente genera una

salida exacta utilizando uno de los métodos de defuzzificación mencionados anteriormente. La estructura del FIS (Sistema de Inferencia Difusa en español) de Mamdani, detallado en el sub-capítulo anterior, es presentada en la Figura 4.8 para un sistema MISO de dos entradas (A y B) y una salida C.

Figura 4.8

Estructura del FIS de Mamdani.



Fuente propia del Autor

4.3 Sistemas de Control Difuso

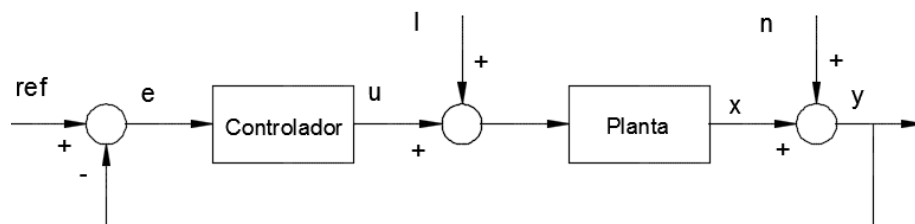
A diferencia de los controladores PID clásicos, un controlador PID difuso es un controlador fuzzificador proporcional-integral-derivativo (PID). Este controlador difuso actúa sobre las mismas señales que el controlador PID clásico, pero la estrategia de control se formula mediante reglas difusas. Los efectos de sobre impulso, tiempo de subida y establecimiento de la respuesta del sistema a una

entrada escalón unitario, son difíciles de predecir al cambiar las reglas difusas o las ganancias de ajuste del controlador, debido a que este controlador inteligente generalmente es no lineal y tiene una estructura compleja. Un controlador PID es una combinación lineal simple compuesta por tres señales: la acción proporcional al error (e), la acción integral al error ($\int edt$) y la acción derivativa proporcional a la derivada temporal del error (de/dt).

Un sistema de control retroalimentado viene dado por la Figura 4.9 en donde el objetivo es controlar la variable de salida $y = x + n$, la cual incluye un ruido n , referenciada por la entrada ref y contrarrestando los efectos a los cambios de esta referencia, a un cambio de carga l o al mismo ruido n . Si la planta es monótona, de tal manera que la salida de la planta x aumenta al aumentar el valor de la entrada $u + l$, el controlador deberá aumentar la señal de control (u) cuando la salida controlada (y) este por debajo de la referencia (ref), mientras que disminuirá (u) cuando (y) este por encima del valor de la referencia. La retroalimentación negativa es la encargada de que la señal de control se mueva en dirección opuesta a la salida de la planta.

Figura 4.9

Sistema de control en lazo cerrado.



Fuente propia del Autor

Es común emplear al error $e = ref - y$ como una entrada al sistema de control difuso (FLC), al igual que en el controlador clásico PID, y observar como la acción de control actúa en magnitud y signo del error e . Cuando el problema de control es la regulación de la variable de salida en torno a la referencia, la señal del error es

tomada como entrada al FLC seguido de su integral y derivada. No obstante en el proceso de fuzzificación del FLC es difícil establecer el efecto de cada valor de ganancia, tiempo de subida, sobre impulso y tiempo de establecimiento pues el controlador suele ser no lineal y tiene más parámetros de sintonía que un PID clásico.

Existen diversos tipos de estructuras difusas que surgen de la combinación de las estructuras difusas básicas del PID difuso, algunas con ciertas características que las diferencian de otras según el tipo de aplicación. La estructura que puede presentar dicho controlador PID difuso es muy variada según el número de entradas y salidas que se trabajen. Las posibles estructuras derivadas elementales son presentadas en la Tabla 4.2 en las que las entradas al controlador difuso pueden ser el error, su derivada o la integral del mismo, y la salida un valor absoluto o incremental de la acción de control.

Tabla 4.2

Posibles estructuras elementales de los controladores PID-difusos.

Una entrada y una salida (posicional)	Proporcional $U = f(E)$	Derivativo $U = f(dE/dt)$	Integral $U = f(\int E)$
Una entrada y una salida (incremental)	Proporcional $dU = f(dE/dt)$	Derivativo $dU = f(d^2E/d^2t)$	Integral $dU = f(E)$
Dos entradas y una salida (posicional)	PD $U = f(E, dE)$	PI $U = f(E, \int E)$	-
Dos entradas y una salida (incremental)	PD $dU = f(E, dE)$	PI $dU = f(dE, d^2E)$	-
Tres entradas y una salida	PID (posicional) $U = f(E, dE, \int E)$	PID (incremental) $dU = f(dE, d^2E, E)$	-

La función f denota el comportamiento del controlador difuso, E es el error, dE es la derivada del error, $\int E$ es la integral del error, U es la acción de control y dU es la derivada de la acción de control. Esta función f representa el comportamiento del controlador difuso, el cual está determinado por el enunciado de sus reglas que se ejecutan en un único bloque o sistema de inferencia difusa.

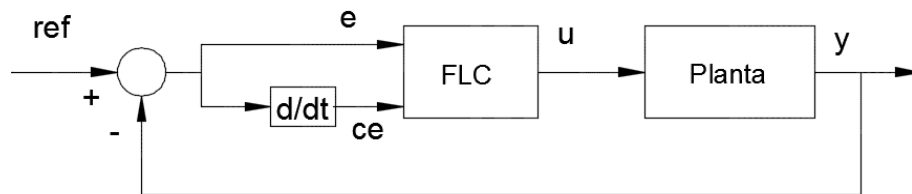
4.3.1 Diagrama básico de un Controlador Difuso.

La configuración más sencilla de un controlador con lógica difusa en un sistema retroalimentado se muestra en la Figura 4.10. Estos sistemas FLC, comúnmente están conformados por:

- **Entradas de control:** error (e), cambio del error (ce), entrada del proceso (setpoint).
- **Conjuntos difusos con funciones de pertenencia de entrada:** El número de conjuntos difusos, el tipo de la función de pertenencia y las ubicaciones de las funciones de pertenencia para cada entrada.
- **Reglas difusas:** Conformado por el conocimiento del experto expuestas de la forma IF-THEN.
- **Salida de control:** ley de control (u), cambio de la ley de control (cu).
- **Conjuntos difusos con funciones de pertenencia de salida:** El número de conjuntos difusos, el tipo de la función de pertenencia y las ubicaciones de las funciones de pertenencia para la salida.
- **Razonamiento difuso:** implicación difusa y agregación (máx.-min o sum-producto).
- **Defuzzificación:** Centro de gravedad, medios de máximos o pesos promedios.

Figura 4.10

Diagrama de bloques de un sistema controlado por un sistema FLC.



Fuente propia del Autor

Los controladores FLC más usados presentan dos entradas y una salida, las cuales son el error y el cambio del error.

Estos controladores difusos se pueden expresar mediante la siguiente expresión:

$$u = f(e, ce) \quad (4.33)$$

En esta notación la función f representa el algoritmo de inferencia de Mamdani, donde las entradas al controlador son las mencionadas anteriormente.

4.3.2 Edición de reglas.

La formulación de las reglas representa el conocimiento del experto sobre el proceso a controlar y define la estrategia de control. Para poder ejecutar el control del sistema Viga y Bola se deben establecer una serie de reglas de la forma IF-THEN para indicar las acciones que el controlador va a realizar. Estas reglas de control están basadas en la intuición, sentido común y experiencia del experto acerca del funcionamiento del sistema. El controlador FLC será más robusto frente a perturbaciones y el ruido de los sensores mientras mayor sea el número de reglas de dicho controlador. Una buena base de reglas es la propuesta por 4 reglas difusas. En ella, tanto el error, el cambio del error y la salida están subdivididas en conjuntos difusos dentro del universo de discurso. Tomaremos solo 4 reglas difusas y medir su performance, estas son listadas a continuación.

R1: If E is N and CE is N then u is N

R2: If E is N and CE is P then u is Z

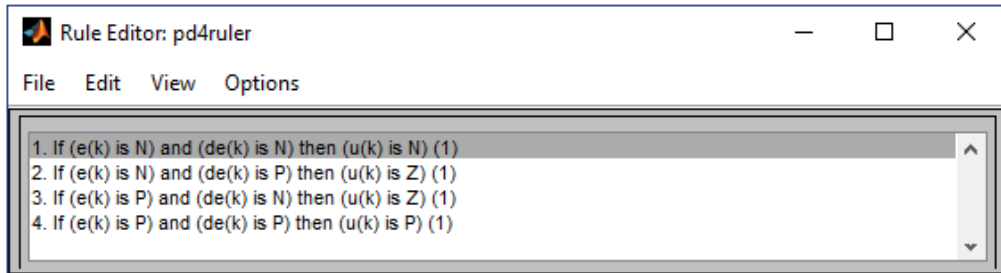
R3: If E is P and CE is N then u is Z

R4: If E is P and CE is P then u is P

La regla base pueden ser implementadas en el Toolbox de MATLAB (ver Figura 4.11).

Figura 4.11

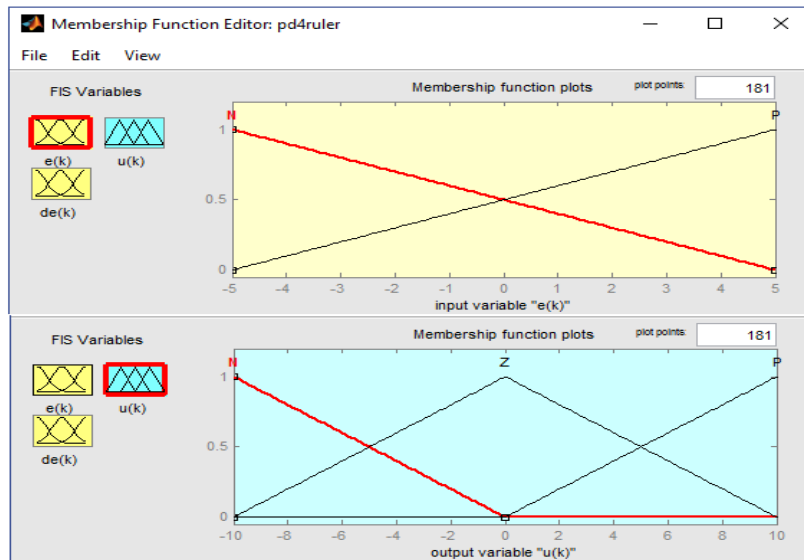
Tabla de regla base.



En torno a lo anterior mencionado las variables lingüísticas para las entradas de las reglas difusas (e y ce) y la salida (u), pueden definirse mediante conjuntos difusos según la Figura 4.12, donde los límites $[-a +a]$ de su universo de discurso serán definidos en el capítulo del diseño del controlador.

Figura 4.12

Conjuntos difusos definidos para las entradas y salida del sistema FLC.



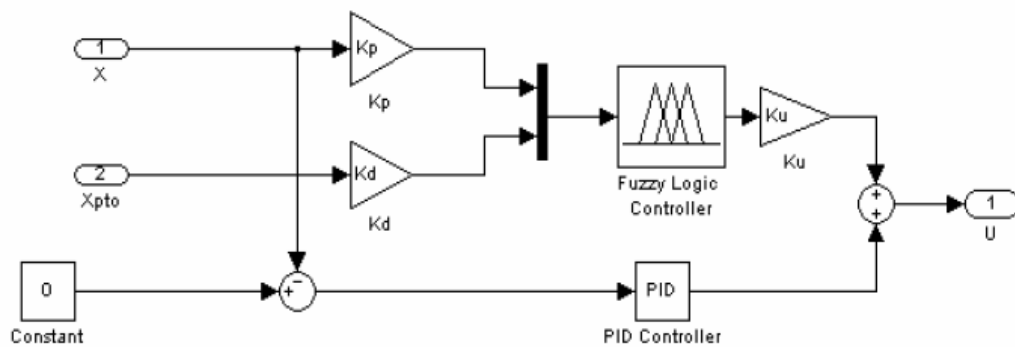
Fuente propia del Autor

4.4 Estructura de Control Difuso para el Robot

La idea por la cual se propone esta estructura es tomar en cuenta la parte clásica del controlador (PID) complemente la acción de control del FLC, esto es porque la superficie de control o el sistema pueden presentar mínimos locales dado por que no se puede asegurar estabilidad global sin un análisis riguroso (como el uso del Teorema de la Mínima ganancia o de Estabilidad de Lyapunov). Cuando el vector de estado cae en un mínimo local, puede ser por que el controlador no está lo suficientemente bien diseñado para sacar al proceso de tal mínimo o por que el controlador provoca un mínimo no deseado. En ambos casos, el resultado más evidente es que se mantiene un error en estado estacionario. Con la acción integral del controlador PID este error puede ser anulado, en conjunto con una posible mejora en la respuesta temporal del sistema si los parámetros proporcional y derivativos del controlador PID están correctamente ajustados (ver Figura 4.13).

Figura 4.13

Estructura del FLC con un PID paralelo



Fuente propia del Autor

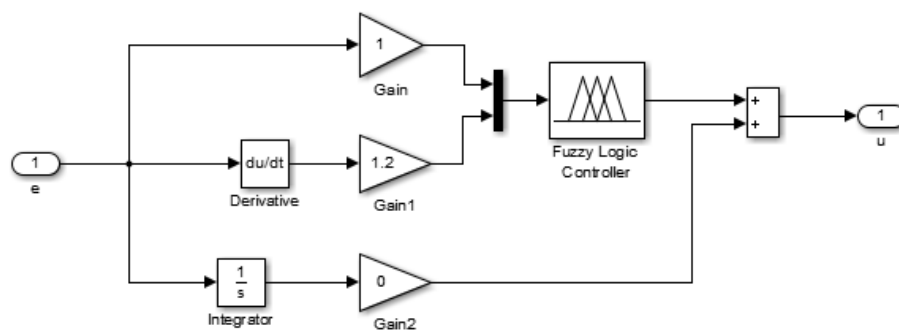
4.4.1. Controlador Difuso Puro (FPID)

Esta estructura típica es la base para muchos otros controladores. En este caso, el reprocesado y pos procesado, estará limitado solo al escalamiento de las señales de entrada (el error y su derivada, e y de) y de salida. La variable de salida es del tipo posicional y no incremental, es decir que el valor que obtiene el sistema de inferencia difusa es aplicado directamente al actuador del proceso, el segundo tipo,

es decir el incremental (la salida del sistema de inferencia difusa es un incremento sobre el valor aplicado al actuador del proceso) es deseable en muchos casos, por lo que deberá ser necesario que el pos procesado resuelva esta situación. Este controlador está diseñado para emular a un controlador clásico PID (proporcional, integral y derivativo). Por lo tanto se lo define como un controlador PID difuso (FPID). La actual estructura de FLC será empleada para regulación con referencia igual a cero, entonces será en este punto donde se buscará cumplir con los objetivos de control. Esta estructura es mostrada en la Figura 4.14.

Figura 4.14

Estructura del FLC tipo PID



Fuente propia del Autor

4.5 Población y muestra.

Por el tipo de investigación experimental no corresponde determinar el tamaño de muestra.

4.6 Técnicas e instrumentos de recolección de datos

Debido a que no se ha determinado el tamaño de la muestra, no corresponde utilizar las técnicas e instrumentos para la recolección de datos.

4.7 Procedimientos de recolección de datos

No corresponde al presente trabajo de investigación.

4.8 Procedimiento estadístico y análisis de datos

No es necesario utilizar procedimientos y análisis de datos.

V.- RESULTADOS

5.1 Simulaciones Control Difuso tipo PID

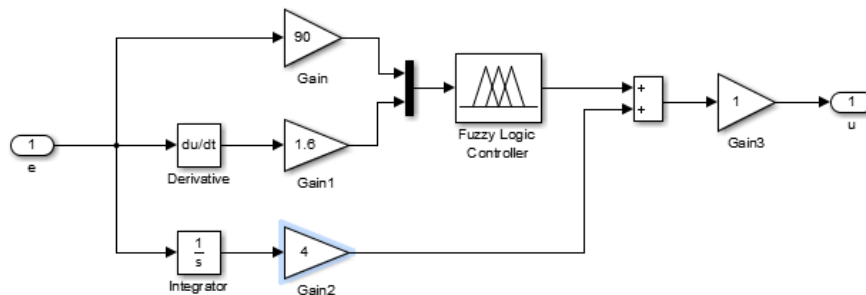
Mediante el programa simulink, nos permite obtener la respuesta en las articulaciones.

Primera Articulación

Para las simulaciones es necesario tener en cuenta las ganancias de la acción integral ya que se tiene las ganancias del controlador PID previamente diseñadas.

Figura 5.1

Sistema de control difuso primera articulación

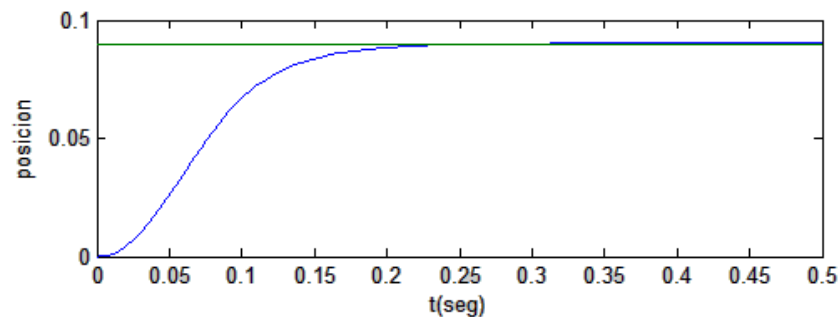


Fuente propia del Autor

Por ejemplo, para el control de posición de la primera articulación no presenta sobre impulso y error en estado estacionario nulo, además que la respuesta es rápida. Esto lo podemos apreciar en la Figura 5.2

Figura 5.2

Control de posición articulación 1.

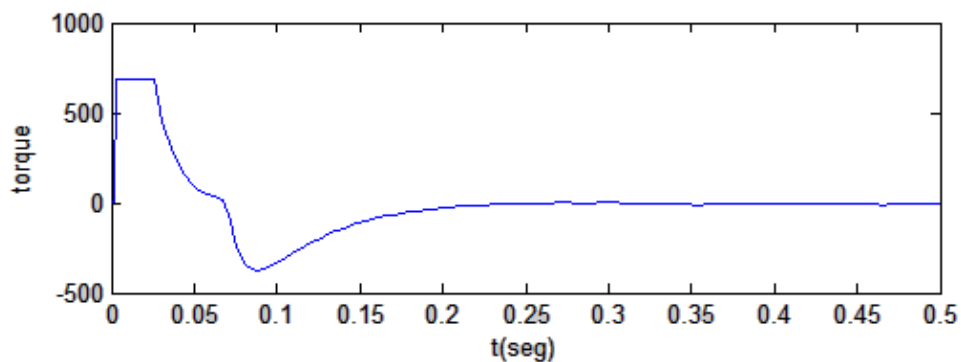


Fuente propia del Autor

Es importante obtener la simulación para la acción de control del robot de 3GDL, en este caso se observa que la acción de control se equilibra cada vez que la posición no varía en el estado estacionario, esto lo podemos apreciar en la curva de la Figura 5.3.

Figura 5.3

Torque ejercido en la articulación 1



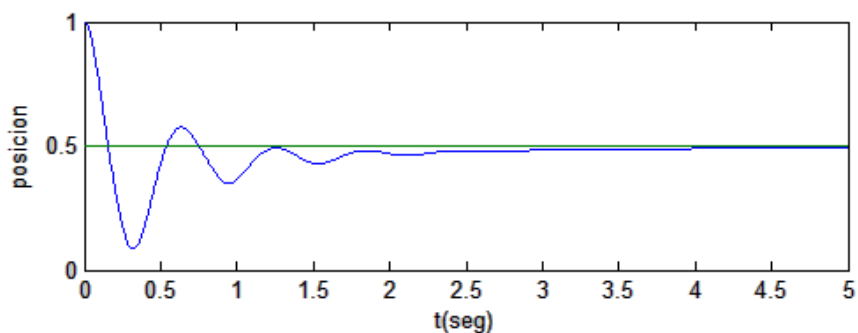
Fuente propia del Autor

Segunda Articulación

Por ejemplo para el control de posición de la segunda articulación presenta sobre impulso y error en estado estacionario nulo, además que la respuesta es relativamente rápida. Esto lo podemos apreciar en la Figura 5.4.

Figura 5. 4

Control de posición articulación 2.

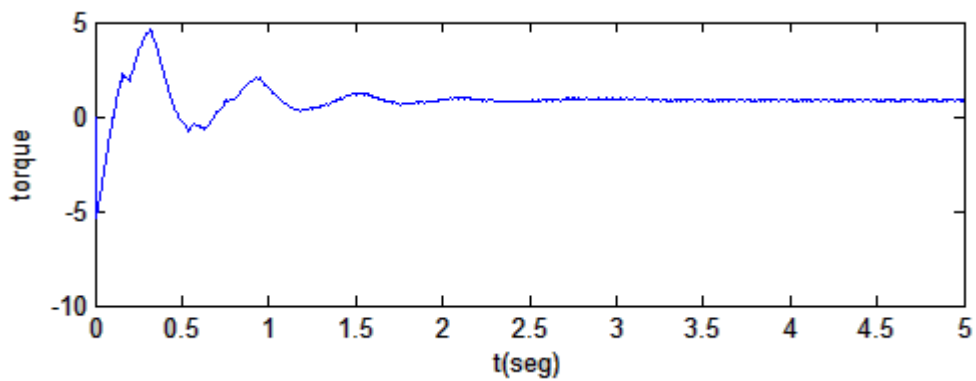


Fuente propia del Autor

Es importante obtener la simulación para la acción de control del robot de 3GDL, en este caso se observa que la acción de control viene limitada y se atenúa rápidamente, esto lo podemos apreciar en la curva de la Figura 5.5.

Figura 5.5

Torque ejercido en la articulación 2



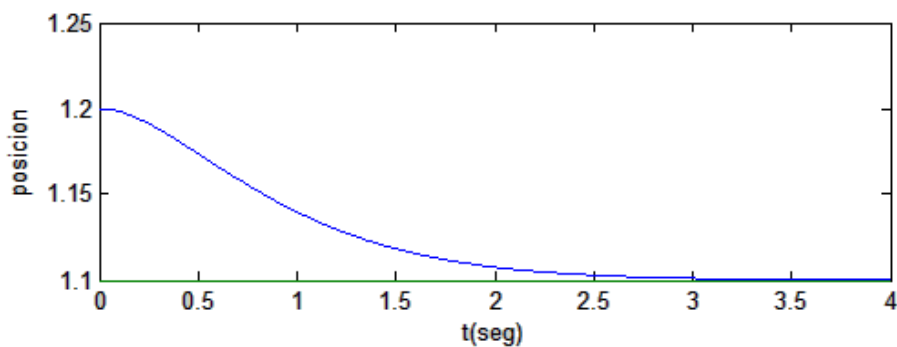
Fuente propia del Autor

Tercera Articulación

Por ejemplo, para el control de posición de la tercera articulación su respuesta es bastante suave y se estabiliza en menos de 2 segundos. Esto lo podemos apreciar en la Figura 5.6

Figura 5.6

Control de posición articulación 3.

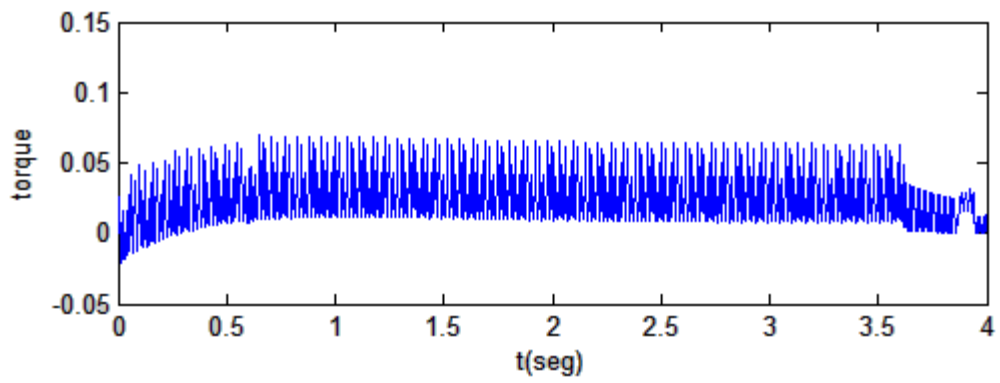


Fuente propia del Autor

Se observa que la acción de control un tanto oscilatoria en la tercera articulación, es debido a la inercia del robot que debe vencer y compensar (ver Figura 5.7).

Figura 5.7

Torque ejercido en la articulación 3.



Fuente propia del Autor

VI.- DISCUSION DE RESULTADOS

Los robots son entidades mecánicas físicas, pero también estos pueden ser emulados virtualmente que por apariencia y movimientos parecen tener vida. Si bien muchos mecanismos son llamados robots, estos son identificados como mecanismos dotados de movimiento capaces de interactuar dinámicamente y en tiempo real con su entorno para desarrollar una variedad de tareas que le son encargadas mediante enseñanza o programación. El objetivo de este capítulo es realizar las simulaciones para el sistema brazo robótico de 3GDL conformado por 3 eslabones rectos y 3 articulaciones de tipo revoluta. Comenzando desde el eslabón unido al origen y continuando la cadena cinemática la longitud de los eslabones l_1 , l_2 y l_3 respectivamente.

6.1 Simulaciones de la Cinemática Directa

Estas simulaciones son desarrolladas desde la tabla de los parámetros Denavit Hartenberg desarrollador en el capítulo 2. Usaremos el Toolbox de Robótica den el ambiente de programación de MATLAB. Se trata de una simulación que resuelve la cinemática directa mediante el comando robot que es un objeto que maneja estructuras de campo que hace posible mover las articulaciones y mostrar diferentes posturas del robot de 3GDL.

```
clear all; close all; clc;

%-----

hp = 0.21; % Altura del prisma
hd = 0.01; % Altura del disco
rm2= 0.05; % Distancia extremo superior del prisma al DC2
l1=hp+hd+rm2;
l2 = 0.20; % Longitud de la 1ra varilla
l3 = 0.20; % Longitud de la 2da varilla

%-----

% Coordenadas
q=[0 0 0];
```

```

q1=q(1); q2=q(2); q3=q(3);

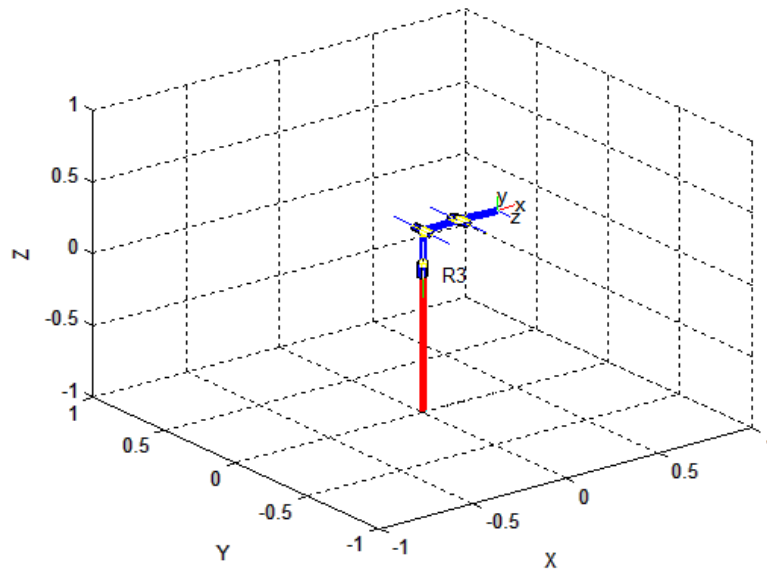
%-----
% Parámetros Denavit-Hartenberg
% -----
%   alpha  a  theta  d  sigma  offset
% -----
L{1} = link([pi/2  0  q1  l1  0  0]);
L{2} = link([0  l2  q2  0  0  0]);
L{3} = link([0  l3  q3  0  0  0]);
%-----

R3=robot(L,'R3','UNAC','Robot3DOF(RRR)');
R3.name    = 'R3';
R3.manuf   = 'UNAC';
R3.plotopt = {'workspace',[-4 4 -4 4 -1 5.8]};
R3.lineopt = {'color','blue','linewidth',4};
R3.shadowopt = {'color','black','linewidth',1};
%-----

T=fkine(R3,q);
drivebot(R3)
axis([-1 2 -1 2 -1 2])

```

Figura 6.1
Robot de 3GDL



El movimiento del robot de 3GDL puede ser resuelto mediante la implementación de dos funciones a que a continuación detallaremos:

La función [directa.m](#) es la encargada de resolver las matrices de transformación homogénea para cada coordenada del efector final.

```
function [x y z]=directa(q)
hp = 0.21; % Altura del prisma
hd = 0.01; % Altura del disco
rm2= 0.05; % Distancia extremo superior del prisma al DC2
l1=hp+hd+rm2;
l2 = 0.20; % Longitud de la 1ra varilla
l3 = 0.20; % Longitud de la 2da varilla
```

```
A01=[cos(q(1)) 0 sin(q(1)) 0
sin(q(1)) 0 -cos(q(1)) 0
0 1 0 l1
0 0 0 1];
```



```

A12=[cos(q(2)) -sin(q(2)) 0 l2*cos(q(2))
sin(q(2)) cos(q(2)) 0 l2*sin(q(2))
0 0 1 0
0 0 0 1];

```

```

A23=[cos(q(3)) -sin(q(3)) 0 l3*cos(q(3))
sin(q(3)) cos(q(3)) 0 l3*sin(q(3))
0 0 1 0
0 0 0 1];

```

```

A02=A01*A12;
A03=A02*A23;
xyz=[A01(1:3,4) A02(1:3,4) A03(1:3,4)];
x=xyz(1,:);
y=xyz(2,:);
z=xyz(3,:);
end

```

Función [R3.m](#) es la encargada de plotear la base y las asignaciones de los ejes coordinados en 3D.

```

function R3(q)
[x y z]=directa(q);
hold on
plot3([0 x],[0 y],[0 z],'b','linewidth',4),grid;
plot3([0 0],[0 0],[0 -1],'k','linewidth',4);
plot3([0 x(3)],[0 y(3)],[-1 -1],'k','linewidth',2);
for i=1:3
    plot3(x(i),y(i),z(i),'r','MarkerSize',17);
end
xb=[1 1 -1 -1 1]*0.1;
yb=[1 -1 -1 1 1]*0.1;

```

```

zb=[0 0 0 0 0]*0.1;
fill3(xb,yb,zb,'y')
hold off

```

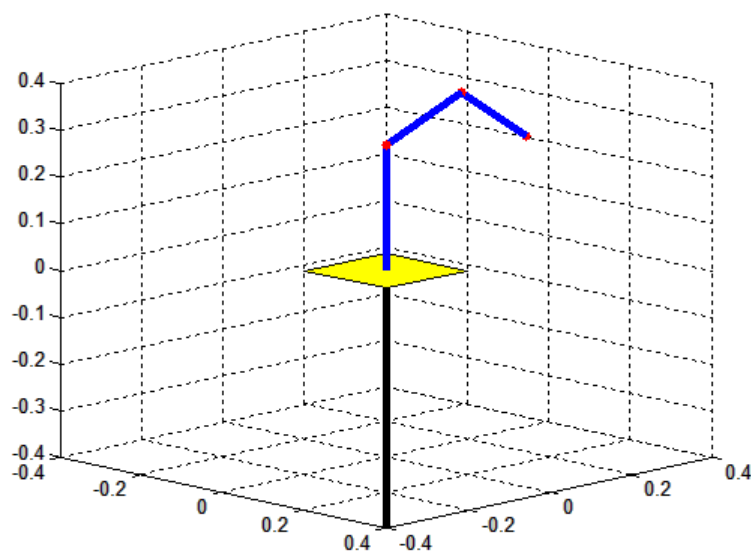
Finalmente el programa `movimiento.m` es la encargado de dar movimiento bajo asignación de las posiciones angulares arbitrarias.

```

for j=1:6
    clf
    q=[pi/3-pi/4*j pi/8 -pi/3];
    view(45,15);
    axis ([-0.4 0.4 -0.4 0.4 -0.4 0.4]);
    R3(q);
    pause(0.4);
end

```

Figura 6.2
Robot de 3GDL y sus posturas



Fuente propia del Autor

6.2 Simulaciones del Control Difuso tipo PID

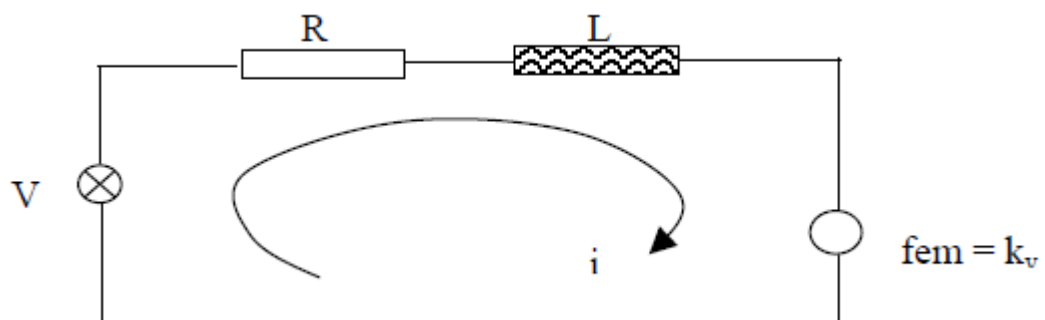
El software de simulación empleado para el control del robot de 3GDL mediante la técnica de control difuso tipo PID presenta una variedad de abordar el control del robot, una de las técnicas es estudiar al robot de 3GDL en forma desacoplada, es decir realizar la sintonía del controlador PID en forma individual.

En primer término, es importante escoger el actuador para el movimiento de la cadena cinemática del robot, en esta ocasión se ha previsto tener motores dc de igual modelo por simplicidad.

Se va a utilizar un modelo eléctrico válido en régimen permanente. El comportamiento dinámico del motor para régimen permanente se puede representar mediante el siguiente circuito eléctrico, como se observa en la Figura 6.3, y ecuaciones siguientes.

Figura 6.3

Diagrama de bloques del Motor DC

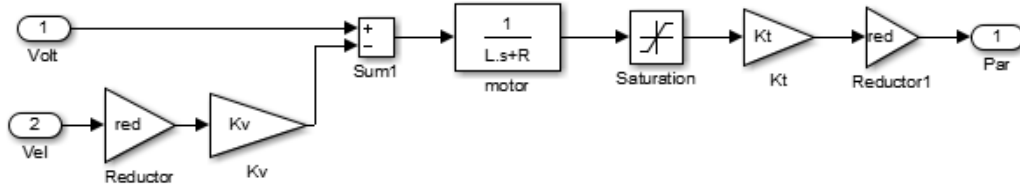


Fuente propia del Autor

$$V = Ri + L \frac{di}{dt} + K_v \omega$$

El par del motor es proporcional a la corriente circulante. ($\tau = K_t i$). El diagrama de bloques del motor dc se muestra en la Figura 6.4. En el modelo solo se ha considerado la parte eléctrica ya que la parte mecánica que corresponde a la inercia y fricción se han tomado dentro del modelo del robot de 3GDL.

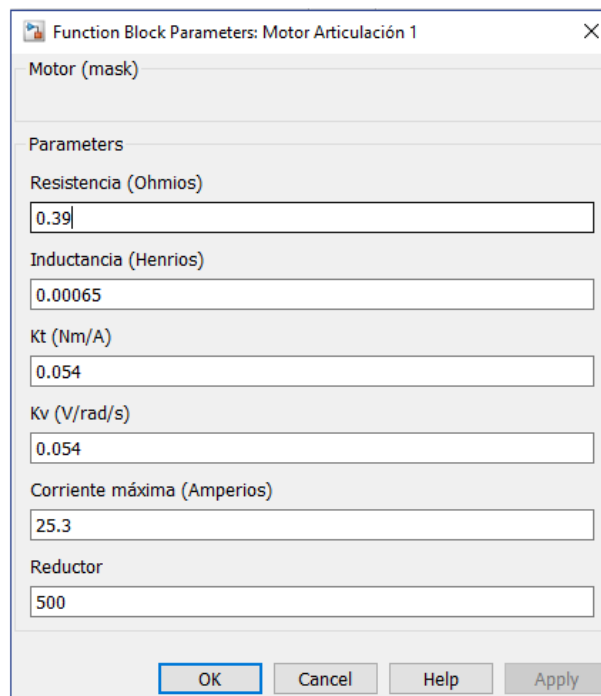
Figura 6.4
Diagrama de bloques del Motor DC.



Fuente propia del Autor

Los parámetros que se han considerado son los mostrados en la ventana de Simulink (ver Figura 6.5), entre ellos tenemos, la resistencia de armadura, la inductancia, la constante de motor, la constante contra electromotriz, la corriente máxima y el reductor.

Figura 6.5
Parámetros característicos del motor DC.



Fuente propia del Autor

Otro parámetro importante es el valor de amplificación del motor dc que puede oscilar entre dos valores 5 y 25 y la saturación para limitar la acción de control, cuyos valores están comprendidos en el intervalo mínimo -10, y máximo 10.

Hay dos estructuras de control típicas utilizadas en los robots: control acoplado y control desacoplado.

CONTROL DESACOPLADO

En el control desacoplado se considera que las articulaciones del robot están desacopladas, de modo que un par en un determinado actuador únicamente tendrá efecto sobre el movimiento de la articulación correspondiente. De esta forma existirá un controlador para cada articulación. La ventaja del control desacoplado radica en que el diseño del regulador más adecuado para cada articulación puede hacerse utilizando las técnicas más frecuentes de diseño.

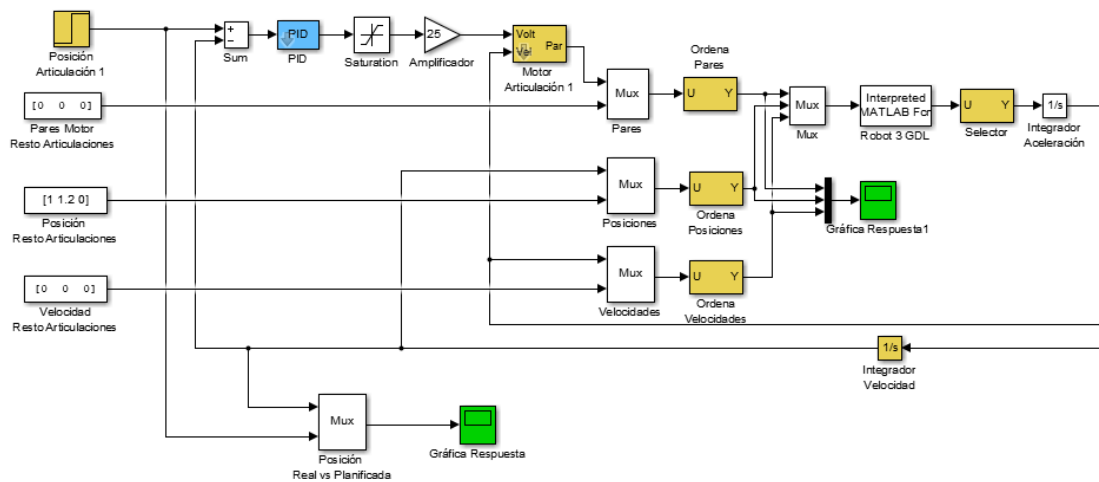
CONTROL ACOPLADO

En ocasiones, la suposición de que el robot es una serie de eslabones dinámicamente desacoplados, de forma que el movimiento de uno de ellos no afecta a los demás, no es siempre aceptable. Las técnicas de control acoplado consideran el modelo dinámico real del robot, haciendo uso del conocimiento del mismo para tratar de desacoplar el sistema.

Adoptaremos una estructura de control desacoplado, de manera que en cada articulación existe un regulador PID. Durante el sintonizado de una articulación, los actuadores del resto de articulaciones permanecen parados. En la Figura 6.6 se muestra la estructura de control desacoplado típica de un robot de tres grados de libertad.

Figura 6.6

Diagrama de bloques del control PID del robot de 3GDL en desacoplo.



Fuente propia del Autor

6.3 Técnicas de Sintonizado.

El sintonizado de un robot consiste en ajustar cada uno de los controladores para que el robot satisfaga un comportamiento especificado. Para ello, en un esquema de control desacoplado, se deberá sintonizar de manera independiente cada articulación del robot. El sintonizado de una articulación consiste en ajustar los parámetros del regulador que actúa sobre ella para que su respuesta cumpla unas determinadas especificaciones. El controlador que utilizaremos en cada una de las articulaciones será un regulador PID. La función de transferencia de un regulador PID es la siguiente:

Para realizar el ajuste de los parámetros de un regulador PID existen múltiples métodos. A continuación se van a estudiar dos métodos muy utilizados: el manual y el de Ziegler- Nichols.

MÉTODO MANUAL (PRUEBA Y ERROR)

Este método propone una manera de ir modificando los parámetros del regulador para conseguir que la respuesta de la articulación cumpla las especificaciones impuestas.

1. En primer lugar, se aumenta la constante proporcional P hasta obtener el tiempo de cruce deseado (el menor posible sin tener una sobre oscilación exagerada).
2. A continuación, se procede a aumentar la constante derivativa D para disminuir la sobre oscilación (esto incrementará el tiempo de cruce).
3. Por último, se aumenta la constante integral I para eliminar el error que exista en régimen permanente (esto incrementará la sobre oscilación y disminuirá el tiempo de respuesta).

MÉTODO DE ZIEGLER-NICHOLS

Es un método experimental en el que es necesario que la respuesta de la articulación en bucle cerrado ante escalón sea periódica. Para ello solamente se introduce una ganancia proporcional que se va aumentando hasta conseguir la respuesta periódica. Si no se consigue que la respuesta sea periódica, no se puede aplicar este método. Sea K_{cr} la ganancia proporcional crítica a la cual la respuesta del sistema se vuelve periódica y P_{cr} el período de la respuesta. En base a estos valores, el método de Ziegler- Nichols proporciona los valores del regulador que aseguran una respuesta aceptable. Los valores que proporciona este método se muestran en la Tabla 6.1.

Tabla 6.1
Valores de los parámetros del controlador PID

K_p	=	$0.6K_{cr}$
T_i	=	$0.5P_{cr}$
T_d	=	$0.125P_{cr}$

Hay que destacar que es posible que la respuesta obtenida aplicando los valores obtenidos por este método no satisfaga las especificaciones impuestas, por lo que será necesario modificar estos valores mediante prueba y error.

La utilidad de este método es que nos proporciona un punto de partida para realizar el ajuste de los parámetros.

Vamos a suponer que el esquema de control de este robot es desacoplado, por lo que cada articulación dispondrá de un regulador PID y durante el proceso de sintonizado de una articulación concreta, los actuadores del resto de articulaciones permanecerán apagados.

Para efectuar el sintonizado del robot hay que situarlo en la peor posición, introducir a la articulación que se desea ajustar un escalón y comprobar que la respuesta de la articulación cumple las especificaciones deseadas.

La peor posición en la que se puede encontrar el robot viene dada por:

$$q = (0 \ 1 \ 1.2 \ 0)$$

Para un funcionamiento satisfactorio del robot exigiremos que la respuesta de cada articulación cumpla las siguientes especificaciones

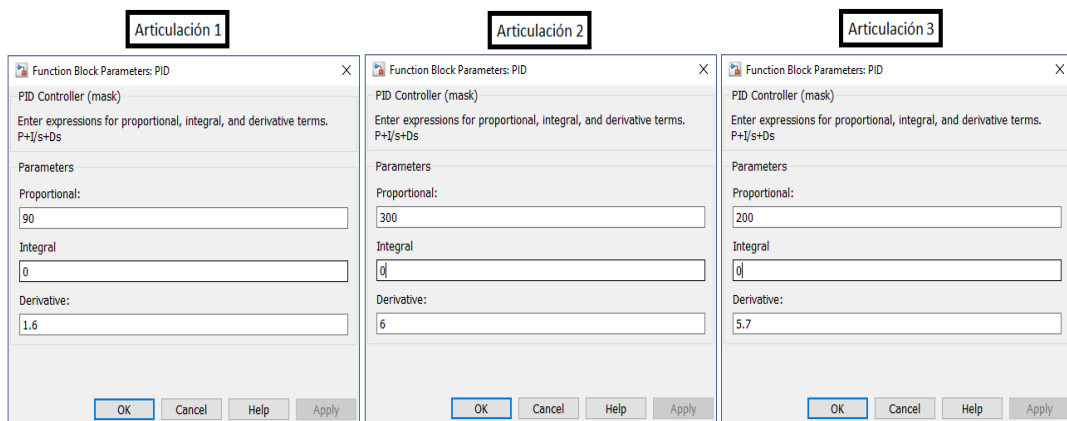
- Tiempo de cruce: < 100 mseg
- Sobreoscilación: < 20 %
- Tiempo de establecimiento: < 200 mseg

La entrada a las articulaciones rotacionales (articulación 1 y 3) para realizar el sintonizado será un escalón de 0.08 rad. En las articulaciones prismáticas se introducirá un escalón de 100mm (0.1m) para sintonizar los reguladores. El escalón será positivo o negativo en función de la articulación que se desee sintonizar. El escalón siempre debe elegirse de modo que precise los máximos requerimientos de par. En el caso de las articulaciones 1 y 4 es independiente considerar el escalón positivo o negativo, ya que, a efectos de requerimientos de par, no hay diferencia en girar la articulación en un sentido o en otro. En es posible introducir el escalón en el sentido positivo del eje z. Por esta razón, en las articulaciones 2 y 3 se ha introducido un escalón negativo.

Se ha desarrollado en Simulink un esquema de control en lazo cerrado, para cada una de las articulaciones, que permite ajustar un regulador PID y comprobar si la respuesta del sistema ante una entrada escalón satisface las especificaciones impuestas.

Los parámetros previstos para las tres articulaciones son los que se listan a continuación.

Figura 6.7
Parámetros de sintonía del robot 3GDL.



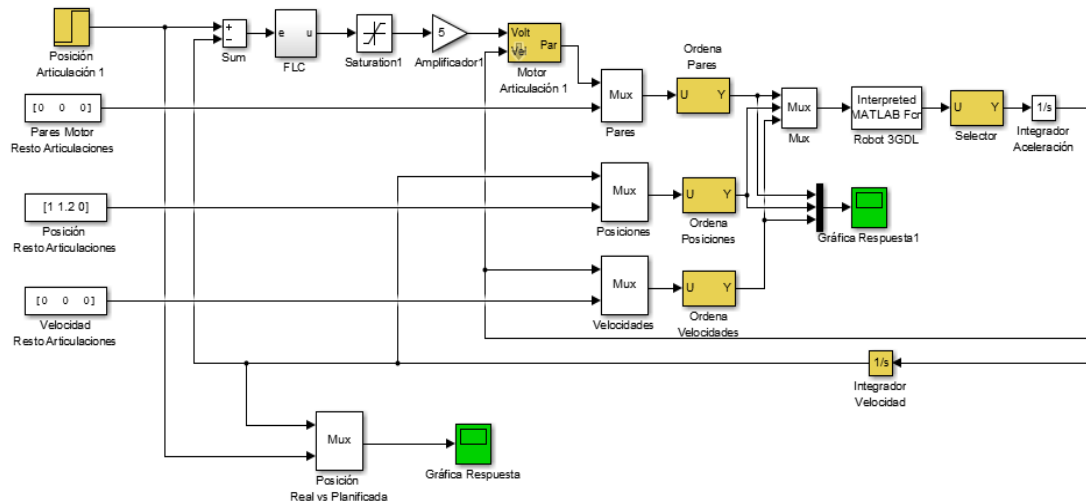
6.4. Simulaciones del Control Difuso tipo PID

Como se vio en el capítulo anterior se aplicará la técnica de control difuso tipo PID. En este caso nuestro esquema de control presenta un controlador difuso el mismo que ha sido analizado para cada articulación. Además, se ha visto la inferencia difusa de Mamdani y se ha utilizado la regla base de 4 reglas y 9 reglas según la articulación. Las medidas de la performance de control se demostrarán cuando se realiza las simulaciones y poder validar la técnica empleada. En este caso la acción de control debe ser capaz de obtener un mínimo sobrepaso, error en estado estacionario nulo y tiempo de establecimiento según las consideraciones de diseño planteadas para el robot de 3GDL.

El esquema de la Figura 6.8 corresponde a la primera articulación, es decir la de la base del robot. Como se mencionó anteriormente se está haciendo el análisis del robot desacoplado.

Figura 6.8

Sistema de control difuso tipo PID del robot 3GDL.

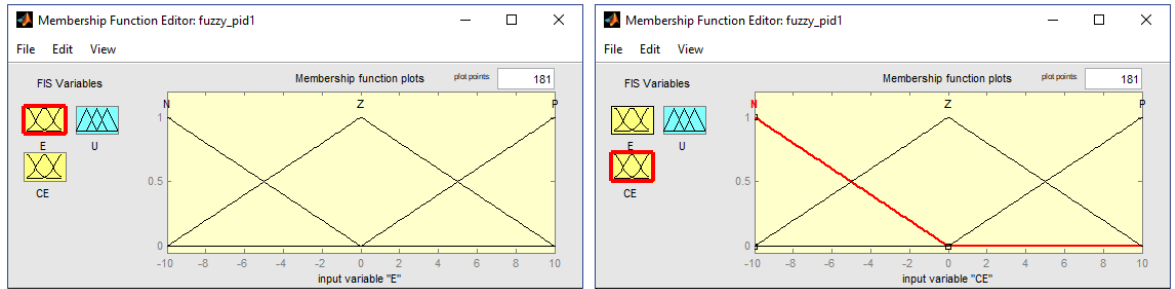


El sistema FLC (Fuzzy Logic Control) del esquema de la Figura 6.8 tiene un interés especial por lo que hay un diseño del sistema de control. En este caso lo primero a tratar es la regla base para la inferencia difusa. Debemos mencionar que el sistema difuso tiene dos entradas, el error y el cambio del error. La parte integrativa se suma con la acción de control difusa.

6.4.1 Inferencia Difusa Articulación 1-2

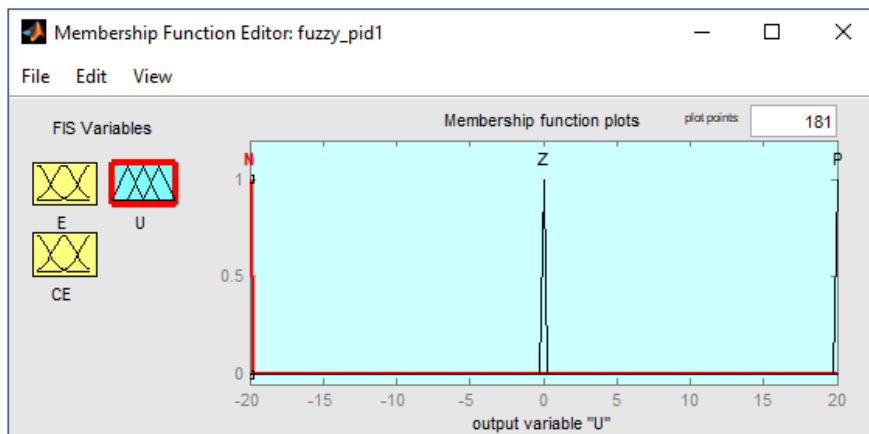
Las funciones de pertenencia para las entradas y salidas para las articulaciones 1 y 2 que corresponden a la base y el primer brazo del robot de 3GDL. También presentan funciones de pertenencia triangulares y el universo de discurso, que pueden ser apreciados en el esquema de la Figura 6.9. Podemos ver las etiquetas lingüísticas denotas por Negativo (N), Cero (Z) y Positivo (P). El universo de discurso a considerar es de -10V a 10V que corresponde a la variable física.

Figura 6.9
Funciones de pertenencia de entrada.



La función de pertenencia aproximada de Singleton difusos a la salida es mostrada en la Figura 6.10. En este caso las etiquetas lingüísticas son las mismas que las entradas. El universo de discurso es el doble que el de la entrada (consideración importante en el diseño).

Figura 6.10
Funciones de pertenencia de salida.



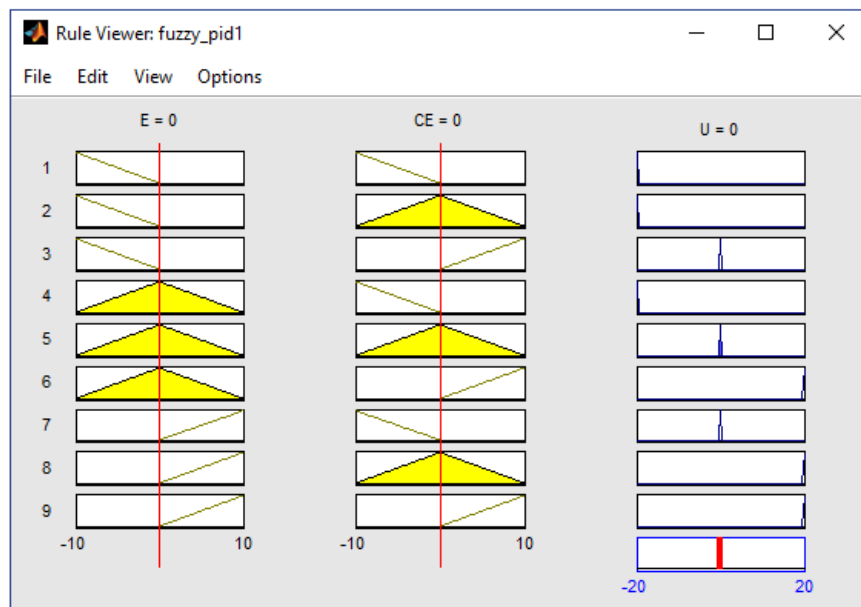
Otra parte importante es la regla base que está conformada por 9 reglas difusas propuestas por Mamdani.

1. If (E is N) and (CE is N) then (U is N) (1)
2. If (E is N) and (CE is Z) then (U is N) (1)
3. If (E is N) and (CE is P) then (U is Z) (1)
4. If (E is Z) and (CE is N) then (U is N) (1)
5. If (E is Z) and (CE is Z) then (U is Z) (1)
6. If (E is Z) and (CE is P) then (U is P) (1)
7. If (E is P) and (CE is N) then (U is Z) (1)
8. If (E is P) and (CE is Z) then (U is P) (1)
9. If (E is P) and (CE is P) then (U is P) (1)

Los resultados de la inferencia para las articulaciones 1 y 2 es mostrada en el editor de reglas del esquema de la Figura 6.11. En él se aprecia las variables del error (E) y cambio de error (CE) del lado izquierdo, mientras que del lado derecho esta la salida. Además, los pasos de fuzzificación, operación difusa, implicación, agregación y defuzzificación son evidentes en el momento de realizar la inferencia difusa.

Figura 6.11

Ventana de visualizador de reglas base.

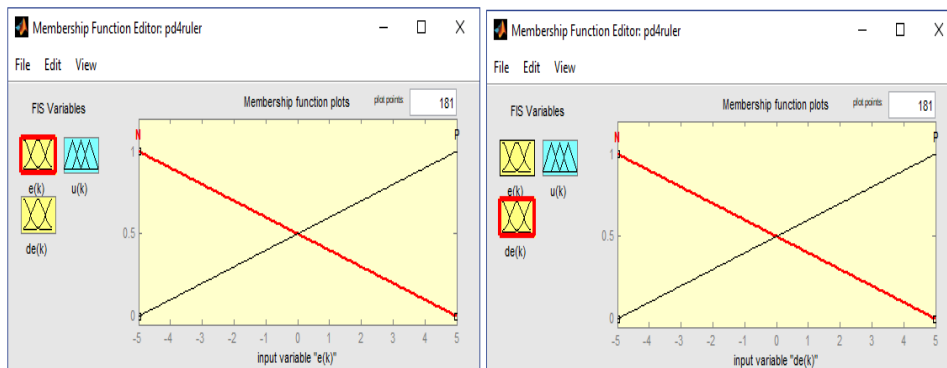


6.4.2 Inferencia Difusa Articulación 3

Las funciones de pertenencia para las entradas y salidas para la articulación 3 que corresponden al segundo brazo del robot de 3GDL. También presentan funciones de pertenencia triangulares y el universo de discurso, que pueden ser apreciados en el esquema de la Figura 6.12. Podemos ver las etiquetas lingüísticas denotas por Negativo (N), Cero (Z) y Positivo (P). El universo de discurso a considerar es de -5V a 5V que corresponde a la variable física.

Figura 6.12

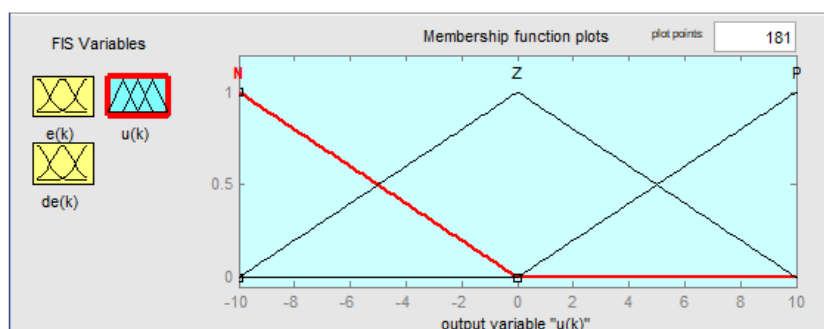
Funciones de pertenencia de entrada articulación 3.



La función de pertenencia aproximada de Singleton difusos a la salida es mostrada en la Figura 6.13. En este caso las etiquetas lingüísticas son las mismas que las entradas. El universo de discurso es el doble que el de la entrada (consideración importante en el diseño).

Figura 6.13

Funciones de pertenencia de salida.



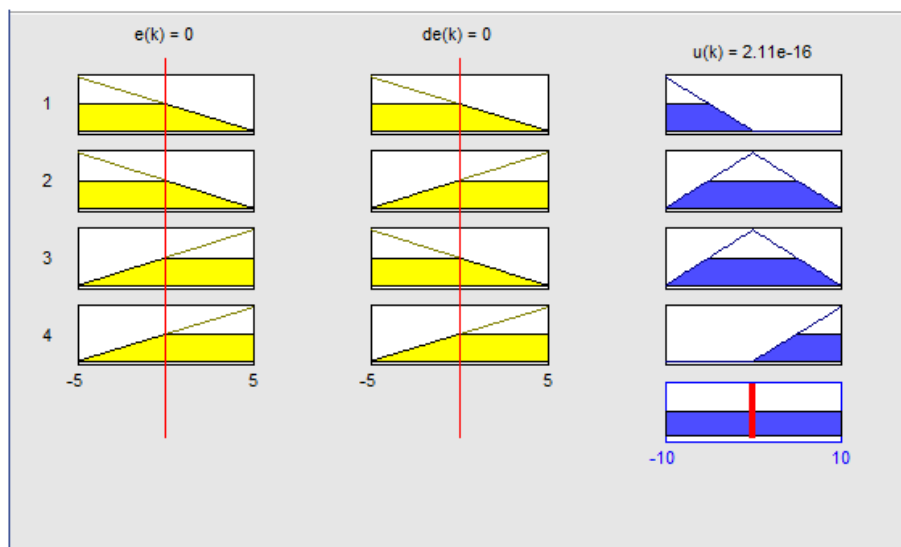
Otra parte importante es la regla base que está conformada por 4 reglas difusas propuestas por Mamdani.

1. If (e(k) is N) and (de(k) is N) then (u(k) is N) (1)
2. If (e(k) is N) and (de(k) is P) then (u(k) is Z) (1)
3. If (e(k) is P) and (de(k) is N) then (u(k) is Z) (1)
4. If (e(k) is P) and (de(k) is P) then (u(k) is P) (1)

Los resultados de la inferencia para la articulación 3, es mostrada en el editor de reglas del esquema de la Figura 6.14. En él se aprecia las variables del error (E) y cambio de error (CE) del lado izquierdo, mientras que del lado derecho esta la salida. Además, los pasos de fuzzificación, operación difusa, implicación, agregación y defuzzificación son evidentes en el momento de realizar la inferencia difusa.

Figura 6.14

Ventana de visualizador de reglas base articulación 3.



VII.- CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES

Concluimos esta tesis con un resumen corto de las contribuciones y algunas recomendaciones para investigaciones del futuro que son principalmente los problemas que no han sido resueltos en esta tesis.

- De la técnica desarrollada en la presente tesis, control difuso tipo PID, se concluye que ofrece un buen control en el resultado para el control angular simultáneo de las articulaciones del robot. Esto es posible a su etapa de abordar el conocimiento basado en reglas inherente a su algoritmo de control, el cual es resuelto fuera de línea.
- La técnica de control empleada requiere del modelo del motor de 3GDL, en este caso, nuestro modelo es bastante exacto ya que el algoritmo de control proporciona las señales de control que hacen que el sistema siga a las trayectorias impuestas con mínimo sobre impulso y error en estado estacionario nulo.
- Para las simulaciones se desarrolla el control difuso para el robot de 3GDL usando el software de simulación de MATLAB herramienta importante en la teoría de control ya que cuenta con las herramientas y comandos necesarios para realizar los algoritmos y simulaciones.

7.2 RECOMENDACIONES

- A partir de los resultados obtenidos del modelado y sintonización del controlador PID de robot puma de tres grados de libertad utilizando la lógica difusa se recomienda implementar el sistema de control.
- Se puede y debe optimizar el controlador Fuzzy, utilizando otros algoritmos.
- Se puede sugerir que para futuros trabajos de investigación de los sistemas de control donde los parámetros cambien constantemente se utilice la lógica difusa para optimización en el modelado y sintonización de los controladores PID aplicado a los robots puma.
- Se debe tener en cuenta la relación de masa- longitud para cada articulación, no exigir demasiado el movimiento del torque y los eslabones de la base.

VIII.- BIBLIOGRAFÍA

- [1] A. Barrientos. P. Balaguer. **Fundamentos de Robótica**. McGraw- Hill, Enero 2007.
- [2] Alberto Isidori. **Nonlinear Control Systems** (Second Edition). Springer-Verlag, Noviembre 1989.
- [3] Arturo Rojas Moreno. **Control No Lineal Multivariable: Aplicaciones en Tiempo Real**. Publicacion Independiente – Lima- Perú, Marzo 2007.
- [4] A. Vivas, V. Mosquera. **Predictive functional control of a puma robot**. ACSE 05 Conference, Cairo, Egypt, Volume 4, Number 2, 2005.
- [5] Bruno Siliciano, Lorenzo Sciavicco. **Robotcs: Modelling, plinning and control**. Springer, Julio 2008.
- [6] E.F.Camacho, C. Bordons. **Model Predictive Control** (Second Edition). Springer, Marzo 2003.
- [7] K.S. Fu, R.C. Gonzales, C.S.G Lee. **Robótica, Control, Detección, Visión, Inteligencia**. McGraw-Hill, Noviembre 1990.
- [8] Hedjar, R., Boucher, P. **Nonlinear receding-horizon control of rigid link robot manipulators**. **International Journal of Advanced Robotic Systems**, Volume 2, Number 1, 2005.
- [9] Ramdane Hedjar, Redouane Toumi, Patrick Boucher, Didier Dumu. **Finite Horizon Nonlinear Predictive Control by the Tylor Approximation: Application to robot tracking trajectory**. **International Journal of Applied Mathematics and Computer**, vol. 15, Number 4, 2005.
- [10] S. Joe Qin and Thomas A. Badgwell. **A survey of industrial model predictive control technology**. **Control Engineering Practice**, vol. 11, 2003.
- [11] Tokunbo Ogunfunmi. **Adaptive Nonlinear System Identification (The Volterra and Wiener Model Approaches)**. Springer, Septiembre 2006.

ANEXOS

ANEXO 1: Códigos MATLAB

Programa 1

El programa resuelve el cálculo de la ecuación dinámica del robot manipulador esférico de 3GDL aplicando el Algoritmo Lagrange-Euler desarrollado con matemática simbólica de MATLAB.

```
clear all; close all; clc
```

```
syms q1 q2 q3 m1 hp hd rm2 hd Lys1 m2 L2 Lcs2 Lxs2 m3 L3 Lcs3 Lxs3
```

```
syms g g1 I1 I2 I3 Dq1 Dq2 Dq3 X Y S1 C1 S2 C2 TT1
```

```
% -----
```

```
% A. Datos de los parámetros D-H
```

```
% -----
```

```
teta1=q1; teta2=q2; teta3=q3;
```

```
dDH1=hp+hd+rm2; dDH2=0; dDH3=0;
```

```
aDH1=0; aDH2=L2; aDH3=L3;
```

```
alfa1=90; alfa2=0; alfa3=0;
```

```
% -----
```

```
% B. Cálculo de las matrices de transformación: Aij
```

```
% -----
```

```
A01=[cos(q1) 0 sin(q1) 0
```

```
sin(q1) 0 -cos(q1) 0
```

```
0 1 0 hp+hd+rm2
```

```
0 0 0 1];
```

```
A12=[cos(q2) -sin(q2) 0 L2*cos(q2)
```

```
sin(q2) cos(q2) 0 L2*sin(q2)
```

```
0 0 1 0
```

```
0 0 0 1];
```

```
A02=A01*A12;
```

```

A23=[cos(q3) -sin(q3) 0 L3*cos(q3)
      sin(q3) cos(q3) 0 L3*sin(q3)
      0 0 1 0
      0 0 0 1];
A03=A02*A23;

% X=cos(q2)*cos(q3)-sin(q2)*sin(q3)=cos(q2+q3);
% Y=sin(q2)*cos(q3)+sin(q3)*cos(q2)=sin(q2+q3);

% -----
% C.- Cálculo de las matrices de transformación homogénea: Uij}
% -----

U11=diff(A01,'q1'); U12=diff(A01,'q2'); U13=diff(A01,'q3');
U21=diff(A02,'q1'); U22=diff(A02,'q2'); U23=diff(A02,'q3');
U31=diff(A03,'q1'); U32=diff(A03,'q2'); U33=diff(A03,'q3');

A = -sin(q1)*L3*X-sin(q1)*L2*cos(q2);
B = cos(q1)*L3*X+cos(q1)*L2*cos(q2);

U31=[-sin(q1)*X sin(q1)*Y cos(q1) A
      cos(q1)*X -cos(q1)*Y sin(q1) B
      0 0 0 0
      0 0 0 0];

U32=[-cos(q1)*Y -cos(q1)*X 0 -L3*cos(q1)*Y-L2*cos(q1)*sin(q2)
      -sin(q1)*Y -sin(q1)*X 0 -L3*sin(q1)*Y-L2*sin(q1)*sin(q2)
      X -Y 0 L3*X+L2*cos(q2)
      0 0 0 0];

```

```

U33=[-cos(q1)*Y -cos(q1)*X 0 -L3*cos(q1)*Y
      -sin(q1)*Y -sin(q1)*X 0 -L3*sin(q1)*Y
      X      -Y 0 L3*X
      0      0 0 0];

```

```
X = cos(q2)*cos(q3)-sin(q2)*sin(q3);
```

```
Y = sin(q2)*cos(q3)+sin(q3)*cos(q2);
```

```

UU31=[-sin(q1)*X sin(q1)*Y cos(q1) A
       cos(q1)*X -cos(q1)*Y sin(q1) B
       0      0 0 0
       0      0 0 0];

```

```

UU32=[-cos(q1)*Y -cos(q1)*X 0 -L3*cos(q1)*Y-L2*cos(q1)*sin(q2)
       -sin(q1)*Y -sin(q1)*X 0 -L3*sin(q1)*Y-L2*sin(q1)*sin(q2)
       X      -Y 0 L3*X+L2*cos(q2)
       0      0 0 0];

```

```

UU33=[-cos(q1)*Y -cos(q1)*X 0 -L3*cos(q1)*Y
       -sin(q1)*Y -sin(q1)*X 0 -L3*sin(q1)*Y
       X      -Y 0 L3*X
       0      0 0 0];

```

```
% -----
```

```
% D.- Matrices de seudoinercias: JJi}
```

```
% Para la base (disco m\s prisma): JJ1 }
```

```
% -----
```

```

JJ1=[I1 0 0 0
     0 -I1 0 -m1*Lys1
     0 0 I1 0
     0 -m1*Lys1 0 m1];

```

```
% Para la primera varilla: JJ2}
```

```
Lxs2=L2-Lcs2;
```

```
JJ2=[ I2 0 0 -m2*Lxs2
```

```
0 0 0 0
```

```
0 0 0 0
```

```
-m2*Lxs2 0 0 m2];
```

```
% Para la segunda varilla: JJ3}
```

```
Lxs3=L3-Lcs3;
```

```
JJ3=[ I3 0 0 -m3*Lxs3
```

```
0 0 0 0
```

```
0 0 0 0
```

```
-m3*Lxs3 0 0 m3];
```

```
% -----
```

```
% E. Calculo de la matriz de inercia simétrica: Hik(q)}
```

```
% -----
```

```
% Matriz H11
```

```
H11=simplify(trace(U11*JJ1*U11')) + ...
```

```
simple(simplify(trace(U21*JJ2*U21')))+ ...
```

```
simplify(trace(UU31*JJ3*UU31'));
```

```
% Matriz H12
```

```
H12=simplify(trace(U22*JJ2*U21')+trace(UU32*JJ3*UU31'));
```

```
% Matriz H13
```

```
H13=simplify(trace(UU33*JJ3*UU31'));
```

```
% Matriz H21
```

```
H21=simplify(trace(U21*JJ2*U22')+trace(UU31*JJ3*UU32'));
```

```
% Matriz H22
H22=simplify(trace(U22*JJ2*U22))+ simplify(trace(UU32*JJ3*UU32'));
```

```
% Matriz H23
H23=simplify(trace(UU33*JJ3*UU32'));
```

```
% Matriz H31
H31=simplify(trace(UU31*JJ3*UU33'));
```

```
% Matriz H32
H32=simplify(trace(UU32*JJ3*UU33'));
```

```
% Matriz H33
H33=simplify(trace(UU33*JJ3*UU33'));
```

```
% -----
% F. Cálculo de las matrices: Uijk }
% -----
```

```
U111=diff(U11,'q1'); U112=diff(U11,'q2'); U113=diff(U11,'q3');
U121=diff(U12,'q1'); U122=diff(U12,'q2'); U123=diff(U12,'q3');
U131=diff(U13,'q1'); U132=diff(U13,'q2'); U133=diff(U13,'q3');
U211=diff(U21,'q1'); U212=diff(U21,'q2'); U213=diff(U21,'q3');
U221=diff(U22,'q1'); U222=diff(U22,'q2'); U223=diff(U22,'q3');
U231=diff(U23,'q1'); U232=diff(U23,'q2'); U233=diff(U23,'q3');
U311=diff(U31,'q1'); U312=diff(U31,'q2'); U313=diff(U31,'q3');
U321=diff(U32,'q1'); U322=diff(U32,'q2'); U323=diff(U32,'q3');
U331=diff(U33,'q1'); U332=diff(U33,'q2'); U333=diff(U33,'q3');
```

```
% -----
% G. Cálculo del vector: hikm }
```

% -----

```
h111=simplify(trace(U111*JJ1*U11'))+...
    simple(simplify(trace(U211*JJ2*U21')));
h112=simple(simplify(trace(U212*JJ2*U21')));
h113=simple(trace(U313*JJ3*UU31'));
h121=simple(simplify(trace(U221*JJ2*U21')));
h122=simple(simplify(trace(U222*JJ2*U21')));
h123=simple(simplify(trace(U323*JJ3*U31')));
h131=simple(simplify(trace(U331*JJ3*UU31')));
h132=simple(simplify(trace(U332*JJ3*UU31')));
h133=simple(simplify(trace(U333*JJ3*UU31')));
h211=simple(simplify(trace(U211*JJ2*U22')));
h212=simple(simplify(trace(U212*JJ2*U22')));
h213=simple(simplify(trace(U313*JJ3*U32')));
h221=simple(simplify(trace(U221*JJ2*U22')));
h222=simple(simplify(trace(U222*JJ2*U22')));
hh223=simple(simplify(trace(U323*JJ3*UU32')));
h231=simple(simplify(trace(U331*JJ3*U32')));
hh232=simple(simplify(trace(U332*JJ3*UU32')));
hh233=simple(simplify(trace(U333*JJ3*UU32')));
hh311=simple(simplify(trace(U331*JJ3*UU33')));
h312=simple(simplify(trace(U312*JJ3*UU33')));
h313=simple(simplify(trace(U313*JJ3*UU33')));
h321=simple(simplify(trace(U321*JJ3*UU33')));
hh322=simple(simplify(trace(U322*JJ3*UU33')));
h323=simple(simplify(trace(U323*JJ3*UU33')));
h331=simple(simplify(trace(U331*JJ3*UU33')));
h332=simple(simplify(trace(U332*JJ3*UU33')));
h333=simple(simplify(trace(U333*JJ3*UU33')));
X = cos(q2)*cos(q3)-sin(q2)*sin(q3);
```

```
Y = sin(q2)*cos(q3)+sin(q3)*cos(q2);
```

```
h113=simple(hh113);
```

```
h223=simple(simplify(hh223));
```

```
h232=h223;
```

```
h233=h232;
```

```
h311=simple(simplify(hh311));
```

```
h322=simple(simplify(hh322));
```

```
% -----
```

```
% H. Cálculo del vector de fuerza Centrifuga y de
```

```
% Coriolis: Ci(q,dq)}
```

```
% -----
```

```
C1=simple (h111*Dq1*Dq1 + h112*Dq1*Dq2 + h113*Dq1*Dq3 + ...  
          h121*Dq1*Dq2 + h122*Dq2*Dq2 + h123*Dq2*Dq3 + ...  
          h131*Dq1*Dq3 + h132*Dq2*Dq3 + h133*Dq3*Dq3);
```

```
C2=simple (h211*Dq1*Dq1 + h212*Dq1*Dq2 + h213*Dq1*Dq3 + ...  
          h221*Dq1*Dq2 + h222*Dq2*Dq2 + h223*Dq2*Dq3 + ...  
          h231*Dq1*Dq3 + h232*Dq2*Dq3 + h233*Dq3*Dq3);
```

```
C3=simple (h311*Dq1*Dq1 + h312*Dq1*Dq2 + h313*Dq1*Dq3 + ...  
          h321*Dq1*Dq2 + h322*Dq2*Dq2 + h323*Dq2*Dq3 + ...  
          h331*Dq1*Dq3 + h332*Dq2*Dq3 + h333*Dq3*Dq3);
```

```
% -----
```

```
% I. Calculo del vector de fuerza gravitacional: di(q)}
```

```
% -----
```

```
r11=[0; -Lys1; 0; 1];
```

```
r22=[-Lxs2; 0; 0;1];
```

```
r33=[-Lxs3; 0; 0;1];
```

```
g1=[0 0 -g 0];
```



```

d1=-m1*g1*U11*r11 - m2*g1*U21*r22 - m3*g1*UU31*r33;
d2=simplify(-m2*g1*U22*r22 - m3*g1*UU32*r33);
d3=simplify(-m3*g1*UU33*r33);

% -----
% J. Parámetros del sistema
% -----

a1=2*I1;
a2=I2 + 2*m2*L2*Lcs2 - m2*L2*L2;
a3=I3 + 2*m3*L3*Lcs3 - m3*L3*L3;
a4=m3*L2*Lcs3;
a5=m3*L2*L2;
a6=m2*g*Lcs2 + m3*g*L2;
a7=m3*g*Lcs3;

% -----
% K. Resultados: Matriz de inercia simétrica
% -----

H11=a1+a2*cos(q2)*cos(q2)+a3*cos(q2+q3)*cos(q2+q3)+ ...
2*a4*cos(q2)*cos(q2+q3)+a5*cos(q2)*cos(q2);
H12=0;
H13=0;
H21=0;
H22=a2+a3+2*a4*cos(q3)+a5;
H23=a3+a4*cos(q3);
H31=0;
H32=H23;
H33=a3;

% -----

```

```
% Vector de fuerza centrífuga y Coriolis
```

```
% -----
```

```
C1=-2*a2*sin(2*q2)*Dq1*Dq2-2*(a3*sin(q2+q3)*cos(q2+q3))+ ...  
a4*sin(2*q2+2*q3)*cos(q2)*Dq1*Dq3;
```

```
C2=a2*sin(q2)*cos(q2)*Dq1*Dq1-2*a4*sin(q3)*Dq2*Dq2*Dq3*Dq3 - ...  
a4*sin(q3)*Dq3*Dq3;
```

```
C3=(a3*sin(q2+q3)*cos(q2+q3)+a4*sin(2*q2+2*q3)*cos(q2))*Dq1*Dq1 + ...  
a4*sin(q3)*Dq2*Dq2;
```

```
% -----
```

```
% Vector de fuerza gravitacional
```

```
% -----
```

```
g1=0;  
g2=a6*cos(q2)+a7*cos(q2+q3);  
g3=a7*cos(q2+q3);
```

Programa de control en lazo abierto

```
clear all; close all; clc  
hd = 0.01;  
Rd = 0.07;  
m_dt = 0.55;  
hp = 0.21;  
ap = 0.044;  
m_pt = 2.475-0.072;  
m_hueco = 0.072;  
R_hueco = 0.063;  
I1=0.5*(m_pt*(ap^2)/6 + m_dt*(Rd^2)/2-m_hueco*(R_hueco^2)/2)+0.00059;
```

```

m2=0.25+0.5+0.05;
mo=0.025;
L2=0.20;
Lcs2=0.10;
I2=m2*(L2*L2)/3;
m3=0.25;
mo=0.025;
L3=0.20;
Lcs3=0.10;
Io=mo*(ap*ap/6);
Iv=m3*(L3*L3)/3;
I3=Iv+Io;
g =9.81;
% Datos del sub-sistema eléctrico
Km1=0.04364; Kb1=0.04364; Ra1=3.5; Jm1=0.00059; n1=18.5; KA1=2.17;
Km2=0.04364; Kb2=0.04364; Ra2=3.5; Jm2=0.00059; n2=18.5; KA2=2.17;
Km3=0.04217; Kb3=0.04217; Ra3=5.3; Jm3=0.000238; n3=12.5; KA3=2.17;
Cc1=0.001488; Cc3=0.00015;
% Carga
Jeq1 = 0.2177; Beq1 = 0.054;
Jeq2 = 0.3954; Beq2 = 0.0589;
Jeq3 = 0.000305; Beq3 = 0.000538;
% Equivalencia de resistencias
RT1 = Ra1/(n1*Km1*KA1);
RT2 = Ra2/(n2*Km2*KA2);
RT3 = Ra3/(n3*Km3*KA3);
% Equivalencia de N
NT1 = n1*Kb1/KA1;
NT2 = n2*Kb2/KA2;
NT3 = n3*Kb3/KA3;
% Parámetros del Robot

```

```

a1=2*I1;
a2=I2 + 2*m2*Lcs2*L2 - m2*L2*L2;
a3=I3 + 2*m3*Lcs3*L3 - m3*L3*L3;
a4=m3*Lcs3*L2;
a5=m3*L2*L2;
a6=m2*g*Lcs2+m3*g*L2;
a7=m3*g*Lcs3;
% Condiciones y parámetros iniciales
p1(1)=0; p1(2)=0; p1(3)=0; p1(4)=0;
dp1(1)=0; dp1(2)=0; dp1(3)=0; dp1(4)=0;
ddp1(1)=0; ddp1(2)=0; ddp1(3)=0; ddp1(4)=0;
p2(1)=0; p2(2)=0; p2(3)=0; p2(4)=0;
dp2(1)=0; dp2(2)=0; dp2(3)=0; dp2(4)=0;
ddp2(1)=0; ddp2(2)=0; ddp2(3)=0; ddp2(4)=0;
p3(1)=0; p3(2)=0; p3(3)=0; p3(4)=0;
dp3(1)=0; dp3(2)=0; dp3(3)=0; dp3(4)=0;
ddp3(1)=0; ddp3(2)=0; ddp3(3)=0; ddp3(4)=0;
x1(1)=0;x1(2)=0;x1(3)=0;x1(4)=0;
x2(1)=0;x2(2)=0;x2(3)=0;x2(4)=0;
x3(1)=0;x3(2)=0;x3(3)=0;x3(4)=0;
x4(1) =0;x4(2)=0;x4(3)=0;x4(4)=0;
x5(1)=0; x5(2)=0; x5(3)=0; x5(4)=0;
x6(1)=0; x6(2)=0; x6(3)=0; x6(4)=0;
% Tiempo de muestreo
Nm=2000; T =0.010;
% sistema de control en lazo abierto
for k=4:Nm;
    % Trayectorias senoidales deseadas: p1, p2 y p3
    p1(k)=pi*cos(2*pi*k*T/11);
    p2(k)=3*pi/2*cos(2*pi*k*T/11);
    p3(k)=2*pi*cos(2*pi*k*T/11);

```

```

% Transformación de coordenadas
z1(k)=x1(k);
z2(k)=x4(k);
z3(k)=x2(k);
z4(k)=x5(k);
z5(k)=x3(k);
z6(k)=x6(k);

% Aproximación de derivadas para discretización
dp1(k)=(p1(k)+3*p1(k-1)-3*p1(k-2)-p1(k-3))/(6*T);
ddp1(k)=(dp1(k)+ 3*dp1(k-1)-3*dp1(k-2)-dp1(k-3))/(6*T);
dp2(k)=(p2(k)+ 3*p2(k-1)-3*p2(k-2)-p2(k-3))/(6*T);
ddp2(k)=(dp2(k)+ 3*dp2(k-1)-3*dp2(k-2)-dp2(k-3))/(6*T);
dp3(k)=(p3(k)+3*p3(k-1)-3*p3(k-2)-p3(k-3))/(6*T);
ddp3(k)=(dp3(k)+3*dp3(k-1)-3*dp3(k-2)-dp3(k-3))/(6*T);

% Matriz de inercias Hij:
H11(k)=a1+a2*cos(z3(k))*cos(z3(k))+...
    a3*cos(z3(k)+z5(k))*cos(z3(k)+z5(k))+...
    2*a4*cos(z3(k))*cos(z3(k)+z5(k))+a5*cos(z3(k))*cos(z3(k));
H22(k)=a2+a3+2*a4*cos(z5(k))+a5;
H23(k)=a3+a4*cos(z5(k));
H32(k)=H23(k);
H33(k)=a3;

% Vector de fuerzas centrífuga y de Coriolis Ci:
C1(k)=-a2*z2(k)*z4(k)*sin(2*z3(k))-...
    2*a3*z2(k)*z6(k)*sin(z3(k)+z5(k))*cos(z3(k)+z5(k))...
    -2*a4*z2(k)*z6(k)*sin(z3(k)+z5(k))*cos(z3(k));
C2(k)=a2*z2(k)*z2(k)*sin(z3(k))*cos(z3(k))-...
    a4*z6(k)*z6(k)*sin(z5(k))-2*a4*z4(k)*z6(k)*sin(z5(k));
C3(k)=a3*z2(k)*z2(k)*sin(z3(k)+z5(k))*cos(z3(k)+z5(k))+...
    a4*z2(k)*z2(k)*cos(z3(k))*sin(z3(k)+z5(k))+...
    a4*z4(k)*z4(k)*sin(z5(k));

```

```

% Vector de fuerza gravitacional di:
d1(k)=0;
d2(k)=a6*cos(z3(k))+a7*cos(z3(k)+z5(k));
d3(k)=a7*cos(z3(k)+z5(k));
% Primer actuador: base
u1(k) = RT1*(H11(k)+Jeq1);
% Segundo actuador: primera varilla
u2(k) = RT2*(H22(k)+Jeq2) +RT2*H23(k);
% Tercer actuador: segunda varilla
u3(k) =RT3*H32(k) + RT3*(H33(k)+Jeq3);
% Proceso de discretizacion
Tc1=Cc1*sign(x4(k));
Tc2=Cc1*sign(x5(k));
Tc3=Cc3*sign(x6(k));
x1(k+1)=x1(k)+T*x4(k);
x4(k+1)=x4(k)+(T*(-n1*RT1*Tc1+u1(k)))/(RT1*(H11(k)+Jeq1));
Det(k)=RT2*RT3*((H22(k)+Jeq2)*(H33(k)+Jeq3)-H23(k)*H23(k));
x2(k+1)=x2(k)+T*x5(k);
x5(k+1)=x5(k)+(T/Det(k))*(RT3*(H33(k)+Jeq3)*u2(k)-RT2*H23(k)*u3(k)-...
RT3*(H33(k)+Jeq3)+RT2*H23(k)-...
RT3*(H33(k)+Jeq3)*n2*RT2*Tc2+RT2*H23(k)*n3*RT3*Tc3);
x3(k+1)=x3(k)+T*x6(k);
x6(k+1)=x6(k)+(T/Det(k))*(RT2*(H22(k)+Jeq2)*u3(k)-RT3*H32(k)*u2(k)-...
RT2*(H22(k)+Jeq2)+RT3*H32(k)-...
RT2*(H22(k)+Jeq2)*n3*RT3*Tc3+RT3*H32(k)*n2*RT2*Tc2);
end
% Simulaciones
ejex = linspace(0,Nm*T,Nm-3);
figure
subplot(311)
plot(ejex,p1(4:Nm)*180/pi,ejex,x1(4:Nm)*180/pi,'k:'), grid

```

```

ylabel('q_1')
legend('q_{d1}','q_1',-1)
subplot(312)
plot(ejex,p2(4:Nm)*180/pi,ejex,x2(4:Nm)*180/pi,'k:'), grid
ylabel('q_2')
legend('q_{d2}','q_2',-1)
subplot(313)
plot(ejex,p3(4:Nm)*180/pi,ejex,x3(4:Nm)*180/pi,'k:'), grid
ylabel('q_3')
xlabel('t [seg]')
legend('q_{d3}','q_3',-1)
figure
subplot(311)
plot(ejex,x4(4:Nm)), grid
ylabel(['Rad/Seg.'])
subplot(312)plot(ejex,x5(4:Nm)), grid
ylabel(['Rad/Seg.'])
subplot(313)
plot(ejex,x6(4:Nm)), grid
ylabel(['Rad/Seg.']), xlabel('t [seg]')
figure
subplot(311)
plot(ejex,u1(4:Nm)), grid
ylabel(['Voltios'])
subplot(312)
plot(ejex,u2(4:Nm)), grid
ylabel(['Voltios'])
subplot(313)
plot(ejex,u3(4:Nm)), grid
ylabel(['Voltios']), xlabel('t [seg]')

```

ANEXO 2: MATRIZ DE CONSISTENCIA DE LA TESIS DE MAESTRÍA EN LA UNIVERSIDAD NACIONAL DEL CALLAO.

Título de la Tesis : “MODELADO Y SINTONIZACIÓN DE UN CONTROLADOR PID DE UN ROBOT TIPO PUMA DE TRES GRADOS DE LIBERTAD UTILIZANDO LÓGICA DIFUSA”			
PROBLEMAS	OBJETIVOS	HIPÓTESIS	VARIABLES E INDICADORES
<p>1.2.1. Problema General: ¿Cómo se evalúa el desempeño de la lógica difusa en el modelado y la sintonización de un controlador PID de un robot tipo PUMA de tres grados de libertad.</p> <p>Problemas Específicos: a.- ¿Cómo se elabora un algoritmo de inferencia difusa? b.- ¿Cómo se sintoniza el controlador PID del robot tipo PUMA de tres grados de libertad utilizando lógica difusa? c.- ¿Cómo se simula el controlador del robot PUMA de tres grados de libertad, con la finalidad de verificar el desempeño de la lógica difusa utilizada en la sintonización del controlador PID libertad?</p>	<p>1.3.1. Objetivo General: Evaluar el desempeño de la lógica difusa en la sintonización de un controlador PID de un robot tipo PUMA de tres grados de libertad.</p> <p>Objetivos Específicos: a.- Elaborar el algoritmo de lógica difusa para la sintonización de un controlador PID. b.- Sintonizar el controlador PID del robot tipo PUMA de tres grados de libertad utilizando lógica difusa. c.- Simular el controlador del robot PUMA de tres grados de libertad, con la finalidad de verificar el desempeño de la lógica difusa utilizada en la sintonización del controlador PID.</p>	<p>3.3.1. Hipótesis General En función de las interrogantes planteadas del problema, así como de los objetivos generales y específicos que se persigue, el siguiente trabajo de tesis plantea la siguiente hipótesis: Es posible el modelado y sintonización del controlador PID de un Robot Tipo PUMA de tres grados libertad utilizando lógica difusa.</p> <p>Hipótesis Específicas. a.- Es posible la elaboración del algoritmo de lógica difusa para el modelado y la sintonización de un controlador PID. b.- Es posible la sintonización del controlador PID del robot tipo PUMA de tres grados de libertad utilizando lógica difusa. c.- Es posible la simulación del controlador del robot PUMA de tres grados de libertad, con la finalidad de verificar el desempeño de la lógica difusa utilizada en la sintonización del controlador PID.</p>	<p>Operacionalización de variables: La eficiencia computacional de X, mejorará la precisión de Y</p> <p>X: Sistema de control del robot tipo PUMA.</p> <p>Indicadores X1: K_p del controlador del robot tipo PUMA. X2: T_i del controlador del robot tipo PUMA. X3 : T_d del controlador del robot tipo PUMA</p> <p>Y: Lógica difusa Indicadores: Y1 : Conjunto difusos Y2 : Función de pertenencia Y3 : Reglas heurísticas</p>

Tabla 2.2 Parámetros de la Planta No – Lineal

u_1, u_2, u_3	Señal de control servo DC1, DC2, DC3	(En simulación) V	J_{cq1}	M.I. equivalente servomotor DC1	0.2177 kgm2
g	Aceleración de la gravedad	9.81 m/seg ²	B_{cq1}	C.F.V. equivalente servomotor DC1	0.054 Nms/rad
Γ_{m2}	Distancia extremo superior del prisma al DC2	0.05 m	J_{cq2}	M.I. equivalente servomotor DC2	0.3954 Kgm2
C_{c1}	Coefficiente fricción de Coulumb 1 y 2DCF	0.001488 kgm2/seg	B_{cq2}	C.F.V. equivalente servomotor DC2	0.0589 Nms /rad
C_{c2}	Coefficiente fricción de Coulumb 3 DOF	0.00015 kgm2/seg	J_{cq3}	M.I. equivalente servomotor DC3	0.000305 kgm2
m_d	Masa del disco	0.55 kg	B_{cq3}	C.F.V. equivalente servomotor DC3	0.000538 Nms/rad
m_p	Masa del prisma	1.0 kg	n_1, n_2	Reducción servomotor DC1, DC2	18.5
m_2, m_3	Masa de la 1ra. y 2da. Varilla	0.20 kg	n_3	Reducción servomotor DC3	12.5
h_d	Altura del disco	0.01 m	R_{a1}, R_{a2}	Resist armadura servomotor DC1, DC2	3.50 Ohmios
R_d	Radio del disco	0.07 m	R_{a3}	Resist armadura servomotor DC3	5.30 Ohmios
h_p	Altura del prisma	0.20 m	L_{a1}, L_{a2}	Inductancia armadura servom DC1, DC2	Despreciable
Q_p	Ancho y largo de la sección del prisma	0.44 m	L_{a3}	Inductancia armadura servom DC3	Despreciable
L_2, L_3	Longitud de la 1ra. y 2da. Varilla	0.2 m	V_{a1}, V_{a2}	Voltaje armadura servomotor DC1, DC2	24V
L_{xc2}	Long. del eje de ref. S1 al CG 1ra. Varilla	0.14 m	V_{a3}	Voltaje armadura servomotor DC3	24V
L_{xc2}	Long. del eje de ref. S2 al CG 2da. Varilla	0.14 m	K_{A1}, K_{A2}, K_{A3}	Constantancia ganancia driver 1, 2, 3	2.17
L_{xc2}	Long. del eje de ref. S2 al CG 1ra. Varilla	(L2 – Lxs2) m	K_{m1}, K_{m2}	Cte. Torque servomotor DC1, DC2	0.04364 N.m/A
L_{xc2}	Long. del eje de ref. S2 al CG 2da. Varilla	(L3 – Lxs3) m	K_{m3}	Cte. Torque servomotor DC3	0.04217 N.m/A
I_1	M.I. del sólido (disco más prisma)	0.0013 kgm2	K_{b1}, K_{b2}	Cte. Fem servomotor DC1, DC2	0.04364 V/rad/s
I_2	M.I. de la 1ra. Varilla	0.0273 kgm2	K_{b3}	Cte. Fem servomotor DC3	0.04217 V/rad/s
I_3	M.I. de la 2da. Varilla	0.0273 kgm2			
J_{m1}, J_{m2}	M.I. servomotor DC1, DC2	0.00059 kgm2			
J_{m3}	M.I. servomotor DC3	0.000238 kgm2			

