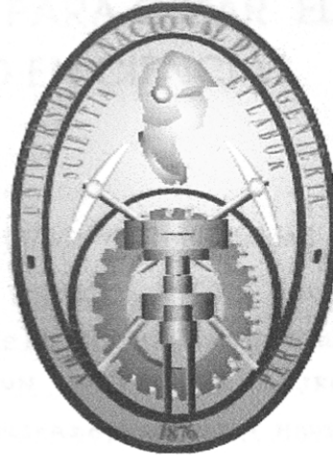


**UNIVERSIDAD NACIONAL DE INGENIERÍA**  
**FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**



**Diseño e Implementación de un Sistema de Control No  
Lineal Multivariable por Realimentación de Estado  
Aplicado a un Robot Manipulador Esférico  
de 2 Grados de Libertad**

**TESIS**

Para optar el Grado de Maestro en Ciencias  
Mención: Automática e Instrumentación

**PRESENTADO POR:**

**Ronald Divio Maravi Pinto**

**LIMA - PERÚ**  
**2004**

UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad de Ingeniería Eléctrica y Electrónica

Diseño e Implementación de un Sistema de Control No Lineal Multivariable por  
Realimentación de Estado Aplicado a un Robot Manipulador Esférico de 2  
Grados de Libertad

TESIS

Para optar el grado de Maestro en Ciencias

Mención: Automática e Instrumentación

Presentada por:

Ronald Divio Maravi Pinto

Lima - Perú

### **Extracto**

La presente tesis trata sobre el diseño e implementación de un controlador no lineal multivariable por realimentación de estado, aplicado a un robot manipulador esférico de dos grados de libertad (2DOF: 2 Degree-of-Freedoms). Este robot se compone de una base giratoria y un eslabón, cuyas posiciones angulares son sujetos de control. Las metas impuestas son: diseño, construcción y modelado del proceso, diseño del controlador no lineal multivariable de 2 entradas y 2 salidas, y simulación e implementación en tiempo real del sistema de control diseñado. Los resultados experimentales demuestran que las salidas del proceso son capaces de seguir la evolución de trayectorias arbitrarias deseadas.

El manipulador esférico usado en esta tesis fue construido e implementado, en las aulas de la Sección de Postgrado de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Ingeniería, siendo los gestores de este trabajo el Dr. Arturo Rojas y mi persona.



Universidad Nacional de Ingeniería  
Facultad de Ingeniería Eléctrica y Electrónica

Design and Implementation of a System of Control NonLinear Multivariable  
Feedback Linearization Applied to a Manipulator Robotic Spherical Two  
Degrees of Freedoms

## THESIS

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

by:

Ronald Divio Maravi Pinto

Lima - Perú

### **Abstract**

This thesis deals with design and implementation of a nonlinear multivariable control system, which employs the feedback linearization methodology. Such a nonlinear control system is applied to a spherical robot manipulator of 2DOF (Two-Degrees-of-Freedoms). This manipulator is composed of a DC servomotor with optical encoder that drives the rotary motion of the base of a metallic disk which is fixed to a metallic bar. Another DC servomotor with optical encoder is mounted on the free end of the bar and employed to drives the rotary motion of a metallic link. The goals imposed are: process modelling, design of the multivariable (two inputs and two outputs) control system, simulation, and real time implementation. The experimental results demonstrate that the process outputs are capable of tracking desired trajectories.

The spherical robot manipulator used in this thesis work was conceived, designed and constructed by the research team for control systems of the Graduate School of the Faculty of Electrical and Electronic Engineering. The members of such a team are Professor Arturo Rojas Moreno, and graduate students Ronald Maravi Pinto and Edgardo Espinoza Malqui.

# Tabla de Contenido

<b>Lista de Figuras</b> . . . . .	1
<b>Lista de Tablas</b> . . . . .	3
<b>Capítulo 1</b>	
<b>Introducción</b> . . . . .	4
1.1 Formulación del Problema . . . . .	5
1.2 Perspectiva Histórica . . . . .	6
1.3 Objetivos . . . . .	6
1.4 Organización de la tesis . . . . .	6
<b>Capítulo 2</b>	
<b>Modelado de la Planta</b> . . . . .	8
2.1 Introducción . . . . .	8
2.2 Descripción del Sistema . . . . .	8
2.3 El Modelo Dinámico . . . . .	11
2.3.1 El Procedimiento Denavit-Hartenberg . . . . .	11
2.3.2 El Algoritmo de Lagrange-Euler . . . . .	14
2.4 Aplicación y Programación del Modelo . . . . .	16
2.4.1 Lista de variables . . . . .	16
2.4.2 Cálculos . . . . .	16
2.4.3 Programación del Algoritmo Lagrange-Euler . . . . .	21
2.4.4 El Modelo de los Actuadores . . . . .	21
2.4.5 Modelo en el Espacio Estado . . . . .	24
<b>Capítulo 3</b>	
<b>Linealización No Lineal</b> . . . . .	27
3.1 Herramientas de Geometría Diferencial . . . . .	27
3.1.1 Derivada de Lie y Corchete de Lie . . . . .	28
3.1.2 Difeomorfismo y Transformación de Coordenadas . . . . .	29
3.1.3 El Teorema de Frobenius . . . . .	30
3.2 Linealización por Realimentación para Procesos SISO . . . . .	31
3.2.1 Condiciones para Linealización . . . . .	31
3.2.2 Grado Relativo del Proceso . . . . .	32
3.2.3 Forma Normal SISO para Linealización Exacta . . . . .	33
3.2.4 Ley de Control SISO para Linealización Exacta . . . . .	34
3.3 Linealización por Realimentación para Procesos MIMO . . . . .	35

3.3.1	Modelando Procesos Cuadrados No Lineales . . . . .	35
3.3.2	Grado Relativo Total . . . . .	36
3.3.3	Forma Normal para Linealización Exacta MIMO . . . . .	36
3.3.4	Ley de Control de Desacoplamiento MIMO . . . . .	38
<b>Capítulo 4</b>		
	<b>Implementación en Tiempo Real</b>	40
4.1	Implementación del Hardware . . . . .	41
4.1.1	El Actuador . . . . .	41
4.1.2	El Sensor . . . . .	46
4.1.3	La Interfaz Lab-PC+ . . . . .	47
4.2	Software de Control . . . . .	48
4.2.1	Diagrama flujo . . . . .	48
4.2.2	Programa Principal XX.CPP . . . . .	49
4.2.3	Funciones Propias del Programa XX.CPP . . . . .	49
4.2.4	La Función Principal Main . . . . .	50
<b>Capítulo 5</b>		
	<b>Simulación del Sistema</b>	52
5.1	Caso 1: Control de Trayectoria Constante . . . . .	53
5.2	Caso 2: Control de Trayectoria Senoidal . . . . .	58
5.3	Caso 3: Simulación con la Interfaz Gráfica de Usuario . . . . .	60
<b>Capítulo 6</b>		
	<b>Resultados Experimentales</b>	67
6.1	Experimento 1: Control de Trayectoria Constante . . . . .	67
6.2	Experimento 2: Control de Trayectoria Senoidal . . . . .	70
<b>Capítulo 7</b>		
	<b>Conclusiones</b>	73
7.1	Conclusiones . . . . .	73
7.2	Recomendaciones para Trabajos Futuros . . . . .	74
<b>Apéndice A</b>		
	<b>Identificación de Parámetros</b>	75
A.1	Modelo del Servomotor DC . . . . .	75
A.2	Implementación . . . . .	78
A.3	Aplicación . . . . .	79
A.3.1	Datos de Fabricante . . . . .	79
A.3.2	Simulación en Tiempo Real . . . . .	79
A.3.3	Cálculos . . . . .	80
A.4	Coefficiente de Fricción Viscosa del Brazo . . . . .	82
<b>Apéndice B</b>		
	<b>Programas en MATLAB</b>	87
B.1	Listado de Programas en MATLAB . . . . .	87
B.2	Programa del Cálculo de la Ecuación Dinámica del Manipulador . . . . .	87

B.3 Programa de Simulación con Trayectoria Constante . . . . .	91
B.4 Programa de Simulación con Trayectoria Senoidal . . . . .	97

**Apéndice C**

<b>Programas en C++</b> . . . . .	103
C.1 Listado de Programas en C++ . . . . .	103
C.2 Programa para Identificación de Parámetros del Servomotor DC .	103
C.3 Programa para el Experimento con Trayectoria Constante . . . .	108
C.4 Programa para el Experimento con Trayectoria Senoidal . . . . .	115

**Bibliografía**

## Lista de Figuras

2.1	Esquema del Robot Manipulador Esférico de 2DOF . . . . .	9
2.2	Esquema con las medidas del Manipulador . . . . .	9
2.3	Sistemas de referencias del Manipulador . . . . .	10
2.4	Momento de inercia del disco respecto al sistema de referencia 1 ( $S_1$ ). . . . .	19
2.5	Momento de inercia del prisma respecto al sistema de referencia 1 ( $S_1$ ) . . . . .	19
2.6	Servomotor DC1 de la base del Manipulador . . . . .	22
2.7	Servomotor DC2 del brazo del Manipulador . . . . .	23
4.1	Esquema de Implementación del sistema. . . . .	40
4.2	Planta SRM de 2DOF real . . . . .	42
4.3	Implementación del sistema real . . . . .	42
4.4	Circuito generador de PWM. . . . .	43
4.5	Circuito amplificador de potencia de PWM. . . . .	44
4.6	Esquema del circuito de disparo y conmutación. . . . .	45
4.7	Estructura del codificador óptico rotatorio. . . . .	46
4.8	Sensado de posición . . . . .	47
4.9	Diagrama de flujo del algoritmo de control. . . . .	51
5.1	Diagrama de bloques del controlador no lineal . . . . .	52
5.2	Trayectoria constante: salidas controladas para la base y el brazo. . . . .	57
5.3	Trayectoria constante: señales de control para los servomotores de la base y el brazo. . . . .	58
5.4	Trayectoria constante: velocidad para los servomotores de la base y el brazo. . . . .	59
5.5	Trayectoria senoidal: salidas controladas para la base y el brazo . . . . .	61
5.6	Trayectoria senoidal: señales de control para los servomotores de la base y el brazo . . . . .	62
5.7	Trayectoria senoidal: velocidad de los servomotores de la base y el brazo. . . . .	62

5.8	Panel de Control de Simulación de Trayectoria Constante del SRM de 2DOF con GUI. . . . .	64
5.9	Panel de Control de Simulación de Trayectoria Senoidal del SRM de 2DOF con GUI. . . . .	66
6.1	Trayectoria Constante Experimental: salidas controladas para la base y el brazo. . . . .	68
6.2	Trayectoria Constante Experimental: señal de control para el servomotor de la base y el brazo. . . . .	69
6.3	Trayectoria Constante Experimental: velocidad del servomotor de la base y el brazo. . . . .	69
6.4	Trayectoria Senoidal Experimental: salidas controladas para la base y el brazo. . . . .	71
6.5	Trayectoria Senoidal Experimental: señal de control para el servomotor de la base y el brazo. . . . .	71
6.6	Trayectoria Senoidal Experimental: velocidad del servomotor de la base y el brazo. . . . .	72
A.1	Circuito Eléctrico del Servomotor DC . . . . .	76
A.2	Curva de velocidad . . . . .	77
A.3	Implementación para la obtención de datos experimentales del servomotor DC . . . . .	78
A.4	Gráfico para identificación de parámetros del servomotor DC en vacío. . . . .	81
A.5	Gráfico para identificación de parámetros del servomotor DC1 de la base. . . . .	81
A.6	Gráfico para identificación de parámetros del servomotor DC2 del brazo. . . . .	82
A.7	Modelo del servomotor DC. . . . .	84
A.8	Simulación del modelo simplificado. . . . .	85
A.9	Respuesta tensión de entrada versus velocidad. . . . .	85
A.10	Experimento de la compensación de la fricción no lineal. . . . .	86

# Capítulo 1

## Introducción

En procesos multivariables, caracterizados por presentar un comportamiento dinámico complicado, se usa e implementa actualmente sistemas de control avanzado, los cuales en los últimos años han alcanzado un elevado nivel de aceptación industrial.

Las técnicas de control avanzado que han logrado mayor aceptación son las siguientes: control no lineal por el método de linealización de la realimentación de estado, control adaptivo no lineal, control no lineal por modos deslizantes, control predictivo no lineal, control no lineal usando backstepping, y control con inteligencia artificial (control difuso, neurocontrol, etc.).

La técnica a emplear en la presente tesis es la de control no lineal basada en el método de linealización por realimentación de estado y se basa en la linealización del proceso mediante un cambio adecuado de coordenadas y una realimentación del vector de estado de dicho proceso, de modo tal que el proceso resultante en su forma normal nos permita aplicar técnicas de control lineal en el caso de linealización exacta o bien atacar los problemas de control con un grado de dificultad menor en el caso de linealización parcial.

El método de linealización exacta se aplica cuando el proceso no lineal MIMO (multiple-input-multiple-output) de orden  $n$ , de  $m$  entradas y  $m$  salidas es completamente linealizable y desacoplado si su grado relativo total ( $r$ ) es igual al orden del proceso, o sea  $r=n$ , y para determinar dicho grado se aplica los conceptos de geometría diferencial : derivada y corchete de Lie. Para el diseño del controlador no lineal multivariable existen nuevas leyes de control de entrada  $v_i$ , los cuales son diseñadas por ubicación de polos del correspondiente subsistema lineal equivalente.

El control por realimentación de estado es bien entendible en el caso de sistemas lineales con múltiples entradas y múltiples salidas. Sin embargo para sistemas no lineales con múltiples entradas y múltiples salidas, el estudio se complica, requiriendo una investigación más profunda. Debido al reciente desarrollo

de máquinas con dinámicas no lineales de aplicación industrial, tales como los robots manipuladores, aeroplanos, submarinos, y vehículos espaciales, la comunidad científica le está dedicando actualmente bastante atención al campo de la ingeniería de control no lineal de procesos no lineales.

El proceso a controlar en este trabajo de tesis, es un robot manipulador esférico de dos grados de libertad (2 DOF). Para este propósito, se ha diseñado un controlador no lineal multivariable de 2 entradas y 2 salidas de tal manera que permita al robot seguir en forma asintótica una trayectoria cualquiera continua y diferenciable donde las entradas al sistema son los voltajes de armadura de los servomotores DC usados en las articulaciones y las salidas son las posiciones del robot ( $q_1$  y  $q_2$ ).

## 1.1 Formulación del Problema

Los manipuladores pueden ser construídos y modelados con muchos cuerpos rígidos conectados en serie por articulaciones rotativas y conducidas cada una de ellas por actuadores eléctricos, que son servomotores DC.

El robot manipulador esférico de dos grados de libertad está compuesto por un servomotor DC con encoder y una base de aluminio (disco y prisma) para el primer grado de libertad; otro servomotor DC también con encoder y una varilla rígida de aluminio que cumple la función de un brazo para el segundo grado de libertad. Dicho robot puede coger y trasladar objetos dentro de su espacio de acción.

Los propósitos de esta tesis son: el control simultáneo del movimiento rotacional de la base acoplada al eje del servomotor en posición vertical y el movimiento del brazo acoplado al eje del otro servomotor. Para este trabajo de tesis se ha diseñado un controlador no lineal MIMO (Multiple-input-Multiple-output) de 2 entradas y 2 salidas basado en el método de linealización exacta por realimentación de estado, de tal manera que el manipulador siga trayectorias de referencia arbitrarias empleando fuerzas de control adecuada.



## 1.2 Perspectiva Histórica

La linealización por realimentación tiene una gran atracción de investigación en los recientes años.

El método de linealización exacta en espacio-estado para sistemas de simple entrada fue propuesta por Brockett en 1978. Una solución para sistemas de múltiples entradas fue encontrado por Jakubcczyk-Respondek en 1980.

El problema de encontrar subsistemas en una simple entrada de un sistema no lineal fue encontrado y resuelto por Krener en 1983. La solución del problema correspondiente para sistemas de múltiples entradas fue encontrada por Marino en 1986. El problema de linealización exacta de un sistema no lineal con salidas es extensamente discutido por Cheng en 1988. Un completo análisis del problema para sistemas de múltiples salidas fue encontrado por Krener-Respondek en 1985.

La linealización exacta de la respuesta entrada-salida fue estudiada por Isidori-Rubeti en 1984. El comportamiento de la entrada-salida de un sistema fue estudiada en el libro de Isidori en 1985 [5].

## 1.3 Objetivos

El objetivo de este trabajo de tesis es resolver el problema de control de posición y seguimiento a una trayectoria dada, mediante la aplicación del método de linealización por realimentación de estado a un proceso no lineal multivariable.

La idea básica en linealización por realimentación de estado, es primero transformar un sistema no lineal en un sistema lineal y luego usar las técnicas de diseño de control lineal conocidas. En esta tesis se hace uso de la técnica de control no lineal MIMO (multiple input-multiple output) de linealización exacta por realimentación de estado para implementar un control de trayectoria para un robot manipulador esférico de 2DOF cuyos actuadores son dos servomotores DC.

## 1.4 Organización de la tesis

La tesis está organizada como sigue:

El modelado matemático del proceso del robot manipulador de 2 DOF es desarrollado en el Capítulo 2. denominado **Modelado de la Planta**. El modelo se ha ejecutado basado en el método de Lagrange-Euler y el cálculo de la ecuación

dinámica se ha determinado usando matemática simbólica del software MATLAB.

En el Capítulo 3. denominado **Linealización No Lineal** se presenta el fundamento teórico de la técnica de control no lineal para procesos SISO (simple entrada y simple salida) y para procesos MIMO (múltiple entrada y múltiple salida).

En el Capítulo 4. denominado **Implementación en Tiempo Real**, se presenta el hardware y el software del sistema.

En el Capítulo 5. denominado **Simulación del Sistema** se ilustran los resultados obtenidos de las simulaciones hechas en MATLAB. Las simulaciones han sido realizadas para el control de trayectoria constante y control de trayectoria senoidal. También se realiza las simulaciones aplicando la interfaz gráfica para el usuario (GUI) con MATLAB. Los programas se muestran en el apéndice B.

En el Capítulo 6 denominado **Resultados Experimentales** se presenta los resultados obtenidos de los experimentos referidos a trayectorias constantes y senoidales realizadas en pruebas de laboratorio, usando para ello la Tarjeta de Adquisición de Datos Lab-PC+.

En el Capítulo 7. denominado **Conclusiones** , se presenta las conclusiones y recomendaciones encontradas a lo largo del desarrollo teórico y práctico de este trabajo.

Por último se presentan los apéndices. En el apéndice A se presenta la identificación de parámetros de los servomotores DC; en el apéndice B se presenta el listado de programas en MATLAB y en el apéndice C se presenta el listado de programas en Borland C++.

## Capítulo 2

# Modelado de la Planta

### 2.1 Introducción

El modelado de la planta o proceso consiste en determinar la descripción matemática (usualmente un conjunto de ecuaciones diferenciales) del proceso a ser controlado, que en nuestro caso es el robot manipulador esférico. Tal modelado describe la dinámica del proceso y tiene por objetivo conocer la relación entre el movimiento del robot y las fuerzas implicadas en el mismo. Para determinar las ecuaciones del modelo usaremos el algoritmo de Lagrange-Euler [1], [2], [6], [9].

### 2.2 Descripción del Sistema

El Sistema Robot Manipulador Esférico de 2 Grados de Libertad que en adelante lo llamaremos SRM (Spherical Robot Manipulator) de 2DOF es mostrado en la figura 2.1, en la figura 2.2 se muestra una vista del esquema anterior con las medidas del robot, y en la figura 2.3 se muestra los sistemas de referencias usados. A continuación se describe los dos grados de libertad usados en el proceso.

- a. **Primer Grado de Libertad:** compuesto por un servomotor DC1 de 24 VDC con encoder óptico, y una base conformado por un disco y un prisma de aluminio; de tal manera que en la parte inferior del **disco** de radio **R** y altura **h**, irá acoplado con una bocina de bronce dicho servomotor en posición vertical y sobre la parte superior del disco irá un prisma de sección cuadrada **a**, y de altura **b**, a su vez en la parte superior de dicho prisma irá el otro servomotor en posición horizontal que corresponde al segundo grado de libertad.
- b. **Segundo Grado de Libertad:** compuesto por un servomotor DC2 de 24 VDC con encoder óptico, y una varilla rígida de aluminio, de tal manera que en el eje de dicho servomotor, en posición horizontal, irá acoplado con una bocina de bronce dicha varilla de longitud  $L_2$  que cumple la función de un brazo.

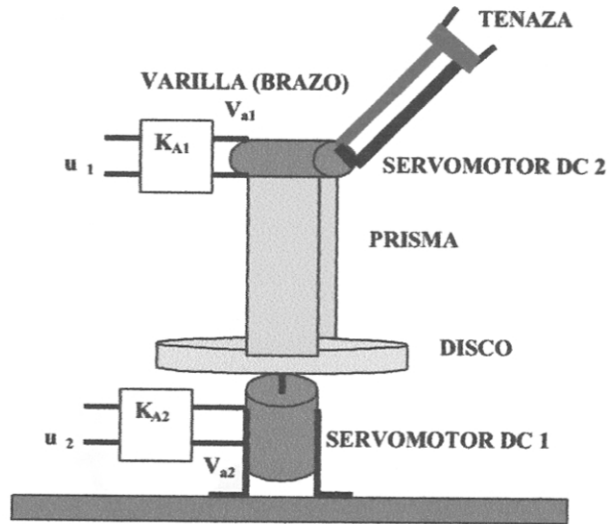


Figura 2.1: Esquema del Robot Manipulador Esférico de 2DOF

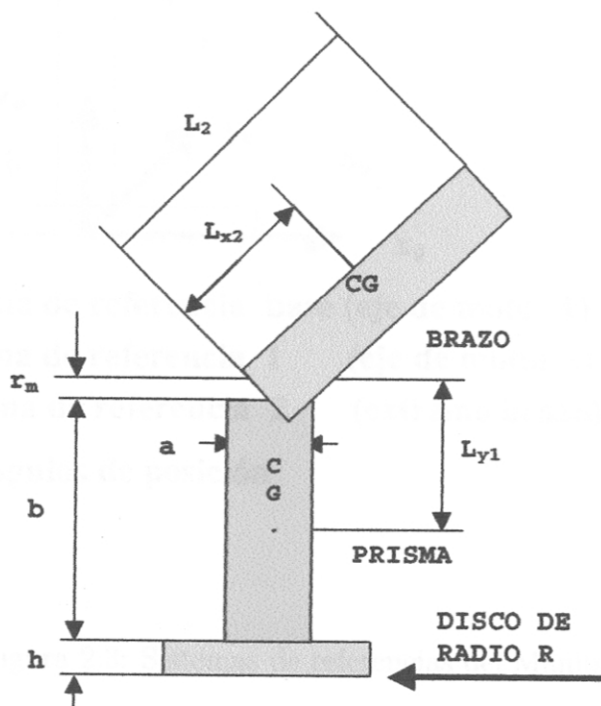
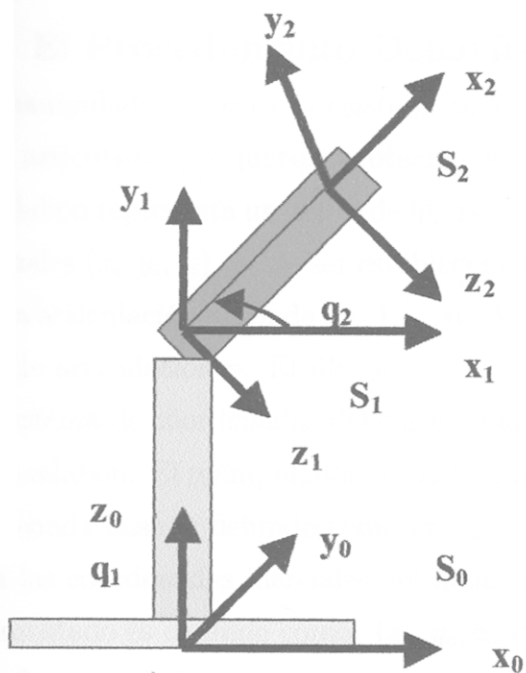


Figura 2.2: Esquema con las medidas del Manipulador



- $S_0$ : sistema de referencia base (eje de motor 1)**
- $S_1$ : sistema de referencia 1 (eje de motor 2)**
- $S_2$ : sistema de referencia 2 (extremo brazo)**
- $q_2$  y  $q_1$ : ángulos de posición**

Figura 2.3: Sistemas de referencias del Manipulador

## 2.3 El Modelo Dinámico

Para determinar la ecuación dinámica de un manipulador existen varios métodos tales como el método de Newton-Euler, el método de Lagrange-Euler, entre otros. Para el manipulador en estudio usaremos el método de Lagrange-Euler.

El método de Lagrange-Euler para el modelado de manipuladores robóticos emplea la fórmula Lagrangiana conjuntamente con la representación de Denavit-Hartenberg (D-H). Un completo estudio de dicho procedimiento puede ser encontrado en [1], [2], [6], [9].

### 2.3.1 El Procedimiento Denavit-Hartenberg

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

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

#### 2.3.1.1 Estableciendo el Sistema de Coordenadas D-H

Para establecer el sistema de coordenadas D-H  $(x_i, y_i, z_i)$ ,  $i = 0 \dots n$ , se usa las siguientes reglas:

1. El sistema de coordenadas base  $(x_o, y_o, z_o)$  se determina usando la regla de la mano derecha. El eje  $z_o$  se escoge libremente y está localizado a lo largo del eje del movimiento del primer eslabón.

del eje del movimiento del primer eslabón.

2. El eje  $z_{i-1}$  es alineado con el eje del movimiento (giratorio o prismático) de la  $i$ -ésima articulación.
3. El origen del sistema de coordenadas  $i$ -ésima está en la intersección del eje  $z_i$  y  $z_{i-1}$  o en la intersección de la normal común entre los ejes de dos articulaciones consecutivas  $z_i$  y  $z_{i-1}$  y el eje  $z_i$ .
4. El eje  $x_i$  es perpendicular al eje  $z_{i-1}$ .
5. Aplicar la regla de la mano derecha en la articulación  $i$ -ésima para determinar  $y_i$ .
6. Generalmente, la articulación  $n$ -ésima es una articulación giratoria. El sistema de coordenadas  $(x_n, y_n, z_n)$ , puede estar en cualquier parte del efector final con tal de que el eje  $z_n$  esté a lo largo de la dirección del eje  $z_{i-1}$  y apuntando hacia afuera del manipulador. El eje  $x_n$  es perpendicular a ambos ejes:  $z_n$  y  $z_{n-1}$ .

### 2.3.1.2 Parámetros D-H

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

1.  $\theta_i$ , es el ángulo de rotación del eje  $x_{i-1}$  con el eje  $x_i$  en el plano del eje  $z_{i-1}$ . Este es variable si la articulación es giratoria.
2.  $d_i$ , es la distancia del origen del sistema de coordenadas  $(i-1)$ -ésima a lo largo del eje  $z_{i-1}$ , hasta la intersección del eje  $z_{i-1}$  con el eje  $x_i$ . Esta distancia  $d_i$  es variable si la articulación  $i$  es prismática.
3.  $a_i$ , es la distancia a lo largo del eje  $x_i$ , que va desde la intersección del eje  $z_{i-1}$  con el eje  $x_i$  hasta el origen del sistema de coordenadas  $i$ -ésima en el caso de articulaciones giratorias.
4.  $\alpha_i$ , es el ángulo de separación entre los ejes  $z_{i-1}$  y  $z_i$  medido en un plano perpendicular al eje  $x_i$ , utilizando la regla de la mano derecha.

### 2.3.1.3 Matriz de Transformación Homogénea D-H una con respecto al

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

1.  $T(z, \theta_i)$ : rotación de un ángulo  $\theta_i$  alrededor del eje  $z_{i-1}$ , para alinear el eje  $x_{i-1}$  con el eje  $x_i$ .
2.  $T(z, d_i)$ : traslación a lo largo del eje  $z_{i-1}$  de una distancia  $d_i$ , hasta hacer coincidir el eje  $z_{i-1}$  con el eje  $x_i$ .
3.  $T(x, a_i)$ : traslación a lo largo del eje  $x_i$  de una distancia  $a_i$ , que va desde la intersección del eje  $z_{i-1}$  y el eje  $x_i$  al origen del sistema de coordenadas  $i$ -ésimo.
4.  $T(x, \alpha_i)$ : rotación de un ángulo  $\alpha_i$ , del eje  $z_{i-1}$  con el eje  $z_i$  alrededor del eje  $x_i$ .

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

$$A_i^{i-1} = T(z, \theta_i)T(0, 0, d_i)T(a_i, 0, 0)T(x, \alpha_i) =$$

$$\begin{bmatrix} c(\theta_i) & -s(\theta_i) & 0 & 0 \\ s(\theta_i) & c(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c(\alpha_i) & -s(\alpha_i) & 0 \\ 0 & s(\alpha_i) & c(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

donde  $c$  y  $s$ , son las funciones *cos* y *sen* respectivamente. La siguiente matriz de transformación homogénea:

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



determina la localización del sistema de coordenada ( $i$ )-ésima con respecto al sistema de coordenada base.

### 2.3.2 El Algoritmo de Lagrange-Euler

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

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

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

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

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

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

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

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

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

$$J_i = \begin{bmatrix} \frac{-I_{xx_i} + I_{yy_i} + I_{zz_i}}{2} & -I_{x_i y_i} & -I_{x_i z_i} & m_i \bar{x}_i \\ -I_{x_i y_i} & \frac{I_{xx_i} - I_{yy_i} + I_{zz_i}}{2} & -I_{y_i z_i} & m_i \bar{y}_i \\ -I_{x_i z_i} & -I_{y_i z_i} & \frac{I_{xx_i} + I_{yy_i} - I_{zz_i}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (2.5)$$

donde:  $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$  es el centro de gravedad (CG) del eslabón  $i$  con respecto al sistema de coordenadas  $i$ -ésimo,  $I_{xx_i}, I_{yy_i}, I_{zz_i}$  son los momentos inercia con respecto al sistema de coordenadas  $(x_i, y_i, z_i)$ ,  $m_i$  es la masa del cuerpo  $i$ ,  $\bar{x}_i, \bar{y}_i, \bar{z}_i$  son las distancias del CG del cuerpo  $i$  al sistema de coordenadas  $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$ , e  $I_{x_i y_i}, I_{y_i z_i}, I_{x_i z_i}$  son los productos correspondientes del momento de inercia.

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

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

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

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

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

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

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

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

$$d_i = \sum_{j=1}^n (-m_j g U_{ji} \bar{r}_j^j) \quad i = 1, 2, \dots, n \quad (2.9)$$

donde,  $g = [g_{x0} \ g_{y0} \ g_{z0} \ 0]$  es el vector gravedad fila expresado en el sistema de coordenadas base y  $\bar{r}_j^j = [\bar{x}_j \ \bar{y}_j \ \bar{z}_j \ 1]^T$  (el elemento 1 es un factor de escala) es el vector centro de masa del eslabón  $j$  y expresado en el sistema de coordenadas  $j$ -ésimo.

Finalmente la ecuación dinámica del manipulador, con el método L-E toma la forma:

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

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

## 2.4 Aplicación y Programación del Modelo

### 2.4.1 Lista de variables

Todas las variables que intervienen en el sistema se describen en la tabla 2.1, con sus respectivos valores que se han obtenido y se explican en el apéndice A (Identificación de Parámetros). En dicha tabla se usan las siguientes abreviaciones: M.I. (Momento de Inercia), C.F.V. (Coeficiente de fricción viscosa) y F.E.M. (Fuerza electromotriz).

### 2.4.2 Cálculos

Para el cálculo de la ecuación dinámica del manipulador se aplica el algoritmo de Lagrange-Euler. A continuación se presenta los siguientes cálculos que usaremos para la programación de dicho modelo con matemática simbólica de MATLAB 6.0 [11].

#### 2.4.2.1 Cálculo de los Parámetros D-H

Primeramente se fijan los sistemas de referencia en las articulaciones como se muestra en la figura 2.3, posteriormente se determina los parámetros D-H (DENAVID-HARTENBERG) [1], [2], [6], [9], que se muestran en la Tabla 2.2.

#### 2.4.2.2 Cálculo de las Matrices de Transformación Homogénea

Reemplazando los valores de la tabla 2.2 en la ecuación (2.1), obtenemos las matrices de transformación homogénea  $A_1^0$  y  $A_2^1$ :

$A_1^0$ : posición y orientación del sistema de referencia 1 ( $S_1$ ) de la base con respecto al sistema de referencia base ( $S_0$ ):

Tabla 2.1: Valores de las Variables

Símbolo	Descripción	Valor
$m_d$	Masa del disco	0.55 kg
$m_p$	Masa del prisma	1.00 kg
$m_2$	Masa del brazo	0.1 kg
$h$	Altura del disco	0.01 m
$R$	Radio del disco	0.07 m
$b$	Altura del prisma	0.20 m
$a$	Ancho, largo de sección del prisma	0.044 m
$L_2$	Longitud del brazo	0.2 m
$L_{x_2}$	Longitud del eje de $S_1$ al CG brazo	0.10 m
$L_{c_2}$	Longitud del eje de $S_2$ al CG brazo	$(L_2 - L_{x_2})$ m
$I_1$	Momento de inercia del sólido (disco más prisma)	0.0016 $kgm^2$
$I_2$	Momento de inercia del brazo	0.0013 $kgm^2$
$J_{m1}$ y $J_{m2}$	M.I. servomotor DC1 y DC2	0.00059 $kgm^2$
$B_{m1}$ y $B_{m2}$	C.F.V. servomotor DC1 y DC2	0.0001415 $Nms/rad$
$J_{eq1}$	M.I. equivalente del servom. DC1 de la base	0.2177 $kgm^2$
$B_{eq1}$	C.F.V. equivalente del servom. DC1 de la base	0.054 $Nms/rad$
$J_{eq2}$	M.I. equivalente del servom. DC2 del brazo	0.3954 $kgm^2$
$B_{eq2}$	C.F.V. equivalente del servom. DC2 del brazo	0.0589 $Nms/rad$
$n_1$ y $n_2$	Reducción servomotor DC1 y DC2	18.5
$R_{a1}$	Resistencia armadura servomotor DC1	3.50 Ohmios
$R_{a2}$	Resistencia armadura servomotor DC2	3.50 Ohmios
$L_{a1}$	Inductancia armadura servomotor DC1	Despreciable
$L_{a2}$	Inductancia armadura servomotor DC2	Despreciable
$V_{b1}$ y $V_{b2}$	F.E.M. servomotor DC1 y DC2	21.56 V
$V_{a1}$ y $V_{a2}$	Voltage armadura servomotor DC1 y DC2	24 V
$I_{a1}$	Corriente armadura servomotor DC1	0.7 A
$I_{a2}$	Corriente armadura servomotor DC2	0.7 A
$K_{A1}$ y $K_{A2}$	Constante ganancia driver 1 y 2	2.17
$K_{m1}$ y $K_{m2}$	Constante torque servomotor DC1 y DC2	0.04364 $N - m/A$
$K_{b1}$ y $K_{b2}$	Constante fem servomotor DC1 y DC2	0.04364 $V/rad/s$
$u_1$	Señal de control servomotor DC1	(En simulación) V
$u_2$	Señal de control servomotor DC2	(En simulación) V
$g$	Aceleración de la gravedad	9.81 $m/seg^2$
$r_m$	Dist. del extremo superior del prisma al eje DC2	0.03 m
$C_c$	Coefficiente de fricción de Coulomb	0.001488 $Kgm^2/seg$

$$A_1^0 = \begin{bmatrix} \cos(q_1) & 0 & \sin(q_1) & 0 \\ \sin(q_1) & 0 & -\cos(q_1) & 0 \\ 0 & 1 & 0 & b + h + r_m \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Tabla 2.2: Valores de los Parámetros  $D - H$

Juntura	$\theta_i$	$d_i$	$a_i$	$\alpha_i$
1	$q_1$	$b + h + r_m$	0	$90^\circ$
2	$q_2$	0	$L_2$	$0^\circ$

$A_2^1$ : posición y orientación del sistema de referencia 2 ( $S_2$ ) del brazo con respecto al sistema de referencia 1 ( $S_1$ ):

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

$A_2^0$ : posición y orientación del sistema de referencia 2 ( $S_2$ ) del brazo con respecto al sistema de referencia base ( $S_0$ ):

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

$A_2^0$  es determinado mediante matemática simbólica en el apéndice B.

#### 2.4.2.3 Cálculo del momento de inercia del disco [3], [8]

El momento de inercia del disco respecto al sistema de referencia 1 ( $S_1$ )  $x_1$ ,  $y_1$ ,  $z_1$ , (ver figura 2.4), está dado por:

$$\begin{aligned} I_{x_1d} &= 0 \\ I_{y_1d} &= \frac{m_d R^2}{2} \\ I_{z_1d} &= 0 \end{aligned}$$

y los productos de los momentos de inercia son:

$$I_{x_1d y_1d} = I_{x_1d z_1d} = I_{y_1d z_1d} = 0$$

#### 2.4.2.4 Cálculo del momento de inercia del prisma [3], [8]

El momento de inercia del prisma respecto al sistema de referencia 1 ( $S_1$ )  $x_1$ ,  $y_1$ ,  $z_1$ , (ver figura 2.5), está dado por:

$$\begin{aligned} I_{x_1p} &= 0 \\ I_{y_1p} &= \frac{m_p a^2}{6} \end{aligned}$$

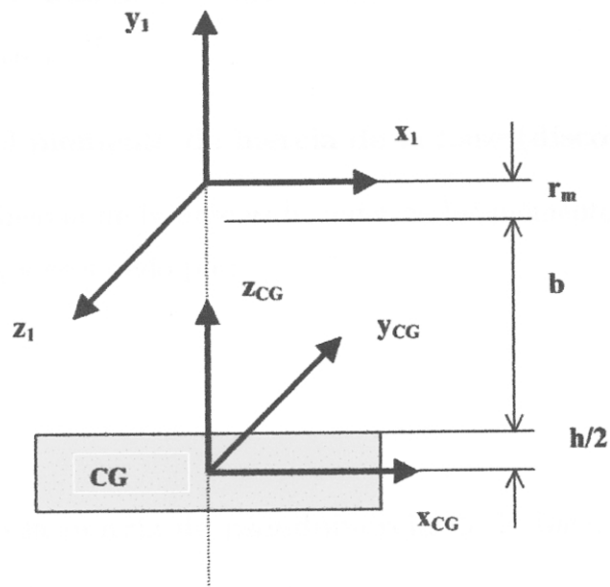


Figura 2.4: Momento de inercia del disco respecto al sistema de referencia 1 ( $S_1$ ).

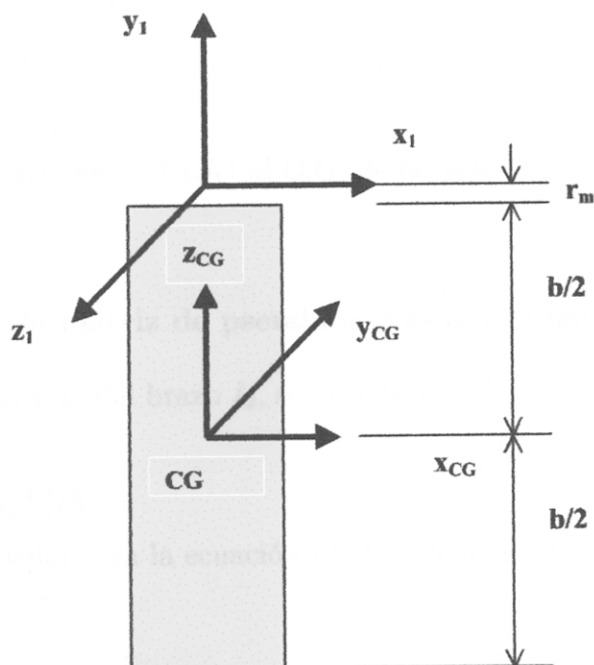


Figura 2.5: Momento de inercia del prisma respecto al sistema de referencia 1 ( $S_1$ )

$$I_{z_1p} = 0$$

y los productos de los momentos de inercia son:

$$I_{x_1d y_1d} = I_{x_1d z_1d} = I_{y_1d z_1d} = 0$$

#### 2.4.2.5 Cálculo del momento de inercia de la base (disco más prisma)

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

$$\begin{aligned} I_{x_1} &= 0 \\ I_{y_1} &= \frac{m_p a^2}{6} + \frac{m_d R^2}{2} \\ I_{z_1} &= 0 \end{aligned} \tag{2.10}$$

#### 2.4.2.6 Cálculo de la matriz de pseudoinercia de la base $J_1$

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

$$J_1 = \begin{bmatrix} I_1 & 0 & 0 & 0 \\ 0 & -I_1 & 0 & -m_1 L_{y_1} \\ 0 & 0 & I_1 & 0 \\ 0 & -m_1 L_{y_1} & 0 & m_1 \end{bmatrix} \tag{2.11}$$

donde:

$$m_1: m_d + m_p + m_{servodc1} + m_{accesorios}$$

$L_{y_1}$ : long. del eje de referencia 1 ( $S_1$ ) al C.G. de la base

$$I_1 = \frac{I_{y_1}}{2}$$

#### 2.4.2.7 Cálculo de la matriz de pseudoinercia del brazo $J_2$

El momento de inercia del brazo  $I_2$ , se obtiene de [3], [8]:

$$I_{xx_2} = 0$$

$$I_2 = I_{yy_2} = I_{zz_2} = m_2 L_2^2 / 3$$

reemplazando estos valores en la ecuación (2.5), obtenemos la matriz de pseudoinercia del brazo  $J_2$ :

$$J_2 = \begin{bmatrix} I_2 & 0 & 0 & -m_2 L_{x_2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -m_2 L_{x_2} & 0 & 0 & m_2 \end{bmatrix} \tag{2.12}$$

donde:

$m_2$ :  $m_{brazo} + m_{objeto}$

$L_{x_2}$ : long. del eje de referencia 1 ( $S_1$ ) al C.G. del brazo

### 2.4.3 Programación del Algoritmo Lagrange-Euler

El programa de cálculo de la ecuación dinámica del robot manipulador esférico de dos grados de libertad (2 DOF) se determina aplicando Matemática Simbólica de MATLAB 6.0 [11] y se presenta en el apéndice B. La ecuación dinámica obtenida para los dos grados de libertad son:

$$\text{para el primer grado de libertad: } T_1 = H_{11}\ddot{q}_1 + C_1 \quad (2.13)$$

$$\text{para el segundo grado de libertad: } T_2 = H_{22}\ddot{q}_2 + C_2 + d_2 \quad (2.14)$$

que son los elementos de  $T_i$ , dado en la ecuación (2.10).

### 2.4.4 El Modelo de los Actuadores

#### 2.4.4.1 Modelado del Actuador No. 1 de la Base del Manipulador

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

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

$$V_{A1} = K_{A1}u_1 = L_{a1}\dot{I}_{a1} + R_{a1}I_{a1} + V_{b1} \quad (2.15)$$

la fuerza electromotriz del servomotor DC es:

$$V_{b1} = K_{b1}w_{m1} = K_{b1}n_1\dot{q}_1 \quad (2.16)$$

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

$$T_{m1} = K_{m1}I_{a1} = J_{m1}n_1\ddot{q}_1 + B_{m1}n_1\dot{q}_1 + T_{g1} \quad (2.17)$$

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

$$T_{g2} = n_1T_{g1} = J_{g1}\ddot{q}_1 + B_{g1}\dot{q}_1 + T_1 + n_1T_{C1} \quad (2.18)$$

donde  $T_1$  es el torque de la carga y  $T_{C1}$  es el torque de fricción de Coulomb. La (2.18) en (2.17) queda:

$$n_1K_{m1}I_{a1} = J_{eq1}\ddot{q}_1 + B_{eq1}\dot{q}_1 + T_1 + n_1T_{C1} \quad (2.19)$$

donde:



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

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

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

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

$$C_1 = -a_2 \dot{q}_1 \dot{q}_2 \sin(2q_2)$$

$$a_1 = 2I_1$$

$$a_2 = I_2 + 2m_2 L_2 L_{c2} - m_2 L_2^2$$

despreciando la inductancia de armadura o sea  $L_{a1} = 0$ , se obtiene de (2.15) y (2.16)

$$K_{A1} u_1 = R_{a1} I_{a1} + K_{b1} n_1 \dot{q}_1 \quad (2.20)$$

despejando  $I_{a1}$  en (2.19) y reemplazando en (2.20), se obtiene la ley de control para la primera entrada:

$$u_1 = R_{T1} (J_{eq1} \ddot{q}_1 + B_{eq1} \dot{q}_1 + T_1) + N_{T1} \dot{q}_1 + n_1 R_{T1} T_{C1} \quad (2.21)$$

donde:

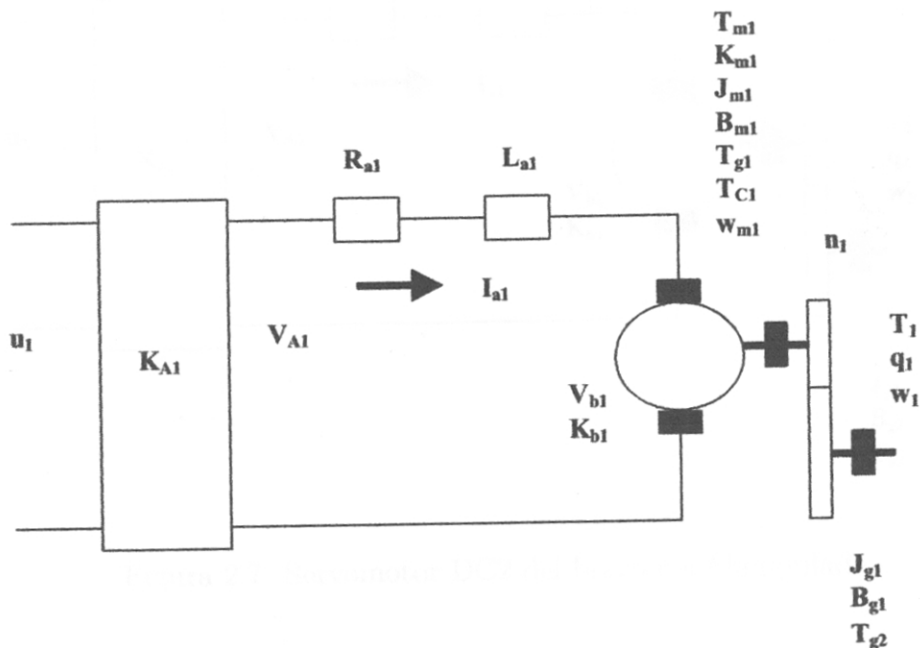


Figura 2.6: Servomotor DC1 de la base del Manipulador

$$R_{T1} = \frac{R_{a1}}{n_1 K_{m1} K_{A1}}$$

$$N_{T1} = \frac{n_1 K_{b1}}{K_{A1}}$$

$$T_{C1} = C_c \text{sign}(\dot{q}_1)$$

#### 2.4.4.2 Modelado del Actuador No. 2 del Brazo del Manipulador

En la figura 2.7, se muestra el servomotor DC2 correspondiente al brazo. En la misma figura se muestra el circuito eléctrico del servomotor DC2, controlado también por la armadura. Dicho servomotor se encuentra ubicado en la parte superior del prisma. Para obtener la ley de control para la segunda entrada se procede de la misma manera que para el primer servomotor, obteniendo de esta manera:

$$u_2 = R_{T2}(J_{eq2}\ddot{q}_2 + B_{eq2}\dot{q}_2 + T_2) + N_{T2}\dot{q}_2 + n_2 R_{T2} T_{C2} \quad (2.22)$$

donde:

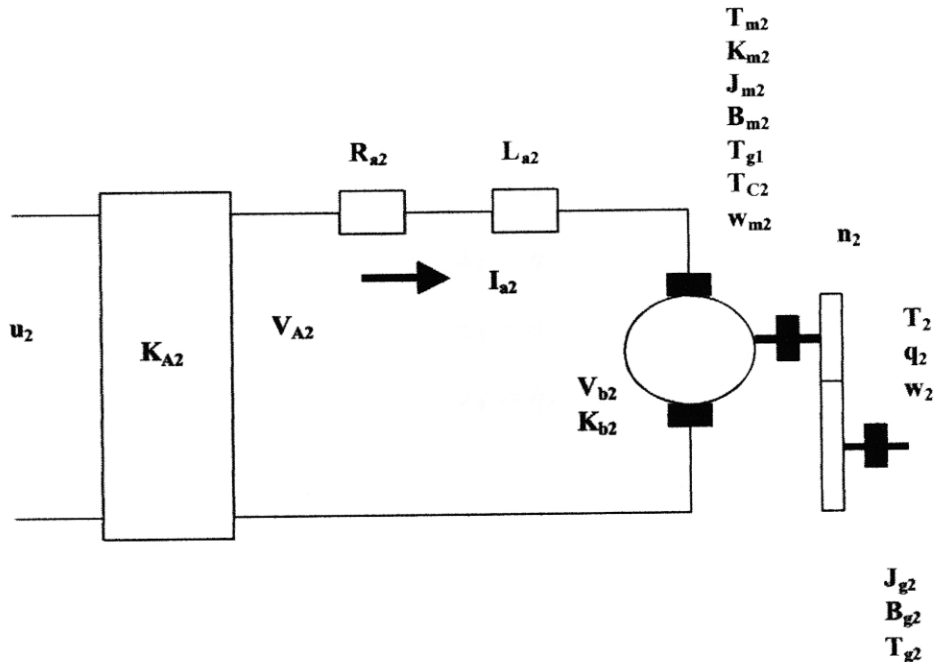


Figura 2.7: Servomotor DC2 del brazo del Manipulador

$$T_{C2} = C_c \text{sign}(\dot{q}_2)$$

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

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

$$R_{T2} = \frac{R_{a2}}{n_2 K_{m2} K_{A2}} \quad (2.27)$$

$$N_{T2} = \frac{n_2 K_{b1}}{K_{A2}}$$

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

$$H_{22} = a_2 \quad (2.28)$$

$$C_2 = a_2 \dot{q}_1^2 \sin(q_2) \cos(q_2)$$

$$d_2 = a_3 \cos(q_2)$$

$$a_2 = I_2 + 2m_2 L_2 L_{c2} - m_2 L_2^2$$

$$a_3 = m_2 g L_{c2}$$

## 2.4.5 Modelo en el Espacio Estado

La ecuación dinámica del manipulador obtenida en las ecuaciones (2.13) y (2.14), se representa en el espacio de estados mediante la siguiente asignación de variables:

$$x_1 = q_1 \quad (2.23)$$

$$x_2 = q_2 \quad (2.24)$$

$$x_3 = \dot{q}_1 \quad (2.25)$$

$$x_4 = \dot{q}_2 \quad (2.26)$$

donde:

$x_1$  = posición angular de la base

$x_2$  = posición angular del brazo

$x_3$  = velocidad angular de la base

$x_4$  = velocidad angular del brazo

La ecuación no lineal para sistemas multivariables es:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \quad (2.27)$$

su representación en forma matricial:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ f_4(x) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ G_{31} & 0 \\ 0 & G_{42} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (2.28)$$

reemplazando las ecuaciones (2.23) y (2.25) en la ecuación (2.21), se obtiene  $u_1$  en variables de estado:

$$u_1 = R_{T1}(J_{eq1}\dot{x}_3 + B_{eq1}x_3 + T_1) + N_{T1}x_3 + C_c n_1 R_{T1} \text{sign}(x_3) \quad (2.29)$$

así mismo, reemplazando las ecuaciones (2.24) y (2.26) en la ecuación (2.22), se obtiene  $u_2$  en variables de estado:

$$u_2 = R_{T2}(J_{eq2}\dot{x}_4 + B_{eq2}x_4 + T_2) + N_{T2}x_4 + C_c n_2 R_{T2} \text{sign}(x_4) \quad (2.30)$$

la ecuación dinámica del manipulador obtenida en las ecuaciones (2.13) y (2.14), también se representa en variables de estado:

$$T_1 = H_{11}\dot{x}_3 + C_1 \quad (2.31)$$

$$T_2 = H_{22}\dot{x}_4 + C_2 + d_2 \quad (2.32)$$

posteriormente, reemplazando la ecuación (2.31) en (2.29), se obtiene la primera entrada  $u_1$ :

$$u_1 = R_{T1}(J_{eq1}\dot{x}_3 + B_{eq1}x_3 + H_{11}\dot{x}_3 + C_1) + N_{T1}x_3 + C_c n_1 R_{T1} \text{sign}(x_3) \quad (2.33)$$

finalmente, reemplazando la ecuación (2.32) en (2.30), se obtiene la segunda entrada  $u_2$ :

$$u_2 = R_{T2}(J_{eq2}\dot{x}_4 + B_{eq2}x_4 + H_{22}\dot{x}_4 + C_2 + d_2) + N_{T1}x_4 + C_c n_2 R_{T2} \text{sign}(x_4) \quad (2.34)$$

así mismo de (2.33) se obtiene:

$$\dot{x}_3 = \frac{u_1 - R_{T1}B_{eq1}x_3 - N_{T1}x_3 - R_{T1}C_1 - C_c n_1 R_{T1} \text{sign}(x_3)}{R_{T1}(J_{eq1} + H_{11})} \quad (2.35)$$

y de (2.34) se obtiene:

$$\dot{x}_4 = \frac{u_2 - R_{T2}B_{eq2}x_4 - N_{T2}x_4 - R_{T2}(C_2 + d_2) - C_c n_2 R_{T2} \text{sign}(x_4)}{R_{T2}(J_{eq2} + H_{22})} \quad (2.36)$$

finalmente reemplazando las ecuaciones (2.25), (2.26), (2.35) y (2.36), en la ecuación (2.28) :

$$f_1(x) = x_3$$

$$f_2(x) = x_4$$

$$f_3(x) = \frac{-(R_{T1}B_{eq1} + N_{T1})x_3 - R_{T1}C_1 - C_c n_1 R_{T1} \text{sign}(x_3)}{R_{T1}(J_{eq1} + H_{11})}$$

$$f_4(x) = \frac{-(R_{T2}B_{eq2} + N_{T2})x_4 - R_{T2}(C_2 + d_2) - C_c n_2 R_{T2} \text{sign}(x_4)}{R_{T2}(J_{eq2} + H_{22})}$$

$$G_{31} = \frac{1}{R_{T1}(J_{eq1} + H_{11})}$$

$$G_{42} = \frac{1}{R_{T2}(J_{eq2} + H_{22})}$$

Como el proceso tiene dos salidas se tiene:

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (2.37)$$

## Capítulo 3

# Linealización No Lineal

### 3.1 Herramientas de Geometría Diferencial

Los métodos de geometría diferencial son principalmente aplicados a procesos no lineales para determinar si tales procesos son linealizables por realimentación de estado; esto es si la descripción dinámica de un proceso en particular puede ser linealizada después de una transformación adecuada de coordenadas y la realimentación del vector de estado al proceso no lineal. De acuerdo a la terminología usada en la geometría diferencial, llamaremos a una función vectorial  $n$  dimensional:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{bmatrix} = \begin{bmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_n(x_1, \dots, x_n) \end{bmatrix} \quad (3.1)$$

un campo vectorial, si para cada función vectorial  $\mathbf{f}$  le corresponde un campo de vectores en un espacio  $n$  dimensional  $\mathbf{R}^n$ . Posteriormente, solo trataremos con campo vectoriales suaves de la forma  $\mathbf{f}(\mathbf{x})$ , lo que significa que tales vectores poseen derivadas parciales continuas de cualquier orden. Suavidad también se aplica a funciones.

El gradiente de una función escalar suave  $h(\mathbf{x})$  es definida como:

$$\nabla h(\mathbf{x}) = \frac{\partial h}{\partial \mathbf{x}} = \left[ \frac{\partial h(x)}{\partial x_1} \quad \dots \quad \frac{\partial h(x)}{\partial x_n} \right] \quad (3.2)$$

Los elementos de  $\nabla h$ , son:

$$(\nabla h)_i = \frac{\partial h}{\partial x_i}$$

Similarmente, el Jacobiano de  $\mathbf{f}(\mathbf{x})$ , de elementos:  $(\nabla \mathbf{f})_{ij} = \partial \mathbf{f}_i / \partial x_j$ , es definido como:

$$\nabla \mathbf{f}(\mathbf{x}) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \dots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \dots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix} \quad (3.3)$$

### 3.1.1 Derivada de Lie y Corchete de Lie

#### Derivada de Lie

La derivada de Lie de una función escalar  $h(\mathbf{x})$  con respecto a un campo vectorial  $\mathbf{f}(\mathbf{x})$  es una nueva función escalar  $L_{\mathbf{f}}h$  definida como:

$$L_{\mathbf{f}}h = \nabla h \mathbf{f} = \frac{\partial h}{\partial x_1} f_1 + \dots + \frac{\partial h}{\partial x_n} f_n \quad (3.4)$$

Obsérvese que la derivada de Lie es el producto interno entre  $\nabla h(\mathbf{x})$  y  $\mathbf{f}(\mathbf{x})$ . Las derivadas de Lie repetidas pueden ser formuladas en forma recursiva:

$$L_{\mathbf{f}}^0 h = h$$
$$L_{\mathbf{f}}^i h = L_{\mathbf{f}}(L_{\mathbf{f}}^{i-1} h) = \nabla(L_{\mathbf{f}}^{i-1} h) \mathbf{f}, \quad \text{para } i=1,2,\dots \quad (3.5)$$

Si  $\mathbf{f}$  y  $\mathbf{g}$  son campos vectoriales, luego:

$$L_{\mathbf{g}} L_{\mathbf{f}} = \nabla(L_{\mathbf{f}}) \mathbf{g} \quad (3.6)$$

#### Corchete de Lie

El corchete de Lie de dos campos vectoriales  $\mathbf{f}$  y  $\mathbf{g}$  es otro campo vectorial  $[\mathbf{f}, \mathbf{g}]$ , definido como:

$$[\mathbf{f}, \mathbf{g}](x) = \nabla \mathbf{g} \mathbf{f} - \nabla \mathbf{f} \mathbf{g} \quad (3.7)$$

La notación  $ad_{\mathbf{f}} \mathbf{g}$  es frecuentemente usado en vez de  $[\mathbf{f}, \mathbf{g}]$ . Los corchetes de Lie repetidos pueden ser formulados en forma recursiva:

$$ad_{\mathbf{f}}^0 \mathbf{g} = \mathbf{g}$$
$$ad_{\mathbf{f}}^i \mathbf{g} = [\mathbf{f}, ad_{\mathbf{f}}^{i-1} \mathbf{g}], \quad \text{para } i=1,2,\dots \quad (3.8)$$

Si  $\mathbf{f}$ ,  $\mathbf{f}_1$ ,  $\mathbf{f}_2$ ,  $\mathbf{g}$ ,  $\mathbf{g}_1$ ,  $\mathbf{g}_2$ , son campo vectoriales suaves,  $\alpha_1$  y  $\alpha_2$  son constantes escalares y  $h(\mathbf{x})$  es una función escalar suave de  $\mathbf{x}$ , podemos enunciar las siguientes propiedades del corchete de Lie:

1) Bilinealidad:

$$[\alpha_1 \mathbf{f}_1 + \alpha_2 \mathbf{f}_2, \mathbf{g}] = \alpha_1 [\mathbf{f}_1, \mathbf{g}] + \alpha_2 [\mathbf{f}_2, \mathbf{g}]$$

$$[\mathbf{f}, \alpha_1 \mathbf{g}_1 + \alpha_2 \mathbf{g}_2] = \alpha_1 [\mathbf{f}, \mathbf{g}_1] + \alpha_2 [\mathbf{f}, \mathbf{g}_2] \quad (3.9)$$

2) Anticonmutatividad:

$$[\mathbf{f}, \mathbf{g}] = -[\mathbf{g}, \mathbf{f}] \quad (3.10)$$

3) Identidad de Jacobi:

$$L_{ad_{\mathbf{f}}}\mathbf{g}h = L_{\mathbf{f}}L_{\mathbf{g}}h = L_{\mathbf{g}}L_{\mathbf{f}}h \quad (3.11)$$

### 3.1.2 Difeomorfismo y Transformación de Coordenadas

Una función  $\Phi$  definida en una región  $\Omega$  es conocida como un difeomorfismo si ambos,  $\Phi$  y su inversa  $\Phi^{-1}$  son funciones suaves. Si la región  $\Omega$  es igual a  $\mathbf{R}^n$  (el espacio total), entonces  $\Phi(\mathbf{x})$  es un difeomorfismo global. Por otro lado un difeomorfismo local es definido solo en un campo finito de un punto dado  $\mathbf{x}_o$ .

Una función suave  $\Phi(\mathbf{x})$  definida en una región  $\Omega$  en  $\mathbf{R}^n$  es un difeomorfismo local si la matriz Jacobiana  $\nabla_{\mathbf{x}}$  es no singular en un punto  $\mathbf{x} = \mathbf{x}_o$  de la región  $\Omega$ .

Empleando un difeomorfismo, podemos transformar los estados de un proceso no lineal dentro de un nuevo conjunto de estados correspondientes a la transformación de un proceso no lineal. Consideremos un proceso no lineal SISO (single-input-single-output) descrito por:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u, \quad y = h(\mathbf{x}) \quad (3.12)$$

donde la entrada simple es  $u$  y la salida simple es  $y$ . Definiremos ahora un nuevo conjunto de estados  $\mathbf{z} = \Phi(\mathbf{x})$ . Ejecutando la diferenciación en  $\mathbf{z}$ :

$$\dot{\mathbf{z}} = \frac{\partial \Phi}{\partial \mathbf{x}} \dot{\mathbf{x}} = \frac{\partial \Phi}{\partial \mathbf{x}} (\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u) = \mathbf{f}^*(\mathbf{z}) + \mathbf{g}^*(\mathbf{z})u, \quad y = h^*(\mathbf{z}) \quad (3.13)$$

donde:

$$\mathbf{f}^* = \frac{\partial \Phi}{\partial \mathbf{x}} \mathbf{f}$$

$$\mathbf{g}^* = \frac{\partial \Phi}{\partial \mathbf{x}} \mathbf{g}$$

$$\mathbf{x} = \Phi^{-1}(\mathbf{z})$$

$$h^* = h$$



### 3.1.3 El Teorema de Frobenius

Sea  $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m$  un conjunto de campos vectoriales linealmente independientes en  $\mathbf{R}^n$ . El Teorema de Frobenius establece que tal conjunto es completamente integrable sí y sólo sí es involutivo, según se explica posteriormente.

El conjunto de campos vectoriales definidos anteriormente completamente integrable sí y sólo sí existen  $(n - m)$  funciones escalares  $h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_{n-m}(\mathbf{x})$ , que satisfacen el siguiente sistema de ecuaciones diferenciales parciales:

$$\nabla h_i \mathbf{f}_j = 0, \quad 1 \leq i \leq n - m, \quad 1 \leq j \leq m \quad (3.14)$$

donde los gradientes  $\nabla h_1, \nabla h_2, \dots, \nabla h_{n-m}$ , son también linealmente independientes.

El conjunto de campos vectoriales considerado es llamado a ser involutivo sí y sólo sí hay funciones escalares  $\alpha_{ijk}$  en  $\mathbf{R}$  (el espacio unidimensional) satisfaciendo:

$$[\mathbf{f}_i, \mathbf{f}_j](\mathbf{x}) = \sum_{k=1}^m \alpha_{ijk}(\mathbf{x}) \mathbf{f}_k(\mathbf{x}), \quad \forall i, j \quad (3.15)$$

La última ecuación indica que el campo vectorial resultante  $[\mathbf{f}_i, \mathbf{f}_j](\mathbf{x})$  es una combinación lineal del conjunto original de campos vectoriales  $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m$ . De (3.15), tenemos los siguientes postulados:

- 1) Los campos vectoriales constantes son siempre involutivos porque el corchete de Lie para dos vectores  $\mathbf{f}$  y  $\mathbf{g}$  resulta  $[\mathbf{f}, \mathbf{g}] = 0$ , el cual puede ser representado como una combinación lineal de los campos vectoriales  $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m$ .
- 2) Un conjunto compuesto por un sólo campo vectorial  $\mathbf{f}$  es involutivo porque  $[\mathbf{f}, \mathbf{f}] = 0$ .
- 3) El conjunto de los campos vectoriales  $\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m$  es involutivo si  $\forall \mathbf{x}$  y  $\forall i, j$ :

$$rango[\mathbf{f}_1(\mathbf{x}) \dots \mathbf{f}_m(\mathbf{x})] = rango[\mathbf{f}_1(\mathbf{x}) \dots \mathbf{f}_m(\mathbf{x}), [\mathbf{f}_i(\mathbf{x}), \mathbf{f}_j(\mathbf{x})]]$$

donde la notación  $rango [.]$  representa el rango de la matriz formada por la columna de los campos vectoriales.

El sistema (3.15) puede ser reescrito en forma matricial como sigue:

$$[\nabla h_1, \nabla h_2, \dots, \nabla h_{n-m}] \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \dots \\ \mathbf{f}_m \end{bmatrix} = \mathbf{0} \quad (3.16)$$

## 3.2 Linealización por Realimentación para Procesos SISO

Esta sección presenta la metodología para diseñar controladores estables basados en la linealización entrada-salida de procesos SISO no lineales, [7], [12], [13], [14], los cuales pueden ser descritos por:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u, \quad y = h(\mathbf{x}) \quad (3.16)$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} \quad \mathbf{g} = \begin{bmatrix} g_1 \\ \vdots \\ g_n \end{bmatrix}$$

donde  $\mathbf{x}$  es el vector de estado de orden  $n \times 1$ ,  $u$  es la señal de control o entrada,  $y$  es la salida del proceso,  $\mathbf{f}$  y  $\mathbf{g}$  son campos vectoriales suaves de orden  $n \times 1$  y  $h$  es una función suave que describe la salida.

### 3.2.1 Condiciones para Linealización

El proceso no lineal descrito en la ecuación (3.16) es linealizable sí y sólo si las siguientes condiciones son verdaderas para una región  $\Omega$ :

- 1) Los campos vectoriales  $\mathbf{g}, ad_{\mathbf{f}}\mathbf{g}, \dots, ad_{\mathbf{f}}^{m-1}\mathbf{g}$ , son linealmente independientes en  $\Omega$
- 2) El conjunto  $\mathbf{g}, ad_{\mathbf{f}}\mathbf{g}, \dots, ad_{\mathbf{f}}^{m-2}\mathbf{g}$  es involutivo en  $\Omega$

Para procesos lineales SISO:

$$\begin{aligned} \mathbf{f}(\mathbf{x}) &= \mathbf{A} \\ \mathbf{g}(\mathbf{x}) &= \mathbf{B} \end{aligned}$$

donde  $\mathbf{A}$  de orden  $n \times n$  y  $\mathbf{B}$  de orden  $n \times 1$  son las matrices de estado y control (o distribución) respectivamente. La primera condición arriba mencionada, puede ser interpretada como la condición de controlabilidad de un proceso lineal de la forma:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u, \quad y = C\mathbf{x} \quad (3.17)$$

El sistema (3.17) es controlable si su matriz de controlabilidad  $\mathcal{C}$  posee rango total; es decir:

$$rango \mathcal{C} = rango[\mathbf{B} \ \mathbf{A}\mathbf{B} \ \dots \ \mathbf{A}^{n-1}\mathbf{B}] = n \quad (3.18)$$

Por consiguiente la primera condición descrita arriba puede ser interpretada como la *Condición de Controlabilidad Generalizada* para procesos no lineales.

### 3.2.2 Grado Relativo del Proceso

Diferenciando la salida  $y$  de (3.16):

$$\dot{y} = \nabla h(\mathbf{x})(\mathbf{f} + \mathbf{g}u) = L_{\square}h(\mathbf{x}) + L_{\square}h(\mathbf{x})u$$

Si  $L_{\square}h(\mathbf{x}) \neq 0$  para algún  $\mathbf{x} = \mathbf{x}_0$ , entonces la siguiente transformación de entrada:

$$u = \frac{1}{L_{\square}h(\mathbf{x})}(-L_{\square}h(\mathbf{x}) + v)$$

genera una relación diferencial lineal entre la salida  $y$  y la nueva entrada  $v$  de la forma:

$$\dot{y} = v$$

Caso contrario si  $L_{\square}h(\mathbf{x}) = 0$  para todo  $\mathbf{x}$ , la diferenciación de  $\dot{y}$  resulta:

$$\ddot{y} = L_{\square}^2h(\mathbf{x}) + L_{\square}L_{\square}h(\mathbf{x})u$$

Si  $L_{\square}L_{\square}h(\mathbf{x}) = 0$  para todo  $\mathbf{x}$ , necesitamos diferenciar  $\ddot{y}$  y así sucesivamente  $y^3$ ,  $y^4$ , ...,  $y^i$ , como sigue:

$$y^i = \frac{d^i y}{dt^i} = L_{\square}^i h(\mathbf{x}) + L_{\square}L_{\square}^{i-1}h(\mathbf{x})u$$

hasta que para algún entero  $r = i$  y para algún  $\mathbf{x} = \mathbf{x}_0$ :

$$L_{\square}L_{\square}^{r-1}h(\mathbf{x}) \neq 0$$

En esta situación, sustituyendo la ley de control:

$$u = \frac{1}{L_{\mathbf{g}}L_{\mathbf{f}}^{r-1}h(\mathbf{x})}(-L_{\square}^r h(\mathbf{x}) + v)$$

en:

$$y^r = L_{\square}^r h(\mathbf{x}) + L_{\square}L_{\square}^{r-1}h(\mathbf{x})u$$

genera una relación diferencial lineal entre la salida  $y$  y la nueva entrada  $v$  de la forma:

$$y^r = \frac{d^r y}{dt^r} = v$$

Por lo tanto, un proceso **SISO** posee un grado relativo  $r$  en un punto  $x_0$  de una región  $\Omega$ , si para todo  $\mathbf{x} \in \Omega$ :

$$L_{\square} L_{\square}^i h(\mathbf{x}) = 0, \quad 0 \leq i < r - 1 \quad (3.19)$$

$$L_{\square} L_{\square}^{r-1} h(\mathbf{x}) \neq 0 \quad (3.20)$$

Observe que  $r$  es el número de diferenciaciones ejecutadas en la salida  $y$ . En general  $r \leq n$  (donde  $n$  es el orden del proceso). Si  $r = n$ , la linealización entrada-salida se denomina exacta.

### 3.2.3 Forma Normal SISO para Linealización Exacta

El proceso no lineal (3.12) puede ser transformado a una **forma normal** usando los resultados establecidos en la subsección 3.1.2. La linealización exacta de un proceso no lineal descrito por (3.12) ocurre cuando el grado relativo  $r$  es igual al orden  $n$  del proceso. Para linealización exacta, la transformación de estado:

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \Phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_n(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} h(\mathbf{x}) \\ L_{\square} h(\mathbf{x}) \\ \vdots \\ L_{\square}^n h(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \\ \vdots \\ y^n \end{bmatrix} \quad (3.21)$$

origina la siguiente ecuación de estado:

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \vdots \\ \dot{z}_{n-1} \\ \dot{z}_n \end{bmatrix} = \begin{bmatrix} z_2 \\ z_3 \\ \vdots \\ z_n \\ b(\mathbf{z}) + a(\mathbf{z})u \end{bmatrix} \quad y = z_1 \quad (3.22)$$

donde:

$$a(\mathbf{z}) = L_{\square} L_{\square}^{n-1} h(\Phi^{-1}(\mathbf{z}))$$

$$b(\mathbf{z}) = L_{\square}^n h(\Phi^{-1}(\mathbf{z})) \quad (3.23)$$

### 3.2.4 Ley de Control SISO para Linealización Exacta

Consideremos la siguiente ley de control linealizada:

$$u = \frac{1}{a(\mathbf{z})}[-b(\mathbf{z}) + v] \quad (3.24)$$

donde  $v$  es una nueva entrada a ser determinada,  $a(\mathbf{z})$  y  $b(\mathbf{z})$  son dadas por (3.23), y  $\mathbf{z}$  es el estado linealizado. Sustituyendo (3.24) en (3.22), llegamos a la representación canónica lineal invariante en el tiempo:

$$\dot{\mathbf{z}} = \mathbf{M}\mathbf{z} + \mathbf{N}v, \quad y = \mathbf{C}\mathbf{z} \quad (3.25)$$

donde:

$$\mathbf{M} = \begin{bmatrix} 0 & 1 & 0 & \cdot & \cdot & 0 \\ 0 & 0 & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & 1 \\ 0 & 0 & 0 & \cdot & \cdot & 0 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \end{bmatrix}$$

La función de transferencia del proceso linealizado es:

$$\frac{y(s)}{v(s)} = H(s) = \mathbf{C}[s\mathbf{I} - \mathbf{M}]^{-1}\mathbf{N} = \frac{1}{s^n} \quad (3.26)$$

donde  $s$  es el operador de Laplace. La entrada  $v$  puede ser diseñada empleando la técnica por ubicación de polos en el sistema lineal equivalente dado en (3.25). Luego, la entrada real  $u$  se calcula usando (3.24). Empleando (3.26), el sistema lineal equivalente se puede describir como:

$$v = \frac{d^n y}{dt^n} = y^{(n)} = \rho^{(n)} + K_1 \bar{y}^{(n-1)} + \dots + K_{n-1} \dot{\bar{y}} + K_n \bar{y} \quad (3.27)$$

donde  $\rho(t)$  es la trayectoria deseada e  $\bar{y}(t) = \rho(t) - y(t)$  es la señal de error de seguimiento. La ecuación característica del sistema lineal equivalente es hallada de:

$$\bar{y}^{(n)} + K_1 \bar{y}^{(n-1)} + \dots + K_{n-1} \dot{\bar{y}} + K_n \bar{y} = 0 \quad (3.28)$$

Para un comportamiento asintóticamente estable, con  $\bar{y} = (\rho - y) \rightarrow 0$  para  $t \rightarrow \infty$ , ninguno de las raíces de (3.28) debe ser localizado en el lado derecho del plano  $s$ , incluyendo el eje imaginario. Los parámetros del controlador  $K_i$ ,

$i = 1, \dots, n$  son seleccionados para encontrar tales requerimientos. La entrada  $v$  toma la forma (tener presente que  $\rho^1 \equiv d\rho/dt$  y  $\rho^0 \equiv \rho$ ):

$$v_i = \rho_i^{(n)} + \sum_{j=1}^n K_j \rho^{(n-j)} - \mathbf{Kz} \quad (3.29)$$

donde:

$$\mathbf{K} = \begin{bmatrix} K_1 & \dots & K_n \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} z_n \\ z_{n-1} \\ \vdots \\ z_1 \end{bmatrix} = \begin{bmatrix} y^{(n-1)} \\ y^{(n-2)} \\ \vdots \\ y \end{bmatrix}$$

### 3.3 Linealización por Realimentación para Procesos MIMO

#### 3.3.1 Modelando Procesos Cuadrados No Lineales

Un proceso no lineal **MIMO** [13], es llamado cuadrado cuando el número de entradas del proceso es igual al número de salidas del proceso. El proceso cuadrado a ser considerado se describe como:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} = \mathbf{f}(\mathbf{x}) + \sum_{j=1}^m \mathbf{g}_j(\mathbf{x})u_j(\mathbf{x}) \quad \mathbf{y} = \mathbf{h}(\mathbf{x}) \quad (3.30)$$

donde:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} h_1(\mathbf{x}) \\ \vdots \\ h_m(\mathbf{x}) \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \mathbf{g}_1(\mathbf{x}) & \dots & \mathbf{g}_m(\mathbf{x}) \end{bmatrix}$$

donde  $\mathbf{x}$  es el vector de estado de orden  $n \times 1$ ,  $\mathbf{u}$  es el vector de control de entrada (de componentes  $u_i$ ) de orden  $m \times 1$ ,  $\mathbf{y}$  es la salida del vector (de componentes  $y_i$ ) de orden  $m \times 1$ ,  $\mathbf{f}$  es un campo vectorial suave de orden  $n \times 1$ ,  $\mathbf{h}$  es un campo vectorial suave de orden  $m \times 1$ , y  $\mathbf{G}$  es una matriz  $n \times m$  cuyas columnas son los campos vectoriales  $\mathbf{g}_j$  de orden  $n \times 1$ . Los componentes de  $\mathbf{g}$  y  $\mathbf{h}$  son funciones suaves.

### 3.3.2 Grado Relativo Total

El proceso cuadrado descrito por (3.30) posee un *grado relativo total*  $r = r_1 + \dots + r_m$ , [12], [13], [14] sí:

- 1) Para todo  $\mathbf{x}$  alrededor de  $\mathbf{x}_o$ :

$$L_{\square_j} L_{\square}^k h_i(\mathbf{x}) = 0 \quad j, i = 1, \dots, m \quad k < r_i - 1 \quad (3.32)$$

- 2) La siguiente matriz  $m \times m$  es no singular en  $\mathbf{x} = \mathbf{x}_o$ :

$$A(\mathbf{x}) = \begin{bmatrix} L_{\square_1} L_{\square}^{r_1-1} h_1(\mathbf{x}) & \dots & L_{\square_m} L_{\square}^{r_1-1} h_1(\mathbf{x}) \\ L_{\square_1} L_{\square}^{r_2-1} h_2(\mathbf{x}) & \dots & L_{\square_m} L_{\square}^{r_2-1} h_2(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ L_{\square_1} L_{\square}^{r_m-1} h_m(\mathbf{x}) & \dots & L_{\square_m} L_{\square}^{r_m-1} h_m(\mathbf{x}) \end{bmatrix} \quad (3.31)$$

Observando las filas de  $A$ , podemos decir que cada entero  $r_i$  está relacionado con la  $i$ -ésima salida del proceso. También notamos que  $r_i$  es el mínimo número de diferenciaciones ejecutadas en la salida  $y_i$ , requeridas para que las componentes de  $u$  no desaparezcan. La no singularidad de  $A(\mathbf{x})$  en  $\mathbf{x} = \mathbf{x}_o$  es la versión multivariable de la condición impuesta por (3.20).

### 3.3.3 Forma Normal para Linealización Exacta MIMO

Si el grado relativo total de un proceso cuadrado descrito por (3.30) es  $r$ , luego:

$$r = r_1 + r_2 + \dots + r_m \leq n$$

La linealización exacta de un proceso no lineal **MIMO** descrito por (3.30), ocurre cuando el grado relativo total :  $r = r_1 + \dots + r_m$ , es igual al orden  $n$  del proceso; es decir, la dimensión  $n$  del vector de estado. Para linealización exacta, se tiene la siguiente transformación de estado  $\mathbf{z}$ :

$$z = \Phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \vdots \\ \phi_{r_1}(\mathbf{x}) \\ \psi_1(\mathbf{x}) \\ \vdots \\ \psi_{r_2}(\mathbf{x}) \\ \xi_1(\mathbf{x}) \\ \vdots \\ \xi_{r_m}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} z_1 \\ \vdots \\ z_{r_1} \\ z_{r_1+1} \\ \vdots \\ z_{r_1+r_2} \\ z_{r_1+r_2+\dots+1} \\ \vdots \\ z_{r_1+r_2+\dots+r_m} \end{bmatrix} \quad (3.32)$$

donde:

$$\begin{aligned} \phi_1(\mathbf{x}) &= h_1(\mathbf{x}) = y_1(\mathbf{x}); \quad \phi_2(\mathbf{x}) = L_f h_1(\mathbf{x}); \quad \dots; \quad \phi_{r_1}(\mathbf{x}) = L_f^{r_1-1} h_1(\mathbf{x}) \\ \psi_1(\mathbf{x}) &= h_2(\mathbf{x}) = y_2(\mathbf{x}); \quad \psi_2(\mathbf{x}) = L_f h_2(\mathbf{x}); \quad \dots; \quad \psi_{r_2}(\mathbf{x}) = L_f^{r_2-1} h_2(\mathbf{x}) \\ &\vdots \end{aligned}$$

$$\xi_1(\mathbf{x}) = h_m(\mathbf{x}) = y_m(\mathbf{x}); \quad \xi_2(\mathbf{x}) = L_f h_m(\mathbf{x}); \quad \dots; \quad \xi_{r_m}(\mathbf{x}) = L_f^{r_m-1} h_m(\mathbf{x})$$

da lugar a las siguientes ecuaciones de estado:

$$\dot{y}_1 = \dot{\phi}_1(\mathbf{x}) = \phi_2(\mathbf{x})$$

$\vdots$

$$y_1^{(r_1-1)} = \dot{\phi}_{r_1-1}(\mathbf{x}) = \phi_{r_1}(\mathbf{x})$$

$$y_1^{(r_1)} = \dot{\phi}_{r_1}(\mathbf{x}) = L_f^{r_1} h_1(\Phi^{-1}(\mathbf{z})) + \sum_{j=1}^m L_{g_j} L_f^{r_1-1} h_1(\Phi^{-1}(\mathbf{z})) u_j$$

$$\dot{y}_2 = \dot{\psi}_1(\mathbf{x}) = \psi_2(\mathbf{x})$$

$\vdots$

$$y_2^{(r_2-1)} = \dot{\psi}_{r_2-1}(\mathbf{x}) = \psi_{r_2}(\mathbf{x})$$

$$y_2^{(r_2)} = \dot{\psi}_{r_2}(\mathbf{x}) = L_f^{r_2} h_2(\Phi^{-1}(\mathbf{z})) + \sum_{j=1}^m L_{g_j} L_f^{r_2-1} h_2(\Phi^{-1}(\mathbf{z})) u_j$$

$$\dot{y}_m = \dot{\xi}_1(\mathbf{x}) = \xi_2(\mathbf{x}) \quad (3.33)$$



$$y_m^{(r_m-1)} = \dot{\xi}_{r_m-1}(\mathbf{x}) = \xi_{r_m}(\mathbf{x})$$

$$y_m^{(r_m)} = \dot{\xi}_{r_m}(\mathbf{x}) = L_f^{r_m} h_m(\Phi^{-1}(\mathbf{z})) + \sum_{j=1}^m L_{g_j} L_f^{r_m-1} h_m(\Phi^{-1}(\mathbf{z})) u_j \quad (3.33)$$

### 3.3.4 Ley de Control de Desacoplamiento MIMO

Es importante señalar que el grado relativo  $r_i$ ,  $i = 1, \dots, m$ , es el entero más pequeño requerido para que por lo menos una de las entradas aparezca en (ver(3.33)):

$$y_i^{(r_i)} = L_f^{r_i} h_i(\Phi^{-1}(\mathbf{z})) + \sum_{j=1}^m L_{g_j} L_f^{r_i-1} h_i(\Phi^{-1}(\mathbf{z})) u_j$$

Consideremos ahora la siguiente *ley de control desacoplada*:

$$u = \mathbf{A}^{-1}(\mathbf{z})[-\mathbf{B}(\mathbf{z}) + \mathbf{v}] \quad (3.34)$$

donde  $\mathbf{v} = [v_1 \dots v_m]^T$ , es una nueva entrada a ser determinada, y  $\mathbf{A}(\mathbf{z})$ , (ver (3.31)), y  $\mathbf{B}(\mathbf{z})$  son matrices con elementos:

$$b_i = L_f^{r_i} h_i(\Phi^{-1}(\mathbf{z})), \quad i = 1, \dots, m$$

$$a_{ij} = L_{g_j} L_f^{r_j-1} h_i(\Phi^{-1}(\mathbf{z})), \quad i, j = 1, \dots, m$$

respectivamente. Sustituyendo los elementos de  $u_i$ ,  $i = 1, \dots, m$  (ecuación (3.34)) en (3.33), se obtiene la representación canónica lineal invariante en el tiempo del proceso **MIMO**; es decir, la ecuación (3.33) con:  $y_i^{(r_i)} = v_i$ .

Por otro lado, de la relación  $\mathbf{y}(s) = \mathbf{H}(s)\mathbf{v}(s)$ , se puede demostrar que la transferencia  $\mathbf{H}(s)$ , del proceso linealizado **MIMO** [13]:

$$\mathbf{H}(s) = \begin{bmatrix} 1/s^{r_1} & 0 & \dots & 0 \\ 0 & 1/s^{r_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/s^{r_m} \end{bmatrix} \quad (3.35)$$

lo cual significa que  $y_i^{(r_i)} = d^{r_i} y_i / dt^{r_i} = v_i$ ,  $i = 1, \dots, m$  donde cada entrada  $v_i$  es diseñada por ubicación de polos aplicado al correspondiente subsistema lineal equivalente. Tal subsistema puede ser seleccionada como:

$$v_i = \frac{d^{r_i} y_i}{dt^{r_i}} = y_i^{(r_i)} = \rho_i^{(n)} + K_{i,1} \bar{y}_i^{(r_i-1)} + \dots + K_{i,r_i-1} \dot{\bar{y}}_i + K_{i,r_i} \bar{y}_i, \quad i = 1, \dots, m \quad (3.36)$$

donde  $\rho_i(t)$  es la  $i$ -ésima trayectoria deseada e  $\bar{y}_i(t) = \rho_i(t) - y_i(t)$  es la  $i$ -ésima señal de error de seguimiento. La  $i$ -ésima ecuación característica del subsistema es:

$$\bar{y}_i^{(r_i)} + K_{i,1}\bar{y}_i^{(r_i-1)} + \dots + K_{i,r_i-1}\dot{\bar{y}}_i + K_{i,r_i}\bar{y}_i = 0, \quad i = 1, \dots, m \quad (3.37)$$

Para comportamiento asintóticamente estable, es decir con  $\bar{y}_i \rightarrow 0$  o  $y_i \rightarrow \rho_i$  para  $t \rightarrow \infty$ , ninguno de las raíces de (3.37) debe ser localizado en el lado derecho del plano  $s$ , incluyendo el eje imaginario. Los parámetros del controlador  $K_{i,j}$ ;  $i = 1, \dots, m$ ;  $j = 1, \dots, r_i$ , son seleccionados para encontrar tales requerimientos, y la entrada  $v_i$  tomará la forma:

$$v_i = \rho_i^{(r_i)} + \sum_{j=1}^{r_i} K_{i,j}\rho_i^{(r_i-j)} - \mathbf{K}_i \mathbf{z}_i, \quad i = 1, \dots, m \quad (3.38)$$

donde:

$$\mathbf{K}_i = \begin{bmatrix} K_{i,1} & \dots & K_{i,r_i} \end{bmatrix} \quad \mathbf{z}_i = \begin{bmatrix} y_i^{(r_i-1)} \\ \vdots \\ y_i \end{bmatrix} \quad i = 1, \dots, m \quad (3.39)$$

Figura 3.2. Esquemas de lazo cerrado para el control de seguimiento.

## Capítulo 4

# Implementación en Tiempo Real

La implementación del sistema SRM de 2DOF se muestra en la figura 4.1, y comprende la implementación del hardware y la implementación del software.

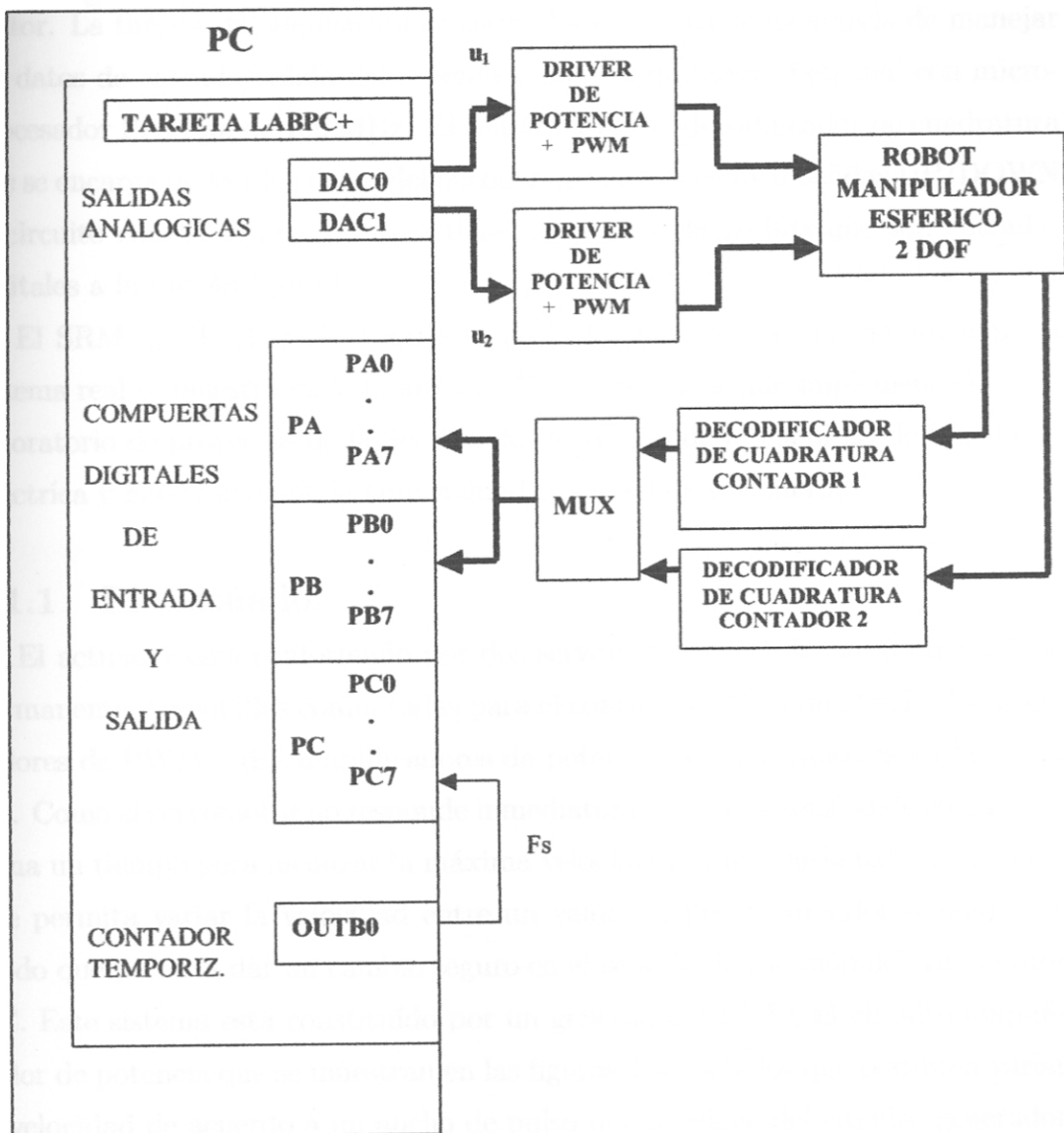


Figura 4.1: Esquema de Implementación del sistema.

## 4.1 Implementación del Hardware

La implementación de los componentes físicos usados en el sistema de control son: dos servomotores DC (modelos iguales) con encoder óptico incremental de 1850 cuentas por vuelta con salida a dos canales en cuadratura. Los actuadores están compuesto por tarjetas similares, como son el generador de PWM que modula las señales de control para que luego sean desfasadas en  $90^\circ$  y el amplificador de potencia que es el encargado de amplificar las señales que provienen del generador de PWM y cuya salida genera el control de inversión del giro del servomotor. La tarjeta de adquisición de datos Lab-PC+ es la encargada de manejar los datos de entrada/salida del sistema y una computadora personal con microprocesador Pentium de 100 MHz. El sensado tiene el decodificador de cuadratura que se encarga de leer los datos del encoder óptico para emitir salidas UP/DOWN al circuito contador, que a su vez tiene una salida de 16 bits que son entradas digitales a la tarjeta Lab-PC+.

El SRM de 2DOF real se muestra en la figura 4.2. La implementación del sistema real se muestra en la figura 4.3. Este sistema ha sido implementado en el laboratorio de proyectos de la Sección de Postgrado de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Ingeniería.

### 4.1.1 El Actuador

El actuador está conformado por dos servomotores DC de campo magnético permanente y escobillas conmutadas para el control del SMR de 2DOF, dos generadores de PWM y dos amplificadores de potencia como se muestra en la figura 4.1. Como el servomotor no responde inmediatamente a una tensión de entrada, le toma un tiempo para alcanzar la máxima velocidad, es necesario usar un sistema que permita variar la velocidad entre un valor mínimo y un valor máximo, de modo que permita dar un cambio seguro en el sentido de rotación del servomotor DC. Este sistema está constituido por un generador PWM y el circuito amplificador de potencia que se muestran en las figuras 4.4 y 4.5, los que permiten variar la velocidad de acuerdo a un ancho de pulso que proviene del circuito generador de PWM. El LM3524 genera una modulación por ancho de pulso a una frecuencia de 15.4KHz (puede variar), esta señal es amplificada por un amplificador de potencia que alimenta al servomotor, al que entrega una tensión de alimentación

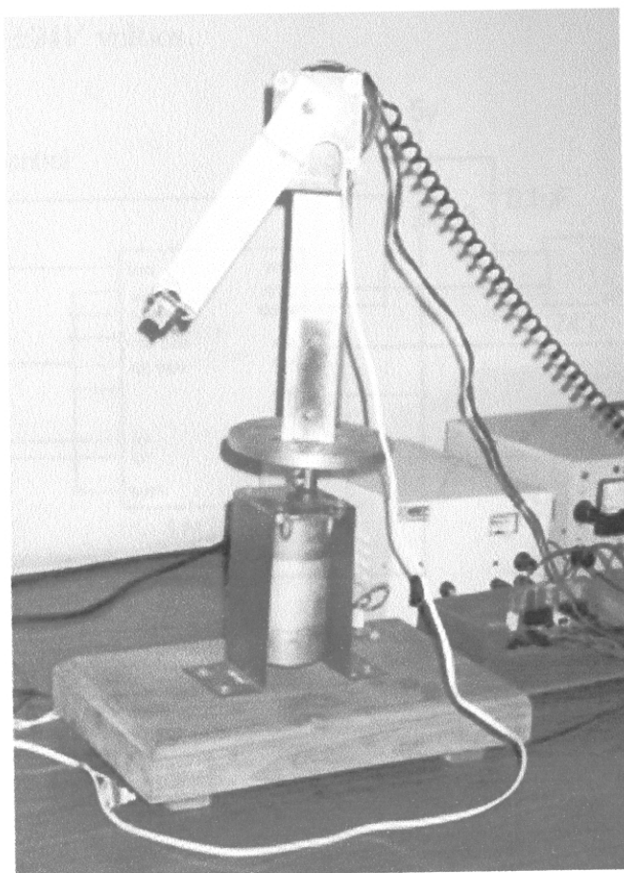


Figura 4.2: Planta SRM de 2DOF real

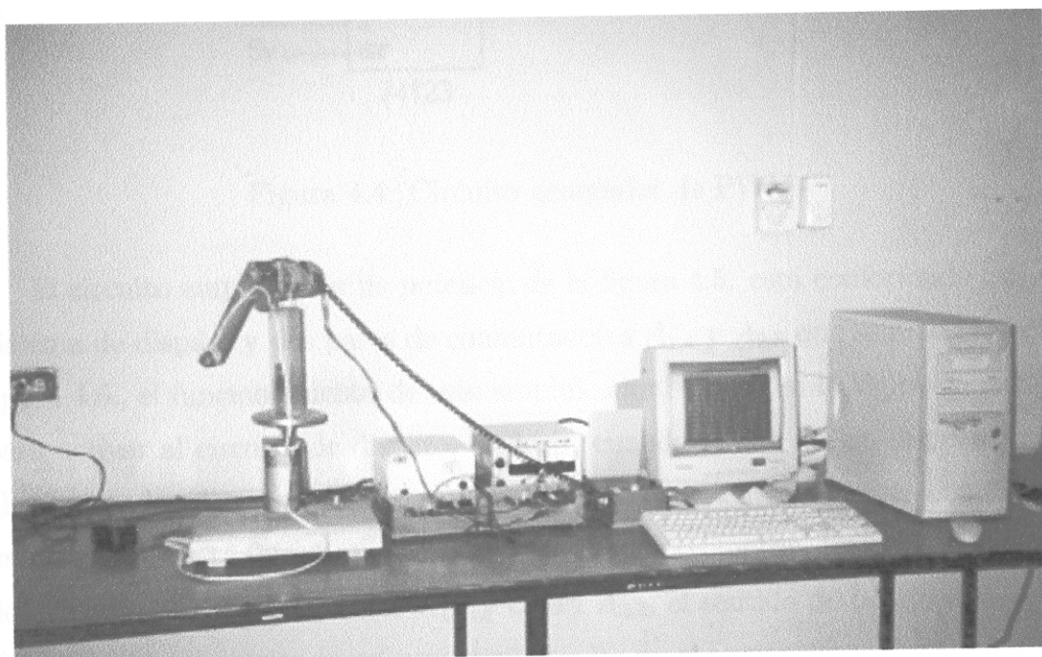


Figura 4.3: Implementación del sistema real

al servomotor de  $\pm 24V$  voltios.

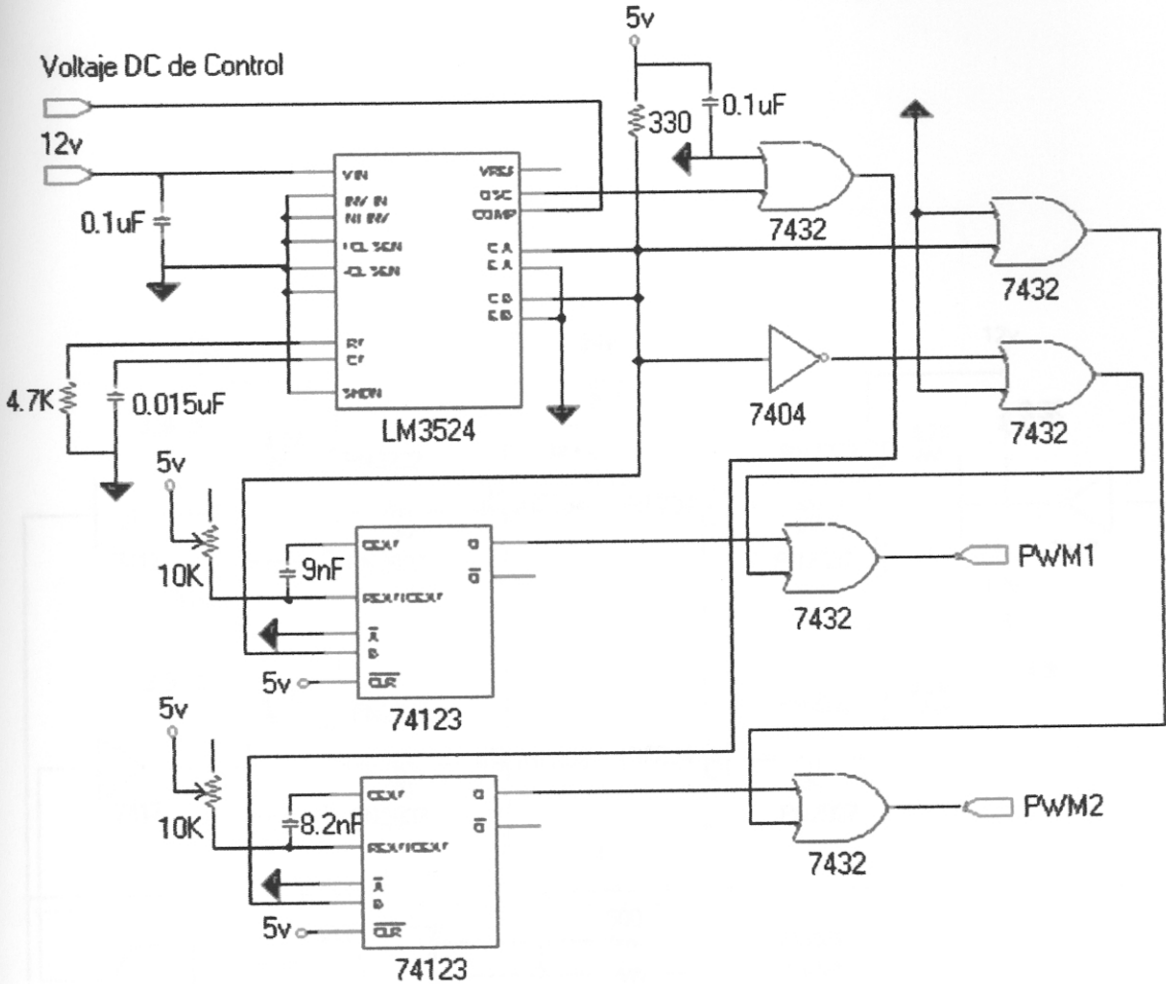


Figura 4.4: Circuito generador de PWM.

El circuito amplificador de potencia de la figura 4.5. está conformado por un sistema de disparo y dos pares de conmutadores  $A_{1,4}$  y  $A_{2,3}$  que se muestra en la figura 4.6., el funcionamiento de este amplificador se basa en la lógica de control para activar al circuito de disparo, es decir, cuando el sistema de disparo cierra al conmutador  $A_{1,4}$  y se abre  $A_{2,3}$ , el sentido de la corriente es la línea con puntos induciendo de esta forma una tensión  $+V_{cc}$  en el servomotor. Luego si el sistema de disparo abre al conmutador  $A_{1,4}$  y cierra  $A_{2,3}$ , el sentido de la corriente es la línea continua, induciendo así una tensión  $-V_{cc}$  en el servomotor. Por lo tanto el servomotor ve en sus bornes una onda de voltaje cuadrada, variando entre  $\pm V_{cc}$  y la corriente que puede absorber dependerá de los conmutadores.

En resumen, de lo anterior, podemos decir que el objetivo del sistema de

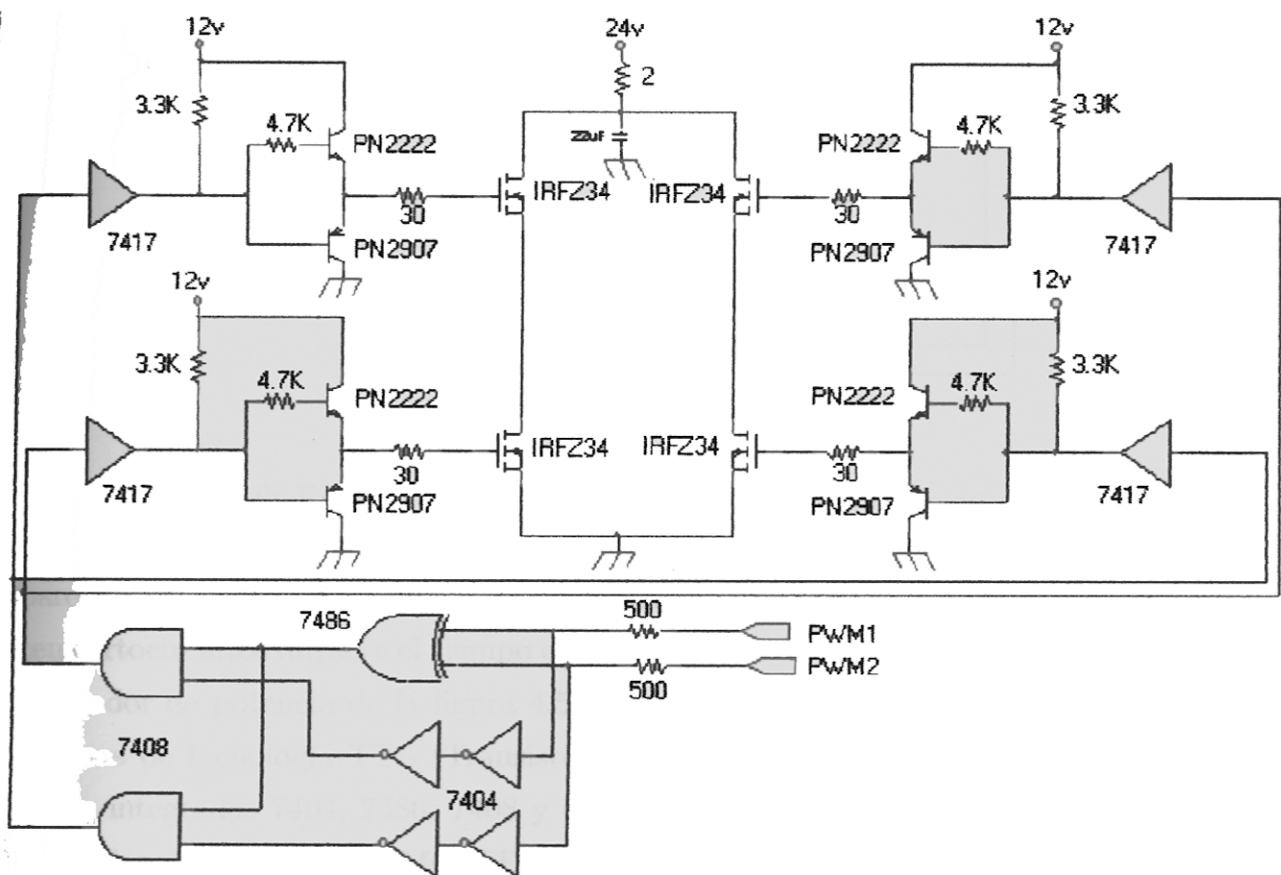


Figura 4.5: Circuito amplificador de potencia de PWM.

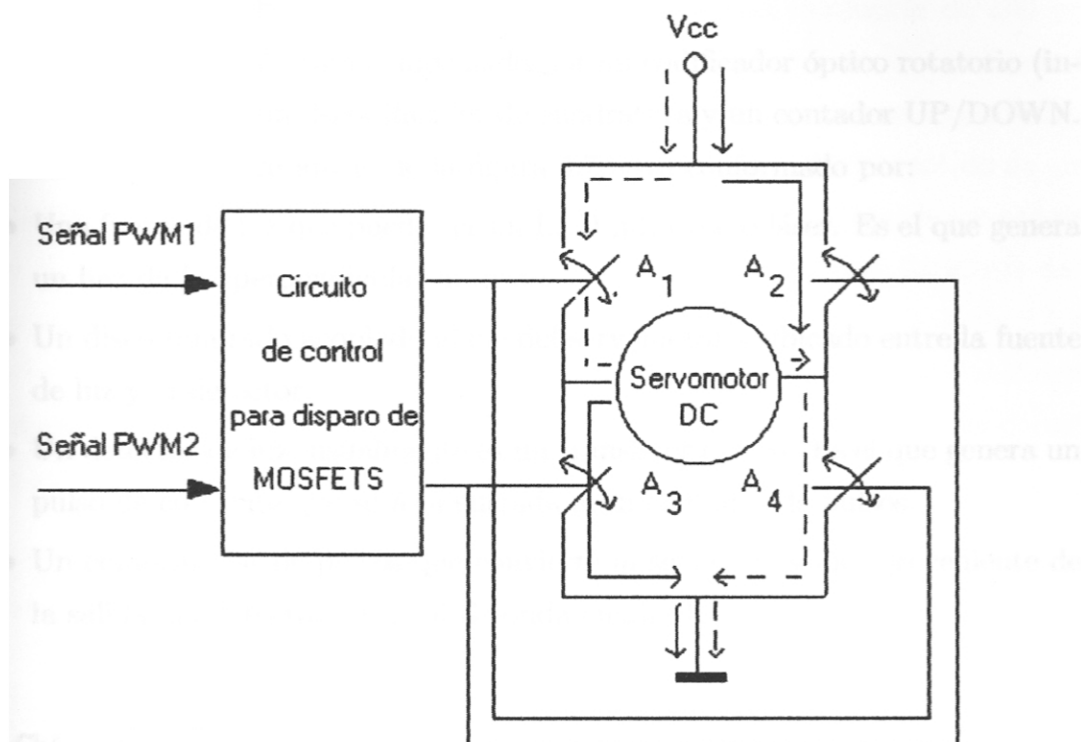


Figura 4.6: Esquema del circuito de disparo y conmutación.

disparo es realizar la conmutación de cada par  $A_{1,4}$  y  $A_{2,3}$ , de tal forma que se eviten cortocircuitos durante el tiempo de conmutación de  $A_{1,4}$  y  $A_{2,3}$ . El circuito amplificador de potencia de la figura 4.5, tiene un circuito basado en compuertas lógicas de tecnología TTL (Transistor Transistor Logic) que conforman los circuitos integrados 7404, 7486, 7408 y dos resistencias de  $500\Omega$ . Este circuito gobierna los disparos de los MOSFET's de manera que las señales PWM que ingresan al circuito amplificador de potencia que tengan igual amplitud estarán siempre en nivel bajo (OFF), como se muestra en la tabla 4.1.

Tabla 4.1: Conmutación de los MOSFET's.

$PWM_1$	$PWM_2$	$A_1$	$A_2$	$A_3$	$A_4$
0	0	OFF	OFF	OFF	OFF
0	1	ON	OFF	OFF	ON
1	0	OFF	ON	ON	OFF
1	1	OFF	OFF	OFF	OFF



### 4.1.2 El Sensor

El sensor de posición está conformado por un codificador óptico rotatorio (interno al servomotor), un decodificador de cuadratura y un contador UP/DOWN. El codificador óptico rotatorio de la figura 4.7, está conformado por:

- Una fuente de luz que puede ser un LED infrarrojo o láser. Es el que genera un haz de luz perpendicular al disco.
- Un disco ranurado acoplado al eje del servomotor y ubicado entre la fuente de luz y el detector.
- Un detector de luz, usualmente es un transistor óptico. Es el que genera un pulso de corriente que será la entrada a un contador de pulsos.
- Un conformador de pulsos que convierte la señal sinusoidal proveniente de la salida del detector en señal de onda cuadrada.

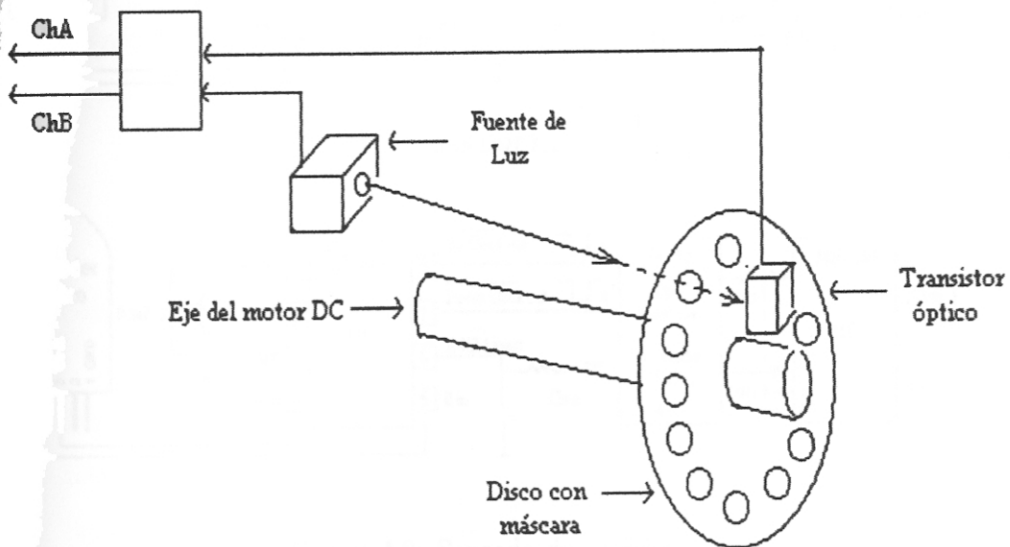


Figura 4.7: Estructura del codificador óptico rotatorio.

El codificador usado en esta implementación, incluye dos pistas de marcas o ranuras en cuadratura (canales A y B), cada una asociada a una fuente de luz y un fotodetector independientes. Los canales A y B producen dos ondas cuadradas desfasadas en  $\pm 90^\circ$  respecto al primero, de acuerdo al sentido de giro del disco. Este desfase se utiliza para determinar la dirección del movimiento. Por ejemplo, si el canal B adelanta al canal A, el eje está girando en el sentido

antihorario y viceversa. El codificador usado tiene una resolución de 100 ppr (pulsos por revolución). La figura 4.8, muestra el sensor de posición conformado por el decodificador de cuadratura LS7083, que activará la cuenta ascendente o descendente del contador UP/DOWN de 16 bits, dependiendo del sentido de giro del servomotor, captado por el desfase entre los trenes de pulsos de los canales A y B. El contador UP/DOWN lo conforman cuatro contadores 74HC193 de 4 bits conectados en cascada; dos de ellos conforman el bus de datos A de 8 bits que se dirige hacia la tarjeta Lab-PC+; mientras que los dos contadores restantes conforman el bus B, de modo tal que para cada medición de posición tendremos una salida de 16 bits que se almacena en un registro.

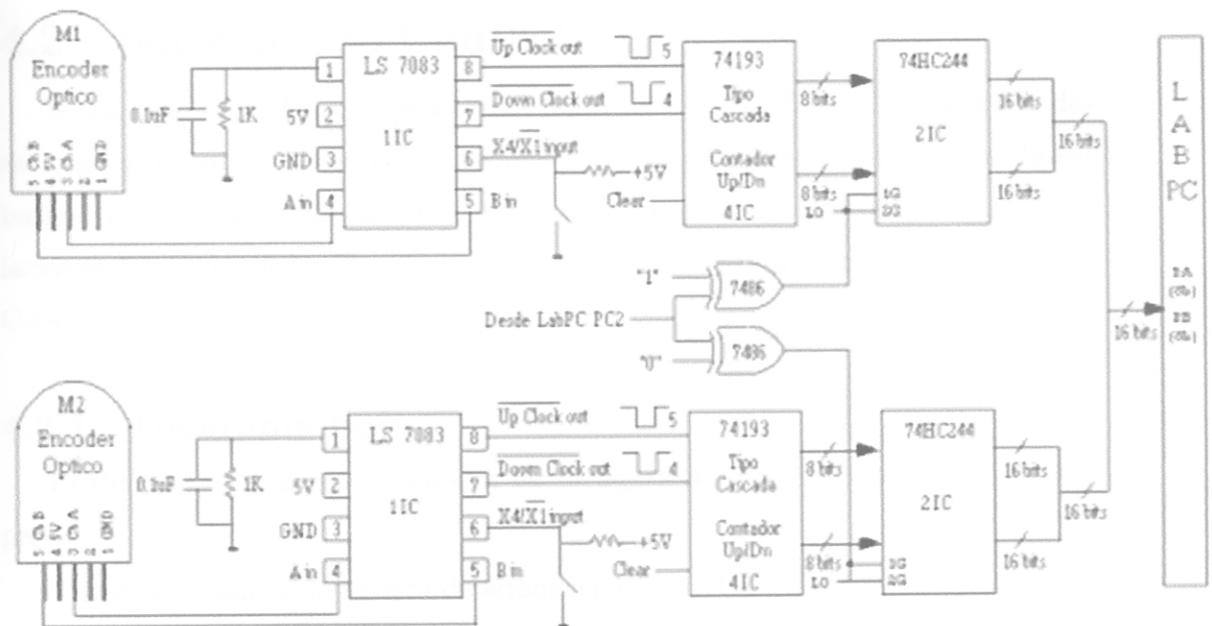


Figura 4.8: Sensado de posición

### 4.1.3 La Interfaz Lab-PC+

La tarjeta de adquisición de datos Lab-PC+ que se muestra en la figura 4.1, es la interfaz empleada para el envío de la señal de control y para la adquisición de datos con el amplificador y los sensores se realizan en la forma siguiente:

1. Dos salidas análogas (DAC0 y DAC1), estas señales de control envían por los terminales del LM3524 (generador PWM) con una resolución de 12 bits para generar la señal analógica.

2. Dos puertos digitales de entrada (PA y PB) de 8 bits cada uno. Estos puertos reciben los datos provenientes del contador UP/DOWN.
3. El terminal OUTB0 que se utiliza como salida de una señal de reloj para el funcionamiento del sistema de control.
4. El terminal PC0 (bit 0 del puerto C) que se utiliza como entrada de la señal de reloj proveniente del OUTB0.

La señal de reloj OUTB0 se genera a partir de una base de tiempos de 2 MHz que activa la salida del contador/temporizador 8253. Los puertos de entrada/salida proviene del PPI (Interface Paralela Programable) 8255.

## 4.2 Software de Control

Se ha usado una PC con procesador Pentium de 100MHz. El software se desarrollo en lenguaje C++, el cual consiste en un programa de control no lineal basado en el método de linealización por realimentación de estado. La simulaciones han sido realizadas en MATLAB y luego se han traducido al lenguaje C++.

### 4.2.1 Diagrama flujo

El diagrama de flujo se muestra en la figura 4.9. y describe los siguientes pasos:

1. Inicialización de la tarjeta de adquisición de datos Lab-PC+ y de las variables utilizadas en el programa.
2. Se detecta el flanco de subida del reloj (clock) y se lee el primer dato de la posición inicial, inicializa el algoritmo.
3. Se comprueba que la variable  $k$  sea menor que el número de muestras  $N_m$ . La variable  $N_m$  junto con la frecuencia de muestreo determinan el tiempo de duración del experimento.
4. Se detecta un flanco de subida del reloj, se inicia un período para la adquisición de datos y control. Se lee la posición  $y$  y se aplica restricciones, luego se calcula la velocidad.
5. Se calcula la señal de control  $u$ , se compensa las tensiones de las fricciones estáticas y de Coulomb  $v_p$ , se aplican restricciones en la señal de control,

luego se envía la señal de control  $u = u + v_p + offset$  por el DAC0 y DAC1 a los generadores de PWM.

6. Se actualizan las variables y se repite el lazo de control hasta que  $k$  alcance a  $N_m$ .

#### 4.2.2 Programa Principal XX.CPP

El programa XX.CPP realiza las siguientes declaraciones y funciones:

1. Declaración de los archivos de cabecera : librería de funciones, etc. Así mismo se declara el archivo "func\_pen.CPP", en el cual se emplea la librería de funciones del NIDAQ (National Instruments Data Acquisition) para configurar el hardware empleado, utilizando las siguientes funciones: ConfigurarHardware, EnviarVoltaje1, EnviarVoltaje2, LeerPosición, NivelClock, ResetContadores. De la misma manera se declara el archivo "func\_pen.H", el cual nos permite facilitar el manejo de la tarjeta de adquisición de datos Lab-PC+, utilizando para ello las mismas funciones anteriores.
2. Declaración de las constantes globales.
3. Declaración de los parámetros de la planta.
4. Declaración de las funciones propias del programa:  
*Mide\_Salidas*, *Aplica\_Control1* y *Aplica\_Control2*.
6. Declaración de las variables globales.
7. Definición de la función principal main.

#### 4.2.3 Funciones Propias del Programa XX.CPP

Las funciones propias del sistema de control son:

1. *Mide\_Salidas*: Especifica las señales de referencia consideradas. Luego mide la posición actual del servomotor DC1 (base) y la posición actual del servomotor DC2 (brazo). Posteriormente calcula las velocidades en función de las posiciones de cada servomotor.
2. *Aplica\_Control1* y *Aplica\_Control2*: Compensa las fricciones estática y de Coulomb para cada servomotor ( $v_{p1}$  y  $v_{p2}$ ) y aplica las restricciones a las señales de entrada  $u_1$  y  $u_2$ . Luego envía los voltajes por medio de las funciones:  $EnviarVoltaje1(u_1 + v_{p1} + offset1)$ , y  $EnviarVoltaje2(u_2 + v_{p2} + offset2)$ .

3. Crea archivo de datos XX.DAT, para hacer tablas y gráficos.

#### 4.2.4 La Función Principal Main

En la función principal tenemos:

1. Declaración de las variables locales.
2. Asegurar la memoria que nos permita almacenar los datos para las variables a usarse para:  
la primera entrada:  $p_1$  (referencia),  $y_1$  (posición),  $w_1$  (velocidad angular),  $u_1$  (señal de control), y para  
la segunda entrada:  $p_2$  (referencia),  $y_2$  (posición),  $w_2$  (velocidad angular),  $u_2$  (señal de control).
3. Ejecutar el lazo de control del programa desde  $k = 0$  hasta  $k = N_m$  ( $N_m$ : máximo número de muestras). Así mismo se define la función *Mide\_Salidas* en cada flanco de subida del reloj.
4. Luego se genera las señales de control  $u_1$  y  $u_2$  y se envía a las salidas del DAC0 y DAC1 las tensiones ordenadas por la función *Aplica\_Control1* y *Aplica\_Control2*.
5. Posteriormente se incrementan  $k$  y  $T$  y se visualizan en la pantalla los valores instantáneos de  $t$ ,  $p_1$ ,  $y_1$ ,  $w_1$ ,  $u_1$ ,  $p_2$ ,  $y_2$ ,  $w_2$ ,  $u_2$ , y se actualiza el pulso del reloj siempre que  $k \leq N_m$ .
6. Al finalizar el tiempo de ejecución, enviar los voltajes de offset1 y offset2 necesarios para que los servomotores se encuentren en posición de reposo.



Figura 4.9 Diagrama de flujo del algoritmo de control.

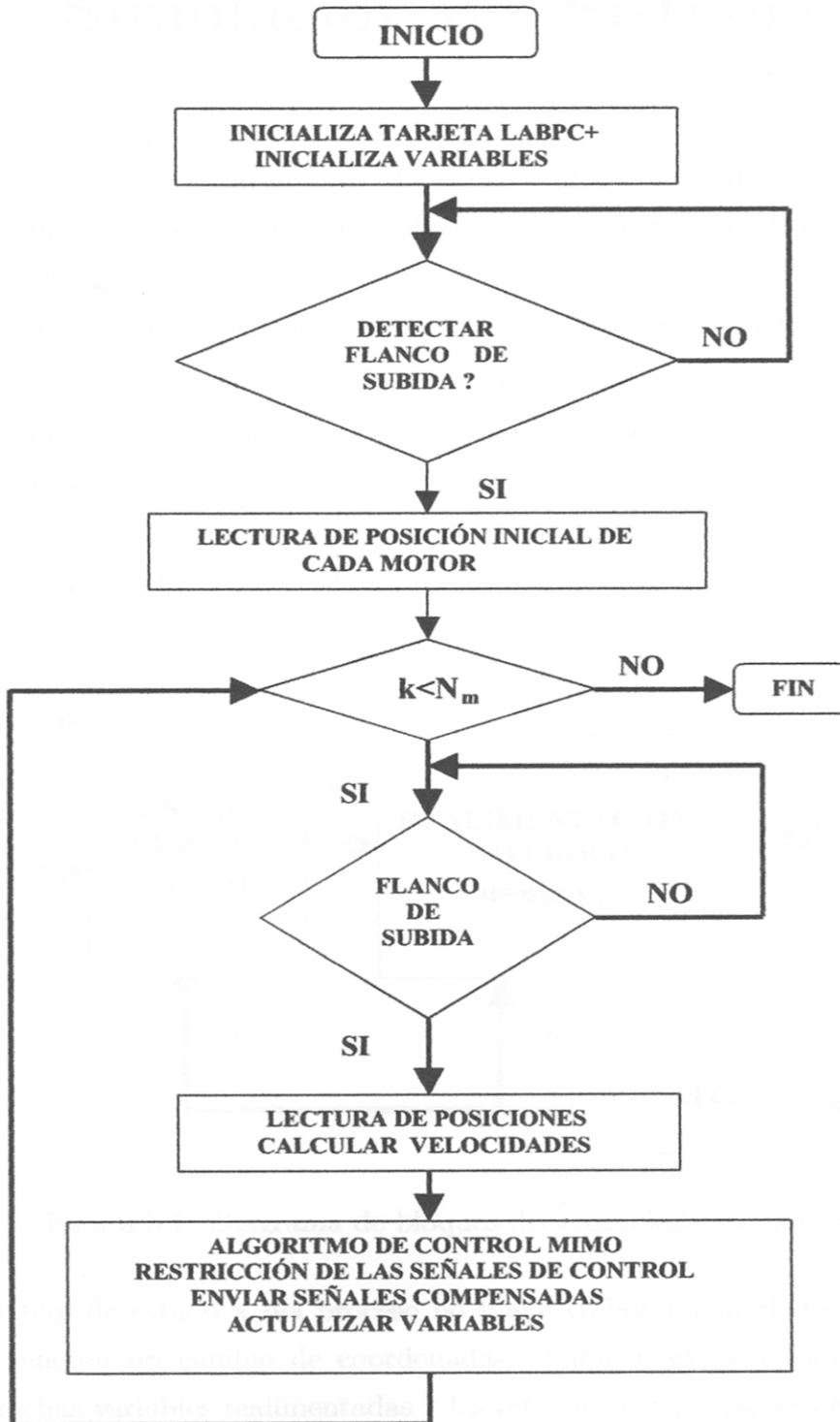


Figura 4.9: Diagrama de flujo del algoritmo de control.

## Capítulo 5

# Simulación del Sistema

En esta sección se muestran las simulaciones del sistema de control no lineal multivariable de dos entradas y dos salidas, por realimentación de estado para el SRM de 2DOF.

Para obtener las simulaciones respectivas se ha tenido en cuenta:

1. El modelo en el espacio-estado, visto en la subsección 2.4.5.
2. El diseño del controlador no lineal multivariable basado en el método de linealización exacta por realimentación de estado, visto en la subsección 3.3.4.
3. El diagrama de bloques del controlador no lineal ilustrado en la figura 5.1.

### Referencias

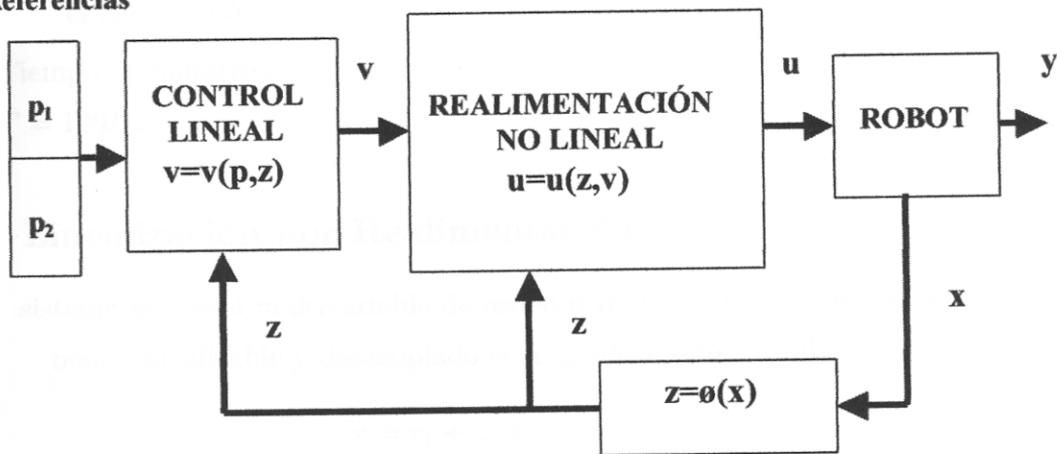


Figura 5.1: Diagrama de bloques del controlador no lineal

Las variables de estado  $x$  del proceso no lineal (robot manipulador esférico de 2DOF) generan un cambio de coordenadas en una nueva ecuación de estado  $z$ . Con dichas variables realimentadas y las referencias  $p_1$  y  $p_2$ , se determina las nuevas señales de control lineal  $v$ , mediante la técnica por ubicación de polos. Así mismo con estas señales de control  $v$  y con las variables de estado realimentadas

$z$ , se generan las señales de control real  $u$  que intervienen en el proceso no lineal. Basado en los conceptos anteriores se han realizado las simulaciones utilizando el programa MATLAB [10], [11]. El desarrollo de las simulaciones se ha dividido para control de trayectoria constante y para control de trayectoria senoidal.

## 5.1 Caso 1: Control de Trayectoria Constante

Para esta simulación se ha asumido una trayectoria constante con los siguientes datos:

- Condiciones iniciales:
  - $x_1 = 0$ , (base)
  - $x_2 = 0$ , (brazo)
- Trayectorias deseadas:
  - $p_1 = \pi/2$ , posición angular de la base
  - $p_2 = 3\pi/2$ , posición angular del brazo
- Polos usados:
  - para la base: -8 y -8
  - para el brazo: -7.5 y -8
- Tiempo de muestreo:
  - $T = 100Hz$

### 5.1.1 Linealización por Realimentación

Un sistema no lineal multivariable de orden  $n$  de  $m$  entradas y  $m$  salidas es completamente linealizabile y desacoplado si su grado relativo total es:

$$r = r_1 + \dots + r_m$$

Si el grado relativo total  $r$ , es igual al número de orden del sistema o sea " $r = n$ ", entonces se aplica el método de linealización exacta. Aplicando los conceptos de geometría diferencial [7], [13], [14], [15], determinamos el grado relativo total.

### 5.1.2 Determinación del Grado Relativo Total: $r$

Aplicando los conceptos de la subsecciones 3.2.2 y 3.3.2, se obtiene:



- Primera salida:  $y_1 = h_1(x) = x_1$ . La segunda derivada de  $y_1$ , es diferente de cero, por lo tanto el grado relativo de  $r_1 = 2$ .
- Segunda salida:  $y_2 = h_2(x) = x_2$ . La segunda derivada de  $y_2$ , es diferente de cero, por lo tanto el grado relativo de  $r_2 = 2$ .
- Grado Relativo Total:  $r = r_1 + r_2 = 4$ , que es igual al orden del sistema, por lo tanto se puede aplicar el método de linealización exacta.

### 5.1.3 Forma Normal para Linealización

Aplicando los conceptos de la subsecciones 3.2.3 y 3.3.3, se obtiene la transformación de coordenadas  $z = \Phi(x)$ , en la siguiente forma matricial:

$$z = \Phi(x) = \begin{bmatrix} h_1(x) \\ L_f h_1(x) \\ h_2(x) \\ L_f h_2(x) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \\ x_2 \\ x_4 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}$$

por lo tanto la nueva ecuación de estado es:

$$\dot{z} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} z_2 \\ v_1 \\ z_4 \\ v_2 \end{bmatrix} \quad (5.1)$$

### 5.1.4 Diseño del Controlador No Lineal MIMO

Para la nueva ecuación de estado (5.1) y aplicando la ecuación (3.36), se obtiene las nuevas entradas lineales equivalentes  $v_1$  y  $v_2$ , y se expresan de la siguiente forma, [12], [13], [14]:

$$v_1 = \dot{x}_3 = \ddot{p}_1 + K_1 \dot{\bar{y}}_1 + K_2 \bar{y}_1 \quad (5.2)$$

$$v_2 = \dot{x}_4 = \ddot{p}_2 + K_3 \dot{\bar{y}}_2 + K_4 \bar{y}_2 \quad (5.3)$$

donde:

$p_1$  y  $p_2$ : trayectorias deseadas para la primera y segunda entrada

$\bar{y}_1$ : error en la primera salida

$\bar{y}_2$ : error en la segunda salida

así mismo se tienen las siguientes equivalencias:

$$\bar{y}_1 = p_1 - y_1, \quad x_1 = y_1, \quad \dot{x}_1 = x_3 \quad (5.4)$$

$$\bar{y}_2 = p_2 - y_2, \quad x_2 = y_2, \quad \dot{x}_2 = x_4 \quad (5.5)$$

reemplazando (5.4) en (5.2), se obtiene:

$$v_1 = \dot{x}_3 = \ddot{p}_1 + K_1\dot{p}_1 + K_2p_1 - K_1x_3 - K_2x_1 \quad (5.6)$$

de la misma manera, reemplazando (5.5) en (5.3), se obtiene:

$$v_2 = \dot{x}_4 = \ddot{p}_2 + K_3\dot{p}_2 + K_4p_2 - K_3x_4 - K_4x_2 \quad (5.7)$$

se sabe que para trayectoria constante se tiene:

$$\dot{p}_1 = \dot{p}_2 = \ddot{p}_1 = \ddot{p}_2 = 0$$

reemplazando estos datos en (5.6) y (5.7) se obtiene:

$$v_1 = \dot{x}_3 = K_2p_1 - K_1x_3 - K_2x_1 \quad (5.8)$$

$$v_2 = \dot{x}_4 = K_4p_2 - K_3x_4 - K_4x_2 \quad (5.9)$$

donde,  $K_1$ ,  $K_2$ ,  $K_3$  y  $K_4$ , son los valores de la ecuación característica que depende de los polos del sistema lineal equivalente.

Con (5.8) y (2.33) se obtiene la ley control  $u_1$  para la primera entrada:

$$u_1 = P_1 + v_1R_{T1}(H_{11} + J_{eq1}) \pm C_cn_1R_{T1}x_3 \quad (5.10)$$

Con (5.9) y (2.34) se obtiene la ley control  $u_2$  para la segunda entrada:

$$u_2 = P_2 + v_2R_{T2}(H_{22} + J_{eq2}) \pm C_cn_2R_{T2}x_4 \quad (5.11)$$

donde:

$$P_1 = R_{T1}(B_{eq1}x_3 + C_1) + N_{T1}x_3$$

$$P_2 = R_{T2}(B_{eq2}x_4 + C_2 + d_2) + N_{T2}x_4$$

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

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

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

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

$$R_{T1} = \frac{R_a}{n_1 K_{m1} K_{A1}}$$

$$R_{T2} = \frac{R_a}{n_2 K_{m2} K_{A2}}$$

$$N_{T1} = \frac{n_1 K_{b1}}{K_{A1}}$$

$$N_{T2} = \frac{n_2 K_{b2}}{K_{A2}}$$

$$H_{11} = a_1 + a_2 \cos^2(x_2)$$

$$H_{22} = a_2$$

$$C_1 = -a_2 x_3 x_4 \sin(2x_2)$$

$$C_2 = a_2 x_3^2 \sin(x_2) \cos(x_2)$$

$$d_2 = a_3 \cos(x_2)$$

$a_1$ ,  $a_2$ , y  $a_3$ : parámetros que dependen de la masa y la inercia del proceso

$$a_1 = 2I_1$$

$$a_2 = I_2 + 2m_2 L_2 L_{c2} - m_2 L_2^2$$

$$a_3 = m_2 g L_{c2}$$

$C_c$ : constante de fricción de Coulomb

$H_{11}$ ,  $H_{22}$ : componentes de la matriz de inercia

$C_1$ ,  $C_2$ : componentes del vector fuerza centrífuga y Coriolis

$d_2$ : componente del vector fuerza de gravedad

$I_1$ : inercia de la base (prisma y disco)

$I_2$ : inercia del brazo

### 5.1.5 Programación del Algoritmo de Control No Lineal Multivariable

Para la programación del algoritmo de control se ha usado MATLAB [11], teniendo como dato las condiciones iniciales indicadas anteriormente y las ecuaciones (5.1), (5.8), (5.9), (5.10) y (5.11). Dicho programa se muestra en el apéndice B.

### 5.1.6 Simulaciones

Las simulaciones respectivas son mostradas desde la figura 5.2 hasta la 5.4. En la figura 5.2 se muestra las salidas controladas para la posición angular de la base para una referencia de  $90^\circ$  y posición angular del brazo para una referencia

de  $270^\circ$ , el tiempo de establecimiento para la base es de 1 seg. y para el brazo el tiempo de establecimiento es de 2.8 seg presentando un sobreimpulso mínimo.

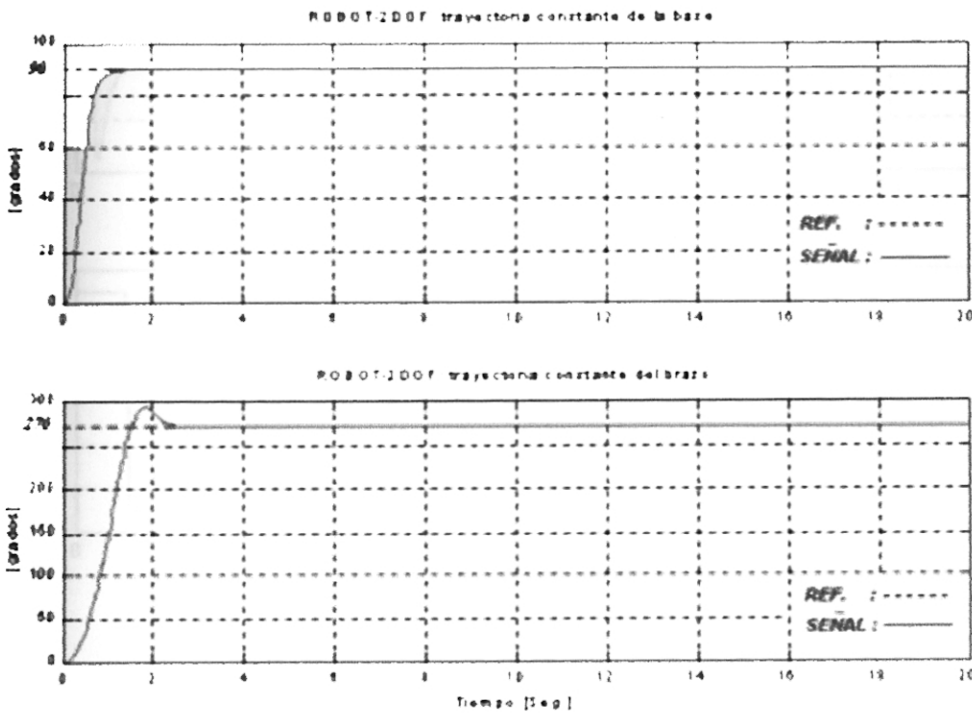


Figura 5.2: Trayectoria constante: salidas controladas para la base y el brazo.

En la figura 5.3 se muestran las señales de control para el servomotor de la base y para el servomotor del brazo. La señal del servomotor de la base varía de +5 voltios a -3.5 voltios por 0.5 seg., para luego aproximarse a cero a los 1.5 seg., mientras que la señal del servomotor del brazo varía de +5 voltios a -5 voltios por 2 seg., para luego aproximarse a cero a los 3 seg. Esto quiere decir que cuando las señales de control se aproximan a cero, el proceso ha sido controlado.

En la figura 5.4 se muestra la respuesta del sistema para la velocidad angular del servomotor de la base y del brazo, se aprecia que en el arranque la velocidad para ambos casos se incrementa para luego aproximarse a cero a los 1.5 seg. para el servomotor de la base y a los 3 seg. para el servomotor del brazo. Esto significa que cuando dichas velocidades se aproximan a cero, el proceso ha sido controlado.

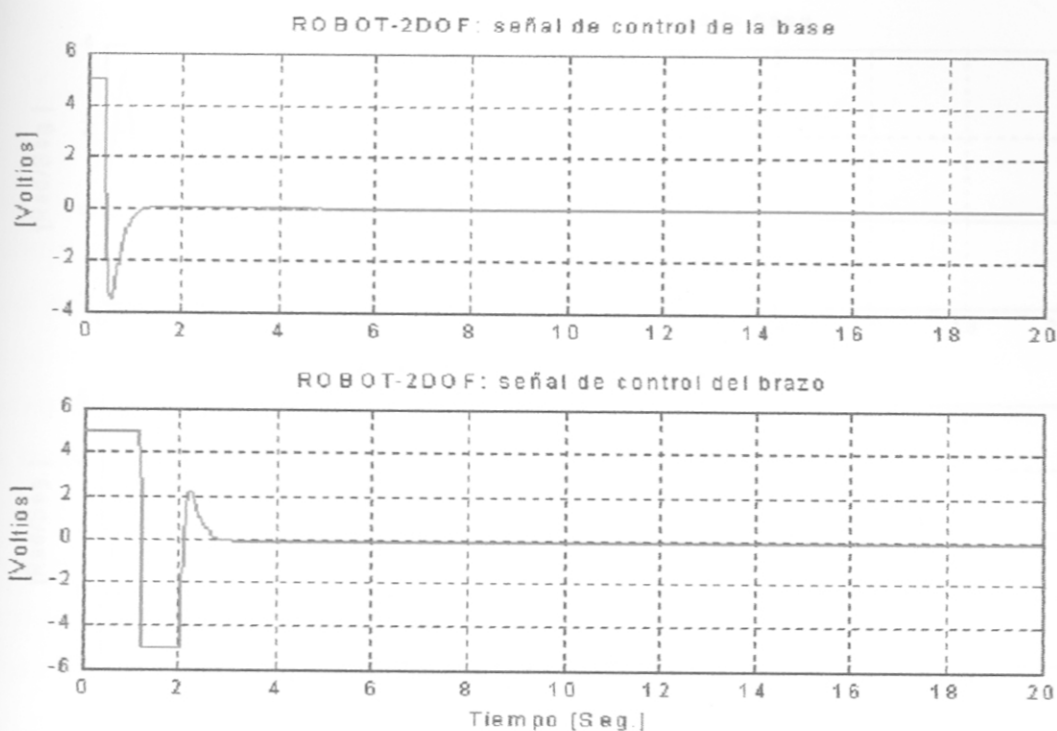


Figura 5.3: Trayectoria constante: señales de control para los servomotores de la base y el brazo.

## 5.2 Caso 2: Control de Trayectoria Senoidal

Para esta simulación se ha asumido una trayectoria senoidal con los siguientes datos:

- Condiciones iniciales:
  - $x_1 = 0$ , (base)
  - $x_2 = 0$ , (brazo)
- Trayectorias deseadas:
  - $p_1 = \pi/4 \cos(2\pi kT/21)$ , posición angular de la base
  - $p_2 = 2\pi/3 \cos(2\pi kT/21)$ , posición angular del brazo
- Polos usados:
  - para la base: -8 y -8
  - para el brazo: -7.5 y -8
- Tiempo de muestreo:
  - $T = 100Hz$

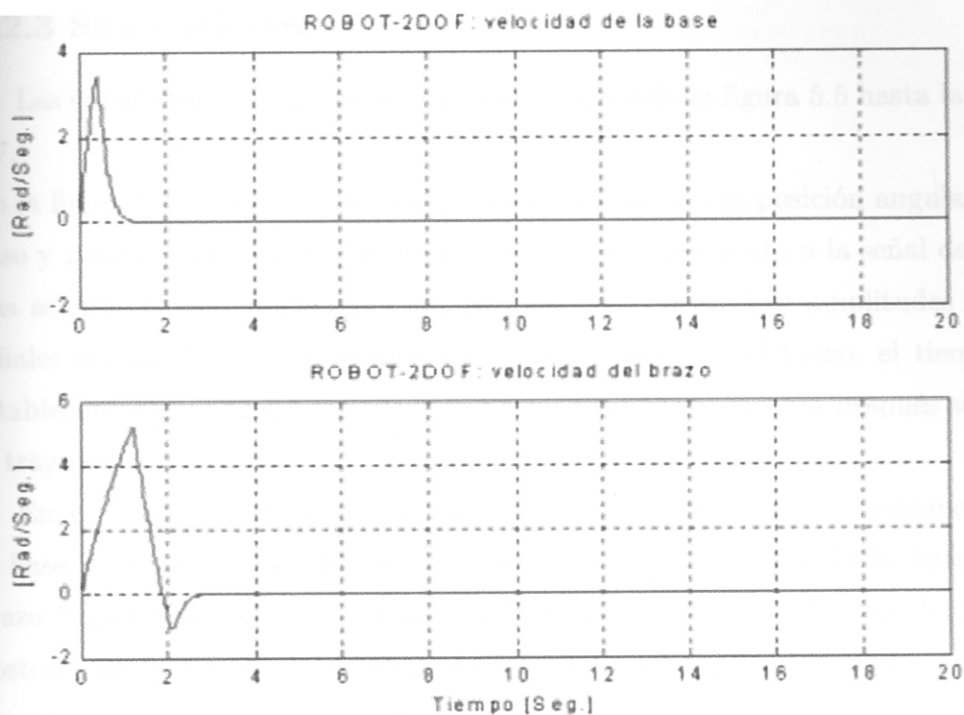


Figura 5.4: Trayectoria constante: velocidad para los servomotores de la base y el brazo.

### 5.2.1 Diseño del Controlador No Lineal MIMO

Para el diseño del controlador no lineal MIMO, se usan las ecuaciones (5.6) y (5.10) para la primera entrada y las ecuaciones (5.7) y (5.11) para la segunda entrada.

### 5.2.2 Programación del Algoritmo de Control No Lineal Multivariable

Para la programación del algoritmo de control se ha usado MATLAB [11], teniendo como dato las condiciones iniciales indicadas anteriormente y las ecuaciones (5.1), (5.6), (5.7), (5.10) y (5.11). Dicho programa se muestra en el apéndice B.

### 5.2.3 Simulaciones

Las simulaciones respectivas son mostradas desde la figura 5.5 hasta la figura 5.7.

En la figura 5.5 se muestra las salidas controladas para la posición angular de la base y posición angular del brazo, realizando un seguimiento a la señal deseada. Las señales se han fijado para un período de 21 Hz. Las amplitudes de las señales senoidales indican el movimiento de la base y del brazo, el tiempo de establecimiento es de 1.5 seg. para la base y para el brazo, para después seguir a la trayectoria deseada.

En la figura 5.6 se muestran las señales de control para el servomotor de la base y el servomotor del brazo. La señal del servomotor de la base y del brazo llega hasta 5 voltios en forma abrupta por un lapso de tiempo de 1.5 seg. Posteriormente las señales de control disminuyen en amplitud hasta hacerse bien pequeñas (0.2 y 0.5 voltios), pero siguiendo una trayectoria senoidal aproximada.

En la figura 5.7 se muestra la respuesta del sistema para la velocidad angular del servomotor de la base y velocidad angular del servomotor del brazo. Se aprecia que en el arranque la trayectoria de la velocidad se incrementa para ambos casos por un lapso de tiempo de 1.5 seg., para luego disminuir hasta una amplitud pequeña de 0.23 rad/seg para la base y 0.6 rad/seg para el brazo, y seguir la trayectoria senoidal.

## 5.3 Caso 3: Simulación con la Interfaz Gráfica de Usuario

Los programas de simulación anteriormente expuestos, han servido para llevar a cabo los cálculos requeridos y producir el ploteo para trayectorias constantes y senoidales. En esta sección se presenta el diseño de la interfaz gráfica para el usuario (GUI: Graphic User Interface) [11], la misma que emplea la estrategia de control por realimentación de estado basado en el método de linealización exacta y permite visualizar tales resultados en un panel de control para trayectoria

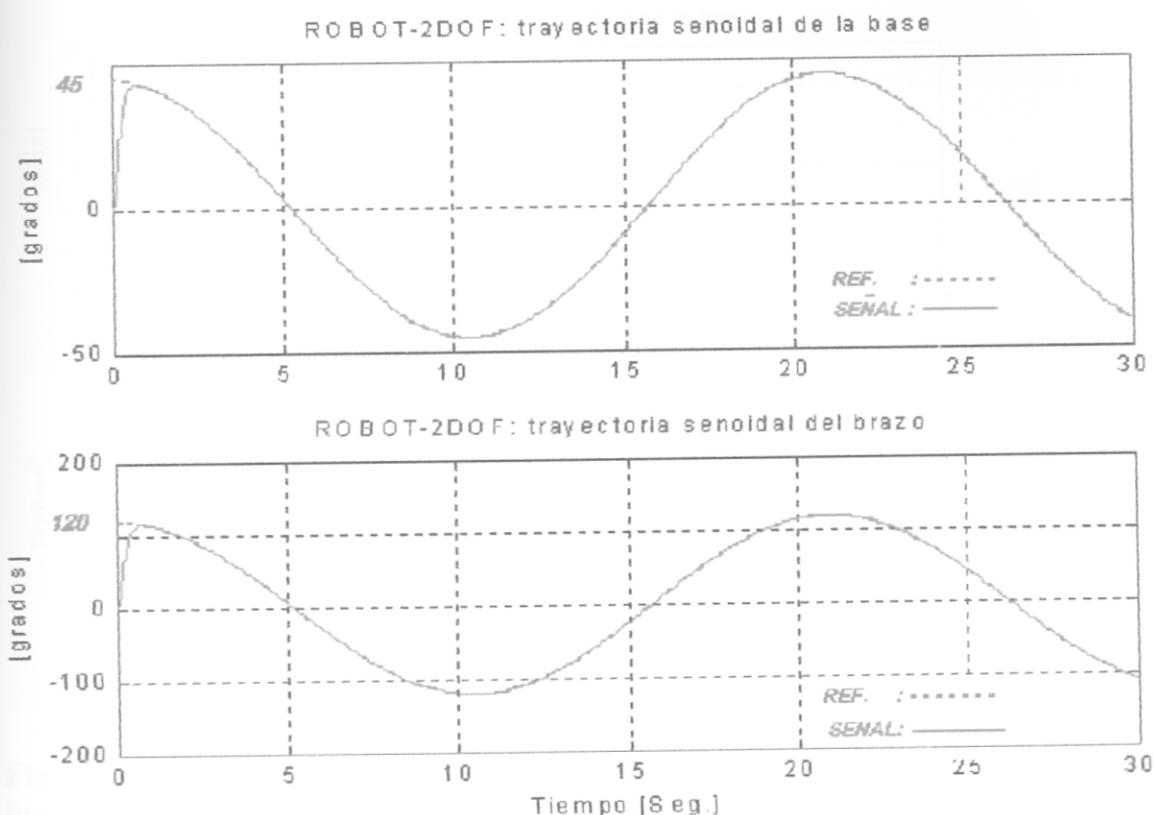


Figura 5.5: Trayectoria senoidal: salidas controladas para la base y el brazo

constante y para trayectoria senoidal.

El propósito del uso de la Interfaz Gráfica del Usuario es una ayuda visual para el usuario, que permite realizar cambios sin entrar al archivo de los programas. Los gráficos generados por la simulación son los estados del sistema que involucran la posición, la señal de control y la velocidad.

### 5.3.1 Simulación con GUI para Trayectoria Constante

Para ejecutar la GUI para trayectoria constante se necesitan 4 archivos desarrollados con MATLAB 5.3 [11]. Los archivos son copiados en un directorio de MATLAB y se ejecutan de acuerdo al siguiente orden:

- El programa *gr2\_pos8.m*, contiene la interfaz y el control, siendo de gran ayuda para que el usuario pueda manipularlos.
- El programa *r2pc\_g.m* contiene la función callback de la GUI y el archivo necesario para correr la simulación del algoritmo de control no lineal



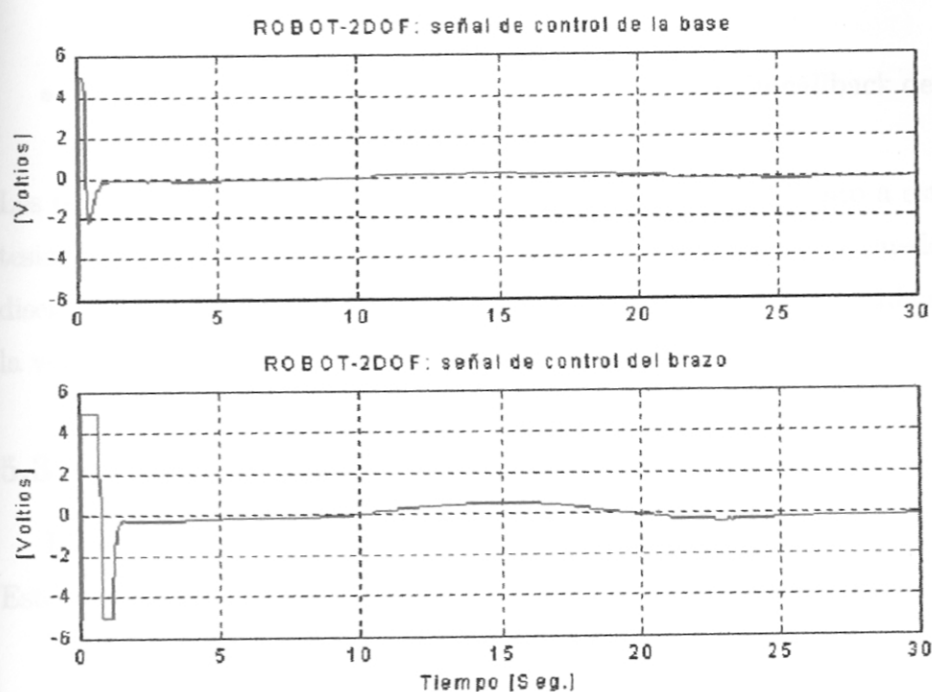


Figura 5.6: Trayectoria senoidal: señales de control para los servomotores de la base y el brazo

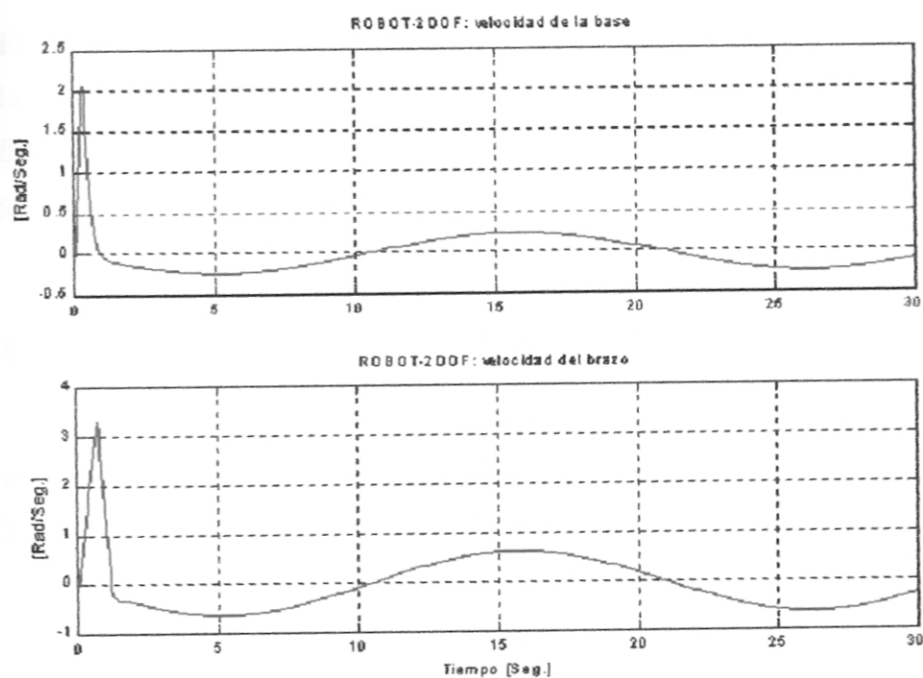


Figura 5.7: Trayectoria senoidal: velocidad de los servomotores de la base y el brazo.

multivariable.

- El programa *pos\_1.m* y *pos\_2.m*, contiene la función callback de la GUI para el funcionamiento de los slider.

Los programas descritos se encuentran en el CD-ROM adjunto a este trabajo de tesis. El panel de control se muestra en la figura 5.8. Como se puede observar, el diseño de la interfaz es de fácil uso y accesible para realizar modificaciones desde la ventana principal del computador.

### 5.3.1.1 Uso del Panel de Control de Trayectoria Constante

El primer grupo de controles contiene los pushbutton RUN, RESET y CLOSE.

Estos controles ejecutan lo siguiente:

- **RUN:** este botón ejecuta la simulación y plotea las respuestas de posición, señal de control y velocidad para los dos grados de libertad.
- **RESET:** este botón borra todos los gráficos ploteados, quedando listo para otra simulación.
- **CLOSE:** este botón cierra totalmente el panel.

El segundo grupo de controles permite el cambio de entradas de referencias para la base y el brazo, mediante el uso de los **SLIDER** donde se hace click en la flecha del respectivo slider para disminuir y/o avanzar el ángulo de posición en grados de  $0^\circ$  a  $360^\circ$  y su equivalente en radianes.

### 5.3.2 Simulación con GUI para Trayectoria Senoidal

Para ejecutar la GUI para trayectoria senoidal se necesitan 5 archivos desarrollados con MATLAB 5.3 [11]. Los archivos son copiados en un directorio de MATLAB y se ejecutan de acuerdo al siguiente orden:

- El programa *gr2\_sin1.m*, contiene la interface y el control, siendo de gran ayuda para que el usuario pueda manipularlos.
- Los programas *r2sin-g.m* y *r2cos-g.m*, contiene la función callback de la GUI y los archivos necesarios para correr la simulación del algoritmo de control no lineal multivariable.
- El programa *sin\_1.m* y *sin\_2.m*, contiene la función callback de la GUI para el funcionamiento de los slider.

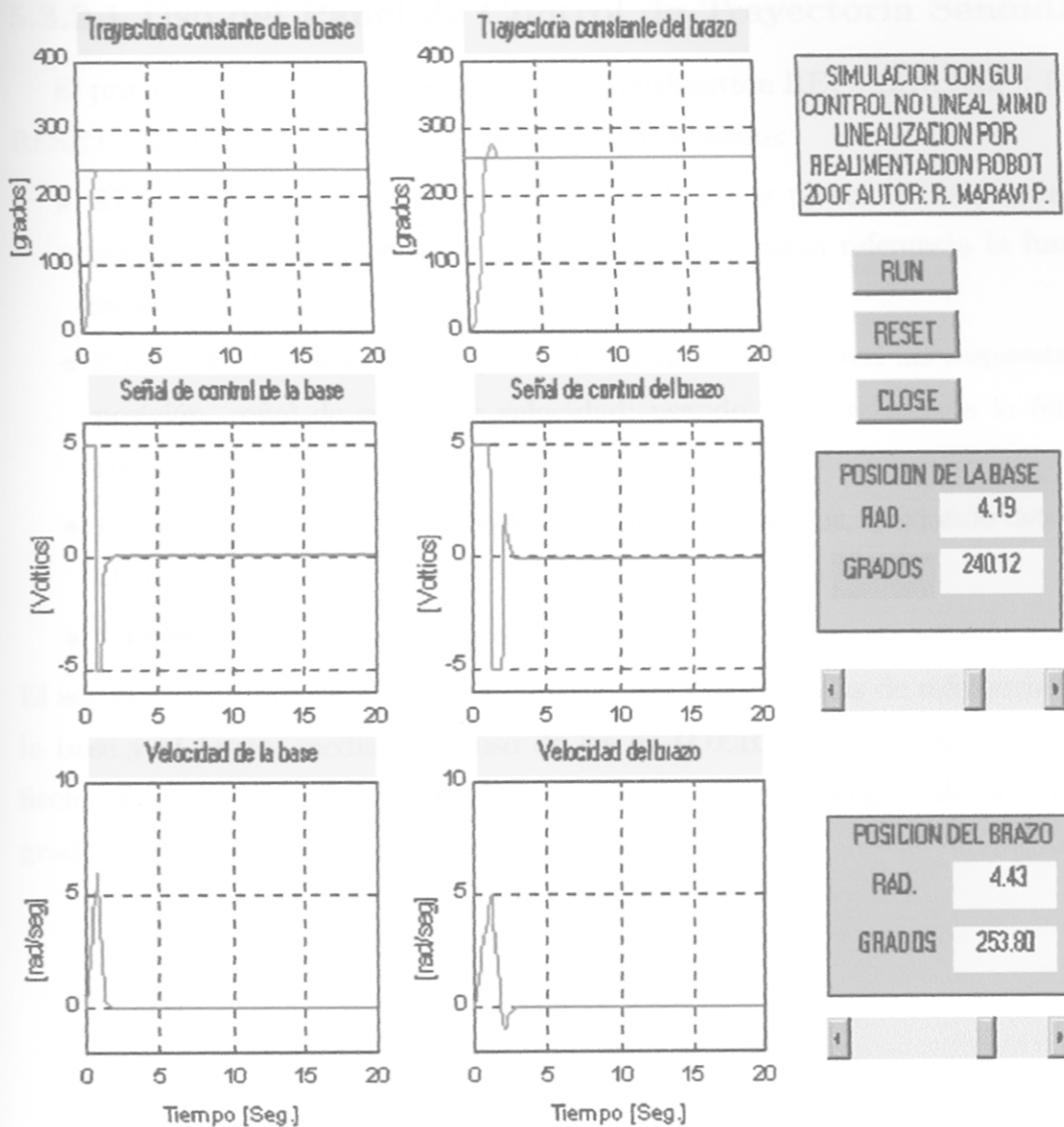


Figura 5.8: Panel de Control de Simulación de Trayectoria Constante del SRM de 2DOF con GUI.

Los programas descritos se encuentran en el CD-ROM adjunto a este trabajo de tesis. El panel de control se muestra en la figura 5.9. Como se puede observar, el diseño de la interfaz es de fácil uso y accesible para realizar modificaciones desde la ventana principal del computador.

### 5.3.2.1 Uso del Panel de Control de Trayectoria Senoidal

El primer grupo de controles contiene los pushbutton REF.: SIN, REF.: COS, RESET y CLOSE. Estos controles ejecutan lo siguiente:

- **REF.: SIN:** este botón ejecuta la simulación y plotea las respuestas de posición, señal de control y velocidad; usando como referencia la función seno para los dos grados de libertad.
- **REF.: COS:** este botón ejecuta la simulación y plotea las respuestas de posición, señal de control y velocidad; usando como referencia la función coseno para los dos grados de libertad.
- **RESET:** este botón borra todos los gráficos ploteados, quedando listo para otra simulación.
- **CLOSE:** este botón cierra totalmente el panel.

El segundo grupo de controles permite el cambio de entradas de referencias para la base y el brazo, mediante el uso de los **SLIDER** donde se hace click en la flecha del respectivo slider para disminuir y/o avanzar el ángulo de posición en grados de  $0^\circ$  a  $360^\circ$  y su equivalente en radianes.

## Simulaciones

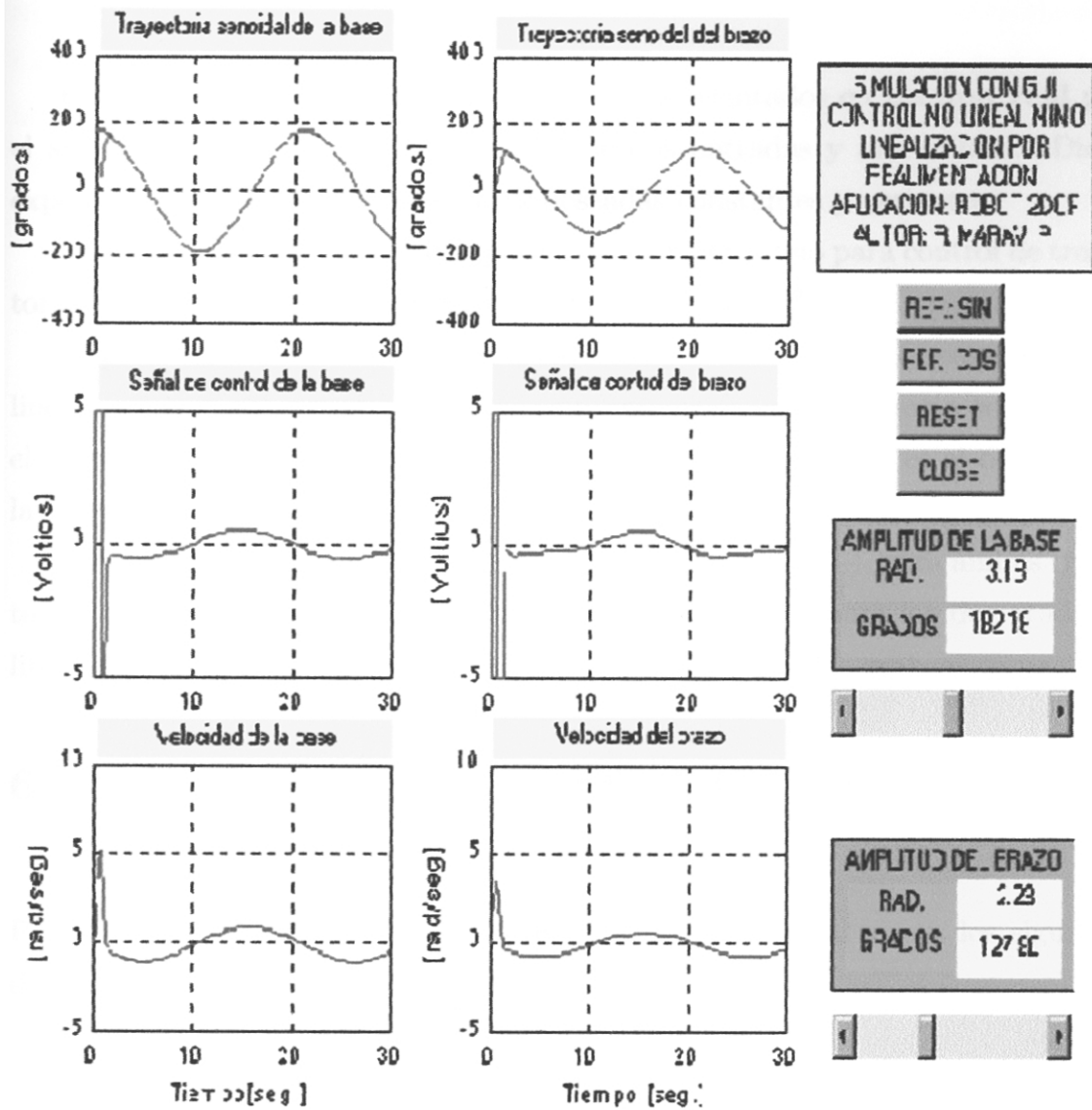


Figura 5.9: Panel de Control de Simulación de Trayectoria Senoidal del SRM de 2DOF con GUI.

## Capítulo 6

# Resultados Experimentales

En esta sección se presentan y discuten los resultados en **tiempo real** para el sistema de control no lineal MIMO de dos entradas y dos salidas. Dichos experimentos consideran trayectorias deseadas constantes y variantes.

Los resultados se han dividido para 2 experimentos uno para control de trayectoria constante y otro para control de trayectoria senoidal.

El programa usado en **tiempo real** es el Borland C++. El controlador no lineal multivariable empleado para la programación de trayectoria constante es el obtenido en la subsección 5.1.4, y para trayectoria senoidal es el obtenido en la subsección 5.2.1.

Los datos experimentales obtenidos del programa en C++ para ambas trayectorias, como **posición, señal de control y velocidad** para los dos grados de libertad, son ploteados en MATLAB para luego obtener los gráficos respectivos.

### 6.1 Experimento 1: Control de Trayectoria Constante

Para este experimento se ha usado una trayectoria constante con los siguientes datos:

- Condiciones iniciales:
  - $x_1 = 0$ , (base)
  - $x_2 = 0$ , (brazo)
- Trayectorias deseadas:
  - $p_1 = 3\pi/2$ , (posición angular de la base)
  - $p_2 = \pi$ , (posición angular del brazo)
- Polos usados:
  - para la base:  $-7.5 + 1.9365i$  y  $-7.5 - 1.9365i$
  - para el brazo:  $-7.5$  y  $-8$
- Tiempo de muestreo:  $T = 100Hz$

Los resultados obtenidos para este experimento se muestran desde la figura 6.1 hasta la figura 6.3.

En la figura 6.1 se muestra las salidas controladas para la posición angular de la base y posición angular del brazo, el tiempo de establecimiento es de 8 seg. para la base y 5 seg. para el brazo. Se observa que en la trayectoria de la base y del brazo no existe sobreimpulso. Se puede observar que en ambos casos la trayectoria desde el inicio hasta llegar al tiempo de establecimiento presenta ciertas oscilaciones.

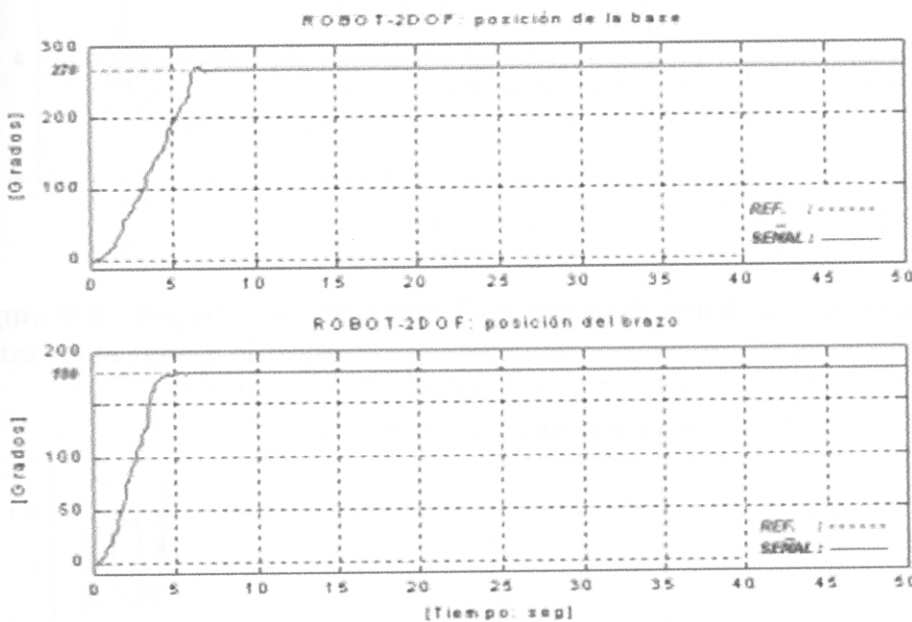


Figura 6.1: Trayectoria Constante Experimental: salidas controladas para la base y el brazo.

En la figura 6.2 se muestra las señales de control para ambos grados de libertad. La señal de control para el servomotor de la base llega hasta 3 voltios, presentando una oscilación por 8 seg. para luego converger aproximadamente a cero. Para el servomotor del brazo la señal llega hasta 3 voltios presentando una oscilación por 7 seg. para luego converger aproximadamente a cero.

En la figura 6.3 se muestra la respuesta al sistema para la velocidad angular del servomotor de la base y el brazo. En el gráfico se aprecia que en el arranque la velocidad en ambos se incrementa, para luego llegar a converger a cero a los 8 seg para la base y a los 7 seg para el brazo. Para ambos casos, la velocidad angular desde el inicio hasta converger a cero presenta oscilaciones abruptas.

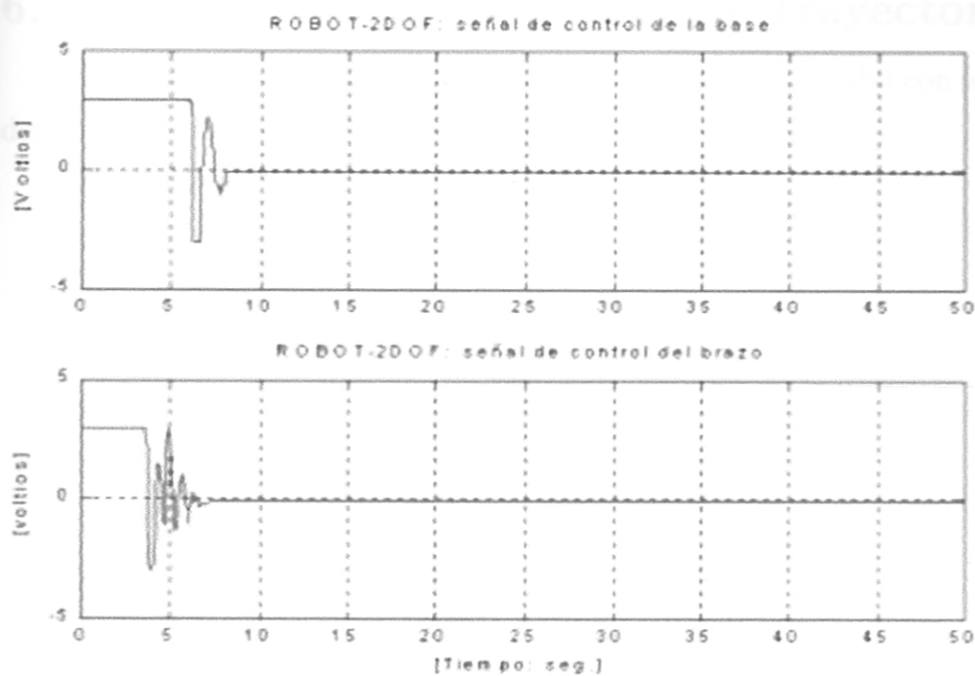


Figura 6.2: Trayectoria Constante Experimental: señal de control para el servomotor de la base y el brazo.

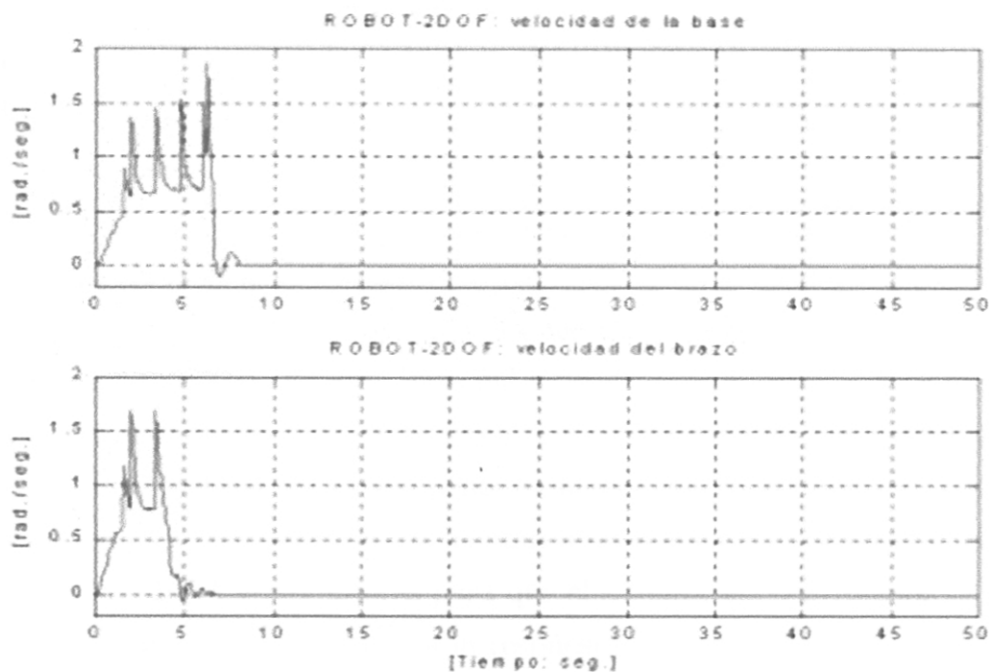


Figura 6.3: Trayectoria Constante Experimental: velocidad del servomotor de la base y el brazo.



## 6.2 Experimento 2: Control de Trayectoria Senoidal

Para este experimento se ha usado una trayectoria senoidal con los siguientes datos:

- Condiciones iniciales:  
 $x_1 = 0$ , (base)  
 $x_2 = 0$ , (brazo)
- Trayectorias deseadas:  
 $p_1 = \pi/2 \cos(2\pi kT/21)$ , posición angular de la base  
 $p_2 = \pi/2 \cos(2\pi kT/21)$ , posición angular del brazo
- Polos usados:  
para la base:  $-7.5 + 2.7839i$  y  $-7.5 - 2.7839i$   
para el brazo:  $-7.5 + 1.9365i$  y  $-7.5 - 1.9365i$
- Tiempo de muestreo:  $T = 100Hz$

Los resultados obtenidos para este experimento se muestran desde la figura 6.4 hasta la figura 6.6.

En la figura 6.4 se muestra las salidas controladas para la posición angular de la base y el brazo, y que a partir de los 8 seg. para ambos casos sigue a la referencia. Se observa que en la trayectoria de la base y el brazo no existe sobreimpulso. Las señales se han fijado a una frecuencia de  $1/21$  Hz de ciclo senoidal y con amplitudes de  $\pi/2$  rad y  $\pi/2$  rad, que indican el movimiento angular de la base y el brazo.

En la figura 6.5 se muestran las señales de control para el servomotor de la base y el servomotor del brazo. La señal de control para ambos llega hasta 3 voltios por un instante de tiempo (4 seg.), posteriormente presenta oscilaciones hasta el final del tiempo del experimento siguiendo una aproximación a la trayectoria senoidal. La oscilación que presentan ambas señales es debido al gran esfuerzo que realizan los servomotores para que el sistema sea controlado.

En la figura 6.6 se muestra la respuesta al sistema para la velocidad angular del servomotor de la base y el brazo. En el gráfico se aprecia que en el arranque la velocidad en ambos casos no es senoidal presentando ciertas oscilaciones, después de 6 seg. para ambos casos siguen a la trayectoria senoidal de la referencia con pequeñas oscilaciones, debido a que los servomotores están ejerciendo fuerza de control.

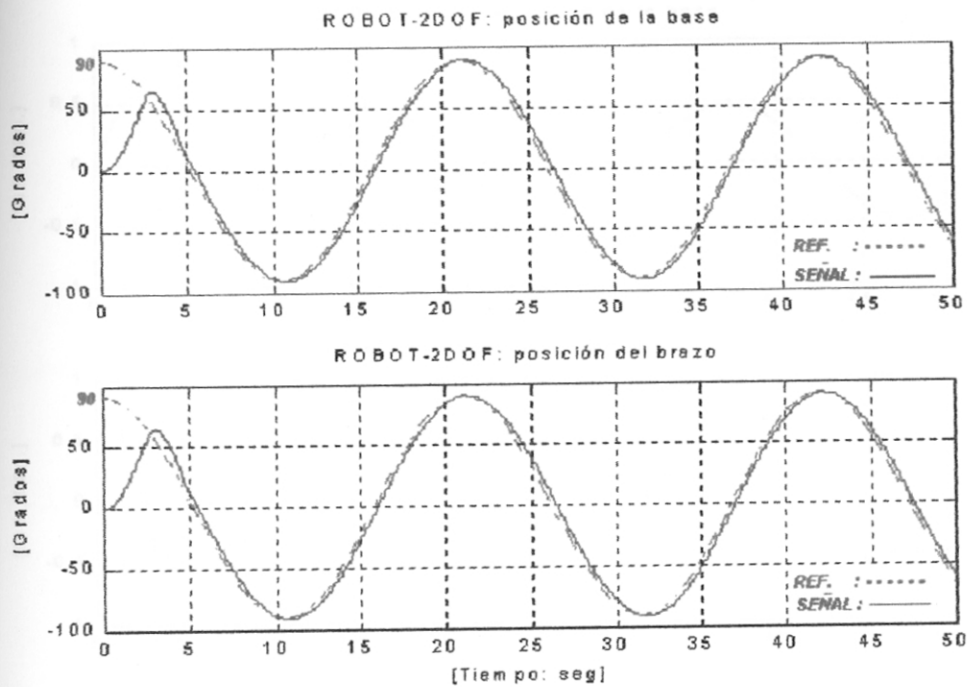


Figura 6.4: Trayectoria Senoidal Experimental: salidas controladas para la base y el brazo.

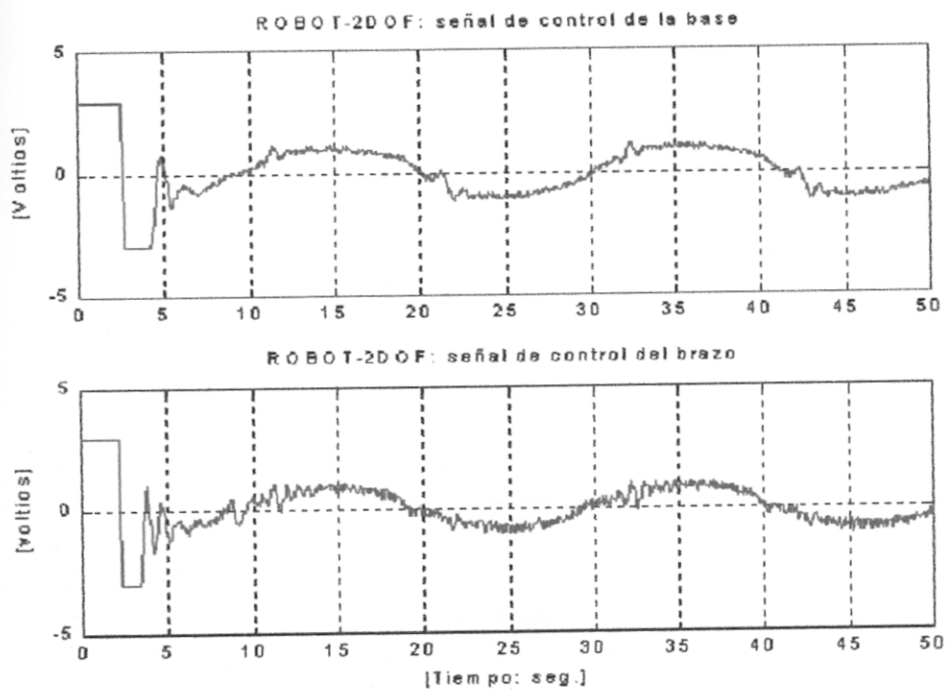


Figura 6.5: Trayectoria Senoidal Experimental: señal de control para el servomotor de la base y el brazo.

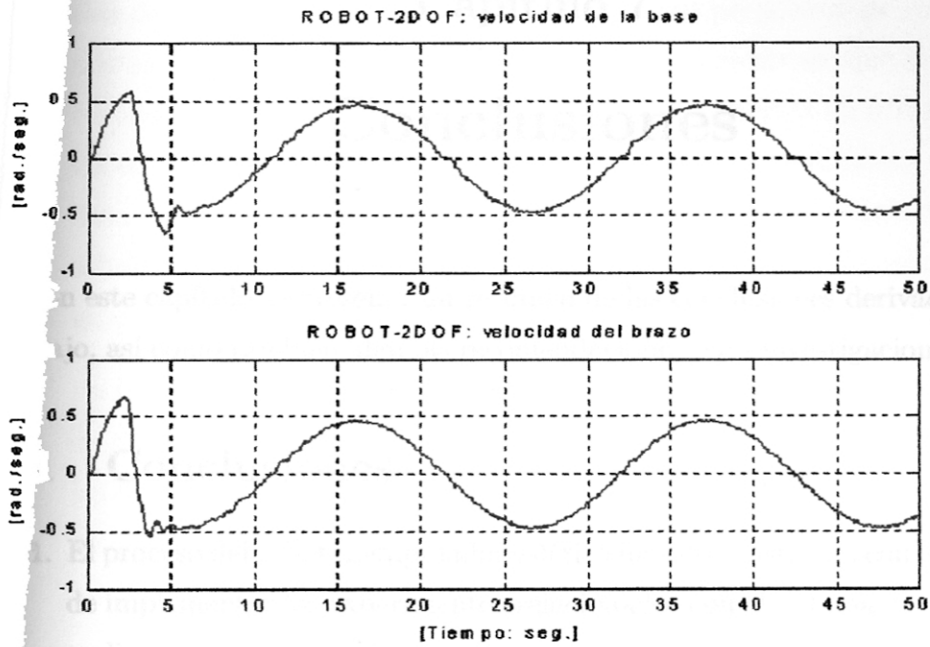


Figura 6.6: Trayectoria Senoidal Experimental: velocidad del servomotor de la base y el brazo.

## Capítulo 7

# Conclusiones

En este capítulo se presenta un resumen de las conclusiones derivadas de este trabajo, así como también algunas recomendaciones para investigaciones futuras.

### 7.1 Conclusiones

1. El proceso del robot manipulador esférico ha sido construido con el propósito de implementar los experimentos relacionados con el sistema de control por realimentación de estado así como también se puede usar para otras técnicas de control. En su construcción se ha optado por utilizar materiales de aluminio en la fabricación de la base y el brazo. En lo que respecta a la parte mecánica se ha tomado en cuenta el buen diseño de las bocinas de acoplamiento entre el eje de salida del servomotor y el eslabón, para evitar posibles desajustes que puedan ocasionar errores en la experiencia. El proceso construido se encuentra disponible en la SPGSE-FIEE para futuros experimentos.
2. Para el diseño del sistema de control fue necesario determinar el modelo dinámico del proceso, ya que la técnica de control no lineal empleada lo requiere. Los resultados experimentales determinados en tiempo real fueron satisfactorios según consta en los gráficos obtenidos en el capítulo 6. Tales resultados experimentales han demostrado que el sistema de control no lineal diseñado, es apto para el seguimiento de trayectorias arbitrarias de la base y del brazo del robot manipulador esférico.
3. El problema planteado fue el que las posiciones de la base y del brazo sigan trayectorias arbitrarias usando control por realimentación de estado. Esta estrategia de control, como es habitual, requiere del modelo del proceso controlado. Las simulaciones realizadas en el capítulo 5 han demostrado que el modelo obtenido en el capítulo 2 es válido y cumple con las exigen-

cias de un modelo no lineal multivariable con propósitos de control. Del mismo modo, tales simulaciones también han demostrado que el sistema de control no lineal multivariable empleando dicha técnica de control, cumple las condiciones de diseño impuestas previamente: seguimiento de las salidas controladas con respecto a señales arbitrarias de referencia con mínimo sobreimpulso, mínimo tiempo de estabilización y error en estado estable nulo.

1. En la implementación del sistema de control diseñado se perfeccionó el diseño de la tarjeta del amplificador de potencia. Los circuitos correspondientes a los diseños finales se muestran en el capítulo 4. Los experimentos realizados en esta tesis y otras, han demostrado el buen desempeño de esta tarjeta, la cual podrá ser empleada sin problemas para futuros trabajos.

## 7.2 Recomendaciones para Trabajos Futuros

Partiendo de los resultados obtenidos en este trabajo, se recomienda las siguientes tareas:

1. Diseño de un controlador que trabaje expuesto a cambios dinámicos desconocidos (disturbios) y que exijan al controlador afrontar con éxito el problema de seguimiento de trayectorias ante la presencia de dichos disturbios.
2. Para un diseño más completo, es necesario agregar un eslabón más y un efector final (por ejemplo una pinza rotórica) al sistema construido, de tal manera que se tenga un robot de 3 grados de libertad por controlar. El efector final puede contar dos servomotores para giro y apertura o cierre de la pinza, los cuales pueden ser monitoreados a lazo abierto. De esta manera, los diseños de los sistemas de control correspondientes, podrían ya ser orientados a aplicaciones industriales.
3. También se requiere investigar el control del manipulador robótico, cuando éste presente en su construcción juntas flexibles.
4. El proceso construido puede ser usado para futuros trabajos de tesis en las áreas de control no lineal avanzado e inteligencia artificial aplicado a la ingeniería de control.