UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

IMPLEMENTACIÓN DE CONTROLADORES DIFUSOS EN UN SISTEMA DE ROBOTS INDUSTRIALES

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO ELÉCTRICO ELECTRÓNICO

P R E S E N T A :

LUIS ENRIQUE ARANDA MELO

DIRECTOR DE TESIS: MARCO ANTONIO ARTEAGA PÉREZ

MEXICO, D.F. NOVIEMBRE 2003

A

PAGINACIÓN DISCONTINUA

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Ingeniería

Implementación de controladores Difusos en un Sistema de Robots Industriales

LUIS E. ARANDA M.

Asesor: Dr. Marco Antonio Arteaga Pérez

México, Noviembre 2003

Agradecimientos :

Agradezco al Dr. Marco Antonio Arteaga Pérez por su apoyo y ayuda para realización de este trabajo .

Dedico esta tesis a:

A Dios por darme vida y permitirme culminar esta importante etapa de mi vida.

A mi familia que siempre me ha apoyado en mis deciciones, gracias Maria Luisa, Irma, Paty, Moy, Tere y Bety.
A Miriam por enseñarme a soñar.
Amis amigos que siempre me acompañaron a lo largo de la carrera. Bety, Erika, Rosalba, Cecilia, Claudia,gracias a todos y tambien a todos lo que me faltaron nombrar
Amis amigos del laboratorio gracias Algelica, Jorge, Juan C,Eduardo, Adrian.

En este camino donde no hay nada,
si vamos los dos juntos talvez encontremos algo como esa luna en el cielo.

V

Índice General

Índice de Figuras

Capítulo 1

Introducción

Las bases de la lógica difusa fueron enunciadas a principios de 1960 por Lotfit A. Zadeh , profesor de Ingeniería en la Universidad de California en Berkeley; pero no es hasta 1973 que presenta la teoría básica de los conjuntos difusos (Martin del Brio and Sanz Molina 2001). Por otra parte, es Mandami el primero que aplica la lógica difusa para controlar un sistema de vapor. Es así como se comienzan a hacer aplicaciones. Otra aplicación bien conocida es la que Smidth hizo para el control de hornos rotativos en una cementera.

Especial mención merece la creación de LIFE (*Laboratory for International Fuzzy Engineering Research*) en marzo de 1989, auspiciado por el ministerio de Comercio Internacional e Industria de Japón (*MITI*). Su capital es 50% de compañías privadas japonesas y en él trabajan alrededor de 30 investigadores. En Estados Unidos y Europa se empezó a dar importancia a la lógica difusa cuando desde Japón llegó información sobre numerosas aplicaciones prácticas. A partir de entonces, empresas norteamericanas como la NASA, Boeing y Ford comenzaron a aplicar la lógica difusa debido a que permite tratar información imprecisa, como estatura "*media*", temperatura "*baja*" o "*mucha*" fuerza. A lo largo de esta tesis se verá la teoría de la lógica difusa aplicada al control, así como sus reglas para definir acciones. Es importante mencionar que para problemas no lineales o no bien definidos, la lógica difusa permite controlar cualquier proceso y aprender de los datos haciendo uso de determinados algoritmos de aprendizaje como los sistema expertos.

Desde su aparición en 1960 hasta nuestros días, las aplicaciones de la lógica difusa se han consolidado paulatinamente. Se encuentran por ejemplo soluciones a problemas de control industrial, en investigación operacional, en estrategias de mantenimiento preventivo y en otros campos más. Las principales razones para tal proliferación quizás sean la sencillez conceptual de los sistemas basados en lógica difusa, su facilidad para adaptarse a casos particulares con pocas variaciones de parámetros, su habilidad para combinar en forma unificada expresiones lingüísticas con datos numéricos, y no requiere de algoritmos muy sofisticados para su implementación. El propósito fundamental de esta tesis es implementar algoritmos de control basados en la lógica difusa.

1.1 Planteamiento del Problema

El uso de técnicas de control clásico como el control PID de ganancia fija, resulta ser una buena alternativa para controlar sistemas dinámicos, ya que proporciona tiempos de respuesta rápidos. Sin embargo, entre mayor sea la precisión requerida en el sistema, el ajuste de este tipo de controladores es más difícil, ya que son bastante sensibles a las señales de ruido y en ocasiones introducen oscilaciones cuando se presentan retardos en el sistema. Cuando la dinámica de los sistemas es no lineal, el control debe tener la capacidad de compensar la no-linealidad, usualmente un controlador PID no ofrece un desempeño aceptable en estos casos. Esta no-linealidad difícilmente puede ser caracterizada por una ecuación dinámica, por lo que en la mayoría de los casos se trata subjetivamente por el operador del proceso. Esta subjetividad tiene implicaciones profundas al modelar este tipo de sistemas a través de la lógica difusa. La implementación de controladores PID en hardware basados en lógica difusa, es motivada por su habilidad para capturar estrategias cualitativas de control y su capacidad de implementación. El comportamiento del control PID como difuso permite lograr que los sistemas puedan ajustarse a condiciones cambiantes que son muchas veces imposibles de predecir. En contraste, un control PID está basado en un modelo matemático del proceso. Estos modelos se desarrollan por medio del cálculo del error de posición y coeficientes asignados al control que son del tipo proporcional, integral y derivativo del sistema.

En la presente tesis se describe la implementación de dos algoritmos de control difusos propuestos por Tao y Taur (2000): Control PID difuso (PIDF), Control PID flexible como difuso (FPIDF). También se implementó un controlador difuso utilizando el método de Síntesis de Lyapunov (Margaliot and Langholz 2000). En el Capítulo 2 se describen los conceptos matemáticos de los controles así como los de la lógica difusa. En el Capítulo 3 se muestran los controladores propuestos. El Capítulo 4 presenta los parámetros y resultados obtenidos. El Capítulo 5 incluye las conclusiones.

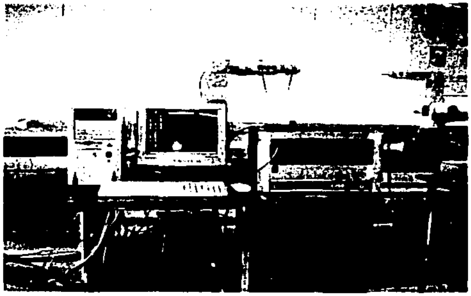


Figura 1.1: Sistema de robots experimentales del laboratorio de robótica DEPFI

Capítulo 2

Preliminares Matemáticas

Este capítulo presenta las bases matemáticas de las cinemática de los robots, así como la planeacion de las trayectorias a seguir conclutyendo con las bases de la lógica difusa.

2.1 Cinemática de un brazo Robot

En el análisis de un robot manipulador es necesario conocer su cinemática, dinámica y entradas de control, para así poder tener un buen conocimiento del mismo. A continuacion se analizan los conceptos básicos de la dinámica y cinemática de los robots A465 y A255 del laboratorio de la DEPFI, así como los de la lógica difusa.

2.1.1 Cinemática directa

El problema de la cinemática directa consiste en determinar la posición del *efector final* de un robot de acuerdo con la posición angular que tiene cada una de las articulaciones. La forma más común de representar la cinemática directa es a través de una matriz de transformación homogénea ${}^0T_n \in \mathbb{R}^{4\times 4}$, donde n es el número de articulaciones del robot. La matriz ${}^0T_n$ se define como:

$$ {}^0T_n = {}^0A_1 \; {}^1A_2 \; \ldots \; {}^{i-1}A_1 $$

Donde ${}^{i-1}A_1$ con $i = 1, 2, \ldots, n$, es la matriz homogénea que transforma la posición de un punto del sistema de coordenadas i al sistema $i-1$. Las Figuras 2.1 y 2.2 muestran un esquema de los robots y su correspondiente diagrama de cuerpo libre. Para el caso del robot A465 se utilizó el algoritmo de Denavit-Hartenberg para obtener las matrices de transformación (Spong and Vidyasagar 1989). Para el robot

A255, las matrices de transformación se obtienen por inspección debido a su configuración mecánica. Las matrices de transformación homogénea quedan como sigue

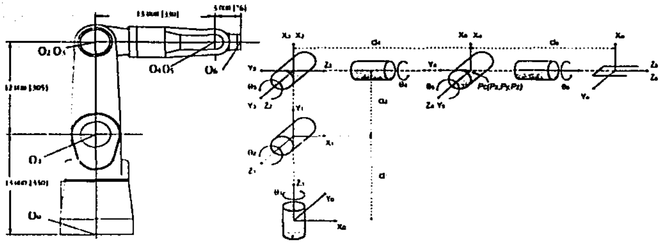
A	Robot A255		Robot A465	
	a_i(metros)	d_i(metros)	a_i(metros)	d_i(metros)
1	0	0.254	0	0.330
2	0.254	0	0.305	0
3	0.254	0	0	0
4	0	0	0	0.330
5	0	0.051	0	0
6	—	—	0	0.076

Tabla 2.1: Parámetros de los robot A255 y A465.

Debido a que nuestro sistema esta compuesto de dos robots y solo se emplearan las tres primeras articulaciones de cada uno de los robots, la cinematica directa de los robots queda de la siguente forma.

Robot A465

La cinematica directa del robot $A465$ se muetra a continuacion utilizando el método de Denavit-Hartenberg, esta matriz de transformación homogénea fue tomada de (Castillo 2002) y los parametros de la Tabla 2.1.



Esquema robot A465 Diagrama de cuerpo libre

Figura 2.1: Esquema y diagrama de cuerpo libre para el robot A465.

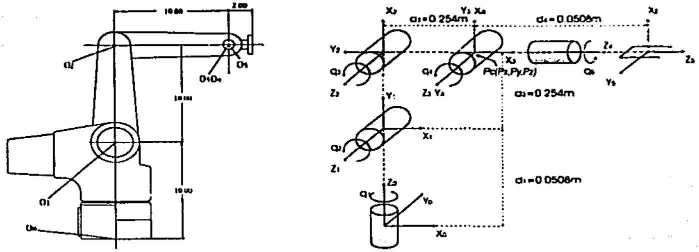
$$
{}^{0}A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0.33 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{1}A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & 0.305\,C_2 \\ S_2 & C_2 & 0 & 0.305\,S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{2}A_3 = \begin{bmatrix} C_3 & 0 & S_3 & 0 \\ S_3 & 0 & -C_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{3}A_4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & 0.33 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}A_5 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{5}A_6 = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0.076 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Robot A255

La cinematica directa del robot $A255$,se obtiene por el método de inspección debido a la configuración mecánica del robot, esta matriz de transformación homogénea fue tomada de (Castillo 2002) y los parametros de la Tabla 2.1.



Esquema robot A255 — Diagrama de cuerpo libre

Figura 2.2: Esquema y diagrama de cuerpo libre para el robot A255.

$$
{}^0A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^1A_2 = \begin{bmatrix} 1 & 0 & 0 & a_2C_2 \\ 0 & 1 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^2A_3 = \begin{bmatrix} 1 & 0 & 0 & a_3C_3 \\ 0 & 1 & 0 & a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^3A_4 = \begin{bmatrix} -S_4 & 0 & C_4 & 0 \\ C_4 & 0 & S_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^4A_5 = \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

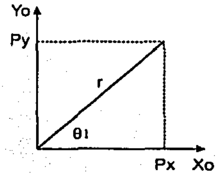
con $s_i = \sin(\theta i)$ y $c_i = \cos(\theta i)$

Una vez obtenida la dinámica de los robots, se utilizarán solo las tres primeras articulaciones en ambos casos, por lo que mecánicamente se harán $\theta 4 = \theta 5 = \theta 6 \equiv 0$ para el robot A465 y $\theta 4 = \theta 5 \equiv 0$ para el A255.

2.1.2 Cinemática inversa

A continuacion se calculará la cinemática inversa de los robots. El problema de la cinemática inversa consiste en determinar los ángulos de cada una de las articulaciones de los robots a partir de la posición, en coordenadas rectangulares (X, Y, Z), del efector final. La cinemática inversa de posición para los robots A465 y A255 se describe a continuación.

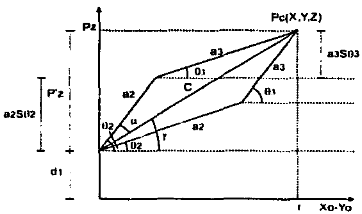
Calculo de θ_1. El calculo de θ_1 es idéntico para los dos robots. La Figura 2.3 muestra el plano $X_0 - Y_0$, donde se puede ver la relación de θ_1 con P_x y P_y.



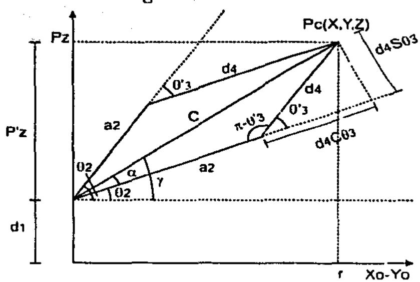
$$\therefore \theta_1 = \arctan\left(\frac{P_y}{P_x}\right)$$

Figura 2.3: Plano $X_0 - Y_0$.

Cálculo de θ_2 y θ_3. El cálculo de θ_2 y θ_3 se basa en la Figura 2.4



Esquema para el robot A255



Esquema para el robot A465

Figura 2.4: Diagrama vectorial de las articulaciones 2 y 3 de los robots.

Robot A255	$\theta_2 = \arctan\left(\frac{\pm\sqrt{1-\sigma^2}}{\sigma}\right) + \arctan\left(\frac{P_z-d_1}{\sqrt{P_x^2+P_y^2}}\right)$ $\theta_3 = \arctan\left(\frac{\psi}{\pm\sqrt{a_3^2-\psi^2}}\right)$
Robot A465	$\theta_2 = \arctan\left(\frac{P_z-d_1}{\sqrt{P_x^2+P_y^2}}\right) - \arctan\left(\frac{d_4 \sin\theta_3'}{a_2+d_4\cos\theta_3'}\right)$ $\theta_3 = \theta_3' + \frac{\pi}{2}$

donde

$$\sigma \triangleq \frac{(a_2)^2+(P_z')^2+(P_x)^2+(P_y)^2-(a_3)^2}{2a_2\sqrt{(P_z')^2+(P_x)^2+(P_y)^2}}$$

$$\psi \triangleq P_z - (a_2 S_2 + d_1)$$

$$\theta_3' = \arctan\left(\frac{\pm\sqrt{1-D^2}}{D}\right)$$

$$D \triangleq \frac{(P_z-d_1)^2+P_x^2+P_y^2-a_2^2-d_4^2}{2a_2 d_4}$$

2.2 Planeación de trayectorias

La rutina para manipulación de un objeto se necesita la planeación de los movimientos de los robots en el espacio de tres dimensiones (X, Y, Z). Las trayectorias presentadas se tomaron de (Bentrup 2002).

Las trayectorias deseadas están compuestas por segmentos de curva descritas por los siguientes polinomios de quinto orden:

$$\begin{aligned}
&Art1\ A465: && x_{d1} = x_{01} + a_{x_{31}}(t^3) + a_{x_{41}}(t^4) + a_{x_{51}}(t^5)\\
&Art2\ A465: && y_{d1} = y_{01} + a_{y_{31}}(t^3) + a_{y_{41}}(t^4) + a_{y_{51}}(t^5)\\
&Art3\ A465: && z_{d1} = z_{01} + a_{z_{31}}(t^3) + a_{z_{41}}(t^4) + a_{z_{51}}(t^5)\\
&Art1\ A255: && x_{d2} = x_{02} + a_{x_{32}}(t^3) + a_{x_{42}}(t^4) + a_{x_{52}}(t^5)\\
&Art2\ A255: && y_{d2} = y_{02} + a_{y_{32}}(t^3) + a_{y_{42}}(t^4) + a_{y_{52}}(t^5)\\
&Art3\ A255: && z_{d2} = z_{02} + a_{z_{32}}(t^3) + a_{z_{42}}(t^4) + a_{z_{52}}(t^5)
\end{aligned}$$

Los coeficientes de estas trayectorias son los siguientes:

$x_{01} = 0.506$	$ax_{31} = 0.008688$	$ax_{41} = -0.0026064$	$ax_{51} = 0.000208512$
$y_{01} = 0$	$ay_{31} = 0$	$ay_{41} = 0$	$ay_{51} = 0$
$z_{01} = 0.635$	$az_{31} = -0.022$	$az_{41} = 0.0066$	$az_{51} = -0.000528$
$x_{02} = 1.0432$	$ax_{32} = -0.008688$	$ax_{42} = 0.0026064$	$ax_{52} = -0.000208512$
$y_{02} = 0$	$ay_{32} = 0$	$ay_{42} = 0$	$ay_{52} = 0$
$z_{02} = 0.635$	$az_{32} = -0.022$	$az_{42} = 0.0066$	$az_{52} = -0.000528$

Tabla 2.2: **Intervalo:** $(0, 5] \rightarrow 0\text{s} < t \leq 5\text{s}$.

$x_{01} = 0.6146$	$ax_{31} = 0.0008$	$ax_{41} = -0.00024$	$ax_{51} = 0.0000192$
$y_{01} = 0$	$ay_{31} = 0$	$ay_{41} = 0$	$ay_{51} = 0$
$z_{01} = 0.36$	$az_{31} = 0$	$az_{41} = 0$	$az_{51} = 0$
$x_{02} = 0.9346$	$ax_{32} = -0.0008$	$ax_{42} = 0.00024$	$ax_{52} = -0.0000192$
$y_{02} = 0$	$ay_{32} = 0$	$ay_{42} = 0$	$ay_{52} = 0$
$z_{02} = 0.36$	$az_{32} = 0$	$az_{42} = 0$	$az_{52} = 0$

Tabla 2.3: **Intervalo:** $(5, 10] \rightarrow 5\text{s} < t \leq 10\text{s}$.

$x_{01} = 0.6246$	$ax_{31} = 0.004$	$ax_{41} = -0.0012$	$ax_{51} = 0.000096$
$y_{01} = 0$	$ay_{31} = 0$	$ay_{41} = 0$	$ay_{51} = 0$
$z_{01} = 0.36$	$az_{31} = 0.022$	$az_{41} = -0.0066$	$az_{51} = 0.000528$
$x_{02} = 0.9246$	$ax_{32} = 0.004$	$ax_{42} = -0.0012$	$ax_{52} = 0.000096$
$y_{02} = 0$	$ay_{32} = 0$	$ay_{42} = 0$	$ay_{52} = 0$
$z_{02} = 0.36$	$az_{32} = 0.022$	$az_{42} = -0.0066$	$az_{52} = 0.000528$

Tabla 2.4: **Intervalo:** $(10, 15] \rightarrow 10\text{s} < t \leq 15\text{s}$.

Para $15\text{s} < t \leq 25\text{s}$, se tiene los siguientes polinomios

$$w = w_0 + ax_{31}(t^3) + ax_{41}(t^4) + ax_{51}(t^5)$$
$$x_{d1} = xr + rcos(w)$$
$$y_{d1} = yr + rsin(w)$$
$$z_{d1} = zr + rcos(w) - r$$
$$x_{d2} = x_{d1} + largo$$
$$y_{d2} = y_{d1}$$
$$z_{d2} = z_{d1}$$

$w_0 = 0$	$ax_{31} = 0.062832$	$ax_{41} = -0.0094248$	$ax_{51} = 0.000376992$
$x_r = 0.6246$	$y_r = 0$	$z_r = 0.635$	$r = 0.05$
$largo = 0.3$			

Tabla 2.5: **Intervalo:** $(15, 25] \rightarrow 15\text{s} < t \leq 25\text{s}$.

$x_{01} = 0.6746$	$ax_{31} = -0.004$	$ax_{41} = 0.0012$	$ax_{51} = -0.000096$
$y_{01} = 0$	$ay_{31} = 0$	$ay_{41} = 0$	$ay_{51} = 0$
$z_{01} = 0.635$	$az_{31} = -0.022$	$az_{41} = 0.0066$	$az_{51} = -0.000528$
$x_{02} = 0.9746$	$ax_{32} = -0.004$	$ax_{42} = 0.0012$	$ax_{52} = -0.000096$
$y_{02} = 0$	$ay_{32} = 0$	$ay_{42} = 0$	$ay_{52} = 0$
$z_{02} = 0.635$	$az_{32} = -0.022$	$az_{42} = 0.0066$	$az_{52} = -0.000528$

Tabla 2.6: **Intervalo:** $(25, 30] \rightarrow 25\text{s} < t \leq 30\text{s}$.

$x_{01} = 0.6246$	$ax_{31} = -0.0008$	$ax_{41} = 0.00024$	$ax_{51} = -0.0000192$
$y_{01} = 0$	$ay_{31} = 0$	$ay_{41} = 0$	$ay_{51} = 0$
$z_{01} = 0.36$	$az_{31} = 0$	$az_{41} = 0$	$az_{51} = 0$
$x_{02} = 0.9246$	$ax_{32} = 0.0008$	$ax_{42} = -0.00024$	$ax_{52} = 0.0000192$
$y_{02} = 0$	$ay_{32} = 0$	$ay_{42} = 0$	$ay_{52} = 0$
$z_{02} = 0.36$	$az_{32} = 0$	$az_{42} = 0$	$az_{52} = 0$

Tabla 2.7: **Intervalo:** $(30, 35] \rightarrow 30\text{s} < t \leq 35\text{s}$.

$x_{01} = 0.6146$	$ax_{31} = -0.008688$	$ax_{41} = 0.0026064$	$ax_{51} = -0.000208512$
$y_{01} = 0$	$ay_{31} = 0$	$ay_{41} = 0$	$ay_{51} = 0$
$z_{01} = 0.36$	$az_{31} = 0.022$	$az_{41} = -0.0066$	$az_{51} = 0.000528$
$x_{02} = 0.9346$	$ax_{32} = 0.008688$	$ax_{42} = -0.0026064$	$ax_{52} = 0.000208512$
$y_{02} = 0$	$ay_{32} = 0$	$ay_{42} = 0$	$ay_{52} = 0$
$z_{02} = 0.36$	$az_{32} = 0.022$	$az_42 = -0.0066$	$az_{52} = 0.000528$

Tabla 2.8: **Intervalo:** $(35.40] \rightarrow 35\text{s} < t \leq 40\text{s}$.

Las Figuras 2.5 y 2.6 muestran las trayectorias que deben seguir los robots.

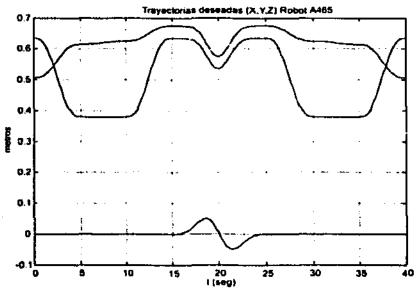


Figura 2.5: Trayectoria deseada en coordenadas rectangulares (X, Y, Z) para el robot A465.

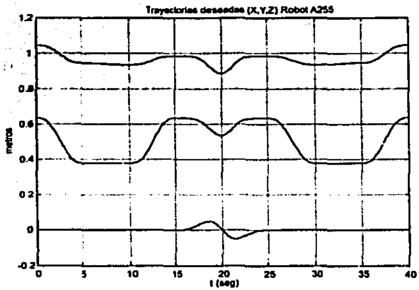


Figura 2.6: Trayectoria deseada en coordenadas rectangulares (X, Y, Z) para el robot A255.

2.3 Lógica difusa

El profesor Zadeh propuso los principios de la lógica difusa en los cuales permite manejar la ambigüedad o incertidumbre; por ello ha tenido un gran éxito en aplicaciones donde se requiere de un procesamiento rápido de la información(Martin del Brio and Sanz Molina 2001).

Un ejemplo de ambigüedad es cuando se habla de la estatura de un hombre. Una persona de 1.50 m puede decir que alto es quien mide 1.70 m; sin embargo, alguien de esta estatura dirá que 1.80 m es alto y esta ultima persona que son altas las mayores a 2 m. Sin embargo, también existe ambigüedad en las respuestas, esto es que alguien dirán que una persona alta es aquella que mide entre 1.70 y 2 m. Esto quiere decir que si una persona mide 1.69 no es alta. Como se observa existe un rango de estaturas en los que se puede denominar a un hombre como "*alto*"; es decir el concepto "*hombre alto*" no está bien definido. Como este ejemplo existen muchos más: hombre viejo, temperatura "*baja*", etc. Además estos enunciados muchas veces requieren de mayor información para saber a que se aplica (Beale 1994).

2.3.1 Principios de Lógica difusa

Una buena estrategia para presentar la teoría de conjuntos difusos, consiste en recordar algunos aspectos de la teoría de conjuntos convencionales (que en adelante se denominaran conjuntos concretos). A partir de allí se puede hacer una extensión a los conjuntos difusos: Un conjunto concreto se define como una colección de elementos que existen dentro de un universo. Si el universo consta de los números enteros no negativos menores que 10, estos forman el universo representado por el conjunto U:

$$U = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Algunos subconjuntos de U son:

$$A = \{0, 2, 4, 6, 8\}$$

$$B = \{1, 3, 5, 7, 9\}$$

$$C = \{1, 4, 5, 8\}$$

Con base en estas definiciones se establece que cada uno de los elementos del universo que pertenece o no a un conjunto determinado. Por lo tanto, cada conjunto puede definirse completamente por una función de pertenencia, que opera sobre los elementos del universo y a la que se le asigna un valor de 1 si el elemento pertenece al conjunto, y de 0 en caso contrario. Tomando como ejemplo el conjunto C mencionado anteriormente, su función de pertenencia o membresía $U_c(x)$, es de la siguiente forma:

$$U_c(0) = 0, U_c(1) = 1, U_c(2) = 0, U_c(3) = 0, U_c(4) = 1,$$
$$U_c(5) = 1, U_c(6) = 0, U_c(7) = 0, U_c(8) = 1, U_c(9) = 0.$$

Ahora bien, un conjunto difuso se define de forma similar, con una diferencia conceptual importante: un elemento puede pertenecer parcialmente a un conjunto. De este modo, un conjunto difuso D definido sobre el mismo universo U puede ser el siguiente:

$$D = \{0.2/1, 0.3/4, 0.5/5, 1/8\}$$

La definición anterior significa que el elemento 1 pertenece 0.2 al conjunto D (y por tanto pertenece 0.8 al complemento de D), en tanto que el elemento 4 pertenece 0.3 al conjunto D. El elemento 5 pertenece 0.5 y el elemento 8 pertenece 1.
En forma alternativa, se dice que la función de pertenecía o membresía $U_D(x)$, del conjunto D es la siguiente:

$$U_D(0) = 0.0, U_D(1) = 0.2, U_D(2) = 0.0, U_D(3) = 0.0, U_D(4) = 0.3,$$
$$U_D(5) = 0.5, U_D(6) = 0.0, U_D(7) = 0.0, U_D(8) = 1.0, U_D(9) = 0.0.$$

Las primeras diferencias que se hacen evidentes entre los conjuntos concretos y difusos son las siguientes:

- La función de pertenencia asociada a los conjuntos concretos sólo puede tener dos valores 0 ó 1, mientras que en los conjuntos difusos puede tomar cualquier valor entre 0 y 1.

- Un elemento puede pertenecer (parcialmente) a un conjunto difuso y simultáneamente pertenecer (parcialmente) al complemento de dicho conjunto. Lo anterior no es posible en los conjuntos concretos, ya que constituiría una violación al principio de probabilidad del tercer excluido.

2.3.2 Conjuntos difusos

Las fronteras de un conjunto concreto son exactas, en tanto que las de uno difuso son imprecisos. Esto es, existen elementos en las fronteras mismas que están a la vez dentro y fuera del conjunto. Un conjunto difuso también se puede representar gráficamente, como se muestra en la Figura 2.7. La función de membresía es de forma trapezoidal y consta de un núcleo (N), un soporte (S) y unos límites (L), donde esta definido el conjunto a trabajar; esta función proporciona el grado de pertenencia o verdad de un elemento dentro del conjunto.

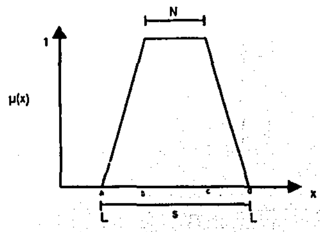


Figura 2.7: Función de Membresía Trapezoidal

La función se define como $f = (x, a, b, c, d)$ y su expresión matemática es:

$$f = (x, a, b, c, d) = \begin{pmatrix} 0 & \text{si } x \leq a \\ x - a/b - a & \text{si } a \leq x \leq b \\ 1 & \text{si } b \leq x \leq c \\ d - x/d - c & \text{si } c \leq x \leq d \end{pmatrix} \tag{2.1}$$

En la Figura 2.7 se muestra que sólo en el núcleo (N) la función de membresía toma el valor de 1. Estas son las principales características de los conjuntos difusos. Otro tipo de representaciones son:

Distribución Gausiana la Figura 2.8. Satisface la ecuación (2.2). Este tipo de distribución es suave y solo en un punto toma el valor de uno.

$$f = (x, \sigma, c) = e^{-(x-c)^2/2\sigma^2} \tag{2.2}$$

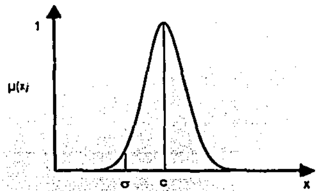


Figura 2.8: Función de distribución Gausiana

Distribución triangular. Satisface la ecuación (2.3) y su forma se muestra en la Figura 2.9.

$$f = (x, a, b, c, d) = \begin{pmatrix} 0 & \text{si } x \leq a \\ x - a/b - a & \text{si } a \leq x \leq b \\ c - x/c - d & \text{si } c \leq x \leq d \\ 0 & \text{si } x \geq d \end{pmatrix} \tag{2.3}$$

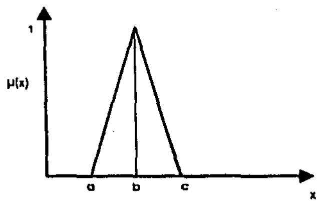


Figura 2.9: Función de distribución triangular

Distribución S. Satisface la ecuación (2.4) y su forma se muestra en la Figura 2.10:

$$f = (x, a, c) = 1/1 - e^{-a(x-c)} \tag{2.4}$$

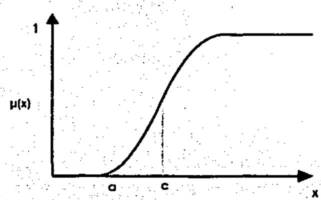


Figura 2.10: Función de distribución S

Distribución Z. Satisface la ecuación (2.5) y su forma se muestra en la Figura 2.11.

$$f = (x, a, b) = \begin{pmatrix} 1 & \text{si } x \leq a \\ x - b/a - b & \text{si } a \leq x \leq b \\ 0 & \text{si } x \geq b \end{pmatrix} \tag{2.5}$$

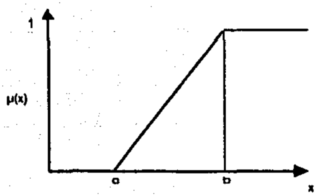


Figura 2.11: Función de distribución Z

Zadeh utiliza la siguiente notación para expresar un conjunto difuso discreto:

$$A = \sum_{i=1}^{n} \mu(x_i)/x_i \tag{2.6}$$

Los conjuntos continuos se representan por medio de la formula:

$$A = \int \mu(x)/x\mathrm{d}x \tag{2.7}$$

Asi mismo, en la lógica aristotélica los conjuntos deben de cumplir con las condiciones:

$$A \cdot A' = \phi \tag{2.8}$$

$$A \cap A' = \phi \tag{2.9}$$

$$A \cup A' = U. \tag{2.10}$$

Para la lógica difusa se emplea una teoría de conjuntos que cumplen con (Martin del Brio and Sanz Molina 2001):

$$A \cdot A' \neq \phi \tag{2.11}$$

$$A \cap A' \neq \phi \tag{2.12}$$

$$A \cup A' = U. \tag{2.13}$$

2.3.3 Teoría de los subconjuntos difusos

En la teoría de los subconjuntos difusos las operaciones de la intersección y de la unión se definen a través de normas y conormas triangulares, respectivamente. Estas funciones fueron introducidas por Schweizer y Sklar (1983) en el estudio de espacios métricos probabilísticos.

Una función T $:[0,1]^2 \longrightarrow [0,1]$ es una norma triangular ($t-norma$) si y sólo si satisface las siguientes propiedades:

1. $T(1,x) = x \quad \forall x \in [0,1]$.
2. $T(y,x) = T(y,x) \quad \forall x,y \in [0,1]$.
3. $T(y,x) \leq T(u,v) \quad \forall x.y,u,v \in [0,1]$ *talque* $x \leq u, y \leq v$.
4. $T(y,T(y,z)) = T(T(x,y),z) \quad \forall x,y,z \in [0,1]$.

Una función S $:[0,1]^2 \longrightarrow [0,1]$ es una conorma triangular ($t-conorma$) si y sólo si satisface las siguientes propiedades:

1. $S(0,x) = x \quad \forall x \in [0,1]$.
2. $S(y,x) = S(y,x) \quad \forall x,y \in [0,1]$.
3. $S(y,x) \leq S(u,v) \quad \forall x,y,u,v \in [0,1]$ *talque* $x \leq u, y \leq v$.
4. $S(y,T(y,z)) = S(S(x,y),z) \quad \forall x,y,z \in [0,1]$.

En este ámbito existen diversas formas de factorizar una relación difusa. Ello es debido a la utilización de diferentes t-normas y tconormas para la representación de la intersección y la unión de conjuntos difusos, así como al empleo de distintos conjuntos (Llamazares 2001).

2.3.4 Operaciones básicas en conjuntos difusos

Las tres operaciones básicas entre conjuntos son: unión, intersección y complemento. Estas se definen también para los conjuntos difusos, intentando mantener el mismo significado. La definición se hace empleando el concepto de función de pertenencia de los conjuntos.

Unión: El resultado de efectuar la operación de unión entre dos conjuntos difusos A y B definidos sobre el mismo universo, y con funciones de pertenencia $\mu_A(x)$ y $\mu_B(x)$ respectivamente, es un nuevo conjunto difuso $A \cup B$ definido sobre el mismo universo, con función de pertenencia $\mu_{A \cup B}(x)$, que esta dada por:

$$\begin{aligned} A \cup B &= \mu_{A \cup B}(x) \\ &= \mu_A(x) \vee \mu_B(x) \end{aligned} \tag{2.14}$$

$$\mu_{A \cup B}(x) = \{x \mid \mu_A(x) \circ \mu_B(x)\} \tag{2.15}$$

$$\mu_{A \cup B}(x) = MAX\ \{\mu_A(x), \mu_B(x)\} \tag{2.16}$$

Las expresiones (2.14) y (2.16) representan la unión de dos conjuntos difusos. Esto es equivalente a la expresión OR de los circuitos lógicos definido en la Tabla 2.9.

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 2.9: Función OR

Siendo la operación unión la suma de los conjuntos (A+B), su representación gráfica se muestra en la Figura 2.12:

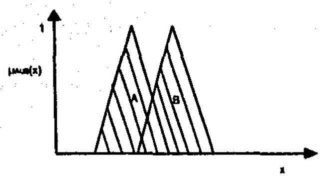


Figura 2.12: Función Unión de A y B

Intersección. El resultado de efectuar la operación intersección entre dos conjuntos difusos A y B definidos sobre el mismo universo, y con funciones de pertenencia $\mu_A(x)$ y $\mu_B(x)$, respectivamente, es un nuevo conjunto difuso $A \cap B$ definido sobre el mismo universo, con función de pertenencia $\mu_{A\cap B}(x)$, dada por:

$$\begin{aligned} A \cap B =& \mu_{A\cap B} \\ =& \mu_A(x) \wedge \mu_B(x) \end{aligned} \tag{2.17}$$

$$\mu_{A\cap B} = \{x \mid \mu_A(x) \quad \text{and} \quad \mu_B(x)\} \tag{2.18}$$

$$\mu_{A\cap B} = MIN\,\{\mu_A(x), \mu_B(x)\} \tag{2.19}$$

Las ecuaciones (2.17) y (2.19) representan la operación "*and*", que es la multiplicación en los circuitos lógicos $A \cdot B$ que cumple con la Tabla 2.10 :

A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 2.10: Función AND

Y su operación para la teoría de conjuntos difusos se representa de la siguiente forma:

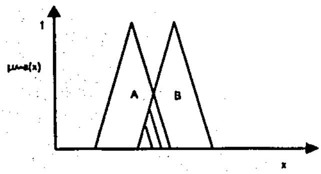


Figura 2.13: Función Intersección de A y B

Complemento. El resultado de efectuar la operación complemento sobre un conjunto difuso A definido sobre un universo, y con función de pertenencia $\mu_A(x)$, es un nuevo conjunto difuso $\bar{A}$ definido sobre el mismo universo, con función de pertenencia $\mu_{\bar{A}}(x)$, dada por:

$$\bar{A} = 1 - \mu_A(x) \tag{2.20}$$

A esta operación se le representa con el complemento en los circuitos lógicos de la Tabla 2.11. La representación grafica se muestra en la Figura 2.14.

A	$\bar{A}$
0	1
1	0

Tabla 2.11: Función complemento

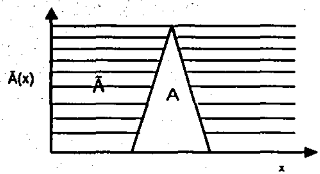


Figura 2.14: Función de distribución Complemento

2.3.5 Relaciones

Una relación es la correspondencia que existe entre dos o más conjuntos. En una relación se pueden dar varios casos.

1)Un miembro de un grupo se relaciona con varios de otro.
2)A varios miembros de un conjunto le corresponde uno de otro.
3)Un miembro de un conjunto se relaciona sólo con un elemento de otros; a esta relación se le conoce como uno a uno.

Para este caso se puede obtener el valor del segundo teniendo el primero y viceversa. A este tipo de relaciones se les conoce también como funciones y solo es posible cuando en los conjuntos existe una relación definida. Por ejemplo la de pertenencias que dice que un elemento solo existe si se encuentra dentro de un conjunto. Es decir, el elemento cumple con las condiciones del conjunto y en consecuencia tendrá el valor de uno o cero en caso contrario. Para el caso de la lógica difusa no se aplica lo anterior, ya que existen un valor de pertenencia que varia entre cero y uno.
Una relación especialmente importante es la composición, esta se puede entender con el siguiente ejemplo.

Suponiendo a R como la relación entre X y Y, y S la relación entre Y y Z, para encontrar la relación entre X y Z, se necesita hacer una composición T, que esta dada por la composición entre R y S:

$$T = R \circ S. \tag{2.21}$$

Para los conjuntos difusos la relación de conjuntos está definida por la función MÁX.-MIN. Es decir, que si un elemento de un conjunto esta relacionado con otro tendrá el valor de 1 y cero de lo contrario. De esta forma se obtiene cada par ordenado el valor mínimo y posteriormente el máximo del par hasta terminar con todas las relaciones. En Figura 2.15 se observan las relaciones que existen entre los conjuntos X,Y y Z:

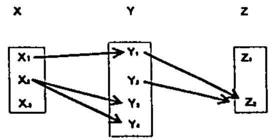


Figura 2.15: Ejemplo de la relación entre los conjuntos X,Y y Z

Para poder encontrar las relaciones R y S se supondrá que:

$$R = \begin{bmatrix} & y_1 & y_2 & y_3 & y_4 \\ x_1 & 1 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 1 & 1 \\ x_3 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.22}$$

$$S = \begin{bmatrix} & z_1 & z_2 \\ y_1 & 0 & 1 \\ y_2 & 0 & 1 \\ y_3 & 0 & 0 \\ y_4 & 0 & 0 \end{bmatrix} \tag{2.23}$$

Entonces la relación T en (2.21), utiliza la operación MÁX.-MIN, que está dada por:

$$X_T = (X_1, Z_1) = MAX\left\{\min(X_R(x,y), X_S(y,z))\right\} \tag{2.24}$$

Para el caso de $X_T = (X_1, Z_1)$ se tiene que:

$$X_T = (X_1, Z_1) = MAX\left\{\min((x_1,y_1),(y_1,z_1))\cdots,\min((x_1,y_4),(y_4,z_1))\right\} \tag{2.25}$$

$$X_T = (X_1, Z_1) = MAX\left\{\min(1,0),\min(0,0),\min(0,0),\min(0,0)\right\} \tag{2.26}$$

$$X_T = (X_1, Z_1) = MAX\{0,0,0,0\} \tag{2.27}$$

$$X_T = (X_1, Z_1) = 0 \tag{2.28}$$

Siguiendo el mismo procedimiento para los demás elementos se obtiene:

$$T = R \circ S = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{cc} z_1 & z_2 \\ \left[\begin{array}{c} 0 \\ 0 \\ 0 \end{array}\right. & \left.\begin{array}{c} 1 \\ 1 \\ 0 \end{array}\right] \end{array} \tag{2.29}$$

2.3.6 Fusificación

La fusificación consiste en convertir un valor real a un valor difuso que representará el grado de pertenecía de los diferentes conjuntos difusos. Para definir a estos existen deferentes métodos, como son:

Por Intuición: En este método se utiliza el conocimiento intuitivo del experto. Un ejemplo es la temperatura ambiente de una habitación como puede ser.

-Una temperatura muy fría es menor a los 5^oC.

-Una temperatura fría es a partir de los 3^oC hasta 18^oC

-Una temperatura templada es de 15^oC a 25^oC

-Una temperatura caliente va desde los 20^oC

Cada una de las variables de entrada y salida tiene una representación dentro del sistema de lógica difusa en forma de variables lingüísticas. Una variable lingüística tiene, entre otras cosas, una colección de atributos que puede adquirir la variable, cada uno representado por un conjunto difuso. Tomando el ejemplo de la Figura 2.16, la variable temperatura tiene cuatro atributos: **muy fría, fría, templada y caliente**. Cada uno de estos atributos está representado por el conjunto difuso respectivo. Estos atributos reciben el nombre de valores lingüísticos.

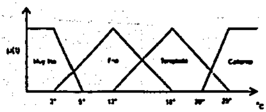


Figura 2.16: Conjuntos difusos de temperatura por intuición

Por Inferencia: Para este método se parte de un conocimiento previo (Martin del Brio and Sanz Molina 2001). Por ejemplo, se sabe que el punto de congelación del agua es de 0^oC y que esta comienza a descongelarse a los 4^oC mientras que el punto de ebullición es de 100^oC. De este modo se puede definir los conjuntos difusos como se muestra en la Figura 2.17

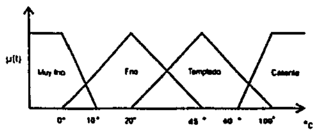


Figura 2.17: Conjuntos difusos de temperatura por inferencia

Se recomienda que para cualquiera de los métodos empleados se utilicen conjuntos impares y no menos de tres. Lo más recomendable son siete y además para la función de membresía $\mu_A(x)$ el punto de cruce debe ser menor o igual que 0.5 (Jamshidi 1993).

2.3.7 Inferencia Difusa:

La inferencia difusa consiste en la combinación de proposiciones para producir nuevas proposiciones. Por ejemplo, al combinar la proposición *X es A* con la proposición *Si X es A entonces Y es B*, se puede inferir la proposición *Y es B*. Una inferencia como la anterior sólo es posible en la lógica tradicional si la proposición *X es A* es idénticamente igual. Sin embargo, en la lógica difusa estas dos proposiciones no necesariamente deben ser idénticas, debido a que las fronteras de los conjuntos no son precisas. Por ello, al combinar la proposición *X es* A^* con la proposición *Si X es A entonces Y es B*, puede obtenerse la proposición *Y es* B^*.

La estructura básica de un sistema de lógica difusa se muestra en la Figura 2.18, ya que el sistema recibe y entrega variables numéricas. El bloque difuso se encarga de convertir las entradas en conjuntos difusos, que son entregados al bloque llamado máquina de inferencia. Este bloque consta de un conjunto de reglas de la forma *Si... entonces...* que se encuentran almacenadas en su memoria, son la base de las reglas que producen los conjuntos difusos para su posterior Defuzzificación.

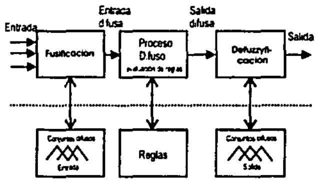


Figura 2.18: Máquina de inferencia difusa

Debido a que un sistema de lógica difusa puede en general, tener varias entradas y salidas, la forma genérica de las reglas presentes en la base de reglas es la siguiente :

$$\begin{array}{l} Si\ x_1 \quad \text{es } a_1 \quad \text{y } x_2 \quad \text{es } a_2 \quad \text{y} \ldots \text{ y } x_m \quad \text{es } a_m \\ entonces\ y_1 \quad \text{es } b_1 \quad \text{y } y_2 \quad \text{es } b_1 \text{ y} \ldots \text{ y } \quad y_n \quad \text{es } b_n \end{array}$$

En estas reglas, a_1,a_2,..., a_m, b_1,b_2,...,b_m son valores de las variables lingüísticas respectivas.

2.3.8 Defuzzificación:

El resultado de salida en la máquina difusa debe ser un número real para poder ser aplicado al sistema. Como se describió en la sección anterior, al conjunto de variables de la entrada en el sistema difuso se le asigna un valor difuso para aplicar el conjunto de reglas difusas. Esto da un resultado difuso, que es un grado de pertenencia de la variable de salida. Para poder obtener un valor real se utiliza la Defuzzificación que se puede llevar a cabo por diversos métodos:

Altura máxima o máxima función de pertenencia En este método se toma el máximo valor de membresía $\mu(x)$, y es valido para conjuntos con un solo máximo Figura (2.19).

$$X^* = MAX\mu(x) \tag{2.30}$$

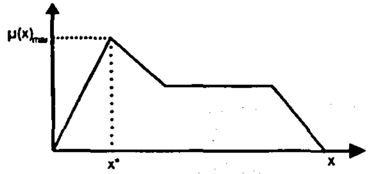


Figura 2.19: Método del Máximo

Centroide Se puede aplicar a cualquier conjunto, pero requiere de un proceso computacional más , Figura (2.20). Su expresión matemática está dada por: :

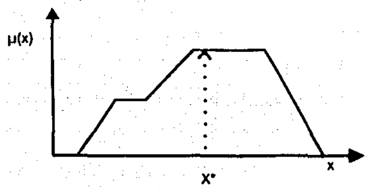


Figura 2.20: Método del centroide

$$X^* = \frac{\int \mu(x) * x dx}{\int \mu(x) dx} \tag{2.31}$$

Promedio pesado Se obtiene al calcular los centroides, de acuerdo con:

$$X^* = \frac{\sum \mu(x) * \bar{x}}{\sum \mu(x)} \tag{2.32}$$

Su representación se muestra an la Figura 2.21:

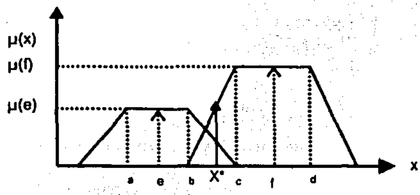


Figura 2.21: Promedio pesado

Singleton Este método es el más utilizado ya que es muy fácil de programar y muy eficiente; además, simplifica considerablemente el proceso de inferencia y posibilita la implementación electrónica eficiente de los sistemas de inferencia difusos.

$$X^* = \frac{\sum \mu(x_i) * x_i}{\sum \mu(x_i)} \tag{2.33}$$

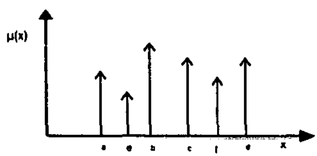


Figura 2.22: Singleton

2.3.9 Reglas difusas

Los conjuntos y operadores difusos son los sujetos y predicados de la lógica difusa. Las reglas *si-entonces* son usadas para formular las expresiones condicionales que abarca la lógica difusa *si x es A entonces y es B.* A y B son los valores lingüísticos definidos por los conjuntos en los rangos de los universos de discurso llamados X e Y, respectivamente. La parte *si* de la regla *x es A* es llamada el antecedente o premisa, mientras la parte *entonces* de la regla *y es B* es llamada la consecuencia o conclusión (Kosko 1999)

Se usan las reglas para representar las relaciones entre las variables lingüísticas, de entrada y salida. Estas pueden ser del tipo Mandami, que se expresan como se muestra en la ecuación (2.34), donde x_n es la variable lingüística que pertenece a un conjunto F_n^1 (Martin del Brio and Sanz Molina 2001):

$$R^{(1)}: Si \quad x_1 \quad es \quad F_1^1 \quad y........y \quad x_n \quad es \quad F_n^1 \quad entonces \quad y \quad es \quad G^1 \tag{2.34}$$

Estas reglas utilizan el conocimiento previo del sistema. Por otra parte las del tipo Sugeno que simplifican los cálculos de la salida pero no expresan el conocimiento de los expertos, esta se expresada por (Martin del Brio and Sanz Molina 2001):

$$R^{(1)}: Si \quad x_1 \quad es \quad F_1^1 \quad y........y \quad x_n \quad es \quad F_n^1 \quad entonces \quad y = F^1(x) \tag{2.35}$$

Las reglas son un conjunto de juicios que tienen un carácter de asignación condicional o incondicional. Un ejemplo es el ya mencionado : *Si el agua tiene una temperatura de 4°C entonces está fría.* Las reglas incondicionales se aplicará sin preguntar: por ejemplo *calienta el agua.* La lógica difusa condicional, consiste en una condición (*Si* $\rightarrow$ *cláusula*), y una conclusión (*entonces* $\rightarrow$ *cláusula*). Las cláusulas condicionales pueden consistir en una o más condiciones ligadas entre si por "*y*" u "*o*". La base de las reglas se puede representar por una tabla llamada Memoria Asociativa Difusa *FAM* (por sus siglas en ingles) (Kosko 1999). Las FAM son matrices que representan la combinación de cada una de las reglas definidas de las variables de entrada con las de salida. Este método se puede representar gráficamente (Kosko 1999) para ello, las relaciones entre las variables de entrada y salida requieren que se indiquen explícitamente las reglas; al computo asociado se le conoce como inferencia de las reglas difusas. La inferencia es un cálculo que consiste en dos pasos. El primero es la agregación, que determina la condición de la regla, (*Si* $\rightarrow$ *cláusula es satisfecha*). Con este propósito se emplean operadores para determinar los grados de validez, un ejemplo es:

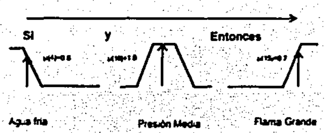


Figura 2.23: Relación Si → entonces

Si temperatura es baja y la presión es media *entonces* la flama es grande

o bien:

Si temperatura = baja *y* presión = media *entonces* flama=grande

En el ejemplo de la Figura 2.23 se tienen dos condiciones, por lo que se debe de observar el grado de validez o el valor de la función de membresía de cada condición. Esto se puede obtener con la ayuda de los conjuntos difusos ya mencionados. Por ejemplo, utilizando la ecuación (2.5) se obtiene el grado de verdad. De este modo, para *agua = fría*, el grado de verdad es de 0.8 y para *presión = media* el grado de verdad es de 1.0. La conjunción Y representada por las ecuaciones (2.17),(2.18) y (2.19), indica que hay que obtener el mínimo de los dos conjuntos de membresía, lo que proporciona un grado de validez a la condición. Considerando las ecuaciones ya mencionadas, se toma el mínimo de los dos conjuntos agua fría y presión media (Beale 1994). El segundo paso es el cálculo de la composición, que usa la validez de las ecuaciones de los conjuntos difusos para determinar la regla. Para el ejemplo se emplea la ecuación (2.24) y se toma directamente porque sólo cuenta con una regla. En caso contrario se tendrían que obtener los máximos de todos los mínimos de las reglas que cumplan con *flama grande*. Es decir, lo que se está realizando es la aplicación de MÁX.-MIN. Otra forma de crear las reglas es por medio de un mapa de asociación difusa: *FAM* (King 1999).

Capítulo 3

Diseño de controles difusos

En este capítulo se presentan los cuatro controladores implementados en el sistema de robots A255 y A465: El primero es un control PID clásico utilizando el método de Ziegler y Nichols para sintonización de ganancias; el segundo es un control $PIDF$ (Control PID difuso), que consta de un control difuso en serie con un control PID clásico, el tercero es $FPIDF$ (Control PID flexible difuso), que consta de un control PID clásico en serie con retroalimentación; el último es un control difuso utilizando el método de síntesis de Lyapunov.

3.1 Control PID utilizando el método de ajuste de Ziegler-Nichols (Z-N)

El método de ajuste de ganancias de Ziegler-Nichols que consiste en un control proporcional de lazo cerrado, como se muestra en la Figura 3.1 (Åström and Wittenmark 1989).

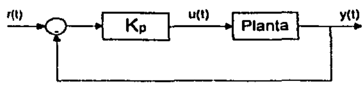


Figura 3.1: Sistema en lazo cerrado con ganancia proporcional

Con un escalón como referencia y un control proporcional como el de la Figura 3.1, se comienza a incrementar K_p hasta que el sistema oscila de manera periódica. Entonces se obtiene la ganancia crítica $K_c = K_p$ y un periodo de oscilación T_c a la salida de la planta, como se muestra el de la Figura 3.2.

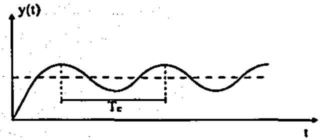


Figura 3.2: Respuesta de la planta a lazo cerrado con ganancia crítica

Con K_c y T_c se ajustan las ganancias del controlador PID, de acuerdo con la Tabla 3.1 (Åström and Wittenmark 1989).

	K_p	T_n	T_v
P	$0.5K_c$	–	–
PI	$0.45K_c$	$0.85T_c$	–
PID	$0.6K_c$	$0.5T_c$	$0.12T_c$

Tabla 3.1: Ajuste de parámetros

La formula del controlador es:

$$u(t) = K_p \left(e(t) + \frac{1}{T_n} \int_0^t e(\tau) \mathrm{d}t + T_v \dot{e}(t) \right), \tag{3.1}$$

o bien

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) \mathrm{d}t + K_d \dot{e}(t), \tag{3.2}$$

donde

$$K_i = \frac{K_p}{T_n}, \qquad K_d = K_p T_v. \tag{3.3}$$

3.2 Control PIDF

El control PID difuso propuesto por Tao and Taur (2000) se compone de un controlador PID conectado en cascada con un mecanismo difuso. Su complejidad está en función del número de reglas, ya que entre más variables existan a la entrada del sistema difuso estas crecerán de forma exponencial. Para reducir la complejidad del control se tomará una sola variable a la entrada del mecanismo difuso, que es la salida del control convencional PID. En la Figura 3.3 se muestra el diagrama esquemático del control PID difuso (PIDF), donde la variable $x(t)$ es la suma del error $e(t)$, la derivada del error $\dot{e}(t)$ y la integral del error $\int_0^t e dt$:

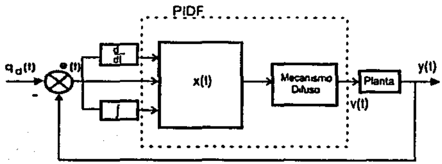


Figura 3.3: Control PIDF

Para el diseño del control $PIDF$ se utilizó la ecuación (3.1) de la manera siguiente

$$x(t) = \left(e(t) + \frac{1}{T_n} \int_0^t e(\tau) \mathrm{d}t + T_v \dot{e}(t) \right) \quad (3.4)$$

$$v(t) = K_p x(t) \quad (3.5)$$

La variable de entrada al mecanismo difuso de cada articulación de los robots es $x_i(t)$ y la salida $u_i(t)$ con $i = 1, 2, 3$ para cada manipulador. Estos conjuntos se dividieron en siete conjuntos difusos. Para el caso de la entrada se tomó: **Negativo Grande** ng, **Negativo Mediano** nm, **Negativo Pequeño** np, **Cero** ze, **Positivo Pequeño** pp, **Positivo Mediano** pm y **Positivo Grande** pg. Es importante mencionar que las funciones pueden ser de varias formas, como se describió en el capítulo anterior. En la Figura 3.4 se muestran los conjuntos a la entrada que se utilizaron.

Para la variable de salida $u(t)$ del controlador $PIDF$ se consideró la misma nomenclatura de los conjuntos de entrada, pero con una función de pertenencia del tipo Singleton (ver Figura 3.5).

Para el caso de las reglas del conjunto difuso se consideraron la siguientes: La j-ésima regla difusa *SI-ENTONCES* se define como.

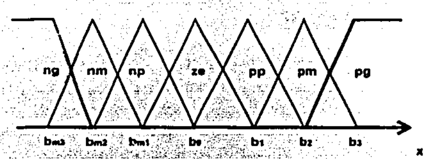


Figura 3.4: Función de pertenencia de los conjuntos de entrada x_i(t)

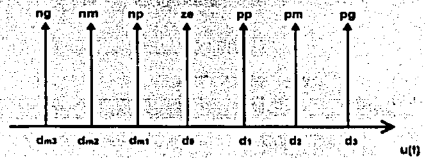


Figura 3.5: Función de pertenencia de la variable de salida u_i(t)

$$R_j : Si \quad (x_i \quad es \quad A_j) \quad entonces \quad (u_i \quad es \quad C_j) \tag{3.6}$$

donde A_j y C_j, son los siete conjuntos de las variables de entrada $x_i(t)$ y de la salida $u_i(t)$. Estas reglas son sencillas, ya que existe una relación biunívoca entre la salida y la entrada. En caso del conjunto difuso de salida se consideró la del tipo Singleton y se utilizó el método del centroide para la Defuzzificación empleando la ecuación (3.7).

$$u_i = \frac{\sum_{j=-3}^{3} d_j f_j(x_i)}{\sum_{j=-3}^{3} f_j(x_i)}, \tag{3.7}$$

donde d_j y f_j son las funciones de pertenencia de las variables de entrada y salida respectivamente. En este caso, el propósito del algoritmo de control es adaptar la ganancia K_p a través del control difuso.

3.3 Control FPIDF

El control flexible PID difuso $FPIDF$ también propuesto por Tao and Taur (2000), es similar al control $PIDF$. En este caso se escalan las ganancias K_{d} y K_{i} para tener un mejor desempeño. Esto se logra haciendo que las ganancias estén en función de la salida del mecanismo difuso $u(t)$. En la Figura 3.6 se muestra el esquema del controlador $FPIDF$.

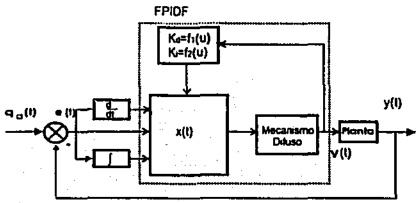


Figura 3.6: Control FPIDF

Utilizando las ecuaciones (3.4) y (3.5) se tiene que:

$$x(t) = \left(e(t) + K_{\mathrm{i}}' \int_0^t e(\tau)\mathrm{d}t + K_{\mathrm{d}}'\dot{e}(t)\right), \tag{3.8}$$

$$v(t) = K_{\mathrm{p}}x(t) \tag{3.9}$$

donde las ganancias K_{d}' y K_{i}' cumplen con la condición

$$K_{\mathrm{D}}' = f(u) = \begin{cases} \frac{\alpha_1}{|u|} & \text{si } |u| \geq \beta_1 \\ \frac{\alpha_1}{\beta_1} & \text{otro caso} \end{cases}$$

$$K_{\mathrm{I}}' = f(u) = \begin{cases} \frac{\alpha_2}{|u|} & \text{si } |u| \geq \beta_2 \\ \frac{\alpha_2}{\beta_2} & \text{otro caso} \end{cases}$$

Donde α_1 y α_2 son factores constantes y β_1 y β_2 son constantes positivas.
Se utilizó la misma nomenclatura de los conjuntos difusos del control $PIDF$: **Negativo Grande** ng, **Negativo Mediano** nm, **Negativo Pequeño** np, **Cero** ze, **Positivo Pequeño** pp, **Positivo Mediano** pm y **Positivo Grande** pg.

En el caso de la defuzzycación se consideró nuevamente el método Singleton:

$$u_i = \frac{\sum_{j=-3}^{3} d_j f_j(x_i)}{\sum_{j=-3}^{3} f_j(x_i)} \tag{3.10}$$

Sin embargo, hay que notar que la salida $u(t)$ modifica las ganancias del control, K_d y K_I, que dependen de las condiciones ya mencionadas.

3.4 Control por el Método de Síntesis de Lyapunov Difuso

Los dos controladores antes mencionados se basan en la experiencia obtenida por el controlador PID convencional. En esta sección se estudiará un controlador difuso cuyo diseño se basa en el Método de Síntesis de Lyapunov Difuso (Margaliot and Langholz 2000). En la Figura 3.7 se muestra en diagrama esquemático del control. En el caso del control PD se utilizó la sintonización de la Tabla 3.1, tomando como cero la parte integrativa.

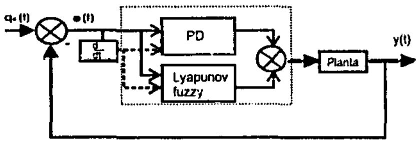


Figura 3.7: Control por Síntesis de Lyapunov

Para emplear este método no es necesario obtener el modelo matemático exacto del robot, si no una aproximación. Para el diseño del controlador difuso se toma cada articulación como un motor, cuyo modelo más simple se puede escribir como:

$$G_i(s) = \frac{K_i}{s(s+a_i)} = \frac{Q_i(s)}{V_i(s)} \tag{3.11}$$

donde K_i y a_i son constantes del motor, $Q_i(s)$ es el ángulo de salida y $V_i(s)$ es el voltaje de entrada. Transformando (3.11) al dominio del tiempo se obtiene:

$$\ddot{q}_i + a\dot{q}_i = K_i V_i, \tag{3.12}$$

donde q_i es la posición de motor y V_i es el voltaje de control. La ecuación (3.12), es el caso ideal, una mejor aproximación del sistema real es:

$$\ddot{q}_i + a_i\dot{q}_i = K_i V_i + p_i'(t) \tag{3.13}$$

donde $p_i'(t)$ es una perturbación del sistema cuya dinámica no esta modelada. Para poder trabajar con ella se hace la siguiente suposición:

$$\|p_i'(t)\| \leq C_{0i} \qquad \forall t \tag{3.14}$$

donde C_{0i} es una constante positiva. Definiendo q_{di} como la posición deseada, ahora para no alterar la ecuación restamos y sumamos q_{di}, cuya primera y segunda derivada estan acotadas, (3.13), se puede reescribir como:

$$\begin{aligned} \ddot{q}_i - \ddot{q}_{di} + a_i\dot{q}_i - a_i\dot{q}_{di} + \ddot{q}_{di} + a_i\dot{q}_{di} &= K_i V_i + p_i'(t) \\ (\ddot{q}_i - \ddot{q}_{di}) + a_i(\dot{q}_i - \dot{q}_{di}) + \ddot{q}_{di} + a_i\dot{q}_{di} &= K_i V_i + p_i'(t) \\ \ddot{\tilde{q}}_i + a_i\dot{\tilde{q}}_i &= K_i V_i + p_i'(t) - \ddot{q}_{di} - a_i\dot{q}_{di}, \end{aligned} \tag{3.15}$$

donde $\tilde{q}_i$ es el error de seguimiento. Cosinderando un voltaje de control difuso V_{fi} generado por el control difuso. La ecuación (3.15) queda:

$$\ddot{\tilde{q}}_i + a_i\dot{\tilde{q}}_i = K_i V_{fi} + p_i'(t) - \dot{q}_{di} - a_i\dot{q}_{di} \tag{3.16}$$

Para el diseño del controlador difuso se propone la siguiente función candidata de Lyapunov (Margaliot and Langholz 2000):

$$V_i = \frac{1}{2}(\tilde{q}_i^2 + \dot{\tilde{q}}_i^2) \tag{3.17}$$

Derivando la ecuación (3.17) se tiene:

$$\dot{V}_i = \tilde{q}_i\dot{\tilde{q}}_i + \dot{\tilde{q}}_i\ddot{\tilde{q}}_i \tag{3.18}$$

Despejando $\ddot{\tilde{q}}_i$ de (3.16):

$$\ddot{\tilde{q}}_i = K_i V_{fi} + p_i'(t) - a_i\dot{\tilde{q}}_i - \ddot{q}_{di} - a_i\dot{q}_{di} \tag{3.19}$$

Sustituyendo en (3.18):

$$\dot{V}_i = \tilde{q}_i\dot{\tilde{q}}_i + \dot{\tilde{q}}_i(K_iV_{fi} + p_i'(t) - a_i\dot{\tilde{q}}_i - \ddot{q}_{di} - a_i\dot{q}_{di}) \tag{3.20}$$

Proponiendo también

$$p_i''(t) \triangleq \ddot{q}_{di} + a_i\dot{q}_{di}.$$

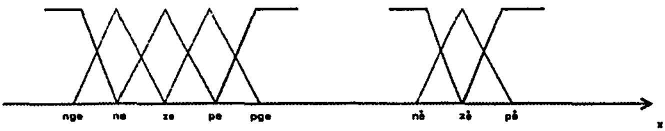
Entonces la ecuación (3.20) se puede escribir como:

$$\begin{aligned} \dot{V}_i &= \tilde{q}_i\dot{\tilde{q}}_i + \dot{\tilde{q}}_i(K_iV_{fi} + p_i'(t) - a_i\dot{\tilde{q}}_i - p_i''(t)) \\ &= \tilde{q}_i\dot{\tilde{q}}_i + \dot{\tilde{q}}_iK_iV_{fi} + \dot{\tilde{q}}_ip_i'(t) - a_i\dot{\tilde{q}}_i^{\,2} - \dot{\tilde{q}}_ip_i''(t) \end{aligned} \tag{3.21}$$

Para asegurar la estabilidad del punto de equilibrio se debe cumplir que $\dot{V}_i \leq 0$ (Margaliot and Langholz 2000). Como se puede observar de las ecuaciones anteriores. los dos primeros terminos, $\tilde{q}_i$ y $\dot{\tilde{q}}_i$, ayudan a la estabilidad del sistema solo cuando son de signo contrario, para $p'(t)$ y $p''(t)$, en el peor de los casos se tomarán el valor absoluto. Entonces las estabilidad del sistema depende de V_{fi}. Por este motivo el sistema debe cumplir con la siguiente condición.

$$\dot{V} = \tilde{q}_i\dot{\tilde{q}}_i + \dot{\tilde{q}}_iK_iV_{fi} + \dot{\tilde{q}}_i \mid p_i'(t) \mid -a_i\dot{\tilde{q}}_i^{\,2} - \dot{\tilde{q}}_i \mid p_i''(t) \mid \leq 0. \tag{3.22}$$

En adelante se supondra que $\mid p'(t) \mid$ y $\mid p''(t) \mid$ estan acotadas para todo el tiempo. Expresando la ecuación anterior en forma lingüística y proponiendo cinco conjuntos difusos para el error $\tilde{q}_i$ **(Negativo Grande, Negativo, Cero, Positivo, Positivo Grande)**, tres para la velocidad del error $\dot{\tilde{q}}_i$ **(Negativo, Cero, Positivo)** y cinco para V_f **(Grande Negativa, Negativa, Cero, Positiva, Positiva grande)** se obtienen las reglas difusas de la Tabla 3.2. Considerando las reglas anteriores, se puede proponer las reglas que se implementarán en el control, de acuerdo con la Figura 3.8.



Conjunto difusos del error $\tilde{q}_i$ Conjunto difusos de $\dot{\tilde{q}}_i$

Figura 3.8: Conjuntos difusos del error y derivada del error.

Si	$\tilde{q}_i$ es	Negativo Grande Y $\dot{\tilde{q}}_i$ Negativo	*entonces* V_{fi}	debe ser Positivo Grande
Si	$\tilde{q}_i$ es	Negativo Grande Y $\dot{\tilde{q}}_i$ Cero	*entonces* V_{fi}	debe ser Positivo Grande
Si	$\tilde{q}_i$ es	Negativo Grande Y $\dot{\tilde{q}}_i$ Positivo	*entonces* V_{fi}	debe ser Positivo
Si	$\tilde{q}_i$ es	Negativo Y $\dot{\tilde{q}}_i$ Negativo	*entonces* V_{fi}	debe ser Positivo Grande
Si	$\tilde{q}_i$ es	Negativo Y $\dot{\tilde{q}}_i$ Cero	*entonces* V_{fi}	debe ser Positivo
Si	$\tilde{q}_i$ es	Negativo Y $\dot{\tilde{q}}_i$ Positivo	*entonces* V_{fi}	debe ser Cero
Si	$\tilde{q}_i$ es	Cero Y $\dot{\tilde{q}}_i$ Negativo	*entonces* V_{fi}	debe ser Positivo
Si	$\tilde{q}_i$ es	Cero Y $\dot{\tilde{q}}_i$ Cero	*entonces* V_{fi}	debe ser Cero
Si	$\tilde{q}_i$ es	Cero Y $\dot{\tilde{q}}_i$ Positivo	*entonces* V_{fi}	debe ser Negativo
Si	$\tilde{q}_i$ es	Positivo Y $\dot{\tilde{q}}_i$ Negativo	*entonces* V_{fi}	debe ser Cero
Si	$\tilde{q}_i$ es	Positivo Y $\dot{\tilde{q}}_i$ Cero	*entonces* V_{fi}	debe ser Negativo
Si	$\tilde{q}_i$ es	Positivo Y $\dot{\tilde{q}}_i$ Positivo	*entonces* V_{fi}	debe ser Negativo Grande
Si	$\tilde{q}_i$ es	Positivo Grande Y $\dot{\tilde{q}}_i$ Negativo	*entonces* V_{fi}	debe ser Negativo
Si	$\tilde{q}_i$ es	Positivo Grande Y $\dot{\tilde{q}}_i$ Cero	*entonces* V_{fi}	debe ser Negativo Grande
Si	$\tilde{q}_i$ es	Positivo Grande Y $\dot{\tilde{q}}_i$ Positivo	*entonces* V_{fi}	debe ser Negativo Grande

Tabla 3.2: Reglas del controlador de Síntesis de Lyapunov Difuso.

De esta forma se obtene un conjunto de reglas para cada uno de los grados de libertad. Para el caso de la salida del control de síntesis de Lyapunov, se empleo el método de singleton para facilidad de programación.

Capítulo 4

Resultados experimentales

En este capítulo se presenta la validación de los controladores, también se muestran los resultados obtenidos en cada uno de los ejes de los robots $A255$ y $A465$, así como una comparación de estos.

4.1 Lenguaje de programación

El software con el cual se programaron los algoritmos de control es LabWindows/CVI que admite programación estructurada basada en un lenguaje abierto y versátil.

4.1.1 LabWindows/CVI y Software FlexMotion

LabWindows/CVI es un ambiente de desarrollo de software para programadores de C que se puede utilizar para: a) desarrollar programas interactivos, b) acceder a librerías de funciones para crear aplicaciones de control de instrumentos y adquisición de datos, c) tomar ventaja de un amplio conjunto de herramientas para adquisición, análisis y presentación de datos.

Los programas escritos en el ambiente de *LabWindows/CVI* deben respetar las especificaciones del ANSI C. Además, se pueden utilizar libremente los módulos de objeto compilados de C, las librerías de enlace dinámico (DLLs), las librerías de C, y los drivers de instrumentos conjuntamente con archivos fuente del ANSI C cuando se desarrollan los programas.

El poder de *LabWindows/CVI* radica en sus librerías. Estas tienen funciones para desarrollar todas las fases del sistema de adquisición de datos y de control de instrumentos. Debido a que *LabWindows/CVI* es flexible, se puede hacer casi cualquier planteamiento para construir un programa. La mayoría de los proyectos

que se construyen incluyen los elementos siguientes: una interfaz de usuario, un programa de control, la adquisición de datos y el análisis de datos. La Figura 4.1 muestra de forma general los elementos del programa. El programa de control recibe la entrada de la interfaz de usuario, de los elementos de adquisición y de análisis de datos.

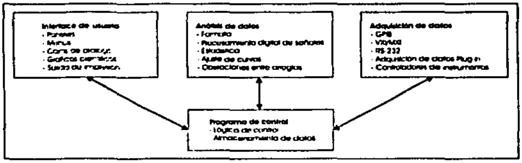


Figura 4.1: Relación entre los elementos de un programa en LabWindows/CVI.

Por otra parte el software FlexMotion es una poderosa interfaz de programación entre la aplicación de control de movimiento y las tarjetas *FlexMotion* de *National Instruments* para computadoras con buses ISA y PCI. El software *FlexMotion* proporciona funcionalidad y poder para los sistemas integrados de movimiento para su uso en laboratorios y ambientes de producción.

El software *FlexMotion* incluye una utilería de configuración para la tarjeta de movimiento. Esta utilería asocia direcciones físicas del bus con las identificaciones de la tarjeta. También es utilizado para verificar que la tarjeta *FlexMotion* está instalada correctamente y se está comunicando con la computadora maestra. Las funciones que constituyen la librería del software *FlexMotion* se pueden llamar desde *LabWindows/CVI*.

4.2 Control PID

El controlador se programó de acuerdo con la ecuación (4.1) utilizando el método de Ziegler-Nichols. Con base en la Tabla 3.1 se obtienen las ganancias K_{pi}, K_{ii} y K_{di}, quedando de la forma:

$$u_i(t) = K_{pi} e_i(t) + K_{ii} \int_0^t e_i(\tau) \mathrm{d}t + K_{di} \dot{e}_i(t). \tag{4.1}$$

Este método se utilizó en cada una de las articulaciones de los robots. En la tabla siguiente se muestran las ganancias del control *PID* de los robots *A465* y *A255*.

Sin embargo, los resultados no fueron del todo satisfactorios por lo que se tuvo que mejorar las ganancias proporcional por el método de prueba y error.

	A465 ARTICULACIONES			A255 ARTICULACIONES		
	1	2	3	1	2	3
K_P	42	48	72	72	108	36
K_I	1.41	1.38	2.21	1.468	1.75	2.16
K_d	300	400	562.5	847	1600	144

Tabla 4.1: Ganancias del controlador PID.

El experimento que se realizó consistió en programar una trayectoria suave de quinto orden en cada una de las articulaciones del robot, mostradas en el Capítulo 2.

Los resultados del error obtenido al implementar el controlador PID se muestran en las graficas de las Figura 4.5.a hasta la Figura 4.10.a. Como se logra ver el error promedio obtenido es de 0.2 grados. Y la salida en promedio de los controladores es de 1.5 volt. (ver Figura 4.11.a hasta Figura 4.16.a).

4.3 Control PIDF

En el caso del control $PIDF$ se consideró la ecuación (4.2), para variar K_p hasta obtener un bues resultado con el controlador:

$$x_i(t) = \left(e_i(t) + \frac{1}{T_{ni}} \int_0^t e_i(\tau)\mathrm{d}t + T_{vi}\dot{e}_i(t)\right) \tag{4.2}$$

$$u_i(t) = K_{pi} x_i(t) \tag{4.3}$$

En este caso la entrada al mecanismo difuso es $x_i(t)$, con $\frac{K_p}{T_n}$ y K_pT_v constantes. Siendo $x_i(t)$ el universo de entrada; esté universo se tomo de la forma triangular para facilitar la programación, quedando de la siguiente manera:

Cabe hacer notar que cada articulación tiene su conjunto difuso específico. Los valores de los conjuntos de membresía de la entrada de cada una de las articulaciones se diseñaron con las reglas definidas por la FAM, utilizando las nomenclatura siguientes :

Para la entrada $x_i(t)$: **Negativo Grande** (ng), **Negativo Mediano** (nm), **Negativo Pequeño** (np), **Cero** (ze), **Positivo Pequeño** (pp), **Positivo Mediano** (pm) y **Positivo Grande** (pg). A la salida $u_i(t)$ se consideró para cada

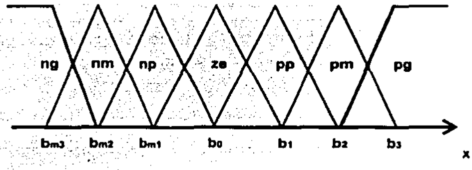


Figura 4.2: Función de pertenencia de la variable de entrada al control PIDF

articulación la misma nomenclatura. Entonces se tiene la siguiente *FAM*

$x_i(t)$	ng_i	nm_i	np_i	ze_i	pp_i	pm_i	pg_i
$u_i(t)$	ng_i	nm_i	np_i	ze_i	pp_i	pm_i	pg_i

Tabla 4.2: *FAM* del control PIDF.

Se utilizó el método de Mandami para programar las reglas, ya que permite expresar el conocimiento obtenido con el control *PID* sobre el sistema. Obteniendo los valores de los conjuntos de la entrada y salida de la siguiente forma.

Valores de los conjuntos de membresía de entrada del control PIDF

En la Tabla 4.3 se muestran los valores de cada uno de los conjuntos de entrada para cada una de las articulaciones. Donde los limites de los conjuntos de membresía esta definidos por b_{m_3} b_{m_2} b_{m_1} b_0 b_1 b_2 b_3, para su programación:

Robot A465	b_{m_3}	b_{m_2}	b_{m_1}	b_0	b_1	b_2	b_3
Articulación 1	-1	-0.1	-0.05	0	0.05	0.1	1
Articulación 2	-2.3	-0.23	-0.115	0	0.115	0.23	2.3
Articulación 3	-1	-0.1	-0.05	0	0.05	0.1	1
Robot A255							
Articulación 1	-1	-0.1	-0.05	0	0.05	0.1	1
Articulación 2	-1	-0.1	-0.05	0	0.05	0.1	1
Articulación 3	-0.8	-0.064	-0.04	0	0.04	0.064	0.8

Tabla 4.3: Parámetros de los conjuntos de entrada $x_i(t)$ del control PIDF.

Para la Defuzzificación se empleó el método Singleton expresado por la ecuación (3.7), por ser un método rápido y eficiente quedando de la forma mostrada en la Figura 4.3.

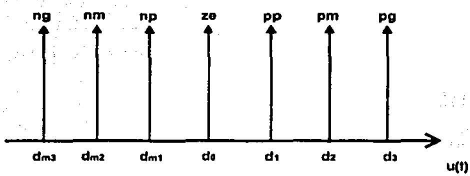


Figura 4.3: Función de pertenencia de la variable de salida $u_i(t)$

Valores de los conjuntos de membresía de salida del controlador ***PIDF***
La tabla de salida esta dada por:

Robot A465	d_{m_3}	d_{m_2}	d_{m_1}	d_0	d_1	d_2	d_3
Articulación 1	-0.5	-0.475	-0.45	0	0.45	0.475	0.5
Articulación 2	-2.5	-2.37	-2.25	0	2.25	2.37	2.5
Articulación 3	-2.5	-1.8	-1.75	0	1.75	1.8	2.5
Robot A255							
Articulación 1	-1.5	-1.25	-1.05	0	1.05	1.25	1.5
Articulación 2	-2.75	-2.62	-1.92	0	1.92	2.62	2.75
Articulación 3	-2.6	-0.13	-0.52	0	0.52	0.13	2.6

Tabla 4.4: Parámetros de los conjuntos de salida $u_i(t)$ del control PIDF.

Donde los limites de los conjuntos de membresía esta definidos por d_{m_3} d_{m_2} d_{m_1} d_0 d_1 d_2 d_3 , para su programación:

Las ganancias para las que se obtuvieron los mejores resultados son las mostra en la Tabla 4.5:

Los resultados del error obtenido al implementar el controlador $PIDF$ se muestran en las graficas de las Figura 4.5.b hasta la Figura 4.10.b. Como se logra ver el error promedio obtenido es de 0.1 grados. Y la salida en promedio de los controladores es de 0.2 volt. (ver Figura 4.11.b hasta Figura 4.16.b).

	A465 ARTICULACIONES			A255 ARTICULACIONES		
	1	2	3	1	2	3
K_p	12	6	7.2	15	5.4	15
K_i	0.672	0.288	0.374	0.51	0.145	1.5
K_d	142	83.33	92.3	294	133	100

Tabla 4.5: Ganancias del controlador PIDF.

4.4 Control FPIDF

En el caso del control $FPIDF$ se empleó una ecuación similar a (4.2) y (4.3), sólo que para este tipo de control existen variables que permite escalar las ganancias K_d y K_i en función de la salida $u(t)$. Como se mencionó en el capítulo anterior, las ecuaciones que se programaron son:

$$x_i(t) = \left(e_i(t) + K'_{Ii}\int_0^t e(\tau)\mathrm{d}t + K'_{Di}\dot{e}_i(t)\right) \tag{4.4}$$

$$v_i(t) = K_{pi}x_i(t) \tag{4.5}$$

Las ganancias K'_{Di} y K'_{Ii} cumplen con la condición:

$$K'_{Di} = f_i(u) = \begin{cases} \frac{\alpha_{1i}}{|u_i|} & \text{si } |u_i| \geq \beta_{1i} \\ \frac{\alpha_{1i}}{\beta_{1i}} & \text{otro caso} \end{cases}$$

$$K'_{Ii} = f_i(u) = \begin{cases} \frac{\alpha_{2i}}{|u_i|} & \text{si } |u_i| \geq \beta_{2i} \\ \frac{\alpha_{2i}}{\beta_{2i}} & \text{otro caso} \end{cases}$$

Con base en la Tabla 3.1 se obtuvo $K'_{Ii} = \frac{f_i(u)}{T_{ni}}$ y $K'_{Di} = T_{vi}f_i(u)$, donde $f(u)$ una función de salida del mecanismo difuso. Estos valores se presentan a continuación. Por conveniencia se proponen que los valores de α_{1i}, α_{2i} sean similares a las ganancias derivativa e integrativa correspondiente, previamente obtenidas del controlador PID. β_{1i}, β_{2i} fueron sintonizadas con valor unitario. El resultado se muesra en las siguiente Tabla 4.6

	A465 ARTICULACIONES			A255 ARTICULACIONES		
	1	2	3	1	2	3
K_P	12	6	7.2	15	5.4	21
α_1	0.672	0.288	0.374	0.51	0.145	2.1
α_2	142	83.33	92.3	294.11	133	140
β_1	1	1	1	1	1	1
β_2	1	1	1	1	1	1

Tabla 4.6: Ganancias del controlador FPIDF.

Para el caso de la función $f_i(u)$ se consideraron también siete conjuntos triangulares para las ganancias. Considerando la misma nomenclatura a la entrada del controlador $PIDF$ se obtuvieron los siguientes valores.

Valores de los conjuntos de membresía de entrada del controlador $FPIDF$

Robot A465	b_{m_3}	b_{m_2}	b_{m_1}	b_0	b_1	b_2	b_3
Articulación 1	-1	-0.1	-0.05	0	0.05	0.1	1
Articulación 2	-2.3	-0.23	-0.115	0	0.115	0.23	2.3
Articulación 3	-1	-0.1	-0.05	0	0.05	0.1	1
Robot A255							
Articulación 1	-1	-0.1	-0.05	0	0.05	0.1	1
Articulación 2	-1	-0.1	-0.05	0	0.05	0.1	1
Articulación 3	-0.8	-0.064	-0.04	0	0.04	0.064	0.8

Tabla 4.7: Parámetros de los conjuntos de entrada $f_i(u)$ del controlador FPIDF.

Donde los limites de los conjuntos de membresía esta definidos por b_{m_3} b_{m_2} b_{m_1} b_0 b_1 b_2 b_3, para su programación:
En el caso de las reglas se empleó de igual forma el método de Mandami por las ventajas ya mencionadas. Asimismo, la FAM para este control se manejó de igual manera (ver Tabla 4.8).

Hay que mencionar que estas reglas se aplicaron a cada una de las articulaciones de los robots variando su tamaño. El ajuste de parámetros se llevó a cabo por prueba y error. Los valores de salida son también del tipo Singleton (ver Figura 4.3) y los resultados obtenidos se muestran en la Tabla 4.9.

$f_i(u)$	ng_i	nm_i	np_i	ze_i	pp_i	pm_i	pg_i
$u_i(t)$	pp_i	pm_i	pg_i	pg_i	pg_i	pm_i	pp_i

Tabla 4.8: FAM de K_{di} y K_{ii} del control FPIDF.

Valores de los conjuntos de membresía de salida del controlador $FPIDF$

Robot A465	d_{m_3}	d_{m_2}	d_{m_1}	d_0	d_1	d_2	d_3
Articulación 1	-0.5	-0.47	-0.45	0	0.47	0.47	0.5
Articulación 2	-2.5	-2.37	-2.25	0	2.25	2.37	2.5
Articulación 3	-2.5	-1.8	-1.75	0	1.75	1.8	2.5
Robot A255							
Articulación 1	-1.5	-1.12	-1.05	0	1.05	1.12	1.5
Articulación 2	-2.75	-2.06	-1.92	0	1.1.37	2.72	3.3
Articulación 3	-2.75	-2.06	-1.92	0	1.1.37	2.72	3.3

Tabla 4.9: Parámetros de los conjuntos de salida u_i del controlador $FPIDF$.

Donde los limites de los conjuntos de membresía esta definidos por d_{m_3} d_{m_2} d_{m_1} d_0 d_1 d_2 d_3 , para su programación:
Los resultados del error obtenido al implementar el controlador $FPIDF$ se muestran en las graficas de las Figura 4.5.c hasta la Figura 4.10.c. Como se logra ver el error promedio obtenido es de 0.1 grados. Y la salida en promedio de los controladores es de 0.2 volt.. (ver Figura 4.11.c hasta Figura 4.16.c).

4.5 Control por el método de Síntesis de Lyapunov

El último esquema de control propuesto consiste en un controlador PD con un termino difuso. Para el diseño de las reglas se utilizó el Método de Síntesis de Lyapunov. Las ganancias del PD son las mismas que las del PID, solo que esta vez se utilizó $K_i = 0$:

$$v_i(t) = K_{pi}\left(\tilde{q}_i(t) + t_{vi}\dot{\tilde{q}}_i(t)\right) . \tag{4.6}$$

Las ganancias proporcional y derivativa se ajustaron en línea, tomando en cuenta la parte del control por síntesis de Lyapunov, por lo que se obtuvieron los siguientes resultados por el método de prueba y error.

	A465 ARTICULACIONES			A255 ARTICULACIONES		
	1	2	3	1	2	3
K_p	42	126	120	135	126	90
K_d	2.352	6.048	0.624	4.59	3.420	9

Tabla 4.10: Ganancias del controlador PD.

Para la parte de síntesis de Lyapunov se consideraron los siguientes conjuntos y nomenclatura: **error grande negativo** $\tilde{q}_{gn_i}$,**error negativo** $\tilde{q}_{n_i}$,**error cero** $\tilde{q}_{0_i}$,**error positivo** $\tilde{q}_{p_i}$ **y error grande positivo** $\tilde{q}_{gp_i}$,quedando:
Valores de los conjuntos de membresía del error $\tilde{q}_i$

Robot A465	$\tilde{q}_{gn_i}$	$\tilde{q}_{n_i}$	$\tilde{q}_{0_i}$	$\tilde{q}_{p_i}$	$\tilde{q}_{gp_i}$
Articulación 1	-0.3	-0.15	0	0.005	0.3
Articulación 2	-0.3	-0.15	0	0.005	0.3
Articulación 3	-0.5	-0.025	0	0.025	0.5
Robot A255					
Articulación 1	-0.5	-0.025	0	0.025	0.5
Articulación 2	-0.5	-0.05	0	0.05	0.5
Articulación 3	-0.5	-0.05	0	0.05	0.5

Tabla 4.11: Parámetros de los conjuntos de entrada del error.

Valores de los conjuntos de membresía de la derivada del error .
Se consideró a la velocidad de error en tres conjuntos por lo que se propuso: **derivada del error negativa** $\dot{\tilde{q}}_{n_i}$, **derivada del error cero** $\dot{\tilde{q}}_{0_i}$ **y derivada del error positiva** $\dot{\tilde{q}}_{p_i}$ mostrados en la Tabla 4.12.

Robot A465	$\widetilde{q}_{n_i}$	$\widetilde{q}_{0_i}$	$\widetilde{q}_{p_i}$
Articulación 1	-0.5	0	0.5
Articulación 2	-0.5	0	0.5
Articulación 3	-0.5	0	0.5
Robot A255			
Articulación 1	-0.5	0	0.5
Articulación 2	-0.5	0	0.5
Articulación 3	-0.5	0	0.5

Tabla 4.12: Parámetros de los conjuntos de entrada de la velocidad del error.

Para el caso de control por síntesis de Lyapunov debe de cumplir con:

$$\dot{V} = \widetilde{q}_i\dot{\widetilde{q}}_i + \dot{\widetilde{q}}_i K_i V_{fi} + \dot{\widetilde{q}}_i \mid p_i'(t) \mid - a_i \dot{\widetilde{q}}_i^{\,2} - \dot{\widetilde{q}}_i \mid p_i''(t) \mid \leq 0 \tag{4.7}$$

Se propuso el método de Mandami por las características ya mencionadas. En este caso se cuenta con quince reglas, a diferencia de los anteriores controladores difusos, puesto que existen dos conjuntos que son el error y su derivada. Se empleó la teoría de relaciones mencionada en la Sección 2.3.5 ocupando la relación MAX-MIN de las ecuaciones (2.24), (2.25) y (2.26). Se propusieron los datos de la Tabla 4.11 para el error y los de la Tabla 4.12 para la velocidad.

Para formar las reglas del control, se utilizó una FAM que se muestra en la Figura 4.4. La Defuzzificación se hizo por el método de singleton (ver Tabla 4.13). Hay que hacer notar que también estos datos se fueron variando en línea por el método de prueba y error. Este tipo de controlador fue el menos eficiente debido a su forma en cascada con el control PD, que ocasiona que la parte del controlador de Síntesis de Lyapunov actué como la parte integrativa del controlador por lo que ocaciona que la eficiencia sea menor.

ė \ e	ng	n	ze	p	pg
n	pg	pg	p	ze	n
ze	pg	p	ze	n	ng
p	p	ze	n	ng	ng

Figura 4.4: FAM del control de síntesis de Lyapunov.

Valores de los conjuntos de membresía de salida

Robot A465	d_{m_2}	d_{m_1}	d_{m_0}	d_1	d_2
Articulación 1	-1	-0.8	0	0.8	1
Articulación 2	-1	-0.8	0	1	1.5
Articulación 3	-1.8	-1.44	0	1.44	1.8
Robot A255					
Articulación 1	-1	-0.8	0	0.8	1
Articulación 2	-1	-0.8	0	0.8	1
Articulación 3	-1	-0.8	0	0.8	1

Tabla 4.13: Parámetros de los conjuntos de salida u_i.

Los resultados del error obtenido al implementar el controlador de Síntesis de Lyapunov se muestran en las graficas de las Figura 4.5.d hasta la Figura 4.10.d. Como se logra ver el error promedio obtenido es de 0.5 grados. Y la salida en promedio de los controladores es de 1.5 volt. (ver Figura 4.11.d hasta Figura 4.16.d).

También se realizó un análisis comparativo con el propósito de observar el desempeño de cada uno de los algoritmos de control para así determinar cual presentó el mejor desempeño con respecto a los otros. El desempeño se evaluó mediante el índice $\mathcal{I}_i$, que mide el promedio de la raíz cuadrada del error de seguimiento de acuerdo con la formula:

$$\mathcal{I}_i = \sqrt{\frac{1}{T}\int_0^T \parallel \tilde{q}_i \parallel^2 dt} \tag{4.8}$$

Dando los siguientes resultados mostrados en la Tabla 4.14:

Robot A465	PID	$PIDF$	$FPIDF$	Síntesis Lyapunov
$\mathcal{I}_1$	0.045279	0.015022	0.012682	0.126870
$\mathcal{I}_2$	0.346636	0.385654	0.657369	0.613523
$\mathcal{I}_3$	0.078954	0.070227	0.118364	0.363232
$\mathcal{I}$	0.15695	0.15696	0.262805	0.367875
Robot A255				
$\mathcal{I}_1$	0.071956	0.081564	0.087369	0.328897
$\mathcal{I}_2$	0.041251	0.025972	0.034832	0.511481
$\mathcal{I}_3$	0.158904	0.102832	0.147213	0.811781
$\mathcal{I}$	0.09070	0.07012	0.194300	0.55071

Tabla 4.14: Tabla comparativa de índice de desempeño $\mathcal{I}$.

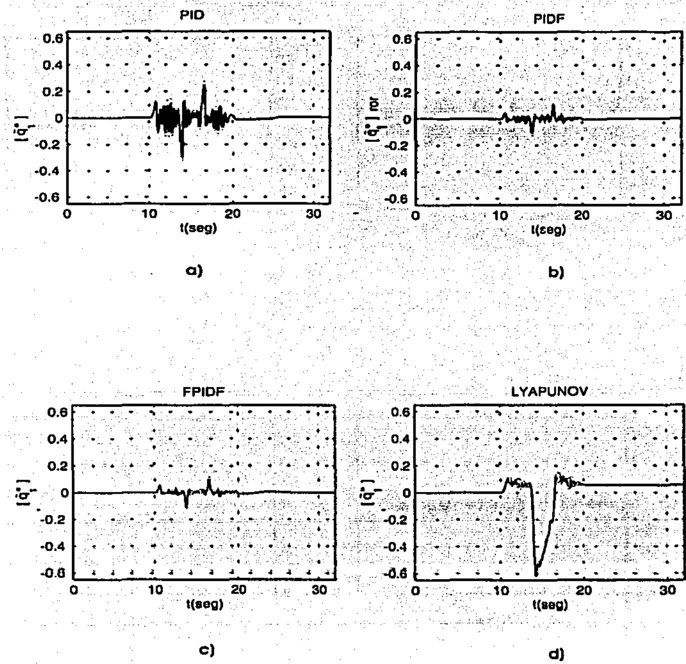


Figura 4.5: Error de seguimiento de la articulación 1 del robot A465 .

Errores de la articulación 1 con el algoritmo de control *PID*, *PIDF*, *FPIDF* y Síntesis de Lyapunos propuesto en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente.

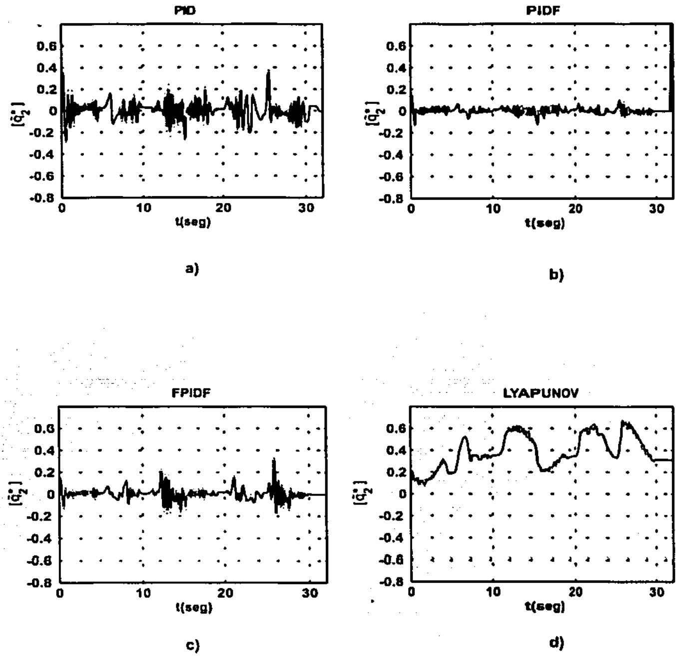


Figura 4.6: Error de seguimiento de la articulación 2 del robot A465.

Errores de la articulación 2 con el algoritmo de control PID, $PIDF$, $FPIDF$ y Síntesis de Lyapunos propuestos en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente.

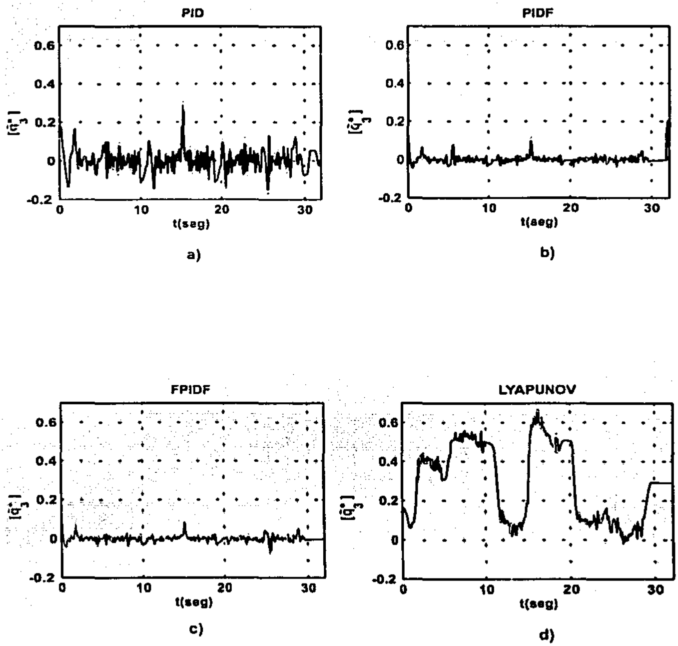


Figura 4.7: Error de seguimiento de la articulación 3 del robot A465.

Errores de la articulación 3 con el algoritmo de control PID, $PIDF$, $FPIDF$ y Síntesis de Lyapunos propuestos en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente.

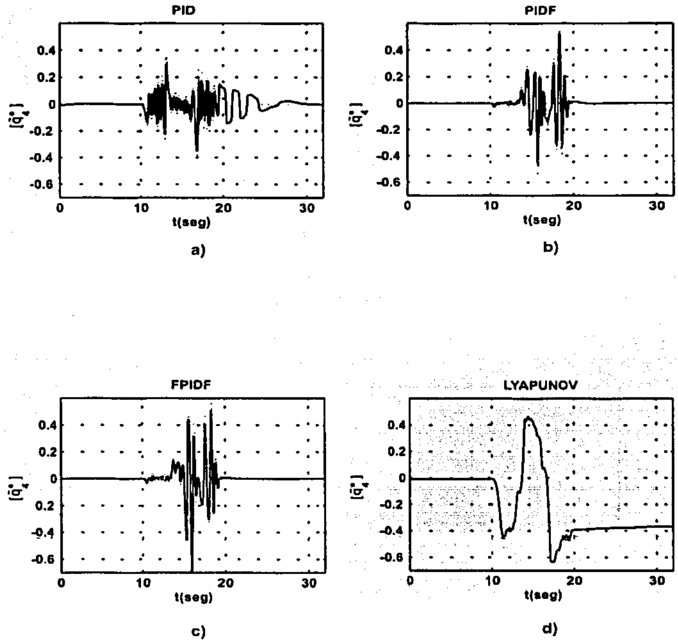


Figura 4.8: Error de seguimiento de la articulación 1 del robot A255.

Errores de la articulación 1 con el algoritmo de control PID, $PIDF$, $FPIDF$ y Síntesis de Lyapunos propuestos en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente.

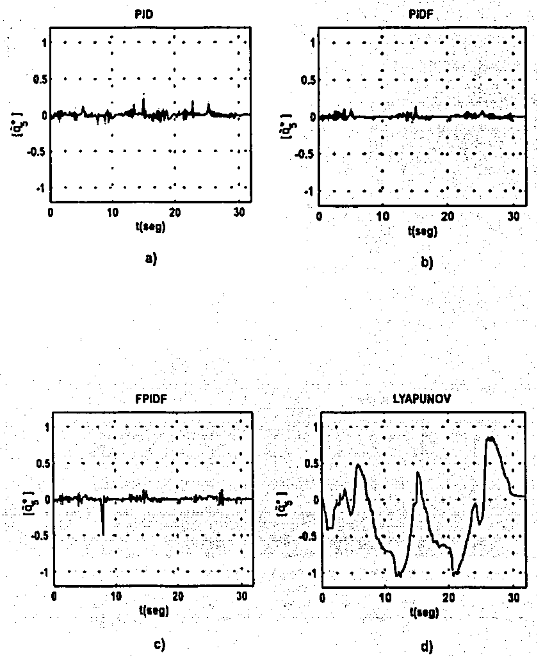


Figura 4.9: Error de seguimiento de la articulación 2 del robot A255.

Errores de la articulación 2con el algoritmo de control PID, $PIDF$, $FPIDF$ y Síntesis de Lyapunos propuestos en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente.

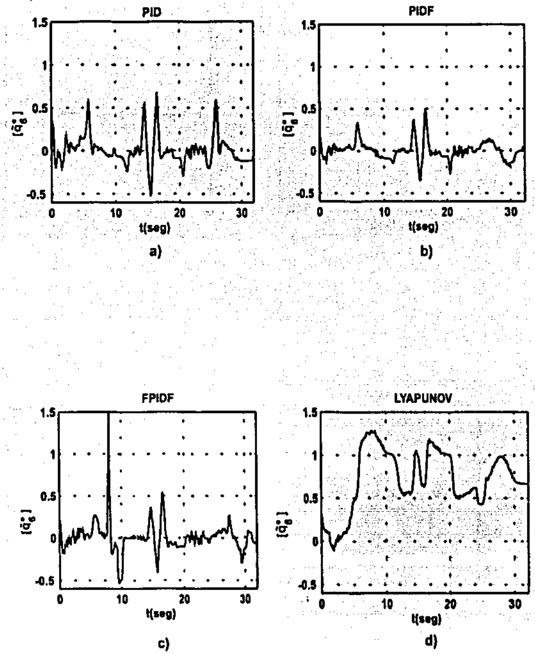


Figura 4.10: Error de seguimiento de la articulación 3 del robot A255.

Errores de la articulación 3 con el algoritmo de control *PID*, *PIDF*, *FPIDF* y Síntesis de Lyapunos propuestos en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente.

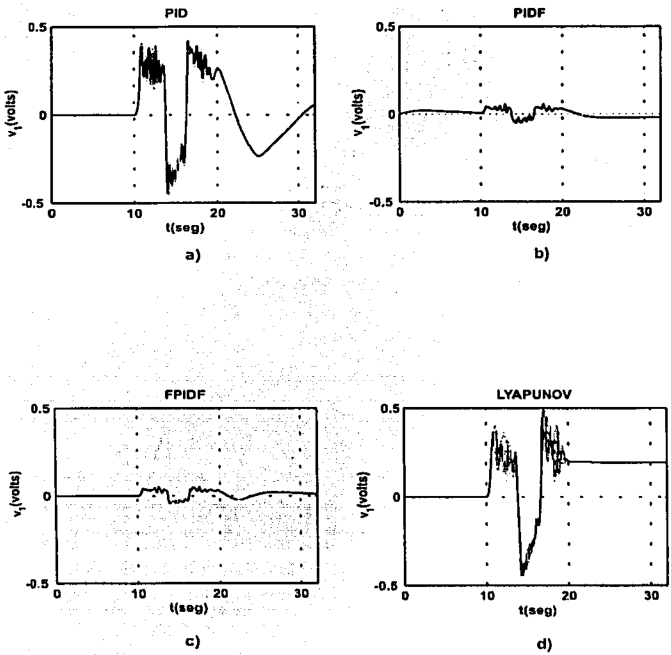


Figura 4.11: Salida $v_i(t)$ de la articulación 1 del robot A465.

Salida de voltaje de los controladores de la articulación 1 con los algoritmos de control PID, $PIDF$, $FPIDF$ y Síntesis de Lyapunos propuestos en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente.

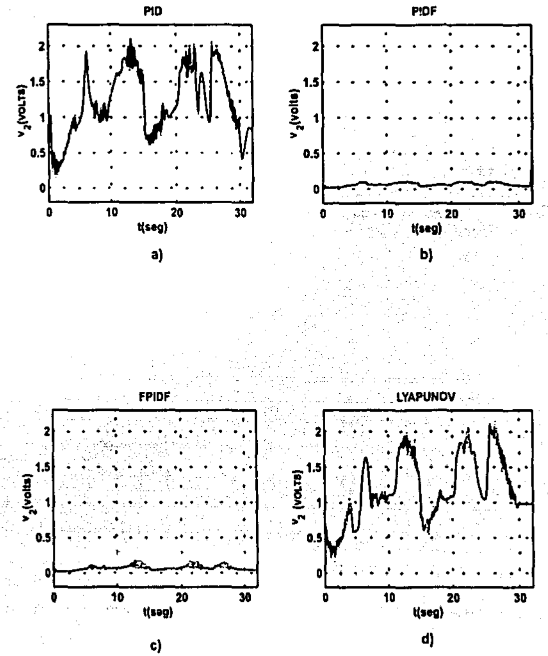


Figura 4.12: Salida $v_i(t)$ de la articulación 2 del robot A465.

Salida de voltaje de los controladores de la articulación 2 con los algoritmos de control PID, $PIDF$, $FPIDF$ y Síntesis de Lyapunos propuestos en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente. .

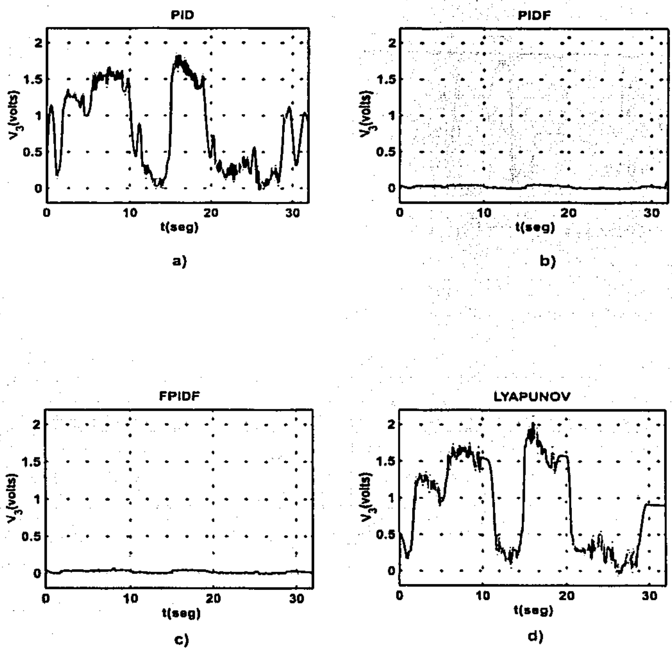


Figura 4.13: Salida $v_i(t)$ de la articulación 3 del robot A465.

Salida de voltaje de los controladores de la articulación 3 con los algoritmos de control PID, $PIDF$, $FPIDF$ y Síntesis de Lyapunos propuestos en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente.

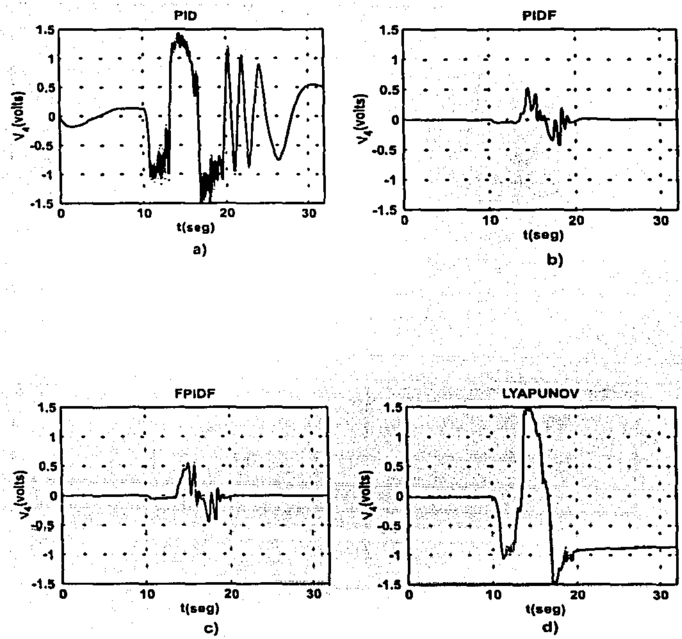


Figura 4.14: Salida $v_i(t)$ de la articulación 1 del robot A255.

Salida de voltaje de los controladores de la articulación 1 con los algoritmos de control PID, $PIDF$, $FPIDF$ y Síntesis de Lyapunos propuestos en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente.

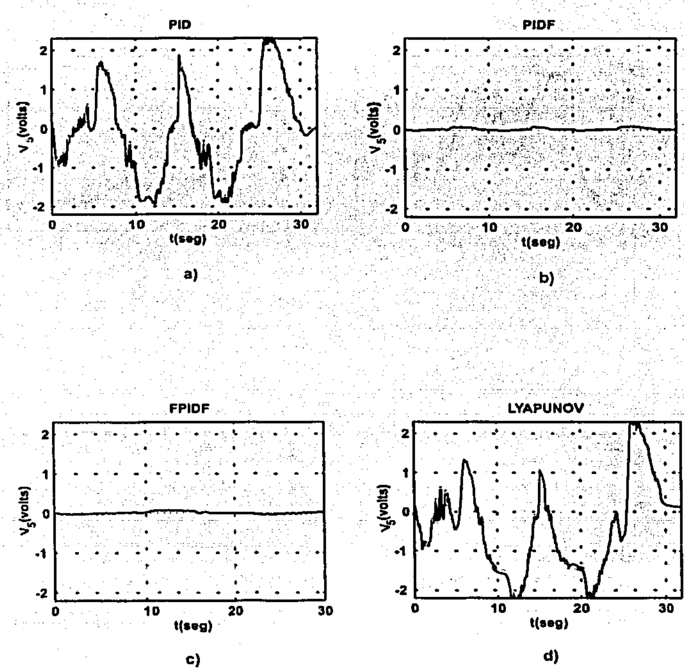


Figura 4.15: Salida $v_i(t)$ de la articulación 2 del robot A255.

Salida de voltaje de los controladores de la articulación 2 con los algoritmos de control PID, $PIDF$, $FPIDF$ y Síntesis de Lyapunos propuestos en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente.

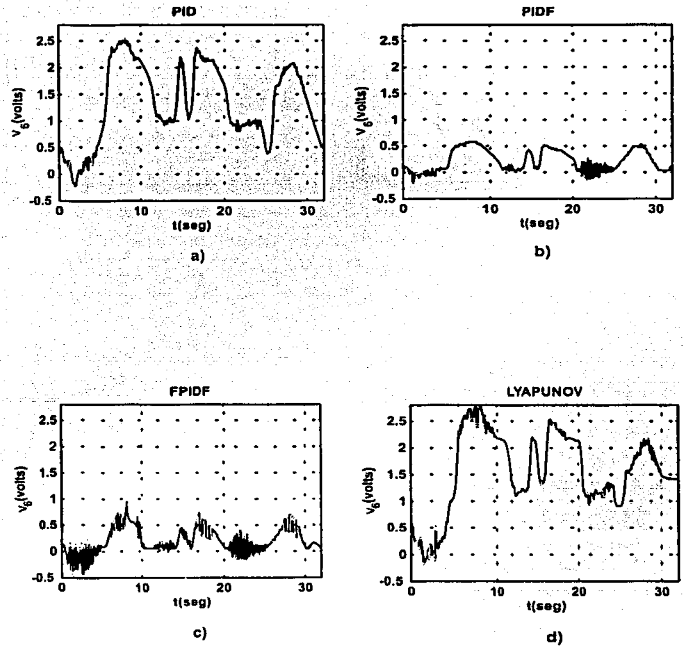


Figura 4.16: Salida $v_i(t)$ de la articulación 3 del robot A255.

Salida de voltaje de los controladores de la articulación 3 con los algoritmos de control PID, $PIDF$, $FPIDF$ y Síntesis de Lyapunos propuestos en las ecuaciones (4.1), (4.3), (4.5) y (4.8) respectivamente.

Capítulo 5

Conclusiones

Durante el desarrollo de esta tesis se mostró el desempeño de cuatro controladores del sistema de robots A465 y A255 del Laboratorio de Robótica de la DEPFI. El comportamiento de los controladores $PIDF$, $FPIDF$ y control por el Método de Síntesis de Lyapunov se comparó con un PID convencional, utilizando un índice de desempeño para tener una mejor comparación. Los algoritmos diseñados por Tao and Taur (2000) y método propuesto por Margaliot and Langholz (2000) no requieren del modelo de los robots. Las ganancias de controlador PID se obtuvieron a partir de método de Ziegler y Nichols. Estas ganancias fueron la base de partida para los distintos controladores aunque, se fueron variando conforme al desempeño en los robots, con la finalidad de reducir el error de posición al mínimo posible. De este modo se ganó experiencia en cada una de las seis articulaciones de forma independiente, lo que ayudó a sintonizar los controles difusos. Con esto se obtuvieron errores menores a 0.4 de grado en promedio. Para el control PID difuso ($PIDF$), se emplearon al principio las mismas ganancias del control PID. Como se observó que el sistema oscilaba, se modificaron las ganancias en línea hasta obtener un desempeño aceptable cuidando de que no aumentara el error. Las siete reglas del control difuso ayudaron a un mejor desempeño del sistema, ya que mejoraron los errores de posición. Estos datos también se variaron en línea, porque al modificar las ganancias se tenían que variar los valores de los conjuntos difusos tanto a la entrada como a la salida. El empleo del controlador flexible PID difuso $FPDF$ fue similar al controlador $PIDF$, siendo estos los resultados más aceptables de todos los controladores.

El controlador de síntesis de Lyapunov fue diferente a los dos anteriores por que está constituido por un control PD en paralelo con un control difuso. Esté tuvo errores mayores a 0.5. Cabe mencionar que el código fue el más grande, ya que se implementaron los conjuntos del error y derivada, con un total de quince reglas, Esto tuvo como consecuencia mayor carga computacional.

El índice de desempeño ($\mathcal{I}_i$) obtenido de cada una de las articulaciones muestra que $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$, del $PIDF$ y $FPIDF$ son casi iguales al control PID y que el de menor desempeño resultó ser el de síntesis de Lyapunov. La mayor diferencia se ve a la salida de voltajes donde los controles PIDF y FPDIF resultaron con un menor voltaje en la salida, en el caso del robot A465. Para el robot A255 se obtuvieron también los índices de desempeño, solo que en éste se ve una diferencia mayor del desempeño de los diferentes controladores, siendo también los mejores los controles $PIDF$ y $FPIDF$ y el menor el de síntesis de lyapunov.

Con todo lo anterior hay que mencionar que los controladores difusos mejoran el desempeño del los robots, debido a que la lógica difusa permite implementar la experiencia obtenida. Esto logra que el error este cercano a cero y que la salida sea suave por lo que los motores no sufren de sobre voltajes.

Apéndice A

Interfaz PC–Robot para la implementación de algoritmos de control

La simulación de algoritmos de control para robots por medio de una computadora ha tenido mucho impulso y ha desplazado la comprobación experimental. Una de las desventajas de la simulación por medio de la computadora radica en no asegurar la confiabilidad de los resultados que la teoría pueda validar. Es decir, fenómenos tales como perturbaciones, incertidumbres paramétricas o dinámicas no modeladas son difíciles de incluir de manera apropiada en la simulación. Crear laboratorios experimentales de robótica permite implementar algoritmos de control de varios tipos y probar el desempeño en condiciones reales de operación.

La mayoría de las compañías que venden robots suelen entregar manipuladores en los que no es posible separar la etapa de potencia, protección y control. Esta sección describe la conversión de un robot industrial a experimental mediante la creación de una interfaz entre las etapas de potencia, protección, motores y encoders. Esto permite la interacción PC–Robot mediante programación estructurada.

A fin de conseguir el diseño de la interfaz PC–Robot, fue necesario conocer las características y componentes de cada uno de los sistemas involucrados.

Robot manipulador CRS Robotics A465.

El manipulador CRS Robotics A465 fue diseñado con el mismo campo de acción de movimiento y levantamiento de carga promedio de un brazo humano. El Robot

CRS A465 es una combinación de potencia y alta velocidad; los motores de CD que utiliza este robot poseen una constante mecánica y eléctrica que los hace tener buen desempeño y alto torque de respuesta (CRS 1997*a*). Sus características principales son:

- Robot articulado de seis grados de libertad.
- Servomotores de C.D.
- Encoders ópticos.
- Transmisión de engranes armónica.
- 31 Kg de peso.
- Salidas de señales de encoder.

Modulo de procesamiento, protección y amplificación C500 CRS.

El módulo C500 CRS permite la salida de señales de encoder y la entrada de señales de control a la etapa amplificadora para su posterior salida al robot A465 (CRS 1997*a*). Sus principales características son las siguientes:

- Módulo multitarea C500C diseñado bajo el concepto de arquitectura abierta.
- Tarjeta de procesamiento PC-104/I486.
- Lenguaje de Programación y aplicaciones RAPL-3.
- Plataforma de control con esquema PID.
- Circuitos de paro y emergencia.
- Detector continuo de fallas.
- 16 entradas digitales aisladas óptimamente.
- 12 salidas digitales aisladas óptimamente.
- Dos puertos seriales.
- Seis unidades de amplificación.

Accesorio de interfaz de movimiento universal (UMI-Flex6)

Este accesorio de *National Instruments* permite la conexión de señales de encoders del robot a una tarjeta de adquisición de datos *Flex Motion*. También permite la conexión de otra tarjeta *Flex Motion* con señales de control de motores al módulo C500 para su amplificación correspondiente. Con su etapa de protección el módulo C500 evita daños en motores y tarjetas de adquisición de datos.

- 6 unidades de conexión de encoders, con seis terminales de conexión cada unidad $(A, \bar{A}, B, \bar{B}, I, \bar{I})$
- 6 unidades de conexión de Amplificador/motores.
- Conexión de switch límite (límite hacia delante, límite de reversa, entrada de home).
- Conexiones de energía.
- Unidad con ocho canales analógicos.
- Cable especial de entrada de 100 pines a tarjeta de adquisición de datos.

Tarjeta de adquisición de datos PCI Flex Motion 6C.

Tarjeta de *NationalInstruments* encargada de procesar señales digitales de encoders de robot hacia la PC; permite la salida de comandos de voltaje de motores en forma analógica (DAC's).

- Procesador DSP de tiempo real MC68331 a 32 bits.
- Control de servomotores.
- Seis entradas de encoders de alta resolución $(A, \bar{A}, B, \bar{B}, I, \bar{I})$.
- Ocho canales analógicos de entrada.
- Seis salidas analógicas de ± 10 volts.
- Conexión de switch límite (límite hacia delante, límite de reversa, entrada de home).

Computadora Personal

La computadora asignada al control de los robots permite el uso de programación estructurada para el procesamiento de algoritmos de control, teniendo como entrada señales de posición de articulación y como salida señales de control de actuadores de c.d.

- Procesador Pentium IV a 1.5 Ghz.
- Lenguaje de programación LabWindows/CVI basado en ANSI "*C*" de National Instruments con características graficas interactivas.

A.0.1 Interfaz

El diseño de esta interfaz tiene tres objetivos.

- Eliminar la etapa de procesamiento de control en el módulo original de fábrica mediante la interrupción en las señales de los sensores de posición. Utilizar la etapa de amplificación para las señales de comandos de voltaje de motores.
- Permitir que las señales de los sensores de posición del Robot CRS A465 se conecten desde una tarjeta UMI (Interfaz de Movimiento Universal) a una computadora que tiene incorporada una tarjeta de adquisición de datos número 1 de 32 bits a través de un cable especial. Hay 36 canales (conexiones) de lectura de encoders con información digital del desplazamiento angular.
- Por medio de una unidad UMI son conectadas las seis señales analógicas de control de cada motor al módulo amplificador de potencia; tales señales de control son previamente procesadas en la PC, y posteriormente enviadas a la unidad DAC de la tarjeta de adquisición de datos número 2 a través de un cable especial.

El siguiente diagrama simplifica el diseño de la interfaz PC–Robot

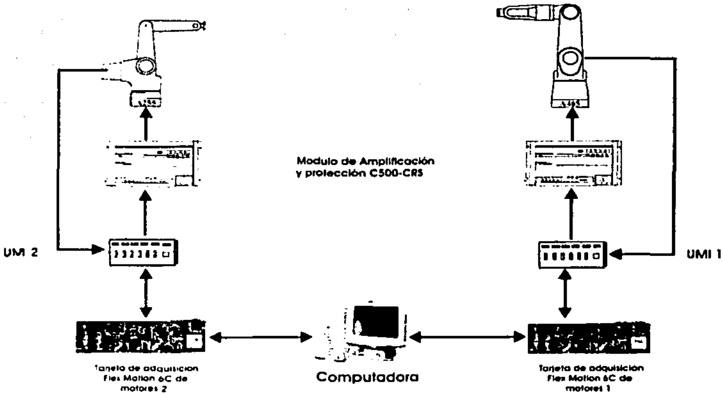


Figura A.1: Diseño de interfaz para control de los Robots A465 y A255.

Apéndice B

Especificaciones de los sistemas robot A465 y A255

Este apéndice describe a grandes rasgos las características más importantes de los sistemas robot A465 y A255. La Tabla B.1 muestra el área de trabajo de cada robot. Las Figuras B.1, B.2, B.3 y B.4 muestran el área de trabajo de los robots.

RANGO DE TRABAJO Y VELOCIDAD			
ROBOT	EJES	RANGO	VELOCIDAD MÁXIMA
A465	J1 (cintura)	± 175°	180 °/seg.
	J2 (hombro)	± 90°	180 °/seg.
	J3 (codo)	± 110°	180 °/seg.
	J4 (rotación de muñeca)	± 180°	171 °/seg.
	J5 (inclinación de muñeca)	± 105°	173 °/seg.
	J6 (rotación de la herramienta)	± 180°	171 °/seg.
A255	J1 (cintura)	350° (totales)	210 °/seg.
	J2 (hombro)	110° (totales)	210 °/seg.
	J3 (codo)	125° (totales)	210 °/seg.
	J4 (inclinación de muñeca)	220° (totales)	675 °/seg.
	J5 (rotación de muñeca)	360° (totales)	1350 °/seg.

Tabla B.1: Rango de trabajo y velocidad de los robot A465 y A255

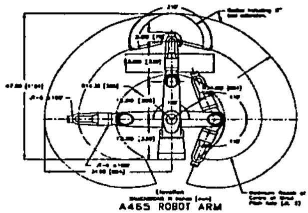


Figura B.1: Área de trabajo del robot A465 (vista lateral)

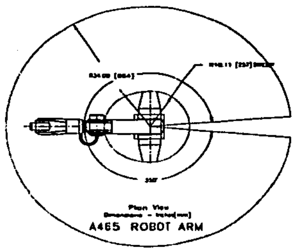


Figura B.2: Área de trabajo del robot A465 (vista superior)

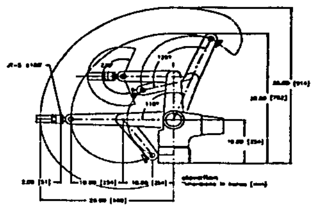


Figura B.3: Área de trabajo del robot A255 (vista lateral)

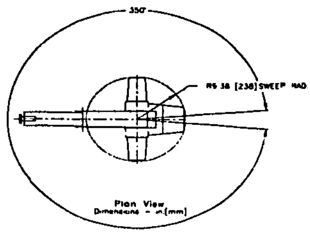


Figura B.4: Área de trabajo del robot A255 (vista superior)

Pin #	Function	Signature	Description
1	Vcom1	Analog	Command voltage for axis 1
2	Vcom5	Analog	Command voltage for axis 5
3	Vcom2	Analog	Command voltage for axis 2
4	Vcom6	Analog	Command voltage for axis 6
5	Vcom3	Analog	Command voltage for axis 3
6	Vcom7	Analog	Command voltage for axis 7
7	Vcom4	Analog	Command voltage for axis 4
8	Vcom8	Analog	Command voltage for axis 8
9	HSW1	OC	Homing switch for axis 1
10	PSW1	OC	Positive Travel switch for axis 1
11	NSW1	OC	Negative travel switch for axis 1
12	TSW1	OC	Thermal switch for axis 1
13	HSW2	OC	Homing switch for axis 2
14	PSW2	OC	Positive Travel switch for axis 2
15	NSW2	OC	Negative travel switch for axis 2
16	TSW2	OC	Thermal switch for axis 2
17	HSW3	OC	Homing switch for axis 3
18	PSW3	OC	Positive Travel switch for axis 3
19	NSW3	OC	Negative travel switch for axis 3
20	TSW3	OC	Thermal switch for axis 3
21	HSW4	OC	Homing switch for axis 4
22	PSW4	OC	Positive Travel switch for axis 4
23	NSW4	OC	Negative travel switch for axis 4
24	TSW4	OC	Thermal switch for axis 4
25	HSW5	OC	Homing switch for axis 5
26	PSW5	OC	Positive Travel switch for axis 5
27	NSW5	OC	Negative travel switch for axis 5
28	TSW5	OC	Thermal switch for axis 5
29	HSW6	OC	Homing switch for axis 6
30	PSW6	OC	Positive Travel switch for axis 6

Tabla B.2: Asignación de pines del conector J6 de la tarjeta CB del controlador C500

TESIS CON
FALLA DE ORIGEN

Pin#	Function	Signature	Description
31	NSW6	OC	Negative travel switch for axis 6
32	TSW6	OC	Thermal switch for axis 6
33	HSW7	OC	Homing switch for axis 7
34	PSW7	OC	Positive Travel switch for axis 7
35	NSW7	OC	Negative travel switch for axis 7
36	TSW7	OC	Thermal switch for axis 7
37	HSW8	CC	Homing switch for axis 8
38	PSW8	OC	Positive Travel switch for axis 6
39	NSW8	OC	Negative travel switch for axis 8
40	TSW8	OC	Thermal switch for axis 8
41	TPEStop+	Swith	Teach pendant E-Stop pair
42	TPEStop-		Teach pendant E-Stop pair
43	LiveMan+	Swith	Teach pendant liveman switch pair
44	LiveMan-		Teach pendant E-Stop pair
45	Brake+	Swith	Brake relay source
46	Brake-		Switch Brake relay return (normally open)
47	ArmOn+	Swith	Arm Power relay source
48	ArmOn-	Swith	Arm Power relay return (normally open)
49	AnalogIn1		
50	AnalogIn2		
51	AEEStop+	Switch	Auxiliary E-Stop switch pair. Used for exp Amp connector. Can be by passed on the auxiliary board with a switch.
52	AEEStop-		
53	N/C		
54	N/C		
55	N/C		
56	N/C		
57	N/C		
58	N/C		
59	N/C		
60	N/C		

Tabla B.3: Asignación de pines del conector J6 de la tarjeta CB del controlador C500.

Para más información acerca del controlador C500 consulte (CRS 1995), para obtener información mas detallada de los robots A465 y A255 puede consultar (CRS 1997*a*), (CRS 2001*a*), (CRS 2001*b*) y (CRS 1997*b*).

Pin#	Function	Signature	Description
1	A1+	RS-422	Axis 1 Channel A input
2	A1-	RS-422	Axis 1 Channel A input (complementary)
3	B1+	RS-422	Axis 1 Channel B input
4	B1-	RS-422	Axis 1 Channel B input (complementary)
5	Z1+	RS-422	Axis 1 Channel Z input
6	Z1-	RS-422	Axis 1 Channel Z input (complementary)
7	A2+	RS-422	Axis 2 Channel A input
8	A2-	RS-422	Axis 2 Channel A input (complementary)
9	B2+	RS-422	Axis 2 Channel B input
10	B2-	RS-422	Axis 2 Channel B input (complementary)
11	Z2+	RS-422	Axis 2 Channel Z input
12	Z2-	RS-422	Axis 2 Channel Z input (complementary)
13	A3+	RS-422	Axis 3 Channel A input
14	A3-	RS-422	Axis 3 Channel A input (complementary)
15	B3+	RS-422	Axis 3 Channel B input
16	B3-	RS-422	Axis 3 Channel B input (complementary)
17	Z3+	RS-422	Axis 3 Channel Z input
18	Z3-	RS-422	Axis 3 Channel Z input (complementary)
19	A4+	RS-422	Axis 4 Channel A input
20	A4-	RS-422	Axis 4 Channel A input (complementary)
21	B4+	RS-422	Axis 4 Channel B input
22	B4-	RS-422	Axis 4 Channel B input (complementary)
23	Z4+	RS-422	Axis 4 Channel Z input
24	Z4-	RS-422	Axis 4 Channel Z input (complementary)
25	A5+	RS-422	Axis 5 Channel A input
26	A5-	RS-422	Axis 5 Channel A Input (complementary)
27	B5+	RS-422	Axis 5 Channel B input
28	B5-	RS-422	Axis 5 Channel B input (complementary)
29	Z5+	RS-422	Axis 5 Channel Z input
30	Z5-	RS-422	Axis 5 Channel Z input (complementary)
31	A6+	RS-422	Axis 6 Channel A input
32	A6-	RS-422	Axis 6 Channel A Input (complementary)
33	B6+	RS-422	Axis 6 Channel B input
34	B6-	RS-422	Axis 6 Channel B input (complementary)
35	Z6+	RS-422	Axis 6 Channel Z input
36	Z6-	RS-422	Axis 6 Channel Z input (complementary)
37	NC	N/A	No Connect
38	Shield		No Connect
39			
40			
41	SGPos	Analog	Servo gripper position
42	SGTor	Analog	Servo gripper torque (not used)
43	AirGrip-	Power	Solenoid return
44	Gnd	Power	
45			
46	Gnd	Power	
47	+12 V	Power	+12 Volt supply to the servo gripper / Air Gripper Solenoid
48	Gnd	Power	
49	+12 V	Power	+12 Volt supply to the servo gripper / Air Gripper Solenoid
50	Gnd	Power	

Tabla B.4: Asignación de pines del conector J7 de la tarjeta CB del controlador C500

Rendimiento	
- Una tasa de actualización de 70 μs por eje - Tasa de actualización de instrucciones cada 10 ms - Búsqueda de entradas y salidas cada 20 ms - Actualización del servo a 1 kHz - Alcance de posición de 2^{24} (número de pulsos por revolución contados por el contador) - Alcance de la velocidad $\pm 2^{11}$ (pulsos por segundo) - Control del amplificador del motor de ±10 VDC - Resolución de 12 bites para comando de los motores - Encoder diferenciales o asimétricos - Realiza hasta 30 tareas concurrentes	
Hardware	
Procesadores	- Intel 80286/80287
Transputadores	- Inmos 20 MHz T400 - Inmos 30 MHz T805 (opcional) - 4 lugares sobre el tablero de circuitos para TRAM - Acceso externo para TRAM net
Ejes	- Soporte completo de 8 ejes - Soporte de servo manipulador
Entradas y Salidas (I/O)	- 16 entradas ópticamente aisladas - 4 salidas relé de contacto (24 V at 3 A) - 12 salidas ópticamente aisladas (150 mA at 24 V) - Entrada/Salida (I/O) expansibles a través de Interfaces PLC
Memoria disponible para el usuario	- 256 kilobytes (kbytes) de memoria de acceso al azar (RAM) respaldada por baterías - 512 kilobites (kbytes) de memoria FLASH para reserva no volátil (opcional)
Comunicacion	- Puertos RS232 duales, estándar 38.4 k-baud (baudios),con un máximo de 78.4 k-baudios - Protocolo de comunicaciones con detección de errores ACI
Temperatúra	- De + 10 a +400 centigrados
Energía	- 100/115/230 VAC, 60/50 Hz (seleccionable), 350 VA (A255 y controlador de movimiento), 900 VA (A465 y robots de función corrediza (gantry))
Estándares	- Diseñado para cumplir con los requisitos estandares de seguridad para robots de UL1740/ANSI/RIA 15.06 - Inmunidad EN50082-2 - Emisiones EN5008 1-2
Software	
Programación	- Lenguaje de programación RAPL-II y sistema operativo de multitarea - Robcomm-II - Herramientas de desarrollo para redes de transpútadores
Tipos de robots	- Articulados y de función corrediza (gantry)
Coordenadas	- Motor, articulación, mundo, y marcas de referencia remota
Tipo de control	- PID (proporcional-integral-derivativo)
Bosquejo de velocidad	- Trapezoide o parabolica
Programación por aprendizaje	- fuera de línea - consola de operacion
Tipos de vías (paths)	- articulacion interpolada (de punto a punto) - línea recta - vía continua - movimiento relativo - movimiento combinado - interpolacion circular
Consola de operación	
Configuracion	- portátil con un cable de 3 metros de largo
Pantalla de despliegue	- 4 filas x 20 caracteres (LCD)
Seguridad	- diseñado para cumplir con los requisitos estándar para consolas de operación de ANSI/RIA 15.02 - interruptor de activacion - parada de emergencia (asegurada)
Características físicas	
Peso	- 68 libras [31 kilogramo]
Tamaño del controlador	- 19 pulgadas [483 mm] de ancho x 10.5 pulgadas [267 mm] de largo x 15.75 pulgadas [400 mm] de espesor - 19 pulgadas instalable en armazón

Tabla B.5: Características del controlador C500

Apéndice C

Especificaciones de la tarjeta FlexMotion 6C

Este apéndice lista las especificaciones de funcionamiento del hardware y software para los controladores FlexMotion 6C.

Funcionamiento Servo

Rango de velocidad de actualización del PID	**62.5 a 500 μs/muestra**
Máxima velocidad de actualización del PID	**62.5 μs / eje**
Velocidad de actualización del PID en los 6 ejes	**375 μs total**
Velocidad de actualización de trayectoria	**62.5 μ/eje**
Sincronización multiejes	**<1 muestra**
Exactitud en posición	
Retroalimentación del Encoder	**± 1 cuenta de cuadratura**
Retroalimentación analógica	**± 1 LSB**
Parámetros de trayectoria	
Rango de posición absoluta	**± 2^{31} cuentas**
Máximo tamaño de movimiento relativo	**± 2^{31} cuentas**
Rango de velocidad	**1 a ± 16.000,000 cuentas / s**
Rango de RPM	**10^{-5} a 10^{6} revoluciones / min**
Aceleración / desaceleración	**4,000 a 128,000,000 cuentas / s^2**
Rango de RPS/s	**10^{-1} a 10^{8} revoluciones / s^2**
Rango de seguimiento de error	**0 a 32,767 cuentas**
Modos del lazo de servocontrol	**PID, PIVff, S-Curve, doble lazo**
Ganancias del PID (Kp, Ki y Kd)	**0 a 32,767**
Limite de integración (Ilim)	**0 a 32,767**
Periodo de muestreo derivativo(Td)	**1 a 63 muestras**
Ganancias de la alimentación adelantada (Aff, Vff)	**0 a 32,767**

Ganancia de retroalimentación de velocidad (Kv)	0 a 32,767
Salidas analógicas de los servo comandos	
Rango de voltaje	±10 V
Resolución	16 bits (0.000305 V / LSB)
Limites del torque programable	
Limite positivo	±10 V (-32,768 a +32,768)
Limite negativo	±10 V (-32,768 a +32,768)
Offset programable	±10 V (-32,768 a +32,768)

Funcionamiento a pasos

Rango de velocidad de actualización de trayectoria	125 a 500 μs/muestra
Máxima velocidad de actualización	125 μs/eje
Sincronización multieje	<1 muestra
Exactitud de posición	
Escalonamiento en lazo abierto	1 completo, medio o micro paso
Retroalimentación del encoder	±1 cuenta de cuadratura
Retroalimentación analógica	±1 LSB
Parámetros de trayectoria	
Rango de posición	$\pm 2^{31}$ pasos
Máximo tamaño de movimiento relativo	$\pm 2^{31}$ cuentas
Rango de velocidad	1 a ± 1,500,000 paso / s
Rango de RPM	10^{-5} a 10^{6} revoluciones / min
Aceleración / desaceleración	4,000 a 128,000,000 pasos / s^2
Rango de RPS/s	10^{-1} a 10^{8} revoluciones / s^2
Rango de seguimiento de error	0 a 32,767 cuentas
Salidas de escalonamiento	
Velocidad máxima del pulso	1.5 MHz (completo,medio,micro paso)
Ancho de pulso mínimo	300 ns a 1.5 MHz
Modo de salida del paso	paso y dirección o CW/CCW
Salidas de escalonamiento	TTL con colector abierto
Rango de voltaje	0 a 5 V
Salida de voltaje en bajo	<0.6 V a 64mA
Polaridad	Programable, activo en alto o activo en bajo

Sistema de seguridad

Función del relevador	Restablece el controlador al estado de inicio
Tiempo de respuesta	63 ms
Entrada E-Stop (paro de emergencia)	Optoacoplada
Rango de voltaje	0 a 24 V
Optoacoplador	TLP626 o equivalente

Corriente máxima de entrada	10 mA
Control	Deshabilita todos los ejes y comandos de salida

Movimiento de entrada / salida

Entrada de encoders	Cuadratura
Máxima velocidad de cuenta	
Encoder 1 y 2	16 MHz
Encoder 3 y 4	1 Mhz
Encoder 5 y 6	2 MHz
Opciones de las señales de encoder	Simple o complementada: A, $\overline{A}$, B, $\overline{B}$, Index, $\overline{\text{Index}}$ por eje
Voltaje de umbral de la entrada	± 0.3 V (típico)
Rango de voltaje	0 a 5 V
Ancho mínimo del pulso de Index	83 ns
Entradas adelante, reversa y home	Optoacopladas
Número de entradas	18 (3 por eje)
Rango de voltaje	0 a 24 V
Optoacoplador	TLP626 o equivalente
Corriente máxima de entrada	10 mA
Polaridad	Programable, activo en alto o activo en bajo
Entradas de señal de disparo	Optoacopladas
Numero de entradas	4 (del encoder 1 al 4)
Rango de voltaje	0 a 24 V
Optoacoplador	TLP2631 o equivalente
Corriente máxima de entrada	10 mA
Polaridad	Programable, activo en alto o activo en bajo
Ancho de pulso mínimo	83 ns
Salidas Breakpoint	Optoacopladas
Número de salidas	4 (del encoder 1 al 4)
Rango de voltaje	0 a 24 V
Optoacoplador	TLP627 o equivalente
Polaridad	Programable, activo en alto o activo en bajo
Salidas Habilitar / Deshabilitar	Colector abierto
Número de salidas	6 (1 por eje)
Rango de voltaje	0 a 12 V
Polaridad	Programable, activo en alto o activo en bajo
Entradas analógicas	
Número de entradas	8 (multiplexadas)
Rango de voltaje	± 10 V
Resistencia de entrada	20 KW min
Resolución	12 bit (0.0049 V/LSB)
Salidas analógicas	

Número de salidas	6
Rango de voltaje	± 10 V
Corriente de salida	± 10 mA
Resolución	16 bits (0.000305 V/LSB)

E/S digitales de 24 bits

Puertos	3, puertos de 8 bits
Máximo número de puertos de entrada	2
Máximo número de puertos de salida	3

Entradas	
Rango de voltaje	0 a 5 V
Voltaje bajo de entrada	0.8 V
Voltaje alto de entrada	2.0 V
Polaridad	Programable, activo en alto o activo en bajo

Salidas	
Rango de voltaje	0 a 5 V
Voltaje bajo de entrada	<0.5 V a 24 mA
Voltaje alto de entrada	>2.4 V a 3 mA
Polaridad	Programable, activo en alto o activo en bajo

Salidas PWM	
Número de salidas PWM	2
Frecuencia máxima PWM	32 KHz
Resolución	8 bits
Rango del ciclo de trabajo	0 a (255/256)%
Fuentes de reloj	Entrada de contador externa o interna

Requerimientos de energía (Máximos)

+5V (±3%)	2.0 A
+12V (±3%)	150 mA
-12V (±3%)	200 mA
Consumo de energía	14.2 W

Dimensiones (sin incluir conector)

PCI-FlexMotion 6C	33.8 x 9.9 cm (13.3 x 3.9 in)

Ambiente

Temperatura de operación	0 a 55 °C
Temperatura de almacenamiento	-20 a 70°C
Rango de humedad relativa	10 a 90% (sin condensación)

La Figura C.1 muestra la asignación de pines para el conector de 100 pines de E/S de movimiento para la tarjeta PCI-FlexMotion-6C. La línea arriba del nombre del pin indica que la señal se activa en bajo.

La Figura C.2 muestra la asignación de pines del conector auxiliar; por default el puerto de 24 bits esta dividido en 3 puertos digitales de E/S de 8 bits cada uno. Cada circuito de salida puede recibir 24mA y proporcionar 3 mA. Para más información sobre la configuración de la tarjeta FlexMotion consulte (NI 1999)

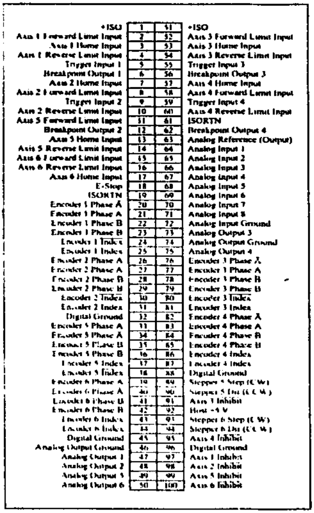


Figura C.1: Asignación de los 100 pines del conector de E/S de movimiento.

Port 3:bit 7	1	2	Digital Ground
Port 3:bit 6	3	4	Digital Ground
Port 3:bit 5	5	6	Digital Ground
Port 3:bit 4	7	8	Digital Ground
Port 3:bit 3	9	10	Digital Ground
Port 3:bit 2	11	12	Digital Ground
Port 3:bit 1	13	14	Digital Ground
Port 3:bit 0	15	16	Digital Ground
Port 2:bit 7	17	18	Digital Ground
Port 2:bit 6	19	20	Digital Ground
Port 2:bit 5	21	22	Digital Ground
Port 2:bit 4	23	24	Digital Ground
Port 2:bit 3	25	26	Digital Ground
Port 2:bit 2	27	28	Digital Ground
Port 2:bit 1	29	30	Digital Ground
Port 2:bit 0	31	32	Digital Ground
Port 1:bit 7	33	34	Digital Ground
Port 1:bit 6	35	36	Digital Ground
Port 1:bit 5	37	38	Digital Ground
Port 1:bit 4	39	40	Digital Ground
Port 1:bit 3	41	42	Digital Ground
Port 1:bit 2	43	44	Digital Ground
Port 1:bit 1	45	46	Digital Ground
Port 1:bit 0	47	48	Digital Ground
+5 V	49	50	Digital Ground

Figura C.2: Asignación de los 50 pines del conector auxiliar de E/S digital de 24 bits.

Nombre de la señal	Referencia	Dirección	Descripción
Analog Output < 1..6 >	Tierra de salida analógica	Salida	Comando analógico del motor
Analog Output Ground	-	-	Referencia para Vout
Stepper < 5..6 > step (CW)	Tierra digital	Salida	Escalonador solamente - paso del motor o control en dirección de las manecillas del reloj
Stepper < 5..6 > dir (CCW)	Tierra digital	Salida	Escalonador solamente - dirección del motor o control en dirección contrario de las manecillas del reloj
Axis < 1..6 > Inhibit	Tierra digital	Salida	Inhabilitación del amplificador/impulsor
Encoder < 1..6 > phase A	Tierra digital	Entrada	Entrada de la fase A del encoder
Encoder < 1..6 > phase A	Tierra digital	Entrada	Entrada de la fase A complementada del encoder
Encoder < 1..6 > phase B	Tierra digital	Entrada	Entrada de la fase B del encoder
Encoder < 1..6 > phase B	Tierra digital	Entrada	Entrada de la fase B complementada del encoder
Encoder < 1..6 > Index	Tierra digital	Entrada	Entrada de índice o marcador
Encoder < 1..6 > Index	Tierra digital	Entrada	Entrada de índice o marcador complementado
Digital Ground	-	-	Referencia para las entradas/salidas digitales
Host +5V	Tierra digital	Salida	Bus de la computadora de 5 Volts
+ISO	ISORTN	Entrada	Fuente de voltaje aislada
Trigger Input < 1..4 >	ISORTN	Entrada	Entrada de la señal de disparo
Breakpoint Output < 1..4 >	ISORTN	Salida	Salida del punto de ruptura
Axis < 1..6 > forward limit input	ISORTN	Entrada	Señal de entrada del limite hacia adelante de los ejes
Axis < 1..6 > home input	ISORTN	Entrada	Señal de entrada de home de los ejes
Axis < 1..6 > reverse limit input	ISORTN	Entrada	Señal de entrada del limite hacia atrás de los ejes
E-stop	ISORTN	Entrada	Entrada del paro de emergencia
ISORTN	-	-	Referencia para señales aisladas
Analog Reference	Analog Input Ground	Salida	7.5 V (nom) de salida
Analog Input < 1..8 >	Analog Input Ground	Entrada	Entradas del ADC de 12 bits
Analog Input Ground	-	-	Referencia para entradas analógicas

Tabla C.1: Asignación de los 100 pines del conector de E/S de movimiento

Apéndice D

Teoría de Lyapunov

Considérese el sistema dinámico no lineal representado por (?):

$$\dot{\mathbf{x}} = f(x), \tag{D.1}$$

donde f es una función vectorial no lineal y $x \in \mathbb{R}^n$ es el vector de estados. Se dice que el sistema no lineal (D.1) es autónomo o invariante en el tiempo dado que f no depende explícitamente del tiempo. La teoría de Lyapunov es una herramienta fundamental para el análisis de estabilidad de sistemas dinámicos, como robots manipuladores. Los conceptos de estabilidad básica están dados por las siguientes definiciones.

Definición 1 **(Equilibrio)** Un estado $\mathbf{x}^*$ es un punto de equilibrio de (D.1) si $f(\mathbf{x}^*) = 0$. $\triangle$

Definición 2 **(Estabilidad)** El punto de equilibrio $\mathbf{x} = 0$ se dice que es estable, si, para cualquier $\rho > 0$ existe una $r > 0$ tal que $\| \mathbf{x}(0) \| < r$, entonces $\| \mathbf{x}(t) \| < \rho \ \forall t \geq 0$. De otra manera el punto de equilibrio es inestable. $\triangle$

Definición 3 **(Estabilidad asintótica)** Un punto de equilibrio mbfx $= 0$ es asintóticamente estable si es estable y existe algún $r > 0$ tal $\| \mathbf{x}(0) \| < r$ implica que $\mathbf{x}(t) \to 0$ cuando $\mathbf{x}(t) \to \infty$. $\triangle$

Definición 4 **(Estabilidad marginal)** Un punto de equilibrio que es estable pero no asintóticamente estable se dice que es marginalmente estable. $\triangle$

Definición 5 (Estabilidad exponencial) Un punto de equilibrio es exponencialmente estable si existen dos números positivos α y λ independientes del tiempo y condiciones iniciales tales que

$$\| \mathbf{x}(t) \| \leq \alpha e^{-\lambda t} \| \mathbf{x}(0) \| \; \forall t > 0 \tag{D.2}$$

en alguna región alrededor del origen. △

Las definiciones anteriores corresponden a las propiedades locales del sistema alrededor del punto de equilibrio. Los conceptos de estabilidad anteriores llegan a ser globales cuando las condiciones correspondientes son satisfechas por cualquier estado inicial.

Método directo de Lyapunov. El método de Lyapunov sirve para determinar si un punto de equilibrio es estable de acuerdo a las definiciones (D.2) y (D.5). Considérense las siguientes definiciones:

Definición 6 Una función escalar continua $V(\mathbf{x})$ es localmente semidefinida positiva si $V(0) = 0$ y $V(\mathbf{x}) > 0$ ($V(\mathbf{x}) \geq 0$) para mbf$\mathbf{x} = 0$. De manera similar, $V(\mathbf{x})$ se dice que es (semi)definida negativa si $-V$(mbf$\mathbf{x}$) es semi-definida positiva.

Definición 7 (Función de Lyapunov) $V(\mathbf{x})$ es denominada función de Lyapunov para el sistema (D.1) si, en una región B, $V(\mathbf{x})$ es definida positiva y tiene derivadas parciales continuas, y sus derivadas con respecto al tiempo, evaluada en (D.1) es definida semidefinida negativa, es decir.

$\dot{V}(\mathbf{x}) = (\frac{\delta V}{\delta \mathbf{x}})f(\mathbf{x}) \leq 0.$ △

Los siguientes teoremas pueden ser utilizados para el análisis de estabilidad local y global respectivamente.

Teorema 1. El punto de equilibrio 0 del sistema D.1es (asintóticamente) estable en una región B, si existe una función escalar $V(\mathbf{x})$ con derivada continua tal que $V(\mathbf{x})$ es definida positiva, $\dot{V}(\mathbf{x})$ es semi-definida negativa (definida negativa) en la región B. △

Teorema 2. El punto de equilibrio del sistema D.1 es global asíntotamente estable si existe una función escalar $V(\mathbf{x})$ con derivada de primer orden continuas tal que $V(\mathbf{x})$ es definida positiva, $\dot{V}(\mathbf{x})$ es definida negativa y $V(\mathbf{x})$ es radialmente no acotada, es decir $V(\mathbf{x}) \to \infty$ cuando $\| \mathbf{x} \| \to \infty$.

Bibliografía

Åström, Karl J. and Björn Wittenmark (1989). *Adaptive Control.* Addison–Wesley. Reading, Masachusetts.

Beale, M. (1994). *Fuzzy Sistems Toolbox.* 1^{th} ed.. Publis Shing Company. Boston,USA.

Bentrup, Carina M. (2002). Manipulación estable de objetos rígidos utilizando dos robots industriales y un sistema de visión. Tesis de Licenciatura. Facultad de Ingeniería, Universidad Nacional Autonoma de México.

Castillo, Adrian M. (2002). Adaptación de dos robots Industriales para su utilización en el desarrollo de nuevas tecnicas y algoritmos de control. Tesis de Licenciatura. Escuela Nacional de Estudios Profecionales Campus Aragon, Universidad Nacional Autonoma de México.

CRS (1995). *Controlador C500 – Guia de usuario.* CRS Robotics Corporation. Ontario, Canada. UMI-17-556-E.

CRS (1997*a*). *A465 Robot Service Manual.* CRS Robotics Corporation. Ontario, Canada. UMS-17-504.

CRS (1997*b*). *Brazo Robot A255 - Guia de usuario.* CRS Robotics Corporation. Ontario, Canada. UMI-14-504-E.

CRS (2001*a*). *A255 Robot System User Guide.* CRS Robotics Corporation. Ontario, Canada. UMI-A255-400.

CRS (2001*b*). *A465 Robot System User Guide.* CRS Robotics Corporation. Ontario, Canada. UMI-A465-400.

Jamshidi, M. (1993). *Fuzzy logic and Control.* 1^{th} ed.. Prentice Hall. New Mexico,USA.

King, E.R. (1999). *Computational Intelligence in Control Engineering.* 2^{th} ed.. Marcel Dekker. New York USA.

Kosko, B. (1999). *Fuzzy Engeneering.* 1^{th} ed.. Prentice Hall. Southern California, USA.

Llamazares, R.B. (2001). *Factorización de prereferencias difusas.* Dpto de economia aplicada(Matemáticas). España.

Margaliot, M. and G. Langholz (2000). *New Approaches to Fuzzy Modeling and Control.* World Scientific. Singapore. 47.

Martin del Brio, B. and A. Sanz Molina (2001). *Redes Neuronales y Sistemas Borrosos..* Ra-ma. España.

NI (1999). *Motion Control - FlexMotion™ - 6C Hardware User Manual.* 1999 ed.. National Instrumenis. Austin, Texas, USA. Part Number: 321944B-01.

Spong, M. W. and M Vidyasagar (1989). *Robot Dynamics and Control.* John Wiley & Sons. New York.

Tao, W. and J.S Taur (2000). Flexible complexity reduced pid-like fuzzy controllers. *IEEE Transactions on systems, man and cybernetics-part b:cybernetics,* 30(4), 510–516.