

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

**“Control Difuso Takagi-Sugeno y Servovisual
aplicado a Robótica ”**

POR

Ubaldo Nájera Ruelas

TESIS

**PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL
GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA**

DIRECTOR DE TESIS

Dr. Miguel Ángel Llama Leal

ISSN: 0188-9060



RIITEC: (17)-TMCIE-2015

Torreón, Coahuila, México
Diciembre 2015



"2015, Año del Generalísimo José María Morelos y Pavón"

Torreón, Coah., **10/Diciembre/2015**
Dependencia: DEPI/CPCIE
Oficio: DEPI/CPCIE/233/2015
Asunto: Autorización de impresión
de tesis.

C. Ubaldo Nájera Ruelas
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA,
PRESENTE

Después de haber sometido a revisión su trabajo de tesis titulado:

"Control Difuso Takagi-Sugeno y Servovisual aplicado a Robótica"

Habiendo cumplido con todas las indicaciones que el jurado revisor de tesis hizo, se le comunica que se le concede la autorización con número de registro **RIITEC: (17)-TMCIE-2015**, para que proceda a la impresión del mismo.

ATEMENTAMENTE
EDUCACIÓN TECNOLÓGICA PUENTE DE INNOVACIÓN

DR. JOSÉ LUIS MEZA MEDINA
Jefe de la División de Estudios de Posgrado e Investigación
del Instituto Tecnológico de la Laguna

SECRETARÍA DE
EDUCACIÓN PÚBLICA
INSTITUTO TECNOLÓGICO
de la Laguna
División de Estudios de Posgrado e
Investigación

ILM/v/RSR





DR. JOSE LUIS MEZA MEDINA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Por medio de la presente, hacemos de su conocimiento que después de haber sometido a revisión el trabajo de tesis titulado:

"Control Difuso Takagi-Sugeno y Servovisual aplicado a Robótica"

Desarrollado por el **C. Ubaldo Nájera Ruelas**, con número de control **M1313063** y habiendo cumplido con todas las correcciones que se le indicaron, estamos de acuerdo que se le conceda la autorización de la fecha de examen de grado para que proceda a la impresión de la misma.

ATENTAMENTE
EDUCACIÓN TECNOLÓGICA FUENTE DE INNOVACIÓN



Dr. Miguel A. Llama Leal
Asesor/Director de Tesis



Dr. Juan Sifuentes Mijares
Comité Tutorial



Dr. Francisco Jurado Zamarripa
Comité Tutorial



Dr. Victor A. Santibañez Dávila
Comité Tutorial



Dedicatoria

A mi esposa, por ser el apoyo incondicional a cada instante durante todo el transcurso de esta maestría.

A mis hijos, Giovanna, Bárbara y Mateo, por ser la inspiración que llena mi vida de alegría.

A mi madre, por quien soy la persona que soy, por siempre estar y cuidar de nosotros desde un mejor lugar.

Agradecimientos

Primeramente quiero dar gracias a Dios, por darme la oportunidad de vivir, por cuidarme, por sacarme adelante y estar conmigo en todo momento.

A CONACYT y a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de la Laguna, por abrirme sus puertas y darme la oportunidad de crecer en mi formación académica.

A mi asesor de tesis, el Dr. Miguel Ángel Llana Leal, por la ayuda recibida para el desarrollo de este proyecto, por los consejos otorgados y por la paciencia para lograr obtener los resultados esperados.

Al comité tutorial formado por el Dr. Víctor Adrián Santibáñez Dávila, el Dr. Francisco Jurado Zamarripa y el Dr. Juan Sifuentes Mijares, por la asesoría y revisión de esta tesis.

Al resto del personal docente de la División de Estudios de Posgrado e Investigación del área de Mecatrónica y Control del Instituto Tecnológico de la Laguna, por transmitirme su conocimiento y la pasión por el estudio.

Resumen

En este trabajo de tesis se presenta el diseño e implementación de un controlador servovisual difuso Takagi Sugeno, el cual ha sido desarrollado tanto en simulación como en experimentación y aplicado al robot manipulador Mitsubishi PA10-7 CE limitando su movilidad a 2 grados de libertad, con el fin de realizar tareas de regulación y seguimiento, agregando una realimentación visual. Los cálculos de las ganancias del controlador, en su parte difusa, se realizan mediante la técnica de Desigualdades Lineales Matriciales, con la que se puede garantizar la estabilidad del sistema en lazo cerrado.

Abstract

This thesis presents the design of a Takagi Sugeno fuzzy servovisual controller, which has been developed in simulation and experimentation, applied to the Mitsubishi PA10-7 CE robot manipulator, limiting its mobility to 2 degrees of freedom, with the main goal of performing regulation and tracking tasks, by adding a visual feedback. The gains (fuzzy part) are obtained using the Linear Matrix Inequalities technique, which guarantees the stability of the closed-loop system.

Índice General

1	Introducción	1
1.1	Objetivo de la tesis.....	2
1.2	Estructura de la tesis.....	2
2	Modelado difuso	4
2.1	Modelo difuso Takagi-Sugeno.....	5
2.1.1	Series de Taylor.....	7
2.2	Compensación paralela distribuida (<i>Parallel Distributed Compensation, PDC</i>).....	8
2.3	Desigualdades lineales matriciales (<i>Linear Matrix Inequalities, LMIs</i>).....	9
2.3.1	Problema LMI.....	10
2.3.2	Problema de valores propios.....	10
2.3.3	Problema de valores propios generalizado.....	11
2.4	Diseño de un controlador PDC para regulación.....	11
2.5	Diseño de un controlador PDC para seguimiento.....	13
3	Mitsubishi PA10-7CE	15
3.1	Componentes del robot Mitsubishi PA10-7CE.....	16
3.1.1	Nivel 1: Mecanismo.....	16
3.1.2	Nivel 2: Controlador de servos.....	17
3.1.3	Nivel 3: Control de movimiento.....	18
3.1.4	Nivel 4: Control de operación.....	18
3.2	Modelado del Mitsubishi PA10-7CE.....	21
3.2.1	Modelo cinemático.....	22
3.2.2	Modelo dinámico.....	25
4	Control difuso Takagi-Sugeno (T-S)	28
4.1	Linealización del modelo dinámico del Mitsubishi PA10.....	28
4.2	Modelo difuso Takagi-Sugeno de la Planta.....	32
4.3	Método de regulación.....	34
4.3.1	Controlabilidad del sistema.....	34
4.3.2	Cálculo de ganancias.....	36
4.4	Método de seguimiento.....	38
4.4.1	Sistema aumentado por el modelo del servocompensador.....	39
4.4.2	Controlabilidad del sistema.....	42

4.4.3 Cálculo de ganancias	45
5 Visión	47
5.1 Sistemas de Visión.....	48
5.2 Adquisición de imágenes en tiempo real.....	49
5.3 Procesamiento de imágenes	50
5.3.1 Separación por color y filtrado de remanentes.....	50
5.3.2 Segmentación.....	51
5.3.3 Detección del centroide	51
5.4 Control servovisual.....	52
5.5 Separación de colores	55
6 Resultados	59
6.1 Simulación control difuso T-S.....	59
6.1.1 Regulación	59
6.1.2 Seguimiento.....	62
6.2 Experimentales control difuso T-S	66
6.2.1 Seguimiento.....	67
6.3 Experimentales control servovisual	70
6.3.1 Seguimiento.....	70
7 Conclusiones y trabajo futuro	73
7.1 Conclusiones	73
7.2 Trabajo futuro	74
8 Apéndice	75
A. Parámetros del robot Mitsubishi PA10-7CE	75
B. Programas en MATLAB.....	76
C. Bloque "PA10 Control"	93
9 Bibliografía.....	96

Índice de Figuras

Figura 1.- Sistema difuso tipo Mamdani	4
Figura 2.- Linealización de un sistema	5
Figura 3.-Niveles funcionales actuales del Mitsubishi PA10-7CE	16
Figura 4.- Mecánica del Mitsubishi PA10-7CE	17
Figura 5.- Caja de control de servomotores del PA10-7	18
Figura 6.- Tarjeta ARCNET PCI20U-4000	18
Figura 7.- Interfaz gráfica de control para el PA10 en Simulink	20
Figura 8.- Articulaciones funcionales PA10	21
Figura 9.- Selección de marcos para el PA10.....	24
Figura 10.- Funciones de membresía del sistema.....	30
Figura 11.- Diagrama a bloques del controlador PID con una planta lineal Takagi-Sugeno.....	33
Figura 12.- Posición de la articulación 2 con el modelo difuso de la planta	33
Figura 13.- Posición de la articulación 4 con el modelo difuso de la planta	34
Figura 14.- a) Sistema con cámara en mano; b) sistema con cámara fija	48
Figura 15.- a) Patrón original; b) aislamiento por umbralización; c) erosión y dilatación... ..	50
Figura 16.- Segmentación de imágenes	51
Figura 17.- Ubicación de centroide en una figura	51
Figura 18.- Cámara "HP HD-3110"	52
Figura 19.- Bloque para la adquisición de imágenes en tiempo real	53
Figura 20.- Ventana "video capture"	54
Figura 21.- Diagrama a bloques del sistema.....	54
Figura 22.- Segmentación por colores a) imagen original, b) segmentación color azul, c) segmentación color verde	56
Figura 23.- Bloque "Rate Transition".....	57
Figura 24.- Marcos establecidos para el sistema	57
Figura 25.- Diagrama a bloques del controlador difuso T-S para regulación	59
Figura 26.- Resultados regulación articulación 2	61
Figura 27.- Resultados regulación articulación 4	62
Figura 28.- Diagrama a bloques del controlador difuso T-S para seguimiento.....	63
Figura 29.- Resultados seguimiento articulación 2	65
Figura 30.- Resultados seguimiento articulación 4	66
Figura 31.- Resultados seguimiento en tiempo real para la articulación 2.....	68
Figura 32.- Resultados seguimiento en tiempo real para la articulación 4.....	70
Figura 33.- Resultados seguimiento en tiempo real con realimentación visual para la articulación 2	72

Índice de Tablas

Tabla 1.- Parámetros Denavith-Hartenberg (D-H) Mitsubishi PA10 limitado a 2 gdl.....	24
Tabla 2.- Segmentación de la imagen.....	55

Capítulo 1

Introducción

La mayor parte de los sistemas que conocemos y utilizamos en nuestra vida diaria son de naturaleza no lineal. Sin embargo, la mayoría de estos sistemas pueden ser representados a través de sistemas lineales. Un ejemplo de esto es a través de los modelos difusos Takagi-Sugeno, los cuales tienen la capacidad de aproximar un sistema no lineal mediante la linealización alrededor de un punto de operación, o como en algunos otros casos, en donde no es bien conocido el modelo del sistema se realiza una identificación alrededor de las señales de entrada y salida [18]. En [19], la estabilidad del sistema de control difuso Takagi-Sugeno ha sido verificada considerando una función de Lyapunov común mediante el uso del recurso de Desigualdades Lineales Matriciales (*Linear Matrix Inequalities*, LMIs).

Un robot manipulador es una plataforma adecuada para hacer pruebas experimentales, ya que presenta grandes inestabilidades en lazo abierto debido a su condición de sistema no lineal. Existen algunas limitantes para el control de robots manipuladores, como lo son los movimientos de cada articulación y su respectivo alcance. En la mayoría de las aplicaciones de control de robots, éstos se realizan cerrando el lazo a través de algún tipo de sensor localizado en las articulaciones del robot, comúnmente un *encoder* digital. Otra alternativa puede ser mediante la utilización de cámaras de visión para obtener la posición del órgano terminal requerida por el controlador de trayectorias del robot. A este tipo de control que utiliza información visual para cerrar el lazo de control se le conoce como "control servovisual" ("visual servoing" en inglés). Una de las ventajas de este tipo de control es que puede trabajar en ambientes no estructurados.

En la literatura existen diferentes metodologías para realizar el control servovisual difuso. En [16] se diseña e implementa un control PD para controlar un robot de 7 grados de libertad (gdl) usando la metodología Takagi-Sugeno. En vez de realizar la identificación del sistema (Jacobianos de imagen y cinemática), se desarrolla un controlador que es función del error y su derivada en el espacio visual deseado. Las ganancias PD son obtenidas con la ayuda de un controlador basado en el modelo, y adaptadas en línea con ayuda de un esquema propuesto que es capaz de compensar las incertidumbres locales asociadas con el sistema y su ambiente. El sistema es capaz de alcanzar un objetivo de punto fijo o uno en movimiento. En [11] un modelo de visión simple basado en 3D es propuesto y un controlador difuso basado sólo en información visual es desarrollado para controlar un robot manipulador.

En [1] se utiliza una metodología basada en el modelo difuso Takagi-Sugeno del robot para abordar el problema de seguimiento servovisual. Para conocer su velocidad se propone el diseño e implementación de un observador difuso. El controlador es implementado utilizando el método de compensación paralela distribuida (PDC, por sus siglas en inglés, *Parallel Distributed Compensation*). El diseño se hace directamente en coordenadas de imagen. Se reporta el uso de dos cámaras y su aplicación a un robot de 6 gdl, aunque sólo se utilizan las tres primeras articulaciones.

En el presente trabajo de tesis se proponen esquemas de control para un robot manipulador de 2 gdl utilizando el recurso del modelado difuso Takagi-Sugeno bajo el enfoque de la metodología de Series de Taylor. El modelo difuso propuesto por Takagi y Sugeno está descrito por reglas difusas del tipo SI-ENTONCES (IF-THEN), las cuales representan la relación entrada-salida de un sistema no lineal.

Siguiendo la metodología en cuestión, se diseña un controlador por realimentación de estados para cada subsistema local, mismos que serán obtenidos mediante una secuencia de imágenes tomadas en tiempo real con las que se determinarán la posición y velocidad de cada articulación.

1.1 Objetivo de la tesis

El objetivo principal de este trabajo de tesis es proponer, desarrollar y validar la eficiencia de un controlador difuso T-S para tareas de regulación y seguimiento del robot Mitsubishi PA10-7CE, adicionando una realimentación visual al sistema.

1.2 Estructura de la tesis

En el capítulo 2 se describe el concepto de modelo difuso T-S y las diferentes técnicas que existen para linealizar un sistema a través de este método. Acto seguido, se resume la teoría del controlador PDC así como las condiciones necesarias para determinar la estabilidad asintótica global. Se desarrolla el concepto de "sistema aumentado" mediante el análisis del modelo de un servocompensador.

En el capítulo 3 de este documento se describe con detalle el robot Mitsubishi PA10-7CE y se presentan los procedimientos para obtener tanto el modelo cinemático directo como el modelo dinámico de un robot manipulador.

En el capítulo 4 se describe el procedimiento para obtener controladores estables, ya sea de regulación o de seguimiento, a través del control difuso T-S.

En el capítulo 5 se detalla lo relacionado al campo de visión artificial, abarcando desde los orígenes de la misma hasta su aplicación en robótica. Asimismo, se explican los

diversos métodos para el procesamiento de una imagen y su obtención en tiempo real, así como también lo relacionado al papel que juega la cámara de visión en el controlador y los parámetros que deben ser tomados en cuenta para lograr una adecuada realimentación visual.

Con el afán de evaluar los procedimientos realizados, en el capítulo 6 se muestran los resultados obtenidos tanto de simulación como de tiempo real de los controladores desarrollados en este trabajo.

Capítulo 2

Modelado difuso

En este capítulo se pretende introducir el significado de "sistema difuso T-S", exponiendo a detalle una de las técnicas para la obtención de sistemas lineales a partir de un sistema no lineal, como lo es un robot de 2 gdl.

La mayoría de los sistemas no lineales pueden ser representados por modelos difusos. Existen tres tipos de modelos difusos para aproximar una función:

- 1.-Mamdani
- 2.-Takagi-Sugeno (T-S)
- 3.-Kosko

En la Figura 1 se muestra la arquitectura general del sistema difuso de tipo Mamdani, el cual consta principalmente de tres partes: la fusificación, base de conocimiento con su sistema de inferencia y la defusificación.

La función de salida para el modelo difuso Mamdani se puede determinar con el centroide del área, el mínimo de los máximos, el máximo de los máximos, etc.



Figura 1.- Sistema difuso tipo Mamdani

En este trabajo se presenta el diseño y desarrollo de controladores difusos T-S, los cuales serán aplicados tanto en simulación como en tiempo real a un robot manipulador.

2.1 Modelo difuso Takagi-Sugeno

El modelo difuso propuesto por Takagi-Sugeno [17] está descrito por reglas difusas del tipo IF-THEN, las cuales representan la relación entrada-salida de un sistema no lineal. Este modelo expresa la dinámica local de cada regla difusa mediante un subsistema lineal.

En la Figura 2 se expone una representación gráfica de la linealización de un sistema no lineal, la cual es una aproximación del mismo sobre una vecindad de un punto de trabajo. De esta manera se puede explicar el objetivo del modelado difuso T-S de un sistema, que consiste en representar completamente un sistema no lineal a través de diversos sistemas lineales.

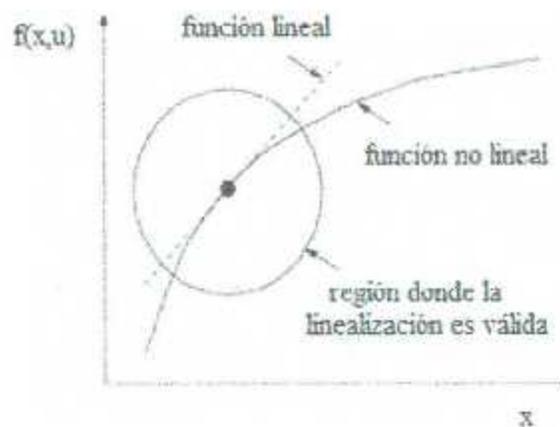


Figura 2.- Linealización de un sistema

La estructura de un modelo difuso T-S para un sistema continuo está dada por:

Regla i del modelo

SI $z_1(t)$ es M_{i1} y ... $z_p(t)$ es M_{ip}

$$\text{ENTONCES} \left\{ \begin{array}{l} \dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) \end{array} \quad i = 1, 2, \dots, r \right\}$$

(2.1)

donde:

- $z_1(t) \dots z_p(t)$: son las variables premisas
- M_{ij} : son los conjuntos difusos
- r : número de reglas del modelo
- $x(t) \in \mathbb{R}^n$: vector de estados
- $u(t) \in \mathbb{R}^m$: vector de entradas
- $y(t) \in \mathbb{R}^w$: vector de salida
- $A_i \in \mathbb{R}^{n \times n}$; $B_i \in \mathbb{R}^{n \times m}$, $C_i \in \mathbb{R}^{w \times n}$

La salida final del sistema T-S es inferida usando un fuzzificador *singleton*, una máquina de inferencia producto y un defuzificador de centro promedio, como se explica a continuación:

$$\dot{x}(t) = \frac{\sum_{i=1}^r w_i(z(t)) \{A_i x(t) + B_i u(t)\}}{\sum_{i=1}^r w_i(z(t))}$$

$$y(t) = \frac{\sum_{i=1}^r w_i(z(t)) C_i x(t)}{\sum_{i=1}^r w_i(z(t))} \quad (2.2)$$

donde:

$$z(t) = [z_1(t), z_2(t) \dots z_p(t)] \quad (2.3)$$

$$w_i(z(t)) = \prod_{j=1}^p M_{ij}(z_j(t))$$

$M_{ij}(z_j(t))$ es el grado de pertenencia de $z_j(t)$ en M_{ij} por lo que se obtiene:

$$h_i(z(t)) = \frac{w_i(z(t))}{\sum_{i=1}^r w_i(z(t))} \quad (2.4)$$

De esta manera, finalmente se tiene la siguiente ecuación con el peso normalizado para cada regla IF-THEN:

$$\begin{aligned}\dot{x}(t) &= \sum_{i=1}^r h_i(z(t)) \{A_i x(t) + B_i u(t)\} \\ y(t) &= \sum_{i=1}^r h_i(z(t)) C_i x(t).\end{aligned}\tag{2.5}$$

Existen tres enfoques para linealizar el sistema y de esta manera obtener el modelo Takagi-Sugeno:

- 1.- No linealidad por sector (Sector nonlinearity)
- 2.- Aproximaciones locales en espacios de partición difusa (Local approximation in fuzzy partition spaces)
- 3.- Series de Taylor (Taylor series)

El modelo difuso para este trabajo se desarrollará a través del enfoque de series de Taylor, en el cual se linealizará alrededor de diferentes puntos de operación basados en una señal de referencia dada.

2.1.1 Series de Taylor

Las series de Taylor surgen de una ecuación en la que se puede encontrar una solución aproximada a una función, mediante una serie de potencias enteras de polinomios.

El teorema de Taylor [14] afirma que si una función $f(x)$ definida en un intervalo de tiempo tiene derivadas de todos los órdenes, ésta se puede aproximar cerca de un punto $x = a$, como la expansión en series de Taylor alrededor de ese punto:

$$f(x) = f(a) + f'(a)(x-a) + \frac{1}{2!} f''(a)(x-a)^2 + \dots + \frac{1}{n!} f^{(n)}(a)(x-a)^n \tag{2.6}$$

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n \tag{2.7}$$

La linealización generalmente consiste en una expansión en series de Taylor de la ecuación de estado (no lineal) alrededor de un punto de operación definido naturalmente por el sistema o seleccionado arbitrariamente para satisfacer alguna necesidad de control.

2.2 Compensación paralela distribuida (*Parallel Distributed Compensation, PDC*)

La PDC ofrece un procedimiento para diseñar controladores difusos a partir del modelo difuso T-S dado. Para realizar la PDC, el sistema a controlar debe de estar representado por un modelo difuso T-S.

En el diseño de la PDC, cada regla de control es diseñada a partir de las reglas difusas correspondientes al modelo difuso T-S.

Regla i de control

$$\begin{array}{ll} \text{SI} & z_1(t) \text{ es } M_{i1} \text{ y...y } z_p(t) \text{ es } M_{ip}, \\ \text{ENTONCES} & u(t) = -K_i x(t), \quad i = 1, 2, \dots, r; \end{array} \quad (2.8)$$

donde:

r es el número de reglas

K_i es la ganancia de realimentación local

Dado lo anterior, el controlador difuso queda establecido de la siguiente forma:

$$u(t) = -\sum_{i=1}^r h_i(z(t)) K_i x(t) \quad (2.9)$$

La ley de control difuso está formada en una parte por un controlador lineal, como lo es el controlador por realimentación de estados para este caso, sin embargo, podría utilizarse cualquier otro controlador lineal para complementar el controlador difuso, como por ejemplo, controladores PID o los controladores de realimentación a la salida.

Con el control por PDC, tenemos un procedimiento simple y natural para manejar los sistemas de control no lineal. Otras técnicas para control no lineal requieren un conocimiento muy especial.

Los cálculos de las ganancias de cada uno de los controladores se hacen a través de la técnica de Desigualdades Lineales Matriciales (LMIs por sus siglas en inglés *Linear Matrix Inequalities*) para garantizar la estabilidad del sistema en lazo cerrado.

2.3 Desigualdades lineales matriciales (*Linear Matrix Inequalities, LMIs*)

Las técnicas LMIs han emergido como herramientas poderosas de diseño en áreas que van desde ingeniería de control hasta identificación de sistemas y diseño estructural.

Esta técnica matemática consiste en llevar los problemas de control a problemas de optimización convexa y resolverlos por medio de eficientes algoritmos computacionales disponibles hoy en día. Usando métodos numéricos se pueden resolver en un tiempo razonable un buen número de problemas de control automático que no se les conoce solución analítica o simplemente no la tienen.

Una de las grandes ventajas de utilizar LMIs es que una vez que se expresa un problema en términos de LMIs, éste se puede resolver mediante algoritmos de optimización, principalmente los basados en métodos de punto interior. El campo de aplicación se extiende a diversas técnicas, como control robusto, control óptimo y control difuso entre otros.

Una LMI es una restricción convexa, expresada usualmente de la forma:

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (2.10)$$

donde $x \in \mathbb{R}^m$, $F_i \in \mathbb{R}^{n \times n}$. El símbolo de la desigualdad > 0 significa que $F(x)$ es definida positiva. Por lo tanto $F(x)$ es una función afín a los elementos de x .

Por lo general en las LMIs las variables son matrices, por ejemplo, en la desigualdad de Lyapunov

$$A^T P + P A < 0 \quad (2.11)$$

donde A es una matriz de $n \times n$ y $P = P^T$ es la variable.

Esta desigualdad se puede expresar de la forma (2.10), de la siguiente manera: Sean P_1, P_2, \dots, P_m una base para las matrices simétricas de $n \times n$, entonces se toma $F_0 = 0$ y $F_i = -A^T P_i - P_i A$, por lo que se puede observar que no hay diferencia en la manera de expresar una LMI.

Los problemas siguientes son necesarios para plantear las aplicaciones de control:

2.3.1 Problema LMI

Dada una LMI $F(x) > 0$, el correspondiente problema LMI consiste en encontrar una x^* tal que $F(x^*) > 0$ o determinar que tal x^* no existe, es decir, que la LMI no es factible. Este solucionador es llamado *feasp* en MATLAB.

2.3.2 Problema de valores propios

El objetivo de este solucionador es minimizar el valor propio máximo, sujeto a una LMI, de una matriz que dependa de alguna variable, por ejemplo

$$\begin{array}{ll} \min & \lambda \\ \text{sujeto a} & \lambda - A(x) > 0 \\ & B(x) > 0 \end{array} \quad (2.12)$$

o también puede aparecer como un programa semidefinido, es decir, de la forma:

$$\begin{array}{ll} \min & c^T x \\ \text{sujeto a} & F(x) > 0 \end{array} \quad (2.13)$$

Este solucionador es llamado *mincx* en MATLAB.

2.3.3 Problema de valores propios generalizado

El problema anterior se puede extender a minimizar el máximo valor propio generalizado de un par de variables que dependen de alguna variable, sujeto a una LMI, de la forma siguiente:

$$\begin{array}{ll}
 \min & \lambda \\
 \text{sujeto a} & \lambda B(x) - A(x) > 0 \\
 & B(x) > 0 \\
 & C(x) > 0
 \end{array} \tag{2.14}$$

Este solucionador es conocido como *gevp* en MATLAB.

Los tres solucionadores *feasp*, *mincx* y *gevp* toman como entrada la representación interna de un sistema LMI y regresan un valor óptimo x^* de las variables de decisión. Los correspondientes valores de la matriz de variables X_1, \dots, X_K son derivadas de x^* con la función *dec2mat*. Estos solucionadores son implementaciones C-MEX (MATLAB Executable) del algoritmo descriptivo tiempo-polinomio de Nesterov y Nemirovski.

Para este trabajo se utilizó el solucionador *feasp* para los programas desarrollados en MATLAB, el cual resuelve el problema de factibilidad para el sistema de restricciones LMI.

Dado un problema de factibilidad LMI, encontrar x tal que $L(x) < R(x)$. El solucionador *feasp* resuelve el problema convexo auxiliar: t sujeto a $L(x) < R(x) + t * I$.

El sistema de LMIs es factible si y sólo si $t < 0$.

2.4 Diseño de un controlador PDC para regulación

El problema de diseño de un controlador PDC se basa en determinar las ganancias K_i de cada uno de los controladores lineales locales.

Teorema 1. [3] El equilibrio del sistema de control difuso continuo en lazo cerrado es globalmente asintóticamente estable si existe una matriz P_C común definida positiva tal que:

$$\begin{aligned} G_i^T P_C + P_C G_i &< 0, \\ \left(\frac{G_i + G_j}{2} \right)^T P_C + P_C \left(\frac{G_i + G_j}{2} \right) &\leq 0 \end{aligned} \quad (2.15)$$

para $i < j$ tal que $h_i \cap h_j \neq \emptyset$, donde $G_{ij} = A_i - B_i K_j$. \triangle

Las condiciones anteriores no son conjuntamente convexas en K_i y P_C , por lo que es necesario multiplicar las desigualdades por la izquierda y por la derecha por P_C^{-1} y definir una nueva variable $X = P_C^{-1}$ para que las condiciones que determinan la estabilidad del sistema se reescriban de la siguiente manera:

$$\begin{aligned} -XA_i^T - A_i X + XK_i^T B_i^T + B_i K_i X &> 0 \\ -XA_i^T - A_i X - XA_j^T - A_j X + XK_i^T B_i^T + B_i K_j X + XK_j^T B_j^T + B_j K_i X &\geq 0 \end{aligned} \quad (2.16)$$

El problema de diseño se reduce a encontrar $X > 0$ y $M_i (i = 1, \dots, r)$ que satisfagan las siguientes desigualdades:

$$\begin{aligned} -XA_i^T - A_i X + M_i^T B_i^T + B_i M_i &> 0 \\ -XA_i^T - A_i X - XA_j^T - A_j X + M_j^T B_i^T + B_i M_j + M_i^T B_j^T + B_j M_i &\geq 0 \end{aligned} \quad (2.17)$$

donde

$$M_i = K_i X$$

Se puede encontrar una matriz X definida positiva y una matriz M_i que satisfagan las LMIs o determinar que tales matrices no existen.

2.5 Diseño de un controlador PDC para seguimiento

Se propone utilizar el modelo de un servocompensador como en [2]. El servocompensador se basa en el error de seguimiento de una señal específica, el cual es definido como:

$$e(t) = y_r(t) - y(t)$$

Dado esto, el modelo del servocompensador está definido por:

$$\dot{x}_c(t) = A_c x_c(t) + B_c e(t) \quad (2.18)$$

donde

$x_c(t)$: estados del compensador

$e(t)$: errores de seguimiento en el sistema

A_c : matriz que acompaña al polinomio característico de la señal de referencia

$$A_c = \text{bloque-diag}\{A_{c1}, A_{c2}, \dots, A_{cn}\}$$

$$B_c = \text{bloque-diag}\{B_{c1}, B_{c2}, \dots, B_{cn}\}$$

$$A_c = \begin{bmatrix} 0 & & & \\ \vdots & & I_{l-1} & \\ 0 & & & \\ -\alpha_0 & -\alpha_1 & \dots & -\alpha_{l-1} \end{bmatrix} \quad B_c = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

Combinando el modelo de la planta con el modelo del servocompensador se obtiene el sistema aumentado:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}_c(t) \end{bmatrix} = \begin{bmatrix} A_p & 0 \\ -B_p C_p & A_c \end{bmatrix} \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix} + \begin{bmatrix} B_p \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ B_c \end{bmatrix} y_r(t). \quad (2.19)$$

El controlador difuso para seguimiento es diseñado via PDC en el cual cada regla de control es diseñada a partir de su correspondiente regla del modelo difuso T-S. Las reglas

del modelo difuso T-S y del controlador PDC comparten los mismos conjuntos difusos en los antecedentes de las reglas. Dado esto, todos los pesos de las reglas del controlador son tomados directamente del modelo difuso conocido, y quedan de la siguiente manera:

Regla i de control

$$\begin{array}{ll}
 \text{SI} & z_1(t) \text{ es } M_{i1} \text{ y...y } z_p(t) \text{ es } M_{ip}, \\
 \text{ENTONCES} & u(t) = -[K_i \quad K_{ci}] \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix}; \quad i=1,2,\dots,r;
 \end{array}
 \tag{2.20}$$

De esta manera, la salida final del controlador difuso estará definida como se indica a continuación:

$$\begin{aligned}
 u(t) &= - \frac{\sum_{i=1}^r w_i(z(t)) [K_i \quad K_{ci}] \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix}}{\sum_{i=1}^r w_i(z(t))} \\
 &= - \sum_{i=1}^r h_i(z(t)) [K_i \quad K_{ci}] \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix}
 \end{aligned}
 \tag{2.21}$$

donde:

$$h_i = \frac{w_i(z(t))}{\sum_{i=1}^r w_i(z(t))}$$

Capítulo 3

Mitsubishi PA10-7CE

Por lo general, un robot es un dispositivo mecánico articulado capaz de imitar ciertas funciones humanas como la manipulación de objetos o el movimiento con el fin de sustituir al humano en la realización de ciertas tareas [1].

Como ya se sabe, existen diferentes tipos de robots que se emplean para algún trabajo específico y se dividen en dos grandes ramas: los manipuladores (de cadena abierta y cerrada) y los móviles.

Un robot manipulador lleva este nombre porque es capaz de manipular objetos y/o herramientas. Mecánicamente está formado por una serie de elementos, llamados eslabones, unidos mediante articulaciones que permiten un movimiento relativo entre los eslabones consecutivos. La constitución física de la mayor parte de los robots industriales guarda cierta similitud con la anatomía del brazo humano, es por esto que en ocasiones se les refiere como brazos manipuladores [1]. Para este trabajo de tesis se utilizará el robot manipulador Mitsubishi PA10-7CE, el cual está ubicado en el Laboratorio de Mecatrónica y Control del Instituto Tecnológico de la Laguna.

El robot Mitsubishi PA10-7CE es un robot manipulador redundante de arquitectura abierta, que se ha convertido en objeto de estudio de múltiples trabajos de investigación. El sistema de control del PA10 incluye una tarjeta de control de movimiento que se puede configurar para llevar a cabo las tareas de control más comunes. Se dice que el sistema PA10 es abierto ya que está separado por niveles (ver Figura 3), a los cuales se puede acceder conociendo la forma en que se comunican entre ellos. [13]

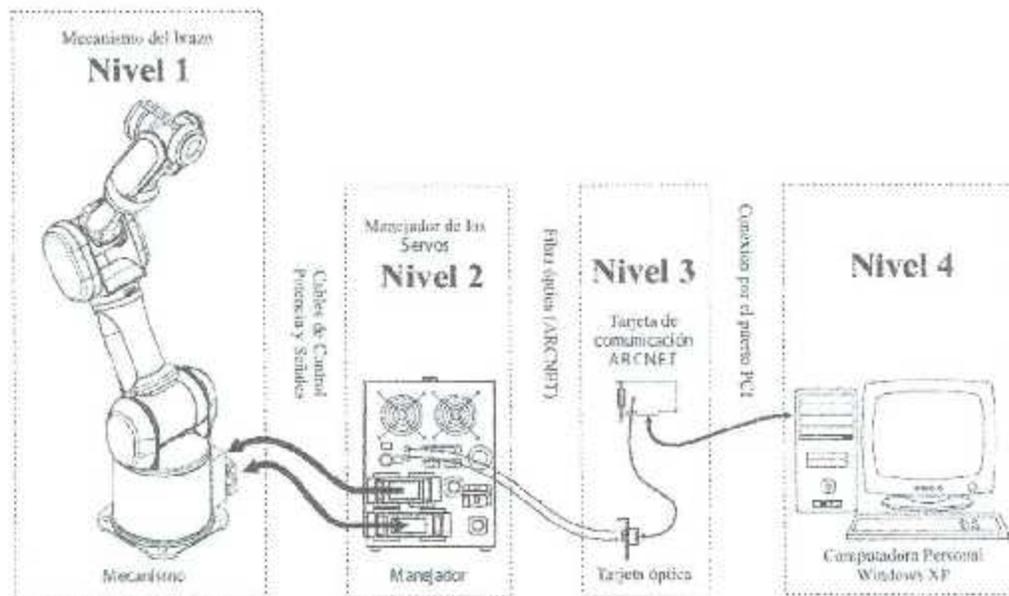


Figura 3.-Niveles funcionales actuales del Mitsubishi PA10-7CE

3.1 Componentes del robot Mitsubishi PA10-7CE

A continuación se presentan los componentes más relevantes de cada uno de los niveles mencionados del robot PA10-7CE.

3.1.1 Nivel 1: Mecanismo

Es el nivel más bajo del robot y abarca toda la estructura mecánica. Los servomotores del PA10 son trifásicos del tipo CD sin escobillas (brushless DC) y están acoplados a los eslabones por medio de transmisiones armónicas (harmonic drives).

Brazo

Es la parte mecánica del sistema, y está compuesto de siete articulaciones conectadas a través de eslabones, como se muestra en la Figura 4. Cada una de las 7 articulaciones cuenta con frenos electromagnéticos los cuales tienen la tarea de sostener el robot.

El dígito del sufijo encontrado en el nombre del robot se refiere a los grados de libertad con los que cuenta el robot. Como se puede observar, y como ya se mencionó, el Mitsubishi PA10-7CE cuenta con 7 grados de libertad, pero para este trabajo de tesis se limitará su funcionamiento a 2 grados de libertad (ejes 2 y 4), dejando los demás ejes fijos.

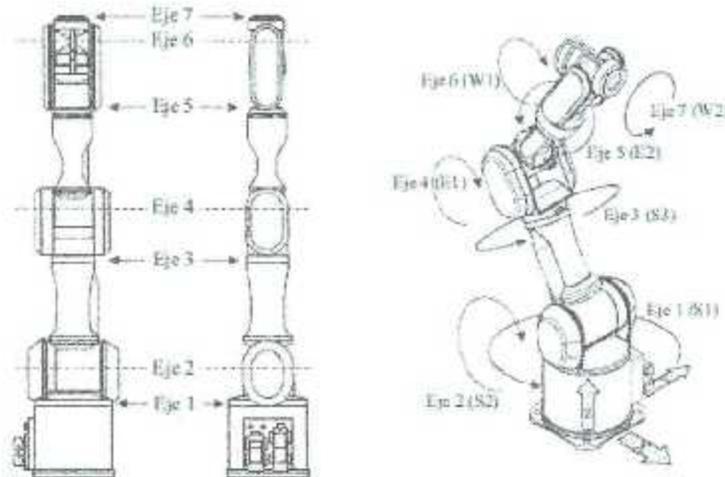


Figura 4.- Mecánica del Mitsubishi PA10-7CE

3.1.2 Nivel 2: Controlador de servos

El manejador de los servos se puede configurar en *modo par* o *modo velocidad* dependiendo de la consigna de control de movimiento para los servomotores, a través del protocolo "ARCNET", el cual, tiene la capacidad de hacer que el sistema trabaje en tiempo real. Estos manejadores, al igual que la fuente de potencia, estarán contenidos en una caja de control (ver Figura 5). De la misma manera, en la caja de control se encuentra el botón de paro de emergencia, el cual activa de forma inmediata los frenos para cada articulación, y apaga los servomotores del robot. A través de cables se puede acceder desde el nivel 2 hacia el nivel 1, ya que uno se encarga de proveer la energía a los servomotores, y otro de transmitir la información de los sensores y otros datos.

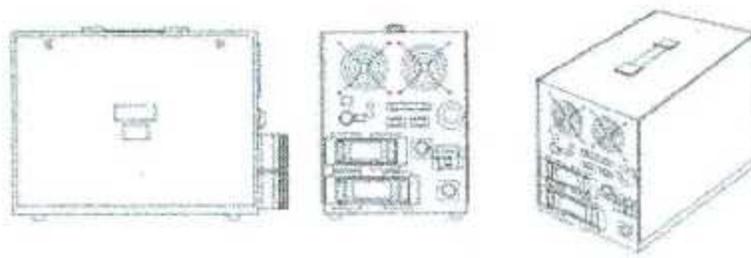


Figura 5. Caja de control de servomotores del PA10-7

3.1.3 Nivel 3: Control de movimiento

Este nivel cuenta con una tarjeta de control de movimiento “PCI20U-4000” de la marca Contemporary Controls (ver Figura 6), la cual se utiliza para controlar el robot en modo par o en modo velocidad, ya que tiene la labor de calcular los valores que se enviarán para el control, basándose en la información que provee la computadora, todo a través del protocolo de comunicación ARCNET. Esta tarjeta para el control de movimiento está conectada a un transceptor óptico, que se encarga de convertir señales eléctricas en ópticas, para enviar los datos entre el controlador y la tarjeta de control de movimiento a través de fibra óptica [13].

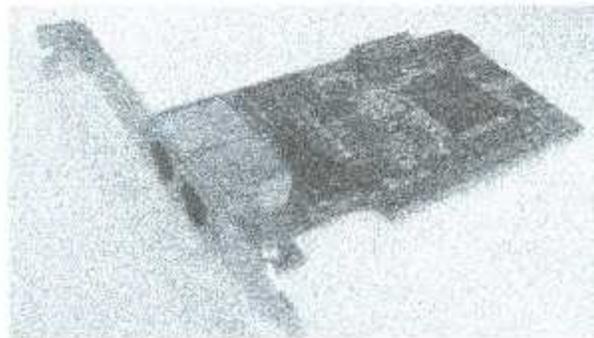


Figura 6.- Tarjeta ARCNET PCI20U-4000

3.1.4 Nivel 4: Control de operación

Este nivel está formado por una PC de control que tiene insertada la tarjeta de comunicación ARCNET (a través de un puerto o bus PCI) y la tarjeta de fibra óptica descrita anteriormente. Esta computadora de control cuenta con los programas necesarios [12] para manejar el robot a través de una interfaz gráfica, desarrollada en [7]

Interfaz de control

Para el funcionamiento en MATLAB/Simulink se utilizará la interfaz gráfica mencionada anteriormente (Figura 7), la cual fue desarrollada exclusivamente para el robot Mitsubishi PA10. Dicha interfaz se basa en el uso del siguiente software:

- Quarc versión 2.5 “Quanser Real-Time Rapid Control Prototyping Software” . Es un software desarrollado por la compañía Quanser, que se integra perfectamente con MatLab/Simulink, de manera que acelera el diseño y la implementación de controladores, debido a que no es necesaria la codificación manual. Este programa ofrece también un conjunto de bloques para dispositivos externos, como el bloque PA10 Control de Mitsubishi, los cuales ayudan a interactuar con los sistemas reales en tiempo real sin ningún problema.
- Microsoft Visual Studio Professional Edition 2010 (versión 10.0). Es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes de programación, como C y C++. Una de sus características principales es la de fungir como compilador (i.e. en MATLAB), haciendo necesaria su instalación para un correcto desempeño de software externo, como lo es Quarc
- MATLAB R2014a con: Simulink, Simulink Coder, MATLAB Coder y Control System Toolbox.

Estos programas fueron instalados en una computadora del Laboratorio de Mecatrónica y Control del Instituto Tecnológico de la Laguna, que cuenta con el sistema operativo Windows 7 de 64-bits, procesador Intel Core i7 y una memoria RAM de 4 GB.

A continuación se describen los controles y demás componentes que se encuentran en la ventana de la interfaz gráfica de control para el PA10 en Simulink (Figura 7):

- Nombre del archivo: ingresar el nombre del archivo de Simulink que contiene el algoritmo de control.
- Cargar: al presionar este botón se abre el archivo de Simulink que se ha indicado. En el caso de que no exista un archivo con el nombre propuesto, se despliega una ventana de error.

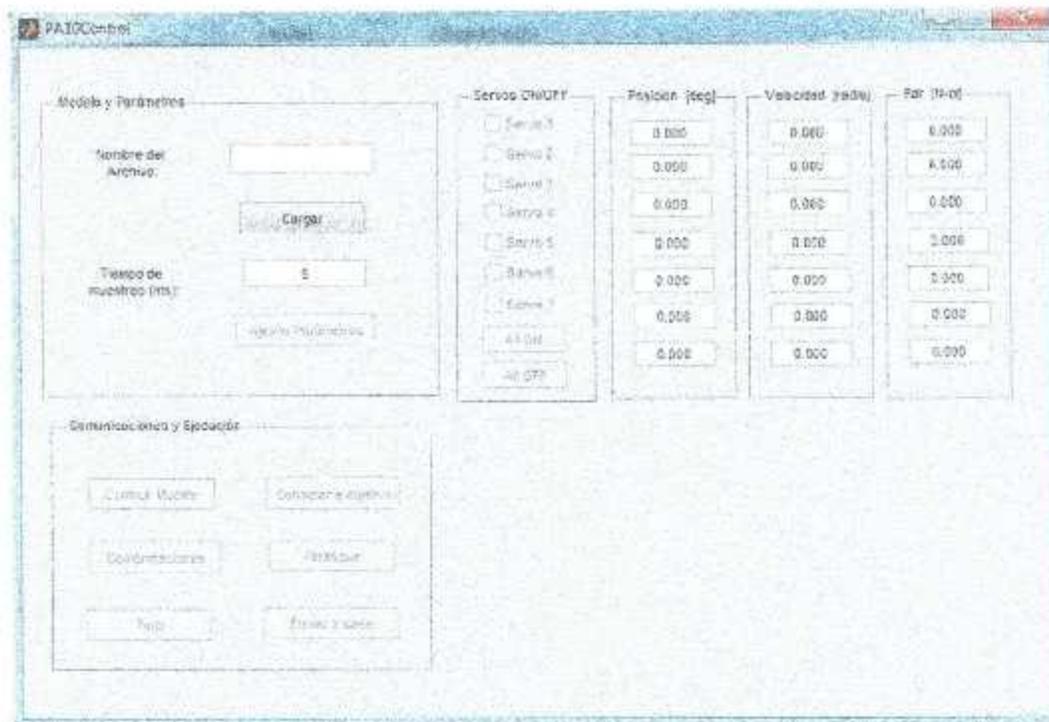


Figura 7.- Interfaz gráfica de control para el PA10 en Simulink

- **Tiempo de muestreo:** ingresar el valor del tiempo de muestreo a utilizar en milisegundos, el cuál debe estar entre los 5 y 100 ms. Por defecto este cuadro de texto tiene un valor asignado de 5 ms.
- **Ajustar parámetros:** establece por defecto los siguientes parámetros en la ventana de configuración de parámetros del archivo de Simulink (Simulation/Configuration Parameters): Integrador odel (Euler), paso fijo, tiempo final de ejecución infinito, modo de ejecución externo. Si el tiempo de muestreo se encuentra fuera de los límites, se despliega una ventana de error.
- **Construir modelo:** se crearán los archivos necesarios para la ejecución del modelo en tiempo real. Esta instrucción es equivalente al comando Tools/CodeGeneration/BuildModel de Simulink.
- **Conectar a objetivo:** esta instrucción es equivalente al comando Simulation/Connect to Target de Simulink para modo externo.
- **Comunicaciones:** establece la comunicación con los manejadores de los servos. El tiempo de simulación de Simulink comienza a correr; en el caso de que la comunicación no sea llevada a cabo con éxito, una ventana de error de Simulink es desplegada. Esta instrucción es equivalente al comando Simulation/Start Real-Time code de Simulink para modo externo.

- Servos ON/OFF: al marcar o desmarcar alguno de los check box contenidos en este panel, se enciende o apaga respectivamente el servomotor correspondiente. El botón all ON enciende todos los servomotores del PA10, mientras que el botón all OFF apaga todos los servomotores.
- Arranque: iniciada la comunicación con el robot y activados los servos, es posible iniciar el controlador del PA10 al presionar este botón, en este momento el tiempo de inicio de control empieza a correr.
- Enviar a casa: inicia el algoritmo para llevar al robot a su posición de casa sin importar lo tarea que esté realizando el robot.
- Paro: detiene completamente el robot, los servos son apagados y los frenos magnéticos que detienen al PA10 se activan; esta instrucción es equivalente al comando Simulation/Stop Real-Time Code para modo externo de Simulink.
- Paneles adicionales: se ubican en la parte superior derecha de la interfaz (ver Figura 11) e indican la posición, velocidad y par actuales de cada una de las articulaciones.

3.2 Modelado del Mitsubishi PA10-7CE

Como se mencionó anteriormente, se limitará el funcionamiento del robot Mitsubishi PA10-7CE de 7 grados de libertad (gdl), para trabajar solamente con dos grados de libertad (articulación 2 y 4), los cuales se moverán en el plano horizontal, como se representa en la Figura 8. Se definirá un sistema de coordenadas cartesianas en donde el origen será el punto de intersección del eje 1 con el plano horizontal x, y .

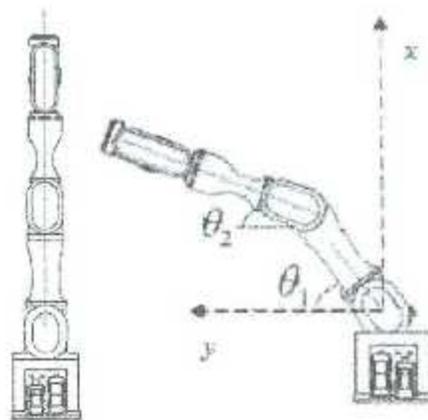


Figura 8.- Articulaciones funcionales PA10

3.2.1 Modelo cinemático

La cinemática de robots establece las relaciones entre las diferentes variables que intervienen en el movimiento del robot, visto desde un punto meramente geométrico. Algunos de los primeros aspectos que se deben de considerar, al estudiar un mecanismo, son la estructura cinemática y el número de grados de libertad (gdl) que posee, ya que en base al modelo cinemático directo de posición de dicho mecanismo, se puede determinar la posición del extremo operativo en un plano.

Se le llama cadena cinemática a un ensamble de eslabones conectados por articulaciones. Un manipulador serial está constituido por una cadena cinemática abierta, ya que solo existe una ruta desde uno de los extremos del robot (la base) hasta el otro (el órgano terminal). Por otra parte, en un robot manipulador paralelo existen varios caminos para llegar desde la base hasta el órgano terminal, por lo que se considera que un robot paralelo tiene, por lo menos, una cadena cinemática cerrada [8]. Para este trabajo de tesis se presentan algoritmos exclusivos para robots seriales.

El método Denavith-Hartenberg (D-H) fue propuesto en 1955 y se considera el método más usado para obtener el modelo cinemático directo de robots manipuladores. Para asignar los marcos coordenados asociados a cada eslabón se deberán seguir los siguientes pasos:

- 1.- Identificar la base fija del robot con el marco 0 .
- 2.- Numerar los marcos del 1 al n iniciando con el primer eslabón móvil y terminando con el último eslabón móvil.
- 3.- Determinar si la articulación es rotacional o prismática. Si es rotacional, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.
- 4.- Para i de 0 a $n-1$ situar el eje z_i sobre el eje de la articulación $i+1$.
- 5.- Situar el origen del sistema de la base \sum_0 en cualquier punto del eje z_0 . Los ejes x_0 o y_0 se situarán de modo que formen un sistema dextrógiro (que cumpla con la regla de la mano derecha) con z_0 .
- 6.- Para i de 1 a $n-1$, situar el sistema \sum_i (solidario al eslabón i) en la intersección del eje z_i con la línea normal común a z_{i-1} y z_i . Si ambos ejes se cortan entre sí, se situará \sum_i en el punto de corte. Si son paralelos \sum_i se situará en la articulación $i+1$.

7.- Situar x_j en la línea normal común a z_{j-1} y z_j . Luego situar y_j de modo que forme un sistema dextrógiro con x_j y z_j .

8.- Situar el sistema \sum_j en el extremo del robot de modo que z_n coincida con la dirección de z_{n-1} y x_n sea normal a z_{n-1} y z_n .

Una vez establecidos los marcos coordenados asociados a cada eslabón, se debe de establecer la postura relativa entre cada par de marcos consecutivos, utilizando para esto, un control de 4 parámetros por articulación. Los 4 parámetros D-H para la articulación i se definen a continuación:

- d_i : distancia entre los ejes x_{i-1} y x_i , medida a lo largo del eje z_{i-1}
- θ_i : ángulo entre los ejes x_{i-1} y x_i , medido alrededor del eje z_{i-1}
- a_i : distancia entre los ejes z_{i-1} y z_i , medida a lo largo del eje x_i
- α_i : ángulo entre los ejes z_{i-1} y z_i , medido a lo largo del eje x_i

Los parámetros a_i y α_i son constantes ya que sólo dependerán de la geometría de las conexiones entre los eslabones consecutivos. El valor de los parámetros θ_i y d_i dependerá del tipo de articulación que conecta los eslabones, ya que:

- si la articulación es de revolución, la variable es θ_i y d_i será constante.
- si la articulación es prismática, la variable es d_i y θ_i será constante.

Una vez obtenidos los parámetros D-H, es necesario formar las n matrices de transformación homogénea T_i^{i-1} como se menciona en [13]:

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

A continuación se muestra la obtención del modelo cinemático directo de posición para el robot Mitsubishi PA10 para 2 gdl. Para simplificar la notación en las ecuaciones que contienen funciones trigonométricas se hace la siguiente sustitución:

$$c_i = \cos(q_i)$$

$$s_i = \text{sen}(q_i)$$

Los marcos fueron seleccionados de la siguiente manera:

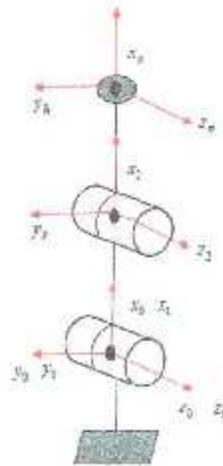


Figura 9.- Selección de marcos para el PA10

Los parámetros D-H obtenidos, están representados en la Tabla 1:

PARÁMETROS DENAVIT-HARTENBERG					
ESLABÓN	σ_i	α_i	d_i	θ_i	r_i
1	0	0	0	θ_1	0
2	0	0	L_2	θ_2	0

Tabla 1.- Parámetros Denavith-Hartenberg (D-H) Mitsubishi PA10 limitado a 2 gdl.

Una vez que se obtienen los parámetros D-H, es necesario formar las n matrices de transformación homogénea T_i^{i-1} , las cuales se muestran a continuación:

$${}^0_1T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1_2T = \begin{bmatrix} c_2 & -s_2 & 0 & L_2 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^2_hT = \begin{bmatrix} 1 & 0 & 0 & L_h \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

El modelo cinemático directo de posición se obtiene como:

$${}^0_hT = {}^0_1T \times {}^1_2T \times {}^2_hT$$

$${}^0_hT = \begin{bmatrix} c_1c_2 - s_1s_2 & -c_1s_2 - s_1c_2 & 0 & L_2^*c_1 + L_h^*c_1c_2 - L_h^*s_1s_2 \\ s_1c_2 + c_1s_2 & -s_1s_2 + c_1c_2 & 0 & L_2^*s_1 + L_h^*c_1s_2 + L_h^*s_1c_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

El modelo cinemático directo de posición de este manipulador que relaciona la posición del extremo operativo en el plano x, y con las variables articulares del robot q_1 y q_2 , viene dado por:

$$\begin{aligned} x &= L_2^*c_1 + L_h^*c_1c_2 - L_h^*s_1s_2 \\ y &= L_2^*s_1 + L_h^*c_1s_2 + L_h^*s_1c_2 \end{aligned} \quad (3.3)$$

3.2.2 Modelo dinámico

El cálculo del modelo dinámico de un robot manipulador juega un papel importante en la simulación de movimiento y el diseño de algoritmos de control

La ecuación que describe la dinámica de un robot manipulador rígido con articulaciones de revolución y de n grados de libertad, puede ser expresada en forma compacta como:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (3.4)$$

Nótese que la ecuación anterior es una ecuación diferencial vectorial no lineal en el estado $[q \ \dot{q}]^T$, $C(q, \dot{q})\dot{q}$ es un vector de $n \times 1$ llamado vector de fuerzas centrífugas y de Coriolis, $g(q)$ es un vector de $n \times 1$ de fuerzas o pares gravitacionales y τ es un vector de $n \times 1$ llamado el vector de fuerzas externas, siendo generalmente los pares y fuerzas aplicadas por los accionadores en las articulaciones.

Donde $q \in \mathbb{R}^n$ es el vector de coordenadas generalizadas, $M(q) \in \mathbb{R}^{n \times n}$ es una matriz simétrica definida positiva, denominada Matriz de Inercia, y que se asocia a la energía cinética del manipulador, $C(q, \dot{q})\dot{q} \in \mathbb{R}^n$ es el vector de fuerzas centrífugas y de Coriolis, $g(q) \in \mathbb{R}^n$ es un vector de fuerzas o pares gravitacionales y $\tau \in \mathbb{R}^n$ es un vector de fuerzas externas, siendo generalmente los pares y fuerzas aplicadas por los accionadores en las articulaciones.

Para que el modelo dinámico de un robot sea más aproximado a la realidad, es necesario incluir los efectos de la fricción en cada una de las articulaciones, por lo que el modelo estaría dado por:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f(\dot{q}) = \tau \quad (3.5)$$

donde $f(\dot{q})$ denota el vector de fuerzas de fricción.

En esta sección se presentará el modelo dinámico del robot manipulador Mitsubishi PA10-7 reducido a 2 grados de libertad. Su obtención queda fuera del objetivo de este proyecto de tesis, sin embargo puede ser consultada en textos especializados como [9], [13].

Como se ha mencionado anteriormente, se trabajará solamente con 2 articulaciones del robot, debido al gran costo computacional que se requiere para trabajar con 7 grados a través de un controlador difuso T-S en tiempo real. Para la obtención del modelo dinámico, se procedió a utilizar el modelo completo para el Mitsubishi PA10-7, con la diferencia de eliminar (colocar en cero) los renglones y columnas de las articulaciones que no se utilizan, o dicho de otra forma, que se encuentran bloqueadas, dando como resultado el sistema a continuación:

$$M = \begin{bmatrix} 2.5217 * \cos(q_4) - .0044 * \cos^2(q_4) + 3.7537 & 1.2707 * \cos(q_4) + 1.0920 \\ 1.2707 * \cos(q_4) + 1.0920 & 1.1018 \end{bmatrix} \quad (3.6)$$

$$C = \begin{bmatrix} C_{(1,1)} & C_{(1,2)} \\ C_{(2,1)} & C_{(2,2)} \end{bmatrix}$$

$$\begin{aligned} C_{(1,1)} &= 0.4388 * \dot{q}_4 * \cos(q_4) * \text{sen}(q_4) - 1.2608 * \dot{q}_4 * \text{sen}(q_4) \\ C_{(1,2)} &= 0.4388 * \dot{q}_2 * \cos(q_4) * \text{sen}(q_4) - 1.2706 * \dot{q}_4 * \text{sen}(q_4) - 1.2608 * \dot{q}_2 * \text{sen}(q_4) \\ C_{(2,1)} &= 1.2608 * \dot{q}_2 * \text{sen}(q_4) - 0.4388 * \dot{q}_2 * \cos(q_4) * \text{sen}(q_4) \\ C_{(2,2)} &= 0 \end{aligned} \quad (3.7)$$

$$g = \begin{bmatrix} [-68.15 * \text{sen}(q_2) - 30.75 * \cos(q_2) * \text{sen}(q_4) - 30.75 * \cos(q_4) * \text{sen}(q_2)] \\ [-30.75 * \cos(q_2) * \text{sen}(q_4) - 30.75 * \cos(q_4) * \text{sen}(q_2)] \end{bmatrix} \quad (3.8)$$

donde:

$$M \in \mathbb{R}^{2 \times 2}, C \in \mathbb{R}^{2 \times 2}, g \in \mathbb{R}^{2 \times 1}$$

Capítulo 4

Control difuso Takagi-Sugeno (T-S)

4.1 Linealización del modelo dinámico del Mitsubishi PA10

Una vez que se obtiene el modelo dinámico del robot se procede a obtener el modelo no lineal despejando las derivadas de mayor orden, como se muestra a continuación:

$$\ddot{q} = M(q)^{-1} * [\tau - C(q, \dot{q})\dot{q} - g(q)]$$

Ahora se utilizará la nomenclatura de variables de estado; considérense los estados del robot Mitsubishi PA10 como:

$$x_1 = q_2$$

$$x_2 = q_4$$

$$x_3 = \dot{q}_2$$

$$x_4 = \dot{q}_4$$

Las entradas de control se definen como τ_2 y τ_4 . De esta manera, la ecuación de estados del sistema está dada por:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \dot{q}_2 \\ \dot{q}_4 \\ \ddot{q}_2 \\ \ddot{q}_4 \end{bmatrix}$$

Las variables premisas serán la posición de la articulación 2 y la posición de la articulación 4. La velocidad, tanto de la articulación 2 (\dot{q}_2), como de la articulación 4 (\dot{q}_4), así como las condiciones iniciales del sistema, son considerados como cero, i.e. $x_3 = 0$, $x_4 = 0$, $\tau_2 = 0$, $\tau_4 = 0$.

Se diseña un modelo difuso de nueve reglas debido a la combinación de los tres puntos de operación propuestos para cada articulación sobre los cuales el sistema será aproximado.

De esta forma, la dinámica no lineal del robot estará representada por el modelo difuso T-S de la siguiente manera:

$$\begin{array}{ll} \text{Regla 1:} & \text{SI } x_1 \text{ es } \mu_1 \text{ y } x_2 \text{ es } \mu_1 \\ & \text{ENTONCES } \dot{x}(t) = A_1x(t) + B_1u(t) \end{array}$$

$$\begin{array}{ll} \text{Regla 2:} & \text{SI } x_1 \text{ es } \mu_1 \text{ y } x_2 \text{ es } \mu_2 \\ & \text{ENTONCES } \dot{x}(t) = A_2x(t) + B_2u(t) \end{array}$$

$$\begin{array}{ll} \text{Regla 3:} & \text{SI } x_1 \text{ es } \mu_2 \text{ y } x_2 \text{ es } \mu_1 \\ & \text{ENTONCES } \dot{x}(t) = A_3x(t) + B_3u(t) \end{array}$$

$$\begin{array}{ll} \text{Regla 4:} & \text{SI } x_1 \text{ es } \mu_2 \text{ y } x_2 \text{ es } \mu_2 \\ & \text{ENTONCES } \dot{x}(t) = A_4x(t) + B_4u(t) \end{array}$$

$$\begin{array}{ll} \text{Regla 5:} & \text{SI } x_1 \text{ es } \mu_3 \text{ y } x_2 \text{ es } \mu_1 \\ & \text{ENTONCES } \dot{x}(t) = A_5x(t) + B_5u(t) \end{array}$$

$$\begin{array}{ll} \text{Regla 6:} & \text{SI } x_1 \text{ es } \mu_3 \text{ y } x_2 \text{ es } \mu_2 \\ & \text{ENTONCES } \dot{x}(t) = A_6x(t) + B_6u(t) \end{array}$$

$$\begin{array}{ll} \text{Regla 7:} & \text{SI } x_1 \text{ es } \mu_4 \text{ y } x_2 \text{ es } \mu_1 \\ & \text{ENTONCES } \dot{x}(t) = A_7x(t) + B_7u(t) \end{array}$$

$$\begin{array}{ll} \text{Regla 8:} & \text{SI } x_1 \text{ es } \mu_4 \text{ y } x_2 \text{ es } \mu_2 \\ & \text{ENTONCES } \dot{x}(t) = A_8x(t) + B_8u(t) \end{array}$$

$$\begin{array}{ll} \text{Regla 9:} & \text{SI } x_1 \text{ es } \mu_5 \text{ y } x_2 \text{ es } \mu_1 \\ & \text{ENTONCES } \dot{x}(t) = A_9x(t) + B_9u(t) \end{array}$$

En la Figura 10 se presentan las funciones de membresía descritas para el sistema.

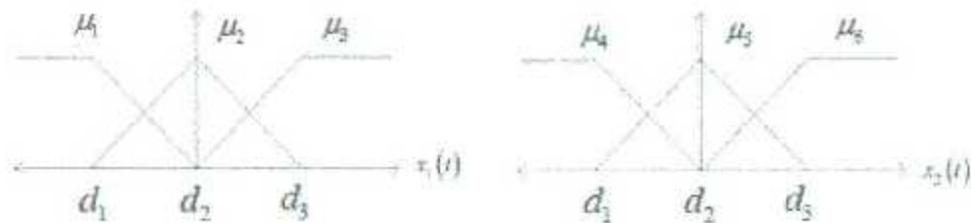


Figura 10.- Funciones de membresía del sistema

donde: $d_1 = -1$ [radianes]; $d_2 = 0$ [radianes]; $d_3 = 1$ [radianes]. El número de reglas ha sido seleccionado después de un proceso de simulación y descartando algunos casos que no se presentarían debido a los alcances de la señal de referencia dada.

La linealización del modelo para ciertos puntos de equilibrio específicos es obtenida utilizando las series de Taylor; por lo tanto, la respectiva linealización es dada como $\dot{x} = Ax(t) + Bu(t)$, donde:

$$A = \left. \frac{\partial F(x(t), u(t), t)}{\partial x} \right|_{x_0, u_0} \quad (4.1)$$

$$B = \left. \frac{\partial F(x(t), u(t), t)}{\partial u} \right|_{x_0, u_0}$$

y $x_0 = [x_1 \ x_2 \ x_3 \ x_4]^T$, $u_0 = [\tau_2 \ \tau_4]$ son los valores de x y u para cada punto de equilibrio.

Para cada una de las reglas dadas, se obtuvo un sistema lineal (A_i, B_i) donde $i = 1, \dots, 9$, los cuales se muestran a continuación:

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 27.35 & -29.21 & 0 & 0 \\ -30.75 & 90.57 & 0 & 0 \end{bmatrix} \quad B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.8302 & -1.78 \\ -1.78 & 4.72 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 14.78 & -15.78 & 0 & 0 \\ -16.61 & 48.93 & 0 & 0 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.8302 & -1.78 \\ -1.78 & 4.72 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 14.78 & -15.78 & 0 & 0 \\ -16.61 & 48.93 & 0 & 0 \end{bmatrix} \quad B_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.8302 & -1.78 \\ -1.78 & 4.72 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 25.82 & 10.85 & 0 & 0 \\ -26.60 & -9.31 & 0 & 0 \end{bmatrix} \quad B_4 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.44 & -0.71 \\ -0.71 & 2.06 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 19.91 & 4.95 & 0 & 0 \\ -43.76 & -2.24 & 0 & 0 \end{bmatrix} \quad B_5 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.44 & -0.71 \\ -0.71 & 2.06 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 7.99 & 6.77 & 0 & 0 \\ 15.01 & -7.82 & 0 & 0 \end{bmatrix} \quad B_6 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.44 & -0.71 \\ -0.71 & 2.06 \end{bmatrix}$$

$$A_7 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 25.82 & 10.85 & 0 & 0 \\ -26.60 & -9.31 & 0 & 0 \end{bmatrix} \quad B_7 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.44 & -0.71 \\ -0.71 & 2.06 \end{bmatrix}$$

$$A_8 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 7.99 & 6.77 & 0 & 0 \\ 15.01 & -7.82 & 0 & 0 \end{bmatrix} \quad B_8 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.44 & -0.71 \\ -0.71 & 2.06 \end{bmatrix}$$

$$A_9 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 19.91 & 4.95 & 0 & 0 \\ -43.76 & -2.24 & 0 & 0 \end{bmatrix} \quad B_9 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.44 & -0.71 \\ -0.71 & 2.06 \end{bmatrix}$$

Como ya es sabido, el modelo obtenido solamente es una aproximación del sistema original, sin embargo, quedará comprobado que el controlador difuso diseñado funciona correctamente cuando se aplica al sistema real.

4.2 Modelo difuso Takagi-Sugeno de la Planta

Una vez que se obtienen las nueve matrices A y B establecidas por el número de reglas propuesto, se procede a la obtención del modelo Takagi-Sugeno de la planta (robot Mitsubishi PA10-7CE). Este modelo es de suma importancia para la determinación del controlador, ya que como se mencionó en el capítulo 2, el controlador difuso T-S se basa en las mismas reglas utilizadas para la obtención del modelo difuso T-S del sistema. La salida final de dicho sistema es inferida usando un fuzzificador *singleton*, una máquina de inferencia producto y un defuzzificador de centro promedio, como se mostró en la ecuación (2.2).

El modelo difuso T-S de la planta queda establecido de la siguiente manera:

$$a_1 = \frac{z_1(A_1x + B_1u) + z_2(A_2x + B_2u) + z_3(A_3x + B_3u) + z_4(A_4x + B_4u)}{z_1 + z_2 + z_3 + z_4 + z_5 + z_6 + z_7 + z_8 + z_9}$$

$$a_2 = \frac{z_5(A_5x + B_5u) + z_6(A_6x + B_6u) + z_7(A_7x + B_7u) + z_8(A_8x + B_8u) + z_9(A_9x + B_9u)}{z_1 + z_2 + z_3 + z_4 + z_5 + z_6 + z_7 + z_8 + z_9}$$

$$\dot{x}(t) = a_1 + a_2 \quad (4.2)$$

Para corroborar el resultado obtenido de la linealización difusa T-S de la planta, se aplicó un controlador PID (probado con anterioridad en el modelo no lineal, donde $K_{p1} = 350$; $K_{i1} = 55$; $K_{d1} = 400$ y $K_{p2} = 200$; $K_{i2} = 30$; $K_{d2} = 180$) al modelo lineal obtenido por el método de T-S (ver Figura 11). Dicho experimento arrojó los siguientes resultados:

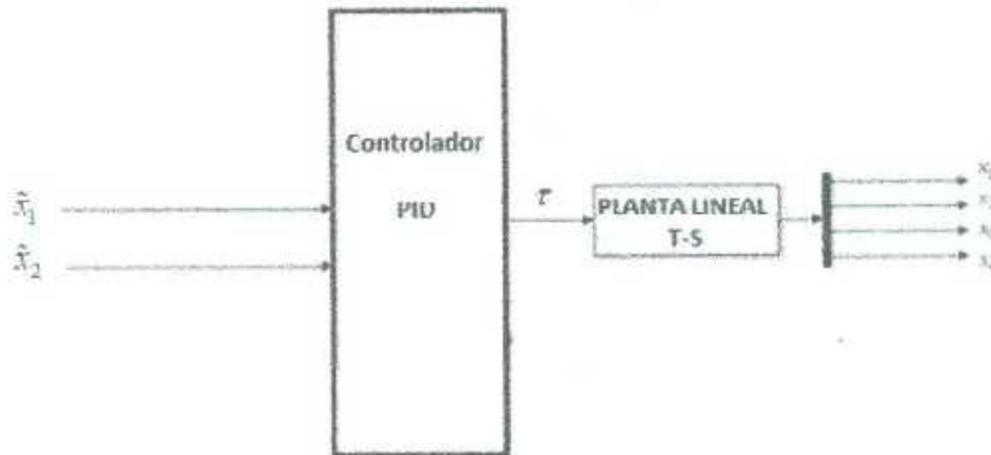


Figura 11.- Diagrama a bloques del controlador PID con una planta lineal Takagi-Sugeno

Para la posición de la articulación 2 (ver Figura 12), la señal de color azul muestra la referencia a seguir, mientras que la línea de color negro muestra la posición obtenida. Dado este resultado se puede observar que el error está acotado, por lo que la linealización del sistema para la articulación 2 es correcta.

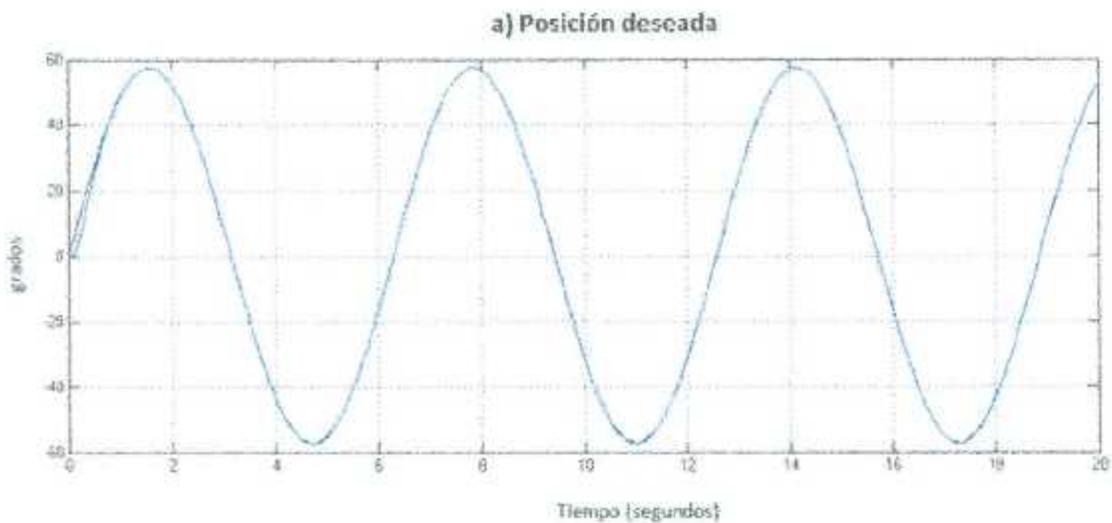


Figura 12.- Posición de la articulación 2 con el modelo difuso de la planta

Para la posición de la articulación 4 (ver Figura 13), la señal de color azul muestra la referencia a seguir mientras que la línea de color negro muestra la posición obtenida. Dado este resultado se puede observar que al igual que en la articulación 2 el error está

acotado, por lo que la linealización del sistema para la articulación 4 también es correcta. De esta manera, se determina que la linealización difusa T-S del sistema es aceptable y se procede a calcular las ganancias del controlador en base a las reglas propuestas para el sistema.



Figura 13.- Posición de la articulación 4 con el modelo difuso de la planta

4.3 Método de regulación

4.3.1 Controlabilidad del sistema

Para poder realizar los cálculos de las matrices de ganancia es necesario realizar la prueba de controlabilidad. A continuación se presenta el requerimiento para la controlabilidad:

Una planta de n -ésimo orden, cuya ecuación de estado es:

$$\dot{x} = Ax + Bu$$

es controlable si la matriz:

$$C_M = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B] \quad (4.3)$$

es de rango n , donde la matriz C_M es la matriz de controlabilidad.

Evaluando los pares de cada subsistema local (A_i, B_i) desde $i=1, \dots, 9$, se define cada matriz de controlabilidad de la siguiente manera:

$$\begin{aligned} C_{M1} &= (A_1, B_1); & C_{M4} &= (A_4, B_4); & C_{M7} &= (A_7, B_7); \\ C_{M2} &= (A_2, B_2); & C_{M5} &= (A_5, B_5); & C_{M8} &= (A_8, B_8); \\ C_{M3} &= (A_3, B_3); & C_{M6} &= (A_6, B_6); & C_{M9} &= (A_9, B_9); \end{aligned}$$

las cuales están conformadas por:

$$\begin{aligned} C_{M1} &= \begin{bmatrix} 0 & 0 & 0.83 & -1.78 & 0 & 0 & 74.72 & -186.76 \\ 0 & 0 & -1.78 & 4.72 & 0 & 0 & -186.76 & 482.68 \\ 0.8302 & -1.78 & 0 & 0 & 74.72 & -186.76 & 0 & 0 \\ -1.7802 & 4.72 & 0 & 0 & -186.76 & 482.68 & 0 & 0 \end{bmatrix} \\ C_{M2} &= \begin{bmatrix} 0 & 0 & 0.83 & -1.78 & 0 & 0 & 40.37 & -100.90 \\ 0 & 0 & -1.78 & 4.72 & 0 & 0 & -100.90 & 260.79 \\ 0.83 & -1.78 & 0 & 0 & 40.37 & -100.90 & 0 & 0 \\ -1.78 & 4.72 & 0 & 0 & -100.90 & 260.79 & 0 & 0 \end{bmatrix} \\ C_{M3} &= \begin{bmatrix} 0 & 0 & 0.83 & -1.78 & 0 & 0 & 40.37 & -100.90 \\ 0 & 0 & -1.78 & 4.72 & 0 & 0 & -100.90 & 260.79 \\ 0.83 & -1.78 & 0 & 0 & 40.37 & -100.90 & 0 & 0 \\ -1.78 & 4.72 & 0 & 0 & -100.90 & 260.79 & 0 & 0 \end{bmatrix} \\ C_{M4} &= \begin{bmatrix} 0 & 0 & 0.44 & -0.71 & 0 & 0 & 3.69 & 3.88 \\ 0 & 0 & -0.71 & 2.06 & 0 & 0 & -5.15 & -0.13 \\ 0.44 & -0.71 & 0 & 0 & 3.69 & 3.88 & 0 & 0 \\ -0.71 & 2.06 & 0 & 0 & -5.15 & -0.13 & 0 & 0 \end{bmatrix} \\ C_{M5} &= \begin{bmatrix} 0 & 0 & 0.44 & -0.71 & 0 & 0 & 5.30 & -4.07 \\ 0 & 0 & -0.71 & 2.06 & 0 & 0 & -17.88 & 26.83 \\ 0.44 & -0.71 & 0 & 0 & 5.30 & -4.07 & 0 & 0 \\ -0.71 & 2.06 & 0 & 0 & -17.88 & 26.83 & 0 & 0 \end{bmatrix} \\ C_{M6} &= \begin{bmatrix} 0 & 0 & 0.44 & -0.71 & 0 & 0 & -1.31 & 8.26 \\ 0 & 0 & -0.71 & 2.06 & 0 & 0 & 12.31 & -26.98 \\ 0.44 & -0.71 & 0 & 0 & -1.31 & 8.26 & 0 & 0 \\ -0.71 & 2.06 & 0 & 0 & 12.31 & -26.98 & 0 & 0 \end{bmatrix} \end{aligned}$$

$$C_{M7} = \begin{bmatrix} 0 & 0 & 0.44 & -0.71 & 0 & 0 & 3.69 & 3.88 \\ 0 & 0 & -0.71 & 2.06 & 0 & 0 & -5.15 & -0.13 \\ 0.44 & -0.71 & 0 & 0 & 3.69 & 3.88 & 0 & 0 \\ -0.71 & 2.06 & 0 & 0 & -5.15 & -0.13 & 0 & 0 \end{bmatrix}$$

$$C_{M8} = \begin{bmatrix} 0 & 0 & 0.44 & -0.71 & 0 & 0 & -1.31 & 8.26 \\ 0 & 0 & -0.71 & 2.06 & 0 & 0 & 12.31 & -26.98 \\ 0.44 & -0.71 & 0 & 0 & -1.31 & 8.26 & 0 & 0 \\ -0.71 & 2.06 & 0 & 0 & 12.31 & -26.98 & 0 & 0 \end{bmatrix}$$

$$C_{M9} = \begin{bmatrix} 0 & 0 & 0.44 & -0.71 & 0 & 0 & 5.30 & -4.07 \\ 0 & 0 & -0.71 & 2.06 & 0 & 0 & -17.88 & 26.83 \\ 0.44 & -0.71 & 0 & 0 & 5.30 & -4.07 & 0 & 0 \\ -0.71 & 2.06 & 0 & 0 & -17.88 & 26.83 & 0 & 0 \end{bmatrix}$$

Dado que las matrices C_{M1}, \dots, C_{M9} son de rango pleno (ver Apéndice B.1), se considera que los subsistemas locales son controlables.

4.3.2 Cálculo de ganancias

Una vez comprobada la funcionalidad de la linealización del sistema, a través de la implementación con el controlador PID, se utilizaron las diversas matrices A y B de cada uno de los nueve subsistemas lineales para calcular las ganancias de cada subsistema a través de las LMIs (ver Apéndice B.2). Para esto se utilizó el "LMI Control Toolbox de MATLAB", el cual nos proporciona una solución viable a las desigualdades propuestas. Es a través de éste que se determina la existencia de una matriz P_C simétrica común definida positiva, lo cual es prueba de la estabilidad del sistema:

$$P_C = 1 \cdot 10^{-8} \cdot \begin{bmatrix} 0.6379 & -0.0660 & -0.0001 & 0.0001 \\ -0.0660 & 0.2248 & -0.0000 & 0.0000 \\ -0.0001 & -0.0000 & 0.0000 & -0.0000 \\ 0.0001 & 0.0000 & -0.0000 & 0.0000 \end{bmatrix} \quad (4.4)$$

De esta matriz P_C se extraen los siguientes polos, a través del solucionador *feasp*, como se menciona en la sección 2.3.1:

$$\begin{aligned}
 i_1 &= 1 \cdot 10^4 \begin{bmatrix} -0.0009 + 1.5192i \\ -0.0009 - 1.5192i \\ -0.0007 + 0.0000i \\ -0.0002 + 0.0000i \end{bmatrix} & i_2 &= 1 \cdot 10^4 \begin{bmatrix} 0.0036 + 1.4931i \\ 0.0036 - 1.4931i \\ -0.0007 + 0.0000i \\ -0.0002 + 0.0000i \end{bmatrix} \\
 i_3 &= 1 \cdot 10^4 \begin{bmatrix} 0.0036 + 1.4931i \\ 0.0036 - 1.4931i \\ -0.0007 + 0.0000i \\ -0.0002 + 0.0000i \end{bmatrix} & i_4 &= 1 \cdot 10^3 \begin{bmatrix} -0.0127 + 8.2606i \\ -0.0127 - 8.2606i \\ -0.0066 + 0.0000i \\ -0.0019 + 0.0000i \end{bmatrix} \\
 i_5 &= 1 \cdot 10^3 \begin{bmatrix} -0.0124 + 8.2607i \\ -0.0124 - 8.2607i \\ -0.0066 + 0.0000i \\ -0.0019 + 0.0000i \end{bmatrix} & i_6 &= 1 \cdot 10^3 \begin{bmatrix} -0.0233 + 8.5114i \\ -0.0233 - 8.5114i \\ -0.0063 + 0.0000i \\ -0.0019 + 0.0000i \end{bmatrix} \\
 i_7 &= 1 \cdot 10^3 \begin{bmatrix} -0.0127 - 8.2606i \\ -0.0127 - 8.2606i \\ -0.0066 + 0.0000i \\ -0.0019 + 0.0000i \end{bmatrix} & i_8 &= 1 \cdot 10^3 \begin{bmatrix} -0.0233 - 8.5114i \\ -0.0233 - 8.5114i \\ -0.0063 - 0.0000i \\ -0.0019 - 0.0000i \end{bmatrix} \\
 i_9 &= 1 \cdot 10^3 \begin{bmatrix} -0.0124 + 8.2607i \\ -0.0124 - 8.2607i \\ -0.0066 + 0.0000i \\ -0.0019 + 0.0000i \end{bmatrix}
 \end{aligned}$$

Una vez que se comprobó la controlabilidad del sistema y se garantizó la estabilidad del mismo, se procede a extraer las ganancias obtenidas de la misma manera que se extraen los polos, (mediante el solucionador *feasp* capítulo 2.3.1, las cuales resultan de la siguiente manera:

$$\begin{aligned}
 K_1 &= 1 \cdot 10^4 \begin{bmatrix} -6.94 & 8.29 & -0.98 & 4.14 \\ -7.41 & 2.67 & -1.05 & 1.33 \end{bmatrix} \\
 K_2 &= 1 \cdot 10^4 \begin{bmatrix} 6.72 & -8.16 & 0.95 & -4.08 \\ 7.24 & -2.64 & 1.03 & -1.31 \end{bmatrix} \\
 K_3 &= 1 \cdot 10^4 \begin{bmatrix} 6.72 & -8.16 & 0.95 & -4.08 \\ 7.24 & -2.64 & 1.03 & -1.31 \end{bmatrix} \\
 K_4 &= 1 \cdot 10^4 \begin{bmatrix} -1.17 & 3.65 & -0.17 & 1.92 \\ -5.98 & 0.74 & -0.91 & 0.39 \end{bmatrix} \\
 K_5 &= 1 \cdot 10^4 \begin{bmatrix} -1.18 & 3.65 & -0.17 & 1.92 \\ -6.03 & 0.74 & -0.91 & 0.39 \end{bmatrix} \\
 K_6 &= 1 \cdot 10^4 \begin{bmatrix} -1.17 & 3.76 & -0.18 & 1.98 \\ -5.89 & 0.76 & -0.94 & 0.40 \end{bmatrix} \\
 K_7 &= 1 \cdot 10^4 \begin{bmatrix} -1.17 & 3.65 & -0.17 & 1.92 \\ -5.98 & 0.74 & -0.91 & 0.39 \end{bmatrix} \\
 K_8 &= 1 \cdot 10^4 \begin{bmatrix} -1.17 & 3.76 & -0.18 & 1.98 \\ -5.89 & 0.76 & -0.94 & 0.40 \end{bmatrix} \\
 K_9 &= 1 \cdot 10^4 \begin{bmatrix} -1.18 & 3.65 & -0.17 & 1.92 \\ -6.03 & 0.74 & -0.91 & 0.39 \end{bmatrix}
 \end{aligned}$$

4.4 Método de seguimiento

El controlador difuso es diseñado vía PDC, donde cada regla de control está dada a partir de su correspondiente regla del modelo difuso T-S. Para el método de seguimiento es necesario utilizar el modelo del servocompensador, en el cual se propone una señal de referencia, y en base a ésta es que se calculan las ganancias del controlador difuso.

Por lo tanto, la salida final del controlador “difuso aumentado” está dada como en la ecuación 2.21, desarrollada en el capítulo 2.5, la cual se muestra a continuación:

$$\begin{aligned}
 u(t) &= \frac{\sum_{i=1}^r \omega_i(z(t)) [K_i \quad K_{\alpha}] \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix}}{\sum_{i=1}^r \omega_i(z(t))} \\
 u(t) &= - \sum_{i=1}^r h_i(z(t)) [K_i \quad K_{\alpha}] \begin{bmatrix} x(t) \\ x_c(t) \end{bmatrix}
 \end{aligned}$$

El objetivo de control es que las posiciones de las articulaciones 2 y 4 del robot Mitsubishi PA10, sigan una señal de referencia sinusoidal $y_r(t) = 0.25 * \text{sen}(t)$. Aplicando la transformada de Laplace, se obtendrán los polinomios característicos para cada una de las articulaciones deseadas de manera que se puedan utilizar en el modelo del servocompensador.

La señal de referencia dada será:

$$\begin{bmatrix} q_{d1} \\ q_{d2} \end{bmatrix} = \begin{bmatrix} 0.25 * \text{sen}(t) \\ 0.25 * \text{sen}(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{s^2 + 1} \\ \frac{1}{s^3 + 1} \end{bmatrix} \quad (4.5)$$

donde:

q_{d1} : es la referencia de la articulación 2

q_{d2} : es la referencia de la articulación 4

4.4.1 Sistema aumentado por el modelo del servocompensador

Como se vio en el capítulo 2, para la parte de seguimiento, en el presente trabajo de tesis se utilizará el modelo de un servocompensador como en [2].

Utilizando la señal de referencia propuesta, se determina que la matriz A_c está dada por:

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Ya con el modelo del servocompensador, se debe aumentar cada subsistema como se indica en la ecuación 2.19, obteniendo los siguientes resultados:

$$A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 27.35 & -29.91 & 0 & 0 & 0 & 0 & 0 & 0 \\ -30.75 & 90.57 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.8302 & -1.7802 \\ -1.7802 & 4.7249 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 14.78 & -15.78 & 0 & 0 & 0 & 0 & 0 & 0 \\ -16.61 & 48.93 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.8302 & -1.7802 \\ -1.7802 & 4.7249 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 14.78 & -15.78 & 0 & 0 & 0 & 0 & 0 & 0 \\ -16.61 & 48.93 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$B_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.8302 & -1.7802 \\ -1.7802 & 4.7249 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 25.82 & 10.85 & 0 & 0 & 0 & 0 & 0 & 0 \\ -26.6 & -9.31 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$B_4 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.4456 & -0.7194 \\ -0.7194 & 2.06 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 19.91 & 4.9564 & 0 & 0 & 0 & 0 & 0 & 0 \\ -43.76 & -2.24 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$B_4 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.4456 & -0.7194 \\ -0.7194 & 2.06 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 7.99 & 6.7741 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15.01 & -7.82 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$B_5 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.4456 & -0.7194 \\ -0.7194 & 2.06 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 25.82 & 10.85 & 0 & 0 & 0 & 0 & 0 & 0 \\ -26.6 & -9.31 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$B_6 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.4456 & -0.7194 \\ -0.7194 & 2.06 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A_7 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 7.99 & 6.77 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15.01 & -7.82 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$B_7 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.4456 & -0.7194 \\ -0.7194 & 2.06 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A_c = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 19.9133 & 4.9564 & 0 & 0 & 0 & 0 & 0 & 0 \\ -43.76 & -2.24 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \quad B_y = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.4456 & -0.7194 \\ -0.7194 & 2.06 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

4.4.2 Controlabilidad del sistema

Para poder realizar los cálculos de las matrices de ganancia, es necesario realizar la prueba de controlabilidad, al igual que en el capítulo 4.2.1, donde se determina la condición para que una planta de n -ésimo orden sea controlable (ecuación 4.2)

Evaluando los nuevos parcs calculados con el modelo del servocompensador

$$\begin{aligned} C_{M1} &= (A_1 \ B_1); & C_{M4} &= (A_4 \ B_4); & C_{M7} &= (A_7 \ B_7); \\ C_{M2} &= (A_2 \ B_2); & C_{M5} &= (A_5 \ B_5); & C_{M8} &= (A_8 \ B_8); \\ C_{M3} &= (A_3 \ B_3); & C_{M6} &= (A_6 \ B_6); & C_{M9} &= (A_9 \ B_9); \end{aligned}$$

se obtienen las siguientes matrices de controlabilidad para los subsistemas locales:

$$C_{Mj} \cdot 10^6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & .0001 & -.0002 & 0 & 0 & .0075 & -.0192 & 0 & 0 & 0.7666 & -1.97 \\ 0 & 0 & 0 & 0 & 0 & 0 & -.0002 & .0005 & 0 & 0 & -.0192 & 0.0495 & 0 & 0 & -1.97 & 5.37 \\ 0 & 0 & 0 & 0 & .0001 & -.0002 & 0 & 0 & .0075 & -.0192 & 0 & 0 & 0.7666 & -1.97 & 0 & 0 \\ 0 & 0 & 0 & 0 & -.0002 & .0005 & 0 & 0 & -.0192 & -.0495 & 0 & 0 & -1.97 & 5.37 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.0001 & .0002 & 0 & 0 & -0.0074 & 0.0190 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.0001 & .0002 & 0 & 0 & -.0074 & .0190 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .0002 & -.0005 & 0 & 0 & 0.0190 & 0.0490 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .0002 & -.0005 & 0 & 0 & .0190 & -.0490 & 0 & 0 \end{bmatrix}$$

$$C_{w1} = 1 \times 10^5 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & .0004 & -.0010 & 0 & 0 & .0219 & -.0561 & 0 & 0 & 1.2092 & -3.1085 \\ 0 & 0 & 0 & 0 & 0 & 0 & -.0010 & .0026 & 0 & 0 & -.0561 & 0.1444 & 0 & 0 & -3.1086 & 7.9975 \\ 0 & 0 & 0 & 0 & .0004 & -.0010 & 0 & 0 & .0219 & -.0561 & 0 & 0 & 1.2092 & -3.1085 & 0 & 0 \\ 0 & 0 & 0 & 0 & -.0010 & .0026 & 0 & 0 & -.0561 & -.1444 & 0 & 0 & -3.1086 & 7.9975 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.0004 & .0010 & 0 & 0 & -.0215 & 0.0551 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.0004 & .0010 & 0 & 0 & -.0215 & .0551 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .0010 & -.0026 & 0 & 0 & 0.0551 & -.01418 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .0010 & 0 & 0 & 0 & .0190 & -.0490 & 0 & 0 \end{bmatrix}$$

$$C_{w2} = 1 \times 10^5 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & .0004 & -.0010 & 0 & 0 & .0219 & -.0561 & 0 & 0 & 1.2092 & -3.1085 \\ 0 & 0 & 0 & 0 & 0 & 0 & -.0010 & .0026 & 0 & 0 & -.0561 & 0.1444 & 0 & 0 & -3.1086 & 7.9975 \\ 0 & 0 & 0 & 0 & .0004 & -.0010 & 0 & 0 & .0219 & -.0561 & 0 & 0 & 1.2092 & -3.1085 & 0 & 0 \\ 0 & 0 & 0 & 0 & -.0010 & .0026 & 0 & 0 & -.0561 & -.1444 & 0 & 0 & -3.1086 & 7.9975 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.0004 & .0010 & 0 & 0 & -.0215 & 0.0551 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.0004 & .0010 & 0 & 0 & -.0215 & .0551 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .0010 & -.0026 & 0 & 0 & 0.0551 & -.01418 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .0010 & -.0026 & 0 & 0 & .0551 & -.1418 & 0 & 0 \end{bmatrix}$$

$$C_{w3} = 1 \times 10^5 \begin{bmatrix} 0 & 0 & .0004 & -.0007 & 0 & 0 & .0037 & -.0039 & 0 & 0 & .0395 & .0988 & 0 & 0 & .4745 & 1.4431 \\ 0 & 0 & -.0007 & .0021 & 0 & 0 & -.0052 & -.0001 & 0 & 0 & -.0504 & -.1020 & 0 & 0 & -.5829 & -1.6770 \\ .0004 & -.0007 & 0 & 0 & .0037 & .0039 & 0 & 0 & .0395 & .0988 & 0 & 0 & .4745 & -1.4431 & 0 & 0 \\ -.0007 & .0021 & 0 & 0 & -.0052 & -.0001 & 0 & 0 & -.0504 & -.1020 & 0 & 0 & -.5829 & -1.6770 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & -.0033 & -.0046 & 0 & 0 & -.00363 & -.00942 \\ 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & -.0033 & .0046 & 0 & 0 & -.0363 & -.0942 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & .0044 & .0022 & 0 & 0 & 0.0455 & .1998 \\ 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & .0044 & -.0022 & 0 & 0 & .0459 & .0998 & 0 & 0 \end{bmatrix}$$

$$C_{w4} = 1 \times 10^5 \begin{bmatrix} 0 & 0 & .0004 & -.0007 & 0 & 0 & .0053 & -.0041 & 0 & 0 & .0171 & .0515 & 0 & 0 & -.6127 & 1.6187 \\ 0 & 0 & -.0007 & .0021 & 0 & 0 & -.0179 & .0268 & 0 & 0 & -.01921 & .1179 & 0 & 0 & -.3148 & -2.5373 \\ .0004 & -.0007 & 0 & 0 & .0053 & -.0041 & 0 & 0 & .0171 & .0515 & 0 & 0 & -.6127 & 1.6187 & 0 & 0 \\ -.0007 & .0021 & 0 & 0 & -.0179 & .0268 & 0 & 0 & -.01921 & .1179 & 0 & 0 & -.3148 & -2.5373 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & -.0049 & .0034 & 0 & 0 & -.0122 & -.0553 \\ 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & -.0049 & .0034 & 0 & 0 & -.0122 & -.0553 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & .0172 & -.0248 & 0 & 0 & 0.1750 & -.0932 \\ 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & .0172 & -.0248 & 0 & 0 & .1750 & -.0932 & 0 & 0 \end{bmatrix}$$

$$C_{w5} = 1 \times 10^5 \begin{bmatrix} 0 & 0 & .0004 & -.0007 & 0 & 0 & -.0013 & .0083 & 0 & 0 & .0730 & -.1167 & 0 & 0 & -.2031 & 1.3375 \\ 0 & 0 & -.0007 & .0021 & 0 & 0 & .0123 & -.0270 & 0 & 0 & -.01160 & .3351 & 0 & 0 & 2.0030 & -4.3739 \\ .0004 & -.0007 & 0 & 0 & -.0013 & -.0083 & 0 & 0 & .0730 & -.1167 & 0 & 0 & -.2031 & 1.3375 & 0 & 0 \\ -.0007 & .0021 & 0 & 0 & .0123 & -.0270 & 0 & 0 & -.01160 & .3351 & 0 & 0 & 2.0030 & -4.3739 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & .0018 & -.0090 & 0 & 0 & -.0747 & .1257 \\ 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & .0018 & -.0090 & 0 & 0 & -.0747 & .1257 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & -.0130 & .0291 & 0 & 0 & 0.1291 & -.3642 \\ 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & -.0130 & .0291 & 0 & 0 & .1291 & -.3642 & 0 & 0 \end{bmatrix}$$

$$C_{A1} = 1 \cdot 10^3 \begin{bmatrix} 0 & 0 & .0004 & -.0007 & 0 & 0 & .0037 & .0039 & 0 & 0 & .0395 & .0988 & 0 & 0 & .4745 & 1.4431 \\ 0 & 0 & -.0007 & .0021 & 0 & 0 & -.0052 & -.0001 & 0 & 0 & -.0504 & -.1020 & 0 & 0 & -.5829 & -1.6770 \\ .0004 & -.0007 & 0 & 0 & .0037 & .0039 & 0 & 0 & .0395 & .0988 & 0 & 0 & .4745 & 1.4431 & 0 & 0 \\ -.0007 & .0021 & 0 & 0 & -.0052 & -.0001 & 0 & 0 & -.0504 & -.1020 & 0 & 0 & -.5829 & -1.6770 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & -.0033 & -.0046 & 0 & 0 & -.0363 & -.0942 \\ 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & -.0033 & -.0046 & 0 & 0 & -.0363 & -.0942 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & .0044 & .0022 & 0 & 0 & .0459 & .0998 \\ 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & .0044 & .0022 & 0 & 0 & .0459 & .0998 & 0 & 0 \end{bmatrix}$$

$$C_{A2} = 1 \cdot 10^3 \begin{bmatrix} 0 & 0 & .0004 & -.0007 & 0 & 0 & -.0013 & .0083 & 0 & 0 & .0730 & .1167 & 0 & 0 & -.2031 & 1.3375 \\ 0 & 0 & -.0007 & .0021 & 0 & 0 & .0123 & -.0270 & 0 & 0 & -.0160 & .3351 & 0 & 0 & 2.0030 & -4.3739 \\ .0004 & -.0007 & 0 & 0 & -.0013 & .0083 & 0 & 0 & .0730 & .1167 & 0 & 0 & -.2031 & 1.3375 & 0 & 0 \\ -.0007 & .0021 & 0 & 0 & .0123 & -.0270 & 0 & 0 & -.0160 & .3351 & 0 & 0 & 2.0030 & 4.3739 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & .0018 & -.0090 & 0 & 0 & -.00747 & .3257 \\ 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & .0018 & -.0090 & 0 & 0 & -.0747 & .3257 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & -.0130 & .0291 & 0 & 0 & .01291 & -.3642 \\ 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & .0130 & .0291 & 0 & 0 & .01291 & .3642 & 0 & 0 \end{bmatrix}$$

$$C_{A3} = 1 \cdot 10^3 \begin{bmatrix} 0 & 0 & .0004 & -.0007 & 0 & 0 & .0253 & .0041 & 0 & 0 & .0171 & .0519 & 0 & 0 & -.5127 & 1.6187 \\ 0 & 0 & -.0007 & .0021 & 0 & 0 & .0179 & .0268 & 0 & 0 & -.01321 & .1179 & 0 & 0 & -.3148 & -2.5373 \\ .0004 & -.0007 & 0 & 0 & .0053 & -.0041 & 0 & 0 & .0171 & .0519 & 0 & 0 & -.5127 & 1.6187 & 0 & 0 \\ -.0007 & .0021 & 0 & 0 & -.0179 & .0268 & 0 & 0 & -.01321 & .1179 & 0 & 0 & -.3148 & -2.5373 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & -.0049 & .0034 & 0 & 0 & -.00122 & -.0553 \\ 0 & 0 & 0 & 0 & -.0004 & .0007 & 0 & 0 & .0049 & .0034 & 0 & 0 & -.0122 & -.0553 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & .0172 & -.0248 & 0 & 0 & .01750 & -.0932 \\ 0 & 0 & 0 & 0 & .0007 & -.0021 & 0 & 0 & .0172 & -.0248 & 0 & 0 & .1750 & -.0932 & 0 & 0 \end{bmatrix}$$

Las cuales son de rango pleno ($n = 8$), por lo tanto todos los subsistemas lineales locales para seguimiento son controlables, y por consecuencia, el sistema en general también.

De la misma manera, a través del LMI Control Toolbox, se determinó la existencia de una matriz simétrica P_c común definida positiva, lo cual es prueba de la estabilidad del sistema, y está dada como:

$$P_c = \begin{bmatrix} 0.0015 & 0.0006 & 0 & 0 & -0.0252 & -0.0103 & -0.0097 & -0.0040 \\ 0.0006 & 0.0003 & 0 & 0 & -0.0096 & -0.0039 & -0.0052 & -0.0021 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0252 & -0.0096 & 0 & 0 & 0.8231 & 0.2818 & 0.2970 & 0.1061 \\ -0.0103 & -0.0039 & 0 & 0 & 0.2818 & 0.1161 & 0.1063 & 0.0435 \\ -0.0097 & -0.0052 & 0 & 0 & 0.2970 & 0.1063 & 0.2083 & 0.0619 \\ -0.0040 & -0.0021 & 0 & 0 & 0.1061 & 0.0435 & 0.0619 & 0.0261 \end{bmatrix}$$

4.4.3 Cálculo de ganancias

Con las matrices obtenidas con el modelo aumentado, y comprobada la estabilidad del sistema mediante la matriz P_C , se determinan las ganancias para el controlador difuso, mediante los eigenvalores de $(A_i + B_i K_i = 0)$ los cuales están localizados como se explica a continuación:

$$\begin{aligned}
 i_1 &= \begin{bmatrix} 7+24i \\ -7-24i \\ -5+24i \\ -5-24i \\ -4+2i \\ -4-2i \\ -4+2i \\ -4-2i \end{bmatrix} &
 i_2 &= \begin{bmatrix} -5+25i \\ -5-25i \\ -7+24i \\ -7-24i \\ -4+2i \\ -4-2i \\ -4+2i \\ -4-2i \end{bmatrix} &
 i_3 &= \begin{bmatrix} -5+25i \\ -5-25i \\ -7+24i \\ -7-24i \\ -4+2i \\ -4-2i \\ -4+2i \\ -4-2i \end{bmatrix} &
 i_4 &= \begin{bmatrix} -7+24i \\ -7-24i \\ -7+24i \\ -7-24i \\ -4+2i \\ -4-2i \\ -4+2i \\ -4-2i \end{bmatrix} &
 i_5 &= \begin{bmatrix} -7+24i \\ -7-24i \\ -7+24i \\ -7-24i \\ -4+2i \\ -4-2i \\ -4+2i \\ -4-2i \end{bmatrix} \\
 i_6 &= \begin{bmatrix} -7+24i \\ -7-24i \\ -7+24i \\ -7-24i \\ -4+2i \\ -4-2i \\ -4+2i \\ -4-2i \end{bmatrix} &
 i_7 &= \begin{bmatrix} -7+24i \\ -7-24i \\ -7+24i \\ -7-24i \\ -4+2i \\ -4-2i \\ -4+2i \\ -4-2i \end{bmatrix} &
 i_8 &= \begin{bmatrix} -7+24i \\ -7-24i \\ -7+24i \\ -7-24i \\ -4+2i \\ -4-2i \\ -4+2i \\ -4-2i \end{bmatrix} &
 i_9 &= \begin{bmatrix} -7+24i \\ -7-24i \\ -7+24i \\ -7-24i \\ -4+2i \\ -4-2i \\ -4+2i \\ -4-2i \end{bmatrix}
 \end{aligned}$$

Con los eigenvalores colocados en los puntos anteriores, se obtienen las siguientes ganancias para cada uno de los nueve subsistemas (ver Apéndice B.3):

$$\begin{aligned}
 K_1 &= \begin{bmatrix} 5093.9 & 1765 & 122.95 & 47.243 & -109040 & -38937 & -36397 & -13349 \\ 1918.4 & 848.21 & 45.754 & 22.468 & -41394 & -14748 & -17079 & -6274.7 \end{bmatrix} \\
 K_2 &= \begin{bmatrix} 4944.3 & 2053.2 & 136.67 & 62.986 & -99984 & -36241 & -36438 & -14846 \\ 1849.2 & 954.67 & 51.563 & 28.177 & -37599 & -13585 & -17260 & -6876.7 \end{bmatrix} \\
 K_3 &= \begin{bmatrix} 4944.3 & 2053.2 & 136.67 & 62.986 & -99984 & -36241 & -36438 & -14846 \\ 1849.2 & 954.67 & 51.563 & 28.177 & -37599 & -13585 & -17260 & -6876.7 \end{bmatrix}
 \end{aligned}$$

$$K_4 = \begin{bmatrix} 4132.4 & 1435.5 & 118.69 & 40.83 & -83768 & -30619 & -28967 & -10563 \\ 1425.1 & 876.59 & 41.307 & 25.292 & -29144 & -10655 & -17990 & -6564.8 \end{bmatrix}$$

$$K_5 = \begin{bmatrix} 4062.2 & 1440.8 & 117.45 & 41.353 & -83884 & -30624 & -29450 & -10741 \\ 1390.1 & 881.61 & 40.802 & 25.565 & -29137 & -10639 & -18130 & -6620 \end{bmatrix}$$

$$K_6 = \begin{bmatrix} 4115.2 & 1413.4 & 118.12 & 41.326 & -83900 & -30637 & -28602 & -10509 \\ 1439.3 & 869.65 & 41.146 & 25.53 & -29199 & -10660 & -17853 & -6544 \end{bmatrix}$$

$$K_7 = \begin{bmatrix} 4132.4 & 1435.5 & 118.69 & 40.83 & -83768 & -30619 & -28967 & -10563 \\ 1425.1 & 876.59 & 41.307 & 25.292 & -29144 & -10655 & -17990 & -6564.8 \end{bmatrix}$$

$$K_8 = \begin{bmatrix} 4115.2 & 1413.4 & 118.12 & 41.326 & -83900 & 30637 & 28602 & -10509 \\ 1439.3 & 869.65 & 41.146 & 25.53 & -29199 & -10660 & -17853 & -6544 \end{bmatrix}$$

$$K_9 = \begin{bmatrix} 4062.2 & 1440.8 & 117.45 & 41.353 & -83884 & -30624 & 29450 & -10741 \\ 1390.1 & 881.61 & 40.802 & 25.565 & -29137 & -10639 & -18130 & -6620 \end{bmatrix}$$

Capítulo 5

Visión

La investigación actual relacionada al control de robots se basa fundamentalmente en la incorporación de información sensorial externa que, a la par de la utilización de algoritmos avanzados de control, permiten la realización de diversas tareas en medios cada vez más hostiles y complicados. Esto da lugar a la apertura de nuevos campos de aplicación para sistemas robóticos.

Entre la gran variedad de sensores externos que se utilizan en la actualidad, se puede mencionar que los sensores de visión proveen una información bastante completa del medio de trabajo, ya que es posible recopilar gran cantidad de información sobre un entorno, como lo es la distancia hacia un objeto, su forma, color y textura, a través de una cámara.

El control basado en visión de sistemas robóticos ha despertado mucho interés en estos últimos años debido a la reducción de costos de las cámaras de visión y de todo el sistema de procesamiento asociado.

El control servovisual se puede clasificar de la siguiente manera:

- Control basado en posición o IBVS (por sus siglas en inglés Image-Based Visual Servoing)
- Control basada en imagen o PBVS (por sus siglas en inglés Position-Based Visual Servoing)

En esquemas de control basados en **posición**, se extraen las características de la imagen las cuales se utilizan junto al modelo geométrico del objeto y el conocimiento del modelo de la cámara para estimar la posición y orientación del objeto de interés con respecto a la cámara. La acción de control se calcula para reducir los errores de posición y orientación estimados [15].

En el control basado en **imagen**, los valores de las acciones de control se calculan directamente sobre la base de las características de la imagen [15]. Este tipo de esquema puede reducir retardos de computación, eliminar la necesidad de interpretación de la

imagen y eliminar errores debidos a las incertidumbres en los modelos de los sensores y de la calibración de la cámara.

Por último, los sistemas se puede clasificar de acuerdo a la posición que ocupe el sensor de visión en: "sistemas con cámara fija"(eye to hand), o "sistemas con cámara en mano"(eye in hand) [1]. En la Figura 14 se representa este tipo de clasificación, en donde el robot puede tener montada la cámara o ésta puede estar fija observando el entorno de trabajo completo.

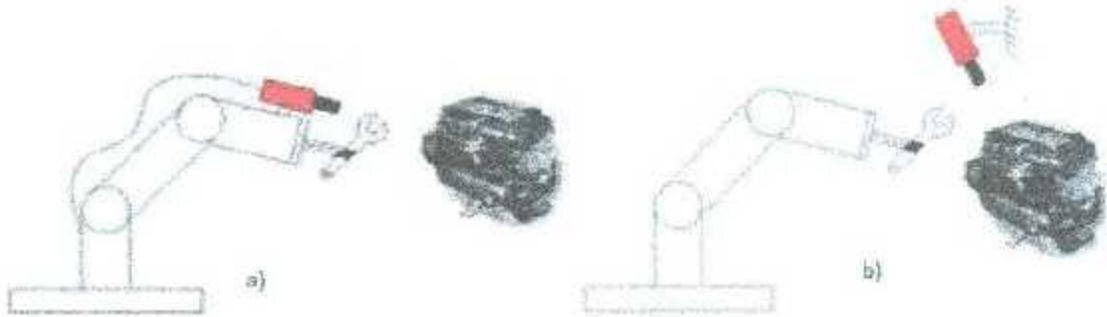


Figura 14. a) Sistema con cámara en mano; b) sistema con cámara fija

En este trabajo se utilizará el sistema "cámara fija", y un esquema de control basado en posición.

El presente capítulo considera que el control se realiza en el espacio articular y no en el de imagen.

5.1 Sistemas de Visión

La visión artificial es el proceso de extraer información del mundo real a partir de imágenes, utilizando como herramienta una cámara y un computador [15].

Los componentes principales de un sistema de visión son por computador son un sensor de imagen y un digitalizador. Un sensor de imagen es un dispositivo físico sensible en una banda del espectro de energía electromagnético que genera una señal eléctrica proporcional al nivel de energía incidente en un instante de tiempo. La señal eléctrica generada es típicamente una señal analógica, por lo que es necesario procesarla mediante un dispositivo digitalizador. Un digitalizador, comúnmente conocido como tarjeta digitalizadora, es un dispositivo capaz de convertir la señal analógica de salida del sensor de imagen en una señal digital, la cual puede ser procesada por un computador [16].

La representación de la información de una imagen puede realizarse de varias formas: a través de imágenes de intensidad (luminosidad), de imágenes térmicas (temperatura) o de imágenes de rango (alcance o profundidad).

La cámara de un sistema de visión por computadora, es el dispositivo que recibe la luz reflejada por la escena y la utiliza para generar imágenes. Como ya se mencionó, el elemento más importante de la cámara es el sensor, el cual transforma la luz incidente en señales eléctricas.

Dicho sensor puede estar construido en base a una tecnología analógica (cámaras de tubo) o en base a una tecnología digital (cámaras de estado sólido). La tecnología digital más utilizada actualmente para la construcción de sensores de imagen es CCD (dispositivo de carga acoplada, por sus siglas en inglés, *Charge-Coupled Device*).

Además de la tecnología de fabricación, un aspecto importante del sensor de una cámara digital es su factor de forma. Los tipos básicos son lineal y matricial. Las cámaras lineales obtienen menos información de la escena, pero poseen una tasa de transferencia más elevada que las cámaras matriciales.

Otros factores a considerar en el sensor de una cámara son su respuesta a las diferentes longitudes de onda, que indica la curva de sensibilidad de la cámara en el espectro; las dimensiones, que indican el número de píxeles con que genera la imagen; la configuración de la ganancia (contraste de la imagen) y del desplazamiento (brillo de la imagen); y finalmente, el ruido que introduce el sensor en la imagen generada debido a las diferentes sensibilidades de cada elemento fotosensible básico.

5.2 Adquisición de imágenes en tiempo real

La primera fase de Visual Servoing es la adquisición y el procesamiento de imágenes. Cuando una cámara captura las imágenes, inmediatamente las procesa en tiempo real y las muestra como un cuadro en una pantalla; analizando propiamente la captura, se pueden observar los parámetros característicos de la imagen, es decir el contorno y la forma. Dichos parámetros vienen dados por valores de intensidad luminosa variante dentro de la escala de grises, que dependen del objeto, del ambiente y de la resolución de la cámara.

La información visual, a diferencia de la información proporcionada por otros tipos de sensores, es muy rica y variada, por lo que requiere complejas transformaciones computacionales antes de que pueda ser utilizada para el control de un sistema robótico. El objetivo de estas transformaciones es la extracción de la información numérica de la imagen, que proporciona una descripción sintética y robusta de los objetos de interés en la escena, a través de los parámetros característicos de la imagen.

5.3 Procesamiento de imágenes

Para poder llevar a cabo la identificación de toda la información visual que obtenemos a través de una cámara, es necesario aplicar a la imagen obtenida ciertas operaciones, las cuales ayudarán a extraer la información necesaria para realizar un control a través de una realimentación visual. A continuación se mencionan las operaciones utilizadas comúnmente para el procesamiento de imágenes:

5.3.1 Separación por color y filtrado de remanentes

Primeramente, es necesario diferenciar el entorno de trabajo, es decir, colocar marcas que distingan unos objetos de otros, otorgando a la cámara la posibilidad de identificar los elementos deseados. Este tipo de marcas podrían ser a través de colores o de formas. Para este trabajo se utilizarán marcas circulares, ya que la geometría del robot lo permite, las cuales se identificarán a través de diferentes colores.

Se utilizará el modelo **RGB**, el cual está formado por los colores primarios de la luz (Rojo, Verde y Azul). La separación por colores se realizará a través del umbralizado y se utilizará alguna técnica de suavización de imagen para tener bordes adecuados y no pixeleados del objeto a observar, por ejemplo la erosión combinada con la dilatación, en operaciones de cerradura y apertura.

En la Figura 15 se expone un ejemplo básico de separación por color, en donde en primera instancia se colocan muestras diversas para después aislar solamente los objetos del color deseado, y por último delimitar correctamente dichos objetos a través del proceso de erosión y dilatación.

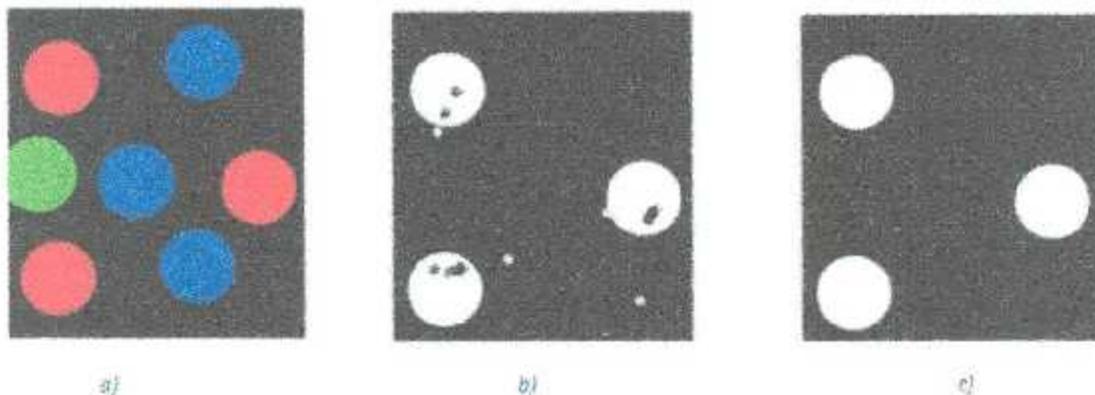


Figura 15.- a) Patrón original; b) aislamiento por umbralización; c) erosión y dilatación

5.3.2 Segmentación

Como se muestra en la Figura 16, una vez realizada la separación por colores se subdivide la imagen en pequeñas regiones, para poder tratar cada objeto como si fuese una imagen individual. Algunas técnicas existentes para este propósito son la asociación de píxeles vecinos (conexos)

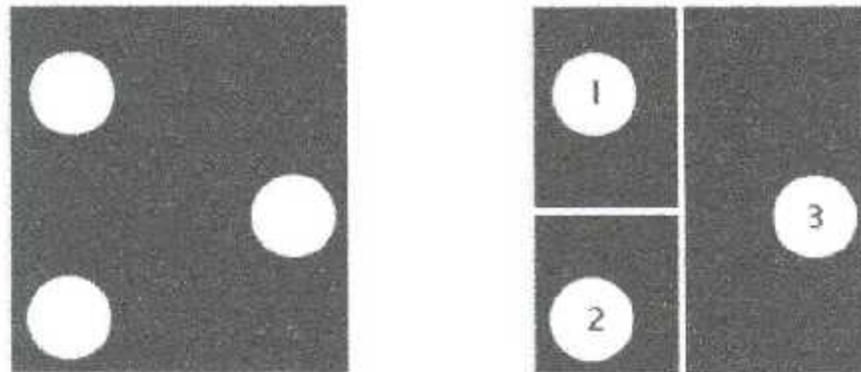


Figura 16.- Segmentación de imágenes

5.3.3 Detección del centroide

Luego de separar los objetos deseados, se procederá a localizar sus centroides para operar con un solo referente coordenado de su posición (u, v). Para este trabajo de tesis la geometría de los objetos a monitorear permite que el cálculo del centroide sea relativamente sencillo, como se muestra en el ejemplo de la Figura 17, pero para otros objetos con geometrías complejas, el cálculo del centroide debe de realizarse minuciosamente, a través de fórmulas más complejas.

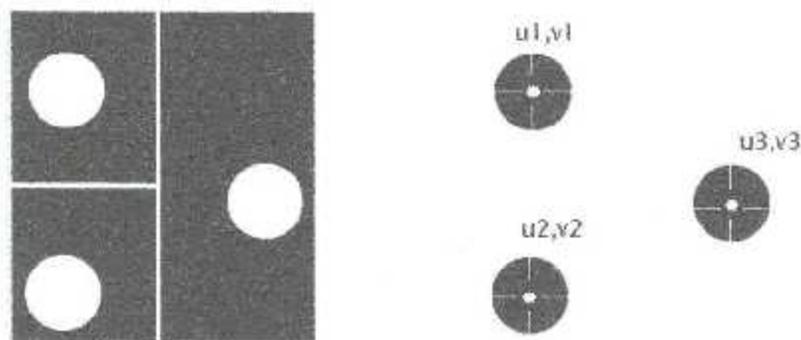


Figura 17.- Ubicación de centroide en una figura

5.4 Control servovisual

En el capítulo 3.2.1 se desarrolló el modelo cinemático directo de posición para el robot Mitsubishi PA10 (limitado a 2 gdl), el cual se utilizará para calcular las posiciones “x, y” de cada una de las articulaciones, una vez que se obtenga el centroide de cada eslabón a través de la adquisición (cámara web) y el procesamiento de la imagen, ya que de esta manera se realimentará el sistema de control, en lugar de utilizar los encoders del robot. En este capítulo se explicarán a detalle cada una de estas operaciones, las cuales forman parte del control Servovisual.

En primer lugar, se debe hacer mención al instrumento a utilizar para la adquisición de imágenes. Ésta se realizará a través de una cámara web, modelo HP HD-3110 de 30 fps (frames per second por sus siglas en inglés, cuadros por segundo, traducido al español) y con una resolución de 720 pixeles, la cual se muestra en la Figura 18.



Figura 18.- Cámara “HP HD-3110”

Para leer la información de la cámara en el programa a desarrollar para el control del robot, es necesario considerar que la información debe ser enviada en tiempo real, de manera que se actualice constantemente.

Es importante recalcar que la interfaz de control para el robot Mitsubishi PA10 diseñada en [7], está desarrollada en MATLAB/Simulink, y que en ésta se utiliza el bloque “PA10 Control” (Apéndice C) el cual se instala en conjunto con el software Quarc versión 2.5 “Quanser Real-Time Rapid Control Prototyping Software”, que es el encargado de que el robot funcione en tiempo real.

Mencionado el punto anterior, en la Figura 19 se muestra el bloque “Video Capture”, utilizado para la adquisición de imágenes en tiempo real en MATLAB/Simulink, el cual puede trabajar a la par del bloque de control “PA10 Control” (encargado de

establecer la comunicación entre MATLAB/Simulink y el robot Mitsubishi PA10), ya que ambos son desarrollados por la misma compañía (Quanser).

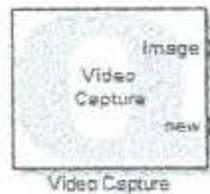


Figura 19.- Bloque para la adquisición de imágenes en tiempo real

A continuación se presenta una breve descripción de los 2 puertos de salida que componen al bloque “Video Capture”, aplicado al robot Mitsubishi PA10-7CE.

- **image:** brinda la secuencia de imágenes adquiridas a través de la cámara en tiempo real. La resolución de dichas imágenes será seleccionada dentro de los componentes del bloque “Video Capture”, los cuales se explican a detalle más adelante.
- **new:** esta salida se produce siempre que una nueva toma está siendo emitida en el puerto de imagen. De otra forma será falso.

Dentro de este bloque existen controles y demás componentes, los cuales son mostrados en la Figura 20.

- **Source of Video:** selecciona la fuente para la adquisición de video, dependiendo del número de dispositivos de captura de imagen o video que se tengan instalados y de sus respectivos paquetes de instalación para MATLAB/Simulink 2014a.
- **Device identifier:** determina el número del dispositivo que se utilizará, haciendo referencia al adaptador al que está ligado.
- **Image resolution:** otorga una lista de posibles resoluciones a elegir para la fuente de video, dependiendo de las características físicas de la misma.
- **Output format:** ofrece diversas opciones para el formato de salida de la imagen adquirida.
- **Output data type:** selecciona el tipo de dato que se contendrá en cada imagen. Puede ser un valor binario, decimal, hexadecimal, etc.
- **Sample time:** se refiere al periodo de muestreo del dispositivo seleccionado. Este rango está determinado en segundos.

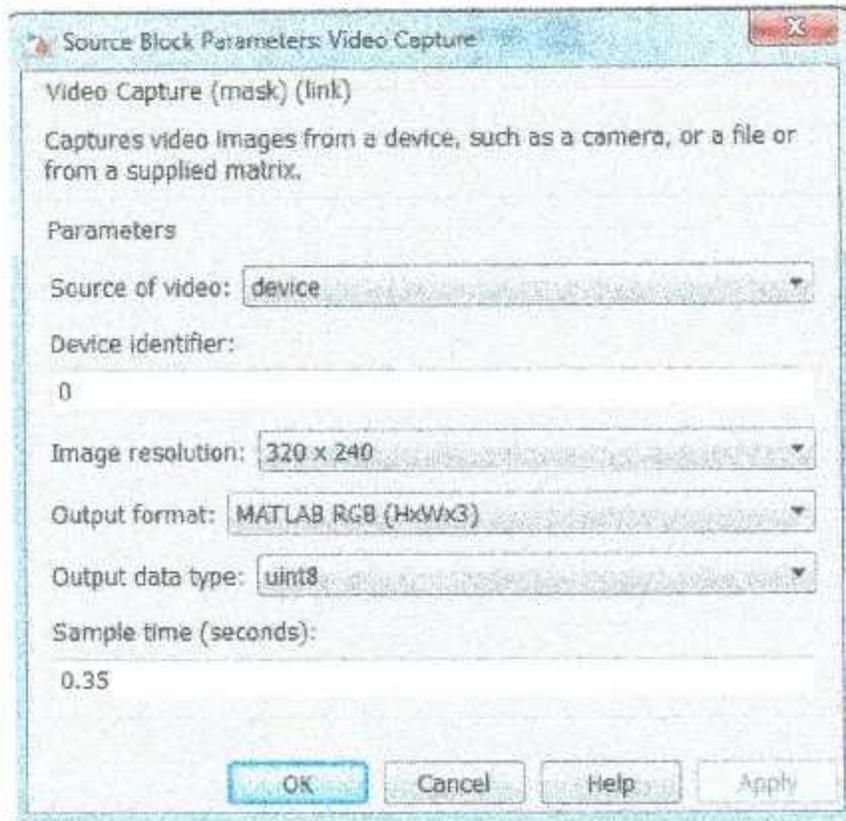


Figura 20.- Ventana "video capture"

El diagrama a bloques del sistema quedará establecido como se muestra en la Figura 21.

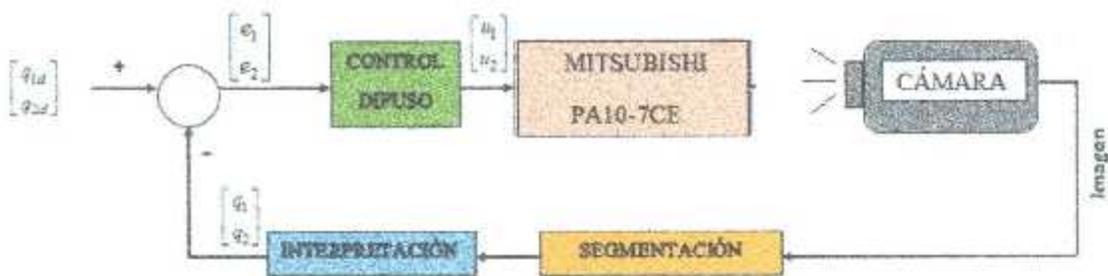


Figura 21.- Diagrama a bloques del sistema

5.5 Separación de colores

El modelo de color RGB (red, green and blue, por sus siglas en inglés) está basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores de luz primarios (rojo verde y azul).

El modelo RGB asigna un valor de intensidad a cada píxel que oscile entre 0 (negro) y 255 (blanco) para cada uno de los componentes RGB de una imagen en color.

Por ejemplo, para adquirir un color rojo brillante se puede combinar el valor R de 246, G de 20 y B de 50. Si el valor de todos los componentes es de 255, el resultado será blanco puro y será negro puro si todos los componentes tienen un valor 0.

De esta manera, una vez que se adquiere una imagen, es necesario separar los componentes RGB para analizarlos por separado, y así poder ubicar el área de interés. Para realizar esto, se utilizaron los parámetros que se muestran en la Tabla 2.

OBJETIVO	Separación por Color	Valor mínimo	Valor máximo
OBJETO VERDE	RED	30	100
	BLUE	122	255
	GREEN	0	150
OBJETO AZUL	RED	30	93
	BLUE	110	255
	GREEN	150	255

Tabla 2.- Segmentación de la imagen

En la Figura 22 se muestran los resultados de la aplicación de los filtros de separación de color sobre el robot Mitsubishi PA10. Se colocaron marcas de color azul y verde sobre el robot, de manera que se pudieran determinar las posiciones de los eslabones 2 y 4. Una vez que se logra separar la marca deseada del resto de la imagen, el programa calcula el centroide de la dicha marca.

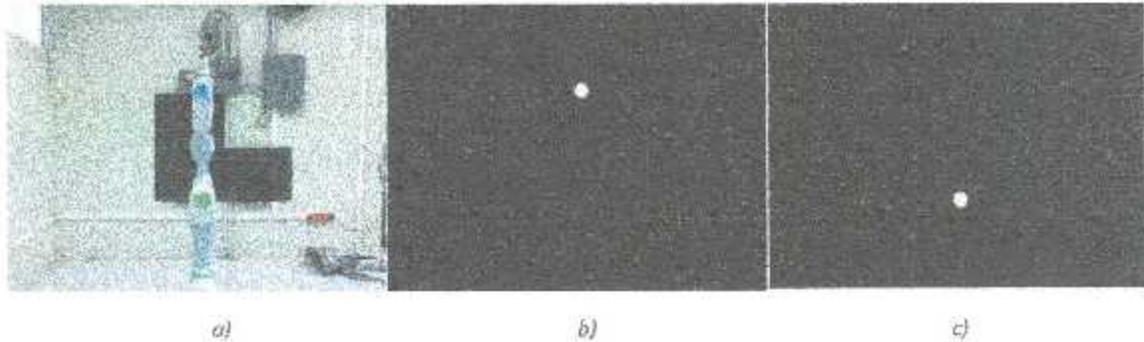


Figura 22.- Segmentación por colores a) imagen original, b) segmentación color azul, c) segmentación color verde

El centroide es obtenido en coordenadas de imagen, dependiendo de la resolución seleccionada para el dispositivo de adquisición de imágenes. La resolución a la que se configuró la cámara web es de 320x240 píxeles, de manera que el ancho de la imagen será de 320 píxeles y su ancho será de 240 píxeles. La cámara se encuentra a una distancia de 1.5 metros de la ubicación del robot, o dicho de otra forma, del plano en el que trabajará el robot (ya que las articulaciones 2 y 4 sólo se mueven en el eje horizontal), lo cual resulta en la siguiente equivalencia:

$$1 \text{ pixel} = 0.56 \text{ centímetros}$$

Se coloca el marco de la base del robot en el punto medio de la imagen, es decir en la posición (160, 240) píxeles, para que la cámara pueda cubrir todo el espacio de trabajo del robot. El periodo de muestreo es seleccionado en base a la velocidad máxima que otorga la cámara de adquisición de imágenes, que por defecto es de 30 fps, equivalente a un periodo de muestreo de 1/30 segundos (aproximadamente 35 ms). Como se mencionó anteriormente, el periodo de muestreo seleccionado para el robot Mitsubishi PA10-7CE es de 5 ms; sin embargo la cámara utilizada sólo puede trabajar con un periodo de 35 ms, por lo que es necesario repetir 7 veces la misma toma para igualar ambos periodos. Para resolver este inconveniente se utilizará el bloque de MATLAB/Simulink "Rate Transition" mostrado en la Figura 23, el cual tiene la función de transmitir la información entre puertos con diferentes velocidades o periodos de muestreo (como es el caso entre la cámara y el robot), de manera que puedan trabajar en conjunto. Lo anterior representa una desventaja para la aplicación en tiempo real, ya que la imagen no es tomada en el mismo instante en que el robot lee instrucciones, lo cual puede introducir ciertos retardos. Sería ideal tener una cámara más rápida (1/200 fps) que pudiera igualar el periodo de muestreo del robot, sin embargo esto representa una mayor inversión, la cual no está dentro del objetivo de este proyecto.

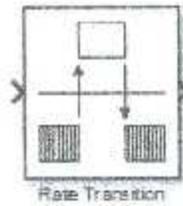


Figura 23.- Bloque "Rate Transition"

Como se menciona anteriormente, el robot Mitsubishi PA10 de 7 grados de libertad es limitado a trabajar solamente con 2 grados de libertad, lo que hace que se mueva exclusivamente en un plano. Se deben de establecer el marco de la cámara y el marco del robot (ver Figura 24), para que de esta manera se puedan realizar los cálculos mencionados en el modelo cinemático directo de posición (Capítulo 3.2.1) para obtener las posiciones de cada articulación.

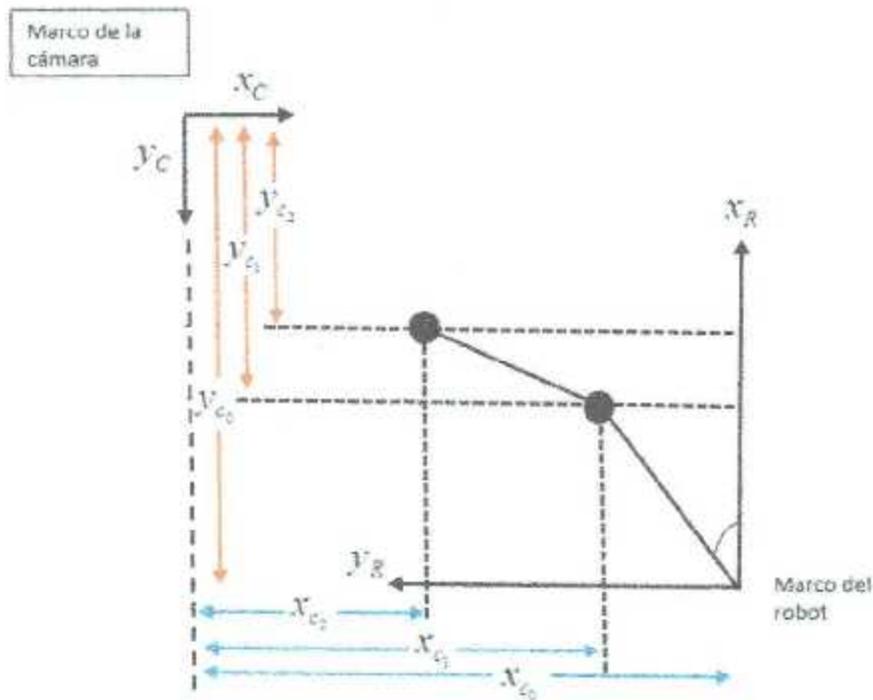


Figura 24.- Marcos establecidos para el sistema

En relación con la Figura 24, se tiene que:

- x_R, y_R determinan el marco del robot
- x_c, y_c determinan el marco de la cámara
- x_{c1}, y_{c1} , distancia del marco de la cámara al eslabón 1
- x_{c2}, y_{c2} , distancia del marco de la cámara al eslabón 2

Una vez ubicado el robot, se procede a calcular las coordenadas articulares de cada eslabón, de la siguiente manera:

$$\begin{aligned}
 x_1 &= y_{c0} - y_{c1} \\
 x_2 &= y_{c0} - y_{c2} \\
 y_1 &= x_{c0} - x_{c1} \\
 y_2 &= x_{c0} - x_{c2} \\
 \theta_1 &= \text{atan2}\left(\frac{y_1}{x_1}\right) \\
 \theta_2 &= \text{atan2}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) - \theta_1
 \end{aligned} \tag{5.1}$$

Calculados los ángulos de cada articulación se procede a realimentarlos al sistema de control.

Capítulo 6

Resultados

6.1 Simulación control difuso T-S

En este apartado se presentan los resultados obtenidos del estudio realizado en simulación (MATLAB/Simulink), considerando el modelo dinámico del robot Mitsubishi PA10-7 obtenido en [13]. Para simular correctamente el sistema no se tomó en cuenta la parte de realimentación visual, de manera que la posición y la velocidad (necesarias para realimentar el sistema) son obtenidas del modelo dinámico.

6.1.1 Regulación

Para realizar correctamente el procedimiento de regulación se incluyó una pequeña perturbación a la salida, de manera que se pueda corroborar que el controlador cumpla con su función adecuadamente.

El diagrama a bloques para el controlador difuso T-S para regulación se representa en la Figura 25.

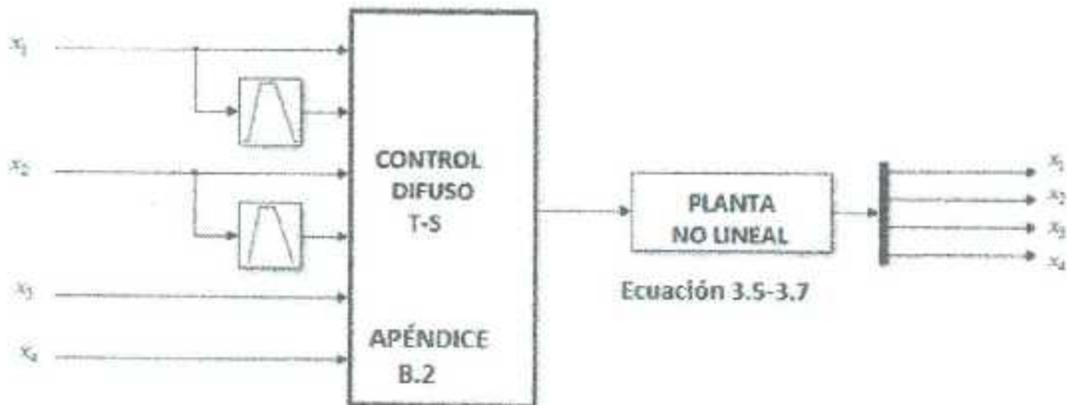
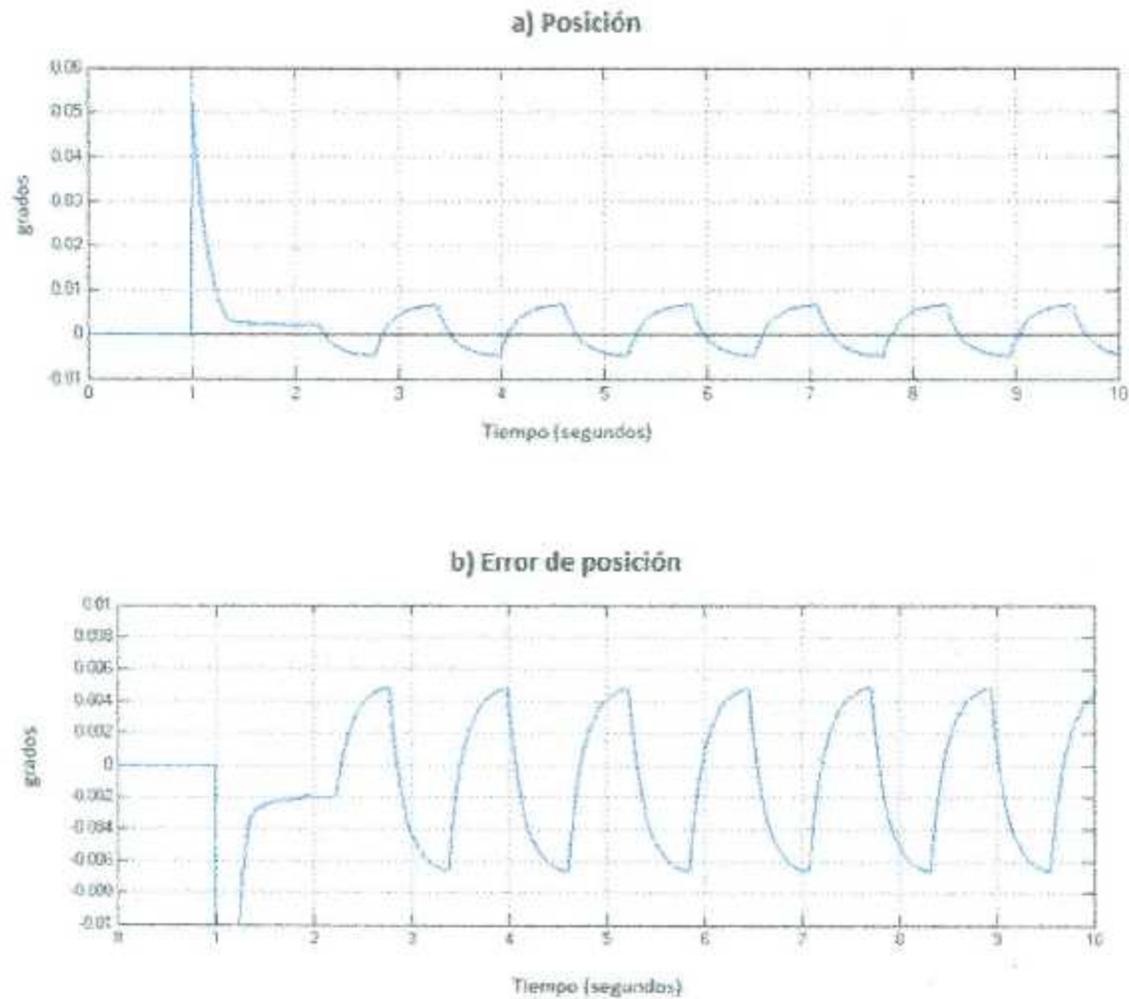


Figura 25.- Diagrama a bloques del controlador difuso T-S para regulación

En la Figura 26 se muestran los resultados obtenidos en simulación para el caso de regulación de la articulación 2. En la gráfica de posición se puede observar el pequeño impulso causado por la perturbación que se colocó a la salida del sistema. Mediante esta perturbación se pretendía que el sistema estuviera en una posición diferente a cero, de manera que se pudiera comprobar que el controlador estaba llevando a cabo la función de regulación.

El error de posición tomado después de la perturbación inducida, oscila en ± 0.006 [grados]. En la gráfica del par aplicado se puede apreciar cómo éste oscila en valores cercanos a cero conforme pasa el tiempo, mostrando algunas variaciones importantes sólo al inicio de la simulación, las cuales son causadas por la perturbación dada al sistema.



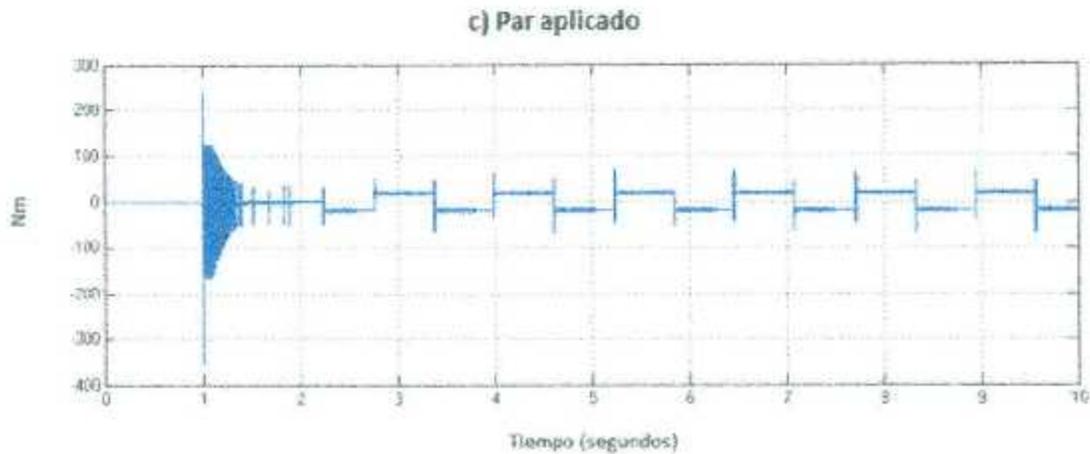
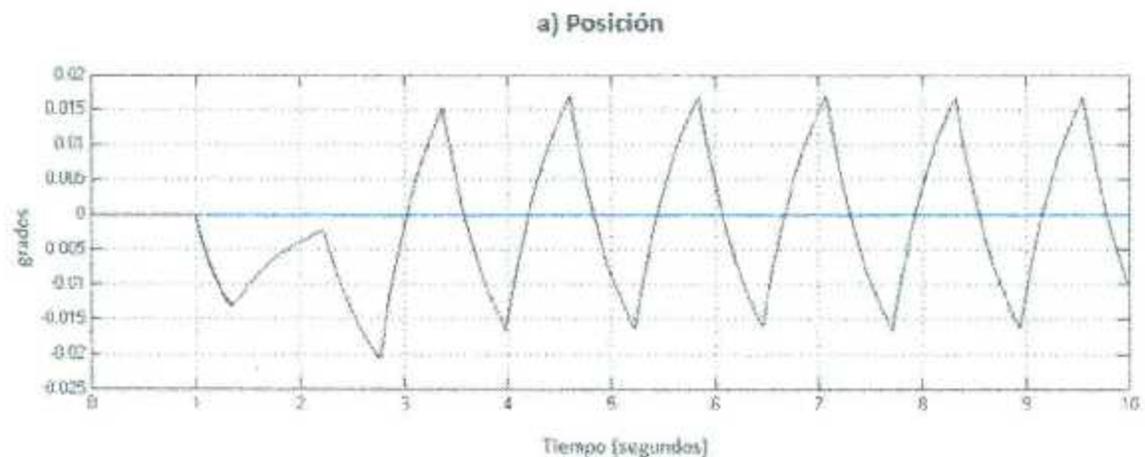


Figura 26.- Resultados regulación articulación 2

En la figura 27 se muestran los resultados obtenidos en simulación para el caso de regulación de la articulación 4. En la gráfica de posición se puede observar que a pesar de que la perturbación es aplicada a la salida de la articulación 2, la articulación 4 presenta también una posición diferente de cero, sin embargo se corrobora que el controlador está llevando a cabo su función y el sistema permanece estable.

El error de posición tomado después de la perturbación inducida a la articulación 2, oscila en -0.02 [grados]. En la gráfica de la señal de control se aprecia cómo dicha señal oscila en valores muy cercanos a cero conforme pasa el tiempo, sin sobrepasar el límite establecido para la articulación que se está graficando.



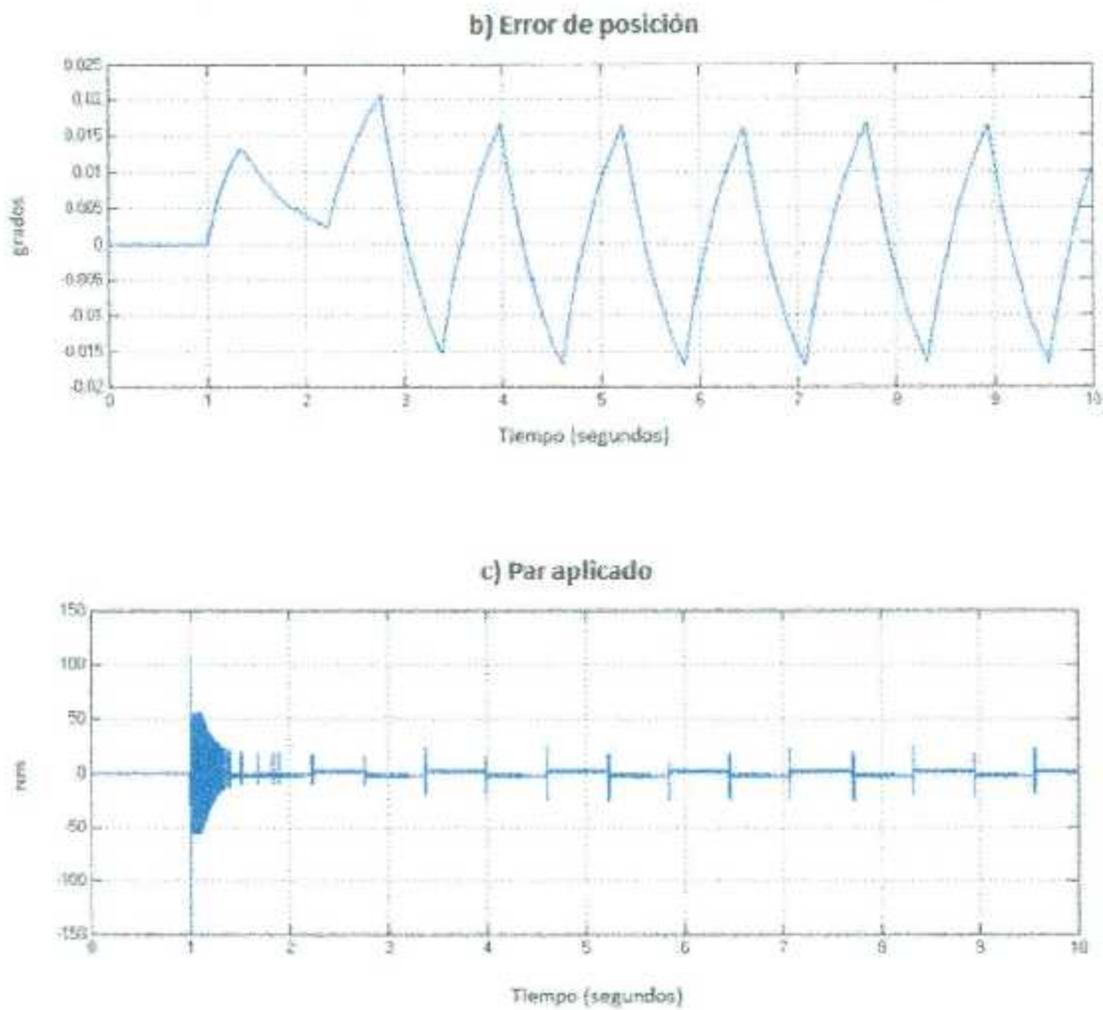


Figura 27.- Resultados regulación articulación 4.

6.1.2 Seguimiento

A continuación se presentan los resultados obtenidos en simulación para la tarea de seguimiento en el robot Mitsubishi PA10-7CE. Para poder realizar esta tarea, es necesario llevar a cabo el procedimiento explicado en el capítulo 4.3.1, relacionado al modelo del servocompensador.

Como primera instancia, se debe de establecer la señal de referencia a seguir, la cual está dada por:

$$\begin{bmatrix} q_{d1} \\ q_{d2} \end{bmatrix} = \begin{bmatrix} 0.25 * \text{sen}(t) \\ 0.25 * \text{sen}(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{s^2 + 1} \\ \frac{1}{s^2 + 1} \end{bmatrix}$$

El diagrama a bloques para el controlador difuso T-S para seguimiento se representa en la Figura 28. Es importante mencionar que para cada señal de referencia dada, existirá un sistema diferente, ya que el controlador obtiene las ganancias en base al modelo del servocompensador, el cual cambia en función a la señal de referencia.

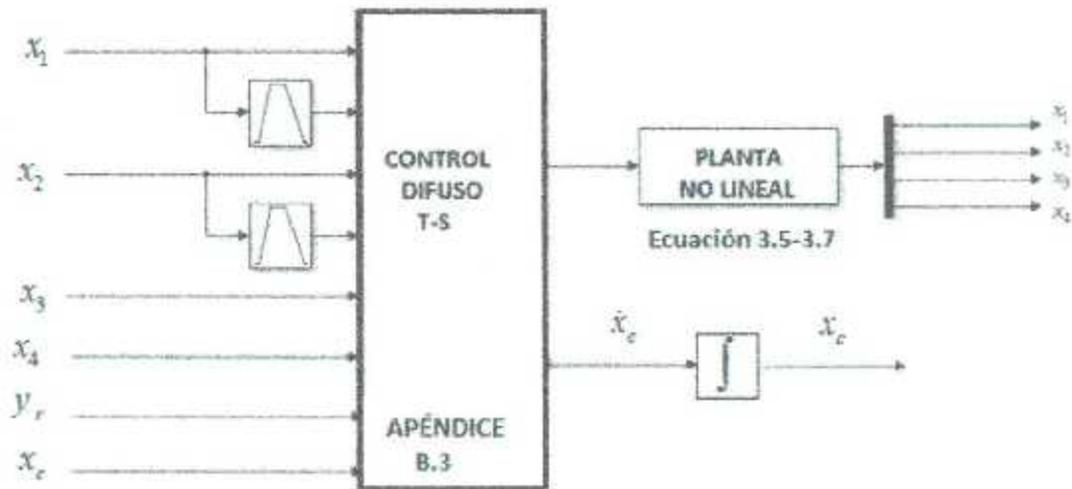


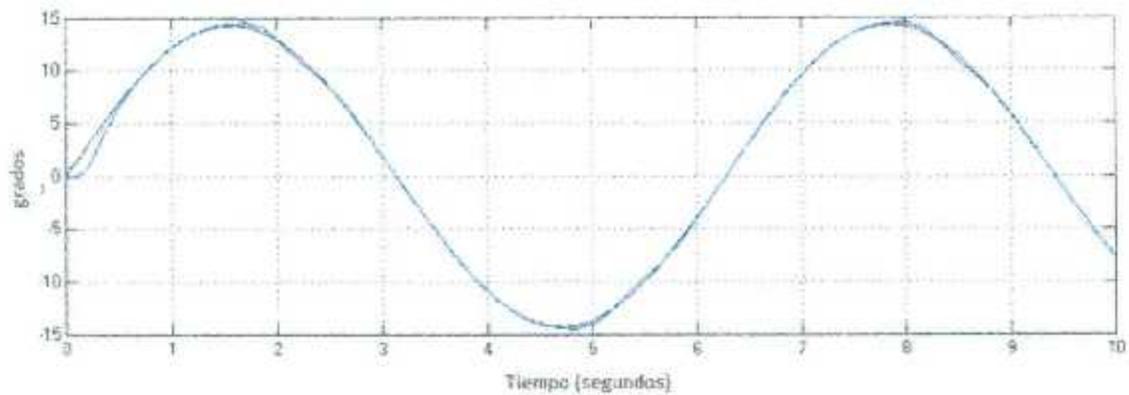
Figura 28.- Diagrama o bloques del controlador difuso T-S para seguimiento

En la Figura 29 se muestran los resultados obtenidos en simulación para el caso de seguimiento de la articulación 2. En la gráfica de posición la señal marcada en color negro representa la señal deseada y la señal marcada en color azul representa la señal obtenida. Se puede observar que en un inicio la respuesta del sistema para la articulación 2 no es tan rápida, sin embargo logra acercarse a la señal deseada a partir de que el tiempo marca 0.5 segundos.

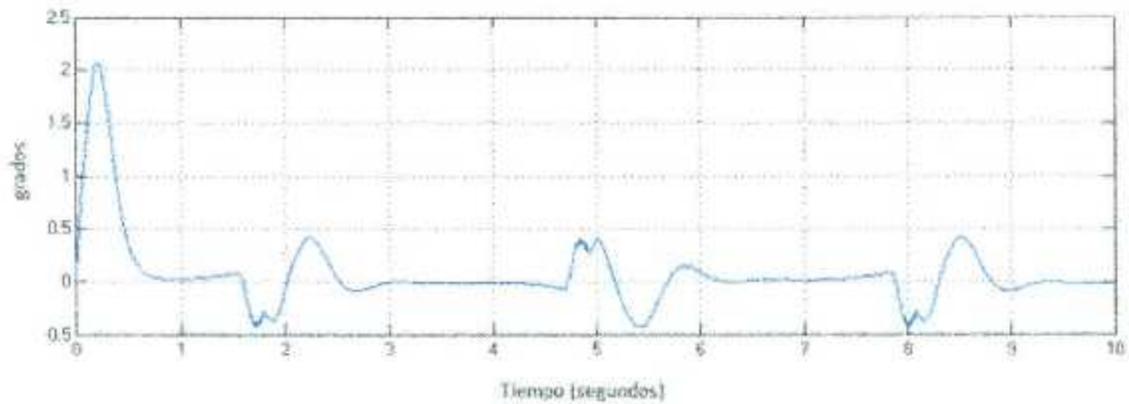
En un inicio, y como ya se mencionó, la respuesta del sistema no es tan rápida, por lo que el error de posición alcanza un pico de 2 [grados], sin embargo, el error de posición es acotado en ± 0.5 [grados] al alcanzar 1 segundo de simulación.

En la gráfica de la señal de control se aprecia que el ruido sólo se presenta en el momento en que la señal deseada cambia de dirección. A pesar de esto, el torque máximo que alcanza en simulación está muy por debajo del límite real que posco la articulación 2.

a) Posición



b) Error de posición



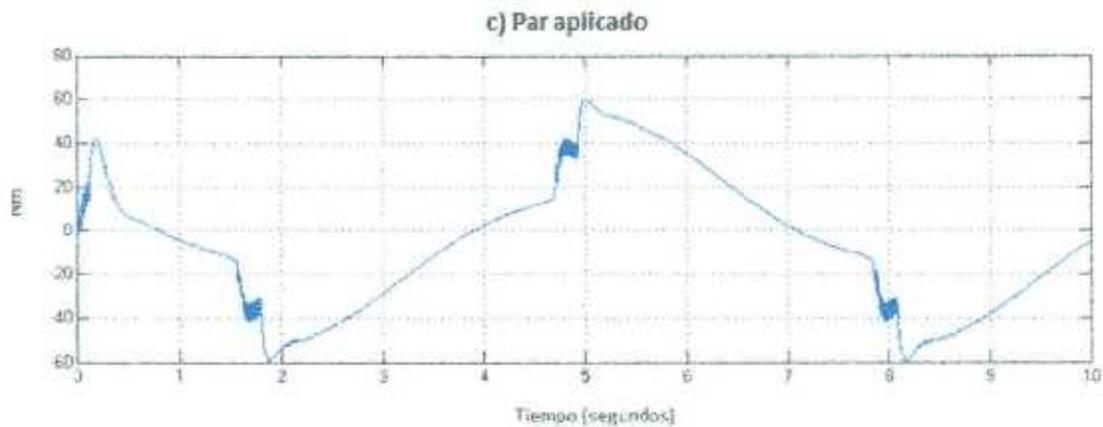
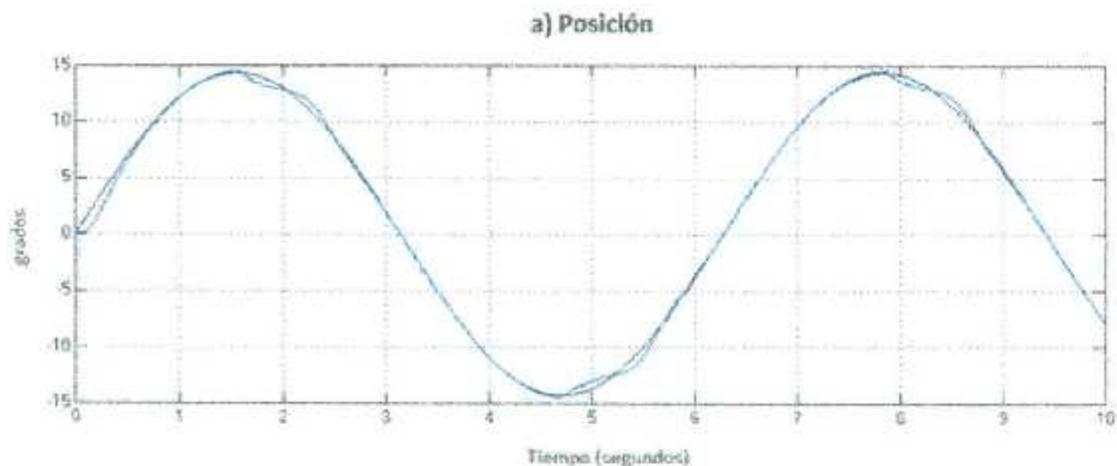


Figura 29.- Resultados seguimiento articulación 2

En la Figura 30 se muestran los resultados obtenidos en simulación para el caso de seguimiento de la articulación 4. En la gráfica de posición la señal marcada en color negro representa la señal deseada y la señal marcada en color azul representa la señal obtenida. En comparación con la articulación 2, la respuesta del sistema para la articulación 4 al inicio de la simulación es un poco más rápida, ya que solamente alcanza un pico de 1.5 [grados]. Sin embargo no es tan exacta a lo largo de la simulación como lo es la articulación 2, ya que estará acotada en ± 1 [grado]

En la gráfica de la señal de control se aprecia que el ruido sólo se presenta en el momento en que la señal deseada cambia de dirección, al igual que en la gráfica de la articulación 2. A pesar de esto, el par máximo que alcanza en simulación está muy por debajo del límite real que posee la articulación 4, ya que el punto máximo que alcanza en la gráfica es de ± 20 [N m].



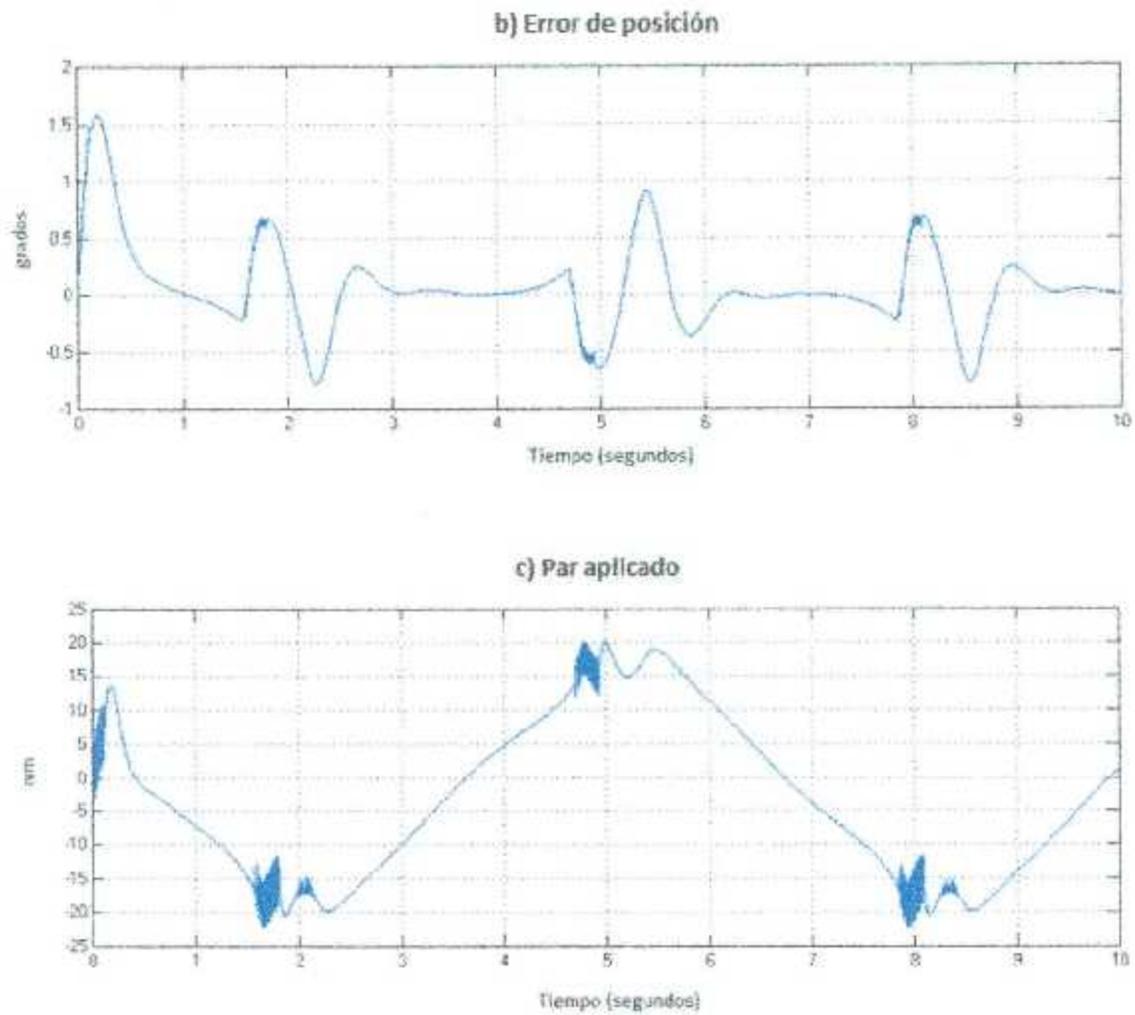


Figura 30.- Resultados seguimiento articulación 4

6.2 Experimentales control difuso T-S

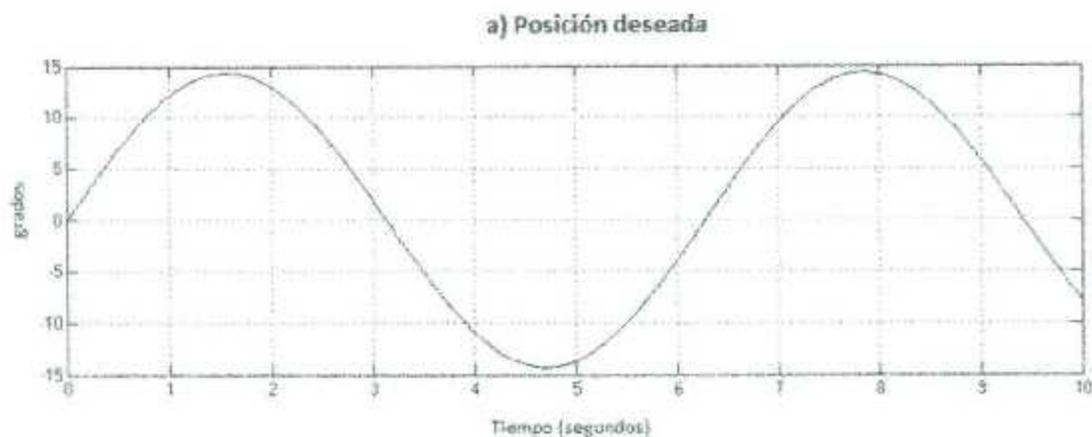
En el presente apartado se muestran los resultados obtenidos de las pruebas en tiempo real del controlador difuso T-S aplicado al robot Mitsubishi PA10. Esta es la primera parte del objetivo de esta tesis, ya que se muestran los resultados obtenidos mediante la realimentación de los encoders. Sin embargo, esta prueba es necesaria para determinar que la parte difusa funciona correctamente.

6.2.1 Seguimiento

A continuación se presentan los resultados obtenidos en tiempo real para la tarea de seguimiento en el robot Mitsubishi PA10-7. Se utilizará la misma señal de referencia que se utilizó en simulación.

En la Figura 31 se muestran los resultados obtenidos para el caso de seguimiento de la articulación 2. En la gráfica de error de posición se puede ver un sobreimpulso de 2 [grados] al inicio de la puesta en marcha, tal y como se vio en la simulación. Sin embargo, se observa también que después de que se cumple 1 segundo de este ejercicio, la señal del error queda acotada en ± 0.5 [grados].

En la gráfica del par aplicado se aprecia que el ruido presentado es aceptable, ya que no existen grandes sobre impulsos en dicha señal. Además, se puede comprobar que la señal de control no excede el límite establecido de ± 232 [N m] para la articulación 2.



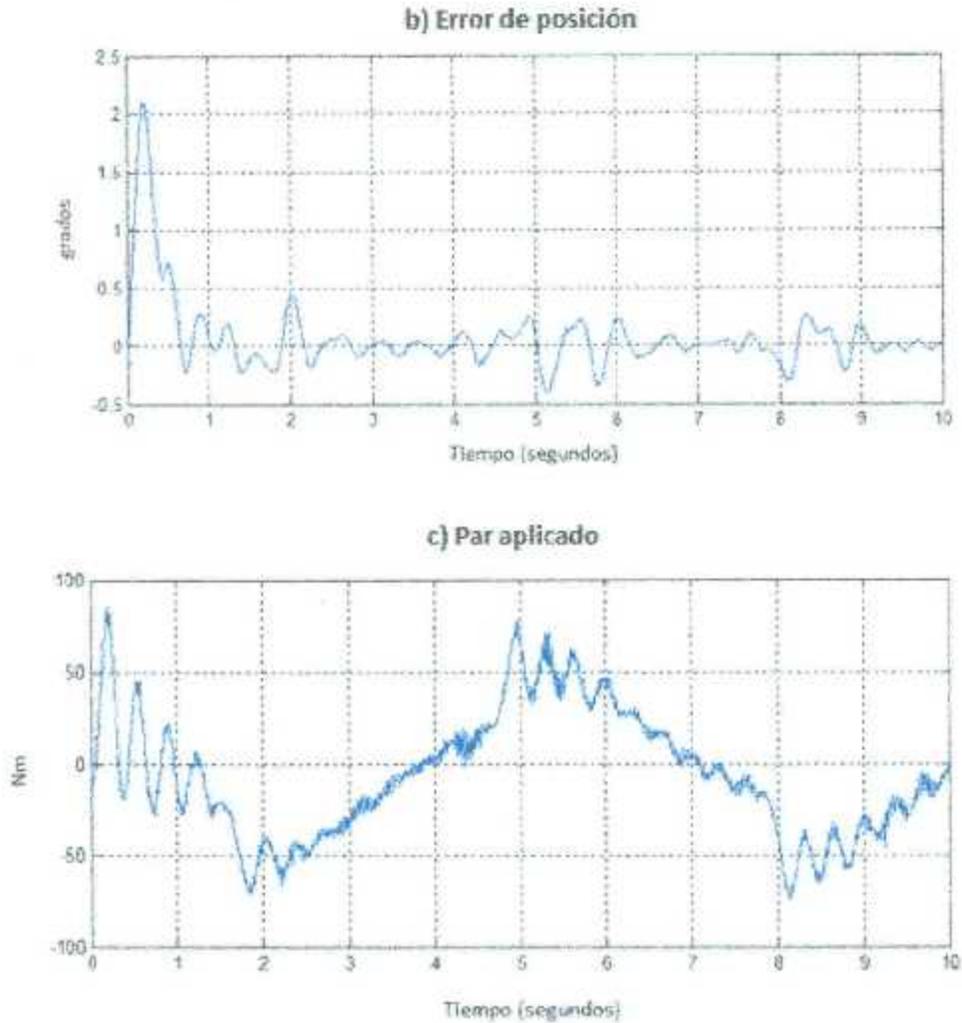
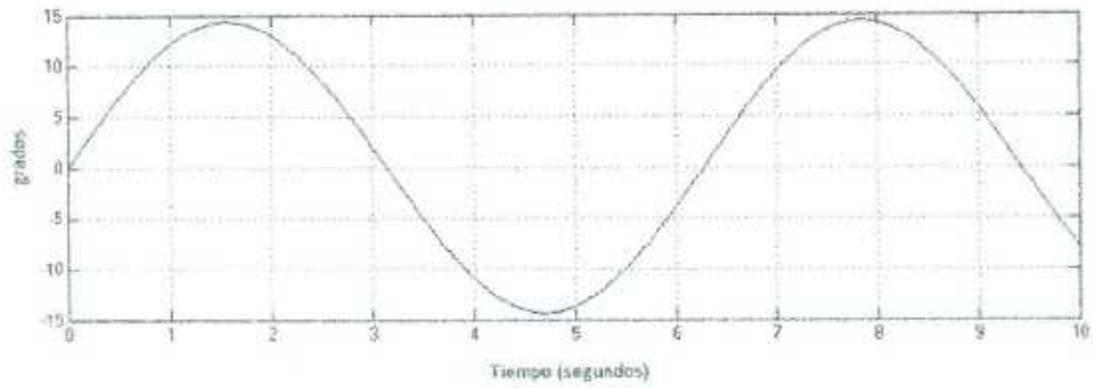


Figura 31.- Resultados seguimiento en tiempo real para la articulación 2

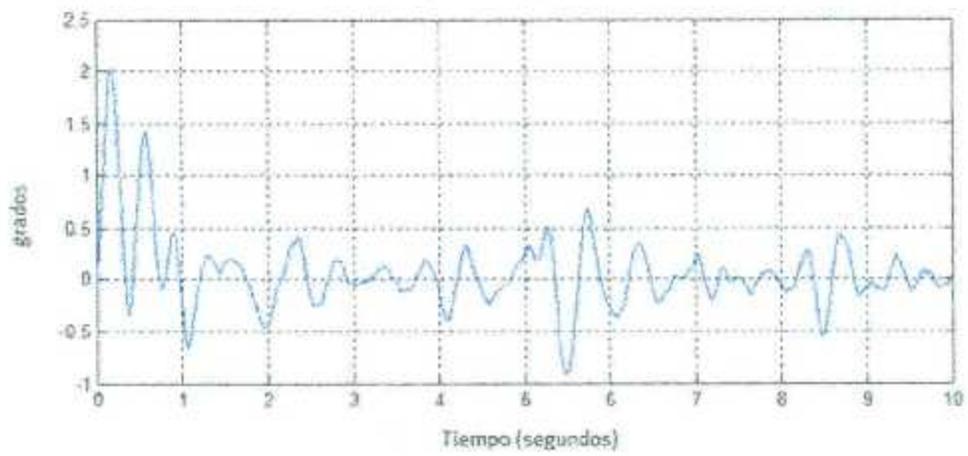
En la Figura 32 se muestran los resultados obtenidos para el caso de seguimiento de la articulación 4. En la gráfica de error de posición se puede ver un sobre impulso de 2 [grados] al inicio de la puesta en marcha, un poco más alto de lo que se esperaba después de realizar la simulación, tal y como se vio en la simulación. Sin embargo, se observa también que después de que se cumple 1 segundo de este ejercicio, la señal del error queda acotada en ± 0.5 [grados], al igual que la articulación 2.

En la gráfica del par aplicado se aprecia que el ruido presentado es aceptable, ya que no existen grandes sobre impulsos en dicha señal. Además, se puede comprobar que la señal de control no excede el límite establecido de ± 100 [N m] para la articulación 4.

a) Posición deseada



b) Error de posición



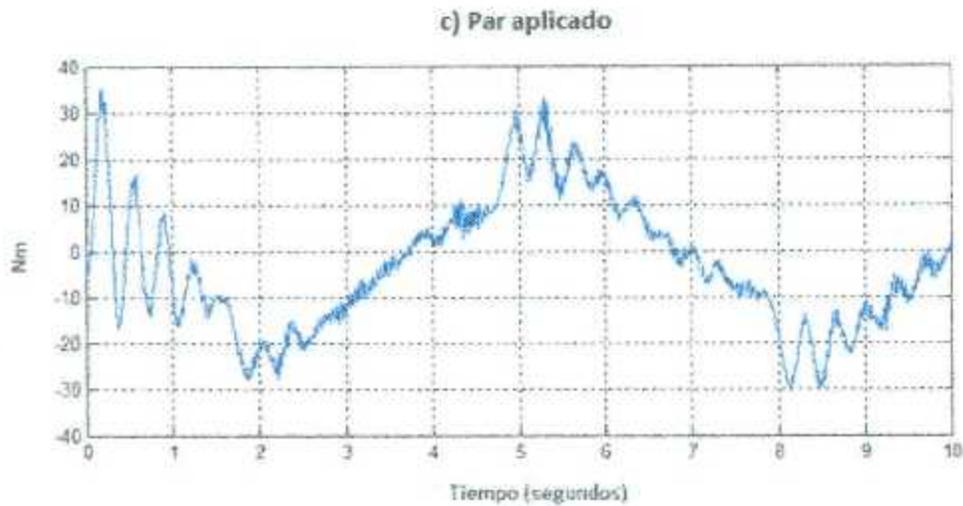


Figura 32.- Resultados seguimiento en tiempo real para la articulación 4

Dados estos resultados se puede decir que el controlador difuso T-S propuesto tiene un desempeño satisfactorio, verificando así los resultados teóricos presentados previamente.

6.3 Experimentales control servovisual

6.3.1 Seguimiento

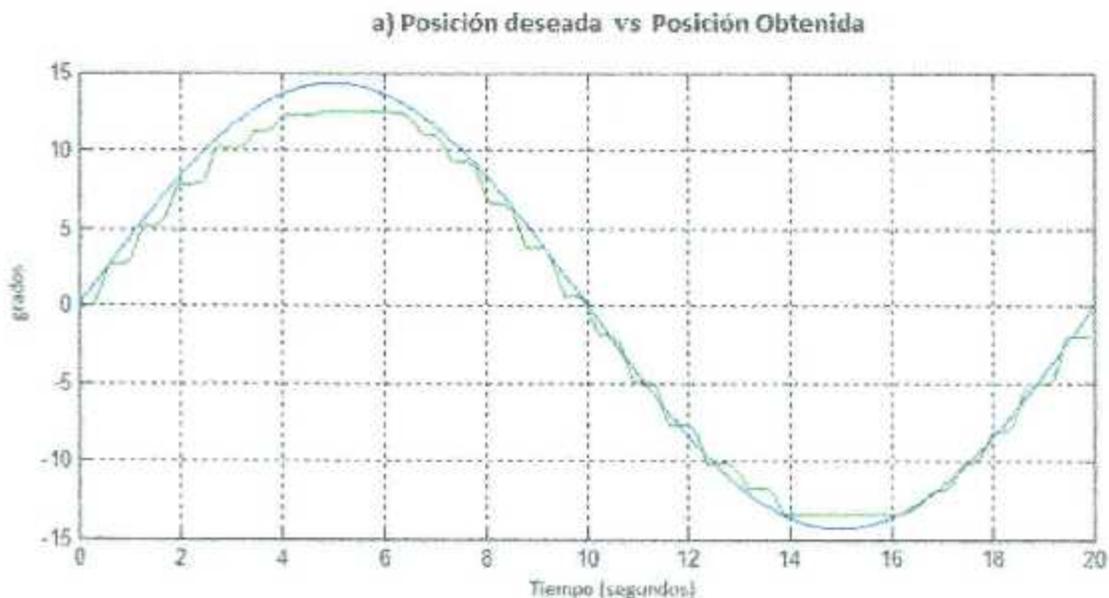
En los objetivos originales de la tesis se planteó que una vez que funcionara el control difuso T-S se sustituiría la acción de los encoders por una realimentación visual, para obtener un control servovisual difuso. Sin embargo, debido a la carencia de una cámara más rápida que pudiera igualar el periodo de muestreo del robot Mitsubishi y debido a limitaciones de tiempo, no se pudo llevar a cabo dicha implementación. Sin embargo, como una fase previa del control servovisual difuso se implementó un controlador PD de ganancias fijas con realimentación visual, el cual consume menor tiempo de cálculo que el Takagi-Sugeno por ser de ganancias fijas, funcionando aceptablemente, por lo cual es el que se reporta en este trabajo de tesis. Se deja como un trabajo futuro la implementación visual del controlador difuso Takagi-Sugeno.

En esta sección se presentan los resultados obtenidos de las pruebas en tiempo real para un controlador tipo PD (Proporcional-Diferencial) aplicado al robot Mitsubishi PA10, con realimentación visual. Se considera una señal de referencia más lenta en comparación a los ejercicios anteriores, debido a que el periodo de muestreo de la cámara es 7 veces más

lento que el del robot, de manera que se pudiera dar al programa la capacidad de adquirir los datos visuales en tiempo real y calcular la salida de control. Dicha señal de referencia está dada por:

$$[q_{d1}] = [0.25 * \text{sen}(0.314 * t)] = \left[\frac{1}{s^2 + 0.314} \right]$$

En la Figura 33 se muestran los resultados obtenidos para el caso de seguimiento de la articulación 2 con realimentación visual. En la gráfica del inciso (a) la nueva señal de referencia está identificada con el color azul y la posición obtenida está marcada con color verde. En dicha gráfica se puede observar que la señal obtenida tiene cierto ruido, el cual es provocado por el tiempo que tarda la cámara en enviar la señal y que se ve reflejado al momento de calcular la posición en la que se encuentra el robot. Sin embargo, se puede observar en la gráfica del error de posición, que la señal está acotada en ± 1.8 [grados] por lo que se puede concluir que el robot sigue la señal de referencia para la articulación 2. Hay que hacer notar que el comportamiento de la articulación 4 no resultó igualmente satisfactorio, ya presentaba un mayor error y ciertas perturbaciones bruscas, por lo que se decidió presentar únicamente los resultados para la articulación 2.



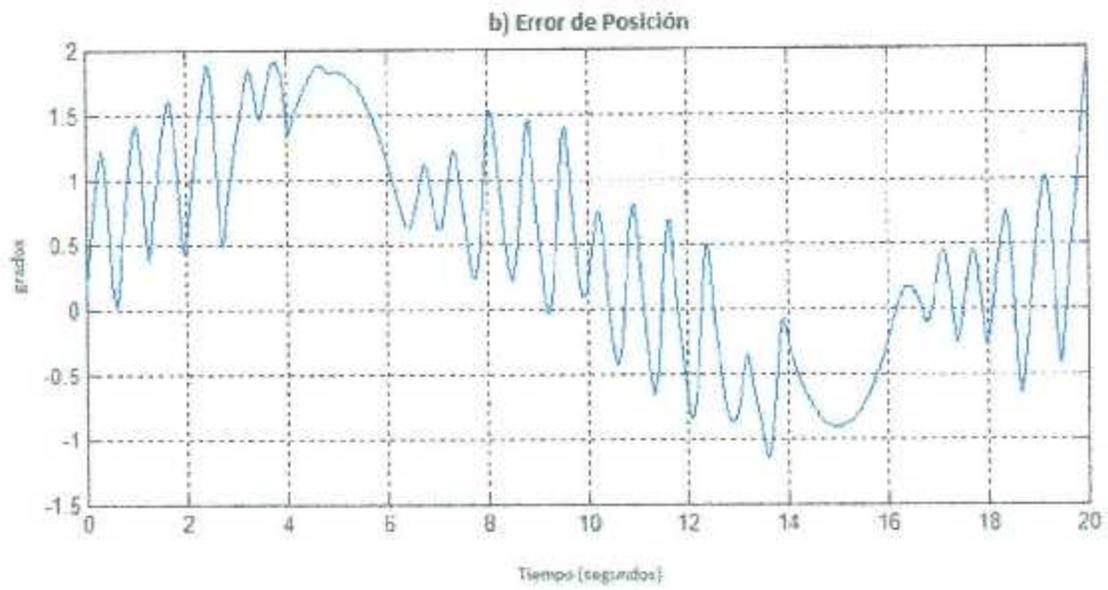


Figura 33.- Resultados seguimiento en tiempo real con realimentación visual para la articulación 2

Capítulo 7

Conclusiones y trabajo futuro

7.1 Conclusiones

En el presente trabajo de tesis se presentaron los resultados obtenidos para el problema de regulación y seguimiento del robot manipulador Mitsubishi PA10-7CF aplicando un controlador difuso Takagi-Sugeno realimentado visualmente, mediante una cámara web.

Como primera instancia de este trabajo se diseñó un controlador difuso Takagi-Sugeno realimentando su posición y su velocidad a través de los encoders que posee el robot. Este controlador fue desarrollado mediante la técnica de Series de Taylor. Para lograr diseñar este controlador, fue necesario obtener el modelo difuso T-S del robot manipulador (planta), ya que para proponer las reglas para el controlador, se utilizan los mismos parámetros difusos usados para las reglas de la planta.

Se utilizaron nueve reglas difusas para cada controlador, lo cual hace que la linealización del sistema se aproxime considerablemente a la realidad, lo que se puede corroborar al momento de la aplicación en tiempo real.

Se demostró la estabilidad del sistema en lazo cerrado mediante el uso de desigualdades del tipo Lyapunov, lo que implica que los estados del sistema en conjunto se encuentran acotados. Las ganancias, tanto del controlador para regulación, como para el de seguimiento, fueron calculadas a través del "LMI Control Toolbox de MATLAB".

Para el diseño del controlador se utilizó MATLAB/Simulink, mediante el cual se pudo corroborar que las ganancias propuestas cumplían con el objetivo de control. Una vez que se comprobó que el controlador funcionaba correctamente en simulación, se procedió a llevarlo a la práctica. Para este procedimiento fue necesario modificar pequeños parámetros relacionados con la mecánica del robot, de manera que el controlador funcionó adecuadamente.

Se puede observar que los resultados obtenidos en la parte de control difuso son satisfactorios, ya que las pruebas en tiempo real demuestran que el algoritmo propuesto es un mecanismo adecuado para el seguimiento de trayectorias aplicado a un robot manipulador.

De la misma forma, se pudo comprobar que el comportamiento del robot, limitando su funcionamiento a tan sólo 2 de las 7 articulaciones con las que cuenta, es adecuado, y

que para la simulación de éste, se puede sólo reducir el número de renglones y columnas del modelo dinámico obtenido para 7 grados de libertad.

Como segunda instancia de este trabajo, se propuso realimentar el sistema visualmente, es decir, a través de imágenes obtenidas en tiempo real. Para llevar a cabo este procedimiento fue necesario el uso de bloques especiales ("Rate Transition" mostrado en la Figura 22 y "Video Capture" mostrado en la Figura 18), ya que a pesar de que MATLAB cuenta con un toolbox para la adquisición de imágenes, éste no es capaz de trabajar en tiempo real. Se calcularon los centroides de las marcas establecidas para cada eslabón, determinando de esta forma, la posición en la que se encontraba el robot en diferentes instantes de tiempo.

Como se mencionó en la sección 6.3.1, se utilizó un controlador PD de ganancias fijas (desarrollado en [13]) para comprobar que la realimentación de la cámara funcionaba correctamente. Cambiando la información obtenida de coordenadas cartesianas a coordenadas articulares se realimentó el sistema, obteniendo resultados aceptables para la parte de control con realimentación visual, pero sólo para la articulación 2, ya que la articulación 4 presentaba un mayor error y ciertas perturbaciones bruscas, por lo que sólo se presentó el resultado de la articulación 2.

7.2 Trabajo futuro

Como trabajo futuro quedaría aplicar la realimentación visual probada con el controlador PD de ganancias fijas al control difuso Takagi-Sugeno desarrollado en esta Tesis, el cual no pudo ser aplicado en tiempo real por falta de tiempo para experimentar con la planta y a la carencia de una cámara más rápida que pudiera igualar el periodo de muestreo del Mitsubishi que es de 7 [ms], mientras que el de la cámara es de 33 [ms].

Otra propuesta para trabajo futuro sería realizar el control difuso utilizando todas las articulaciones del robot Mitsubishi, proponiendo por lo menos 3 puntos de operación para cada articulación, lo cual resultaría en un gran número de subsistemas lineales, los cuales podrían ser calculados en una computadora con mayores capacidades.

Se propone también utilizar una segunda cámara, la cual se coloque perpendicularmente a la primera cámara, con el objetivo de visualizar, no sólo un plano cartesiano, sino todo el espacio en el que se mueve el robot.

Otro de los trabajos a futuros que complementarían a este proyecto de investigación sería el desarrollo de un artículo para una revista indexada.

Apéndice

A. Parámetros del robot Mitsubishi PA10-7CE

Es importante mencionar que la planta a utilizar debe de ser manejada con los máximos cuidados, ya que sus componentes electrónicos y mecánicos son susceptibles a fallas ocasionadas por el mal manejo del mismo. Dado esto, es necesario considerar los siguientes aspectos: la masa total del robot Mitsubishi PA10-7CE es de aproximadamente 40 kg, y puede sostener en su efector final una carga máxima de hasta 10 kg, como se explica en [8], ya que de otra manera, el robot puede sufrir descomposturas en su parte mecánica.

Por otro lado en la Tabla A.1 se muestran las consignas máximas de velocidad y par que se pueden aplicar a cada una de las articulaciones del robot así como los límites de posición. La reducción de engranajes de la transmisión armónica que tienen los motores de las articulaciones es de 50. Es decir, si se le aplica al motor un par de 1 [N m], en realidad son 50 [N m] (después de la transmisión) los que se aplican a la articulación. Es de vital importancia considerar estos parámetros, ya que de no hacerlo, el robot puede sufrir daños importantes tanto en sus motores como otras partes que componen el sistema mecánico.

Articulación	Velocidad [rad/s]	Par [N*m]	Posición[°]
S1	1	232	± 165
S2	1	232	± 85
S3	2	100	± 165
E1	2	100	± 125
W1	2π	14.5	± 245
W2	2π	14.5	± 145
W3	2π	14.5	± 245

Tabla A.1. - Parámetros del robot Mitsubishi PA10

B. Programas en MATLAB

B.1 Linealización y controlabilidad

A continuación se muestra un resumen del procedimiento para obtener el modelo difuso por medio de la técnica de Takagi Sugeno para el robot Mitsubishi PA10-7CE para 2 grados de libertad.

```
clear all;clc;syms q2 q4;syms q2p q4p;syms T2 T4;

q1=0;
q3=0;
q5=0;
q5=0;
q7=0;
q1p=0;
q3p=0;
q5p=0;
q6p=0;
q7p=0;
Qp=[q2p;q4p];
TAU=[T2;T4];

M = [M1;M2;M3;M4;M5;M6;M7];
C = [C1;C2;C3;C4;C5;C6;C7];
g = [g1;q2;q3;q4;q5;q6;q7];

M_mod=[M(2,2) M(2,4);M(4,2) M(4,4)];
C_mod=[C(2,2) C(2,4);C(4,2) C(4,4)];
g_mod=[g2;q4];

QPP=inv(M_mod)*(TAU-g_mod-C_mod*Qp);
%-----Series de Taylor
syms x1 x2 x3 x4;
x1=q2;
x2=q4;
x3=q2p;
x4=q4p;

f=[q2p;q4p;QPP(1);QPP(2)];

%DERIVADAS PARCIALES DE F
```

```

%PARA f1
f1x1=diff(f(1),x1);
f1x2=diff(f(1),x2);
f1x3=diff(f(1),x3);
f1x4=diff(f(1),x4);

%PARA f2
f2x1=diff(f(2),x1);
f2x2=diff(f(2),x2);
f2x3=diff(f(2),x3);
f2x4=diff(f(2),x4);

%PARA f3
f3x1=diff(f(3),x1);
f3x2=diff(f(3),x2);
f3x3=diff(f(3),x3);
f3x4=diff(f(3),x4);

%PARA f4
f4x1=diff(f(4),x1);
f4x2=diff(f(4),x2);
f4x3=diff(f(4),x3);
f4x4=diff(f(4),x4);

%Para las entradas
%Para f1
f1T2=diff(f(1),T2);
f1T4=diff(f(1),T4);

%Para f2
f2T2=diff(f(2),T2);
f2T4=diff(f(2),T4);

%Para f3
f3T2=diff(f(3),T2);
f3T4=diff(f(3),T4);

%Para f4
f4T2=diff(f(4),T2);
f4T4=diff(f(4),T4);

%formar la matriz A
A=[f1x1 f1x2 f1x3 f1x4;
   f2x1 f2x2 f2x3 f2x4;
   f3x1 f3x2 f3x3 f3x4;
   f4x1 f4x2 f4x3 f4x4];

%formar la matriz B
B=[f1T2 f1T4;
   f2T2 f2T4;
   f3T2 f3T4;
   f4T2 f4T4];

%Evaluar numéricamente en cada punto de linealización.
%Todos se evalúan con T=0

```

```

T2=0; T4=0;

q2=0; q4=0; q2p=0; q4p=0;
A_num=inline(A);
A1=A_num(T2,T4,q2,q2p,q4,q4p)
B_num=inline(B);
B1=B_num(q4)

-----
q2=-1; q4=0; q2p=0; q4p=0;
A2=A_num(T2,T4,q2,q2p,q4,q4p)
B2=B_num(q4)

-----
q2=1; q4=0; q2p=0; q4p=0;
A3=A_num(T2,T4,q2,q2p,q4,q4p)
B3=B_num(q4)

-----
q2=0; q4=-1; q2p=0; q4p=0;
x1=0; x2=-1; x3=0; x4=0;
A4=A_num(T2,T4,q2,q2p,q4,q4p)
B4=B_num(q4)

-----
q2=-1; q4=-1; q2p=0; q4p=0;
x1=-1; x2=-1; x3=0; x4=0;
A5=A_num(T2,T4,q2,q2p,q4,q4p)
B5=B_num(q4)

-----
q2=1; q4=-1; q2p=0; q4p=0;
x1=1; x2=-1; x3=0; x4=0;
A6=A_num(T2,T4,q2,q2p,q4,q4p)
B6=B_num(q4)

-----
q2=0; q4=1; q2p=0; q4p=0;
x1=0; x2=1; x3=0; x4=0;
A7=A_num(T2,T4,q2,q2p,q4,q4p)
B7=B_num(q4)

-----
q2=-1; q4=1; q2p=0; q4p=0;
x1=-1; x2=1; x3=0; x4=0;
A8=A_num(T2,T4,q2,q2p,q4,q4p)
B8=B_num(q4)

-----
q2=1; q4=1; q2p=0; q4p=0;
x1=1; x2=1; x3=0; x4=0;
A9=A_num(T2,T4,q2,q2p,q4,q4p)
B9=B_num(q4)
%Controlabilidad

```

```
CM1=ctrb(A1,B1)
CM2=ctrb(A2,B2)
CM3=ctrb(A3,B3)
CM4=ctrb(A4,B4)
CM5=ctrb(A5,B5)
CM6=ctrb(A6,B6)
CM7=ctrb(A7,B7)
CM8=ctrb(A8,B8)
CM9=ctrb(A9,B9)
```

```
rank(CM1)
rank(CM2)
rank(CM3)
rank(CM4)
rank(CM5)
rank(CM6)
rank(CM7)
rank(CM8)
rank(CM9)
```

B.2 Cálculo de ganancias K_i y de la matriz P_C del controlador difuso T-S para regulación

Utilizando los datos proporcionados por el LMI Toolbox de MatLab como se muestran en [3], a continuación se presenta el procedimiento para obtener las ganancias de regulación.

%Código copiado del editor de LMIs Toolbox (se abre con lmedit)

```
setlmis([]);
X=lmivar(1,[4 1]);
M1=lmivar(2,[2 4]);
M2=lmivar(2,[2 4]);
M3=lmivar(2,[2 4]);
M4=lmivar(2,[2 4]);
M5=lmivar(2,[2 4]);
M6=lmivar(2,[2 4]);
M7=lmivar(2,[2 4]);
M8=lmivar(2,[2 4]);
M9=lmivar(2,[2 4]);

lmiterm([-1 1 1 X],1,-A1','s'); % LMI #1: -X*A1'-A1*X
lmiterm([-1 1 1 -M1],.5*1,B1','s'); % LMI #1: M1'*B1'
lmiterm([-1 1 1 M1],.5*B1,1,'s'); % LMI #1: B1*M1
lmiterm([-2 1 1 X],1,-A2','s'); % LMI #2: -X*A2'-A2*X
lmiterm([-2 1 1 -M2],.5*1,B2','s'); % LMI #2: M2'*B2'
lmiterm([-2 1 1 M2],.5*B2,1,'s'); % LMI #2: B2*M2
lmiterm([-3 1 1 X],1,-A3','s'); % LMI #3: -X*A3'-A3*X
lmiterm([-3 1 1 -M3],.5*1,B3','s'); % LMI #3: M3'*B3'
lmiterm([-3 1 1 M3],.5*B3,1,'s'); % LMI #3: B3*M3
lmiterm([-4 1 1 X],1,-A4','s'); % LMI #4: -X*A4'-A4*X
```

```

lmiterm([-4 1 1 -M4],.5*1,B4','s');
lmiterm([-4 1 1 M4],.5*B4,1,'s');
lmiterm([-5 1 1 X],1,-A5','s');
lmiterm([-5 1 1 -M5],.5*1,B5','s');
lmiterm([-5 1 1 M5],.5*B5,1,'s');
lmiterm([-6 1 1 X],1,-A6','s');
lmiterm([-6 1 1 -M6],.5*1,B6','s');
lmiterm([-6 1 1 M6],.5*B6,1,'s');
lmiterm([-7 1 1 X],1,-A7','s');
lmiterm([-7 1 1 -M7],.5*1,B7','s');
lmiterm([-7 1 1 M7],.5*B7,1,'s');
lmiterm([-8 1 1 X],1,-A8','s');
lmiterm([-8 1 1 -M8],.5*1,B8','s');
lmiterm([-8 1 1 M8],.5*B8,1,'s');
lmiterm([-9 1 1 X],1,-A9','s');
lmiterm([-9 1 1 -M9],.5*1,B9','s');
lmiterm([-9 1 1 M9],.5*B9,1,'s');
lmiterm([-10 1 1 X],1,-A1','s');
lmiterm([-10 1 1 X],1,-A2','s');
lmiterm([-10 1 1 -M2],.5*1,B1','s');
lmiterm([-10 1 1 M2],.5*B1,1,'s');
lmiterm([-10 1 1 -M1],.5*1,B2','s');
lmiterm([-10 1 1 M1],.5*B2,1,'s');
lmiterm([-11 1 1 X],1,-A1','s');
lmiterm([-11 1 1 X],1,-A3','s');
lmiterm([-11 1 1 -M3],.5*1,B1','s');
lmiterm([-11 1 1 M3],.5*B1,1,'s');
lmiterm([-11 1 1 -M1],.5*1,B3','s');
lmiterm([-11 1 1 M1],.5*B3,1,'s');
lmiterm([-12 1 1 X],1,-A1','s');
lmiterm([-12 1 1 X],1,-A4','s');
lmiterm([-12 1 1 -M4],.5*1,B1','s');
lmiterm([-12 1 1 M4],.5*B1,1,'s');
lmiterm([-12 1 1 -M1],.5*1,B4','s');
lmiterm([-12 1 1 M1],.5*B4,1,'s');
lmiterm([-13 1 1 X],1,-A1','s');
lmiterm([-13 1 1 X],1,-A5','s');
lmiterm([-13 1 1 -M5],.5*1,B1','s');
lmiterm([-13 1 1 M5],.5*B1,1,'s');
lmiterm([-13 1 1 -M1],.5*1,B5','s');
lmiterm([-13 1 1 M1],.5*B5,1,'s');
lmiterm([-14 1 1 X],1,-A1','s');
lmiterm([-14 1 1 X],1,-A6','s');
lmiterm([-14 1 1 -M6],.5*1,B1','s');
lmiterm([-14 1 1 M6],.5*B1,1,'s');
lmiterm([-14 1 1 -M1],.5*1,B6','s');
lmiterm([-14 1 1 M1],.5*B6,1,'s');
lmiterm([-15 1 1 X],1,-A1','s');
lmiterm([-15 1 1 X],1,-A7','s');
lmiterm([-15 1 1 -M7],.5*1,B1','s');
lmiterm([-15 1 1 M7],.5*B1,1,'s');
lmiterm([-15 1 1 -M1],.5*1,B7','s');
lmiterm([-15 1 1 M1],.5*B7,1,'s');
lmiterm([-16 1 1 X],1,-A1','s');
lmiterm([-16 1 1 X],1,-A8','s');
lmiterm([-16 1 1 -M8],.5*1,B1','s');
lmiterm([-16 1 1 M8],.5*B1,1,'s');
% LMI #4: M4'*B4'
% LMI #4: B4*M4
% LMI #5: -X*A5'-A5*X
% LMI #5: M5'*B5'
% LMI #5: B5*M5
% LMI #6: -X*A6'-A6*X
% LMI #6: M6'*B6'
% LMI #6: B6*M6
% LMI #7: -X*A7'-A7*X
% LMI #7: M7'*B7'
% LMI #7: B7*M7
% LMI #8: -X*A8'-A8*X
% LMI #8: M8'*B8'
% LMI #8: B8*M8
% LMI #9: -X*A9'-A9*X
% LMI #9: M9'*B9'
% LMI #9: B9*M9
% LMI #10: -X*A1'-A1*X
% LMI #10: -X*A2'-A2*X
% LMI #10: M2'*B1'
% LMI #10: B1*M2
% LMI #10: M1'*B2'
% LMI #10: B2*M1
% LMI #11: -X*A1'-A1*X
% LMI #11: -X*A3'-A3*X
% LMI #11: M3'*B1'
% LMI #11: B1*M3
% LMI #11: M1'*B3'
% LMI #11: B3*M1
% LMI #12: -X*A1'-A1*X
% LMI #12: -X*A4'-A4*X
% LMI #12: M4'*B1'
% LMI #12: B1*M4
% LMI #12: M1'*B4'
% LMI #12: B4*M1
% LMI #13: -X*A1'-A1*X
% LMI #13: -X*A5'-A5*X
% LMI #13: M5'*B1'
% LMI #13: B1*M5
% LMI #13: M1'*B5'
% LMI #13: B5*M1
% LMI #14: -X*A1'-A1*X
% LMI #14: -X*A6'-A6*X
% LMI #14: M6'*B1'
% LMI #14: B1*M6
% LMI #14: M1'*B6'
% LMI #14: B6*M1
% LMI #15: -X*A1'-A1*X
% LMI #15: -X*A7'-A7*X
% LMI #15: M7'*B1'
% LMI #15: B1*M7
% LMI #15: M1'*B7'
% LMI #15: B7*M1
% LMI #16: -X*A1'-A1*X
% LMI #16: -X*A8'-A8*X
% LMI #16: M8'*B1'
% LMI #16: B1*M8

```

```

lmiterm([-16 1 1 -M1],.5*1,B8','s');
lmiterm([-16 1 1 M1],.5*B8,1,'s');
lmiterm([-17 1 1 X],1,-A1','s');
lmiterm([-17 1 1 X],1,-A9','s');
lmiterm([-17 1 1 -M9],.5*1,B1','s');
lmiterm([-17 1 1 M9],.5*B1,1,'s');
lmiterm([-17 1 1 -M1],.5*1,B9','s');
lmiterm([-17 1 1 M1],.5*B9,1,'s');
lmiterm([-18 1 1 X],1,-A2','s');
lmiterm([-18 1 1 X],1,-A3','s');
lmiterm([-18 1 1 -M3],.5*1,B2','s');
lmiterm([-18 1 1 M3],.5*B2,1,'s');
lmiterm([-18 1 1 -M2],.5*1,B3','s');
lmiterm([-18 1 1 M2],.5*B3,1,'s');
lmiterm([-19 1 1 X],1,-A2','s');
lmiterm([-19 1 1 X],1,-A4','s');
lmiterm([-19 1 1 -M4],.5*1,B2','s');
lmiterm([-19 1 1 M4],.5*B2,1,'s');
lmiterm([-19 1 1 -M2],.5*1,B4','s');
lmiterm([-19 1 1 M2],.5*B4,1,'s');
lmiterm([-20 1 1 X],1,-A5','s');
lmiterm([-20 1 1 -M5],.5*1,B2','s');
lmiterm([-20 1 1 M5],.5*B2,1,'s');
lmiterm([-20 1 1 -M2],.5*1,B5','s');
lmiterm([-20 1 1 M2],.5*B5,1,'s');
lmiterm([-21 1 1 X],1,-A2','s');
lmiterm([-21 1 1 X],1,-A6','s');
lmiterm([-21 1 1 -M6],.5*1,B2','s');
lmiterm([-21 1 1 M6],.5*B2,1,'s');
lmiterm([-21 1 1 -M2],.5*1,B6','s');
lmiterm([-21 1 1 M2],.5*B6,1,'s');
lmiterm([-22 1 1 X],1,-A2','s');
lmiterm([-22 1 1 X],1,-A7','s');
lmiterm([-22 1 1 -M7],.5*1,B2','s');
lmiterm([-22 1 1 M7],.5*B2,1,'s');
lmiterm([-22 1 1 -M2],.5*1,B7','s');
lmiterm([-22 1 1 M2],.5*B7,1,'s');
lmiterm([-23 1 1 X],1,-A2','s');
lmiterm([-23 1 1 X],1,-A9','s');
lmiterm([-23 1 1 -M8],.5*1,B2','s');
lmiterm([-23 1 1 M8],.5*B2,1,'s');
lmiterm([-23 1 1 -M2],.5*1,B8','s');
lmiterm([-23 1 1 M2],.5*B8,1,'s');
lmiterm([-24 1 1 X],1,-A2','s');
lmiterm([-24 1 1 X],1,-A9','s');
lmiterm([-24 1 1 -M9],.5*1,B2','s');
lmiterm([-24 1 1 M9],.5*B2,1,'s');
lmiterm([-24 1 1 -M2],.5*1,B9','s');
lmiterm([-24 1 1 M2],.5*B9,1,'s');
lmiterm([-25 1 1 X],1,-A3','s');
lmiterm([-25 1 1 X],1,-A4','s');
lmiterm([-25 1 1 -M4],.5*1,B3','s');
lmiterm([-25 1 1 M4],.5*B3,1,'s');
lmiterm([-25 1 1 -M2],.5*1,B4','s');
lmiterm([-25 1 1 M3],.5*B4,1,'s');
lmiterm([-26 1 1 X],1,-A3','s');
lmiterm([-26 1 1 X],1,-A5','s');
% LMI #16: M1'*B8'
% LMI #16: B8*M1
% LMI #17: -X*A1'-A1*X
% LMI #17: -X*A9'-A9*X
% LMI #17: M9'*B1'
% LMI #17: B1*M9
% LMI #17: M1'*B9'
% LMI #17: B9*M1
% LMI #18: -X*A2'-A2*X
% LMI #18: -X*A3'-A3*X
% LMI #18: M3'*B2'
% LMI #18: B2*M3
% LMI #18: M2'*B3'
% LMI #18: B3*M2
% LMI #19: -X*A2'-A2*X
% LMI #19: -X*A4'-A4*X
% LMI #19: M4'*B2'
% LMI #19: B2*M4
% LMI #19: M2'*B4'
% LMI #19: B4*M2
% LMI #20: -X*A5'-A5*X
% LMI #20: M5'*B2'
% LMI #20: B2*M5
% LMI #20: M2'*B5'
% LMI #20: B5*M2
% LMI #21: -X*A2'-A2*X
% LMI #21: -X*A6'-A6*X
% LMI #21: M6'*B2'
% LMI #21: B2*M6
% LMI #21: M2'*B6'
% LMI #21: B6*M2
% LMI #22: -X*A2'-A2*X
% LMI #22: -X*A7'-A7*X
% LMI #22: M7'*B2'
% LMI #22: B2*M7
% LMI #22: M2'*B7'
% LMI #22: B7*M2
% LMI #23: -X*A2'-A2*X
% LMI #23: -X*A8'-A8*X
% LMI #23: M8'*B2'
% LMI #23: B2*M8
% LMI #23: M2'*B8'
% LMI #23: B8*M2
% LMI #24: -X*A2'-A2*X
% LMI #24: -X*A9'-A9*X
% LMI #24: M9'*B2'
% LMI #24: B2*M9
% LMI #24: M2'*B9'
% LMI #24: B9*M2
% LMI #25: -X*A3'-A3*X
% LMI #25: -X*A4'-A4*X
% LMI #25: M4'*B3'
% LMI #25: B3*M4
% LMI #25: M2'*B4'
% LMI #25: B4*M3
% LMI #26: -X*A3'-A3*X
% LMI #26: -X*A5'-A5*X

```

```

lmiterm([-26 1 1 -M5],.5*1,B3','s');
lmiterm([-26 1 1 M5],.5*B3,1,'s');
lmiterm([-26 1 1 -M2],.5*1,B5','s');
lmiterm([-26 1 1 M3],.5*B5,1,'s');
lmiterm([-27 1 1 X],1,-A3','s');
lmiterm([-27 1 1 X],1,-A6','s');
lmiterm([-27 1 1 -M6],.5*1,B3','s');
lmiterm([-27 1 1 M6],.5*B3,1,'s');
lmiterm([-27 1 1 -M2],.5*1,B6','s');
lmiterm([-27 1 1 M3],.5*B6,1,'s');
lmiterm([-28 1 1 X],1,-A3','s');
lmiterm([-28 1 1 X],1,-A7','s');
lmiterm([-28 1 1 -M7],.5*1,B3','s');
lmiterm([-28 1 1 M7],.5*B3,1,'s');
lmiterm([-28 1 1 -M2],.5*1,B7','s');
lmiterm([-28 1 1 M3],.5*B7,1,'s');
lmiterm([-29 1 1 X],1,-A3','s');
lmiterm([-29 1 1 X],1,-A8','s');
lmiterm([-29 1 1 -M6],.5*1,B3','s');
lmiterm([-29 1 1 M8],.5*B3,1,'s');
lmiterm([-29 1 1 -M2],.5*1,B8','s');
lmiterm([-29 1 1 M3],.5*B8,1,'s');
lmiterm([-30 1 1 X],1,-A3','s');
lmiterm([-30 1 1 X],1,-A9','s');
lmiterm([-30 1 1 -M9],.5*1,B3','s');
lmiterm([-30 1 1 M9],.5*B3,1,'s');
lmiterm([-30 1 1 -M2],.5*1,B9','s');
lmiterm([-30 1 1 M3],.5*B9,1,'s');
lmiterm([-31 1 1 X],1,-A4','s');
lmiterm([-31 1 1 X],1,-A5','s');
lmiterm([-31 1 1 -M5],.5*1,B4','s');
lmiterm([-31 1 1 M5],.5*B4,1,'s');
lmiterm([-31 1 1 -M4],.5*1,B5','s');
lmiterm([-31 1 1 M4],.5*B5,1,'s');
lmiterm([-32 1 1 X],1,-A4','s');
lmiterm([-32 1 1 X],1,-A6','s');
lmiterm([-32 1 1 -M6],.5*1,B4','s');
lmiterm([-32 1 1 M6],.5*B4,1,'s');
lmiterm([-32 1 1 -M4],.5*1,B6','s');
lmiterm([-32 1 1 M4],.5*B6,1,'s');
lmiterm([-33 1 1 X],1,-A4','s');
lmiterm([-33 1 1 X],1,-A7','s');
lmiterm([-33 1 1 -M7],.5*1,B4','s');
lmiterm([-33 1 1 M7],.5*B4,1,'s');
lmiterm([-33 1 1 -M4],.5*1,B7','s');
lmiterm([-33 1 1 M4],.5*B7,1,'s');
lmiterm([-34 1 1 X],1,-A4','s');
lmiterm([-34 1 1 X],1,-A8','s');
lmiterm([-34 1 1 -M8],.5*1,B4','s');
lmiterm([-34 1 1 M8],.5*B4,1,'s');
lmiterm([-34 1 1 -M4],.5*1,B8','s');
lmiterm([-34 1 1 M4],.5*B8,1,'s');
lmiterm([-35 1 1 X],1,-A4','s');
lmiterm([-35 1 1 X],1,-A9','s');
lmiterm([-35 1 1 -M9],.5*1,B4','s');
lmiterm([-35 1 1 M9],.5*B4,1,'s');
% LMI #26: M5'*B3'
% LMI #26: B3*M5
% LMI #26: M2'*B5'
% LMI #26: B5*M3
% LMI #27: -X*A3'-A3*X
% LMI #27: -X*A6'-A6*X
% LMI #27: M6'*B3'
% LMI #27: B3*M6
% LMI #27: M2'*B6'
% LMI #27: B6*M3
% LMI #28: -X*A3'-A3*X
% LMI #28: -X*A7'-A7*X
% LMI #28: M7'*B3'
% LMI #28: B3*M7
% LMI #28: M2'*B7'
% LMI #28: B7*M3
% LMI #29: -X*A3'-A3*X
% LMI #29: -X*A8'-A8*X
% LMI #29: M8'*B3'
% LMI #29: B3*M8
% LMI #29: M2'*B8'
% LMI #29: B8*M3
% LMI #30: -X*A3'-A3*X
% LMI #30: -X*A9'-A9*X
% LMI #30: M9'*B3'
% LMI #30: B3*M9
% LMI #30: M2'*B9'
% LMI #30: B9*M3
% LMI #31: -X*A4'-A4*X
% LMI #31: -X*A5'-A5*X
% LMI #31: M5'*B4'
% LMI #31: B4*M5
% LMI #31: M4'*B5'
% LMI #31: B5*M4
% LMI #32: -X*A4'-A4*X
% LMI #32: -X*A6'-A6*X
% LMI #32: M6'*B4'
% LMI #32: B4*M6
% LMI #32: M4'*B6'
% LMI #32: B6*M4
% LMI #33: -X*A4'-A4*X
% LMI #33: -X*A7'-A7*X
% LMI #33: M7'*B4'
% LMI #33: B4*M7
% LMI #33: M4'*B7'
% LMI #33: B7*M4
% LMI #34: -X*A4'-A4*X
% LMI #34: -X*A8'-A8*X
% LMI #34: M8'*B4'
% LMI #34: B4*M8
% LMI #34: M4'*B8'
% LMI #34: B8*M4
% LMI #35: -X*A4'-A4*X
% LMI #35: -X*A9'-A9*X
% LMI #35: M9'*B4'
% LMI #35: B4*M9

```

```

lmiterm([-35 1 1 -M4],.5*1,B9','s');
lmiterm([-35 1 1 M4],.5*B9,1,'s');
lmiterm([-36 1 1 X],1,-A5','s');
lmiterm([-36 1 1 X],1,-A6','s');
lmiterm([-36 1 1 -M6],.5*1,B5','s');
lmiterm([-36 1 1 M6],.5*B5,1,'s');
lmiterm([-36 1 1 -M5],.5*1,B6','s');
lmiterm([-36 1 1 M5],.5*B6,1,'s');
lmiterm([-37 1 1 X],1,-A5','s');
lmiterm([-37 1 1 X],1,-A7','s');
lmiterm([-37 1 1 -M7],.5*1,B5','s');
lmiterm([-37 1 1 M7],.5*B5,1,'s');
lmiterm([-37 1 1 -M5],.5*1,B7','s');
lmiterm([-37 1 1 M5],.5*B7,1,'s');
lmiterm([-38 1 1 X],1,-A5','s');
lmiterm([-38 1 1 X],1,-A8','s');
lmiterm([-38 1 1 -M8],.5*1,B5','s');
lmiterm([-38 1 1 M8],.5*B5,1,'s');
lmiterm([-38 1 1 -M8],.5*1,B9','s');
lmiterm([-38 1 1 M8],.5*B8,1,'s');
lmiterm([-39 1 1 X],1,-A5','s');
lmiterm([-39 1 1 X],1,-A9','s');
lmiterm([-39 1 1 -M9],.5*1,B5','s');
lmiterm([-39 1 1 M9],.5*B5,1,'s');
lmiterm([-39 1 1 -M5],.5*1,B9','s');
lmiterm([-39 1 1 M5],.5*B9,1,'s');
lmiterm([-40 1 1 X],1,-A6','s');
lmiterm([-40 1 1 X],1,-A7','s');
lmiterm([-40 1 1 -M7],.5*1,B6','s');
lmiterm([-40 1 1 M7],.5*B6,1,'s');
lmiterm([-40 1 1 -M6],.5*1,B7','s');
lmiterm([-40 1 1 M6],.5*B7,1,'s');
lmiterm([-41 1 1 X],1,-A6','s');
lmiterm([-41 1 1 X],1,-A8','s');
lmiterm([-41 1 1 -M8],.5*1,B6','s');
lmiterm([-41 1 1 M8],.5*B6,1,'s');
lmiterm([-41 1 1 -M6],.5*1,B8','s');
lmiterm([-41 1 1 M6],.5*B8,1,'s');
lmiterm([-42 1 1 X],1,-A6','s');
lmiterm([-42 1 1 X],1,-A9','s');
lmiterm([-42 1 1 -M9],.5*1,B6','s');
lmiterm([-42 1 1 M9],.5*B6,1,'s');
lmiterm([-42 1 1 -M6],.5*1,B9','s');
lmiterm([-42 1 1 M6],.5*B9,1,'s');
lmiterm([-43 1 1 X],1,-A7','s');
lmiterm([-43 1 1 X],1,-A8','s');
lmiterm([-43 1 1 -M8],.5*1,B7','s');
lmiterm([-43 1 1 M8],.5*B7,1,'s');
lmiterm([-43 1 1 -M7],.5*1,B8','s');
lmiterm([-43 1 1 M7],.5*B8,1,'s');
lmiterm([-44 1 1 X],1,-A7','s');
lmiterm([-44 1 1 X],1,-A9','s');
lmiterm([-44 1 1 -M9],.5*1,B7','s');
lmiterm([-44 1 1 M9],.5*B7,1,'s');
lmiterm([-44 1 1 -M7],.5*1,B9','s');
lmiterm([-44 1 1 M7],.5*B9,1,'s');
lmiterm([-45 1 1 X],1,-A8','s');
% LMI #35: M4**B9'
% LMI #35: B9*M4
% LMI #36: -X*A5'-A5*X
% LMI #36: -X*A6'-A6*X
% LMI #36: M5**B5'
% LMI #36: B5*M6
% LMI #36: M5**B6'
% LMI #36: B6*M5
% LMI #37: -X*A5'-A5*X
% LMI #37: -X*A7'-A7*X
% LMI #37: M7**B5'
% LMI #37: B5*M7
% LMI #37: M5**B7'
% LMI #37: B7*M5
% LMI #38: -X*A5'-A5*X
% LMI #38: -X*A8'-A8*X
% LMI #38: M8**B5'
% LMI #38: B5*M8
% LMI #38: M5**B9'
% LMI #38: B9*M5
% LMI #39: -X*A5'-A5*X
% LMI #39: -X*A9'-A9*X
% LMI #39: M9**B5'
% LMI #39: B5*M9
% LMI #39: M5**B9'
% LMI #39: B9*M5
% LMI #40: -X*A6'-A6*X
% LMI #40: -X*A7'-A7*X
% LMI #40: M7**B6'
% LMI #40: B6*M7
% LMI #40: M6**B7'
% LMI #40: B7*M6
% LMI #41: -X*A6'-A6*X
% LMI #41: -X*A8'-A8*X
% LMI #41: M8**B6'
% LMI #41: B6*M8
% LMI #41: M6**B8'
% LMI #41: B8*M6
% LMI #42: -X*A6'-A6*X
% LMI #42: -X*A9'-A9*X
% LMI #42: M9**B6'
% LMI #42: B6*M9
% LMI #42: M6**B9'
% LMI #42: B9*M6
% LMI #43: -X*A7'-A7*X
% LMI #43: -X*A8'-A8*X
% LMI #43: M8**B7'
% LMI #43: B7*M8
% LMI #43: M7**B8'
% LMI #43: B8*M7
% LMI #44: -X*A7'-A7*X
% LMI #44: -X*A9'-A9*X
% LMI #44: M9**B7'
% LMI #44: B7*M9
% LMI #44: M7**B9'
% LMI #44: B9*M7
% LMI #45: -X*A8'-A8*X

```

```

lmiterm([-45 1 1 X],1,-A9','s'); % LMI #45: -X*A9'-A9*X
lmiterm([-45 1 1 -M9],.5*1,B8','s'); % LMI #45: M9'*B8'
lmiterm([-45 1 1 M9],.5*B8,1,'s'); % LMI #45: B8*M9
lmiterm([-45 1 1 -M8],.5*1,B9','s'); % LMI #45: M8'*B9'
lmiterm([-45 1 1 M8],.5*B9,1,'s'); % LMI #45: B9*M8
lmiterm([-46 1 1 X],1,1); % LMI #46: X

Mitsu=getimis; % Mitsu es el nombre dado al sistema de desigualdades
%% Resolver la LMI, la respuesta es buena si tmin<=0
[tmin,Xfeas]=feasp(Mitsu)% %s es el que calcula la viabilidad de la solución

Xfinal = dec2mat(Mitsu, Xfeas,X) %dec2mat permite extraer las variables
matriciales; por ejemplo extraer X de Xfeas
polosX=eig(Xfinal)'
M1final= dec2mat(Mitsu,Xfeas,M1) %extraer M1 de Xfeas
M2final= dec2mat(Mitsu,Xfeas,M2) %extraer M2 de Xfeas
M3final= dec2mat(Mitsu,Xfeas,M3) %extraer M3 de Xfeas
M4final= dec2mat(Mitsu,Xfeas,M4) %extraer M4 de Xfeas
M5final= dec2mat(Mitsu,Xfeas,M5) %extraer M5 de Xfeas
M6final= dec2mat(Mitsu,Xfeas,M6) %extraer M6 de Xfeas
M7final= dec2mat(Mitsu,Xfeas,M7) %extraer M7 de Xfeas
M8final= dec2mat(Mitsu,Xfeas,M8) %extraer M8 de Xfeas
M9final= dec2mat(Mitsu,Xfeas,M9) %extraer M9 de Xfeas
%% Ganancias y Matriz P
P=inv(Xfinal)
polosp=eig(P)'
K1=M1final*inv(Xfinal) %ganancias finales del controlador 1
K2=M2final*inv(Xfinal) %ganancias finales del controlador 2
K3=M3final*inv(Xfinal) %ganancias finales del controlador 3
K4=M4final*inv(Xfinal) %ganancias finales del controlador 4
K5=M5final*inv(Xfinal) %ganancias finales del controlador 5
K6=M6final*inv(Xfinal) %ganancias finales del controlador 6
K7=M7final*inv(Xfinal) %ganancias finales del controlador 7
K8=M8final*inv(Xfinal) %ganancias finales del controlador 8
K9=M9final*inv(Xfinal) %ganancias finales del controlador 9

```

B.3 Cálculo de ganancias K_i y de la matriz P_C del controlador difuso T-S para seguimiento

Utilizando las matrices A y B para seguimiento (modelo del servocompensador), obtenidas en el capítulo 4.4.1 y seleccionados los eigenvalores de $(A_i + B_i K_i - 0)$ mencionados en el capítulo 4.4.3, las ganancias K_i y la matriz P_C son obtenidas de la siguiente manera:

%Código para obtener las ganancias K_i de seguimiento

```

polos1 =
-7.1712 +24.2145i
-7.1712 -24.2145i

```

```
-5.3840 +24.9725i
-5.3840 -24.9725i
-4.4026 + 2.7388i
-4.4026 - 2.7388i
-4.3820 + 2.6625i
-4.3820 - 2.6625i
```

polos2 =

```
-5.6823 +25.8573i
-5.6823 -25.8573i
-7.1716 +24.2143i
-7.1716 -24.2143i
-4.4044 + 2.7404i
-4.4044 - 2.7404i
-4.0815 + 2.8042i
-4.0815 - 2.8042i
```

polos3 =

```
-5.6823 +25.8573i
-5.6823 -25.8573i
-7.1716 +24.2143i
-7.1716 -24.2143i
-4.4044 - 2.7404i
-4.4044 - 2.7404i
-4.0815 + 2.8042i
-4.0815 - 2.8042i
```

polos4 =

```
-7.0703 +24.2891i
-7.0703 -24.2891i
-7.1935 +24.2103i
-7.1935 -24.2103i
-4.3941 + 2.7335i
-4.3941 - 2.7335i
-4.4028 + 2.7418i
-4.4028 - 2.7418i
```

polos5 =

```
-7.0691 +24.2854i
-7.0691 -24.2854i
-7.1891 +24.1984i
-7.1891 -24.1984i
-4.3953 + 2.7395i
-4.3953 - 2.7395i
-4.4072 + 2.7391i
-4.4072 - 2.7391i
```

polos6 =

```
-7.0726 +24.2943i
-7.0726 -24.2943i
```

```

-7.1908 +24.2039i
-7.1908 -24.2039i
-4.3918 + 2.7410i
-4.3918 - 2.7410i
-4.4055 + 2.7401i
-4.4055 - 2.7401i

poles7 =

-7.0703 +24.2891i
-7.0703 -24.2891i
-7.1935 +24.2103i
-7.1935 -24.2103i
-4.3941 + 2.7395i
-4.3941 - 2.7395i
-4.4028 + 2.7418i
-4.4028 - 2.7418i

poles8 =

-7.0726 +24.2943i
-7.0726 -24.2943i
-7.1908 +24.2039i
-7.1908 -24.2039i
-4.3918 + 2.7410i
-4.3918 - 2.7410i
-4.4055 + 2.7401i
-4.4055 - 2.7401i

poles9 =

-7.0691 +24.2854i
-7.0691 -24.2854i
-7.1891 +24.1984i
-7.1891 -24.1984i
-4.3953 + 2.7395i
-4.3953 - 2.7395i
-4.4072 + 2.7391i
-4.4072 - 2.7391i

K1=place(A1,B1,polos1)
K2=place(A2,B2,polos2)
K3=place(A3,B3,polos3)
K4=place(A4,B4,polos4)
K5=place(A5,B5,polos5)
K6=place(A6,B6,polos6)
K7=place(A7,B7,polos7)
K8=place(A8,B8,polos8)
K9=place(A9,B9,polos9)

%% Matrices G
% Gii para i=1,2,...,r
%
% 3Desigualdades
%1.- Gii'*P+P*Gii<0

```

```
%2.- (0.5*(G1j+Gj1))*P+P*(0.5*(G1j+Gj1))*i<=0
```

```
% %Gii=(Ai-Bi*Ki)
G11=(A1-B1*K1);
G22=(A2-B2*K2);
G33=(A3-B3*K3);
G44=(A4-B4*K4);
G55=(A5-B5*K5);
G66=(A6-B6*K6);
G77=(A7-B7*K7);
G88=(A8-B8*K8);
G99=(A9-B9*K9);
```

```
**G1j=Ai-Bj*Kj);
```

```
%para i=1
G12=A1-B1*K2;
G13=A1-B1*K3;
G14=A1-B1*K4;
G15=A1-B1*K5;
G16=A1-B1*K6;
G17=A1-B1*K7;
G18=A1-B1*K8;
G19=A1-B1*K9;
%para i=2
G23=A2-B2*K3;
G24=A2-B2*K4;
G25=A2-B2*K5;
G26=A2-B2*K6;
G27=A2-B2*K7;
G28=A2-B2*K8;
G29=A2-B2*K9;
%para i=3
G34=A3-B3*K4;
G35=A3-B3*K5;
G36=A3-B3*K6;
G37=A3-B3*K7;
G38=A3-B3*K8;
G39=A3-B3*K9;
%para i=4
G45=A4-B4*K5;
G46=A4-B4*K6;
G47=A4-B4*K7;
G48=A4-B4*K8;
G49=A4-B4*K9;
%para i=5
G56=A5-B5*K6;
G57=A5-B5*K7;
G58=A5-B5*K8;
G59=A5-B5*K9;
%para i=6
G67=A6-B6*K7;
G68=A6-B6*K8;
G69=A6-B6*K9;
%para i=7
G78=A7-B7*K8;
```

```

G79=A7-B7*K9;
%para i=8
G89=A8-B8*K9;

%%Gji=Aj-Bj*Ki;
%para i=1
G21=A2-B2*K1;
G31=A3-B3*K1;
G41=A4-B4*K1;
G51=A5-B5*K1;
G61=A6-B6*K1;
G71=A7-B7*K1;
G81=A8-B8*K1;
G91=A9-B9*K1;
%para i=2
G32=A3-B3*K2;
G42=A4-B4*K2;
G52=A5-B5*K2;
G62=A6-B6*K2;
G72=A7-B7*K2;
G82=A8-B8*K2;
G92=A9-B9*K2;
%para i=3
G43=A4-B4*K3;
G53=A5-B5*K3;
G63=A6-B6*K3;
G73=A7-B7*K3;
G83=A8-B8*K3;
G93=A9-B9*K3;
%para i=4
G54=A5-B5*K4;
G64=A6-B6*K4;
G74=A7-B7*K4;
G84=A8-B8*K4;
G94=A9-B9*K4;
%para i=5
G65=A6-B6*K5;
G75=A7-B7*K5;
G85=A8-B8*K5;
G95=A9-B9*K5;
%para i=6
G76=A7-B7*K6;
G86=A8-B8*K6;
G96=A9-B9*K6;
%para i=7
G87=A8-B8*K7;
G97=A9-B9*K7;
%para i=8
G98=A9-B9*K8;

%código copiado del lmi toolbox
setlmi([1]);
P=lmiivar(1,[0 1]);

lmiTerm([1 1 1 P],G11',1,'s');
% LMI #1: G11'+P+P*G11

```

```

lmiterm([2 1 1 P],G32',1,'s');
lmiterm([3 1 1 P],G33',1,'s');
lmiterm([4 1 1 P],G44',1,'s');
lmiterm([5 1 1 P],G55',1,'s');
lmiterm([6 1 1 P],G66',1,'s');
lmiterm([7 1 1 P],G77',1,'s');
lmiterm([8 1 1 P],G88',1,'s');
lmiterm([9 1 1 P],G99',1,'s');
lmiterm([10 1 1 P],.5*(0.5*(G12+G21))',1,'s');
% LMI #10: (0.5*(G12+G21))*P (NON SYMMETRIC?)
lmiterm([10 1 1 P],.5*1,((0.5*(G12+G21))),'s');
% LMI #10: P*((0.5*(G12+G21))) (NON SYMMETRIC?)
lmiterm([11 1 1 P],.5*(0.5*(G13+G31))',1,'s');
% LMI #11: (0.5*(G13+G31))*P (NON SYMMETRIC?)
lmiterm([11 1 1 P],.5*1,((0.5*(G13+G31))),'s');
% LMI #11: P*((0.5*(G13+G31))) (NON SYMMETRIC?)
lmiterm([12 1 1 P],.5*(0.5*(G14+G41))',1,'s');
% LMI #12: (0.5*(G14+G41))*P (NON SYMMETRIC?)
lmiterm([12 1 1 P],.5*1,((0.5*(G14+G41))),'s');
% LMI #12: P*((0.5*(G14+G41))) (NON SYMMETRIC?)
lmiterm([13 1 1 P],.5*(0.5*(G15+G51))',1,'s');
% LMI #13: (0.5*(G15+G51))*P (NON SYMMETRIC?)
lmiterm([13 1 1 P],.5*1,((0.5*(G15+G51))),'s');
% LMI #13: P*((0.5*(G15+G51))) (NON SYMMETRIC?)
lmiterm([14 1 1 P],.5*(0.5*(G16+G61))',1,'s');
% LMI #14: (0.5*(G16+G61))*P (NON SYMMETRIC?)
lmiterm([14 1 1 P],.5*1,((0.5*(G16+G61))),'s');
% LMI #14: P*((0.5*(G16+G61))) (NON SYMMETRIC?)
lmiterm([15 1 1 P],.5*(0.5*(G17+G71))',1,'s');
% LMI #15: (0.5*(G17+G71))*P (NON SYMMETRIC?)
lmiterm([15 1 1 P],.5*1,((0.5*(G17+G71))),'s');
% LMI #15: P*((0.5*(G17+G71))) (NON SYMMETRIC?)
lmiterm([16 1 1 P],.5*(0.5*(G18+G81))',1,'s');
% LMI #16: (0.5*(G18+G81))*P (NON SYMMETRIC?)
lmiterm([16 1 1 P],.5*1,((0.5*(G18+G81))),'s');
% LMI #16: P*((0.5*(G18+G81))) (NON SYMMETRIC?)
lmiterm([17 1 1 P],.5*(0.5*(G19+G91))',1,'s');
% LMI #17: (0.5*(G19+G91))*P (NON SYMMETRIC?)
lmiterm([17 1 1 P],.5*1,((0.5*(G19+G91))),'s');
% LMI #17: P*((0.5*(G19+G91))) (NON SYMMETRIC?)
lmiterm([18 1 1 P],.5*(0.5*(G23+G32))',1,'s');
% LMI #18: (0.5*(G23+G32))*P (NON SYMMETRIC?)
lmiterm([18 1 1 P],.5*1,((0.5*(G23+G32))),'s');
% LMI #18: P*((0.5*(G23+G32))) (NON SYMMETRIC?)
lmiterm([19 1 1 P],.5*(0.5*(G24+G42))',1,'s');
% LMI #19: (0.5*(G24+G42))*P (NON SYMMETRIC?)
lmiterm([19 1 1 P],.5*1,((0.5*(G24+G42))),'s');
% LMI #19: P*((0.5*(G24+G42))) (NON SYMMETRIC?)
lmiterm([20 1 1 P],.5*(0.5*(G25+G52))',1,'s');
% LMI #20: (0.5*(G25+G52))*P (NON SYMMETRIC?)
lmiterm([20 1 1 P],.5*1,((0.5*(G25+G52))),'s');
% LMI #20: P*((0.5*(G25+G52))) (NON SYMMETRIC?)
lmiterm([21 1 1 P],.5*(0.5*(G26+G62))',1,'s');
% LMI #21: (0.5*(G26+G62))*P (NON SYMMETRIC?)
lmiterm([21 1 1 P],.5*1,((0.5*(G26+G62))),'s');
% LMI #21: P*((0.5*(G26+G62))) (NON SYMMETRIC?)
lmiterm([22 1 1 P],.5*(0.5*(G27+G72))',1,'s');

```

```

% LMI #22: (0.5*(G27+G72))*P (NON SYMMETRIC?)
lmiterm([22 1 1 P],.5*1,((0.5*(G27+G72))),'s');
% LMI #22: P*((0.5*(G27+G72))) (NON SYMMETRIC?)
lmiterm([23 1 1 P],.5*(0.5*(G28+G82))',1,'s');
% LMI #23: (0.5*(G28+G82))*P (NON SYMMETRIC?)
lmiterm([23 1 1 P],.5*1,((0.5*(G28+G82))),'s');
% LMI #23: P*((0.5*(G28+G82))) (NON SYMMETRIC?)
lmiterm([24 1 1 P],.5*(0.5*(G29+G92))',1,'s');
% LMI #24: (0.5*(G29+G92))*P (NON SYMMETRIC?)
lmiterm([24 1 1 P],.5*1,((0.5*(G29+G92))),'s');
% LMI #24: P*((0.5*(G29+G92))) (NON SYMMETRIC?)
lmiterm([25 1 1 P],.5*(0.5*(G34+G43))',1,'s');
% LMI #25: (0.5*(G34+G43))*P (NON SYMMETRIC?)
lmiterm([25 1 1 P],.5*1,((0.5*(G34+G43))),'s');
% LMI #25: P*((0.5*(G34+G43))) (NON SYMMETRIC?)
lmiterm([26 1 1 P],.5*(0.5*(G35+G53))',1,'s');
% LMI #26: (0.5*(G35+G53))*P (NON SYMMETRIC?)
lmiterm([26 1 1 P],.5*1,((0.5*(G35+G53))),'s');
% LMI #26: P*((0.5*(G35+G53))) (NON SYMMETRIC?)
lmiterm([27 1 1 P],.5*(0.5*(G36+G63))',1,'s');
% LMI #27: (0.5*(G36+G63))*P (NON SYMMETRIC?)
lmiterm([27 1 1 P],.5*1,((0.5*(G36+G63))),'s');
% LMI #27: P*((0.5*(G36+G63))) (NON SYMMETRIC?)
lmiterm([28 1 1 P],.5*(0.5*(G37+G73))',1,'s');
% LMI #28: (0.5*(G37+G73))*P (NON SYMMETRIC?)
lmiterm([28 1 1 P],.5*1,((0.5*(G37+G73))),'s');
% LMI #28: P*((0.5*(G37+G73))) (NON SYMMETRIC?)
lmiterm([29 1 1 P],.5*(0.5*(G38+G83))',1,'s');
% LMI #29: (0.5*(G38+G83))*P (NON SYMMETRIC?)
lmiterm([29 1 1 P],.5*1,((0.5*(G38+G83))),'s');
% LMI #29: P*((0.5*(G38+G83))) (NON SYMMETRIC?)
lmiterm([30 1 1 P],.5*(0.5*(G39+G93))',1,'s');
% LMI #30: (0.5*(G39+G93))*P (NON SYMMETRIC?)
lmiterm([30 1 1 P],.5*1,((0.5*(G39+G93))),'s');
% LMI #30: P*((0.5*(G39+G93))) (NON SYMMETRIC?)
lmiterm([31 1 1 P],.5*(0.5*(G45+G54))',1,'s');
% LMI #31: (0.5*(G45+G54))*P (NON SYMMETRIC?)
lmiterm([31 1 1 P],.5*1,((0.5*(G45+G54))),'s');
% LMI #31: P*((0.5*(G45+G54))) (NON SYMMETRIC?)
lmiterm([32 1 1 P],.5*(0.5*(G46+G64))',1,'s');
% LMI #32: (0.5*(G46+G64))*P (NON SYMMETRIC?)
lmiterm([32 1 1 P],.5*1,((0.5*(G46+G64))),'s');
% LMI #32: P*((0.5*(G46+G64))) (NON SYMMETRIC?)
lmiterm([33 1 1 P],.5*(0.5*(G47+G74))',1,'s');
% LMI #33: (0.5*(G47+G74))*P (NON SYMMETRIC?)
lmiterm([33 1 1 P],.5*1,((0.5*(G47+G74))),'s');
% LMI #33: P*((0.5*(G47+G74))) (NON SYMMETRIC?)
lmiterm([34 1 1 P],.5*(0.5*(G48-G84))',1,'s');
% LMI #34: (0.5*(G48+G84))*P (NON SYMMETRIC?)
lmiterm([34 1 1 P],.5*1,((0.5*(G48+G84))),'s');
% LMI #34: P*((0.5*(G48+G84))) (NON SYMMETRIC?)
lmiterm([35 1 1 P],.5*(0.5*(G49+G94))',1,'s');
% LMI #35: (0.5*(G49+G94))*P (NON SYMMETRIC?)
lmiterm([35 1 1 P],.5*1,((0.5*(G49+G94))),'s');
% LMI #35: P*((0.5*(G49+G94))) (NON SYMMETRIC?)
lmiterm([36 1 1 P],.5*(0.5*(G56+G65))',1,'s');
% LMI #36: (0.5*(G56+G65))*P (NON SYMMETRIC?)

```

```

lmiterm([36 1 1 P],.5*1,((0.5*(G56+G65))), 's');
% LMI #36: P*((0.5*(G56+G65))) (NON SYMMETRIC?)
lmiterm([37 1 1 P],.5*(0.5*(G57+G75))',1,'s');
% LMI #37: (0.5*(G57+G75))*P (NON SYMMETRIC?)
lmiterm([37 1 1 P],.5*1,((0.5*(G57+G75))), 's');
% LMI #37: P*((0.5*(G57+G75))) (NON SYMMETRIC?)
lmiterm([38 1 1 P],.5*(0.5*(G58+G85))',1,'s');
% LMI #38: (0.5*(G58+G85))*P (NON SYMMETRIC?)
lmiterm([38 1 1 P],.5*1,((0.5*(G58+G85))), 's');
% LMI #38: P*((0.5*(G58+G85))) (NON SYMMETRIC?)
lmiterm([39 1 1 P],.5*(0.5*(G59+G95))',1,'s');
% LMI #39: (0.5*(G59+G95))*P (NON SYMMETRIC?)
lmiterm([39 1 1 P],.5*1,((0.5*(G59+G95))), 's');
% LMI #39: P*((0.5*(G59+G95))) (NON SYMMETRIC?)
lmiterm([40 1 1 P],.5*(0.5*(G67+G76))',1,'s');
% LMI #40: (0.5*(G67+G76))*P (NON SYMMETRIC?)
lmiterm([40 1 1 P],.5*1,((0.5*(G67+G76))), 's');
% LMI #40: P*((0.5*(G67+G76))) (NON SYMMETRIC?)
lmiterm([41 1 1 P],.5*(0.5*(G68+G86))',1,'s');
% LMI #41: (0.5*(G68+G86))*P (NON SYMMETRIC?)
lmiterm([41 1 1 P],.5*1,((0.5*(G68+G86))), 's');
% LMI #41: P*((0.5*(G68+G86))) (NON SYMMETRIC?)
lmiterm([42 1 1 P],.5*(0.5*(G69+G96))',1,'s');
% LMI #42: (0.5*(G69+G96))*P (NON SYMMETRIC?)
lmiterm([42 1 1 P],.5*1,((0.5*(G69+G96))), 's');
% LMI #42: P*((0.5*(G69+G96))) (NON SYMMETRIC?)
lmiterm([43 1 1 P],.5*(0.5*(G78+G87))',1,'s');
% LMI #43: (0.5*(G78+G87))*P (NON SYMMETRIC?)
lmiterm([43 1 1 P],.5*1,((0.5*(G78+G87))), 's');
% LMI #43: P*((0.5*(G78+G87))) (NON SYMMETRIC?)
lmiterm([44 1 1 P],.5*(0.5*(G79+G97))',1,'s');
% LMI #44: (0.5*(G79+G97))*P (NON SYMMETRIC?)
lmiterm([44 1 1 P],.5*1,((0.5*(G79+G97))), 's');
% LMI #44: P*((0.5*(G79+G97))) (NON SYMMETRIC?)
lmiterm([45 1 1 P],.5*(0.5*(G89+G98))',1,'s');
% LMI #45: (0.5*(G89+G98))*P (NON SYMMETRIC?)
lmiterm([45 1 1 P],.5*1,((0.5*(G89+G98))), 's');
% LMI #45: P*((0.5*(G89+G98))) (NON SYMMETRIC?)
lmiterm([-46 1 1 P],1,1);
% LMI #46: P

```

Mitsu=getlmis:

```

%% Resolver la LMI, la respuesta es buena si tmin<=0
[tmin,pfeas]=feasp(Mitsu);
%% Ahora se extraen las variables del solucionador de IT's (pfeas)
% Asegurarse de utilizar el mismo identificador que en lmiedit
Pfinal = dec2mat(Mitsu,pfeas,P)

```

B.4 Desigualdades LMI

“LMI Toolbox MATLAB”

En la Figura B.1 se muestra la ventana principal del editor de LMI. Como primera instrucción, se debe de asignar un nombre al sistema LMI, en la parte superior de la ventana; al presente trabajo se le llamo “Mitsu”. En el segundo apartado, se deben describir las variables matriciales, las cuales se dividen en 3 partes: en la primera se declara la variable a resolver; en la segunda se define si es una matriz simétrica cuadrada (S) o una matriz rectangular (R); en la tercera se especifica la estructura de la matriz, no las dimensiones de la misma, es decir, el primer término que se encuentra en los corchetes se refiere al número de componentes que posee la diagonal principal, y el segundo término indica el número de diagonales que posee.

En el tercer apartado se escriben las desigualdades matriciales que se deben de cumplir para asegurar la estabilidad del sistema. En este trabajo de Tesis se utilizaron las desigualdades del capítulo 2. Se utilizaron nueve reglas, por lo que $r=9$, dando como resultado un total de 46 desigualdades.

Una vez que el LMI está con los datos completos colocados, es necesario dar un click en la pestaña “view commands”, para poder generar los comandos que se utilizarán para resolver las desigualdades de forma numérica en MATLAB.

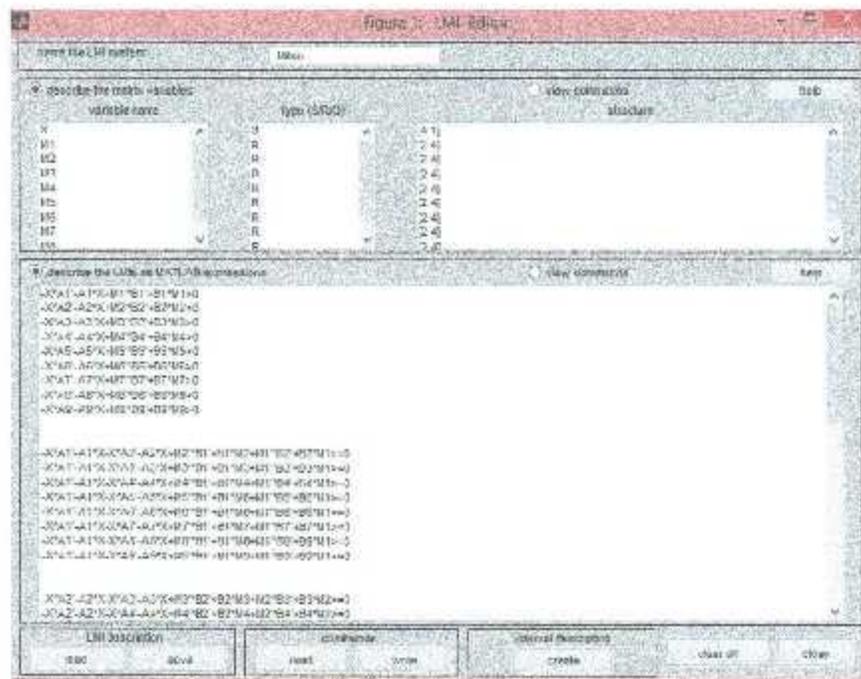


Figura B.1.- Editor de LMI

C. Bloque “PA10 Control”

El motor de la interfaz gráfica de usuario, la cual es necesaria para controlar el robot Mitsubishi PA10-7CE, es el bloque de Simulink “PA10 Control” desarrollado por Quanser (Figura C.1). Este bloque es el medio de comunicación entre MATLAB/Simulink y el robot Mitsubishi PA10-7CE. Dicho bloque tiene la labor de enviar las consignas de par o velocidad hacia el robot, y a la inversa, de recibir ángulos, velocidades y pares desde el robot.

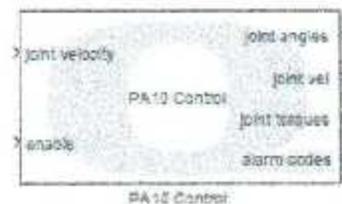


Figura C.1.- Bloque para el control del PA10 en tiempo real

Cada articulación puede ser habilitada, deshabilitada y controlada independientemente. Este bloque utiliza el protocolo ARCNET para comunicarse con el robot. Enseguida se presenta una breve descripción de los puertos de entrada y de salida que componen al bloque “PA10 Control”, aplicado al robot Mitsubishi PA10-7CE.

- **Joint velocity/torque:** vector de 7 elementos que posee las consignas de velocidad [rad/seg] o de par [N m], en función del modo de operación del robot.
- **Enable:** vector de 7 elementos que indica si cada una de las articulaciones está habilitada (1 = *verdadero*) o deshabilitada (0 = *false*).
- **Joint angles:** vector de 7 elementos que posee las posiciones articulares actuales de cada eslabón [rad].
- **Joint vel:** vector de 7 elementos que posee las velocidades articulares actuales de cada eslabón [rad/seg].
- **Joint torques:** vector de 7 elementos que posee los pares articulares actuales de cada eslabón [N m].

- Alarm codes: vector de 7 elementos que posee los códigos de alarmas que entrega el sistema PA10.

Dentro de este bloque se deben configurar algunos parámetros, los cuales se muestran en la Figura C.2, y se detallan a continuación.

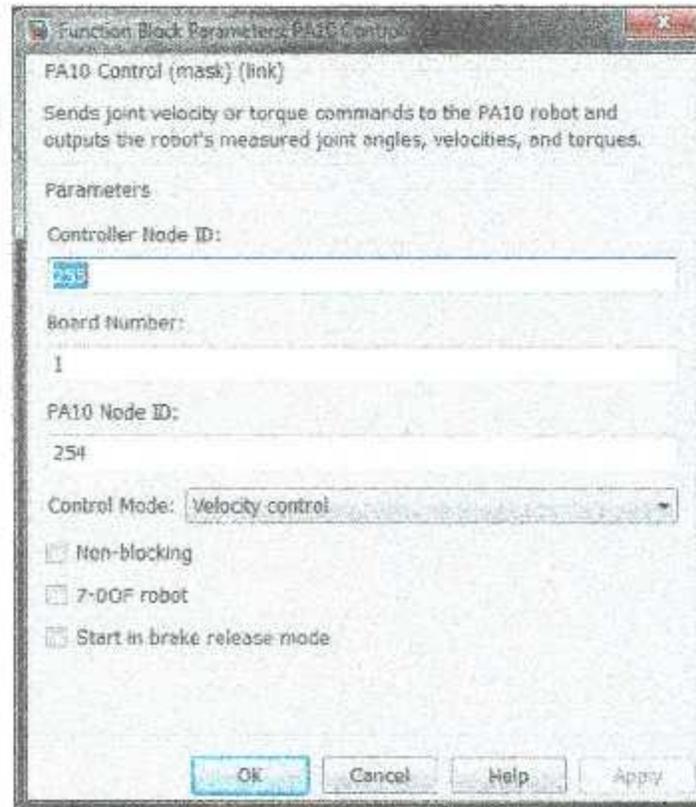


Figura C.2.- Parámetros del bloque PA10 Control

- Controller Node ID: se refiere al nodo identificador de la tarjeta ARCNET que posee la computadora encargada de controlar el robot. Se requiere que este ID sea 255.
- Board Number: es el número de la tarjeta ARCNET en la computadora asignada para el control del robot. El valor establecido para la computadora del Mitsubishi PA10 debe ser 1.
- PA10 Node ID: es el nodo identificador de la tarjeta ARCNET que posee la caja de control del robot. Este nodo debe ser 254.
- Control Mode: selecciona si los comandos dados se utilizarán para el control en modo velocidad o en modo par.
- Non-blocking: selecciona si el bloque funciona con métodos de comunicación no bloqueante. La operación predeterminada es bloqueante, y el bloque puede ser colocado en

un subproceso asincrónico para evitar muestras perdidas en el caso de que la frecuencia de muestreo sea alta.

- 7-DOF robot: se debe seleccionar este recuadro para poder trabajar con la versión de 7 grados de libertad del sistema PA10. Si esta opción no es seleccionada se asume que la versión utilizada es la de 6 grados de libertad.
- Start in brake release mode: esta opción se seleccionará para iniciar el PA10 en el modo de liberación del freno. Al seleccionar este recuadro quedarán activos los frenos de forma conjunta y todas las consignas de control serán ignoradas, por lo que al no marcarlo, el PA10 podrá trabajar normalmente.