

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



**DISEÑO DE UN SISTEMA DE CONTROL DE
POSICION ÓPTIMO MULTIVARIABLE PARA UN
MANIPULADOR ROBÓTICO ESFÉRICO**

TESIS

**PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO ELECTRÓNICO**

PRESENTADO POR:

NICANOR RAÚL BENITES SARAVIA

**PROMOCIÓN
1994-2**

**LIMA - PERÚ
2002**

A mis Padres

A mi Esposa

A mis Hijas

**DISEÑO DE UN SISTEMA DE CONTROL DE
POSICIÓN ÓPTIMO MULTIVARIABLE PARA UN
MANIPULADOR ROBÓTICO ESFÉRICO**

SUMARIO

La presente tesis trata sobre el diseño y simulación de un Sistema de Control Óptimo Multivariable para un Manipulador Robótico Esférico (SRM) de dos grados de libertad. El sistema está compuesto de una tarjeta "PWM+Amplificador de Potencia" (dos unidades), una tarjeta sensora compuesta por un par de decodificadores de cuadratura y dos unidades de contadores de 16 bits, una tarjeta de adquisición de datos LAB-PC+ de NATIONAL INSTRUMENTS insertada en el slot del PC, y dos servomotores DC con codificadores incrementales, localizados en la base y brazo del robot, respectivamente. En el extremo del brazo se ha colocado un efector final constituido por una pinza accionada por un motor paso a paso de 6 hilos.

La configuración de la primera versión del sistema de control combina en el diseño la representación del modelo no lineal del proceso, un observador de estados no lineal (NSO) para estimar los estados del proceso, y un controlador de realimentación de estados proporcional integral (PISFC: Proportional-Integral State Feedback Controller). Por consiguiente, el objetivo del control óptimo es encontrar una función de fuerza de control u capaz de minimizar la diferencia entre el vector de salida del proceso y y el vector de trayectoria deseada r , a pesar de la presencia de los parámetros inciertos y de las perturbaciones estocásticas gaussianas. Después de cada tiempo de muestreo, el NSO estima los estados del proceso \tilde{x} usando la data proporcionada por el vector de entrada u y el vector de salida y del proceso. El Controlador por realimentación de estados toma el vector de estados estimado \tilde{x} y la integral de la salida y para calcular el vector de control u , requerido para controlar las posiciones del SRM, o sea, lograr que las salidas del SRM se hagan iguales a las referencias en tiempo estable. El diseño de la segunda versión, comprende la aplicación de la estrategia de Control Óptimo Proporcional Integral con Observador Óptimo Discreto, con los mismos objetivos que en el diseño de la primera versión.

ÍNDICE

PRÓLOGO	1
CAPÍTULO I	
BREVE REFERENCIA A LA ROBÓTICA	2
1.1 Automatización y Robótica	2
1.2 Clasificación de los Robots	3
1.3 Aportes de la Robótica	4
1.4 Estructura de un Robot	5
1.5 Características de los Robots	5
1.6 Configuraciones de los Robots	8
1.7 Aplicaciones de la Robótica	10
CAPÍTULO II	
MODELADO DEL MANIPULADOR ROBÓTICO ESFÉRICO (SRM)	12
2.1 Componentes Principales del Robot Esférico	12
2.2 Obtención del Modelo Cinemático Directo por el Procedimiento de Denavit-Hartenberg	13
2.3 Obtención del Modelo Dinámico mediante el Procedimiento de Lagrange-Euler	16
2.4 Obtención del Modelo del Subsistema Eléctrico del SRM	23
2.5 El Modelo No Lineal del SRM en el Espacio de Estado y Simulación en Lazo Abierto	27
CAPÍTULO III	
IDENTIFICACIÓN DE PARÁMETROS DE LOS SERVOMOTORES	

DEL SRM	31
3.1 Pasos en la Identificación de Parámetros de los servomotores sin carga	31
3.2 Procedimiento para la Identificación de Parámetros de los servomotores sin carga a partir de la Respuesta Gráfica	32

CAPÍTULO IV

DISEÑO DEL SISTEMA DE CONTROL ÓPTIMO CONTINUO DEL SRM

4.1 Configuración del Sistema de Control Optimo No Lineal	38
4.2 Dinámica del Proceso No Lineal MIMO	39
4.3 Controlador por Realimentación de Estado MIMO	41
4.4 Márgenes de Estabilidad de Realimentación de Estados	48
4.5 Márgenes de Estabilidad y Robustez para el LQSFC	49
4.6 Observador de Estado No Lineal MIMO	50
4.7 Simulación del Sistema de Control Optimo con Observador No Lineal	54

CAPÍTULO V

DISEÑO DEL SISTEMA DE CONTROL ÓPTIMO DISCRETO DEL SRM

5.1 El problema del Control Optimo Cuadrático No Estacionario	61
5.2 El problema del Control Optimo Cuadrático Estacionario	62
5.3 El Controlador Optimo Proporcional	63
5.4 El Controlador Optimo Proporcional Integral	65
5.5 El Controlador Optimo Multivariable Proporcional Integral	68
5.6 El Observador Optimo Cuadrático SISO	73
5.7 El Observador Optimo Cuadrático MIMO	75
5.8 Simulación del Sistema de Control Optimo Multivariable Discreto	77

CAPÍTULO VI

PROPUESTA DE IMPLEMENTACIÓN DEL SISTEMA

6.1 Implementación del Hardware	82
6.2 Implementación del Software de Control	93

CAPÍTULO VII

EVALUACIÓN ECONÓMICA DEL PROYECTO	101
7.1 Costo de los Componentes	101
7.2 Costos de Diseño y Construcción	102
CONCLUSIONES Y RECOMENDACIONES	104
APÉNDICE A: LISTADO DE PROGRAMAS	106
APÉNDICE B: DIAGRAMAS CIRCUITALES	134
BIBLIOGRAFÍA	140

PRÓLOGO

La presente tesis trata sobre el diseño y simulación de un Sistema de Control Optimo Multivariable para un Manipulador Robótico Esférico (SRM) de dos grados de libertad. Este trabajo comprende siete capítulos.

El capítulo I trata en forma resumida sobre la Robótica y sus aplicaciones.

El capítulo II se ocupa del modelado del manipulador robótico, usando el modelo cinemático de Denavit-Hartenberg y el modelo dinámico de Lagrange - Euler, así como las ecuaciones del subsistema eléctrico.

El capítulo III cubre el proceso de identificación de los parámetros de los servomotores DC, que permitan validar los resultados de la simulación del proceso en lazo abierto, obtenidos en el capítulo II.

El capítulo IV abarca el diseño del sistema de control óptimo continuo del SRM, y muestra los resultados de la simulación para trayectorias y para referencias constantes (fijas).

El capítulo V abarca el diseño del sistema de control óptimo discreto con observador no lineal del SRM, y muestra asimismo, los resultados de la simulación para trayectorias y para referencias constantes.

El capítulo VI trata sobre la propuesta de implementación del sistema, tanto en hardware como en software.

El capítulo VII presenta la evaluación económica para la construcción del manipulador robótico de dos grados de libertad. Esta evaluación incluye el costo de los materiales, el de diseño y el de construcción.

Finalmente, se presenta las conclusiones de esta tesis y las recomendaciones para trabajos futuros.

CAPITULO I

BREVE REFERENCIA A LA ROBÓTICA

1.1 Automatización y Robótica

La automatización y la robótica son dos tecnologías estrechamente relacionadas. En un contexto industrial podemos definir la automatización como una tecnología que está relacionada con el empleo de sistemas mecánicos, electrónicos y basados en computadoras en la operación y control de la producción. Ejemplos de esta tecnología son: líneas de transferencia, máquinas de montaje mecanizado, sistemas de control de realimentación (aplicados a los procesos industriales), máquinas-herramienta con control numérico y robots. En consecuencia, la robótica es una forma de automatización industrial.

Hay tres clases amplias de automatización industrial: automatización fija, automatización programable y automatización flexible [6]. La robótica coincide más estrechamente con la automatización programable. Un robot industrial es una máquina programable de uso general que tiene algunas características antropomórficas o "humanoides". La característica humanoide más típica de los robots actuales es la de sus brazos móviles. El robot puede programarse para desplazar su brazo a través de una secuencia de movimientos con el fin de realizar una tarea de utilidad.

Existen ciertas dificultades sobre la definición de un robot industrial. La primera de ellas surge de la diferencia conceptual entre el mercado japonés y el euro-americano de lo que es un robot y lo que es un manipulador. Así, mientras que para los japoneses un robot industrial es cualquier dispositivo mecánico dotado de articulaciones móviles destinado a la manipulación, el mercado occidental es más restrictivo, exigiendo una mayor complejidad, sobre todo en lo relativo al control.

La definición más comunmente adoptada es la de la *Asociación de Industrias Robóticas* (RIA), según la cual: " *Un robot industrial es un manipulador*

multifuncional reprogramable, capaz de manipular materiales, piezas, herramientas o dispositivos especiales según trayectorias variables programadas para realizar tareas diversas ”.

Una peculiaridad de los robots, aparte de su estructura en forma de brazo mecánico, es su adaptabilidad a diferentes aprehensores o herramientas. Otra característica específica del robot, es la posibilidad de llevar a cabo trabajos completamente diferentes e, incluso, tomar decisiones según la Información procedente del mundo exterior, mediante el adecuado programa operativo de su sistema informático.

1.2 Clasificación de los Robots

La maquinaria para la automatización rígida dio paso al robot con el desarrollo de controladores rápidos, basados en el microprocesador, así como con el empleo de servos en bucle cerrado, que permiten establecer con exactitud la posición real de los elementos del robot y establecer el error con la posición deseada. Esta evolución ha dado origen a una serie de tipos de robots, que se mencionan a continuación:

1) Manipuladores

Son sistemas mecánicos multifuncionales, con un sencillo sistema de control, que permite gobernar el movimiento de sus elementos, según los modos manual, de secuencia y de secuencia variable. Muchas operaciones básicas pueden ser realizadas óptimamente mediante manipuladores, cuando las tareas son sencillas y repetitivas.

2) Robots de repetición o aprendizaje

Son manipuladores que se limitan a repetir una secuencia de movimientos, previamente ejecutada por un operador humano, haciendo uso de un controlador manual o un dispositivo auxiliar. El operador en la fase de aprendizaje, se vale de pulsadores o teclas, o bien de joytics, o a veces, desplaza directamente la mano del robot. Estos robots son muy conocidos en el ámbito industrial, y por el tipo de programación que incorporan, reciben el nombre de " *gestual* " [1], y [2].

3) Robots con control por computador

Son manipuladores multifuncionales, controlados por computador. A diferencia de los robots de repetición o aprendizaje, en este tipo de robots controlados por computador se dispone de un lenguaje de programación de bajo o alto nivel que permiten la implementación de un programa de aplicación, sin la intervención del manipulador (" *off-line* "), razón por la cual se le denomina *programación textual* [1], [2], [3], [4] y [7]. Este tipo de robots, van adquiriendo mayor demanda por sus grandes ventajas, lo que implica la necesidad de una preparación adecuada en software.

4) **Robots inteligentes**

Este tipo de robots son idénticos a los robots controlados por computador, con la diferencia de que pueden tomar decisiones en tiempo real, debido a la adición de sensores, que permiten sensar diferentes variables, como por ejemplo, temperatura, presión, posición, velocidad, etc.

Este tipo de robots no son muy conocidos y se encuentran en su etapa experimental. En tal sentido, la incorporación de la Visión Artificial, la Inteligencia Artificial, etc, son las ciencias que permitirán la fabricación de robots inteligentes, con capacidades de " *discernimiento* " parecidas a las del hombre [2], [3] y [4].

5) **Micro-robots o microbots**

Son robots de fines educativos, de entretenimiento o de investigación, con precios al alcance de la economía de los interesados, con estructura y funcionamiento similares a los robots industriales [5] y [1].

1.3 **Aportes de la Robótica**

- 1) Aumento de la productividad.
- 2) Mejora de la flexibilidad en la adaptación de la producción a series cortas.
- 3) Rápida amortización por sustitución de la mano de obra, mayor duración de las herramientas, menos pérdida de material y reducido mantenimiento.
- 4) Mejora de la calidad de los productos.
- 5) Optimización del rendimiento de las máquinas relacionadas con el robot.

- 6) Disminución de los stocks de productos terminados.
- 7) Realización de trabajos en condiciones y ambientes hostiles y peligrosos.

1.4 Estructura de un Robot

- 1) Manipulador o brazo mecánico.
- 2) Controlador basado en computador.
- 3) Elementos motrices o actuadores.
- 4) Elemento terminal, que puede ser una herramienta o aprehensor.
- 5) Sensores de información del entorno de funcionamiento.

1.5 Características de los Robots

Entre las principales características de los robots, podemos citar las siguientes:

- 1) **Grados de libertad:** determinados por el número de movimientos básicos (giros y desplazamientos) independientes. En la figura 1.1 se muestra un robot de 6 grados de libertad, para lo cual cuenta con 6 servomotores de corriente continua.
- 2) **Zonas de trabajo y dimensiones del manipulador:** La zona de trabajo queda definida por las dimensiones del manipulador y los grados de libertad. La zona de trabajo queda restringida por los límites de giro y desplazamiento de las articulaciones.
- 3) **Capacidad de carga:** Es muy importante considerar el peso que puede transportar el aprehensor o garra del manipulador. Esta característica recibe el nombre de *capacidad de carga*. Esta capacidad de carga debe especificarse bajo la condición de que el brazo del robot esté en su posición más débil, es decir cuando el brazo (cualquier sea su configuración) esté en su extensión máxima.
- 4) **Precisión en la repetibilidad:** Definiremos la precisión de la repetibilidad como el grado de exactitud en la repetición de los movimientos al

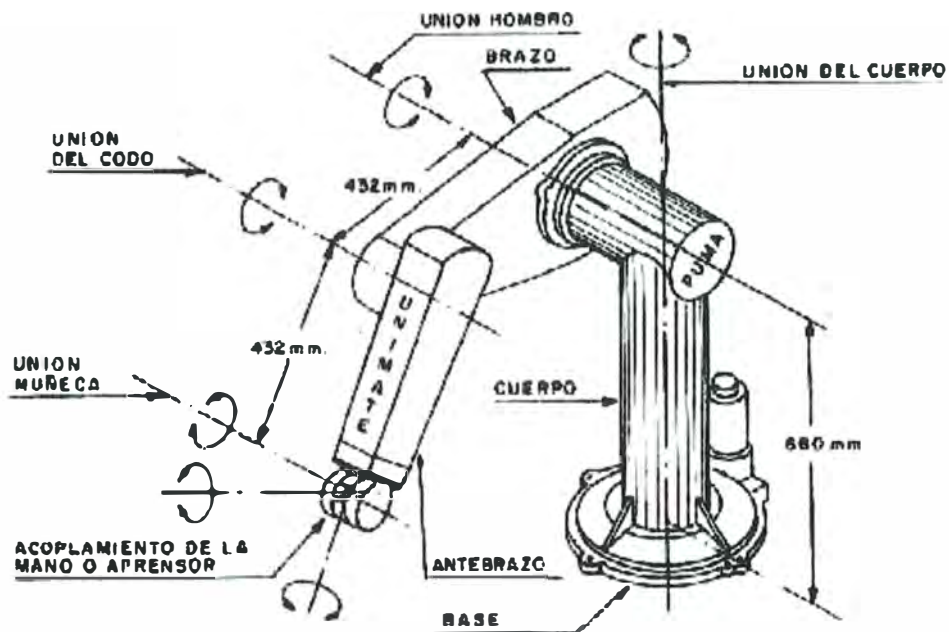
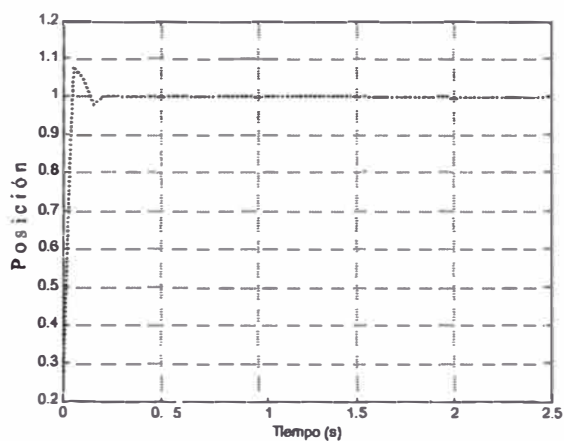


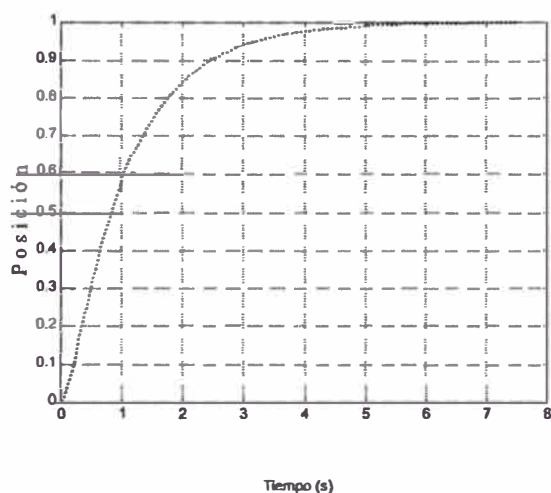
Figura 1.1: Esquema del manipulador PUMA 600 de UNIMATION, con 6 grados de libertad.

efectuar una tarea programada. La repetibilidad es, obviamente muy importante en robótica industrial, en la cual el robot debe repetir un gran número de veces la misma operación en distintas piezas. Esta característica está relacionada con el tipo de trabajo que deba realizarse. Así por ejemplo, en tareas de soldadura, pintura y manipulación de piezas, la precisión en la repetibilidad está comprendida entre 1 y 3 mm. y en las operaciones de mecanizado, la precisión ha de ser menor de 1 mm.

- 5) **Velocidad de respuesta y estabilidad:** Son dos características importantes del rendimiento dinámico en relación con el diseño de los sistemas de control. La velocidad de respuesta se refiere a la capacidad del robot para desplazarse a la siguiente posición en un breve período de tiempo. La estabilidad se suele definir como una medida de las oscilaciones que se producen en el brazo durante el movimiento desde una posición a la siguiente. En el diseño del sistema de control se puede amortiguar las oscilaciones transitorias, sin embargo, esto conlleva a una disminución en la velocidad de respuesta, tal como se puede ver en la figura 1.2.



(a)



(b)

Figura 1.2: Características de velocidad de respuesta y estabilidad: (a) baja amortiguación (respuesta rápida) y (b) alta amortiguación (respuesta lenta).

- 6) **Coordenadas de los movimientos:** Existen cuatro estructuras clásicas en los manipuladores, relacionadas con las coordenadas espaciales. Esta característica se enfocará en la siguiente sección.
- 7) **Tipos de actuadores:** Se entiende por actuadores, a los elementos que generan el movimiento de las articulaciones de los robots. Según el tipo de energía que consumen, pueden ser de tipo eléctrico, hidráulico o neumático.
- Entre los actuadores de tipo eléctrico, están los motores eléctricos (en la

gama de baja y media potencia).

Los actuadores de tipo hidráulico están destinados a tareas que requieren gran potencia y grandes capacidades de carga. Su precisión mecánica y su costo es elevado.

Los actuadores de tipo neumático presentan gran velocidad de respuesta y un bajo costo, pero están siendo sustituidos por elementos de tipo eléctrico.

- 8) **Programabilidad:** Los controladores de tipo microelectrónico en los robots, hacen posible la programación manual (o gestual) y la programación textual, usando para tal fin, de un terminal de pantalla, teclado y un lenguaje de programación.
- 9) **Capacidad de memoria:** La capacidad y el tipo de memoria del computador son muy importantes en las tareas de programación, sea esta de simulación y/o de control en tiempo real, que hagan posible el almacenamiento de la *data* de posición y/o velocidad, durante el tiempo de funcionamiento.

Entre otras características muy importantes están la visión artificial, la adaptabilidad con sensores de información, sincronización, etc.

1.6 Configuraciones de los Robots

La configuración mecánica de los robots lo proporcionan la estructura del manipulador y la relación entre sus elementos, dando origen al establecimiento de los parámetros, para definir la posición y orientación del elemento terminal (cinemática del robot). Existen cuatro configuraciones muy conocidas, relacionadas al tipo de coordenadas, según se muestra en la figura 1.3, y que a continuación se citan:

- Cartesianas.
- Cilíndricas.
- Polares.
- Angulares o articuladas.

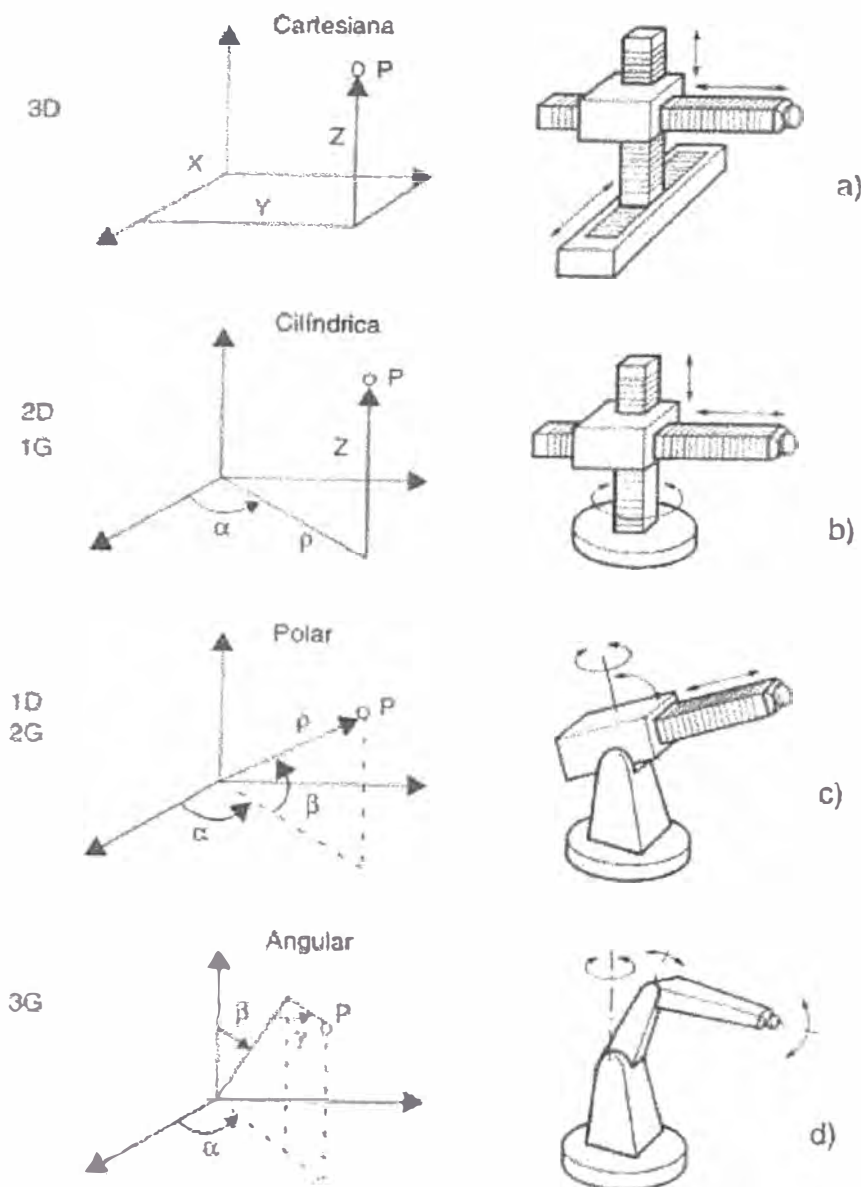


Figura 1.3: Configuraciones básicas de manipuladores, que se relacionan con los tipos de coordenadas espaciales, usadas en la definición de la posición y orientación del elemento terminal.

A la muñeca del manipulador robótico se acopla un elemento terminal, que puede ser una garra o una herramienta, que se encarga de realizar la tarea o el trabajo previsto. Las formas que adoptan estos terminales pueden ser muy variadas, dependiendo de las tareas que ha de desarrollar el robot. En la figura 1.4 se muestra un tipo de elemento terminal o efector final usado en los robots. Basándose en la configuración angular o articular, también se han desarrollado

robots de formas humanas, tal como puede apreciarse en la figura 1.5.

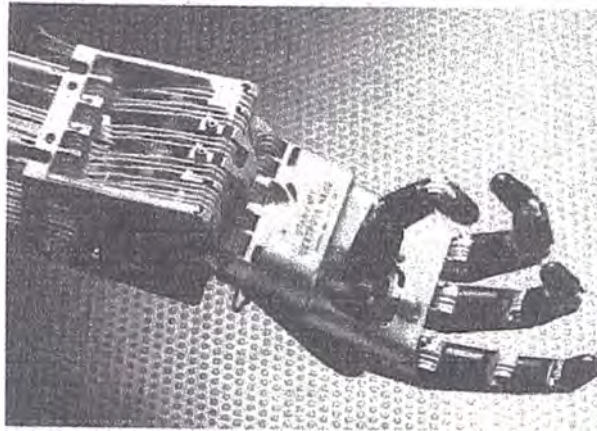


Figura 1.4: Fotografía de una mano robótica desarrollada en la Universidad de Utah.

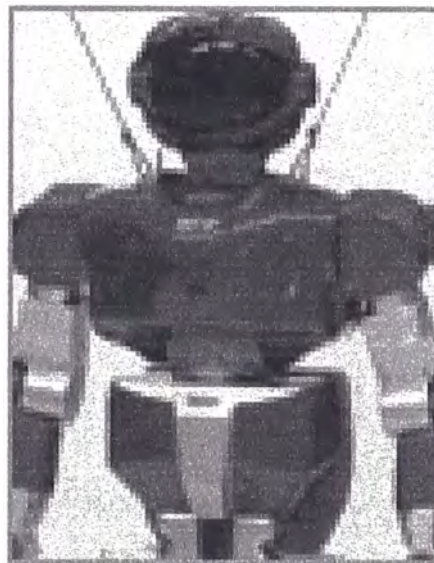


Figura 1.5: Fotografía de un robot humanoide.

1.7 Aplicaciones de la Robótica

La aplicación de los robots en las diferentes actividades del hombre se extiende por todo el mundo a un ritmo acelerado. Entre las diversas labores que realizan los robots podemos mencionar las siguientes:

- Carga y descarga de máquinas herramienta.

- Maquinaria de moldeo por inyección.
- Estampación en frío.
- Hornos.
- Fundición y estampación en caliente.
- Tratamientos térmicos.
- Paletización.
- Soldadura por puntos.
- Soldadura por arco.
- Soldadura por resistencia.
- Pintura.
- Pulverización a la llama.
- Forja, prensa y fundición.
- Esmaltado.
- Pulido.
- Cirugía asistida.
- Vigilancia y control de alarmas en zonas de seguridad.
- Colección de información y materiales en zonas peligrosas, como en las misiones espaciales, etc.

CAPÍTULO II

MODELADO DEL MANIPULADOR ROBÓTICO ESFÉRICO (SRM)

2.1 Componentes Principales del Robot Esférico

El proceso robot esférico de dos grados de libertad se muestra en la figura 2.1, que hace uso de dos servomotores D.C. Servo Systems GMX-7MC019B. Uno de ellos es usado en el hombro o base del robot, y el otro, en el brazo. Como efector final se usa una pinza accionada por un motor paso a paso de 6 hilos, que en este trabajo, no se modela, y cuyo control es en lazo abierto, usando para ello los cuatro bits más bajos del puerto C de la tarjeta de adquisición de datos. De la figura referida, puede observarse que el robot consta de dos estructuras:

1) Primera Estructura:

compuesta por un disco y un prisma de aluminio. En la parte inferior del disco de radio R y altura h , va acoplado, mediante una bocina, un motor DC (motor 1) colocado en posición vertical, para producir un movimiento rotatorio alrededor del eje z_0 . En la parte superior del disco va acoplada un prisma de sección cuadrada a y de altura b . La estructura descrita corresponde a la base del robot (primer grado de libertad).

2) Segunda estructura:

Corresponde a otro motor DC (motor 2) sujeto a la parte superior del prisma rectangular ya descrito, colocado en forma horizontal, para producir un movimiento rotatorio del brazo L_2 alrededor del eje z_1 (segundo grado de libertad). Sobre el eje de este segundo motor va acoplado una varilla rígida de aluminio de 3mm. de espesor, que cumple la función de un brazo. Sobre el extremo de esta varilla va conectado el relé electromagnético, ya indicado.

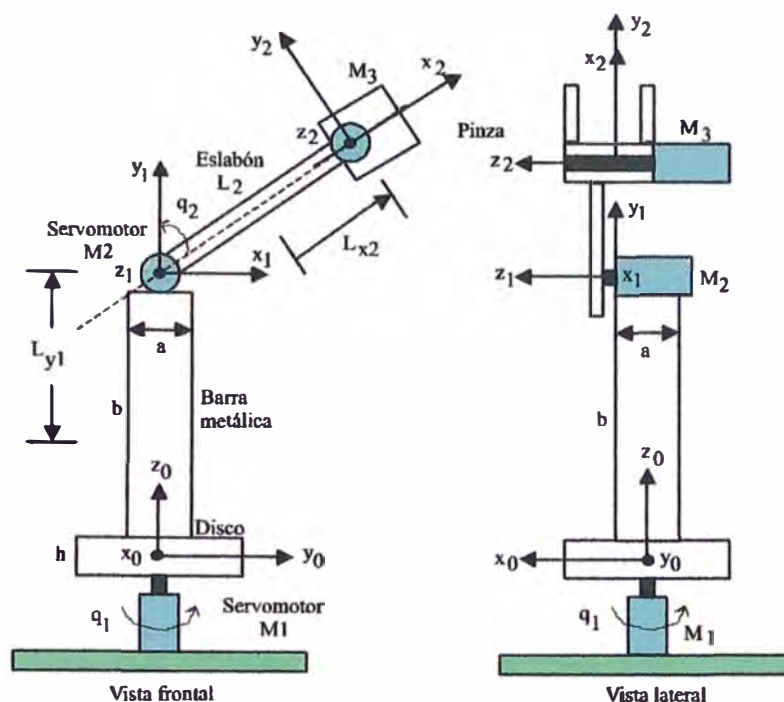


Figura 2.1: Esquema del manipulador robótico esférico (SRM).

De la descripción de los componentes principales del proceso, se puede deducir que tiene dos entradas de control: $K_{A1}u_1$ y $K_{A2}u_2$ aplicados a los terminales de armadura de los motores M_1 y M_2 , respectivamente, y dos salidas: θ_1 y θ_2 . Los parámetros K_{A1} y K_{A2} son las ganancias de cada amplificador. Las variables y parámetros del SRM se listan en la tabla 2.1.

2.2 Obtención del Modelo Cinemático Directo por el Procedimiento de Denavit-Hartenberg

Las ecuaciones diferenciales dinámicas de procesos mecánicos complejos, como en el caso del SRM, pueden ser obtenidos usando las ecuaciones de Lagrange-Euler (L-E), que consideran potenciales, energía cinética, fuerzas y torques. La formulación de Lagrange-Euler se basa en la representación cinemática directa del robot dada por Denavit-Hartenberg (ver [1], [2] y [7], para formular el modelo dinámico del robot. En tal sentido, antes de presentar la formulación de Lagrange-Euler, se presentará el procedimiento de Denavit-Hartenberg.

Un manipulador robótico está formado por un número de eslabones unidos a un número igual de articulaciones, de tal forma que el par articulación-eslabón

Tabla 2.1: Variables y parámetros del SRM.

Símbolo	Descripción	Valor	Unidades
m_d	Masa del disco	0.4	kg
m_b	Masa de la barra o prisma	1.0	kg
m_2	Masa del brazo (varilla)	0.1	kg
h	Altura del disco (espesor)	0.01	m
R_d	Radio del disco	0.07	m
b	Longitud de la barra	0.21	m
a	Longitud de la barra de sección cuadrada	0.044	m
L_{y1}	Long. del eje de ref. 1 al C.G. de la barra		m
L_2	Longitud del brazo del manipulador	0.3	m
L_{x2}	Long. del eje de ref. 2 al C.G. de la varilla	0.14	m
I_1	Moment of inertia of the base	0.0013	kg-m ²
I_2	Moment of inertia of the arm	0.0273	kg-m ²
J_{m1}	Momento de inercia de M_1	$1.4628e^{-6}$	kg-m ²
J_{m2}	Momento de inercia de M_2	$1.2055e^{-6}$	kg-m ²
B_{m1}	Constante de fricción de M_1	$2.0143e^{-7}$	N-m-s/rad
B_{m2}	Constante de fricción de M_2	$5.6877e^{-8}$	N-m-s/rad
J_{g1}, J_{g2}	Mom. inercia del sist. de red. de M_1/M_2	$5.63e^{-5}$	kg-m ²
B_{g1}, B_{g2}	Ctes. de fricción del sistema de reducción	$7.05e^{-5}$	N-m-s/rad
n_1, n_2	Relación de radios de M_1 y M_2	18.5	
R_{a1}, R_{a2}	Resistencias de armadura de M_1 y M_2	3.43	Ω
L_{a1}, L_{a2}	Inductancia de armadura de M_1 y M_2	$1.5e^{-3}$	H
V_{b1}, V_{b2}	Voltaje de f.c.e. de M_1/M_2		V
V_{a1}, V_{a2}	Voltaje de armadura máxima de M_1/M_2	24	V
I_{a1}, I_{a2}	Corriente de armadura máxima de M_1/M_2	0.7	A
K_{A1}, K_{A2}	Ganacia del amplificador M_1/M_2	14.07	
K_{t1}, K_{t2}	Constantes de los motores M_1/M_2	0.0437	N-m/A
K_{b1}, K_{b2}	Ctes. de f.c.e. de M_1/M_2	0.0437	V-s/rad
u_1, u_2	Voltajes de control máximo de M_1/M_2	1.4	V
g	Constante gravitacional	9.81	m/s ²

constituye un grado de libertad. Por consiguiente, a cada eslabón se le puede asociar un sistema de referencia solidario a él y, utilizando las transformaciones homogéneas, es posible representar las rotaciones y traslaciones relativas entre los distintos eslabones que componen el robot.

Según la representación de D-H, escogiendo adecuadamente los sistemas de coordenadas asociadas a cada eslabón, será posible pasar de uno al siguiente mediante 4 transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón.

Estas transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia del elemento i con el sistema del elemento $i - 1$. Dichas transformaciones son las siguientes:

1. Rotación alrededor del eje z_{i-1} un ángulo θ_i .
2. Traslación a lo largo de z_{i-1} una distancia d_i ; vector $d_i(0, 0, d_i)$.
3. Traslación a lo largo de x_i una distancia a_i ; vector $a_i(a_i, 0, 0)$.
4. Rotación alrededor del eje x_i un ángulo α_i .

El producto de esas matrices permite obtener la matriz de transformación homogénea de D-H, que relaciona el sistema de referencia del elemento i con el sistema del elemento $i - 1$. Entonces:

$$A_i^{i-1} = T(z, \theta_i)T(0, 0, d_i)T(a_i, 0, 0)T(x, \alpha_i) \quad (2.1)$$

que resulta en:

$$A_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

donde θ_i , a_i , d_i , α_i son los parámetros D-H del eslabón i . C y S las funciones coseno y seno, respectivamente.

La matriz de transformación homogénea del eslabón final i con respecto a la base, para un robot de n grados de libertad, viene dada por:

$$T = A_n^0 = \prod_{j=1}^i A_j^{j-1} = A_1^0 A_2^1 A_3^2 \dots A_i^{i-1} \quad i = 1, 2, \dots, n \quad (2.3)$$

Para el caso específico del robot manipulador esférico de dos grados de libertad (SRM), la tabla 2.2 muestra los parámetros D-H. Usando estos parámetros,

Tabla 2.2: Parámetros D-H del SRM.

Articulación	θ_i	d_i	a_i	α_i
1	q_1	$b + h$	0	90°
2	q_2	0	L_2	0°

obtenemos la matriz de transformación D-H siguiente:

$$T = A_2^0 = A_1^0 A_2^1 \quad (2.4)$$

Calculemos cada una de las componentes de la matriz de transformación D-H:

$$A_1^0 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & b+h \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad A_2^1 = \begin{bmatrix} C_2 & -S_2 & 0 & L_2 C_2 \\ S_2 & C_2 & 0 & L_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Luego, la matriz de transformación homogénea del eslabón final 2 con respecto a la base 0, viene dada por la ecuación 2.4, que resulta siendo:

$$A_2^0 = A_1^0 A_2^1 = \begin{bmatrix} C_1 C_2 & -C_1 S_2 & S_1 & L_2 C_1 C_2 \\ S_1 C_2 & -S_1 S_2 & -C_1 & L_2 S_1 C_2 \\ S_2 & C_2 & 0 & L_2 S_2 + b + h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

2.3 Obtención del Modelo Dinámico mediante el Procedimiento de Lagrange-Euler

La formulación de Lagrange-Euler (ver [1], [2] y [11]), utiliza las matrices de transformación homogénea A_i^{i-1} que relacionan el sistema de coordenadas de referencia del elemento i con el del elemento $i - 1$. Dicha formulación es la siguiente:

1. Asignar a cada eslabón un sistema de referencia de acuerdo a las normas de D-H.
2. Obtener matrices de transformación A_i^0 para cada elemento i . Para el robot esférico de dos grados de libertad estas matrices ya fueron halladas en la subsección anterior.
3. Obtener las matrices U_{ij} definidas por:

$$U_{ij} = \frac{\partial A_i^0}{\partial q_j} = \begin{cases} A_{j-1}^0 Q_j A_i^{j-1} & \text{si } j \leq i \\ 0 & \text{si } j > i \end{cases} \quad (2.7)$$

que representan el efecto del movimiento de la articulación j sobre todos los puntos del elemento i , y donde:

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{si la articulación } i \text{ es de rotación}$$

$$Q_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{si la articulación } i \text{ es de traslación}$$

Para el robot esférico de dos grados de libertad, las articulaciones son de rotación, por consiguiente:

$$Q_1 = Q_2 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.8)$$

Sustituyendo la ecuación 2.8 en 2.7, se obtiene:

$$U_{11} = \begin{bmatrix} -S_1 & 0 & C_1 & 0 \\ C_1 & 0 & S_1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad U_{12} = 0$$

$$\begin{aligned}
 \mathbf{U}_{21} &= \begin{bmatrix} -S_1C_2 & S_1S_2 & C_1 & -L_2S_1C_2 \\ C_1C_2 & C_1S_2 & S_1 & L_2C_1C_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 \mathbf{U}_{22} &= \begin{bmatrix} -C_1S_2 & -C_1C_2 & 0 & -L_2C_1S_2 \\ -S_1S_2 & S_1C_2 & 0 & -L_2S_1S_2 \\ C_2 & -S_2 & 0 & L_2C_2 \\ 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

4. Obtener las matrices U_{ijk} que representan los efectos de la interacción entre las articulaciones, definidas por:

$$U_{ijk} = \frac{\partial U_{ij}}{\partial q_k} = \frac{\partial}{\partial q_k} \left(\frac{\partial A_i^0}{\partial q_k} \right) = \left\{ \begin{array}{ll} A_{j-1}^0 Q_j A_{k-1}^{j-1} Q_k^{k-1} A_i & \text{si } i \geq k \geq j \\ A_{k-1}^0 Q_k A_{j-1}^{k-1} Q_j^{j-1} A_i & \text{si } i \geq j \geq k \\ 0 & \text{si } k > i \text{ o } j > i \end{array} \right\} \quad (2.9)$$

Para el robot manipulador esférico (SRM), las matrices U_{ijk} son:

$$\begin{aligned}
 \mathbf{U}_{111} &= \begin{bmatrix} -C_1 & 0 & -S_1 & 0 \\ -S_1 & 0 & C_1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 \mathbf{U}_{112} &= 0 \quad \mathbf{U}_{121} = 0 \quad \mathbf{U}_{122} = 0 \\
 \mathbf{U}_{211} &= \begin{bmatrix} -C_1C_2 & C_1S_2 & -S_1 & -L_2C_1C_2 \\ -S_1C_2 & -S_1S_2 & C_1 & -L_2S_1C_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 \mathbf{U}_{212} &= \begin{bmatrix} S_1S_2 & S_1C_2 & 0 & L_2S_1S_2 \\ -C_1S_2 & C_1C_2 & 0 & -L_2S_1C_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 \mathbf{U}_{221} &= \begin{bmatrix} S_1S_2 & S_1C_2 & 0 & L_2S_1S_2 \\ -C_1S_2 & -C_1C_2 & 0 & -L_2C_1S_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

$$U_{222} = \begin{bmatrix} -C_1C_2 & C_1S_2 & 0 & -L_2C_1C_2 \\ -S_1C_2 & S_1S_2 & 0 & -L_2S_1C_2 \\ -S_2 & -C_2 & 0 & -L_2S_2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

5. Obtener las matrices de pseudoinercias J_i para cada elemento, dadas por:

$$J_i = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int y_i x_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int z_i x_i dm & \int z_i y_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix}$$

$$J_i = \begin{bmatrix} \frac{-I_{x_i} + I_{y_i} + I_{z_i}}{2} & -I_{x_i y_i} & -I_{x_i z_i} & m_i \bar{x}_i \\ -I_{x_i y_i} & \frac{I_{x_i} - I_{y_i} + I_{z_i}}{2} & -I_{y_i z_i} & m_i \bar{y}_i \\ -I_{x_i z_i} & -I_{y_i z_i} & \frac{I_{x_i} + I_{y_i} - I_{z_i}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (2.10)$$

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

Para el SRM las pseudoinercias son J_1 y J_2 , donde $i=1$ corresponde a la base (disco + barra) e $i=2$ corresponde al eslabón L_2 . Los momentos de inercia del disco con respecto a la coordenada (x_1, y_1, z_1) (ver figura 2.1) y los correspondientes productos de los momentos de inercia están dados por:

$$I_{x_{1d}} = \frac{m_d(3R_d^2 + h^2 + 12(b + h/2))}{12} \quad I_{y_{1d}} = \frac{m_d R_d^2}{2} + \frac{1}{12} m_{M2}(h_{M2} + 3R_{M2}^2)$$

$$I_{z_{1d}} = I_{x_{1d}} \quad I_{x_{1d}y_{1d}} = I_{x_{1d}z_{1d}} = I_{y_{1d}z_{1d}} = 0$$

Los momentos de inercia de la barra con respecto a la coordenada (x_1, y_1, z_1) (ver figura 2.1) y los correspondientes productos de momentos de inercia están dados por:

$$I_{x_{1b}} = \frac{m_b(a^2 + 4b^2)}{12} \quad I_{y_{1b}} = \frac{m_b a^2}{6} \quad I_{z_{1b}} = I_{x_{1b}}$$

$$I_{x_{1b}y_{1b}} = I_{x_{1b}z_{1b}} = I_{y_{1b}z_{1b}} = 0$$

Entonces, los momentos de inercia de la base (disco + barra) son:

$$I_{x_1} = I_{x_{1d}} + I_{x_{1b}}$$

$$I_{y_1} = I_{y_{1d}} + I_{y_{1b}}, \quad I_{z_1} = I_{x_1}$$

luego J_1 toma la forma:

$$\mathbf{J}_1 = \begin{bmatrix} I_1 & 0 & 0 & 0 \\ 0 & I_{1a} & 0 & -m_1 L_{y1} \\ 0 & 0 & I_1 & 0 \\ 0 & -m_1 L_{y1} & 0 & m_1 \end{bmatrix} \quad (2.11)$$

donde $m_1 = m_d + m_b + m_{M2}$ es la masa de la base, L_{y1} es la distancia desde la coordenada (x_1, y_1, z_1) al C.G. de m_1 , y

$$I_1 = \frac{I_{y1}}{2} \quad y \quad I_{1a} = I_{x1} - I_1$$

La matriz \mathbf{J}_2 para $i = 2$ viene a ser:

$$\mathbf{J}_2 = \begin{bmatrix} I_2 & 0 & 0 & -m_2 L_{x2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -m_2 L_{x2} & 0 & 0 & m_2 \end{bmatrix} \quad (2.12)$$

donde $m_2 = m_L + m_H$ es la masa del brazo del SRM, L_{x2} es la distancia desde la coordenada (x_2, y_2, z_2) al C.G. de m_2 , y

$$I_2 = I_{y2} = I_{z2} = m_2 L_{x2}^2 / 3 + J_H$$

Tomar en cuenta que la masa de la mano m_H y el momento de inercia de la mano J_H también incluyen la masa y el momento de inercia de la carga, respectivamente. Considerando que $m_H \gg m_L$, entonces $m_2 \cong m_H$ y $L_{x2} \cong 0$.

6. Obtener la matriz de inercia simétrica H , de $n \times n$, relacionada con la aceleración de las variables de la articulación, y cuyos elementos son:

$$H_{ik} = \sum_{j=\max(i,k)} Tr[U_{jk} J_j U_{ji}^T] \quad i, k = 1, 2, \dots, n \quad (2.13)$$

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

Para el SRM, $i, k = 1, 2$, por lo que, empleando la ecuación 2.13, obtendremos los siguientes elementos de H:

$$H_{11} = Tr(U_{11}J_1U_{11}^T) + Tr(U_{21}J_2U_{21}^T) = 2I_1 + (I_2 + 2m_2L_{x2}L_2 - m_2L_2^2)\cos^2q_2$$

$$H_{12} = Tr(U_{22}J_2U_{21}^T) = 0 \quad H_{21} = Tr(U_{21}J_2U_{22}^T) = 0$$

$$H_{22} = Tr(U_{22}J_2U_{22}^T) = I_2 + 2m_2L_{x2}L_2 - m_2L_2^2$$

Entonces

$$H_{11} = 2I_1 + H_{22}\cos^2q_2$$

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

$$h_{ikm} = \sum_{j=\max(i,k,m)}^n Tr[U_{jkm}J_jU_{ji}^T] \quad i, k, m = 1, 2, \dots, n \quad (2.14)$$

Luego, para el SRM obtendremos:

$$h_{111} = Tr(U_{111}J_1U_{11}^T) + Tr(U_{211}J_2U_{21}^T) = 0$$

$$h_{112} = Tr(U_{212}J_2U_{21}^T) = (m_2L_2^2 - 2m_2L_2L_{x2} - I_2)\senq_2\cosq_2$$

$$h_{121} = Tr(U_{221}J_2U_{21}^T) = (m_2L_2^2 - 2m_2L_2L_{x2} - I_2)\senq_2\cosq_2$$

$$h_{122} = Tr(U_{222}J_2U_{21}^T) = 0$$

$$h_{211} = Tr(U_{211}J_2U_{22}^T) = (2m_2L_2L_{x2} - m_2L_2^2 + I_2)\senq_2\cosq_2$$

$$h_{212} = Tr(U_{212}J_2U_{22}^T) = 0$$

$$h_{221} = Tr(U_{221}J_2U_{22}^T) = 0$$

$$h_{222} = Tr(U_{222}J_2U_{22}^T) = 0$$

8. Obtener la matriz $n \times 1$ de fuerza de Coriolis y centrípeta c cuyos elementos vienen definidos por:

$$c_i = \sum_{k=1}^n \sum_{m=1}^n h_{ikm}\dot{q}_k\dot{q}_m \quad (2.15)$$

y que aplicado al SRM, tendremos:

$$\begin{aligned} C_1 &= h_{111}\dot{q}_1^2 + h_{112}\dot{q}_1\dot{q}_2 + h_{121}\dot{q}_2\dot{q}_1 + h_{122}\dot{q}_2^2 \\ &= 2(m_2L_2^2 - 2m_2L_2L_{x2} - I_2)\dot{q}_1\dot{q}_2\senq_2\cosq_2 \\ &= -2H_{22}\dot{q}_1\dot{q}_2\senq_2\cosq_2 \end{aligned}$$

$$\begin{aligned} C_2 &= h_{211}\dot{q}_1^2 + h_{212}\dot{q}_1\dot{q}_2 + h_{221}\dot{q}_2\dot{q}_1 + h_{222}\dot{q}_2^2 \\ &= (2m_2L_2L_{x2} - m_2L_2^2 + I_2)\dot{q}_1^2\senq_2\cosq_2 \\ &= H_{22}\dot{q}_1^2\senq_2\cosq_2 \end{aligned}$$

9. Obtener la matriz $n \times 1$ de fuerzas de gravedad d cuyos elementos vienen definidos por:

$$d_i = \sum_{j=1}^n (-m_j g U_{ji} \bar{r}_j^j) \quad i = 1, 2, \dots, n \quad (2.16)$$

donde

$$g = \begin{bmatrix} g_{x0} & g_{y0} & g_{z0} & 0 \end{bmatrix}$$

es el vector de gravedad expresado en el sistema de la base $\{S_0\}$ y

$$\bar{r}_j^j = \begin{bmatrix} \bar{x}_j & \bar{y}_j & \bar{z}_j & 1 \end{bmatrix}^T$$

(el elemento 1 es un factor de escala) es el vector de coordenadas homogéneas del centro de masas del elemento o eslabón j expresado en el sistema S_j .

Para el SRM obtendremos:

$$r_1^1 = [0 \quad -L_{y1} \quad 0 \quad 1]^T \quad r_2^2 = [-L_{x2} \quad 0 \quad 0 \quad 1]^T$$

$$g = [g_{x0} \quad g_{y0} \quad g_{z0} \quad 0] = [0 \quad 0 \quad -g \quad 0]$$

$$d_1 = -m_1 g U_{11} r_1^1 - m_2 g U_{21} r_2^2 \quad d_2 = -m_2 g U_{22} r_2^2$$

Remplazando los valores obtenidos en los pasos anteriores, se obtiene:

$$d_1 = 0 \quad d_2 = m_2 g L_{x2} \cos q_2$$

10. La ecuación dinámica del sistema será:

$$T = H\ddot{q} + c + d \quad (2.17)$$

donde T es el vector de torques y fuerzas aplicados a cada coordenada q_i . Siempre que $c = C\dot{q}$, el modelo de Lagrange de un proceso puede ser obtenido desde el modelo L-E dada por la ecuación 2.17, a saber

$$T = H\ddot{q} + C\dot{q} + d \quad (2.18)$$

En consecuencia, para el SRM tendremos que:

$$\mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{d}(\mathbf{q}) \quad (2.19)$$

donde K_{b1} es la constante de fuerza contra-electromotriz y n_1 is la relación de radios de los engranajes. Reemplazando la ecuación 2.23 en la ecuación 2.22 se obtiene:

$$V_{a1} = K_{A1}u_1 = L_{a1}\dot{I}_{a1} + R_{a1}I_{a1} + K_{b1}n_1\dot{q}_1 \quad (2.24)$$

El torque del servomotor T_{m1} es proporcional a la corriente I_{a1} , dado por:

$$T_{m1} = K_{m1}I_{a1} \quad (2.25)$$

donde k_{m1} es la constante del servomotor M_1 . Aplicando la ley de Newton para el torque, tendremos:

$$T_{m1} = J_{m1}n_1\ddot{q}_1 + B_{m1}n_1\dot{q}_1 + T_{g1} \quad (2.26)$$

El torque requerido para mover la base del SRM tiene la forma:

$$T_{g2} = n_1T_{g1} = J_{g1}\ddot{q}_1 + B_{g1}\dot{q}_1 + T_1 \quad (2.27)$$

entonces el torque del motor T_{m1} reflejado a la carga, se encuentra multiplicando las ecuaciones 2.25 y 2.26 por n_1 , así:

$$n_1T_{m1} = n_1K_{m1}I_{a1} \quad (2.28)$$

$$n_1T_{m1} = n_1^2J_{m1}\ddot{q}_1 + n_1^2B_{m1}\dot{q}_1 + n_1T_{g1} \quad (2.29)$$

Reemplazando la ecuación 2.27 en la ecuación 2.29 se tiene que:

$$\begin{aligned} n_1T_{m1} &= n_1^2J_{m1}\ddot{q}_1 + n_1^2B_{m1}\dot{q}_1 + J_{g1}\ddot{q}_1 + B_{g1}\dot{q}_1 + T_1 \\ n_1T_{m1} &= (n_1^2J_{m1} + J_{g1})\ddot{q}_1 + (n_1^2B_{m1} + B_{g1})\dot{q}_1 + T_1 \end{aligned} \quad (2.30)$$

Efectuando la siguiente asignación:

$$J_{eq1} = n_1^2J_{m1} + J_{g1}; \quad B_{eq1} = n_1^2B_{m1} + B_{g1}$$

la ecuación 2.30 queda reescrita como:

$$n_1T_{m1} = J_{eq1}\ddot{q}_1 + B_{eq1}\dot{q}_1 + T_1 \quad (2.31)$$

Ahora, igualando las ecuaciones 2.28 y 2.31 se obtiene:

$$n_1K_{m1}I_{a1} = J_{eq1}\ddot{q}_1 + B_{eq1}\dot{q}_1 + T_1 \quad (2.32)$$

donde T_1 es el torque de la carga. Sustituyendo la ecuación 2.20 en 2.32 se obtiene:

$$n_1 K_{m1} I_{a1} = J_{eq1} \ddot{q}_1 + B_{eq1} \dot{q}_1 + H_{11} \ddot{q}_1 + C_1 \quad (2.33)$$

Derivando la ecuación 2.33 se obtiene:

$$n_1 K_{m1} \dot{I}_{a1} = H_{11} q_1^{(3)} + J_{eq1} q_1^{(3)} + \dot{H}_{11} \ddot{q}_1 + B_{eq1} \dot{q}_1 + \dot{C}_1 \quad (2.34)$$

donde el superíndice (3) en q_1 denota la tercera derivada con respecto t .

Finalmente, la ley de control u_1 se obtiene reemplazando las ecuaciones (2.33), y (2.34) en (2.24)

$$\begin{aligned} u_1 = & L_{T1} (H_{11} q_1^{(3)} + J_{eq1} q_1^{(3)} + \dot{H}_{11} \ddot{q}_1 + B_{eq1} \dot{q}_1 + \dot{C}_1) \\ & + R_{T1} (J_{eq1} \ddot{q}_1 + B_{eq1} \dot{q}_1 + H_{11} \ddot{q}_1 + C_1) + N_{T1} \dot{q}_1 \end{aligned} \quad (2.35)$$

donde

$$L_{T1} = \frac{L_{a1}}{n_1 K_{A1} K_{m1}} \quad R_{T1} = \frac{R_{a1}}{n_1 K_{A1} K_{m1}} \quad N_{T1} = \frac{n_1 K_{b1}}{K_{m1}}$$

Modelo del servomotor M_2 (Brazo del SRM)

En la figura 2.3 se muestra el circuito eléctrico del servomotor M_2 . Usando el mismo procedimiento para obtener el voltaje de control de armadura u_1 , se encuentra que u_2 está dada por:

$$\begin{aligned} u_2 = & L_{T2} (H_{22} q_2^{(3)} + J_{eq2} q_2^{(3)} + B_{eq2} \ddot{q}_2 + \dot{C}_2 + \dot{d}_2) \\ & + R_{T2} (J_{eq2} \ddot{q}_2 + B_{eq2} \dot{q}_2 + H_{22} \ddot{q}_2 + C_2 + d_2) + N_{T2} \dot{q}_2 \end{aligned} \quad (2.36)$$

donde

$$L_{T2} = \frac{L_{a2}}{n_2 K_{A2} K_{m2}} \quad R_{T2} = \frac{R_{a2}}{n_2 K_{A2} K_{m2}} \quad N_{T2} = \frac{n_2 K_{b2}}{K_{m2}}$$

Despreciando las inductancias de armadura de los servomotores M_1 y M_2 , es decir, considerando L_{T1} y L_{T2} nulos en las ecuaciones 2.35 y 2.36, respectivamente, los voltajes de control u_1 y u_2 resultan:

$$u_1 = R_{T1} (J_{eq1} \ddot{q}_1 + B_{eq1} \dot{q}_1 + H_{11} \ddot{q}_1 + C_1) + N_{T1} \dot{q}_1 \quad (2.37)$$

$$u_2 = R_{T2} (J_{eq2} \ddot{q}_2 + B_{eq2} \dot{q}_2 + H_{22} \ddot{q}_2 + C_2 + d_2) + N_{T2} \dot{q}_2 \quad (2.38)$$

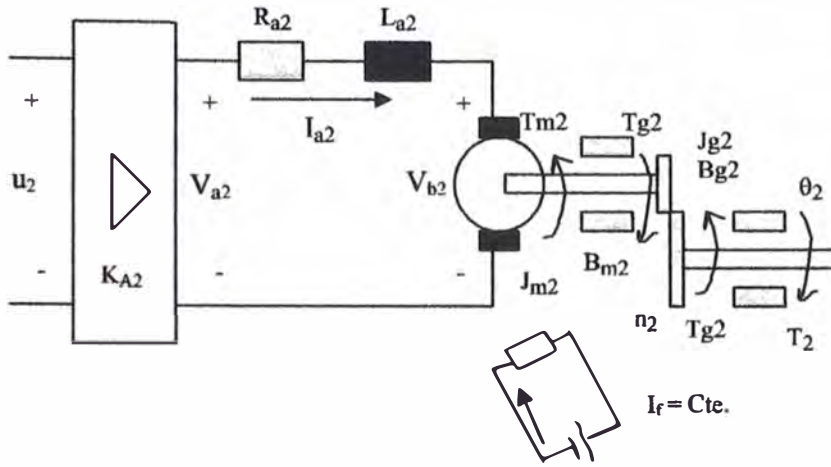


Figura 2.3: Diagrama del Servomotor M_2 para el movimiento del Brazo (Varilla) del SRM4

Reemplazando C_1 , C_2 y d_2 en las ecuaciones 2.37 y 2.38 y ordenando convenientemente, el modelo de Lagrange del SRM queda definido por:

$$\begin{aligned}
 & \begin{bmatrix} R_{T1}(J_{eq1} + 2I_1 + H_{22}\cos^2 q_2) & 0 \\ 0 & R_{T2}(J_{eq2} + H_{22}) \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} \\
 + & \begin{bmatrix} R_{T1}(B_{eq1} - H_{22}\dot{q}_2 \sin q_2 \cos q_2) + N_{T1} & -R_{T1}H_{22}\dot{q}_1 \sin q_2 \cos q_2 \\ R_{T2}H_{22}\dot{q}_1 \sin q_2 \cos q_2 & R_{T2}B_{eq2} + N_{T2} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \\
 & + \begin{bmatrix} 0 \\ R_{T2}m_2 g L_{x2} \cos q_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (2.39)
 \end{aligned}$$

que es de la forma:

$$M(q)\ddot{q} + P(q, \dot{q})\dot{q} + d(q) = u \quad (2.40)$$

donde

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}; \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$M(q) = \begin{bmatrix} R_{T1}(J_{eq1} + 2I_1 + H_{22}\cos^2 q_2) & 0 \\ 0 & R_{T2}(J_{eq2} + H_{22}) \end{bmatrix}$$

$$P(q, \dot{q}) = \begin{bmatrix} R_{T1}(B_{eq1} - H_{22}\dot{q}_2 \sin q_2 \cos q_2) + N_{T1} & -R_{T1}H_{22}\dot{q}_1 \sin q_2 \cos q_2 \\ R_{T2}H_{22}\dot{q}_1 \sin q_2 \cos q_2 & R_{T2}B_{eq2} + N_{T2} \end{bmatrix}$$

$$d(q) = \begin{bmatrix} 0 \\ R_{T2}m_2gL_{x2}\cos q_2 \end{bmatrix}$$

2.5 El Modelo No Lineal del SRM en el Espacio de Estado y Simulación en Lazo Abierto

El proceso SRM es de orden 4, por lo que eligiendo las siguientes variables de estado: $x_1 = q_1$ (posición angular de la base), $x_2 = q_2$ (posición angular del brazo), $x_3 = \dot{q}_1$ (velocidad angular de la base), y $x_4 = \dot{q}_2$ (velocidad angular del brazo), y derivando cada una de las variables de estado, se obtienen las ecuaciones de estado correspondientes, quedando la representación lagrangiana del SRM (ecuación 2.40), en la forma:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} = \mathbf{f}(\mathbf{x}) + \mathbf{g}_1(\mathbf{x})u_1 + \mathbf{g}_2(\mathbf{x})u_2 \quad (2.41)$$

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

donde

$$f_1(\mathbf{x}) = x_3 \quad f_2(\mathbf{x}) = x_4$$

$$f_3(\mathbf{x}) = \frac{-[R_{T1}(B_{eq1} - 2H_{22}x_4\text{sen } x_2\cos x_2) + N_{T1}]x_3}{R_{T1}(J_{eq1} + 2I_1 + H_{22}\cos^2 x_2)}$$

$$f_4(\mathbf{x}) = \frac{-(R_{T2}B_{eq2} + N_{T2})x_4 - R_{T2}(H_{22}x_3^2\text{sen } x_2 + m_2gL_{x2})\cos x_2}{R_{T2}(J_{eq2} + H_{22})}$$

$$G_{31}(\mathbf{x}) = \frac{1}{R_{T1}(J_{eq1} + 2I_1 + H_{22}\cos^2 x_2)}$$

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

La ecuación matricial en espacio de estado para las salidas $y_1 = x_1$ y $y_2 = x_2$ es de la forma:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (2.42)$$

El programa en código Matlab, que obtiene la respuesta gráfica del proceso SRM a entradas escalón unitario, en lazo abierto, se presenta a continuación:

```

% DATOS DEL SISTEMA
% SUB-SISTEMA MECANICO
% SOLIDO Y DISCO
  h=0.01; R=0.07; %ALTURA Y RADIO
% M_DISCO_TOTAL=M_DISCO+M_BOCINA=0.3+0.25
  m_dt=0.55;
% PRISMA
  b=0.21; %ALTURA
  a=0.044; %ANCHO Y LARGO DE LA SECCION DEL PRISMA
%M_PRISMA_TOTAL=M_PRISMA+M_MOTOR 2+M_ACCESORIOS=1+1.235+0.1+0.065+0.075
  m_pt=2.475;
% INERCIA DEL SOLIDO MAS MOTOR DC 2
  I1=0.5*(m_pt*(a^2)/6 + m_dt*(R^2)/2) + 0.0002;
%VARILLA
  m2=0.9; %MASA VARILLA+MASA MUÑECA+MASA MANO+MASA OBJETO=
    %
    % 0.1+0.2+0.4+0.2
  L2=0.3; % LONGITUD TOTAL
  Lx2=0.14; %LONG. DEL PTO. DE REFER. 2 AL CENTRO DE GRAVEDAD
  I2=m2*(L2^2)/3+0.00009+0.00009+0.0001; %INERCIA
  g =9.81;
% DATOS DEL SUB-SISTEMA ELECTRICO
  KK1=0.0437; Kt1=0.0437; La1=1.5e-3; Ra1=3.43; Ja1=1.4628e-6;
  n1=18.5; Ka1=14.07;
  KK2=0.0437; Kt2=0.0437; La2=1.5e-3; Ra2=3.43; Ja2=1.2055e-6;
  n2=18.5; Ka2=14.07;
  bm1=2.0143e-7; bm2=5.6877e-8;
% CARGA
  JL1=5.63e-5; bL1=7.05e-5; JL2=JL1; bL2=bL1;
% EQUIVALENCIA DE J y b
  J1=n1^2*Ja1 + JL1;
  b1=n1^2*bm1 +bL1;
  %J2=J1;
  %b2=b1;
  J2=n2^2*Ja2 + JL2;
  b2=n2^2*bm2 +bL2;
% EQUIVALENCIA DE INDUCTANCIA
  Lt1=La1/(n1*KK1*Ka1);
  Lt2=La2/(n2*KK2*Ka2);
% EQUIVALENCIA DE RESISTENCIAS
  Rt1=Ra1/(n1*KK1*Ka1);
  Rt2=Ra2/(n2*KK2*Ka2);
% EQUIVALENCIA DE N
  Nt1=n1*Kt1/Ka1;
  Nt2=n2*Kt2/Ka2;
% SIMPLIFICACION DE PARAMETROS DEL ROBOT
  a1=2*I1;
  a2=I2 + 2*m2*Lx2*L2 - m2*L2^2;
  a3=m2*g*Lx2;
% MODELO LINEALIZADO EQUIVALENTE DEL PROCESO DE 4to ORDEN (CONTINUO)

```

```

A=[0 0 1 0; 0 0 0 1; 0 0 -(Rt1*b1+Nt1)/(Rt1*(J1+a1+a2)) 0;...
    0 0 0 -(Rt2*b2+Nt2)/(Rt2*(J2+a2))];
B=[0 0; 0 0; 1/(Rt1*(J1+a1+a2)) 0;0 1/(Rt2*(J2+a2))];
C=[1 0 0 0; 0 1 0 0];
D=zeros(2,2);
% CALCULO DE LA CONTROLABILIDAD EN TIEMPO CONTINUO
M=[B A*B A^2*B A^3*B];
rangoM=rank(M); %RANGO DEL SISTEMA = 4
% CALCULO DE LA OBSERVABILIDAD EN TIEMPO CONTINUO
N=[C' A'*C' A'^2*C' A'^3*C'];
rangoN=rank(N);
% RANGO DEL SISTEMA = 4

t=0:0.00025:1.250;
subplot(2,1,1)
[y1,x3,t]=step(A,B,C,D,1,t); plot(t,y1,'-',t,x3,'--'); grid;
text(0.7,8,'y1 = x1 : posición de la base')
text(0.4,19,'x3 : velocidad de la base')
subplot(2,1,2)
[y2,x4,t]=step(A,B,C,D,2,t); plot(t,y2,'-',t,x4,'--'); grid;
text(0.7,8,'y2 = x2 : posición del brazo')
text(0.4,19,'x4 : velocidad del brazo')
print -f -deps srm4_la

```

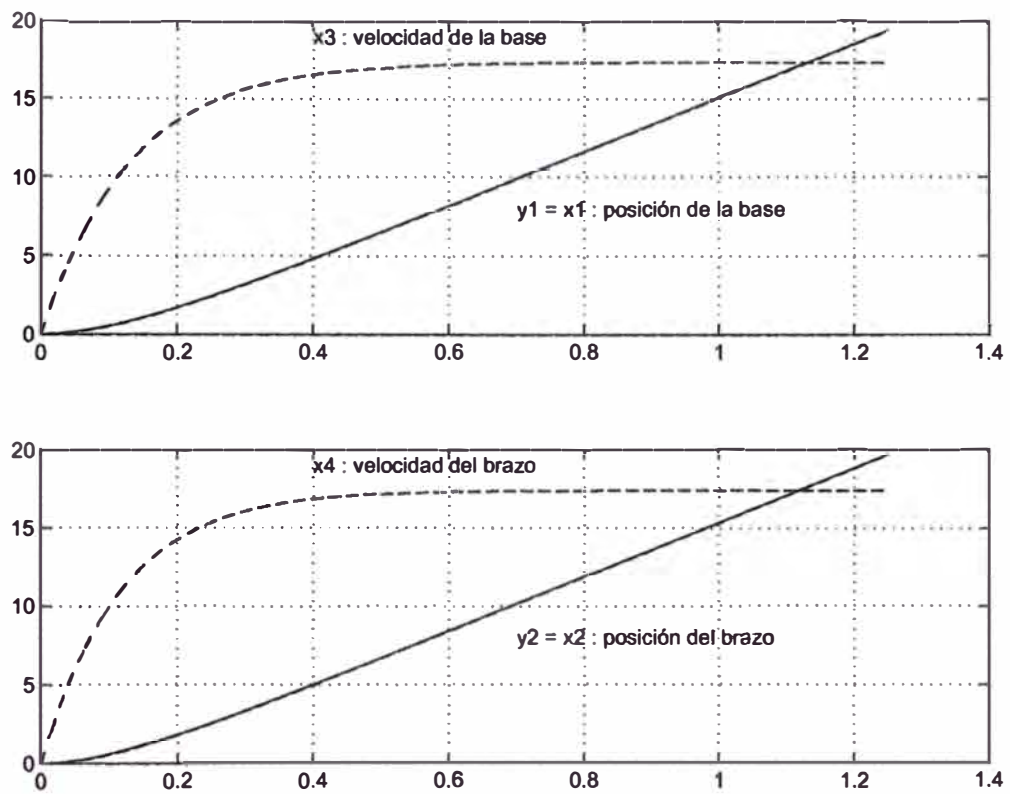


Figura 2.4: Respuesta en lazo abierto de posición y velocidad de la base y brazo del SRM4

CAPÍTULO III

IDENTIFICACIÓN DE PARÁMETROS DE LOS SERVOMOTORES DEL SRM

Este capítulo trata sobre la determinación de los parámetros de los servomotores del SRM, que permiten validar los resultados de la simulación. Existen muchos métodos para la identificación de parámetros agrupados en dos campos muy conocidos [8], como son la respuesta en el tiempo y respuesta en frecuencia; de entre ellos se usará el método de respuesta al escalón (respuesta en el tiempo), método experimental, cuyos resultados se utilizarán para la representación de un modelo muy popular, denominado modelo de primer orden. Varios parámetros listados en la tabla 2.1 y usados en la simulación, fueron hallados por el proceso de identificación que se presenta en este capítulo.

3.1 Pasos en la Identificación de Parámetros de los servomotores sin carga

- 1) Edición de un programa en C para la generación de un archivo de datos de velocidad del proceso en lazo abierto. Los archivos usados son:
 - *adqr.cpp* (contiene el programa principal para la data de velocidad)
 - *fun_d1.cpp* (contiene el programa de configuración de la tarjeta LAB-PC+, que trabaja con la librería *Nidaqbc.lib*)
 - *fun_d1.h* (contiene las funciones para el manejo de la tarjeta LAB-PC+)
 - *Nidaqbc.lib* (contiene las librerías de funciones para el manejo de la tarjeta LAB-PC+)
- 2) Creación del Proyecto *adqr.prj* (que incluye los archivos *adqr.cpp*, *fund1.cpp* y *Nidaqbc.lib*)

- 3) Ejecución del archivo *prj* (ejecución en tiempo real). Como resultado de la ejecución se generan los archivos de datos *dacr1_24.dat* y *dacr2_24.dat* para los motores de la base y del brazo, respectivamente.
- 4) Edición y ejecución del programa en código Matlab *vadqr.m*, que permite obtener los resultados gráficos de velocidad de los dos motores, en base a la información de los archivos de datos indicados.
- 5) Uso de un método de identificación de los parámetros del motor: resistencia de armadura R_a , constante de torque K_t , constante de fuerza contra-electromotriz K_b , fricción viscosa B y momento de inercia J . En esta tesis se usará el método gráfico de aproximación de la respuesta del proceso a uno de primer orden.

3.2 Procedimiento para la Identificación de Parámetros de los servomotores sin carga a partir de la Respuesta Gráfica

A continuación se presenta el procedimiento de identificación de los parámetros del servomotor de la base:

- 1) Resultado gráfico para el motor:
servosystem GMX-7MC019B (motor a ser usado en la base del robot 2dof)
Archivo C: *adqr.cpp* - Archivo M (para ploteo): *vadqr.m*
Archivo de datos: *dacr1_24.dat*
- 2) Trazar una recta tangente en el punto de inflexión de la curva, y anotar la amplitud K_{med1} y la constante de tiempo T_{med1} de la gráfica de velocidad. Los valores de amplitud de la velocidad y de la constante de tiempo son:
 $K_{med1} = 18.3 \text{ rad/seg}$, $T_{med1} = 0.8 \text{ seg}$.
- 3) Disponer de la información técnica del servomotor: Matsushita
Modelo : GMX-7MC019B; Alimentación: 24VDC/0.70A
velocidad sin carga: 255 RPM; Reducción: 18.5:1
- 4) Siguiendo el mismo análisis seguido en el capítulo II, se plantea el juego de ecuaciones para el motor M_1 (base del SRM), pero sin carga. Considerando

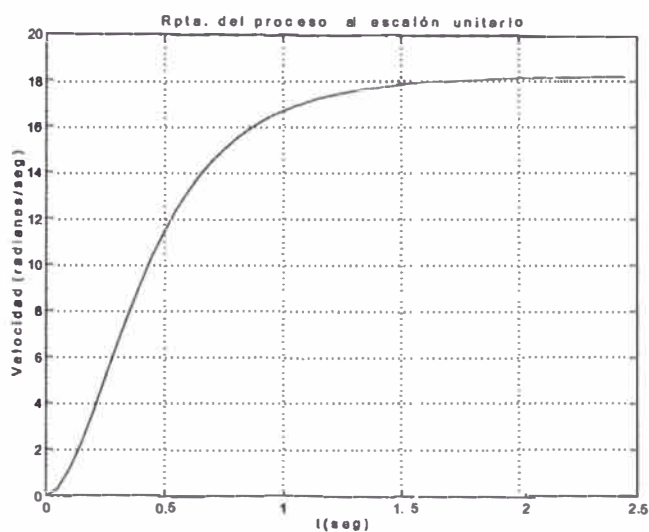


Figura 3.1: Respuesta del servomotor a una entrada escalón unitario.

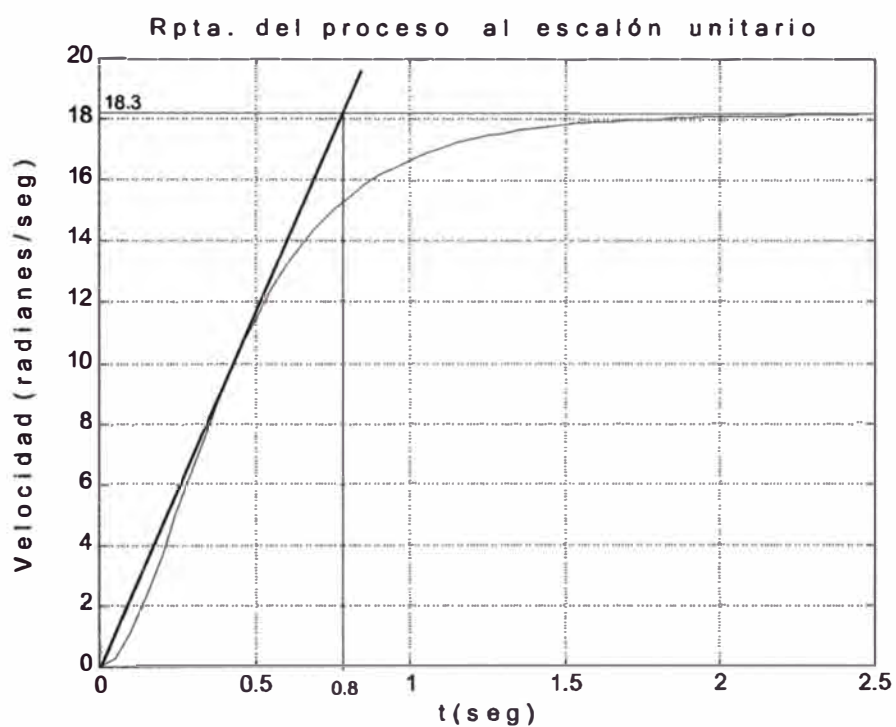


Figura 3.2: Trazado de la recta tangente en el punto de inflexión.

el circuito del motor DC controlado por armadura, que se muestra en la figura 3.3, se tiene que:

Asimismo, la velocidad en el motor es:

$$\omega_{m1} = n_1 \omega_1 \quad (3.8)$$

5) Determinación de la **Función de transferencia**

Tomando transformada de Laplace a la ecuación 3.3 se obtiene:

$$V_{a1}(S) = (R_{a1} + L_{a1}s)I_{a1}(s) + n_1 K_{b1} \omega_1(s) \quad (3.9)$$

e idénticamente, de las ecuaciones 3.4 y 3.5, se obtiene:

$$I_{a1}(s) = \frac{(J_{eq1}s + B_{eq1})\omega_1(s)}{K_{t1}} \quad (3.10)$$

Reemplazando la ecuación 3.10 en 3.9, obtendremos la siguiente función de transferencia:

$$\frac{\omega_1(s)}{V_{a1}(s)} = \frac{K_{t1}}{J_{eq1}L_{a1}s^2 + (L_{a1}B_{eq1} + R_{a1}J_{eq1})s + (R_{a1}B_{eq1} + K_{t1}K_{b1})} \quad (3.11)$$

Si consideramos L_{a1} despreciable, entonces obtendremos la siguiente función de transferencia de primer orden:

$$\frac{\omega_1(s)}{V_{a1}(s)} = \frac{K_{t1}}{R_{a1}J_{eq1}s + (R_{a1}B_{eq1} + K_{t1}K_{b1})} = \frac{K_{med1}}{T_{med1}s + 1} \quad (3.12)$$

donde:

$$K_{med1} = \frac{K_{t1}}{R_{a1}B_{eq1} + K_{t1}K_{b1}}; \quad T_{med1} = \frac{R_{a1}J_{eq1}}{R_{a1}B_{eq1} + K_{t1}K_{b1}} \quad (3.13)$$

6) La resistencia R_{a1} se obtiene de la siguiente relación:

$$R_{a1} = \frac{V_{a1}}{I_{a1} STALL} = \frac{24V}{7A} = 3.43\Omega \quad (3.14)$$

7) Las constantes de torque K_{t1} y la constante de fuerza contra-electromotriz K_{b1} se obtienen de las ecuaciones 3.4 y 3.3 respectivamente (considerando L_{a1} nulo en esta última ecuación):

$$K_{t1} = \frac{T_{m1} STALL}{I_{a1} STALL}; \quad \text{De la tabla 3.1: } n_1 T_{m1}|_{STALL} = 5.6550 Nm$$

$$\Rightarrow T_{m1} STALL = \frac{5.6550 Nm}{18.5} = 0.3057 Nm/A$$

$$K_{t1} = \frac{0.3057 Nm}{7 A} = 0.0437 Nm/A \quad (3.15)$$

$$K_{b1} = \frac{V_{a1} - R_{a1}I_{a1}}{n_1 \omega_1} = \frac{24V - 3.43\Omega \times 0.7 A}{18.5 \times 26.7035 rad/s} = 0.0437 V/rad/s \quad (3.16)$$

- 8) Reemplazando los valores $K_{med1} = 18.3 \text{ rad/seg}$, $T_{med1} = 0.8 \text{ seg}$ (del paso (2)), y usando los resultados de R_{a1} y K_{t1} y K_{b1} (de los pasos (6) y (7)) en la ecuación 3.13 tenemos que:

$$K_{med1} = \frac{K_{t1}}{R_{a1}B_{eq1} + K_{t1}K_{b1}} \quad (3.17)$$

El resultado numérico del paso (7) es $K_{t1} = K_{b1}$, entonces, haciendo $K_{t1} = K_{b1} = K_{m1}$ se tiene que:

$$\begin{aligned} K_{med1} &= \frac{K_{m1}}{R_{a1}B_{eq1} + K_{m1}^2} = 18.3; \quad \Rightarrow B_{eq1} = \frac{(\frac{K_{m1}}{K_{med1}} - K_{m1}^2)}{R_{a1}} \\ B_{eq1} &= \frac{(\frac{0.0437}{18.3} - 0.0437^2)}{3.43} = 1.3944 \times 10^{-4} \end{aligned} \quad (3.18)$$

y

$$T_{med1} = \frac{R_{a1}J_{eq1}}{R_{a1}B_{eq1} + K_{t1}K_{b1}} \quad (3.19)$$

Debido a que $K_{t1} = K_{b1} = K_{m1}$

$$\begin{aligned} T_{med1} &= \frac{R_{a1}J_{eq1}}{R_{a1}B_{eq1} + K_{m1}^2}; \quad \Rightarrow J_{eq1} = T_{med1} \frac{(R_{a1}B_{eq1} + K_{m1}^2)}{R_{a1}} \\ J_{eq1} &= 0.8 \frac{(3.43 \times 1.3944 \times 10^{-4} + 0.0437^2)}{3.43} = 5.5696 \times 10^{-4} \end{aligned} \quad (3.20)$$

- 9) Finalmente, se puede determinar la ganancia del amplificador K_{A1} , por medición directa de la tensión en los terminales de armadura del motor (salida del amplificador), y considerando la tensión de entrada a dicho amplificador, así:

$$K_{A1} = \frac{V_{a1 \text{ medido}}}{u_1} = \frac{19.7 \text{ V}}{1.4 \text{ V}} = 14.07 \quad (3.21)$$

El servomotor utilizado para mover el brazo del SRM es del mismo modelo del servomotor usado en la base del SRM (que se acaba de identificar); sin embargo, siguiendo los mismos procedimientos, se determinan ligeras variaciones en los parámetros B_{eq2} y J_{eq2} , debido a que los valores medidos de la fase experimental (del archivo de datos), se observa que $K_{med2} = 19.7 \text{ rad/s}$ (ligeramente mayor que K_{med1}) y $T_{med2} = 0.725 \text{ s}$ (ligeramente menor que T_{med1}). Finalizado el procedimiento de identificación, se presenta la siguiente tabla resumen de los parámetros identificados de los dos servomotores:

Tabla 3.1: Datos técnicos del servomotor GMX-7MC019B.

Velocidad (ω_1) (RPM)	Velocidad (ω_1) (rad/s)	Torque ($n_1 T_{m1}$) (Nm)	Corriente (A)
255	26.7035	sin carga	0.7
240	25.1327	0.5655	1.4
228	23.8761	1.1310	2.1
214	22.4100	1.6965	2.8
200	20.9440	2.2620	3.5
180	18.8496	3.3930	4.7
155	16.2316	4.5240	6.0
STALL	STALL	5.6550	7.0

Tabla 3.2: Parámetros del servomotor GMX-7MC019B identificados.

Parámetros	Símbolo	Valor	Unidades
Resistencia de armadura	R_{a1}, R_{a2}	3.43	Ω
Constante de torque del motor	K_{t1}, K_{t2}	0.0437	N m/A
Constante de f.c.e.m.	K_{b1}, K_{b2}	0.0437	V/rad/s
Fricción viscosa equivalente de M_1	B_{eq1}	1.3944×10^{-4}	N m/rad/s
Fricción viscosa equivalente de M_2	B_{eq2}	8.9966×10^{-5}	N m/rad/s
Momento de inercia equival. de M_1	J_{eq1}	5.5696×10^{-4}	N m s ²
Momento de inercia equival. de M_2	J_{eq2}	4.6887×10^{-4}	N m s ²
Ganancia del amplificador	K_{A1}, K_{A2}	14.07	—

CAPITULO IV

DISEÑO DEL SISTEMA DE CONTROL ÓPTIMO CONTINUO DEL SRM

4.1 Configuración del Sistema de Control Optimo No Lineal

La estrategia de control óptimo tiene por objetivo la determinación de una señal de control óptima u que minimice la función de costo (denominado también índice de performance o de rendimiento). Este capítulo trata sobre el problema de Control Lineal Cuadrático Gaussiano (LQG) donde el índice de performance es una función cuadrática de los estados y de las señales de control, y las perturbaciones estocásticas gaussianas son consideradas en el modelo no lineal del proceso. La solución del problema LQG conduce a una estructura de control lineal de realimentación de estados.

La configuración del sistema de control óptimo no lineal empleado en este capítulo es ilustrado en la figura 4.1. Tal configuración combina en el diseño la representación del modelo no lineal del proceso, un observador de estados no lineal (NSO) para estimar los estados del proceso, y un controlador de realimentación de estados proporcional integral (PISFC: Proportional-Integral State Feedback Controller) [13]. En definitiva, el objetivo del control óptimo es encontrar una función de fuerza de control u capaz de minimizar la diferencia entre el vector de salida del proceso y y el vector de trayectoria deseada r , a pesar de la presencia de los parámetros inciertos y de las perturbaciones estocásticas gaussianas.

La implementación en tiempo discreto del sistema de control óptimo no lineal mostrado en la figura 4.1 opera como sigue: Después de cada tiempo de muestreo, el NSO estima los estados del proceso \tilde{x} usando la data proporcionada por el vector de entrada u y el vector de salida y del proceso. El Controlador por realimentación de estados toma el vector de estados estimado \tilde{x} y la integral de la salida y para calcular el vector de control u .

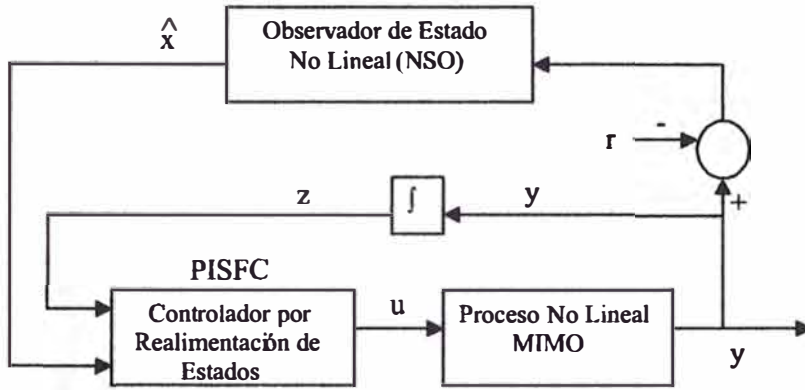


Figura 4.1: Configuración del Sistema de Control Optimo No Lineal.

4.2 Dinámica del Proceso No Lineal MIMO

La dinámica del proceso no lineal MIMO considerado puede ser representado por:

$$\begin{aligned} \dot{x} &= f(x) + G(x)u + v = f(x) + \sum_{i=1}^m g_i u_i + v ; & x(0^-) &= 0 \\ y &= h(x) + w \end{aligned} \quad (4.1)$$

siendo:

$$\begin{aligned} x &= \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} ; & u &= \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_m \end{bmatrix} ; & f &= \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{bmatrix} ; & y &= \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_p \end{bmatrix} \\ h &= \begin{bmatrix} h_1 \\ h_2 \\ \dots \\ h_p \end{bmatrix} ; & G &= \begin{bmatrix} g_1 & g_2 & \dots & g_m \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1m} \\ g_{21} & g_{22} & \dots & g_{2m} \\ \vdots & \vdots & \dots & \vdots \\ g_{n1} & g_{n2} & \dots & g_{nm} \end{bmatrix} \end{aligned}$$

donde x es el vector de estados, u es el vector de control (entrada del proceso), e y es el vector de salida del proceso. Los vectores v de orden $n \times 1$ y w de orden $p \times 1$ son las perturbaciones de entrada y de salida al proceso, respectivamente.

Denominando x_0 , u_0 e y_0 a los vectores de referencia o nominales correspondientes a x , u e y , respectivamente. Si la entrada al proceso se escoge exactamente igual a u_0 , la respuesta al proceso será x_0 . Entonces, x_0 satisface:

$$\begin{aligned} \dot{x}_0 &= f(x_0) + G(x_0)u_0 + v ; & x_0(0^-) &= 0 \\ y_0 &= h(x_0) + w \end{aligned} \quad (4.2)$$

Si la entrada al proceso u no es exacta, pro encierra a u_0 , la trayectoria resultante x es una ligera variación de x_0 . Para tal situación, la trayectoria actual puede ser formulada como:

$$x = x_0 + \delta x; \quad u = u_0 + \delta u; \quad y = y_0 + \delta y \quad (4.3)$$

donde el vector residual δx , δu , y δy , representan las desviaciones de los vectores de referencia correspondientes. Sustituyendo la ecuación (4.3) en (4.1) se obtiene:

$$\begin{aligned} \dot{x}_0 + \delta \dot{x} &= f(x_0 + \delta x) + G(x_0 + \delta x)[u_0 + \delta u] + v \\ y_0 + \delta y &= h(x_0 + \delta x) + w \end{aligned} \quad (4.4)$$

Como las desviaciones actuales se asumen muy pequeñas, el sistema dado por la ecuación (4.4) puede admitir una linealización aproximada constante para x cercano a x_0 . La expansión en series de Taylor de (4.4) alrededor de las trayectorias de referencia x_0 e y_0 conducen a:

$$\begin{aligned} \delta \dot{x} &\approx A\delta x + B\delta u + v; \quad \delta x_0(0^-) = 0 \\ \delta y &\approx C\delta x + w \end{aligned} \quad (4.5)$$

donde:

$$\begin{aligned} A &= \begin{bmatrix} \frac{\partial \dot{x}_1}{\partial x_1} & \frac{\partial \dot{x}_1}{\partial x_2} & \cdots & \frac{\partial \dot{x}_1}{\partial x_n} \\ \frac{\partial \dot{x}_2}{\partial x_1} & \frac{\partial \dot{x}_2}{\partial x_2} & \cdots & \frac{\partial \dot{x}_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \dot{x}_n}{\partial x_1} & \frac{\partial \dot{x}_n}{\partial x_2} & \cdots & \frac{\partial \dot{x}_n}{\partial x_n} \end{bmatrix}_{(x_0, u_0)}; \quad B = \begin{bmatrix} \frac{\partial \dot{x}_1}{\partial u_1} & \frac{\partial \dot{x}_1}{\partial u_2} & \cdots & \frac{\partial \dot{x}_1}{\partial u_m} \\ \frac{\partial \dot{x}_2}{\partial u_1} & \frac{\partial \dot{x}_2}{\partial u_2} & \cdots & \frac{\partial \dot{x}_2}{\partial u_m} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \dot{x}_n}{\partial u_1} & \frac{\partial \dot{x}_n}{\partial u_2} & \cdots & \frac{\partial \dot{x}_n}{\partial u_m} \end{bmatrix}_{(x_0, u_0)} \\ C &= \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial h_p}{\partial x_1} & \frac{\partial h_p}{\partial x_2} & \cdots & \frac{\partial h_p}{\partial x_n} \end{bmatrix}_{(x_0)} \end{aligned} \quad (4.6)$$

Observar que las matrices jacobianas A , B y C necesitan ser evaluadas a lo largo de la trayectoria de referencia x_0 . La representación linealizada de la ecuación 4.1 será formulada como:

$$\begin{aligned} \dot{x} &= Ax + Bu + v; \quad x(0^-) = 0 \\ y &= Cx + w \end{aligned} \quad (4.7)$$

Por consiguiente, las matrices A , B y C dadas por (4.6) para el SRM, pueden ser calculadas considerando el modelo no lineal del SRM en el espacio de estado, representadas por las ecuaciones (2.41) y (2.42), obteniéndose:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{R_{T1}B_{eq1}+N_{T1}}{R_{T1}(J_{eq1}+a_1+a_2)} & 0 \\ 0 & 0 & 0 & -\frac{R_{T2}B_{eq2}+N_{T2}}{R_{T2}(J_{eq2}+a_2)} \end{bmatrix};$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{R_{T1}(J_{eq1}+a_1+a_2)} & 0 \\ 0 & \frac{1}{R_{T2}(J_{eq2}+a_2)} \end{bmatrix}; \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (4.8)$$

donde: $a_1 = 2I_1$, $a_2 = I_2 + 2m_2L_{x2}L_2 - m_2L_2^2$.

4.3 Controlador por Realimentación de Estado MIMO

Es muy conocido el uso de un Controlador por Realimentación de Estados Proporcional (*Proportional State-Feedback Controller:PSFC*) tal como se muestra en la figura 4.2, en la que el controlador es una matriz ganancia de realimentación de estados D , con la siguiente ley de control:

$$u = -Dx \quad (4.9)$$

que minimiza la siguiente función de costo cuadrática:

$$I_{PSFC} = \int_0^{\infty} e^{2\alpha t} [x^T Q x(t) + u^T(t) R u(t)] dt \quad (4.10)$$

sujeto a la ecuación restringida

$$\dot{x} = Ax + Bu + v, \quad x(0^-) = 0 \quad (4.11)$$

donde A es una matriz de orden $n \times n$ y B es una matriz de orden $n \times m$ constantes, $Q = Q^T$ de orden $n \times n$ es una matriz de pesos semidefinida positiva (sdp), y $R = R^T$ de orden $m \times m$ es una matriz de pesos definida positiva (dp). Asimismo, $\alpha \geq 0$ es una constante de peso exponencial.

Una condición necesaria para aplicar control por realimentación de estados es que el proceso linealizado de la forma $\dot{x} = Ax + Bu$ sea *completamente controlable*, es decir el rango de la matriz de controlabilidad C sea igual al orden del

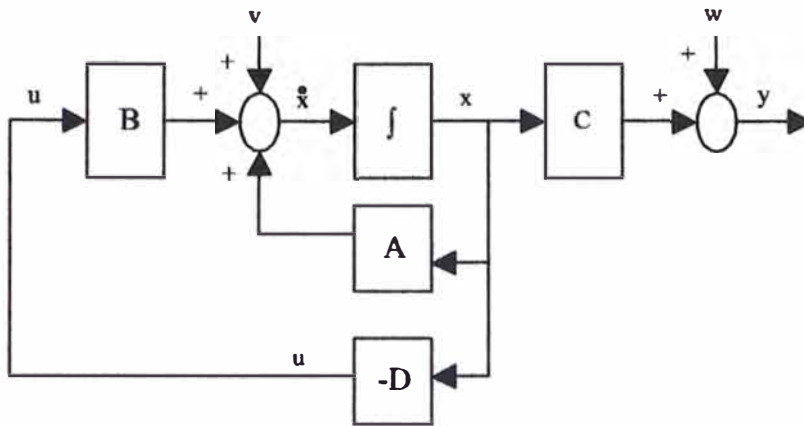


Figura 4.2: Sistema de Control por Realimentaciòn de Estados Proporcional.

sistema, o sea:

$$\text{rango } C = \text{rango} \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} = n \quad (4.12)$$

En nuestro caso se demuestra que el rango de la matriz de controlabilidad es:

$$\text{rango } C = \text{rango} \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix} = 4 \quad (4.13)$$

entonces, el proceso es completamente controlable, por lo que podemos proseguir con los pasos de diseño.

Es necesario mencionar que el esquema de la figura 4.2, representa al controlador por realimentaciòn de estados proporcional lineal, el cual nos permitirà determinar la matriz ganancia de realimentaciòn; sin embargo, en trabajos de implementaciòn en tiempo real el modelo de la planta o proceso se reemplaza por el proceso fìsico real, que para efectos de simulaciòn, se considerarà el modelo no lineal del proceso.

El siguiente paso es encontrar la matriz ganancia de realimentaciòn de estados D , dada por:

$$D = R^{-1}B^TK \quad (4.14)$$

Eligiendo:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad R = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \quad (4.15)$$

donde $K = K^T$ es una matriz definida positiva solución de la matriz asociada de Riccati:

$$K(A + \alpha I) + (A + \alpha I)^T K - KBR^{-1}B^T K + Q = 0 \quad (4.16)$$

Las matrices D y K , con $\alpha=5$, pueden determinarse fácilmente usando el siguiente comando de Matlab:

$$[D, K] = lqr((A + \alpha * eye(4)), B, Q, R)$$

obteniéndose:

➤ D

D =

```

3.5797    -0.0000    0.3730    0.0000
0.0000    3.5424    0.0000    0.3613

```

➤ K

K =

```

12.7145    -0.0000    0.2697    0.0000
-0.0000    12.4485   -0.0000    0.2387
 0.2697    -0.0000    0.0281    0.0000
 0.0000    0.2387    0.0000    0.0243

```

Si deseamos conocer la estabilidad del proceso, es decir conocer sus raíces, debemos determinar la ecuación característica del proceso, dada por:

$$|SI - A| = 0$$

Usando el siguiente comando en Matlab, se determina que las raíces son:

➤ eig(A)

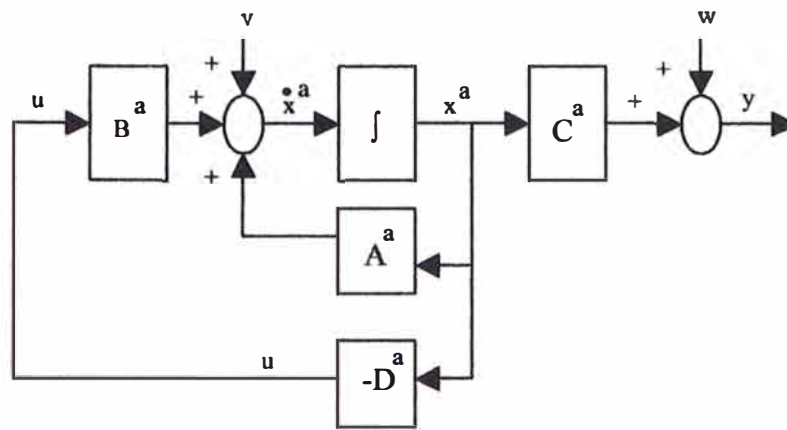
ans =

```

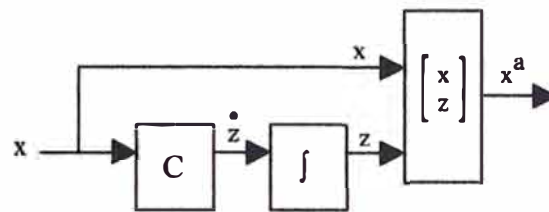
0
0
-7.6328
-8.5302

```

Para mejorar el rendimiento del PSFC, adicionamos un integrador en el controlador (ver [12] y [14]), según se muestra en la figura 4.3. Entonces, la



(a)



(b)

Figura 4.3: (a): Sistema de Control por Realimentación de Estados Proporcional Integral, (b): Esquema explicativo para la determinación de x^a .

descripción en el espacio de estados aumentada toma la forma:

$$\begin{aligned}\dot{x}^a &= A^a x^a + B^a u + v \\ y &= C^a x^a + w\end{aligned}\quad (4.17)$$

El vector x^a y las matrices A^a , B^a y C^a están dadas por:

$$x^a = \begin{bmatrix} x \\ z \end{bmatrix}; \quad A^a = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}; \quad B^a = \begin{bmatrix} B \\ 0 \end{bmatrix}; \quad C^a = [C \ 0] \quad (4.18)$$

con:

$$z = \int_0^t y dt; \quad z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$\Rightarrow x^a = \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ z_1 \\ z_2 \end{bmatrix} \quad (4.19)$$

Para el SRM, dichas matrices son:

$$A^a = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -7.6328 & 0 & 0 & 0 \\ 0 & 0 & 0 & -8.5302 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad B^a = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 132.7418 & 0 \\ 0 & 148.3875 \\ 0 & 0 \\ 0 & 0 \end{bmatrix};$$

$$C^a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Como el proceso linealizado invariante en el tiempo dado por:

$$\dot{x}^a = A^a x^a + B^a u + v; \quad x^a(0^-) = 0 \quad (4.20)$$

cumple con la condición de controlabilidad dada por (4.13), entonces, la matriz de ganancia aumentada del controlador de realimentación de estados proporcional integral, puede calcularse desde:

$$D^a = R^{-1}(B^a)^T K^a \quad (4.21)$$

donde $K^a = (K^a)^T$ es la matriz solución (única) definida positiva de la ecuación asociada de Riccati:

$$K^a(A^a + \alpha I) + (A^a + \alpha I)^T K^a - K^a B^a R^{-1}(B^a)^T K^a + Q^a = 0 \quad (4.22)$$

Entonces la ley de control de realimentación de estados aumentada:

$$u = -D^a x^a = - \begin{bmatrix} D_1 & D_2 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} \quad (4.23)$$

minimiza la siguiente función de costo cuadrática aumentada:

$$I_{SFC}^a = \int_0^\infty e^{2\alpha t} [(x^a)^T(t) Q^a x^a(t) + u^T(t) R u(t)] dt \quad (4.24)$$

sujeto a restricciones dada por la ecuación (4.20), donde $Q^a = (Q^a)^T$ de orden $(n+2) \times (n+2)$ es la matriz de pesos semidefinida positiva, $R = R^T$ de orden $m \times m$ es la matriz de pesos definida positiva, y $\alpha \geq 0$ es una constante de peso exponencial. El orden del vector v en (4.17) y (4.20) es $(n+2) \times 1$.

Idénticamente como en el caso del controlador por realimentación de estados proporcional, el esquema del sistema de control por realimentación de estados proporcional integral debe considerar el modelo no lineal del proceso, que representa más fielmente el proceso físico, tal como puede observarse en la figura 4.4.

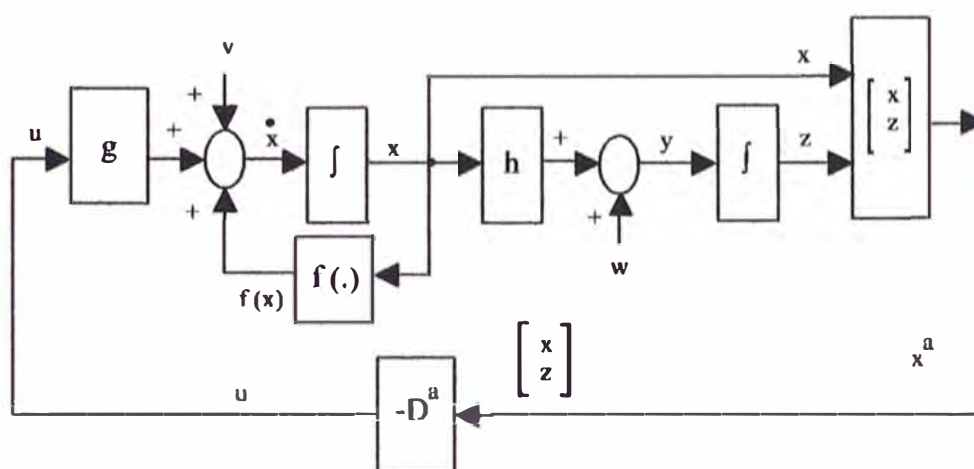


Figura 4.4: (a): Sistema de Control por Realimentación de Estados Proporcional Integral.

Determinemos las matrices ampliadas de ganancia D^a y de Riccati K^a para el SRM, para lo cual (usando Matlab) asumamos que:

```
R=10*eye(2); Qa=10*eye(6); alfa=5;
```

entonces, usando el comando:

$$[Da, Ka] = lqr((Aa + alfa * eye(6)), Ba, Qa, R)$$

se obtienen las siguientes matrices:

> Da

Da =

```
21.6226    0.0000    1.1318    0.0000   104.8427    0.0000
 0.0000    21.4643    0.0000    1.1119    0.0000   104.4503
```

→ Ka

Ka =

1.0e+004 *

0.0233	0.0000	0.0002	0.0000	0.1168	0.0000
0.0000	0.0230	0.0000	0.0001	0.0000	0.1151
0.0002	0.0000	0.0000	0.0000	0.0008	0.0000
0.0000	0.0001	0.0000	0.0000	0.0000	0.0007
0.1168	0.0000	0.0008	0.0000	1.0991	0.0000
0.0000	0.1151	0.0000	0.0007	0.0000	1.0909

Por el teorema de separación, los controladores por realimentación de estados pueden ser implementados empleando un estimador de estados no lineal en lugar de los estados actuales del proceso. Entonces, las leyes de control (4.9) y (4.23) pueden ser implementados como:

$$\begin{aligned}
 u &= -D\hat{x} \\
 u &= -D^a\hat{x}^a = -\begin{bmatrix} D_1 & D_2 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{z} \end{bmatrix}
 \end{aligned} \tag{4.25}$$

Para la obtención del vector de estados estimado \hat{x} se empleará un Observador de Estados No Lineal, y el vector z puede obtenerse por integración directa de la salida y , así:

$$\begin{aligned}
 z &= \int_0^t y dt = \int_0^t C^a x^a dt = \int_0^t \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ z_1 \\ z_2 \end{bmatrix} dt \\
 z &= \int_0^t \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} dt
 \end{aligned} \tag{4.26}$$

Si consideramos referencias, entonces el vector z puede representarse por:

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \int_0^t \begin{bmatrix} x_1(t) - r_1(t) \\ x_2(t) - r_2(t) \end{bmatrix} dt \tag{4.27}$$

lo que discretizando por adelanto se obtiene:

$$\begin{aligned} z_1(k+2) &= z_1(k+1) + T e_1(k+1) \\ z_2(k+2) &= z_2(k+1) + T e_2(k+1) \end{aligned} \quad (4.28)$$

donde:

$$\begin{aligned} e_1(k+1) &= x_1(k+1) - r_1(k+1) \\ e_2(k+1) &= x_2(k+1) - r_2(k+1) \end{aligned}$$

4.4 Márgenes de Estabilidad de Realimentación de Estados

Los márgenes de estabilidad están formuladas con respecto a la representación linealizada de (4.1) dada por (4.7). El proceso descrito por (4.1) con realimentación de estados no lineal de lazo cerrado (4.9) es estable con ganancia finita, si

$$P\{(A - q) - (B - g)D\} + \frac{1}{2}S > 0 \quad (4.29)$$

o

$$P\{(A - \nabla q) - (B - \nabla g)D\} + S > 0 \quad (4.30)$$

donde la función $q(x)$ puede obtenerse de la relación

$$f(x) = q(x)x \quad (4.31)$$

También, la matriz $P = P^T$ de orden $n \times n$ y la matriz dependiente de x $S = S^T$ son simétricas, definidas positivas, soluciones de la siguiente ecuación de Lyapunov:

$$P(A - BD) + (A - BD)^T P + S = 0 \quad (4.32)$$

Para nuestro caso, asumamos que:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.33)$$

entonces, de la ecuación (4.32) puede obtenerse la matriz S . Usando Matlab, tendremos que:

$$S = -(P*(A-B*D) + (A-B*D)'*P)$$

cuyo resultado es:

S =

$$\begin{array}{cccc} 0 & 0 & 474.1803 & 0.0000 \\ 0 & 0 & -0.0000 & 524.6468 \\ 474.1803 & -0.0000 & 114.2944 & 0.0000 \\ 0.0000 & 524.6468 & 0.0000 & 124.2916 \end{array}$$

verificándose que $S = S^T$.

4.5 Márgenes de Estabilidad y Robustez para el LQSFC

El proceso representado por (4.1) y la ley de control dada por (4.9), conforman un lazo cerrado estable con ganancia finita si cualquiera de las dos ecuaciones

$$K[\alpha I + (A - g) - (B - g)D] + \frac{1}{2}[Q + KBR^{-1}B^TK] > 0 \quad (4.34)$$

o

$$K[\alpha I + (A - \nabla q) - (B - \nabla g)D] + \frac{1}{2}[Q + KBR^{-1}B^TK] > 0 \quad (4.35)$$

contiene a todas las x . Las matrices ganancia D y K se calculan de (4.14) y (4.16), tarea que ya ha sido cumplida.

Las condiciones (4.34) y (4.35) pueden demostrarse fácilmente. Sustituyendo

$$S = Q + KBR^{-1}B^TK + 2\alpha K; \quad K = P \quad (4.36)$$

en (4.34), (4.35), y (4.16), se pueden obtener los márgenes de estabilidad (4.29), (4.30) y la ecuación de Lyapunov (4.32). Considerando los resultados de D y K obtenidos en la sección 4.3, se determina la matriz S :

```
> S=Q+K*B*inv(R)*B'*K+2*alfa*K
```

S =

$$\begin{array}{cccc} 256.2897 & -0.0000 & 16.0496 & 0.0000 \\ -0.0000 & 250.9709 & -0.0000 & 15.1867 \\ 16.0496 & -0.0000 & 2.6724 & 0.0000 \\ 0.0000 & 15.1867 & 0.0000 & 2.5490 \end{array}$$

Reemplazando este resultado en la ecuación de Lyapunov, con $P=K$, se obtiene:

> P=K

P =

12.7145	-0.0000	0.2697	0.0000
-0.0000	12.4485	-0.0000	0.2387
0.2697	-0.0000	0.0281	0.0000
0.0000	0.2387	0.0000	0.0243

> P*(A-B*G)+(A-B*G)'*P+S

ans =

1.0e-011 *

-0.5798	-0.0001	-0.0103	-0.0001
-0.0001	-0.4746	-0.0000	-0.0023
-0.0103	-0.0000	-0.0002	-0.0000
-0.0001	-0.0023	-0.0000	0.0001

Se podrá observar que el resultado de la ecuación de Lyapunov es demasiado pequeño (del orden de 10^{-11}), que es prácticamente nulo, cumpliéndose así con las condiciones de estabilidad.

Es importante mencionar que con la elección de la matriz P , en la sección anterior, se determina un valor de S distinto al obtenido en la presente sección; sin embargo se cumple con las condiciones de estabilidad. Las ecuaciones de Lyapunov y de Riccati pueden ser usadas considerando las matrices aumentadas, con sólo reemplazar en las ecuaciones correspondientes las matrices P , A , B , D , K , S , q , g , y Q por las matrices P^a , A^a , B^a , D^a , K^a , S^a , q^a , g^a , y Q^a , cumpliéndose así las condiciones de estabilidad del sistema aumentado.

4.6 Observador de Estado No Lineal MIMO

Consideremos la estructura del sistema de control con estimación de estados no lineal, mostrada en la figura 4.5.

La ventaja de incluir un observador de estados, radica en la estimación en línea (on-line) de los estados del proceso, que permiten un eficiente control de las variables del proceso en tiempo real [13]. De la estructura mostrada, puede observarse que la trayectoria de referencia considerada es $r(t) = 0$. Las ecuaciones que describen al Observador de Estado No Lineal, son:

$$\dot{\hat{x}} = f + gu - E(\hat{y} - y); \quad \hat{x}(0^-) = 0$$

$$\hat{y} = h\hat{x} \quad (4.37)$$

donde \hat{x} e \hat{y} son los vectores estimados de x e y , respectivamente, y E es una

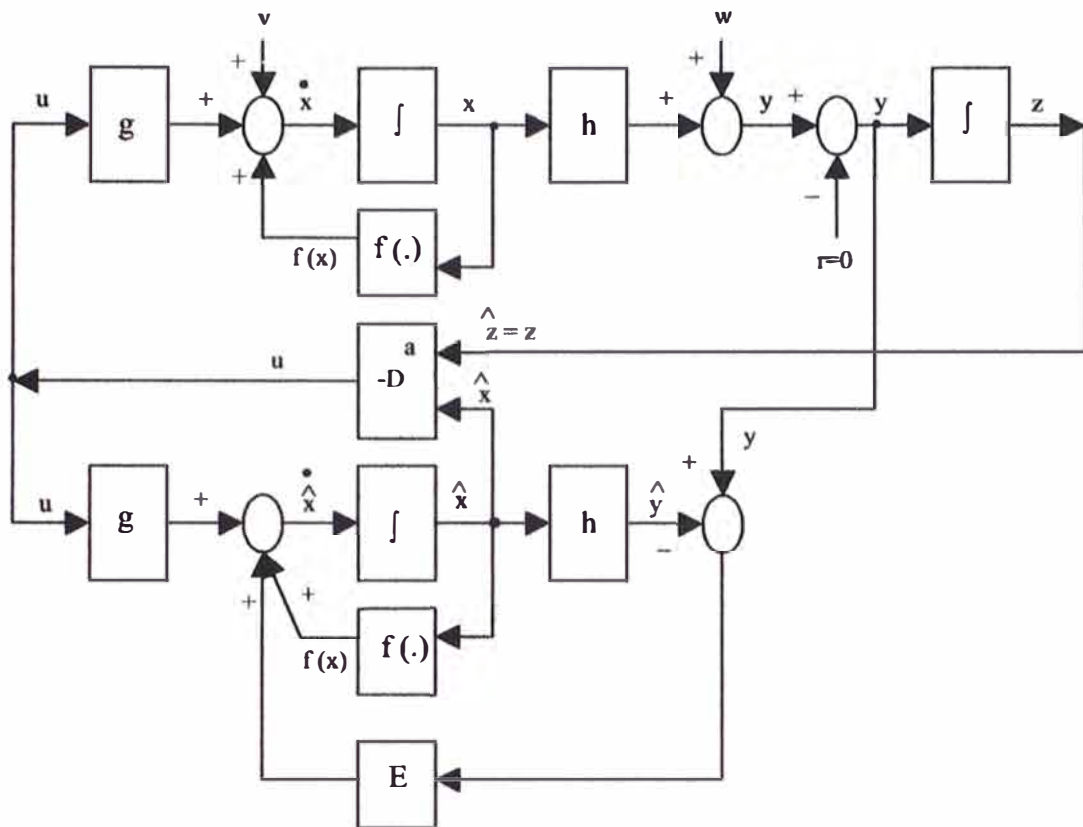


Figura 4.5: (a): Estructura del sistema de Control y Estimación de Estados.

matriz de ganancia residual de $(n \times p)$ a determinarse. Tan solo por simplicidad, se ha omitido la dependencia de f , g , y h en la ecuación (4.37).

Un proceso lineal dado por la ecuación 4.7 con $v = w = 0$ es observable si la matriz de observabilidad correspondiente \mathcal{O} es de rango completo, es decir:

$$\text{rango } \mathcal{O} = \text{rango} \begin{bmatrix} C \\ CM \\ \vdots \\ CM^{n-1} \end{bmatrix} = n \quad (4.38)$$

Considerando que el modelo linealizado dado por la ecuación 4.7 es observable, la ganancia residual E del observador de estado no lineal descrito por 4.37 se

encuentra usando la ganancia del filtro de Kalman:

$$E = \sum C^T W^{-1} \quad (4.39)$$

donde $\sum = \sum^T$ es una matriz definida positiva, solución (única) de la ecuación de Riccati:

$$0 = \sum(A + \alpha I)^T + (A + \alpha I)\sum - \sum C^T W^{-1} C \sum + V \quad (4.40)$$

Las perturbaciones v y w en la ecuación (4.1) pueden ser ruido blanco, ruido gaussiano y tienen valor medio cero. Los ruidos blanco y gaussiano tienen la propiedad de no ser correlacionados. El valor medio cero implica que toda la información estática acerca del ruido es sumariada en la covarianza del ruido. En términos matemáticos

$$\begin{aligned} E_o[v(t)v^T(\tau)] &= V\delta(t - \tau); & E_o[v] &= 0 \\ E_o[w(t)w^T(\tau)] &= W\delta(t - \tau); & E_o[w] &= 0 \end{aligned} \quad (4.41)$$

donde $E_o[.]$ es la operación matemática expectante, $\delta(t)$ es la función delta de Kronecker, y V y W son matrices de covarianza del ruido del proceso y del ruido del sensor definidas positivas, respectivamente. V es de orden $n \times n$ (en nuestro caso de 4×4), W es de orden $m \times m$ (en nuestro caso de 2×2). Para el sistema SRM, dichas matrices de covarianza han sido elegidas como:

$$V = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}; \quad W = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}; \quad G1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.42)$$

El paso siguiente es resolver la ecuación de Riccati para obtener \sum , y luego reemplazar dicho resultado en la ecuación (4.39) para hallar la ganancia del filtro de Kalman E . Toda esta tarea puede resolverse rápidamente con el siguiente comando en Matlab:

```
[E,sum,Eo]=lqe((A+alfa*eye(4)),G1,C,V,W)
```

que produce los siguientes resultados:

➤ E

E =

```

10.1038    0.0000
 0.0000    10.1022
 0.0245    0.0000
 0.0000    0.0164

```

➤ sum

sum =

```

1.0104    0.0000    0.0025    0.0000
0.0000    1.0102    0.0000    0.0016
0.0025    0.0000    0.0190    0.0000
0.0000    0.0016    0.0000    0.0142

```

Puede apreciarse que la ganancia del filtro de Kalman E es de la forma:

$$E = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \\ E_{31} & E_{32} \\ E_{41} & E_{42} \end{bmatrix} \quad (4.43)$$

La ley de control para el SRM, dada por la ecuación (4.25), usando el resultado de la matriz ganancia del controlador D^a , es de la forma:

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = -D^a \hat{x}^a = - \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & d_{26} \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{z}_1 \\ \hat{z}_2 \end{bmatrix} \quad (4.44)$$

con $\hat{z} = z$. A continuación se reescribe la ley de control en forma individual (para u_1 y u_2), así:

$$\begin{aligned} u_1 &= -d_{11}\hat{x}_1 - d_{12}\hat{x}_2 - d_{13}\hat{x}_3 - d_{14}\hat{x}_4 - d_{15}\hat{z}_1 - d_{16}\hat{z}_2 \\ u_2 &= -d_{21}\hat{x}_1 - d_{22}\hat{x}_2 - d_{23}\hat{x}_3 - d_{24}\hat{x}_4 - d_{25}\hat{z}_1 - d_{26}\hat{z}_2 \end{aligned} \quad (4.45)$$

Considerando la ecuación del proceso no lineal (2.41), la ecuación del observador

de estados no lineal en tiempo continuo (4.37), es:

$$\begin{aligned}
 \dot{\hat{x}}_1 &= \hat{x}_3 - E_{11}(\hat{x}_1 - x_1) - E_{12}(\hat{x}_2 - x_2) \\
 \dot{\hat{x}}_2 &= \hat{x}_4 - E_{21}(\hat{x}_1 - x_1) - E_{22}(\hat{x}_2 - x_2) \\
 \dot{\hat{x}}_3 &= -p_{33} \times g_3 \hat{x}_3 + g_3 u_1 - E_{31}(\hat{x}_1 - x_1) - E_{32}(\hat{x}_2 - x_2) \\
 \dot{\hat{x}}_4 &= -q_{42} \cos \hat{x}_2 - q_{43} \hat{x}_3 - q_{44} \hat{x}_4 + g_4 u_2 - E_{41}(\hat{x}_1 - x_1) - E_{42}(\hat{x}_2 - x_2)
 \end{aligned} \tag{4.46}$$

donde:

$$\begin{aligned}
 p_{33} &= R_{T1}(B_{eq1} - 2H_{22}x_4 \text{sen } x_2 \cos x_2) + N_{T1}; \\
 g_3 &= \frac{1}{R_{T1}(J_{eq1} + 2I_1 + H_{22} \cos^2 x_2)}, \quad q_{42} = \frac{R_{T2} a_3}{R_{T2}(J_{eq2} + H_{22})}; \\
 a_3 &= m_2 g L_{x_2}; \quad q_{43} = \frac{R_{T2} H_{22} x_3 \text{sen } x_2 \cos x_2}{R_{T2}(J_{eq2} + H_{22})}; \\
 q_{44} &= \frac{R_{T2} B_{eq2} + N_{T2}}{R_{T2}(J_{eq2} + H_{22})}; \quad g_4 = \frac{1}{R_{T2}(J_{eq2} + H_{22})} \\
 N_{T1} &= \frac{n_1 K_{t1}}{K_{A1}}; \quad N_{T2} = \frac{n_2 K_{t2}}{K_{A2}}
 \end{aligned}$$

4.7 Simulación del Sistema de Control Optimo con Observador No Lineal

Si bien es cierto que el observador no lineal considerado es de tiempo continuo; sin embargo para ejecutarlo en un computador digital, es necesario discretizar directamente la ecuación (4.46), tal como se presenta en la siguiente ecuación:

$$\begin{aligned}
 \hat{x}_1(k+1) &= \hat{x}_1(k) + T[\hat{x}_3(k) - E_{11}(\hat{x}_1(k) - x_1(k)) - E_{12}(\hat{x}_2(k) - x_2(k))] \\
 \hat{x}_2(k+1) &= \hat{x}_2(k) + T[\hat{x}_4(k) - E_{21}(\hat{x}_1(k) - x_1(k)) - E_{22}(\hat{x}_2(k) - x_2(k))] \\
 \hat{x}_3(k+1) &= \hat{x}_3(k) + T[-p_{33}(k) \times g_3(k) \hat{x}_3(k) + g_3(k) u_1(k) \dots \\
 &\quad - E_{31}(\hat{x}_1(k) - x_1(k)) - E_{32}(\hat{x}_2(k) - x_2(k))] \\
 \hat{x}_4(k+1) &= \hat{x}_4(k) + T[-q_{42}(k) \cos \hat{x}_2(k) - q_{43}(k) \hat{x}_3(k) - q_{44}(k) \hat{x}_4(k) \dots \\
 &\quad + g_4 u_2(k) - E_{41}(\hat{x}_1(k) - x_1(k)) - E_{42}(\hat{x}_2(k) - x_2(k))]
 \end{aligned} \tag{4.47}$$

donde:

$$p_{33}(k) = R_{T1}(B_{eq1} - 2H_{22} \hat{x}_4 \text{sen } \hat{x}_2 \cos \hat{x}_2) + N_{T1};$$

$$g_3(k) = \frac{1}{R_{T1}(J_{eq1} + 2I_1 + H_{22}\cos^2 \hat{x}_2)}; \quad q_{42}(k) = q_{42};$$

$$q_{43}(k) = \frac{R_{T2}H_{22}\hat{x}_3 \text{sen } \hat{x}_2 \cos \hat{x}_2}{R_{T2}(J_{eq2} + H_{22})}; \quad q_{44}(k) = q_{44}; \quad g_4(k) = g_4$$

La estructura de la figura 4.5 puede usarse para resolver el problema de seguimiento. En tal caso, la trayectoria de referencia $r(t)$ es una función de tiempo continuo t . Esta consideración es la que se ha usado en esta tesis, cuyo programa, que comprende el controlador óptimo y el observador no lineal, escrito en código Matlab, denominado *srm4opt.m* se presenta en el Apéndice A, y es la que se lista a continuación en forma parcial.

```
% MODELO DISCRETO DEL PROCESO LINEALIZADO CONTINUO
T=0.0014; % Fs=700 Hz
[Ad,Bd,Cd,Dd]=c2dm(A,B,C,D,T,'zoh');
% CALCULO DE LA CONTROLABILIDAD EN TIEMPO DISCRETO
Md=[Bd Ad*Bd Ad^2*Bd Ad^3*Bd];
rangoMd=rank(Md); %RANGO DEL SISTEMA = 4
% CALCULO DE LA OBSERVABILIDAD EN TIEMPO DISCRETO
Nd=[Cd' Ad'*Cd' Ad'^2*Cd' Ad'^3*Cd'];
rangoNd=rank(Nd); %RANGO DEL SISTEMA = 4
% CONTROLADOR POR REALIMENTACION DE ESTADOS NO LINEAL MIMO (CONTINUO)

% MATRICES DE PONDERACION
Q=1*eye(4);
R=10*eye(2);
Qa=10*eye(6); V=0.1*eye(4); W=0.1*eye(2); G1=eye(4);
alfa=5; % sin considerar restricciones de la señal de control
%alfa=2; % considerando restricciones de la señal de control
[D,K]=lqr((A+alfa*eye(4)),B,Q,R);
% MEJORA DEL RENDIMIENTO DEL CONTROLADOR ADICIONANDO UN INTEGRADOR
% MATRICES AUMENTADAS
Aa=[A zeros(4,2); C zeros(2,2)]; % matriz (n+2) x (n+2): 6 x 6
Ba=[B; zeros(2,2)]; % matriz (n+2) x m : 6 x 2
Ca=[C zeros(2,2)];
Da=zeros(2);
% CALCULO DE LAS GANACIAS Ga y Ka
[Da,Ka]=lqr((Aa+alfa*eye(6)),Ba,Qa,R);
% OBSERVADOR DE ESTADOS NO LINEAL MIMO
% xep=q*xe+g*u-E*(ye-y) ; xe(0)=0;
% ye=h*xe;
% matriz E es de orden n x p (4x2)
[E,sum,Eo]=lqe((A+alfa*eye(4)),G1,C,V,W);

% CONDICIONES INICIALES
u1(1)=0.0; u2(1)=0.1;
u1(2)=0.0; u2(2)=0.1;
```

```

u1(3)=0.0; u2(3)=0.1;
x1(1)=0.5; x2(1)=-0.5; x3(1)=0.0; x4(1)=0.0;
x1(2)=0.5; x2(2)=-0.5; x3(2)=0.0; x4(2)=0.0;
x1(3)=0.5; x2(3)=-0.5; x3(3)=0.0; x4(3)=0.0;
xe1(1)=0.0; xe2(1)=0.0; xe3(1)=0.0; xe4(1)=0.0;
xe1(2)=0.0; xe2(2)=0.0; xe3(2)=0.0; xe4(2)=0.0;
xe1(3)=0.0; xe2(3)=0.0; xe3(3)=0.0; xe4(3)=0.0;
z1(1)=0.0; z2(1)=0.0;
z1(2)=0.0; z2(2)=0.0;
z1(3)=0.0; z2(3)=0.0;
MM=4000; % número de muestras

% BUCLE DE CONTROL
for k=2:MM;
% VALORES DE PARAMETROS DEL PROCESO NO LINEAL DISCRETIZADO
p33(k)=Rt1*(b1-2*a2*x4(k)*sin(x2(k))*cos(x2(k)))+Nt1;
pa(k)=Rt1*(J1+a1+a2*cos(x2(k))*cos(x2(k)));
pb(k)=Rt2*(J2+a2);
q42(k)=(Rt2*a3)/(Rt2*(J2+a2));
q43(k)=(Rt2*a2*x3(k)*sin(x2(k))*cos(x2(k)))/(Rt2*(J2+a2));
q44(k)=(Rt2*b2+Nt2)/(Rt2*(J2+a2));
g3(k)=1/pa(k); g4(k)=1/pb(k);
% PROCESO NO LINEAL DISCRETIZADO
x1(k+1)=x1(k)+T*x3(k);
x2(k+1)=x2(k)+T*x4(k);
x3(k+1)=x3(k)+T*[-p33(k)*g3(k)*x3(k)+g3(k)*u1(k)];
x4(k+1)=x4(k)+T*[-q42(k)*cos(x2(k))-q43(k)*x3(k)-q44(k)*x4(k)+ ...
g4(k)*u2(k)];

r1(k+1)=sin(0.0014*k)+ 0.00014*k; r2(k+1)=cos(0.0014*k);
e1(k+1)=-r1(k+1)+x1(k+1); e2(k+1)=-r2(k+1)+x2(k+1);

% OBSERVADOR NO LINEAL DISCRETIZADO DE 4TO. ORDEN
xe1(k+2)=xe1(k+1)+T*[xe3(k+1)-E(1,1)*(xe1(k+1)-e1(k+1))- ...
E(1,2)*(xe2(k+1)-e2(k+1))];
xe2(k+2)=xe2(k+1)+T*[xe4(k+1)-E(2,1)*(xe1(k+1)-e1(k+1))- ...
E(2,2)*(xe2(k+1)-e2(k+1))];
xe3(k+2)=xe3(k+1)+T*[-p33(k)*g3(k)*xe3(k+1)+g3(k)*u1(k)- ...
E(3,1)*(xe1(k+1)-e1(k+1))-E(3,2)*(xe2(k+1)-e2(k+1))];
xe4(k+2)=xe4(k+1)+T*[-q42(k)*cos(xe2(k+1))-q43(k)*xe3(k+1) - ...
q44(k)*xe4(k+1)+g4(k)*u2(k)-E(4,1)*(xe1(k+1)-e1(k+1))- ...
E(4,2)*(xe2(k+1)-e2(k+1))];

% LEY DE CONTROL DISCRETIZADO u1(K) Y u2(k)
z1(k+2)=z1(k+1)+T*e1(k+1); z2(k+2)=z2(k+1)+T*e2(k+1);
u1(k+1)=-Da(1,1)*xe1(k+2)-Da(1,2)*xe2(k+2)-Da(1,3)*xe3(k+2)- ...
Da(1,4)*xe4(k+2)-Da(1,5)*z1(k+2)-Da(1,6)*z2(k+2);
u2(k+1)=-Da(2,1)*xe1(k+2)-Da(2,2)*xe2(k+2)-Da(2,3)*xe3(k+2)- ...
Da(2,4)*xe4(k+2)-Da(2,5)*z1(k+2)-Da(2,6)*z2(k+2);

```

end

que produce el resultado gráfico que se muestra en las figuras 4.6 y 4.7. Del

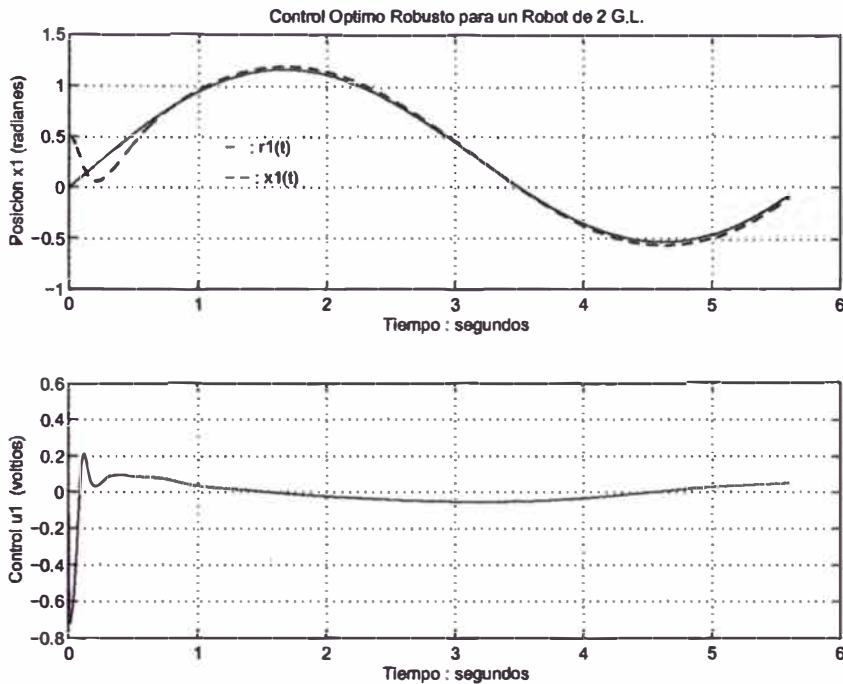


Figura 4.6: Posición de la base x_1 y la señal de control correspondiente u_1 .

extracto del programa presentado, las señales de referencia usadas son $r_1(t) = \sin t + 0.1t$, $r_2(t) = \cos t$, y el período de muestreo considerado es $T = 0.0014$ segundos.

Del resultado gráfico, se puede observar de la figura 4.6 que la señal de control $u_1(t)$ necesario para que la posición angular $x_1(t)$ de la base del SRM siga a la referencia $r_1(t)$, alcanza un valor pico transitorio de aproximadamente 0.2 V , luego decrece hasta $\pm 0.05\text{ V}$ en tiempo estacionario. Idénticamente, de la figura 4.7 se puede observar que la tensión de control $u_2(t)$ necesario para llevar la posición angular del brazo del SRM a la posición deseada $r_2(t)$, alcanza un valor pico transitorio de aproximadamente 2.8 V , decreciendo a valores entre $0\text{ a }0.5\text{ V}$ en tiempo estacionario. Es necesario anotar, que las señales de control $u_1(t)$ y $u_2(t)$ generadas por el controlador son señales sin restricciones. Para la implementación en tiempo real, será necesario restringir dichas señales para evitar saturaciones indeseadas en el amplificador.

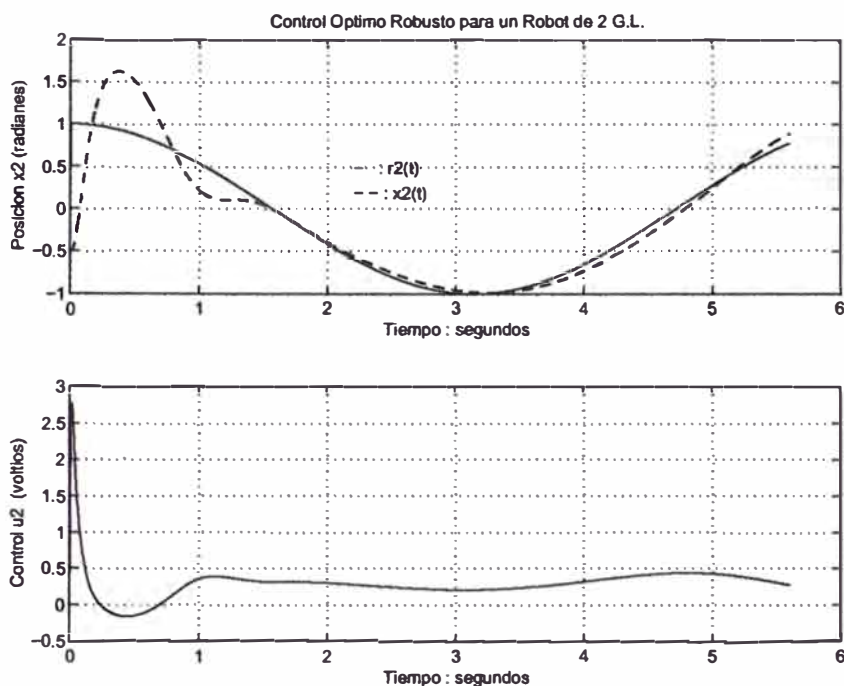


Figura 4.7: Posición del brazo x_2 y la señal de control correspondiente u_2 .

En forma idéntica a los resultados mostrados de respuestas a trayectorias, en las figuras 4.8 y 4.9 se presentan las respuestas a referencias fijas. En este caso, las referencias son: $r_1(t) = 1$, $r_2(t) = 1.5$, considerando las siguientes matrices de ponderación: $R = 100 * eye(2)$, $Qa = 100 * eye(6)$, $V = 0.1 * eye(4)$, $W = 0.1 * eye(2)$, $G1 = eye(4)$. De la figura 4.8 se observa que la posición angular de la base x_1 presenta un sobrepico de aproximadamente 12 %, alcanzando a la referencia en aproximadamente 0.8 segundos, con error estacionario nulo. La señal de control correspondiente u_1 alcanza un valor máximo transitorio de aproximadamente 0.75 V, atenuándose y llegando a cero voltios en aproximadamente 0.5 segundos (tiempo estable). Finalmente, de la figura 4.9, se aprecia que la posición angular del brazo x_2 presenta un sobrepico de aproximadamente 67 %, amortiguándose y tendiendo hacia la referencia en aproximadamente 4 segundos, a partir del cual se mantiene en estado estable. La señal de control correspondiente que hace posible tal respuesta es u_2 , que alcanza un valor máximo transitorio de aproximadamente 3.6 V, luego tiende hacia cero en aproximadamente 4 segundos (tiempo estable).

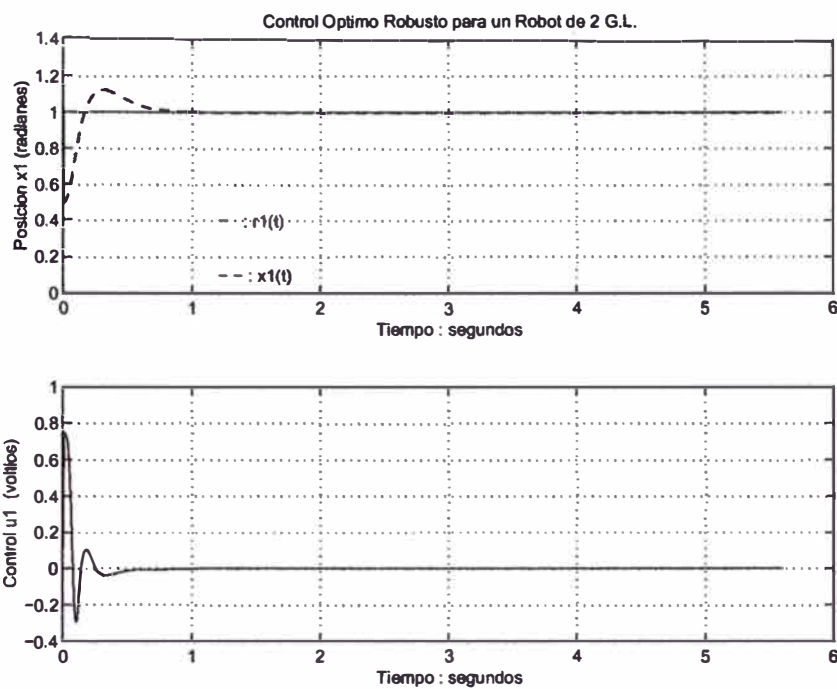


Figura 4.8: Posición de la base x_1 y la señal de control correspondiente u_1 .

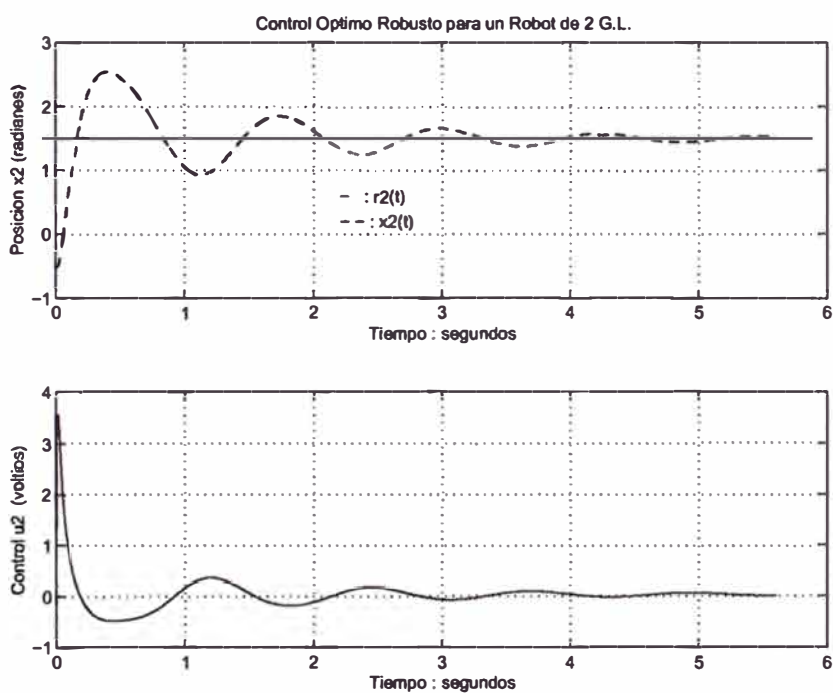


Figura 4.9: Posición del brazo x_2 y la señal de control correspondiente u_2 .

CAPITULO V

DISEÑO DEL SISTEMA DE CONTROL ÓPTIMO DISCRETO DEL SRM

El objetivo del Control Óptimo es el de determinar una ley de control óptima u que minimice la función de costo J . La formulación de la función de costo o índice de rendimiento apropiado es muy importante, porque determina la naturaleza del control óptimo. En aplicaciones prácticas es importante considerar restricciones en la señal de control, para hacerlo físicamente realizable. Asimismo, es importante indicar que un sistema de control que es óptimo bajo un índice de desempeño es, en general, no óptimo bajo otra función de costo o índice de desempeño. Para aplicar el diseño de control óptimo, el proceso debe ser completamente controlable y completamente observable (ver [16] y [12]). Si el proceso no es completamente observable, entonces se puede utilizar un observador óptimo que estime dichos estados.

El problema de optimización de un sistema de control se puede formular si se cuenta con la siguiente información:

- a) Ecuaciones del sistema
- b) Vectores de control permitidos
- c) Restricciones en el problema
- d) Función de costo o índice de desempeño
- e) Parámetros del sistema

La solución de un problema de control óptimo consiste en determinar el vector de control óptimo $u(k)$ que depende de:

- a) La naturaleza de la función de costo

- b) La naturaleza de las restricciones
- c) El estado inicial o la salida inicial
- d) El estado deseado o salida deseada
- e) Parámetros del sistema

En general, una solución analítica es muy complicada, por lo que debe usarse la computadora. En tal sentido podemos decir que el diseño de sistemas de control óptimo es fundamentalmente un problema computacional.

5.1 El problema del Control Óptimo Cuadrático No Estacionario

El problema de control óptimo cuadrático discreto no estacionario o de tiempo finito (ver [16] y [12]), se puede plantear de la siguiente manera: Dado un sistema de control lineal en tiempo discreto

$$x(k+1) = Gx(k) + Hu(k), \quad x(0) = c \quad (5.1)$$

de estado completamente controlable y donde

- $x(k)$ es un vector de estado de orden n
- $u(k)$ es un vector de control de orden r
- G es una matriz de dimensión $n \times n$
- H es una matriz de dimensión $n \times r$

encontrar la señal de control óptima $u(k)$, para $k = 0, 1, \dots, N-1$, que minimice la siguiente función de costo:

$$J = \frac{1}{2}x^T(N)Sx(N) + \frac{1}{2} \sum_{k=0}^{N-1} [x^T(k)Qx(k) + u^T(k)Ru(k)] \quad (5.2)$$

donde

- Q es una matriz Hermítica (o matriz real simétrica) definida positiva o semidefinida positiva de dimensión $n \times n$, que pondera el vector de estado $x(k)$

- R es una matriz Hermítica o matriz real simétrica) definida positiva de dimensión $r \times r$, que pondera el vector de control $u(k)$
- S es una matriz Hermítica o matriz real simétrica) definida positiva o semidefinida positiva de dimensión $n \times n$, que pondera el estado final del vector $x(N)$

La ley de control óptima para un sistema de regulación viene dada por:

$$u(k) = -K(k)x(k) \quad (5.3)$$

donde $K(k)$ es la matriz de ganancia del controlador o matriz de realimentación de estados, de dimensión $r \times n$, dada por la siguiente ecuación:

$$K(k) = [R + H^T P(k+1)H]^{-1} H^T P(k+1)G \quad (5.4)$$

donde $P(k+1)$ es una matriz Hermítica definida positiva de dimensión $n \times n$, que es la solución de la siguiente ecuación de Riccati:

$$P(k) = Q + G^T P(k+1)G - G^T P(k+1)H [R + H^T P(k+1)H]^{-1} H^T P(k+1)G \quad (5.5)$$

La matriz $P(k)$ de la ecuación 5.5 puede determinarse recursivamente hacia atrás, es decir, partiendo de la condición terminal $P(N) = S$, o sea con $k = N$, siguiendo con $P(N-1), P(N-2), \dots, P(0)$. Calculados los valores de la matriz $P(k)$, se deben calcular también en forma recursiva los valores de la matriz ganancia del controlador $K(k)$, con $k = N-1, N-2, \dots, 0$. Luego debe calcularse la solución de la ecuación de estados $x(k+1)$, es decir $x(k)$, con $k = 0, 1, \dots, N-1$. Finalmente, se calcula la señal de control $u(k)$ dada por la ecuación (5.3). Por otro lado, el valor mínimo de la función de costo viene dado por:

$$J_{min} = \frac{1}{2} x^T(0) P(0) x(0) \quad (5.6)$$

5.2 El problema del Control Óptimo Cuadrático Estacionario

En el sistema de control óptimo cuadrático discreto no estacionario, se vio que la dinámica del sistema evoluciona hasta un tiempo finito, haciendo que la matriz de ganancia o matriz de ganancia de realimentación $K(k)$ se convierte en una matriz variante en el tiempo; mientras que en el control óptimo cuadrático estacionario, la dinámica de control evoluciona hasta un tiempo N infinito, por

consiguiente, la ganancia $K(k)$ se convierte en una matriz constante K . Para N infinito, el termino $\frac{1}{2}x^T(N)Sx(N)$ de la ecuación (5.2) desaparece debido a que $x(\infty)$ se hace cero. Por consiguiente, la función de costo para el control óptimo estacionario de un sistema regulador es de la forma:

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [x^T(k)Qx(k) + u^T(k)Ru(k)] \quad (5.7)$$

La ecuación de Riccati en estado estacionario es:

$$P = Q + G^T P G - G^T P H [R + H^T P H]^{-1} H^T P G \quad (5.8)$$

Una forma de resolver la ecuación de Riccati en estado estacionario es usar la ecuación de Riccati en estado no estacionario, dada en la ecuación (5.5) pero con la inversión en la dirección del tiempo, como se presenta a continuación:

$$P(k+1) = Q + G^T P(k)G - G^T P(k)H [R + H^T P(k)H]^{-1} H^T P(k)G \quad (5.9)$$

y comenzar la solución con $P(0) = 0$, usando este valor para calcular $P(1)$, y así sucesivamente hasta llegar a un tiempo discreto k para el cual $P(k) = P(k+1) = P(k+2) = \dots$. Obtenido P , usar esta matriz para calcular la matriz ganancia del controlador K , dada por:

$$K = [R + H^T P H]^{-1} H^T P G \quad (5.10)$$

La ley de control óptimo en estado estacionario está dado por:

$$u(k) = -Kx(k) \quad (5.11)$$

y la función de costo mínima es:

$$J_{min} = \frac{1}{2} x^T(0) P x(0) \quad (5.12)$$

El diagrama de bloques del sistema del control óptimo de estado estacionario tipo regulador se representa en la figura 5.1, en donde se asume que todos los estados se encuentran disponibles.

5.3 El Controlador Optimo Proporcional

El controlador óptimo proporcional es un sistema de control realimentado, en donde la salida controlada $y(k)$ sigue a una señal de referencia $r(k) = r$ (función

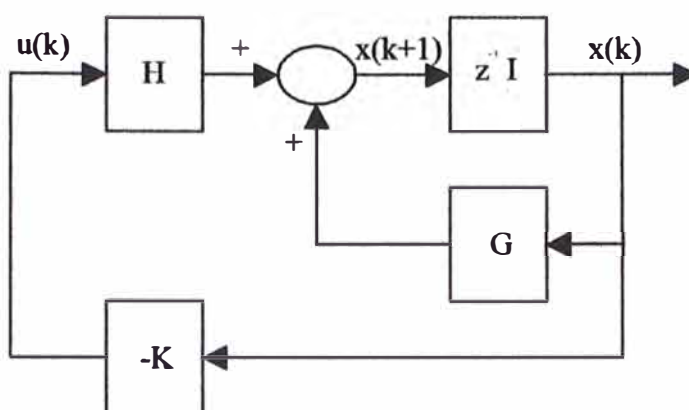


Figura 5.1: Sistema Regulator Optimo Discreto.

escalón), es decir, estamos en el caso de un sistema de seguimiento. Se restringirá el tratamiento a sistemas univariados. Esta estructura sólo es aplicable a procesos que poseen un comportamiento integral. La figura 5.2 muestra el esquema de un controlador proporcional para la variable de estado $x_2(k)$, empleando una ley de control de realimentación de estados que involucra a la matriz de ganancia del controlador K , la referencia $r(k)$ y la señal de salida $y(k)$. De dicha figura, considerando como salida al estado $x_2(k)$ se obtiene la siguiente ley de control:

$$\begin{aligned}
 u(k) &= -k_1x_1(k) - k_2x_2(k) - \dots - k_nx_n(k) + k_2[r(k) - x_2(k)] \\
 u(k) &= - \begin{bmatrix} k_1 & k_2 & k_3 & \dots & k_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{bmatrix} + k_2r(k) \\
 u(k) &= -Kx(k) + k_2r(k)
 \end{aligned} \tag{5.13}$$

Reemplazando la ecuación (5.13) en la ecuación (5.1) se obtiene:

$$\begin{aligned}
 x(k+1) &= Gx(k) + Hu(k) \\
 x(k+1) &= (G - HK)x(k) + Hk_2r(k)
 \end{aligned} \tag{5.14}$$

Aplicando la transformada z a la ecuación (5.14) se obtiene la siguiente solución de la ecuación de estado en términos de z :

$$x(z) = (zI - G + HK)^{-1}Hk_2r(z) \tag{5.15}$$

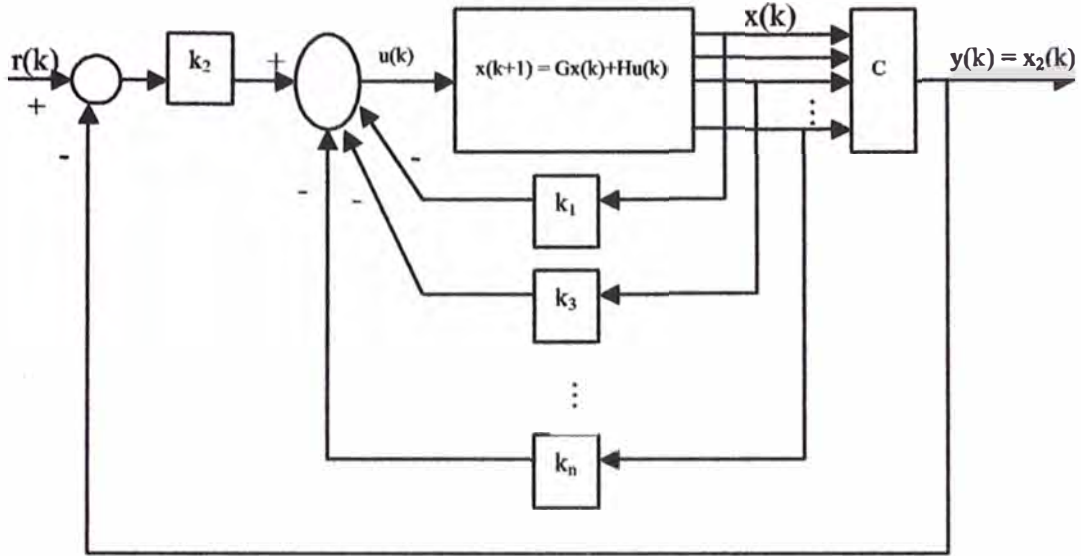


Figura 5.2: Sistema de Control Óptimo Proporcional.

Reemplazando esta última ecuación en la expresión de la salida, tenemos:

$$y(z) = C(zI - G + HK)^{-1} Hk_2 r(z) \quad (5.16)$$

Para obtener la salida en estado estacionario como respuesta a una referencia escalón unitario, aplicamos la propiedad del valor final

$$\begin{aligned} \lim_{k \rightarrow \infty} y(k) &= \lim_{z \rightarrow 1} \frac{z-1}{z} y(z) \\ \lim_{k \rightarrow \infty} y(k) &= C(zI - G + HK)^{-1} Hk_2 \end{aligned} \quad (5.17)$$

Para un perfecto seguimiento, la salida $y = 1$ (escalón unitario), condición que debe cumplirse si $C(zI - G + HK)^{-1} Hk_2 = 1$.

Idénticamente como en el caso no estacionario, las matrices de ponderación Q y R ponderan la importancia del vector de estados $x(k)$ y del vector de control $u(k)$, respectivamente.

5.4 El Controlador Óptimo Proporcional Integral

Para procesos que no poseen propiedades integradoras, la inclusión de acción integral en el sistema de control permite obtener un error estacionario nulo (ver [16] y [12]). Incluso en el caso de procesos que tengan inherentemente un integrador, es conveniente incluir un integrador en el controlador proporcional,

para asegurar error estacionario nulo no solamente para entradas escalón, sino, para trayectorias de referencia, frecuentemente usadas en sistemas de seguimiento. La figura 5.3 ilustra un controlador óptimo proporcional integral para sistemas de una entrada y una salida (SISO). Considerando la figura 5.3 se obtienen las

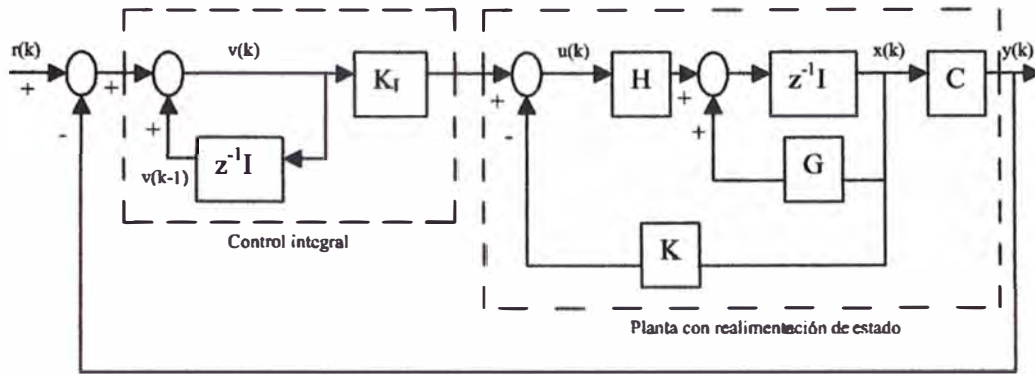


Figura 5.3: Sistema de Control Óptimo Proporcional Integral.

ecuaciones de estado y de salida del proceso, expresadas por:

$$x(k+1) = Gx(k) + Hu(k) \quad (5.18)$$

$$y(k) = Cx(k) + Du(k); \quad \text{con } D \text{ nulo} \quad (5.19)$$

Asimismo, la señal de control viene a ser:

$$u(k) = -Kx(k) + K_I v(k) \quad (5.20)$$

donde K es la matriz de ganancia del controlador, dada por:

$$K = [k_1 \quad k_2 \quad k_3 \quad \dots \quad k_n] \quad (5.21)$$

De dicha figura, podrá apreciarse que la ecuación para el integrador es:

$$v(k) = v(k-1) + r(k) - y(k) \quad (5.22)$$

Aplicando un corrimiento a la derecha a esta última ecuación, se obtiene:

$$\begin{aligned} v(k+1) &= v(k) + r(k+1) - y(k+1) \\ &= v(k) + r(k+1) - C[Gx(k) + Hu(k)] \\ &= (1 - CHK_I)v(k) + (-CG + CHK)x(k) + r(k+1) \end{aligned} \quad (5.23)$$

Reemplazando la ecuación (5.20) en la ecuación (5.18) se obtiene:

$$\begin{aligned} x(k+1) &= Gx(k) + H[-Kx(k) + K_I v(k)] \\ &= (G - HK)x(k) + HK_I v(k) \end{aligned} \quad (5.24)$$

Usando las ecuaciones (5.23) y (5.24) se obtiene la siguiente ecuación de estado matricial:

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} G - HK & HK_I \\ -CG + CHK & 1 - CHK_I \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(k+1) \quad (5.25)$$

La ecuación (5.19) puede ser reescrita de la siguiente forma:

$$y(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} \quad (5.26)$$

En estado estacionario, es decir cuando $k \rightarrow \infty$, los valores de $x(k)$, $u(k)$ y $v(k)$ toman valores invariantes $x(\infty)$, $u(\infty)$ y $v(\infty)$. Entonces, la ecuación (5.25) toma la siguiente expresión:

$$\begin{bmatrix} x(\infty) \\ v(\infty) \end{bmatrix} = \begin{bmatrix} G - HK & HK_I \\ -CG + CHK & 1 - CHK_I \end{bmatrix} \begin{bmatrix} x(\infty) \\ v(\infty) \end{bmatrix} + \begin{bmatrix} 0 \\ r \end{bmatrix} \quad (5.27)$$

Si restamos la ecuación (5.27) de la ecuación (5.25) se obtiene:

$$\begin{aligned} \begin{bmatrix} x_e(k+1) \\ v_e(k+1) \end{bmatrix} &= \begin{bmatrix} G - HK & HK_I \\ -CG + CHK & 1 - CHK_I \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \\ &= \begin{bmatrix} G & 0 \\ -CG & I \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} + \begin{bmatrix} H \\ -CH \end{bmatrix} \begin{bmatrix} -K & K_I \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \end{aligned} \quad (5.28)$$

donde

$$x_e(k) = x(k) - x(\infty) \quad (5.29)$$

$$v_e(k) = v(k) - v(\infty) \quad (5.30)$$

Considerando nuevas asignaciones, la ecuación (5.28) puede ser reescrita como:

$$\zeta(k+1) = \tilde{G}\zeta(k) + \tilde{H}\omega(k) \quad (5.31)$$

donde

$$\omega(k) = -\tilde{K}\zeta(k); \quad \zeta(k) = \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \quad (5.32)$$

$$\tilde{G}(k) = \begin{bmatrix} G & 0 \\ -CG & I \end{bmatrix}; \quad \tilde{H}(k) = \begin{bmatrix} H \\ -CH \end{bmatrix} \quad (5.33)$$

$$\tilde{K} = \begin{bmatrix} K & -K_I \end{bmatrix} \quad (5.34)$$

En tal sentido, la función de costo vendrá definida por:

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [\zeta^T(k) \tilde{Q} \zeta(k) + \omega^T(k) \tilde{R} \omega(k)] \quad (5.35)$$

Podrá apreciarse que al insertar un integrador, el orden del sistema de control aumenta en uno, es decir $(n+1)$.

Finalmente, la ecuación de Riccati dada por (5.8) y la ecuación de la ganancia del controlador dada por (5.37) quedan reformuladas como:

$$\tilde{P} = \tilde{Q} + \tilde{G}^T \tilde{P} \tilde{G} - \tilde{G}^T \tilde{P} \tilde{H} [\tilde{R} + \tilde{H}^T \tilde{P} \tilde{H}]^{-1} \tilde{H}^T \tilde{P} \tilde{G} \quad (5.36)$$

$$\tilde{K} = [\tilde{R} + \tilde{H}^T \tilde{P} \tilde{H}]^{-1} \tilde{H}^T \tilde{P} \tilde{G} \quad (5.37)$$

5.5 El Controlador Optimo Multivariable Proporcional Integral

En esta sección se tratará del diseño del controlador óptimo multivariable proporcional integral para el manipulador robótico esférico (SRM). El esquema del sistema de control es el mismo que para el proceso SISO tratado en la sección 5.4 (ver figura 5.3), con la diferencia que al tratarse de un proceso MIMO, se deben tener presente las siguientes consideraciones:

- $r(k)$ es un vector de referencia de orden r
- $u(k)$ es un vector de control de orden m
- $x(k)$ es un vector de estados de orden n
- $y(k)$ es un vector de salida de orden r
- G es una matriz de estados de dimensión $n \times n$
- H es una matriz de control de dimensión $n \times m$

- C es una matriz de salida de dimensión $r \times n$
- D es una matriz de transmisión directa de dimensión $r \times m$

donde $n = 4$ (número de variables de estado), $r = 2$ (número de salidas del proceso), $m = 2$ (número de entradas al proceso).

Considerando un período de muestreo $T = 0.01$ segundo y asumiendo retención de memoria de orden cero, las ecuaciones de estado y de salida del proceso linealizado y discretizado están dadas por las ecuaciones (5.18) y (5.19), que a continuación se reescriben:

$$x(k+1) = Gx(k) + Hu(k) \quad (5.38)$$

$$y(k) = Cx(k) + Du(k) \quad (5.39)$$

donde:

$$G = \begin{bmatrix} 1 & 0 & 0.0096 & 0 \\ 0 & 1 & 0 & 0.0096 \\ 0 & 0 & 0.9265 & 0 \\ 0 & 0 & 0 & 0.9182 \end{bmatrix}; \quad H = \begin{bmatrix} 0.0065 & 0 \\ 0 & 0.0072 \\ 1.2780 & 0 \\ 0 & 1.4223 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}; \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (5.40)$$

Las ecuaciones de estado, de salida y de control del sistema de control multivariable vienen dadas por:

$$x(k+1) = Gx(k) + Hu(k) \quad (5.41)$$

$$y(k) = Cx(k) \quad (5.42)$$

$$u(k) = -Kx(k) + K_I v(k) \quad (5.43)$$

donde $v(k)$ es el vector de salida del integrador de orden m , K es la matriz ganancia del controlador de dimensión $m \times n$ (en nuestro caso: 2×4) y la matriz ganancia del integrador de dimensión $m \times r$ que adquieren la siguiente forma:

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \end{bmatrix}; \quad K_I = \begin{bmatrix} k_{i11} & k_{i12} \\ k_{i21} & k_{i22} \end{bmatrix} \quad (5.44)$$

Asimismo, la ecuación para el integrador está dada por:

$$v(k) = v(k-1) + r(k) - y(k) \quad (5.45)$$

Finalmente, la ecuación de estado del controlador óptimo multivariable proporcional integral viene dada por:

$$\zeta(k+1) = \tilde{G}\zeta(k) + \tilde{H}\omega(k) \quad (5.46)$$

donde

$$\omega(k) = -\tilde{K}\zeta(k); \quad \zeta(k) = \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \quad (5.47)$$

$$\tilde{G}(k) = \begin{bmatrix} G & 0 \\ -CG & I \end{bmatrix}; \quad \tilde{H}(k) = \begin{bmatrix} H \\ -CH \end{bmatrix} \quad (5.48)$$

$$\tilde{K} = \begin{bmatrix} K & -K_I \end{bmatrix} \quad (5.49)$$

donde $\omega(k)$ es un vector de entrada de orden m .

La función de costo es la misma que para el caso SISO; sin embargo es necesario tener presente el orden de los vectores y las dimensiones de las matrices del sistema multivariable. Por último, la solución al problema de diseño del control óptimo multivariable proporcional integral es, calcular la matriz de realimentación de estado ampliada \tilde{k} que minimice la función de costo dada en la ecuación 5.50. La ecuación asociada de Riccati y la ganancia ampliada del controlador vienen dadas por las ecuaciones (5.36) y (5.37), que a continuación se reescriben.

$$J = \frac{1}{2} \sum_{k=0}^{\infty} [\zeta^T(k) \tilde{Q} \zeta(k) + \omega^T(k) \tilde{R} \omega(k)] \quad (5.50)$$

$$\tilde{P} = \tilde{Q} + \tilde{G}^T \tilde{P} \tilde{G} - \tilde{G}^T \tilde{P} \tilde{H} [\tilde{R} + \tilde{H}^T \tilde{P} \tilde{H}]^{-1} \tilde{H}^T \tilde{P} \tilde{G} \quad (5.51)$$

$$\tilde{K} = [\tilde{R} + \tilde{H}^T \tilde{P} \tilde{H}]^{-1} \tilde{H}^T \tilde{P} \tilde{G} \quad (5.52)$$

donde:

- \tilde{P} es una matriz simétrica definida positiva de dimensión $(n+r) \times (n+r)$
- \tilde{Q} es una matriz simétrica semidefinida positiva de dimensión $(n+r) \times (n+r)$
- \tilde{R} es una matriz simétrica definida positiva de dimensión $m \times m$

Para determinar la ganancia \tilde{K} se siguen los siguientes pasos:

1) Elegir las matrices de ponderación \tilde{Q} y \tilde{R} , que en nuestro caso son:

$$\tilde{Q} = 10 * \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \quad \tilde{R} = 10 * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.53)$$

2) Resolver la ecuación de Riccati en forma recursiva (ver ecuación (5.9)), que a continuación se reescribe:

$$\tilde{P}(k+1) = \tilde{Q} + \tilde{G}^T \tilde{P}(k) \tilde{G} - \tilde{G}^T \tilde{P}(k) \tilde{H} [\tilde{R} + \tilde{H}^T \tilde{P}(k) \tilde{H}]^{-1} \tilde{H}^T \tilde{P}(k) \tilde{G} \quad (5.54)$$

3) Calculada la matriz \tilde{P} (de elementos estacionarios), obtener la matriz ganancia del controlador \tilde{K} , usando la siguiente ecuación:

$$\tilde{K} = [\tilde{R} + \tilde{H}^T \tilde{P} \tilde{H}]^{-1} \tilde{H}^T \tilde{P} \tilde{G} = [K \quad -K_I] \quad (5.55)$$

A continuación se presenta un extracto del programa en código MATLAB que determina las matrices \tilde{P} y \tilde{K} .

```
% MATRICES AMPLIADAS DEBIDO A LA INSERCIÓN DEL INTEGRADOR EN EL
% CONTROLADOR
Ga = [G zeros(4,2)
      -Cd*G eye(2,2)];
Ha = [H; -Cd*H];

% CALCULO DE LA CONTROLABILIDAD EN TIEMPO DISCRETO DEL SISTEMA AMPLIADO
Md=[Ha Ga*Ha Ga^2*Ha Ga^3*Ha Ga^4*Ha Ga^5*Ha];
rangoMd=rank(Md); %RANGO DEL SISTEMA AMPLIADO= 6
% CALCULO DE LA OBSERVABILIDAD EN TIEMPO DISCRETO DEL SISTEMA AMPLIADO
Nd=[Cd' G'*Cd' (G')^2*Cd' (G')^3*Cd'];
rangoNd=rank(Nd); %RANGO DEL SISTEMA AMPLIADO= 4

% CONTROLADOR POR REALIMENTACIÓN DE ESTADOS NO LINEAL MIMO (DISCRETO)

% MATRICES DE PONDERACIÓN
R=10*eye(2);
Qa=10*eye(6);
P=zeros(6,6);
for i=1:1000
P=Qa+Ga'*P*Ga-Ga'*P*Ha*inv(R+Ha'*P*Ha)*Ha'*P*Ga;
```

end

```

K1=inv(R+Ha'*P*Ha)*Ha'*P*Ga;
[KK,PP]=dlqr(Ga,Ha,Qa,R); % K1=KK, P=PP
K=[KK(1,1) KK(1,2) KK(1,3) KK(1,4)
   KK(2,1) KK(2,2) KK(2,3) KK(2,4)];
KI = - [KK(1,5) KK(1,6)
        KK(2,5) KK(2,6)];

Kc1=[KK(1,1) KK(1,2) KK(1,3) KK(1,4)];
Kc2=[KK(2,1) KK(2,2) KK(2,3) KK(2,4)];
KI1= - [KK(1,5)];
KI2= - [KK(2,6)];

```

obteniéndose el siguiente resultado:

➤ P

P =

1.0e+004 *

1.5029	0	0.0131	0	-0.1056	0
0	1.4895	0	0.0121	0	-0.1047
0.0131	0	0.0015	0	-0.0009	0
0	0.0121	0	0.0014	0	-0.0008
-0.1056	0	-0.0009	0	0.0151	0
0	-0.1047	0	-0.0008	0	0.0150

➤ K1

K1 =

7.8204	0	0.5685	0	-0.5188	0
0	7.3374	0	0.5275	0	-0.4887

➤ K

K =

7.8204	-0.0000	0.5685	-0.0000
-0.0000	7.3374	-0.0000	0.5275

➤ KI

KI =

0.5188	-0.0000
-0.0000	0.4887

donde K_1 es la ganancia del controlador óptimo multivariable proporcional integral \tilde{K} .

5.6 El Observador Óptimo Cuadrático SISO

Cuando tan sólo se pueden medir en forma directa algunas de las variables del vector de estado, entonces es necesario estimar dicho vector de estado $x(k)$, es decir obtener un vector de estado estimado (ver [16] y [12]). En el diseño del estimador u observador de estados discreto se calcula su matriz de ganancia K_e , que permite la obtención de la señal de control $u(k)$. En un sistema práctico es necesario observar o estimar las variables de estado no medibles a partir de las variables de salida y las de control. En la figura 5.4 se muestra el diagrama de bloques del observador de estados discreto, donde:

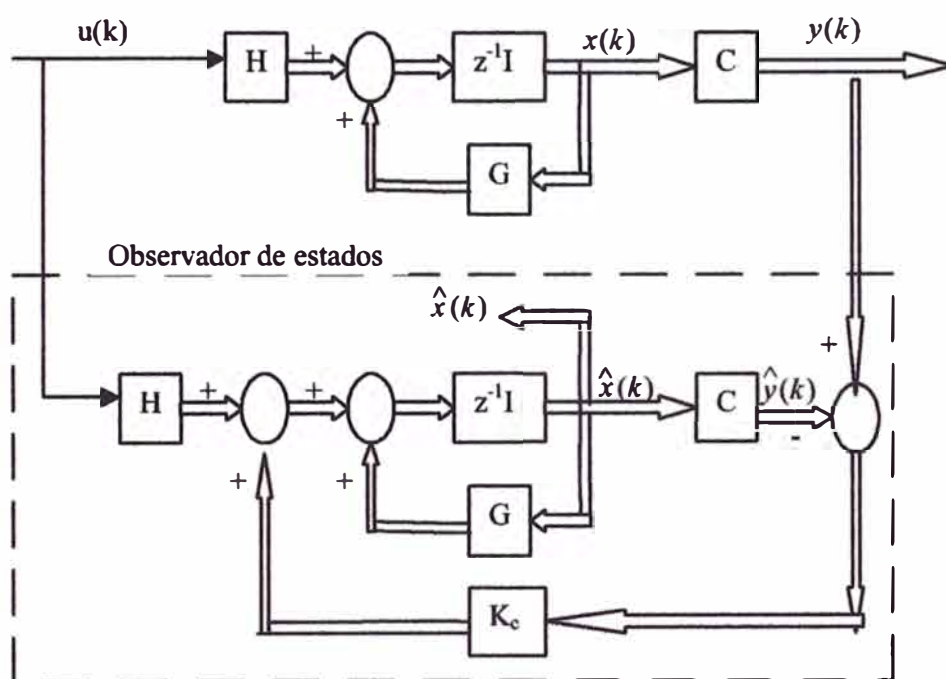


Figura 5.4: Diagrama de bloques del Observador de Estados.

- K_e : matriz de ganancia de realimentación del observador con dimensión $n \times r$ ($n \times 1$).

- C : matriz de salida de dimensión $r \times n$ ($1 \times n$).
- $\hat{x}(k)$: vector de estado estimado de dimensión n .
- $\hat{y}(k)$: vector de salida estimado de dimensión r ($r = 1$).

de donde se pueden escribir las siguientes ecuaciones del proceso

$$x(k+1) = Gx(k) + Hu(k) \quad (5.56)$$

$$y(k) = Cx(k) \quad (5.57)$$

y del observador

$$\begin{aligned} \hat{x}(k+1) &= G\hat{x}(k) + Hu(k) + K_e[y(k) - \hat{y}(k)] \\ &= G\hat{x}(k) + Hu(k) + K_e[y(k) - C\hat{x}(k)] \\ &= G\hat{x}(k) + Hu(k) + K_eC[x(k) - \hat{x}(k)] \end{aligned} \quad (5.58)$$

Al observador o estimador de estados se le denomina también observador de predicción, porque el vector de estado estimado está un período de muestreo delante de la medición $y(k)$.

Ecuación de error del observador

Restando la ecuación (5.58) de la ecuación (5.56) se obtiene la siguiente ecuación de error del observador:

$$e(k+1) = [G - K_eC]e(k); \quad (5.59)$$

donde:

$$e(k) = x(k) - \hat{x}(k) \quad (5.60)$$

Ecuación característica del observador

La estabilidad del observador se determina resolviendo la siguiente ecuación característica:

$$|zI - G + K_eC| = 0 \quad (5.61)$$

debiendo verificarse lo siguiente:

- Las raíces de la ecuación característica deben posicionarse dentro del círculo unitario para operación satisfactoria del observador.

- K_e debe ser escogida apropiadamente para que el error tienda a cero.
- El proceso debe ser completamente observable, condición que se consigue aplicando el criterio de observabilidad:

$$N = \begin{bmatrix} C^T & G^T C^T & (G^T)^2 C^T & \dots & G^T)^{n-1} C^T \end{bmatrix} \quad (5.62)$$

Ecuación de Riccati

La ecuación de Riccati para el observador es:

$$P_e = Q_e + GP_e G^T - GP_e C^T [R_e + CP_e C^T]^{-1} CP_e G^T \quad (5.63)$$

Para determinar P_e aplicamos el mismo procedimiento empleado para calcular P , dada en la ecuación (5.8) pero con la inversión en la dirección del tiempo y efectuando las siguientes modificaciones:

$$G \rightarrow G^T; \quad H \rightarrow C^T; \quad K \rightarrow K_e^T$$

obteniéndose

$$P_e(k+1) = Q_e + GP_e(k)G^T - GP_e(k)C^T [R_e + CP_e(k)C^T]^{-1} CP_e(k)G^T \quad (5.64)$$

Matriz de ganancia K_e

Considerando las mismas modificaciones que para la ecuación de Riccati, obtenemos la matriz de ganancia del observador, que toma la siguiente forma:

$$K_e = [R_e + CP_e C^T]^{-1} CP_e G^T \quad (5.65)$$

Para el caso del observador de estados, las matrices de ponderación R_e y Q_e deben ser elegidas de tal forma que la respuesta del observador sea dos o tres veces más rápida en comparación con la respuesta del proceso. Generalmente para que esto ocurra, los elementos de R_e deben ser bastantes menores que los elementos de Q_e .

5.7 El Observador Óptimo Cuadrático MIMO

Las ecuaciones del observador óptimo cuadrático MIMO son las mismas que para el caso SISO, con las siguientes consideraciones:

- K_e es una matriz de ganancia del observador de dimensión $n \times r$ (en nuestro caso: 4×2)

- P_e es una matriz simétrica definida positiva de dimensión $n \times n$ (en nuestro caso: 4×4)

En consecuencia, reescribimos las ecuaciones que gobiernan la dinámica del observador MIMO:

$$\begin{aligned}\hat{x}(k+1) &= G\hat{x}(k) + Hu(k) + K_e[y(k) - \hat{y}(k)] \\ &= G\hat{x}(k) + Hu(k) + K_e[y(k) - C\hat{x}(k)] \\ &= G\hat{x}(k) + Hu(k) + K_eC[x(k) - \hat{x}(k)]\end{aligned}\quad (5.66)$$

$$e(k+1) = [G - K_eC]e(k); \quad (5.67)$$

$$|zI - G + K_eC| = 0 \quad (5.68)$$

$$P_e(k+1) = Q_e + GP_e(k)G^T - GP_e(k)C^T[R_e + CP_e(k)C^T]^{-1}CP_e(k)G^T \quad (5.69)$$

$$K_e = [R_e + CP_eC^T]^{-1}CP_eG^T \quad (5.70)$$

A continuación se presenta un extracto del programa en MATLAB para el cálculo de la matriz de Riccati y la ganancia del observador de estados.

```
% CALCULO DE LA GANANCIA DEL OBSERVADOR DE ESTADOS
```

```
% MATRICES DE PONDERACION PARA EL OBSERVADOR
```

```
Qe=100*eye(4); Re=10*eye(2);
```

```
Pe = zeros(4,4);
```

```
for i=1:200
```

```
Pe = Qe + G*Pe*G' -G*Pe*Cd'*inv(Re+Cd*Pe*Cd')*Cd*Pe*G';
```

```
end
```

```
KeT = inv(Re+Cd*Pe*Cd')*Cd*Pe*G'; Ke=KeT';
```

```
[KKeT,PPe] = dlqr(G',Cd',Qe,Re); KKe=KKeT';% Pe=PPe, Ke=KKe
```

que produce los siguientes resultados:

```
> Pe
```

```
Pe =
```

```
109.2375      0      6.8051      0
      0 109.2294      0      6.0612
      6.8051      0 703.9833      0
      0      6.0612      0 635.9144
```

```
> Ke
```

$K_e =$

0.9167	0
0	0.9166
0.0529	0
0	0.0467

⇒ KK_e

$KK_e =$

0.9167	0.0000
0.0000	0.9166
0.0529	0.0000
0.0000	0.0467

donde K_e es la matriz de ganancia del observador calculado recursivamente y KK_e es la matriz de ganancia del observador calculado mediante el comando *dlqr* de MATLAB, con los mismos resultados por cualquiera de las formas.

5.8 Simulación del Sistema de Control Optimo Multivariable Discreto

Considerando los resultados obtenidos en las secciones 5.5 y 5.7, se puede efectuar la simulación, contando para ello con el programa en código MATLAB *srm4optd.m* listado en el APENDICE A, cuyo resultado se muestra en las figuras 5.5 y 5.6. Las señales de referencia usadas son $r_1(t) = \text{sen } t + 0.1t$, $r_2(t) = \text{cos } t$, y el período de muestreo considerado es $T = 0.01$ segundos (equivalente a una frecuencia de muestreo de 100 Hz).

Del resultado gráfico, se puede observar de la figura 5.5 que la señal de control $u_1(t)$ necesario para que la posición angular $x_1(t)$ de la base del SRM siga a la referencia $r_1(t)$, alcanza un valor pico transitorio de aproximadamente $0.01 V$, decreciendo hasta un valor negativo muy pequeño en tiempo estacionario. Idénticamente, de la figura 5.6 se puede observar que la tensión de control $u_2(t)$ necesario para llevar la posición angular del brazo del SRM a la posición deseada $r_2(t)$, alcanza un valor pico transitorio de aproximadamente $0.7 V$, decreciendo hasta aproximadamente $0.2 V$ en tiempo estacionario. Se puede observar asimismo, que la base del SRM sigue exactamente a la referencia r_1 , y en el caso de la respuesta del brazo, existe un sobrepico de aproximadamente 12 % en tiempo transitorio de 1 segundo de duración, y siguiendo con exactitud a la referencia r_2

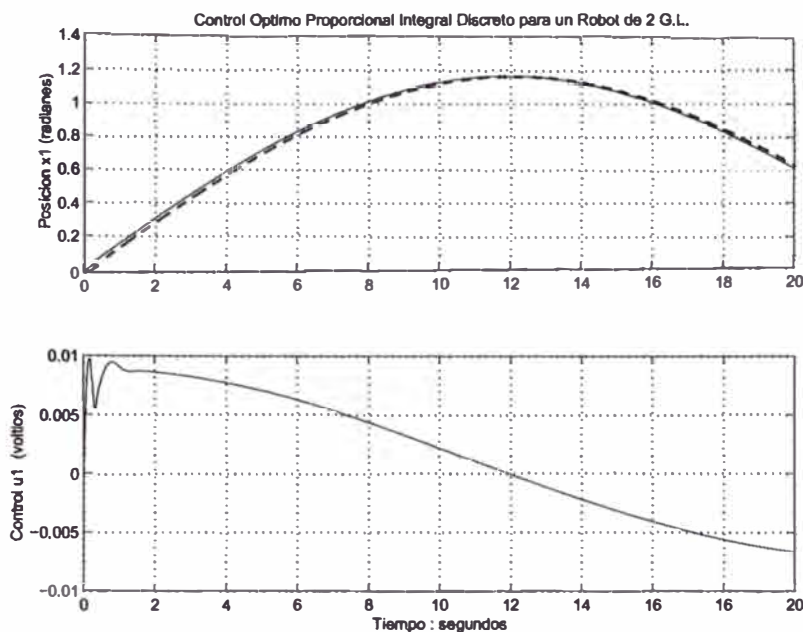


Figura 5.5: Salida controlada x_1 (posición angular de la base del SRM) y señal de control u_1 .

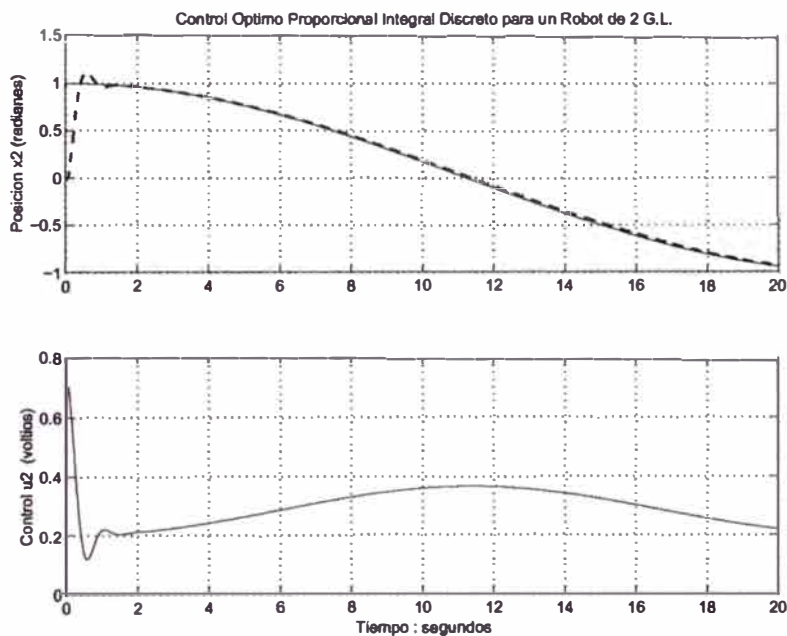


Figura 5.6: Salida controlada x_2 (posición angular del brazo del SRM) y señal de control u_2 .

en tiempo estacionario. Considerando referencias constantes, el resultado de la simulación no es satisfactorio, según puede apreciarse en la figuras 5.7 y 5.8; lo que obliga a usar una frecuencia de muestreo mayor.

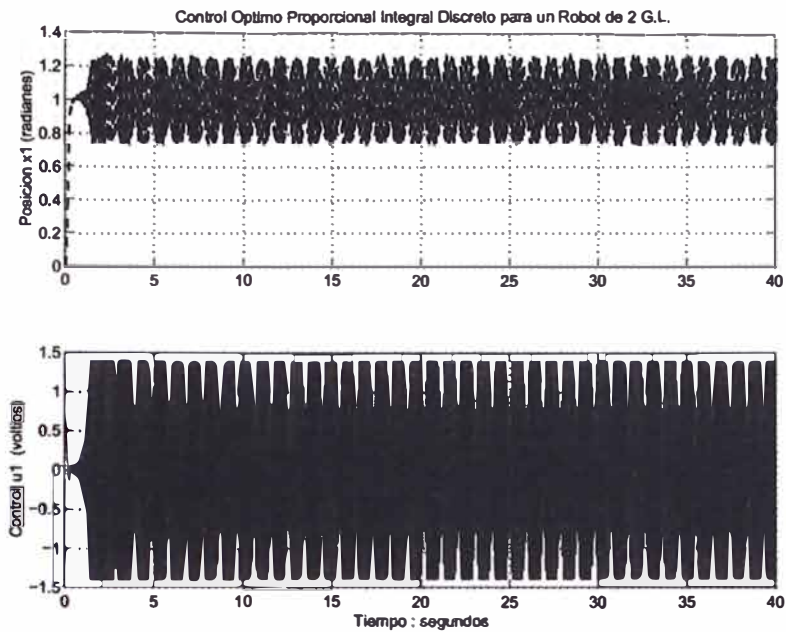


Figura 5.7: Salida x_1 (posición angular de la base del SRM) y señal de control u_1 para un escalón de 1 unidad.

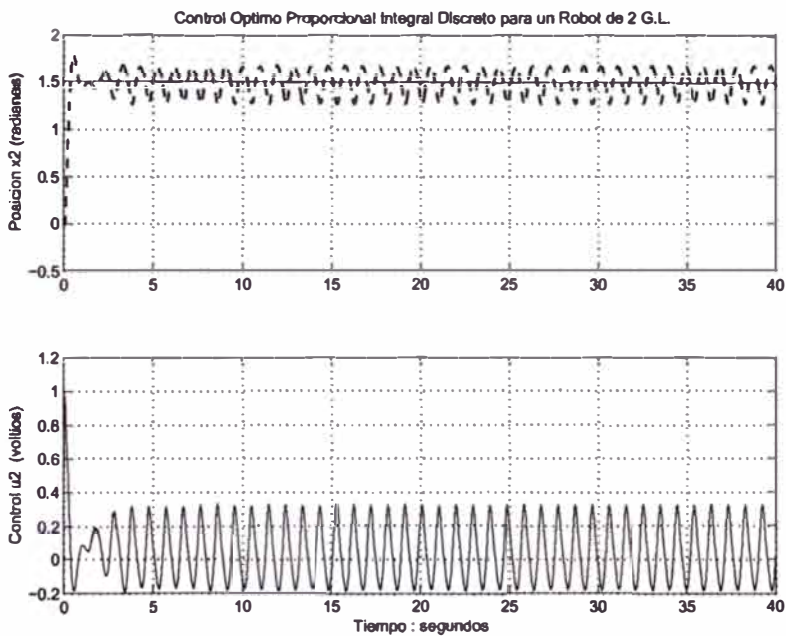


Figura 5.8: Salida x_2 (posición angular del brazo del SRM) y señal de control u_2 para un escalón de 1.5 unidades.

Considerando una frecuencia de muestreo de 200 Hz, equivalente a un período de muestreo de $T = 0.005$ segundos, la respuesta del sistema a trayectorias se muestran en las figuras 5.9 y 5.10, y la respuesta a referencias constantes

se muestran en las figuras 5.11 y 5.12. Debido a estos últimos resultados se

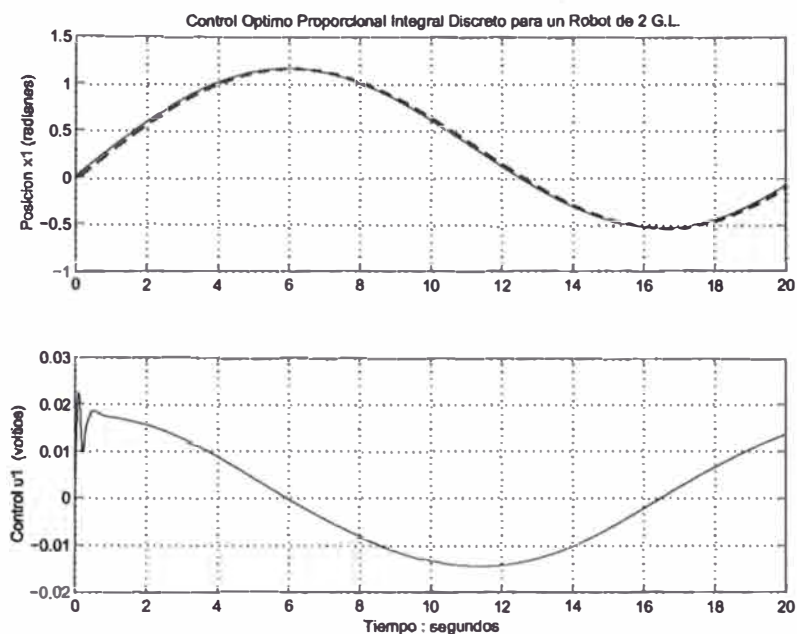


Figura 5.9: Salida x_1 (posición angular de la base del SRM) y señal de control u_1 .

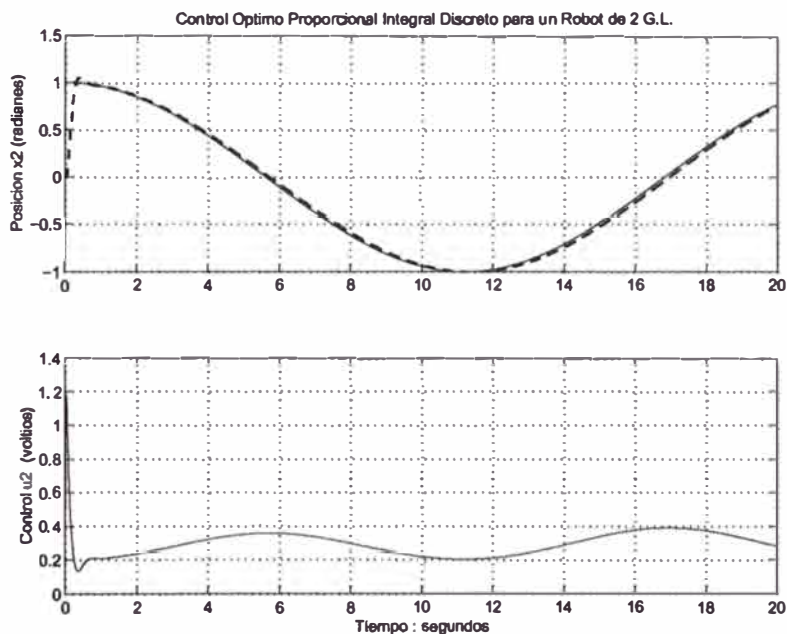


Figura 5.10: Salida x_2 (posición angular del brazo del SRM) y señal de control u_2 .

ha considerado usar un período de muestreo de $T = 0.005$ segundos (200 Hz.)

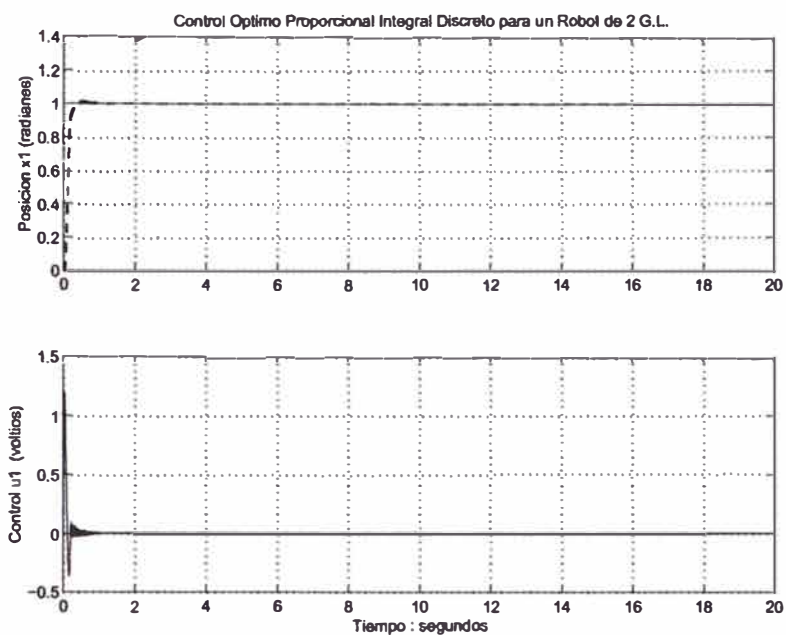


Figura 5.11: Salida x_1 (posición angular de la base del SRM) y señal de control u_1 para un escalón de 1 unidad.

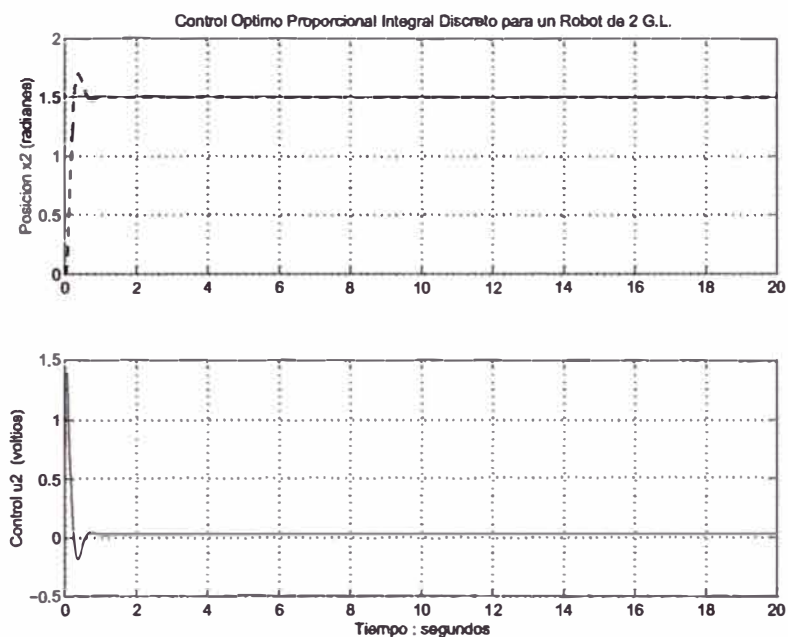


Figura 5.12: Salida x_2 (posición angular del brazo del SRM) y señal de control u_2 para un escalón de 1.5 unidades.

por presentar un desempeño óptimo tanto para referencias constantes como para trayectorias.

CAPÍTULO VI

PROPUESTA DE IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se aborda la propuesta de implementación del sistema de control, tanto del Hardware (parte física) y del software (parte lógica) [21], que involucra a la planta (SRM) y al controlador óptimo con observador de estados, aspectos que se pasarán a explicar.

6.1 Implementación del Hardware

En la figura 6.1 se presenta el esquema de implementación del sistema de control óptimo multivariable con observador de estados, en cualquiera de sus versiones, así como el correspondiente al control en lazo abierto del efector final. En dicho esquema se puede observar que la parte física del sistema de control del SRM consta de:

- Un manipulador robótico de dos grados de libertad (SRM), que dispone de dos servomotores D.C. con codificador óptico incorporado y un sistema de reducción de velocidad. Un servomotor se fija en la base, y el otro, en el brazo. Adicionalmente, se dispone de un motor paso a paso para accionar una pinza que permita coger objetos de pequeño peso. Esta unidad viene a ser el proceso a controlar.
- Una tarjeta interfaz de potencia (Generador PWM + Amplificador de potencia), una unidad para cada servomotor.
- Una tarjeta sensora, conteniendo dos unidades similares de decodificador + contador UP/DOWN (una por cada servomotor).
- Una tarjeta de interfaz del efector final.
- Una PC Pentium.

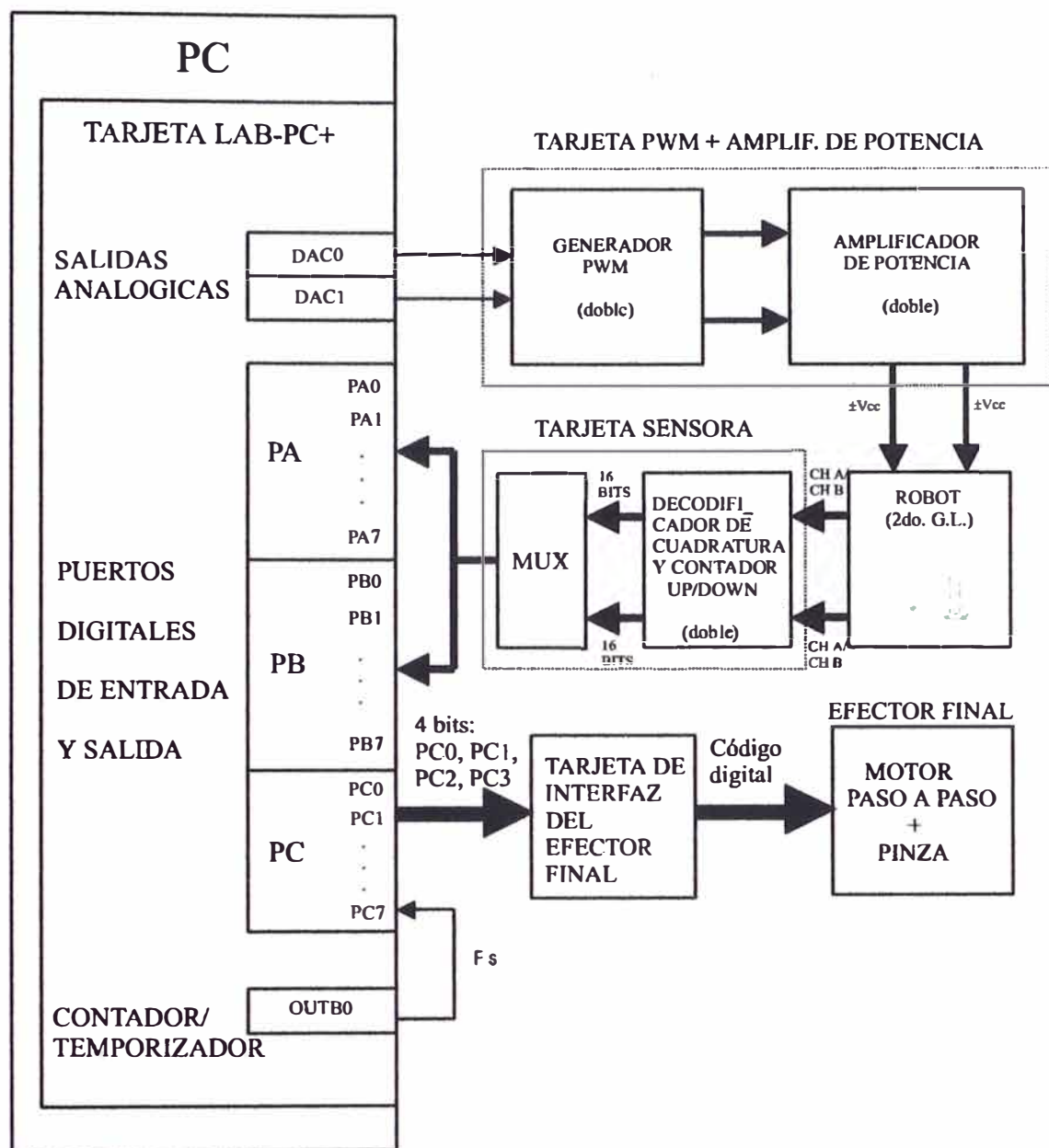


Figura 6.1: Esquema de implementación del sistema de control de posición del SRM.

- Una tarjeta de adquisición de datos LAB-PC+ de NATIONAL INSTRUMENTS insertada en el slot de la PC.

En cuanto a métodos de control de potencia de los motores de corriente continua, podemos citar:

- Rectificación controlada por silicio (tiristores y triacs), permitiendo regular el tiempo de cada semiciclo rectificado, durante el cual deja pasar corriente

al circuito.

- Conmutación electrónica, mediante puente de transistores tipos T o H .
- Modulación por Ancho de pulsos (PWM), permitiendo aplicar potencia al motor mediante impulsos de anchura variante, y que aplicados a la entrada de un amplificador de conmutación tipo T o H , producen tensiones de salida aproximadamente igual a la tensión de alimentación, que se aplica a los bornes del motor. A mayor ancho de pulso se consigue una mayor tensión de salida, alcanzando un máximo de $\pm V_{cc}$.
- Modulación de frecuencia de pulsos (PFM), similar al PWM, pero donde se varía la frecuencia de los pulsos, manteniendo constante la anchura de los mismos.

Como se puede apreciar, en esta tesis se propone el uso del método PWM, por su alta fiabilidad y por contar en la actualidad con transistores MOSFET de alta velocidad de conmutación, así como el soporte de altos niveles de corriente. En la figura 6.2 se muestra un tren de pulsos de anchura variable y de período constante (señales PWM).

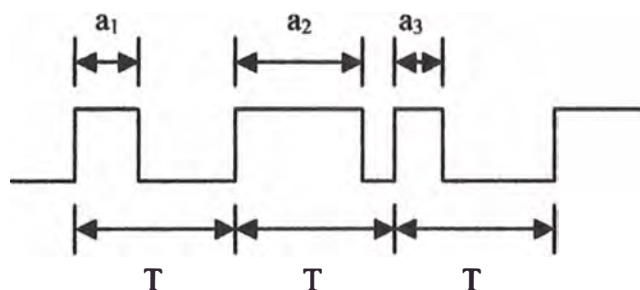


Figura 6.2: Tren de pulsos de anchura variante (PWM).

Tarjeta Interfaz de Potencia

Esta tarjeta está conformada por dos unidades idénticas de Generador PWM y el Circuito Amplificador de Potencia, que nos permite variar la velocidad o la posición angular de la base y del brazo del manipulador robótico de acuerdo al ancho de pulso de la señal PWM. Cada uno de los generadores PWM ("Pulse Width Modulation") LM3524 genera una modulación por ancho de pulsos de 15.4 KHz, siendo amplificada por cada uno de los circuitos de potencia,

los cuales entregan una tensión de alimentación de ± 24 voltios a cada uno de los motores. Cada uno de los Circuitos de Potencia están conformados por un sistema de disparo y dos pares de conmutadores A y B, según se muestra en la figura 6.3.

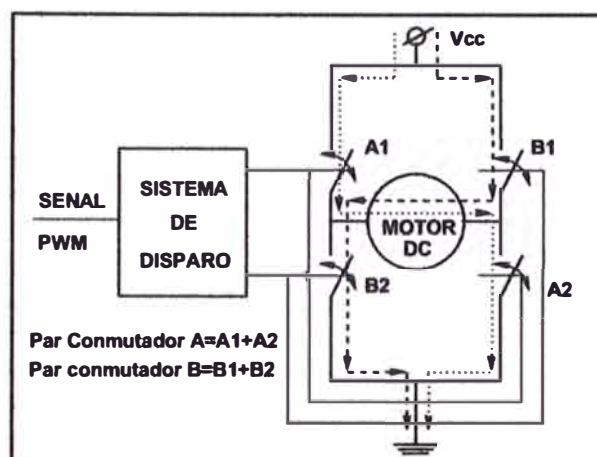


Figura 6.3: Esquema general del sistema de disparo y conmutación

El sistema de disparo permite la conmutación de cada par A y B, de tal forma que se eviten cortocircuitos durante el tiempo de conmutación de A y B. Cuando el sistema de disparo cierra el conmutador A y abre B, el sentido de la corriente es la línea punteada, induciendo de esta forma una tensión de $+V_{cc}$ en el motor. Ahora, si el sistema de disparo abre el conmutador A y cierra B, el sentido de la corriente es la línea segmentada, induciendo así una tensión $-V_{cc}$ en el motor. Esto significa que el motor recibe en sus bornes una tensión de onda cuadrada cuyo valor varía entre $\pm V_{cc}$, con un consumo de corriente que depende de los conmutadores conformados por transistores MOSFET.

Tarjeta Sensora

La tarjeta sensora contiene dos unidades idénticas de decodificadores de cuadratura y contadores UP/DOWN. Cada decodificador recibe señales en cuadratura (Canal A y Canal B) desde cada uno de los codificadores incorporados en cada motor. Asimismo, cada decodificador entrega un par de señales (UP/DOWN) a cada una de las entradas de un contador UP/DOWN de 16 bits. Cada unidad de contadores de 16 bits entregan la información binaria a un MUX, que se encarga de acoplar alternadamente los 16 bits provenientes de cada unidad

contadora a los puertos de entrada digital PA y PB de la tarjeta de adquisición de datos (insertado en el slot del PC). El esquema básico del codificador óptico se muestra en la figura 6.4. El codificador usado tiene una resolución de 100

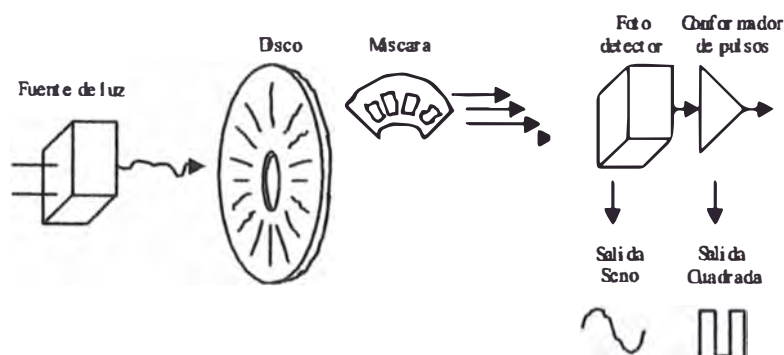


Figura 6.4: Estructura de un codificador óptico rotatorio

ppr (pulsos por revolución). El decodificador de cuadratura es el CI LS7083, que activará la cuenta ascendente o descendente del contador UP/DOWN de 16 bits, dependiendo del sentido de giro del motor, captado por el desfase entre los trenes de pulsos de los canales A y B. Cada unidad de contadores UP/DOWN lo conforman 4 CI 74HC193, haciendo un total de 16 bits por canal, que ingresan al MUX, dispositivo que se encarga de entregar dos grupos alternados de 16 bits a la tarjeta de adquisición de datos.

El esquema general de la estructura de la tarjeta sensora se muestra en la figura 6.5.

En la figura 6.6 se muestra el esquema de principio para la selección de los buffer 74LS244, por consiguiente, para el multiplexaje de datos provenientes de cada unidad contadora de 16 bits. Los XOR 74LS86 habilitan el funcionamiento de cada bloque de 2 C.I. 74LS244, de acuerdo a la tabla 6.1 mostrada, permitiendo que los puertos A y B (PA y PB) de la tarjeta de adquisición de datos capturen los datos correspondientes a las posiciones angulares de los motores M1 y M2. La salida S1 habilita o deshabilita el primer bloque 74LS244, y la salida S2 hace lo mismo pero para el segundo bloque 74LS244. Como puede observarse, la señal *SEL* proviene de un circuito astable, conformado básicamente por un TIMER NE 555 operando a una frecuencia de 100 Hz., frecuencia adecuada para que los datos de cada unidad contadora puedan ser captados por los puertos A y B de la

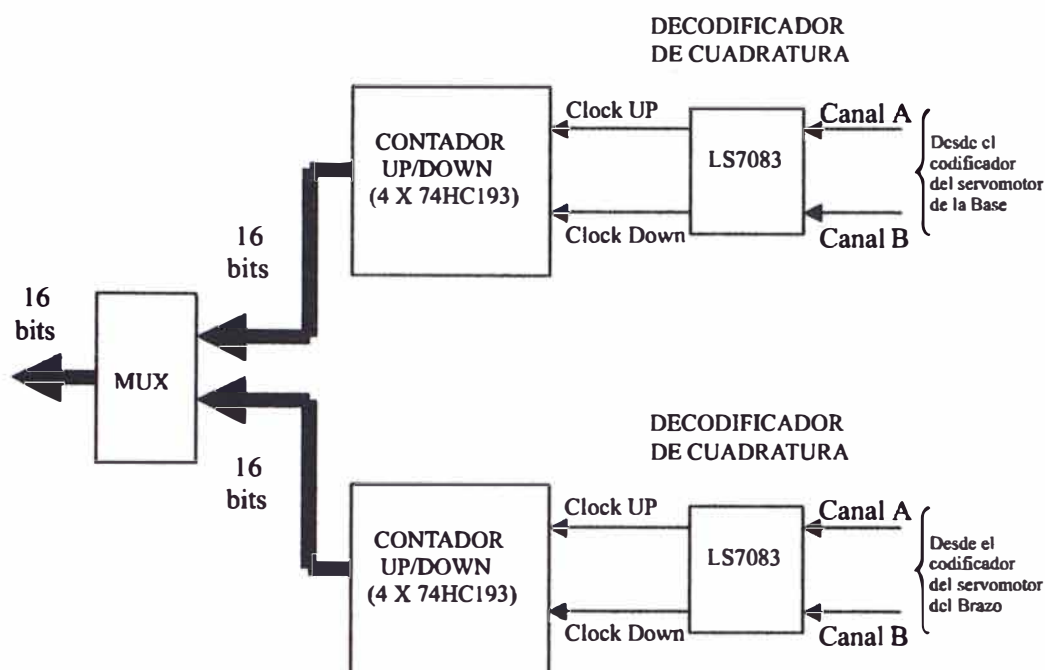


Figura 6.5: Esquema general de la tarjeta sensora.

tarjeta de adquisición de datos a la frecuencia de 200 Hz (debido a los dos canales de datos multiplexados).

Tabla 6.1: Tabla de verdad del multiplexaje de datos.

A	SEL	S1	B	SEL	S2
1	0	1	0	0	0
1	1	0	0	1	1

Tarjeta de Interfaz del Efecto Final

La implementación de sistemas de control para motores paso a paso se pueden realizar mediante control por PC, microcontrolador, DSP, o por medio de circuitos especializados que generan una secuencia de códigos, permitiendo en cualquiera de los casos, el control del sentido de giro y velocidad del motor paso a paso.

Una configuración básica de interfazado con el motor paso a paso (PAP) se muestra en la figura 6.7 [5]. Se puede apreciar la conexión del circuito integrado ULN 2003 (buffer amplificador) a las bobinas del motor PAP unipolar de 6 hilos. La entrada al circuito proviene del bus de datos del puerto C (sólo los 4 bits más

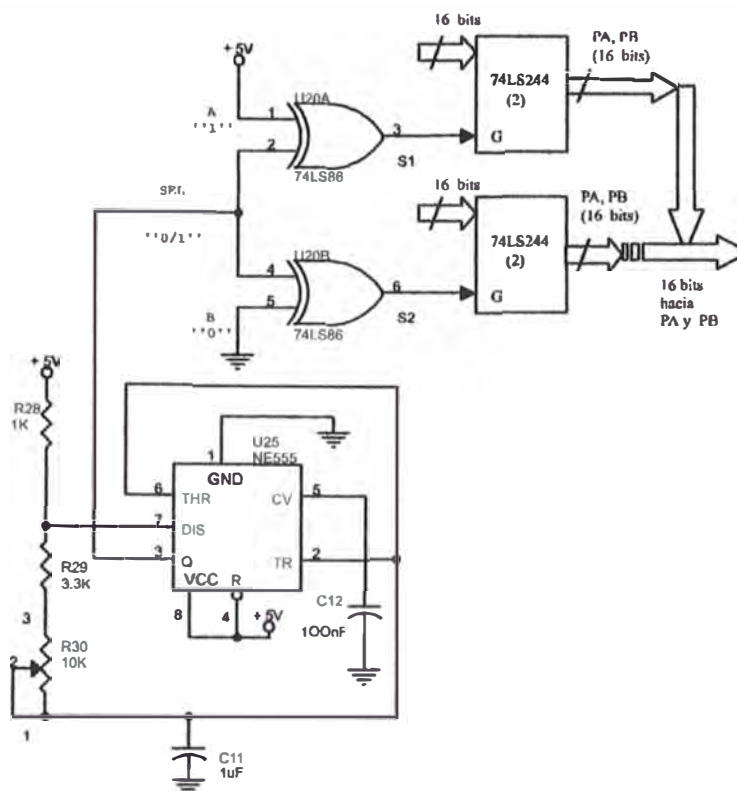


Figura 6.6: Esquema de principio para el multiplexaje de datos de 16 bits provenientes de cada unidad contadora.

bajos), configurado como salida.

Otro circuito integrado que pudiera usarse, y es el que se sugiere en el presente trabajo, es el C.I. L293B (de SGS-THOMSON) con el cual se puede conseguir la conmutación de la corriente por las bobinas del motor PAP, con la necesaria amplificación de las señales provenientes de la tarjeta de adquisición de datos, con alta inmunidad al ruido y protección contra sobre-temperaturas. En la figura 6.8 se muestra el esquema interno del C.I. L293B, que contiene 4 drivers. La pareja de drivers 1 y 2 es habilitada por la señal *EN1* y la otra pareja de drivers 3 y 4 es habilitada por la señal *EN2*. En la tabla 6.2 se muestran las características eléctricas del L293B, y en la figura 6.3 su tabla de verdad, donde H=nivel alto "1", L=nivel bajo "0", Z=alta impedancia "OFF".

Es frecuente que los fabricantes de motores presenten, conjuntamente con la información técnica, la secuencia de códigos que se debe aplicar a las bobinas de un motor PAP, para ambos sentidos de giro. En vista que se cuenta con un motor PAP, extraído de una disquetera, se propone su uso. En la figura 6.9

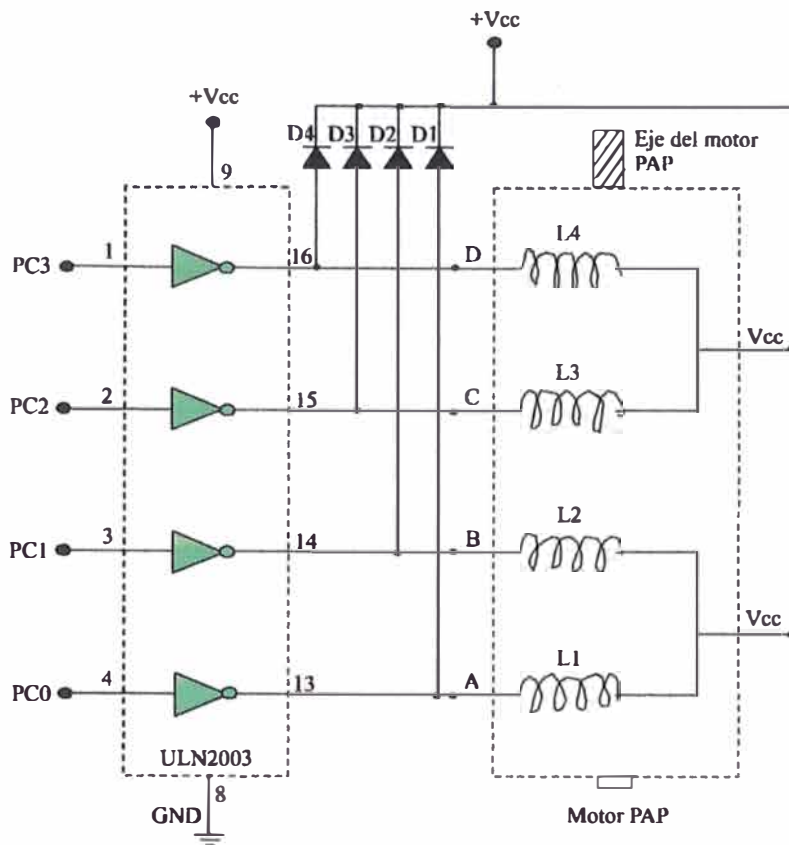


Figura 6.7: Conexión del C.I. ULN2003 a un motor PAP unipolar de 6 hilos.

se muestra dicho motor, con las siguientes características: $7.5^\circ/\text{paso}$, 48 pasos, 12V/fase, 0.25A/fase, 4 fases.

Para determinar los bobinados del motor PAP, se hicieron pruebas de medición de resistencia con el ohmímetro, y, para determinar la secuencia de códigos que deben aplicarse a los terminales de los bobinados, se hicieron pruebas experimentales para ambos sentidos de giro. De acuerdo a la figura 6.7, el terminal D corresponde al cable negro, el terminal C corresponde al cable marrón, el terminal B corresponde al cable naranja, y el terminal A corresponde al cable amarillo. Los terminales centrales V_{cc} de cada par de bobinados corresponden a los cables de color rojo.

La secuencia de códigos del motor PAP referido, se encuentra en la tabla 6.4, y el esquema de implementación de la interface entre el C.I. L293B y el motor PAP se muestra en la figura 6.10.

Tabla 6.2: Características Eléctricas del L293B. Para cada canal, $V_{ss}=24V$, $V_{cc}=5V$, $T_{ambiente}=25^{\circ}C$.

Símbolo	Parámetro	Mín.	Típ.	Máx.	Unidad
V_{ss}	Alimentación de las cargas	V_{ss}		36	V
V_{cc}	Alimentación de la lógica	4.5		36	V
V_{IL}	Tensión de entrada a nivel "0"	-0.3		1.5	V
V_{IH}	Tensión de entrada a nivel "1"	2.3		V_{cc}	V
I_{IL}	Corriente de entrada a nivel "0"			-10	μA
I_{IH}	Corriente de entrada a nivel "1"		30	100	μA
V_{inL}	Tensión de habilitación a "0"	-0.3		1.5	V
V_{inH}	Tensión de habilitación a "1"	2.3		V_{cc}	V
I_{inL}	Corriente de habilitación a "0"		-30	-100	μA
I_{inH}	Corriente de habilitación a "1"			± 10	μA

Tabla 6.3: Tabla de verdad del C. I. L293B.

V_{EN}	V_{IN}	V_{OUT}
H	H	H
H	L	L
L	H	Z
L	L	Z

Tabla 6.4: Secuencia de códigos para el giro del motor PAP.

Paso	Rotación: horaria	Base. Hexad.	Rotación: anti- horaria	Base Hexad.
	D C B A		D C B A	
1	1 0 1 0	A	1 0 1 0	A
2	0 1 1 0	6	1 0 0 1	9
3	0 1 0 1	5	0 1 0 1	5
4	1 0 0 1	9	0 1 1 0	6

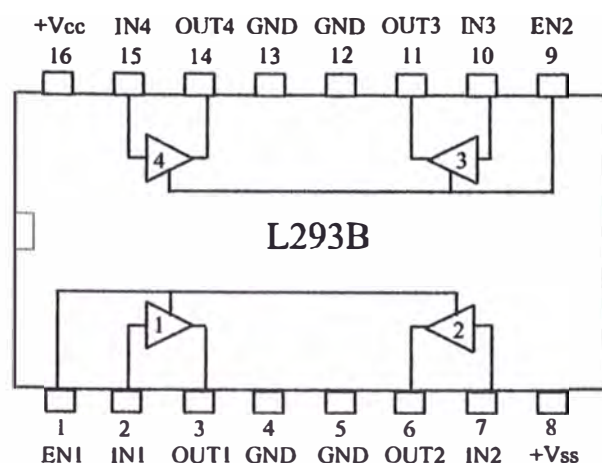


Figura 6.8: Esquema interno del C.I. L293B.

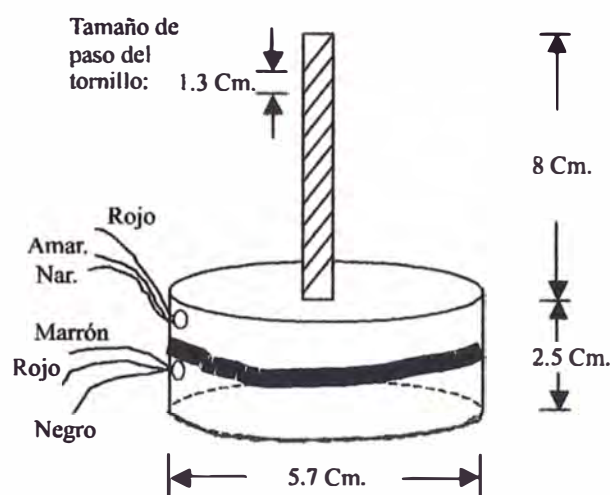


Figura 6.9: Esquema del motor PAP elegido.

La figura 6.11 ilustra el tipo de pinza a utilizarse en el manipulador robótico, donde el eje tornillo del motor PAP permite el desplazamiento horizontal de la barra de sección S_b de la pinza en ambas direcciones. La barra de sección S_a es fija, y en su parte posterior se sujeta el área frontal del motor PAP. El material empleado en la construcción de la pinza es de aluminio.

Tarjeta de Adquisición de Datos LAB-PC+

La interface entre el microcomputador, la tarjeta "PWM+Amplificador de potencia", el robot (planta) y la tarjeta sensora, es la tarjeta de adquisición de datos LAB-PC+ (ver [18]). Los terminales usados para el control son:

1. Dos salidas analógicas (DAC0 y DAC1). Las señales de control se envían

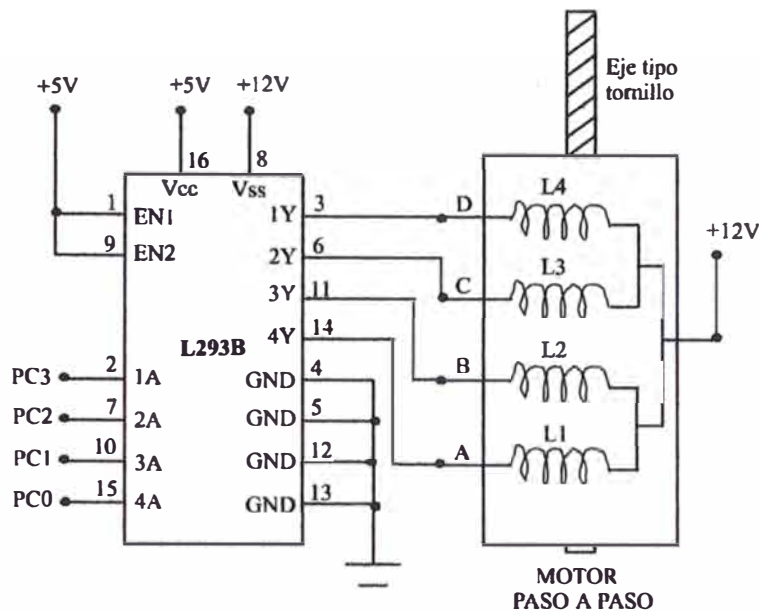


Figura 6.10: Esquema de conexión entre el C.I. L293 y el motor PAP de 6 hilos.

por los terminales DAC0 y DAC1 de la tarjeta a cada uno de los CI LM3524 (Generador PWM) con una resolución de 12 bits para generar la señal analógica correspondiente.

2. Dos puertos digitales de entrada (PA y PB) de 8 bits cada uno. Dichos puertos reciben los datos provenientes del MUX incorporado en la tarjeta sensora.
3. Puerto C de 8 bits, de los cuales se usan los 4 bits más bajos (4 bits: *PC3*, *PC2*, *PC1* y *PC0*) para el control de un motor paso a paso de 6 hilos que accione la pinza del robot.
4. El terminal OUTB0, se utiliza como salida de una señal de reloj para el funcionamiento del sistema de control.
5. El terminal PC7 (bit 8 del puerto C), se utiliza como entrada de la señal de reloj proveniente de 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 provienen del PPI (Interface Paralela Programable) 8255.

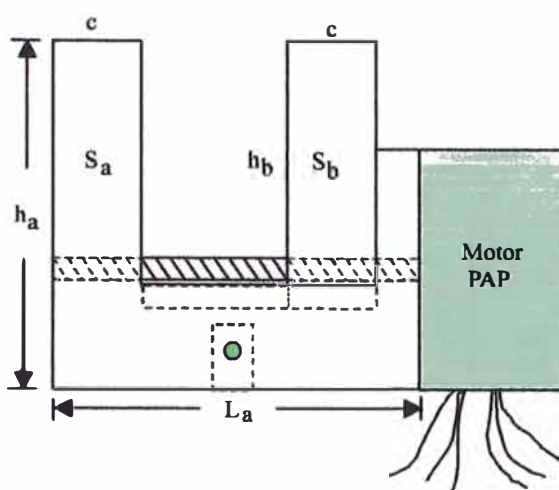


Figura 6.11: Vista lateral de la pinza tipo tornillo.

En la figura 6.12 se muestra el diagrama de bloques de la tarjeta de adquisición de datos Lab_PC +. Asimismo, en la figura 6.13 se muestra el diagrama de bloques correspondiente al puerto de entrada/salida digital.

Las dos salidas analógicas usadas para proveer las señales de control hacia los generadores PWM, se muestran en la figura 6.14.

6.2 Implementación del Software de Control

El software usado para la implementación de los algoritmos de control de posición del SRM, emplea el **lenguaje C** y se ejecuta en una PC Pentium. Previamente, las simulaciones se realizaron empleando **MATLAB**.

Los programas en tiempo real para las dos versiones de control óptimo, se estructuran cada uno de ellos en tres archivos de código fuente y un archivo que corresponde a la librería del NIDAQ:

Control Optimo Continuo

- R2lib.cpp
- R2lib.h
- srm4opt.cpp
- nidaqbc.lib

Control Optimo Discreto

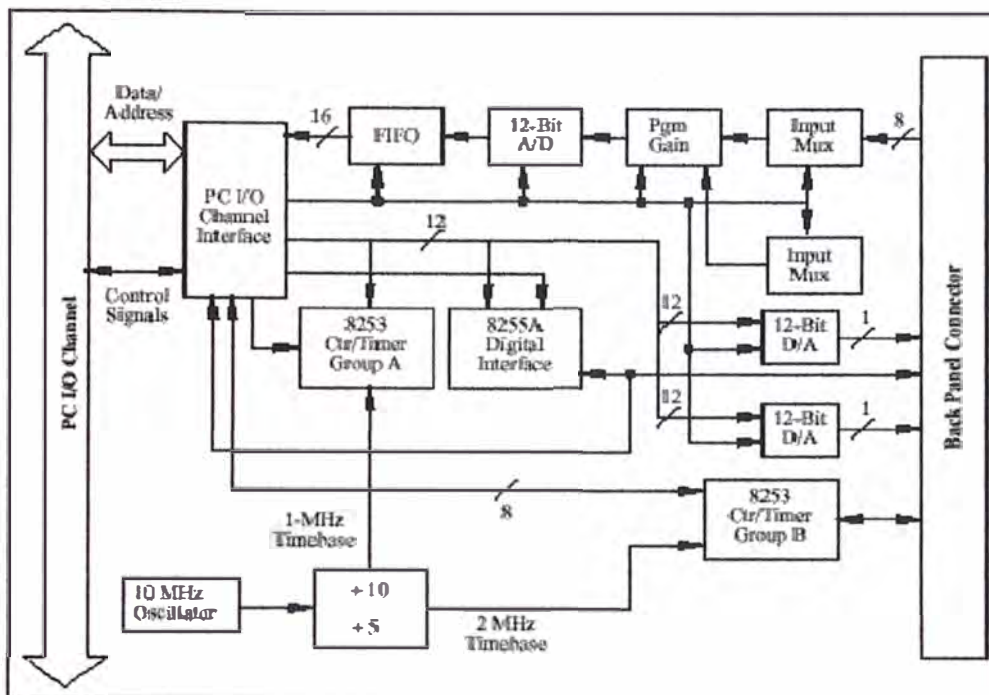


Figura 6.12: Diagrama de bloques de la tarjeta Lab_PC +.

- R2llib.cpp
- R2llib.h
- srm4optd.cpp
- nidaqbc.lib

Los archivos ejecutables para cada una de las versiones del control, se consigue mediante la creación de los respectivos proyectos, denominados *srm4opt.prj* y *srm4optd.prj*. En cada uno de estos archivos de proyecto, se incluyen los dos archivos fuente de extensión *cpp* y el archivo *nidaqbc.lib* (correspondiente a la librería de NIDAQ de la tarjeta de adquisición de datos). Al ejecutarse cada uno de estos proyectos independientemente, se generan los respectivos archivos ejecutables *srm4opt.exe* y *srm4optd.exe*.

El diagrama de flujo mostrada en la figura 6.15 presenta los siguientes pasos:

1. Inicialización de la tarjeta de adquisición de datos *LAB - PC+* y de las variables utilizadas en el programa.

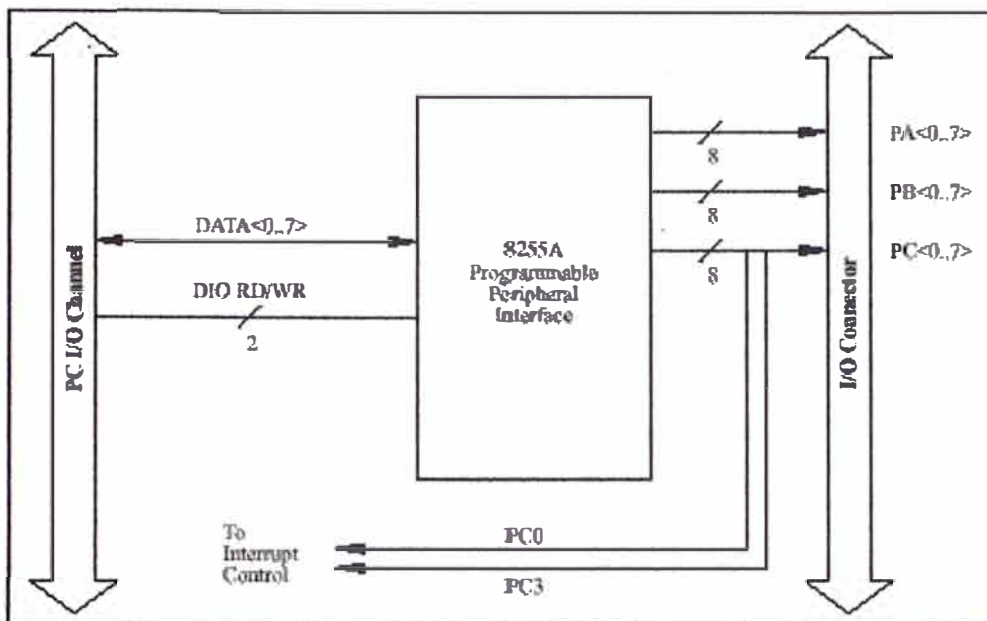


Figura 6.13: Diagrama de bloques del circuito de entrada/salida digital de la tarjeta Lab_PC + .

2. Se detecta el flanco de subida del reloj (clock) y se lee el primer dato de la posición (posición inicial), dándose inicio al algoritmo de control.
3. Se comprueba que la variable k sea menor al número de puntos n . La variable n junto con la frecuencia de muestreo determinan el tiempo de duración del experimento.
4. Se detecta un flanco de subida del reloj, con lo que se inicia un periodo de adquisición de datos y control. Se lee la posición y , y luego se estima la velocidad a partir de ella.
5. Se generan las señales de control u_1 y u_2 .
6. Se compensan las no linealidades, o sea, las tensiones de las fricciones estática y de coulomb v , se aplican restricciones en la señal de control, y luego se envían las señales $u_1 = u_1 + v + \text{offset}$ por el DAC0 y $u_2 = u_2 + v + \text{offset}$ por el DAC1 en dirección de los respectivos generadores PWM.

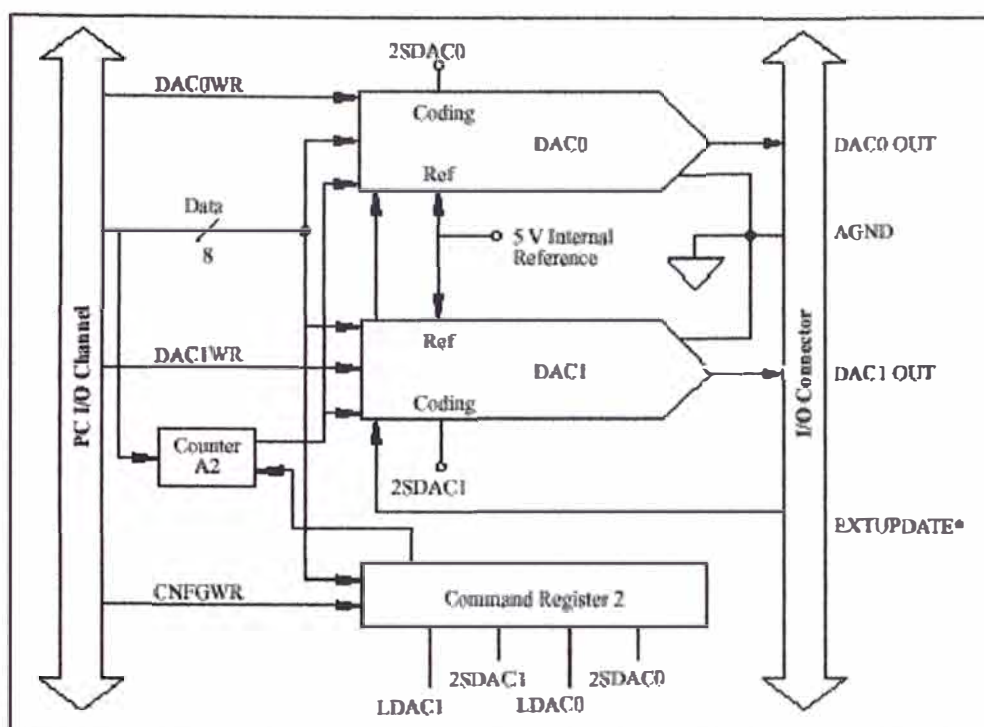


Figura 6.14: Diagrama de bloques del circuito de salida analógica de la tarjeta Lab_PC + .

7. Se actualizan las variables y se repite el lazo de control hasta que k alcance a n .

Los archivos principales del control óptimo, tanto continuo *srm4opt.cpp* como discreto *srm4optd.cpp*, presentan la siguiente estructura:

1. Declaración de Archivos de Cabecera:

Archivo de cabecera	Descripción
<i>r2llib.h</i>	definido por el usuario
<i>stdio.h</i>	para E/S estándar
<i>math.h</i>	para manejar funciones matemáticas
<i>asserth.h</i>	declara las macros de depuración
<i>alloc.h</i>	función de asignación dinámica de memoria
<i>conio.h</i>	funciones de consola y puertos de E/S

2. Declaración de constantes globales.
3. Declaración de las funciones propias del programa:

Inicializa_Sistema, Mide_Salidas, Aplica_Control1 y Aplica_Control2.

4. Declaración de las variables globales:

y1, y2, y1ant, y2ant, a1, a2, a3, b1, b2, b3, etc.

5. Definición de funciones propias del programa:

Inicializa_Sistema, Mide_Salidas, Aplica_Control1, y Aplica_Control2.

6. Definición de la función principal *main*.

La implementación del algoritmo de control secuencial del motor PAP escrito en código C, se muestra a continuación (en líneas generales):

```
do{
    opcion = getche();
    switch(opcion)
    {
        case'D': /*MOVIMIENTO DE APERTURA DE LA PINZA (GIRO A DERECHA) */
            for(count=0; count<=pas; count++)
            {
                do{
                    outportb(puerto, 0x0A);
                    delay(8);
                    outportb(puerto, 0x06);
                    delay(8);
                    outportb(puerto, 0x05);
                    delay(8);
                    outportb(puerto, 0x09);
                    delay(8); break;
                }while(count<=pas);
            } break;
        case'I': /* MOVIMIENTO DE CIERRE DE LA PINZA (GIRO A IZQUIERDA) */
            for(count=0; count<=pas; count++)
            {
                do{
                    outportb(puerto, 0x0A);
                    delay(8);
                    outportb(puerto, 0x09);
                    delay(8);
                    outportb(puerto, 0x05);
                    delay(8);
                    outportb(puerto, 0x06);
                    delay(8); break;
                }while(count<=pas);
            } break;
    }
}
```

Efectuando la siguiente asignación: $pas = \text{No. de pasos}$, $ang/pas = \text{ángulo por paso}$, y $ang_giro = \text{posición angular (en grados)}$, podemos calcular la posición

angular aplicando la siguiente fórmula:

$$ang_giro = pas \times ang/pas$$

Si por ejemplo, $ang/pas = 7.5^\circ/pas$ y la posición angular requerida sea de 150° , entonces, el número de pasos que hay que considerar será de $pas = 20$ pasos.

Las opciones de cierre y apertura de la pinza se consiguen presionando la tecla *I* para cierre y la tecla *D* para apertura. En cada una de las opciones se produce una secuencia de códigos binarios durante el tiempo en que demore cumplirse el número de pasos requerido para girar un ángulo determinado. En el programa, el número de pasos está dado por la variable *pas*. Es importante mencionar que entre paso y paso (entre código y código) se debe considerar un retardo de 8 mseg, con el que se adquiere una velocidad de 8 mseg. por paso.

En la figura 6.16 se muestra el diagrama de flujo del algoritmo de control secuencial para el movimiento de cierre y apertura de la pinza.

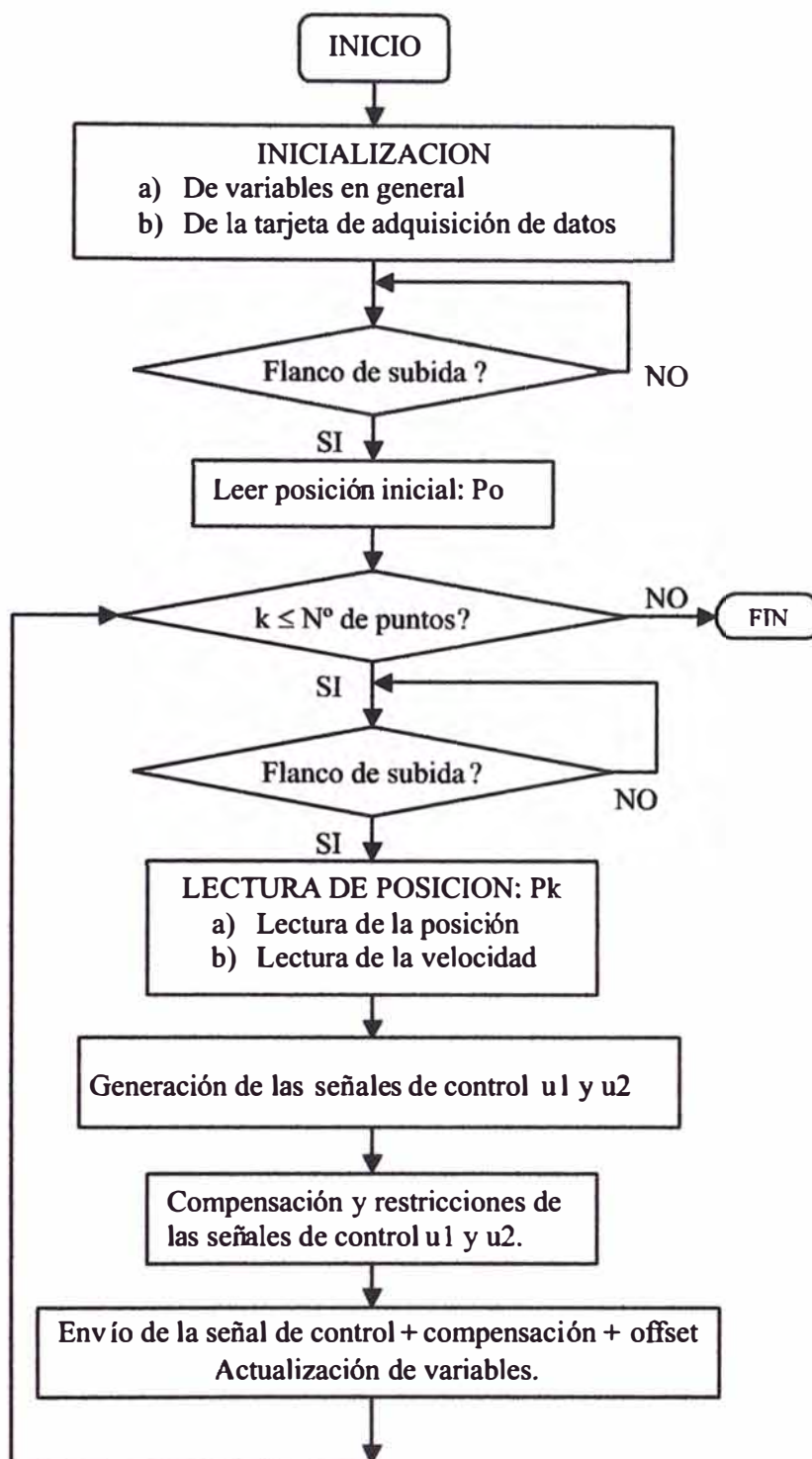


Figura 6.15: Diagrama de flujo del algoritmo de control óptimo.

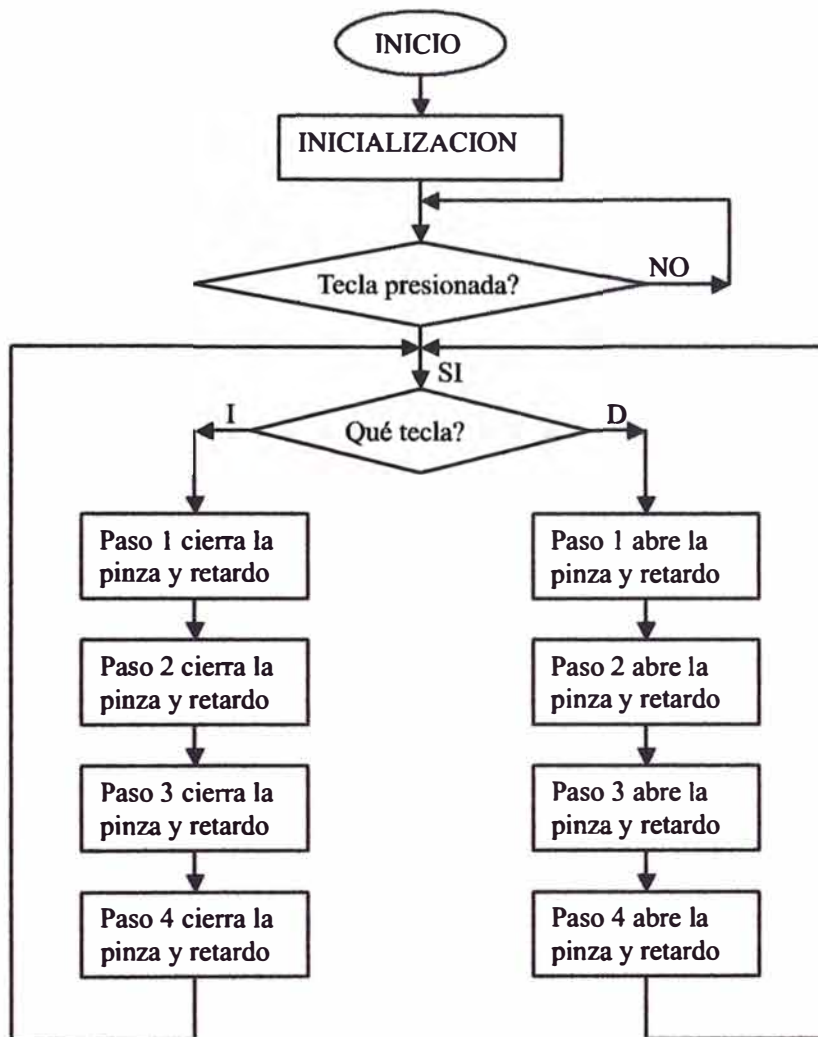


Figura 6.16: Diagrama de flujo del algoritmo de control secuencial de la pinza.

CAPÍTULO VII

EVALUACIÓN ECONÓMICA DEL PROYECTO

7.1 Costo de los Componentes

En este capítulo se especifican los costos de los elementos activos y pasivos de las tarjetas de interfaz de potencia y tarjeta sensora, así como también los costos de diseño, construcción y montaje del sistema. Debido a que el sistema es MIMO (con dos entradas y dos salidas), se utilizan dos tarjetas de interfaz de potencia, cada una conteniendo un "Generador PWM + Amplificador de potencia", y una tarjeta sensora conteniendo dos unidades similares de "decodificador + contador UP/Down" (ver figura 6.1).

Las tablas 7.1 y 7.2 muestran la descripción, cantidad y costo por unidad de cada uno de los componentes de una tarjeta de interfaz de potencia ("Generador PWM + Amplificador de potencia"). Como se dijo, en el proyecto se usan dos tarjetas idénticas.

Tabla 7.1: Circuito Generador PWM con tiempo muerto.

Descripción	Valor	Cant.	S/. Costo unit.	S/. Costo tot.
Resistencia	$4.7k\Omega$	01	0.05	0.05
Resistencia	330Ω	01	0.05	0.05
Potenciómetro	$1k\Omega$	02	12.40	12.40
Cond, Cerámico	$0.1\mu F$	02	0.30	0.60
Cond. Cerámico	$0.015\mu F$	01	0.30	0.30
Cond. Cerámico	$9nF$	01	0.30	0.30
Cond. Cerámico	$1.8nF$	01	0.30	0.30
Compuerta OR	74LS32	02	2.0	4.0
Gen. PWM	LM1524	01	15.0	15.0
Compuerta NOT	74LS02	01	2.0	2.0
Monoestable	74LS123	02	4.0	8.0

Tabla 7.2: Circuito Amplificador de Potencia.

Descripción	Valor	Cant.	S/. Costo unit.	S/. Costo tot.
Resistencia	$3.3k\Omega$	06	0.1	0.6
Resistencia	500Ω	02	0.1	0.2
Resistencia	$4.7k\Omega$	04	0.1	0.4
Resistencia	$2k\Omega$	02	0.1	0.2
Resistencia	30Ω	06	0.1	0.6
Cond. Cerámico	$22\mu F$	03	0.4	1.2
Transistor	2N2222	06	0.5	3.0
Transistor	2N2907	04	0.5	2.0
Diodos	1N4148	02	0.2	0.4
Mosfet	BUZ80	04	9.0	36.0
Compuerta NOT	4049	01	2.0	2.0

El costo de los componentes de las tarjetas de interfaz de potencia hacen un monto de S/. 89.00 por tarjeta, haciendo un total de S/. 178.00.

La tabla 7.3 muestra la descripción, cantidad y costo por unidad de cada uno de los componentes de la tarjeta sensora dual ("decodificador + contador UP/DOWN").

Tabla 7.3: Circuito Sensor: decodificador + contador UP/DOWN.

Descripción	Valor	Cant.	S/. Costo unit.	S/. Costo tot.
Resistencia	$270k\Omega$	02	0.05	0.10
Cond. Cerámico	$0.1\mu F$	02	0.4	0.80
Buffer	74HC244	04	2.80	11.20
Decoder	LS7083	02	15.00	30.00
Contador	74LS193	08	3.20	25.60

El costo de los componentes de la tarjeta sensora hacen un monto de S/. 67.70. Por consiguiente, el costo de las dos tarjetas de interfaz de potencia y de la tarjeta sensora suman S/. 245.70.

7.2 Costos de Diseño y Construcción

El costo de implementación considera lo siguiente:

- a) El proceso de elaboración de las dos tarjetas de interfaz de potencia y la tarjeta sensora: S/. 100.00.

- b) Diseño y construcción del Robot Esférico (incluye los dos servomotores):
S/. 600.00.
- c) Otros: S/. 100.00

El costo total resulta de la suma de los costos anteriores, haciendo un total de S/. 1045.70. Evidentemente este costo corresponde al prototipo; sin embargo, este precio puede bajar razonablemente si es que se fabrican en serie.

CONCLUSIONES Y RECOMENDACIONES

En esta tesis se han aplicado las estrategias de Control Optimo Proporcional Integral Continuo y Discreto, para el control de posición de un manipulador robótico esférico de dos grados de libertad, cuyos resultados gráficos fueron obtenidos en los capítulos IV (sección 4.7) y V (sección 5.8), respectivamente.

Los resultados gráficos del Control Optimo Proporcional Integral Continuo, cuyo algoritmo fue discretizado en la fase final para su ejecución en una PC digital, usando para ello una frecuencia de muestreo de 700 Hz., se muestran en:

- Las figuras 4.6 y 4.7 para trayectorias
- Las figuras 4.8 y 4.9 para referencias fijas

y los resultados para el Control Optimo Proporcional Integral Discreto, usando una frecuencia de muestreo de 200 Hz., se muestran en:

- Las figuras 5.5 y 5.6 para trayectorias
- Las figuras 5.7 y 5.8 para referencias fijas

Antes de emitir un resultado sobre la versión más apropiada de control óptimo, es conveniente presentar el siguiente cuadro de características de respuestas para ambas estrategias:

Características de respuestas de las estrategias de control óptimo.

Cnt.Opt.	Refer.	Tiempo de asent.	Sobrepico	Tensión de control
Continuo	<i>Trayec.</i>	0.7 seg.	50 %(para x_1)	$0.2 V_{max}$ (para u_1)
Continuo	<i>Trayec.</i>	1.5 seg.	60 %(para x_2)	$2.74 V_{max}$ (para u_2)
Continuo	<i>Const.</i>	0.8 seg.(para x_1)	12 %	$0.75 V_{max}$ (para u_1)
Continuo	<i>Const.</i>	4.0 seg.(para x_2)	67 %	$3.6 V_{max}$ (para u_2)
Discreto	<i>Trayec.</i>	0.01 seg.	0 %(para x_1)	$0.022 V_{max}$ (para u_1)
Discreto	<i>Trayec.</i>	0.5 seg.	5 %(para x_2)	$1.2 V_{max}$ (para u_2)
Discreto	<i>Const.</i>	0.75 seg.(para x_1)	2 %	$1.2 V_{max}$ (para u_1)
Discreto	<i>Const.</i>	0.75 seg.(para x_2)	20 %	$1.4 V_{max}$ (para u_2)

Del cuadro en referencia, podemos concluir que la estrategia de Control Optimo Proporcional Integral Discreto es el que presenta un mejor resultado tanto para trayectorias como para referencias constantes, con la ventaja adicional de trabajar a una frecuencia de muestreo de 200 Hz (3.5 veces menor que en la versión continua), lo que permite optimizar memoria en la generación de la data, para la obtención de los resultados gráficos de tiempo real.

Para futuros trabajos, se recomienda aplicar las estrategias de Control Adaptivo Autosintonizado y Control Difuso (Fuzzy) al proceso no lineal, que permitan observar los desempeños correspondientes.

APENDICE A

LISTADO DE PROGRAMAS

A continuación se presentan los listados de los programas en código **MATLAB** y en código **C** correspondientes a la simulación y a la propuesta de implementación en tiempo real, respectivamente.

Programas en Código Matlab

1. Programa de Control Optimo Continuo para Referencia Constante

```
% ARCHIVO : srm4optc.m
% PROCESO : ROBOT MANIPULADOR CILINDRICO DE DOS GRADOS DE
% LIBERTAD CON MUÑECA, CON MANO Y UN PESO
% SIMULACION CON CONTROL OPTIMO ROBUSTO
% M.Sc. RAUL BENITES SARAVIA

clear all; % close all clc

% DATOS DEL SISTEMA
% SUB-SISTEMA MECANICO
% SOLIDO Y DISCO
h=0.01; R=0.07; %ALTURA Y RADIO
% M_DISCO_TOTAL=M_DISCO+M_BOCINA=0.3+0.25
m_dt=0.55;
% PRISMA
b=0.21; %ALTURA
a=0.044; %ANCHO Y LARGO DE LA SECCION DEL PRISMA
%M_PRISMA_TOTAL=M_PRISMA+M_MOTOR 2+M_ACCESORIOS=1+1.235+0.1+0.065+0.075
m_pt=2.475;
% INERCIA DEL SOLIDO MAS MOTOR DC 2
I1=0.5*(m_pt*(a^2)/6 + m_dt*(R^2)/2) + 0.0002;
%VARILLA
m2=0.9; %MASA VARILLA+MASA MUÑECA+MASA MANO+MASA OBJETO=
% 0.1+0.2+0.4+0.2
L2=0.3; % LONGITUD TOTAL
Lx2=0.14; %LONG. DEL PTO. DE REFER. 2 AL CENTRO DE GRAVEDAD
I2=m2*(L2^2)/3+0.00009+0.00009+0.0001; %INERCIA
g =9.81;
% DATOS DEL SUB-SISTEMA ELECTRICO
KK1=0.0437; Kt1=0.0437; La1=1.5e-3; Ra1=3.43; Ja1=1.4628e-6;
```

```

n1=18.5; Ka1=14.07;
KK2=0.0437; Kt2=0.0437; La2=1.5e-3; Ra2=3.43; Ja2=1.2055e-6;
n2=18.5; Ka2=14.07;
bm1=2.0143e-7; bm2=5.6877e-8;
% CARGA
JL1=5.63e-5; bL1=7.05e-5; JL2=JL1; bL2=bL1;
% EQUIVALENCIA DE J y b
J1=n1^2*Ja1 + JL1;
b1=n1^2*bm1 +bL1;
%J2=J1;
%b2=b1;
J2=n2^2*Ja2 + JL2;
b2=n2^2*bm2 +bL2;
% EQUIVALENCIA DE INDUCTANCIA
Lt1=La1/(n1*KK1*Ka1);
Lt2=La2/(n2*KK2*Ka2);
% EQUIVALENCIA DE RESISTENCIAS
Rt1=Ra1/(n1*KK1*Ka1);
Rt2=Ra2/(n2*KK2*Ka2);
% EQUIVALENCIA DE N
Nt1=n1*Kt1/Ka1;
Nt2=n2*Kt2/Ka2;
% SIMPLIFICACION DE PARAMETROS DEL ROBOT
a1=2*I1;
a2=I2 + 2*m2*Lx2*L2 - m2*L2^2;
a3=m2*g*Lx2;
% MODELO LINEALIZADO EQUIVALENTE DEL PROCESO DE 4to ORDEN (CONTINUO)
A=[0 0 1 0; 0 0 0 1; 0 0 -(Rt1*b1+Nt1)/(Rt1*(J1+a1+a2)) 0;...
    0 0 0 -(Rt2*b2+Nt2)/(Rt2*(J2+a2))];
B=[0 0; 0 0; 1/(Rt1*(J1+a1+a2)) 0;0 1/(Rt2*(J2+a2))];
C=[1 0 0 0; 0 1 0 0];
D=zeros(2,2);
% CALCULO DE LA CONTROLABILIDAD EN TIEMPO CONTINUO
M=[B A*B A^2*B A^3*B];
rangoM=rank(M); %RANGO DEL SISTEMA = 4
% CALCULO DE LA OBSERVABILIDAD EN TIEMPO CONTINUO
N=[C' A'*C' A'^2*C' A'^3*C'];
rangoN=rank(N);
% RANGO DEL SISTEMA = 4
% MODELO DISCRETO DEL PROCESO LINEALIZADO CONTINUO
T=0.0014; % Fs=700 Hz
%T=0.0010; % Fs=1000 Hz
%T=0.0050; % Fs=200 Hz
%T=0.0033; % Fs=300 Hz
%T=0.0025; % Fs=400 Hz
%T=0.0016; % Fs=625 Hz. se obtiene una respuesta regular con
%alfa =0.5 o 1
[Ad,Bd,Cd,Dd]=c2dm(A,B,C,D,T,'zoh');
% CALCULO DE LA CONTROLABILIDAD EN TIEMPO DISCRETO

```

```

Md=[Bd Ad*Bd Ad^2*Bd Ad^3*Bd];
rangoMd=rank(Md); %RANGO DEL SISTEMA = 4
% CALCULO DE LA OBSERVABILIDAD EN TIEMPO CONTINUO
Nd=[Cd' Ad'*Cd' Ad'^2*Cd' Ad'^3*Cd'];
rangoNd=rank(Nd); %RANGO DEL SISTEMA = 4
% CONTROLADOR POR REALIMENTACION DE ESTADOS NO LINEAL MIMO (CONTINUO)

% MATRICES DE PONDERACION
Q=1*eye(4);
%R=10*eye(2);
R=100*eye(2); % permite disminuir la magnitud de la señal de control
%Qa=10*eye(6); V=0.1*eye(4); W=0.1*eye(2); G1=eye(4);
Qa=100*eye(6); V=0.1*eye(4); W=0.1*eye(2); G1=eye(4); % mejora la
%respuesta
%alfa=5; % sin considerar restricciones de la señal de control
%(respuesta no tan buena)
%alfa=2; % considerando restricciones de la señal de control
alfa=10; % para respuesta a referencia constante, sin restricciones
% de la señal de control (mejor respuesta que para alfa=5)
[D,K]=lqr((A+alfa*eye(4)),B,Q,R);
% MEJORA DEL RENDIMIENTO DEL CONTROLADOR ADICIONANDO UN INTEGRADOR
% MATRICES AUMENTADAS
Aa=[A zeros(4,2); C zeros(2,2)]; % matriz (n+2) x (n+2): 6 x 6
Ba=[B; zeros(2,2)]; % matriz (n+2) x m : 6 x 2
Ca=[C zeros(2,2)];
Da=zeros(2);
% CALCULO DE LAS GANACIAS Ga y Ka
[Da,Ka]=lqr((Aa+alfa*eye(6)),Ba,Qa,R);
% OBSERVADOR DE ESTADOS NO LINEAL MIMO
% xep=q*x+g*u-E*(ye-y) xe(0)=0;
% ye=h*x;
% matriz E es de orden nxp (4x2)
[E,sum,Eo]=lqe((A+alfa*eye(4)),G1,C,V,W);

% CONDICIONES INICIALES
u1(1)=0.0; u2(1)=0.1;
u1(2)=0.0; u2(2)=0.1;
u1(3)=0.0; u2(3)=0.1;
x1(1)=0.5; x2(1)=-0.5; x3(1)=0.0; x4(1)=0.0;
x1(2)=0.5; x2(2)=-0.5; x3(2)=0.0; x4(2)=0.0;
x1(3)=0.5; x2(3)=-0.5; x3(3)=0.0; x4(3)=0.0;
xe1(1)=0.0; xe2(1)=0.0; xe3(1)=0.0; xe4(1)=0.0;
xe1(2)=0.0; xe2(2)=0.0; xe3(2)=0.0; xe4(2)=0.0;
xe1(3)=0.0; xe2(3)=0.0; xe3(3)=0.0; xe4(3)=0.0;
z1(1)=0.0; z2(1)=0.0;
z1(2)=0.0; z2(2)=0.0;
z1(3)=0.0; z2(3)=0.0;
MM=4000; % número de muestras

```

```

% BUCLE DE CONTROL
for k=2:MM;

% VALORES DE PARAMETROS DEL PROCESO NO LINEAL DISCRETIZADO
p33(k)=Rt1*(b1-2*a2*x4(k)*sin(x2(k))*cos(x2(k)))+Nt1;
pa(k)=Rt1*(J1+a1+a2*cos(x2(k))*cos(x2(k)));
pb(k)=Rt2*(J2+a2);
q42(k)=(Rt2*a3)/(Rt2*(J2+a2));
q43(k)=(Rt2*a2*x3(k)*sin(x2(k))*cos(x2(k)))/(Rt2*(J2+a2));
q44(k)=(Rt2*b2+Nt2)/(Rt2*(J2+a2));
g3(k)=1/pa(k);    g4(k)=1/pb(k);
% PROCESO NO LINEAL DISCRETIZADO
x1(k+1)=x1(k)+T*x3(k);
x2(k+1)=x2(k)+T*x4(k);
x3(k+1)=x3(k)+T*[-p33(k)*g3(k)*x3(k)+g3(k)*u1(k)];
x4(k+1)=x4(k)+T*[-q42(k)*cos(x2(k))-q43(k)*x3(k)-q44(k)*x4(k)+ ...
g4(k)*u2(k)];

%r1(k+1)=sin(0.0014*k)+ 0.00014*k; r2(k+1)=cos(0.0014*k);
r1(k+1)=1; r2(k+1)=1.5;
e1(k+1)=-r1(k+1)+x1(k+1); e2(k+1)=-r2(k+1)+x2(k+1);

% OBSERVADOR NO LINEAL DISCRETIZADO DE 4TO. ORDEN
xe1(k+2)=xe1(k+1)+T*[xe3(k+1)-E(1,1)*(xe1(k+1)-e1(k+1))- ...
E(1,2)*(xe2(k+1)-e2(k+1))];
xe2(k+2)=xe2(k+1)+T*[xe4(k+1)-E(2,1)*(xe1(k+1)-e1(k+1))- ...
E(2,2)*(xe2(k+1)-e2(k+1))];
xe3(k+2)=xe3(k+1)+T*[-p33(k)*g3(k)*xe3(k+1)+g3(k)*u1(k)- ...
E(3,1)*(xe1(k+1)-e1(k+1))-E(3,2)*(xe2(k+1)-e2(k+1))];
xe4(k+2)=xe4(k+1)+T*[-q42(k)*cos(xe2(k+1))-q43(k)*xe3(k+1) - ...
q44(k)*xe4(k+1)+g4(k)*u2(k)-E(4,1)*(xe1(k+1)-e1(k+1))- ...
E(4,2)*(xe2(k+1)-e2(k+1))];

% LEY DE CONTROL DISCRETIZADO u1(K) Y u2(k)
z1(k+2)=z1(k+1)+T*e1(k+1);    z2(k+2)=z2(k+1)+T*e2(k+1);
u1(k+1)=-Da(1,1)*xe1(k+2)-Da(1,2)*xe2(k+2)-Da(1,3)*xe3(k+2)- ...
Da(1,4)*xe4(k+2)-Da(1,5)*z1(k+2)-Da(1,6)*z2(k+2);
u2(k+1)=-Da(2,1)*xe1(k+2)-Da(2,2)*xe2(k+2)-Da(2,3)*xe3(k+2)- ...
Da(2,4)*xe4(k+2)-Da(2,5)*z1(k+2)-Da(2,6)*z2(k+2);

% limitando control

%fu = 1.4;
%if(u1(k+1) > 0 & u1(k+1) > fu)
% u1(k+1) = fu;
%elseif(u1(k+1) < 0 & u1(k+1) < -fu)
% u1(k+1) = -fu;
%end

```

```

%fu = 1.4;
%if(u2(k+1) > 0 & u2(k+1) > fu)
% u2(k+1) = fu;
%elseif(u2(k+1) < 0 & u2(k+1) < -fu)
% u2(k+1) = -fu;
%end

end

ejex = linspace(0,MM*T,MM-1);
figure(1)
subplot(2,1,1)
plot(ejex,r1(2:MM),ejex,x1(2:MM),'--'); grid on
title('Control Optimo Robusto para un Robot de 2 G.L.')
```

ylabel('Posicion x1 (radianes)')

xlabel('Tiempo : segundos')

text(1.2,0.4,'- r1(t)')

text(1.2,0.1,'- - : x1(t)')

subplot(2,1,2)

plot(ejex,u1(2:MM)), grid on

ylabel('Control u1 (voltios)')

xlabel('Tiempo : segundos')

% print -f -deps srm4opt2a

```

figure(2)
subplot(2,1,1)
plot(ejex,r2(2:MM),ejex,x2(2:MM),'--'); grid on
title('Control Optimo Robusto para un Robot de 2 G.L.')
```

ylabel('Posicion x2 (radianes)')

xlabel('Tiempo : segundos')

text(2.2,0.6,'- r2(t)')

text(2.2,0.2,'- - : x2(t)')

subplot(2,1,2)

plot(ejex,u2(2:MM)), grid on

ylabel('Control u2 (voltios)')

xlabel('Tiempo : segundos')

% print -f -deps srm4optc

2. Programa de Control Optimo Continuo para Referencia Variante (trayectoria)

```

% ARCHIVO : srm4opt.m
% PROCESO : ROBOT MANIPULADOR ESFERICO DE DOS GRADOS DE
% LIBERTAD CON MUÑECA, CON MANO Y UN PESO
% SIMULACION CON CONTROL OPTIMO ROBUSTO
% M.Sc. RAUL BENITES SARAIVIA

clear all; % close all clc
```

```

% DATOS DEL SISTEMA
% SUB-SISTEMA MECANICO
% SOLIDO Y DISCO
  h=0.01; R=0.07; %ALTURA Y RADIO
% M_DISCO_TOTAL=M_DISCO+M_BOCINA=0.3+0.25
  m_dt=0.55;
% PRISMA
  b=0.21; %ALTURA
  a=0.044; %ANCHO Y LARGO DE LA SECCION DEL PRISMA
%M_PRISMA_TOTAL=M_PRISMA+M_MOTOR 2+M_ACCESORIOS=
%1+1.235+0.1+0.065+0.075
  m_pt=2.475;
% INERCIA DEL SOLIDO MAS MOTOR DC 2
  I1=0.5*(m_pt*(a^2)/6 + m_dt*(R^2)/2) + 0.0002;
%VARILLA
  m2=0.9; %MASA VARILLA+MASA MUÑECA+MASA MANO+MASA
  %OBJETO=0.1+0.2+0.4+0.2
  L2=0.3; % LONGITUD TOTAL
  Lx2=0.14; %LONG. DEL PTO. DE REFER. 2 AL CENTRO DE GRAVEDAD
  I2=m2*(L2^2)/3+0.00009+0.00009+0.0001; %INERCIA
  g =9.81;
% DATOS DEL SUB-SISTEMA ELECTRICO
  KK1=0.0437; Kt1=0.0437; La1=1.5e-3; Ra1=3.43; Ja1=1.4628e-6;
  n1=18.5; Ka1=14.07;
  KK2=0.0437; Kt2=0.0437; La2=1.5e-3; Ra2=3.43; Ja2=1.2055e-6;
  n2=18.5; Ka2=14.07;
  bm1=2.0143e-7; bm2=5.6877e-8;
% CARGA
  JL1=5.63e-5; bL1=7.05e-5; JL2=JL1; bL2=bL1;
% EQUIVALENCIA DE J y b
  J1=n1^2*Ja1 + JL1;
  b1=n1^2*bm1 +bL1;
  %J2=J1;
  %b2=b1;
  J2=n2^2*Ja2 + JL2;
  b2=n2^2*bm2 +bL2;
% EQUIVALENCIA DE INDUCTANCIA
  Lt1=La1/(n1*KK1*Ka1);
  Lt2=La2/(n2*KK2*Ka2);
% EQUIVALENCIA DE RESISTENCIAS
  Rt1=Ra1/(n1*KK1*Ka1);
  Rt2=Ra2/(n2*KK2*Ka2);
% EQUIVALENCIA DE N
  Nt1=n1*Kt1/Ka1;
  Nt2=n2*Kt2/Ka2;
% SIMPLIFICACION DE PARAMETROS DEL ROBOT
  a1=2*I1;
  a2=I2 + 2*m2*Lx2*L2 - m2*L2^2;
  a3=m2*g*Lx2;

```

```

% MODELO LINEALIZADO EQUIVALENTE DEL PROCESO DE 4to ORDEN (CONTINUO)
A=[0 0 1 0; 0 0 0 1; 0 0 -(Rt1*b1+Nt1)/(Rt1*(J1+a1+a2)) 0;...
    0 0 0 -(Rt2*b2+Nt2)/(Rt2*(J2+a2))];
B=[0 0; 0 0; 1/(Rt1*(J1+a1+a2)) 0;0 1/(Rt2*(J2+a2))];
C=[1 0 0 0; 0 1 0 0];
D=zeros(2,2);
% CALCULO DE LA CONTROLABILIDAD EN TIEMPO CONTINUO
M=[B A*B A^2*B A^3*B];
rangoM=rank(M); %RANGO DEL SISTEMA = 4
% CALCULO DE LA OBSERVABILIDAD EN TIEMPO CONTINUO
N=[C' A'*C' A'^2*C' A'^3*C'];
rangoN=rank(N);
% RANGO DEL SISTEMA = 4
% MODELO DISCRETO DEL PROCESO LINEALIZADO CONTINUO
T=0.0014; % Fs=700 Hz
%T=0.0010; % Fs=1000 Hz
%T=0.0050; % Fs=200 Hz
%T=0.0033; % Fs=300 Hz
%T=0.0025; % Fs=400 Hz
%T=0.0016; % Fs=625 Hz. se obtiene una respuesta regular con
%alfa =0.5 o 1
[Ad,Bd,Cd,Dd]=c2dm(A,B,C,D,T,'zoh');
% CALCULO DE LA CONTROLABILIDAD EN TIEMPO DISCRETO
Md=[Bd Ad*Bd Ad^2*Bd Ad^3*Bd];
rangoMd=rank(Md); %RANGO DEL SISTEMA = 4
% CALCULO DE LA OBSERVABILIDAD EN TIEMPO CONTINUO
Nd=[Cd' Ad'*Cd' Ad'^2*Cd' Ad'^3*Cd'];
rangoNd=rank(Nd); %RANGO DEL SISTEMA = 4
% CONTROLADOR POR REALIMENTACION DE ESTADOS NO LINEAL MIMO (CONTINUO)

% MATRICES DE PONDERACION
Q=1*eye(4);
R=10*eye(2);
%R=100*eye(2); % permite disminuir la magnitud de la señal de control
Qa=10*eye(6); V=0.1*eye(4); W=0.1*eye(2); G1=eye(4);
alfa=5; % sin considerar restricciones de la señal de control
%alfa=2; % considerando restricciones de la señal de control
%alfa=10; % para respuesta a referencia constante, sin restricciones
% de la señal de control
[D,K]=lqr((A+alfa*eye(4)),B,Q,R);
% MEJORA DEL RENDIMIENTO DEL CONTROLADOR ADICIONANDO UN INTEGRADOR
% MATRICES AUMENTADAS
Aa=[A zeros(4,2); C zeros(2,2)]; % matriz (n+2) x (n+2): 6 x 6
Ba=[B; zeros(2,2)]; % matriz (n+2) x m : 6 x 2
Ca=[C zeros(2,2)];
Da=zeros(2);
% CALCULO DE LAS GANACIAS Da y Ka
[Da,Ka]=lqr((Aa+alfa*eye(6)),Ba,Qa,R);
% OBSERVADOR DE ESTADOS NO LINEAL MIMO

```

```

% xep=q*xe+g*u-E*(ye-y)      xe(0)=0;
% ye=h*xe;
% matriz E es de orden npx (4x2)
  [E,sum,Eo]=lqe((A+alfa*eye(4)),G1,C,V,W);

% CONDICIONES INICIALES
u1(1)=0.0; u2(1)=0.1;
u1(2)=0.0; u2(2)=0.1;
u1(3)=0.0; u2(3)=0.1;
x1(1)=0.5; x2(1)=-0.5; x3(1)=0.0; x4(1)=0.0;
x1(2)=0.5; x2(2)=-0.5; x3(2)=0.0; x4(2)=0.0;
x1(3)=0.5; x2(3)=-0.5; x3(3)=0.0; x4(3)=0.0;
xe1(1)=0.0; xe2(1)=0.0; xe3(1)=0.0; xe4(1)=0.0;
xe1(2)=0.0; xe2(2)=0.0; xe3(2)=0.0; xe4(2)=0.0;
xe1(3)=0.0; xe2(3)=0.0; xe3(3)=0.0; xe4(3)=0.0;
z1(1)=0.0; z2(1)=0.0;
z1(2)=0.0; z2(2)=0.0;
z1(3)=0.0; z2(3)=0.0;
MM=4000; % número de muestras

% BUCLE DE CONTROL
for k=2:MM;

% VALORES DE PARAMETROS DEL PROCESO NO LINEAL DISCRETIZADO
p33(k)=Rt1*(b1-2*a2*x4(k)*sin(x2(k))*cos(x2(k)))+Nt1;
pa(k)=Rt1*(J1+a1+a2*cos(x2(k))*cos(x2(k)));
pb(k)=Rt2*(J2+a2);
q42(k)=(Rt2*a3)/(Rt2*(J2+a2));
q43(k)=(Rt2*a2*x3(k)*sin(x2(k))*cos(x2(k)))/(Rt2*(J2+a2));
q44(k)=(Rt2*b2+Nt2)/(Rt2*(J2+a2));
g3(k)=1/pa(k); g4(k)=1/pb(k);
% PROCESO NO LINEAL DISCRETIZADO
x1(k+1)=x1(k)+T*x3(k);
x2(k+1)=x2(k)+T*x4(k);
x3(k+1)=x3(k)+T*[-p33(k)*g3(k)*x3(k)+g3(k)*u1(k)];
x4(k+1)=x4(k)+T*[-q42(k)*cos(x2(k))-q43(k)*x3(k)-q44(k)*x4(k)+
g4(k)*u2(k)];

r1(k+1)=sin(0.0014*k)+ 0.00014*k; r2(k+1)=cos(0.0014*k);
%r1(k+1)=1; r2(k+1)=1.5;
e1(k+1)=-r1(k+1)+x1(k+1); e2(k+1)=-r2(k+1)+x2(k+1);

% OBSERVADOR NO LINEAL DISCRETIZADO DE 4TO. ORDEN
xe1(k+2)=xe1(k+1)+T*[xe3(k+1)-E(1,1)*(xe1(k+1)-e1(k+1))-
E(1,2)*(xe2(k+1)-e2(k+1))];
xe2(k+2)=xe2(k+1)+T*[xe4(k+1)-E(2,1)*(xe1(k+1)-e1(k+1))-
E(2,2)*(xe2(k+1)-e2(k+1))];
xe3(k+2)=xe3(k+1)+T*[-p33(k)*g3(k)*xe3(k+1)+g3(k)*u1(k)-
E(3,1)*(xe1(k+1)-e1(k+1))-E(3,2)*(xe2(k+1)-e2(k+1))];

```



```

xe4(k+2)=xe4(k+1)+T*[-q42(k)*cos(xe2(k+1))-q43(k)*xe3(k+1) - ...
q44(k)*xe4(k+1)+g4(k)*u2(k)-E(4,1)*(xe1(k+1)-e1(k+1))- ...
E(4,2)*(xe2(k+1)-e2(k+1))];

% LEY DE CONTROL DISCRETIZADO u1(K) Y u2(k)
z1(k+2)=z1(k+1)+T*e1(k+1);    z2(k+2)=z2(k+1)+T*e2(k+1);
u1(k+1)=-Da(1,1)*xe1(k+2)-Da(1,2)*xe2(k+2)-Da(1,3)*xe3(k+2)- ...
        Da(1,4)*xe4(k+2)-Da(1,5)*z1(k+2)-Da(1,6)*z2(k+2);
u2(k+1)=-Da(2,1)*xe1(k+2)-Da(2,2)*xe2(k+2)-Da(2,3)*xe3(k+2)- ...
        Da(2,4)*xe4(k+2)-Da(2,5)*z1(k+2)-Da(2,6)*z2(k+2);

% limitando control

%fu = 1.4;
%if(u1(k+1) > 0 & u1(k+1) > fu)
% u1(k+1) = fu;
%elseif(u1(k+1) < 0 & u1(k+1) < -fu)
% u1(k+1) = -fu;
%end

%fu = 1.4;
%if(u2(k+1) > 0 & u2(k+1) > fu)
% u2(k+1) = fu;
%elseif(u2(k+1) < 0 & u2(k+1) < -fu)
% u2(k+1) = -fu;
%end

end

ejex = linspace(0,MM*T,MM-1);
figure(1)
subplot(2,1,1)
plot(ejex,r1(2:MM),ejex,x1(2:MM),'--'); grid on
title('Control Optimo Robusto para un Robot de 2 G.L.')
ylabel('Posicion x1 (radianes)')
xlabel('Tiempo : segundos')
text(1.2,0.4,'- : r1(t)')
text(1.2,0.1,'- - : x1(t)')
subplot(2,1,2)
plot(ejex,u1(2:MM)), grid on
ylabel('Control u1 (voltios)')
xlabel('Tiempo : segundos')
% print -f -deps srm4opt

figure(2)
subplot(2,1,1)
plot(ejex,r2(2:MM),ejex,x2(2:MM),'--'); grid on
title('Control Optimo Robusto para un Robot de 2 G.L.')
ylabel('Posicion x2 (radianes)')

```

```

xlabel('Tiempo : segundos')
text(2.2,0.6,'-      r2(t)')
text(2.2,0.2,'- - : x2(t)')
subplot(2,1,2)
plot(ejex,u2(2:MM)), grid on
ylabel('Control u2 (voltios)')
xlabel('Tiempo : segundos')
% print -f -deps srm4opt

```

3. Programa de Control Optimo Discreto para Referencia Constante y Variante (trayectoria)

```

% ARCHIVO : srm4optd.m
% PROCESO : ROBOT MANIPULADOR ESFERICO DE DOS GRADOS DE
% LIBERTAD CON MUÑECA, CON MANO Y UN PESO
% SIMULACION CON CONTROL OPTIMO PROPORCIONAL INTEGRAL DISCRETO
% TESIS
% M.Sc. RAUL BENITES SARAVIA

clear all; % close all clc

% DATOS DEL SISTEMA
% SUB-SISTEMA MECANICO
% SOLIDO Y DISCO
h=0.01; R=0.07; %ALTURA Y RADIO
% M_DISCO_TOTAL=M_DISCO+M_BOCINA=0.3+0.25
m_dt=0.55;
% PRISMA
b=0.21; %ALTURA
a=0.044; %ANCHO Y LARGO DE LA SECCION DEL PRISMA
%M_PRISMA_TOTAL=M_PRISMA+M_MOTOR 2+M_ACCESORIOS=1+1.235+0.1+0.065+0.075
m_pt=2.475;
% INERCIA DEL SOLIDO MAS MOTOR DC 2
I1=0.5*(m_pt*(a^2)/6 + m_dt*(R^2)/2) + 0.0002;
%VARILLA
m2=0.9; %MASA VARILLA+MASA MUÑECA+MASA MANO+MASA OBJETO
%
%                               =0.1+0.2+0.4+0.2
L2=0.3; % LONGITUD TOTAL
Lx2=0.14; %LONG. DEL PTO. DE REFER. 2 AL CENTRO DE GRAVEDAD
I2=m2*(L2^2)/3+0.00009+0.00009+0.0001; %INERCIA
g =9.81;
% DATOS DEL SUB-SISTEMA ELECTRICO
KK1=0.0437; Kt1=0.0437; La1=1.5e-3; Ra1=3.43; Ja1=1.4628e-6; n1=18.5;
Ka1=14.07;
KK2=0.0437; Kt2=0.0437; La2=1.5e-3; Ra2=3.43; Ja2=1.2055e-6; n2=18.5;
Ka2=14.07;
bm1=2.0143e-7; bm2=5.6877e-8;
% CARGA
JL1=5.63e-5; bL1=7.05e-5; JL2=JL1; bL2=bL1;

```

```

% EQUIVALENCIA DE J y b
J1=n1^2*Ja1 + JL1;
b1=n1^2*bm1 +bL1;
%J2=J1;
%b2=b1;
J2=n2^2*Ja2 + JL2;
b2=n2^2*bm2 +bL2;
% EQUIVALENCIA DE INDUCTANCIA
Lt1=La1/(n1*KK1*Ka1);
Lt2=La2/(n2*KK2*Ka2);
% EQUIVALENCIA DE RESISTENCIAS
Rt1=Ra1/(n1*KK1*Ka1);
Rt2=Ra2/(n2*KK2*Ka2);
% EQUIVALENCIA DE N
Nt1=n1*Kt1/Ka1;
Nt2=n2*Kt2/Ka2;
% SIMPLIFICACION DE PARAMETROS DEL ROBOT
a1=2*I1;
a2=I2 + 2*m2*Lx2*L2 - m2*L2^2;
a3=m2*g*Lx2;
% MODELO LINEALIZADO EQUIVALENTE DEL PROCESO DE 4to ORDEN (CONTINUO)
A=[0 0 1 0; 0 0 0 1; 0 0 -(Rt1*b1+Nt1)/(Rt1*(J1+a1+a2)) 0;...
    0 0 0 -(Rt2*b2+Nt2)/(Rt2*(J2+a2))];
B=[0 0; 0 0; 1/(Rt1*(J1+a1+a2)) 0;0 1/(Rt2*(J2+a2))];
C=[1 0 0 0; 0 1 0 0];
D=zeros(2,2);
% CALCULO DE LA CONTROLABILIDAD EN TIEMPO CONTINUO
M=[B A*B A^2*B A^3*B];
rangoM=rank(M); %RANGO DEL SISTEMA = 4
% CALCULO DE LA OBSERVABILIDAD EN TIEMPO CONTINUO
N=[C' A'*C' A'^2*C' A'^3*C'];
rangoN=rank(N);
% RANGO DEL SISTEMA = 4
% MODELO DISCRETO DEL PROCESO LINEALIZADO CONTINUO
%T=0.01; % Fs=100 Hz. Deseable para trayectorias
%T=0.0014; % Fs=700 Hz
%T=0.0010; % Fs=1000 Hz
T=0.0050; % Fs=200 Hz. Deseable para referencias constantes y tray.
%T=0.0033; % Fs=300 Hz.
%T=0.0025; % Fs=400 Hz
%T=0.0016; % Fs=625 Hz
[G,H,Cd,Dd]=c2dm(A,B,C,D,T,'zoh');
Cd1=[Cd(1,1) Cd(1,2) Cd(1,3) Cd(1,4)];
Cd2=[Cd(2,1) Cd(2,2) Cd(2,3) Cd(2,4)];
H1=[H(1,1); H(2,1); H(3,1); H(4,1)];
H2=[H(1,2); H(2,2); H(3,2); H(4,2)];

% MATRICES AMPLIADAS DEBIDO A LA INSERCIÓN DEL INTEGRADOR
% EN EL CONTROLADOR

```

```

Ga = [G zeros(4,2)
      -Cd*G eye(2,2)];
Ha = [H;-Cd*H];

% CALCULO DE LA CONTROLABILIDAD EN TIEMPO DISCRETO DEL SISTEMA AMPLIADO
Md=[Ha Ga*Ha Ga^2*Ha Ga^3*Ha Ga^4*Ha Ga^5*Ha];
rangoMd=rank(Md); %RANGO DEL SISTEMA AMPLIADO= 6
% CALCULO DE LA OBSERVABILIDAD EN TIEMPO DISCRETO DEL SISTEMA AMPLIADO
Nd=[Cd' G'*Cd' (G')^2*Cd' (G')^3*Cd'];
rangoNd=rank(Nd); %RANGO DEL SISTEMA AMPLIADO= 4

% CONTROLADOR POR REALIMENTACION DE ESTADOS NO LINEAL MIMO (DISCRETO)

% MATRICES DE PONDERACION
R=10*eye(2);
Qa=10*eye(6);
P=zeros(6,6);
for i=1:1000
P=Qa+Ga'*P*Ga-Ga'*P*Ha*inv(R+Ha'*P*Ha)*Ha'*P*Ga;
end

K1=inv(R+Ha'*P*Ha)*Ha'*P*Ga;
[KK,PP]=dlqr(Ga,Ha,Qa,R); % K1=KK, P=PP
K=[KK(1,1) KK(1,2) KK(1,3) KK(1,4)
   KK(2,1) KK(2,2) KK(2,3) KK(2,4)];
KI = - [KK(1,5) KK(1,6)
        KK(2,5) KK(2,6)];

Kc1=[KK(1,1) KK(1,2) KK(1,3) KK(1,4)];
Kc2=[KK(2,1) KK(2,2) KK(2,3) KK(2,4)];
KI1= - [KK(1,5)];
KI2= - [KK(2,6)];

% CALCULO DE LA GANANCIA DEL OBSERVADOR DE ESTADOS

% MATRICES DE PONDERACION PARA EL OBSERVADOR
Qe=100*eye(4); Re=10*eye(2);
Pe = zeros(4,4);
for i=1:200
Pe = Qe + G*Pe*G' -G*Pe*Cd'*inv(Re+Cd*Pe*Cd')*Cd*Pe*G';
end
KeT = inv(Re+Cd*Pe*Cd')*Cd*Pe*G'; Ke=KeT';
[KKeT,PPe] = dlqr(G',Cd',Qe,Re); KKe=KKeT';% Pe=PPe, Ke=KKe

% SIMULACION DEL SISTEMA DE CONTROL OPTIMO MULTIVARIABLE
% CONDICIONES INICIALES
x1=0; x2=0; x3=0; x4=0;
xe1=0; xe2=0; xe3=0; xe4=0;
xe = [xe1;xe2;xe3;xe4];

```

```

v1=0; v2=0;
v=[v1;v2];

% LAZO DE CONTROL
N = 4000; % TIEMPO EN SEGUNDOS: T*N

for k=1:N

% LEY DE CONTROL
u1=-Kc1*xe+KI1*v1;
u2=-Kc2*xe+KI2*v2;
u=[u1;u2]; % REPRESENTACION VECTORIAL

% RESTRICCIONES PARA NO SATURAR A LOS AMPLIFICADORES
if(u1 > 1.4), u1 = 1.4;
elseif(u1 < -1.4), u1 = -1.4;
end

if(u2 > 1.4), u2 = 1.4;
elseif(u2 < -1.4), u2 = -1.4;
end

% ESTADOS ESTIMADOS
xe = G*xe + H*u + Ke*([x1;x2] - Cd*xe);

% VALORES DE LOS PARAMETROS DEL PROCESO NO LINEAL
p33=Rt1*(b1-2*a2*x4*sin(x2)*cos(x2))+Nt1;
pa=Rt1*(J1+a1+a2*cos(x2)*cos(x2));
pb=Rt2*(J2+a2);
q42=(Rt2*a3)/(Rt2*(J2+a2));
q43=(Rt2*a2*x3*sin(x2)*cos(x2))/(Rt2*(J2+a2));
q44=(Rt2*b2+Nt2)/(Rt2*(J2+a2));
g3=1/pa; g4=1/pb;

% PROCESO NO LINEAL DISCRETIZADO
x1=x1+T*x3;
x2=x2+T*x4;
x3=x3+T*[-p33*g3*x3+g3*u1];
x4=x4+T*[-q42*cos(x2)-q43*x3-q44*x4+g4*u2];

% SETPOINT
%r1(k)=sin(0.0014*k)+ 0.00014*k; % TRAYECTORIA 1
%r2(k)=cos(0.0014*k); % TRAYECTORIA 2
r1(k)=1; r2(k)=1.5; % REFERENCIAS 1 Y 2 CONSTANTES

% ERROR DE SEGUIMIENTO
e1=r1(k)-x1; e2=r2(k)-x2;

% SALIDA DEL INTEGRADOR

```

```

v1 = v1 + r1(k) - x1;
v2 = v2 + r2(k) - x2;
v=[v1;v2]; % REPRESENTACION VECTORIAL

% ASIGNACION DE VARIABLES PARA PLOTEO
y1(k)=x1; y2(k)=x2;
U1(k) =u1; U2(k)=u2;

end

% PLOTEO
t = linspace(0,N*T,N);
figure(1)
subplot(2,1,1)
plot(t,r1,'-',t,y1,'--'); grid
title('Control Optimo Proporcional Integral Discreto para un Robot
de 2 G.L. ')
ylabel('Posicion x1 (radianes)')
%xlabel('Tiempo : segundos')
%text(1.2,0.4,'- : r1(t)')
%text(1.2,0.1,'- - : x1(t)')
subplot(2,1,2)
plot(t,U1), grid
ylabel('Control u1 (voltios)')
xlabel('Tiempo : segundos')
%print -f -deps optdisc3

figure(2)
subplot(2,1,1)
plot(t,r2,'-',t,y2,'--'); grid
title('Control Optimo Proporcional Integral Discreto para un Robot
de 2 G.L. ')
ylabel('Posicion x2 (radianes)')
%xlabel('Tiempo : segundos')
%text(2.2,0.6,'- : r2(t)')
%text(2.2,0.2,'- - : x2(t)')
subplot(2,1,2)
plot(t,U2), grid
ylabel('Control u2 (voltios)')
xlabel('Tiempo : segundos')
%print -f -deps optdisc4

```

Programa en Código C

4. Programa de Control Optimo Continuo para Referencia Constante y Variante (trayectoria)

```
/*Archivo : srm4opt.cpp
```

PROGRAMA DEL SISTEMA DE CONTROL OPTIMO CONTINUO DE UN ROBOT
ESFERICO DE DOS GRADOS DE LIBERTAD

motor1: MATSUSHITA GMX-7MCO19B - serie: 5911 NC (BASE)
motor2: MATSUSHITA GMX-7MCO19B - serie: 5807 NC (BRAZO)
M.Sc. Ra'ul Benites Saravia */

```
#include "r2_llib.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <assert.h>
#include <alloc.h>
#include <conio.h>
#include <time.h>

#define IZQ 0
#define DER 1
#define MOTOR1 IZQ
#define MOTOR2 DER
#define A_RADIANES1 2.*M_PI/100.
#define A_RADIANES2 2.*M_PI/100.
#define Nm1 1/18.5 //servosystem serie: 5911 (base)
#define Nm2 1/18.5 //servosystem serie: 5807 (brazo)

/* Declaracion de las funciones propias del programa */

void Inicializa_Sistema();
void Mide_Salidas(float *y1, float *vel1, float *y2, float *vel2);
float Aplica_Control1(float u1, float yvel1);
float Aplica_Control2(float u2, float yvel2);

/* Declaracion de las variables globales */

float y1, y1ant, y2, y2ant;
float T, t, tsimul, Fs, N;
float VcFC, VcFE, Wth, Ofst1, Ofst2;
float E[4][2]; Da[2][6];
float Nt1, Nt2, a1, a2, a3, Rt1, Rt2, J1, J2, b1, b2;

/***** ** FUNCIONES *****/

void Inicializa_Sistema()
{
    VcFE= .2; VcFC= .15;
    Wth = .1;
    Ofst1= 2.16;

```

```

Ofst2= 2.16;
//T=0.0014; Fs=700 Hz
Fs= 700.; T= 1./Fs;
tsimul= 20.;

/* Condiciones iniciales */
u1p=0.; u2p=0.;
xe1p=0.; xe2p=0.; xe3p=0.; xe4p=0.;
z1p=0.; z2p=0.;

/* Matriz Ganancia del Observador de Estados */
E[0][0]=10.1038; E[0][1]=0.;
E[1][0]=0.; E[1][1]=10.1022;
E[2][0]=0.0245; E[2][1]=0.;
E[3][0]=0.; E[3][1]=0.0164;

/* Matriz Ganancia del Controlador Proporcional Integral */
Da[0][0]=21.6226; Da[0][1]=0.; Da[0][2]=1.1318; Da[0][3]=0.;
Da[0][4]=104.8427; Da[0][5]=0.;
Da[1][0]=0.; Da[1][1]=21.4643; Da[1][2]=0.; Da[1][3]=1.1119;
Da[1][4]=0.; Da[1][5]=104.4503;

/* Par'ámetros del sistema */
Nt1=0.0575; Nt2=Nt1;
a1=0.0025; a2=0.0219; a3=1.2361;
Rt1=0.3015; Rt2=0.3015;
J1=5.5694e-4; J2=4.6888e-4;
b1=1.3944e-4; b2=8.9966e-5;

ConfigurarHardware(Fs);
EnviarVoltaje1(Ofst1);
EnviarVoltaje2(Ofst2);
}

void Mide_Salidas(float *y1, float *vel1, float *y2, float *vel2)
{
    static float y1ant=0;
    static float y2ant=0;
    static unsigned contC1_ant=0;
    static unsigned contC2_ant=0;
    unsigned contC1_act, contC2_act;

//Lee Posicion del MOTOR1

    contC1_act=LeerSensor(MOTOR1);
    if(contC1_act<contC1_ant)
        {

```



```

        if((contC1_ant-contC1_act)>32768)
            *y1=y1ant+(65536-contC1_ant+contC1_act)*A_RADIANES1*Nm1;
        else
            *y1=y1ant-(contC1_ant-contC1_act)*A_RADIANES1*Nm1;
    }
else
    {
        if((contC1_act-contC1_ant)>32768)
            *y1=y1ant-(65536-contC1_act+contC1_ant)*A_RADIANES1*Nm1;
        else
            *y1=y1ant+(contC1_act-contC1_ant)*A_RADIANES1*Nm1;
    }
*vel1=(5.*(*y1-y1ant)+*vel1)/(1.+5.*T);
contC1_ant=contC1_act;
y1ant=*y1;

//Lee Posicion del MOTOR2

contC2_act=LeerSensor(MOTOR2);
if(contC2_act<contC2_ant)
    {
        if((contC2_ant-contC2_act)>32768)
            *y2=y2ant+(65536-contC2_ant+contC2_act)*A_RADIANES2*Nm2;
        else
            *y2=y2ant-(contC2_ant-contC2_act)*A_RADIANES2*Nm2;
    }
else
    {
        if((contC2_act-contC2_ant)>32768)
            *y2=y2ant-(65536-contC2_act+contC2_ant)*A_RADIANES2*Nm2;
        else
            *y2=y2ant+(contC2_act-contC2_ant)*A_RADIANES2*Nm2;
    }
*vel2=(5.*(*y2-y2ant)+*vel2)/(1.+5.*T);
contC2_ant=contC2_act;
y2ant=*y2;
}

float Aplica_Control1(float u1, float yvel1) {
    float v1;

    if(fabs(yvel1)<=Wth)
    {
        if(u1<0.) v1 = -VcFE;
        if(u1>0.) v1 = VcFE;
    }
    else
    {
        if(u1<0.) v1 = -VcFC;
    }
}

```

```

        if(u1>0.) v1 = VcFC;
    }

    if(u1+v1<-1.4)    u1= -1.4;
    else if(u1+v1>1.4) u1=  1.4;

    EnviarVoltaje1(u1+v1+Ofst1);

    return u1;
}

float Aplica_Control2(float u2, float yvel2) {
    float v2;

    if(fabs(yvel2)<=Wth)
    {
        if(u2<0.) v2 = -VcFE;
        if(u2>0.) v2 =  VcFE;
    }
    else
    {
        if(u2<0.) v2 = -VcFC;
        if(u2>0.) v2 =  VcFC;
    }

    if(u2+v2<-1.4)    u2= -1.4;
    else if(u2+v2>1.4) u2=  1.4;

    EnviarVoltaje2(u2+v2+Ofst2);

    return u2;
}

void main()
{
    int ant=0, act=0, k, n;
    FILE *out;
    float *yb1, *ub1, *velb1, vel1, y1, u1, y1ant;
    float *yb2, *ub2, *velb2, vel2, y2, u2, y2ant;
    float r1, r2, x1, x2, x3, x4;
    float xe1, xe1p, xe2, xe2p, xe3, xe3p; xe4, xe4p;
    float e1, e2, u1p, u2p, z1, z1p, z2, z2p;
    float p33, pa, pb, q42, q43, q44, g3, g4;
    float f, w;

    Inicializa_Sistema();
    clrscr();
    assert(tsimul<100);
    N= tsimul/T;

```

```

n= (int)N;
t=0.;
vel1=0.;
vel2=0.;

assert(NULL!=(ub1= (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(yb1= (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(velb1= (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(ub2= (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(yb2= (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(velb2= (float *)calloc(n/5,sizeof(float))));

for(k=0; k<n;)
{
act=NivelClock();
if ((ant==0)&&(act==1))
{
Mide_Salidas(&y1, &vel1, &y2, &vel2);

/* Definiendo las referencias */
r1=M_PI/2;
r2=M_PI/4;

/* Definiendo trayectorias */
//r1=sin(t)+0.1*t;
//r2=cos(t);

/* Valores de par'ámetros del proceso */
x1=y1; x2=y2; x3=vel1; x4=vel2;
p33=Rt1*(b1-2.*a2*x4*sin(x2)*cos(x2))+Nt1;
pa=Rt1*(J1+a1+a2*cos(x2)*cos(x2));
pb=Rt2*(J2+a2); q42=(Rt2*a3)/(Rt2*(J2+a2));
q43=(Rt2*a2*x3*sin(x2)*cos(x2))/(Rt2*(J2+a2));
q44=(Rt2*b2+Nt2)/(Rt2*(J2+a2));
g3=1./pa; g4=1./pb;

/* Error de Posici'on */
e1=-r1+x1; e2=-r2+x2;

/* Observador No Lineal */
xe1=xe1p+T*(xe3p-E[0][0]*(xe1p-e1)-E[0][1]*(xe2p-e2));
xe2=xe2p+T*(xe4p-E[1][0]*(xe1p-e1)-E[1][1]*(xe2p-e2));
xe3=xe3p+T*(-p33*g3*xe3p+g3*u1p-E[2][0]*(xe1p-e1)- ...
E[2][1]*(xe2p-e2));
xe4=xe4p+T*(-q42*cos(xe2p)-q43*xe3p-q44*xe4p+g4*u2p-
E[3][0]*(xe1p-e1)-E[3][1]*(xe2p-e2));

/* Leyes de Control u1 y u2 */
z1=z1p+T*e1; z2=z2p+T*e2;

```

```

u1=-Da[0][0]*xe1-Da[0][1]*xe2-Da[0][2]*xe3-Da[0][3]*xe4- ...
Da[0][4]*z1-Da[0][5]*z2;
u2=-Da[1][0]*xe1-Da[1][1]*xe2-Da[1][2]*xe3-Da[1][3]*xe4- ...
Da[1][4]*z1-Da[1][5]*z2;

/* Actualizaci'on de Variables */
u1p=u1; u2p=u2;
xe1p=xe1; xe2p=xe2; xe3p=xe3; xe4p=xe4;
z1p=z1; z2p=z2;

u1= Aplica_Control1(u1, vel1);
if(!(k%5))
{
    yb1[k/5]=y1*180./M_PI; ub1[k/5]=u1;
    velb1[k/5]=vel1;
}

u2=Aplica_Control2(u2, vel2);
if(!(k%5))
{
    yb2[k/5]=y2*180./M_PI; ub2[k/5]=u2;
    velb2[k/5]=vel2;
}

/* Actualizaci'on de Variables */
u1p=u1; u2p=u2;
xe1p=xe1; xe2p=xe2; xe3p=xe3; xe4p=xe4;
z1p=z1; z2p=z2;

k++;
t+=T;
gotoxy(15,5);printf("*****");
gotoxy(18,7);printf("RESPUESTA DEL SISTEMA DE CONTROL OPTIMO");
gotoxy(18,9);printf("DEL ROBOT ESFERICO DE 2 G.L.");
gotoxy(15,11);printf("*****");
gotoxy(21,13);printf("tiempo          : t=%+5.3f",t);
gotoxy(21,15);printf("control 1       : u1=%+5.3f",u1);
gotoxy(21,17);printf("pos. angular 1 : y1=%+5.3f",y1*180./M_PI);
gotoxy(21,19);printf("referencia 1   : r1= %+5.3f", r1*180./M_PI);
gotoxy(21,21);printf("error 1        : e1= %+5.3f", e1*180./M_PI);
gotoxy(21,23);printf("vel. angular 1 : vel1=%+5.3f",vel1);
gotoxy(21,25);printf("control 2       : u2=%+5.3f",u2);
gotoxy(21,27);printf("pos. angular 2 : y2=%+5.3f",y2*180./M_PI);
gotoxy(21,29);printf("referencia 2   : r2= %+5.3f", r2*180./M_PI);
gotoxy(21,31);printf("error 2        : e2= %+5.3f", e2*180./M_PI);
gotoxy(21,33);printf("vel. angular 2 : vel2=%+5.3f",vel2);
gotoxy(15,35);printf("*****");

if(kbhit()    if(getch()==27) break;

```

```

    }
    ant=act;
}

EnviarVoltaje1(Ofst1);
EnviarVoltaje2(Ofst2);

out=fopen("srm4opt.dat","wt");
for (int i=0; i<n/5; i++)
    fprintf(out,"%10f %10f %10f %10f %10f %10f %10f\n",
        i*5*T,ub1[i],yb1[i],velb1[i],ub2[i],yb2[i], velb2[i]);
delete ub1, yb1, velb1, ub2, yb1, velb2;
fclose(out);
}

```

5. Programa de Control Optimo Discreto para Referencia Constante y Variante (trayectoria)

```

/*Archivo : srm4optd.cpp
PROGRAMA DEL SISTEMA DE CONTROL OPTIMO PROPORCIOAL INTEGRAL
DISCRETO DE UN ROBOT ESFERICO DE DOS GRADOS DE LIBERTAD
motor1: MATSUSHITA GMX-7MCO19B - serie: 5911 NC (BASE)
motor2: MATSUSHITA GMX-7MCO19B - serie: 5807 NC (BRAZO)
M.Sc. Ra'ul Benites Saravia */

#include "r2_llib.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <assert.h>
#include <alloc.h>
#include <conio.h>
#include <time.h>

#define IZQ 0
#define DER 1
#define MOTOR1 IZQ
#define MOTOR2 DER
#define A_RADIANES1 2.*M_PI/100.
#define A_RADIANES2 2.*M_PI/100.
#define Nm1 1/18.5 //servosystem serie: 5911 (base)
#define Nm2 1/18.5 //servosystem serie: 5807 (brazo)

/* Declaraci'on de las funciones propias del programa */

void Inicializa_Sistema();
void Mide_Salidas(float *y1, float *vel1, float *y2, float *vel2);
float Aplica_Control1(float u1,float yvel1);

```

```

float Aplica_Control2(float u2, float yvel2);

/* Declaraci'on de las variables globales */

float    y1, y1ant, y2, y2ant; float    T, t, tsimul, Fs, N; float
VcFC, VcFE, Wth, Ofst1, Ofst2; float    Ke[4][2]; Ka[2][6];
G[4][4]; H[4][2]; Cd[2][4];
float Nt1, Nt2, a1, a2, a3, Rt1, Rt2, J1, J2, b1, b2;

/***** ** FUNCIONES *****/

void Inicializa_Sistema() {
    VcFE= .2; VcFC= .15;
    Wth = .1;
    Ofst1= 2.16;
    Ofst2= 2.16;
    //T=0.005; Fs=200 Hz
    Fs= 200.; T= 1./Fs;
    tsimul= 20.;

    /* Condiciones iniciales */
    //u1p=0.; u2p=0.;
    v1p=0.; v2p=0.; x1=0.; x2=0.;
    xe1p=0.; xe2p=0.; xe3p=0.; xe4p=0.;

    /* Matriz Ganancia del Observador de Estados */
    Ke[0][0]=0.9164; Ke[0][1]=0.;
    Ke[1][0]=0.; Ke[1][1]=0.9164;
    Ke[2][0]=0.0562; Ke[2][1]=0.;
    Ke[3][0]=0.; Ke[3][1]=0.0500;

    /* Matriz Ganancia del Controlador Proporcional Integral */
    Ka[0][0]=15.0906; Ka[0][1]=0.; Ka[0][2]=0.7924; Ka[0][3]=0.;
    Ka[0][4]=-0.6995; Ka[0][5]=0.;
    Ka[1][0]=0.; Ka[1][1]=14.4815; Ka[1][2]=0.; Ka[1][3]=0.7530;
    Ka[1][4]=0.; Ka[1][5]=-0.6756;

    KI1=-Ka[0][4]; KI2=-Ka[1][5];

    /* Matrices del proceso */
    G[0][0]=1.; G[0][1]=0.; G[0][2]=0.0049; G[0][3]=0.;
    G[1][0]=0.; G[1][1]=1.; G[1][2]=0.; G[1][3]=0.0049;
    G[2][0]=0.; G[2][1]=0.; G[2][2]=0.9626; G[2][3]=0.;
    G[3][1]=0.; G[3][2]=0.; G[3][3]=0.; G[3][4]=0.9582;

    H[0][0]=0.0016; H[0][1]=0.;
    H[1][0]=0.; H[1][1]=0.0018;

```

```

//Lee Posicion del MOTOR2

contC2_act=LeerSensor(MOTOR2);
if(contC2_act<contC2_ant)
{
    if((contC2_ant-contC2_act)>32768)
        *y2=y2ant+(65536-contC2_ant+contC2_act)*A_RADIANES2*Nm2;
    else
        *y2=y2ant-(contC2_ant-contC2_act)*A_RADIANES2*Nm2;
}
else
{
    if((contC2_act-contC2_ant)>32768)
        *y2=y2ant-(65536-contC2_act+contC2_ant)*A_RADIANES2*Nm2;
    else
        *y2=y2ant+(contC2_act-contC2_ant)*A_RADIANES2*Nm2;
}
*vel2=(5.*(y2-y2ant)+*vel2)/(1.+5.*T);
contC2_ant=contC2_act;
y2ant=*y2;
}

float Aplica_Control1(float u1, float yvel1)
{
    float v1;

    if(fabs(yvel1)<=Wth)
    {
        if(u1<0.) v1 = -VcFE;
        if(u1>0.) v1 = VcFE;
    }
    else
    {
        if(u1<0.) v1 = -VcFC;
        if(u1>0.) v1 = VcFC;
    }

    if(u1+v1<-1.4) u1= -1.4;
    else if(u1+v1>1.4) u1= 1.4;

    EnviarVoltaje1(u1+v1+Ofst1);

    return u1;
}

float Aplica_Control2(float u2, float yvel2)
{
    float v2;

```

```

H[2][0]=0.6512; H[2][1]=0.;
H[3][1]=0.; H[3][2]=0.7263;

Cd[0][0]=1.; Cd[0][1]=0.; Cd[0][2]=0.; Cd[0][3]=0.;
Cd[1][0]=0.; Cd[1][1]=1.; Cd[1][2]=0.; Cd[1][3]=0.;

/* Par'ámetros del sistema */
Nt1=0.0575; Nt2=Nt1;
a1=0.0025; a2=0.0219; a3=1.2361;
Rt1=0.3015; Rt2=0.3015;
J1=5.5694e-4; J2=4.6888e-4;
b1=1.3944e-4; b2=8.9966e-5;

ConfigurarHardware(Fs);
EnviarVoltaje1(Ofst1);
EnviarVoltaje2(Ofst2);
}

void Mide_Salidas(float *y1, float *vel1, float *y2, float *vel2)
{
    static float y1ant=0;
    static float y2ant=0;
    static unsigned contC1_ant=0;
    static unsigned contC2_ant=0;
    unsigned contC1_act, contC2_act;

    //Lee Posicion del MOTOR1

    contC1_act=LeerSensor(MOTOR1);
    if(contC1_act<contC1_ant)
    {
        if((contC1_ant-contC1_act)>32768)
            *y1=y1ant+(65536-contC1_ant+contC1_act)*A_RADIANES1*Nm1;
        else
            *y1=y1ant-(contC1_ant-contC1_act)*A_RADIANES1*Nm1;
    }
    else
    {
        if((contC1_act-contC1_ant)>32768)
            *y1=y1ant-(65536-contC1_act+contC1_ant)*A_RADIANES1*Nm1;
        else
            *y1=y1ant+(contC1_act-contC1_ant)*A_RADIANES1*Nm1;
    }
    *vel1=(5.*(y1-y1ant)+*vel1)/(1.+5.*T);
    contC1_ant=contC1_act;
    y1ant=*y1;
}

```



```

    if(fabs(yvel2)<=Wth)
    {
        if(u2<0.) v2 = -VcFE;
        if(u2>0.) v2 = VcFE;
    }
    else
    {
        if(u2<0.) v2 = -VcFC;
        if(u2>0.) v2 = VcFC;
    }

    if(u2+v2<-1.4)      u2= -1.4;
    else if(u2+v2>1.4) u2=  1.4;

    EnviarVoltaje2(u2+v2+Ofst2);

    return u2;
}

void main()
{
    int ant=0, act=0, k, n;
    FILE *out;
    float *yb1, *ub1, *velb1, vel1, y1, u1, y1ant;
    float *yb2, *ub2, *velb2, vel2, y2, u2, y2ant;
    float r1, r2; x1, x2, x3, x4; xe1, xe1p, xe2, xe2p, xe3, xe3p,
    float xe4, xe4p;
    float e1, e2; //u1p, u2p;
    float p33, pa, pb, q42, q43, q44, g3, g4;
    float f, w;

    Inicializa_Sistema();
    clrscr();
    assert(tsimul<100);
    N= tsimul/T;
    n= (int)N;
    t=0.;
    vel1=0.;
    vel2=0.;

    assert(NULL!=(ub1= (float *)calloc(n/5,sizeof(float))));
    assert(NULL!=(yb1= (float *)calloc(n/5,sizeof(float))));
    assert(NULL!=(velb1= (float *)calloc(n/5,sizeof(float))));
    assert(NULL!=(ub2= (float *)calloc(n/5,sizeof(float))));
    assert(NULL!=(yb2= (float *)calloc(n/5,sizeof(float))));
    assert(NULL!=(velb2= (float *)calloc(n/5,sizeof(float))));

    for(k=0; k<n;)
    {

```

```

act=NivelClock();
if ((ant==0)&&(act==1))
{
  Mide_Salidas(&y1, &vel1, &y2, &vel2);

  /* Definiendo las referencias */
  r1=M_PI/2;
  r2=M_PI/4;

  /* Definiendo trayectorias */
  //r1=sin(t)+0.1*t;
  //r2=cos(t);

  /* Valores de par'ámetros del proceso */
  x1=y1; x2=y2; x3=vel1; x4=vel2;
  p33=Rt1*(b1-2.*a2*x4*sin(x2)*cos(x2))+Nt1;
  pa=Rt1*(J1+a1+a2*cos(x2)*cos(x2));
  pb=Rt2*(J2+a2); q42=(Rt2*a3)/(Rt2*(J2+a2));
  q43=(Rt2*a2*x3*sin(x2)*cos(x2))/(Rt2*(J2+a2));
  q44=(Rt2*b2+Nt2)/(Rt2*(J2+a2));
  g3=1./pa; g4=1./pb;

  /* Error de Posici'on */
  e1=r1-x1; e2=r2-x2;

  /* Observador:
  xe(k+1)=G*xe(k)+H*u(k)+Ke*([x1(k);x2(k)]-Cd*xe(k)) */

  // Asignando las siguientes variables para comodidad del cálculo
  GX1=G[0][0]*xe1p+G[0][1]*xe2p+G[0][2]*xe3p+G[0][3]*xe4p;
  GX2=G[1][0]*xe1p+G[1][1]*xe2p+G[1][2]*xe3p+G[1][3]*xe4p;
  GX3=G[2][0]*xe1p+G[2][1]*xe2p+G[2][2]*xe3p+G[2][3]*xe4p;
  GX4=G[3][0]*xe1p+G[3][1]*xe2p+G[3][2]*xe3p+G[3][3]*xe4p;

  HU1=H[0][0]*u1+H[0][1]*u2;
  HU2=H[1][0]*u1+H[1][1]*u2;
  HU3=H[2][0]*u1+H[2][1]*u2;
  HU4=H[3][0]*u1+H[3][1]*u2;

  XC1=x1-(Cd[0][0]*xe1p+Cd[0][1]*xe2p+Cd[0][2]*xe3p+Cd[0][3]*xe4p);
  XC2=x2-(Cd[1][0]*xe1p+Cd[1][1]*xe2p+Cd[1][2]*xe3p+Cd[1][3]*xe4p);

  KCX1=Ke[0][0]*XC1+Ke[0][1]*XC2;
  KCX2=Ke[1][0]*XC1+Ke[1][1]*XC2;
  KCX3=Ke[2][0]*XC1+Ke[2][1]*XC2;
  KCX4=Ke[3][0]*XC1+Ke[3][1]*XC2;

  xe1=GX1+HU1+KCX1;
  xe2=GX2+HU2+KCX2;

```

```

xe3=Gx3+Hu3+Kcx3;
xe4=Gx4+Hu4+Kcx4;

/* Leyes de Control u1 y u2 */
u1=-(Ka[0][0]*xe1+Ka[0][1]*xe2+Ka[0][2]*xe3-Da[0][3]*xe4)+...
    +KI1*v1p;
u2=-(Ka[1][0]*xe1+Ka[1][1]*xe2+Ka[1][2]*xe3-Da[1][3]*xe4)+...
    +KI2*v2p;

/* Ecuación de los integradores */
v1=v1p+r1-x1;
v2=v2p+r2-x2;

u1= Aplica_Control1(u1, vel1);
if(!(k%5))
{
    yb1[k/5]=y1*180./M_PI; ub1[k/5]=u1;
    velb1[k/5]=vel1;
}

u2=Aplica_Control2(u2, vel2);
if(!(k%5))
{
    yb2[k/5]=y2*180./M_PI; ub2[k/5]=u2;
    velb2[k/5]=vel2;
}

/* Actualizaci'on de Variables */
%u1p=u1; u2p=u2;
xe1p=xe1; xe2p=xe2; xe3p=xe3; xe4p=xe4;
v1p=v1; v2p=v2;

k++;
t+=T;
gotoxy(15,5); printf("*****");
gotoxy(18,7); printf("RESPTA. DEL SIST. DE CONT. OPTIMO DISCRETO");
gotoxy(18,9); printf("DEL ROBOT ESFERICO DE DOS G. L.");
gotoxy(15,11);printf("*****");
gotoxy(21,13);printf("tiempo      : t=%+5.3f",t);
gotoxy(21,15);printf("control 1   : u1=%+5.3f",u1);
gotoxy(21,17);printf("pos. angular 1 : y1=%+5.3f",y1*180./M_PI);
gotoxy(21,19);printf("referencia 1  : r1= %+5.3f", r1*180./M_PI);
gotoxy(21,21);printf("error 1      : e1= %+5.3f", e1*180./M_PI);
gotoxy(21,23);printf("vel. angular 1 : vel1=%+5.3f",vel1);
gotoxy(21,25);printf("control 2   : u2=%+5.3f",u2);
gotoxy(21,27);printf("pos. angular 2 : y2=%+5.3f",y2*180./M_PI);
gotoxy(21,29);printf("referencia 2  : r2= %+5.3f", r2*180./M_PI);
gotoxy(21,31);printf("error 2      : e2= %+5.3f", e2*180./M_PI);
gotoxy(21,33);printf("vel. angular 2 : vel2=%+5.3f",vel2);

```

```
gotoxy(15,35);printf("*****");

    if(kbhit()    if(getch()==27) break;
    }
    ant=act;
}

EnviarVoltaje1(Ofst1);
EnviarVoltaje2(Ofst2);

out=fopen("srm4optd.dat","wt");
for (int i=0; i<n/5; i++)
    fprintf(out,"%10f %10f %10f %10f %10f %10f %10f\n",
        i*5*T,ub1[i],yb1[i],velb1[i],ub2[i],yb2[i], velb2[i]);
delete ub1, yb1, velb1, ub2, yb1, velb2;
fclose(out);
}
```

APÉNDICE B

DIAGRAMAS CIRCUITALES

En el presente apéndice se presentan los siguientes diagramas:

- Figura B.1: Circuito generador PWM (un canal).
- Figura B.2: circuito amplificador de potencia (un canal).
- Figuras B.3 y B.4: Circuitos decodificador - contador de los canales 1 y 2, respectivamente.
- Figura B.5: Circuito de la sección de multiplexaje de los datos provenientes de cada unidad contadora.

Podrá observarse, que en el circuito del driver (generador PWM y amplificador de potencia), sólo se ha presentado el correspondiente a un canal, ya que para el otro canal es idéntico; sin embargo, para la unidad sensora, es decir los diagramas del decodificador y contador, se presentan las dos unidades o canales, debido a la necesidad de presentar con transparencia el multiplexaje de los datos de ambos canales.

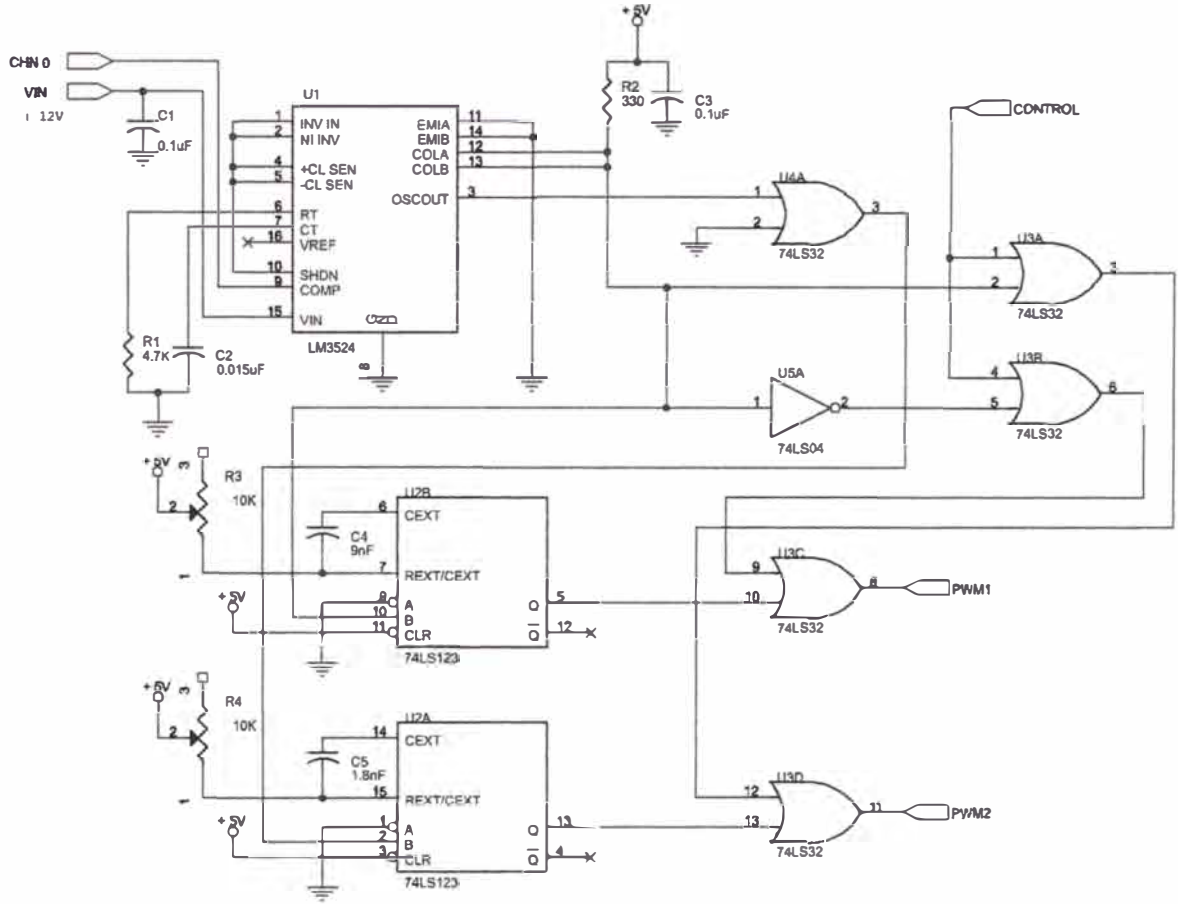
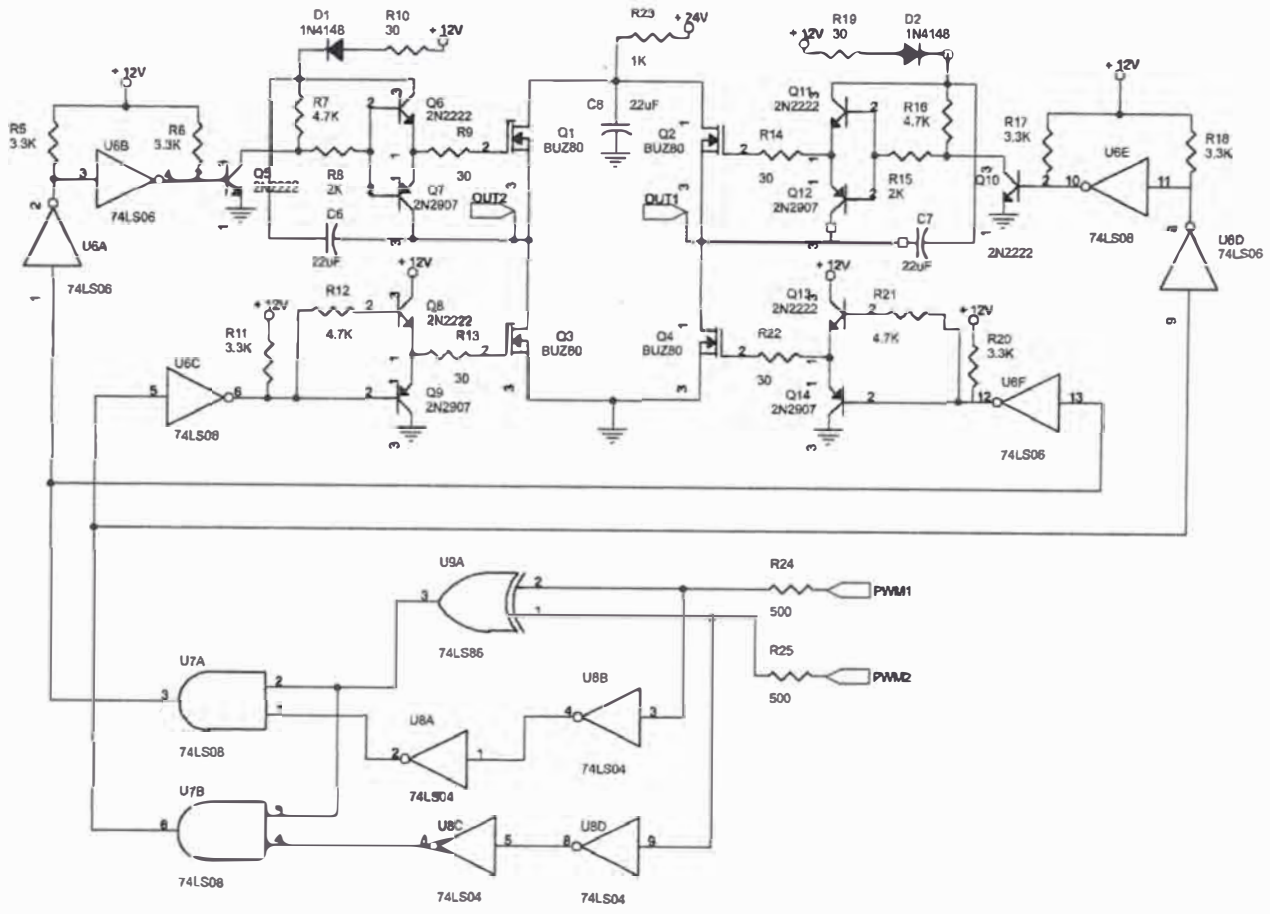


Figura B.1: Circuito Generador PWM con tiempo muerto

Figura B.2: Circuito Amplificador de Potencia



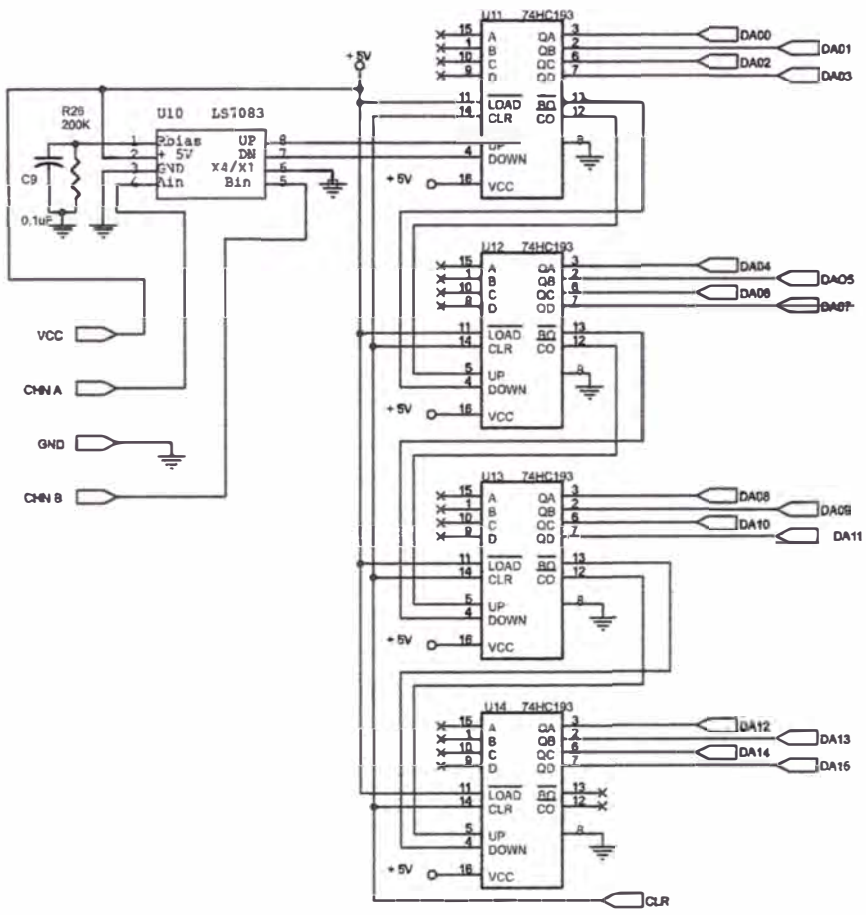


Figura B.3: Circuito Decodificador - Contador 1

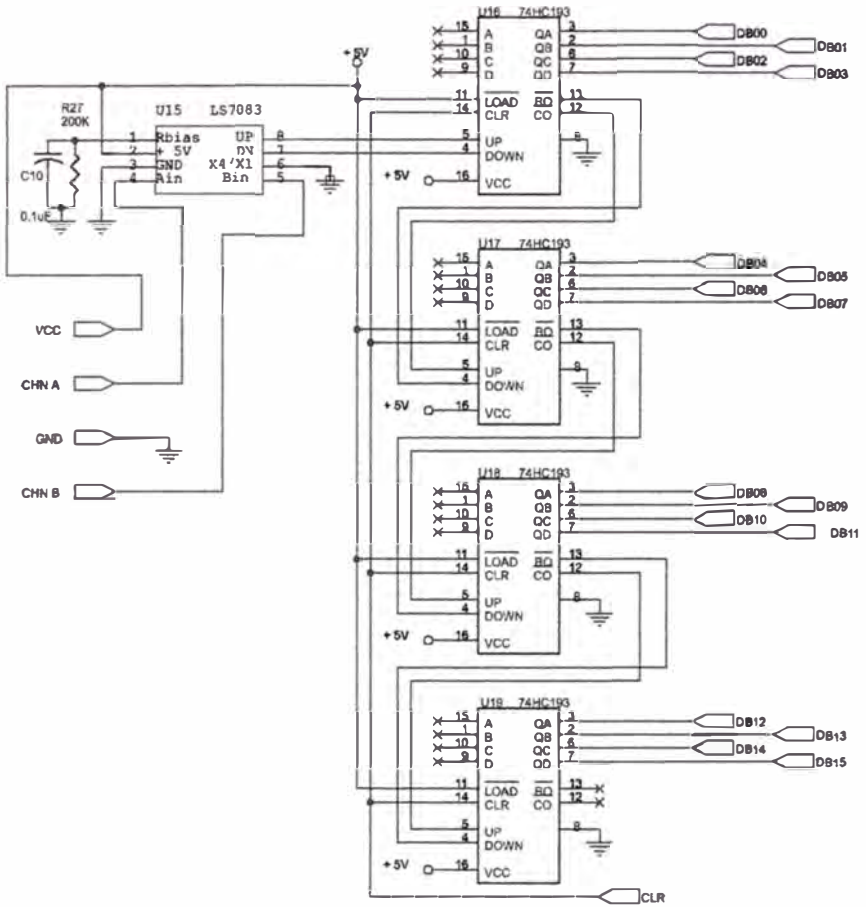


Figura B.4: Circuito Decodificador - Contador 2

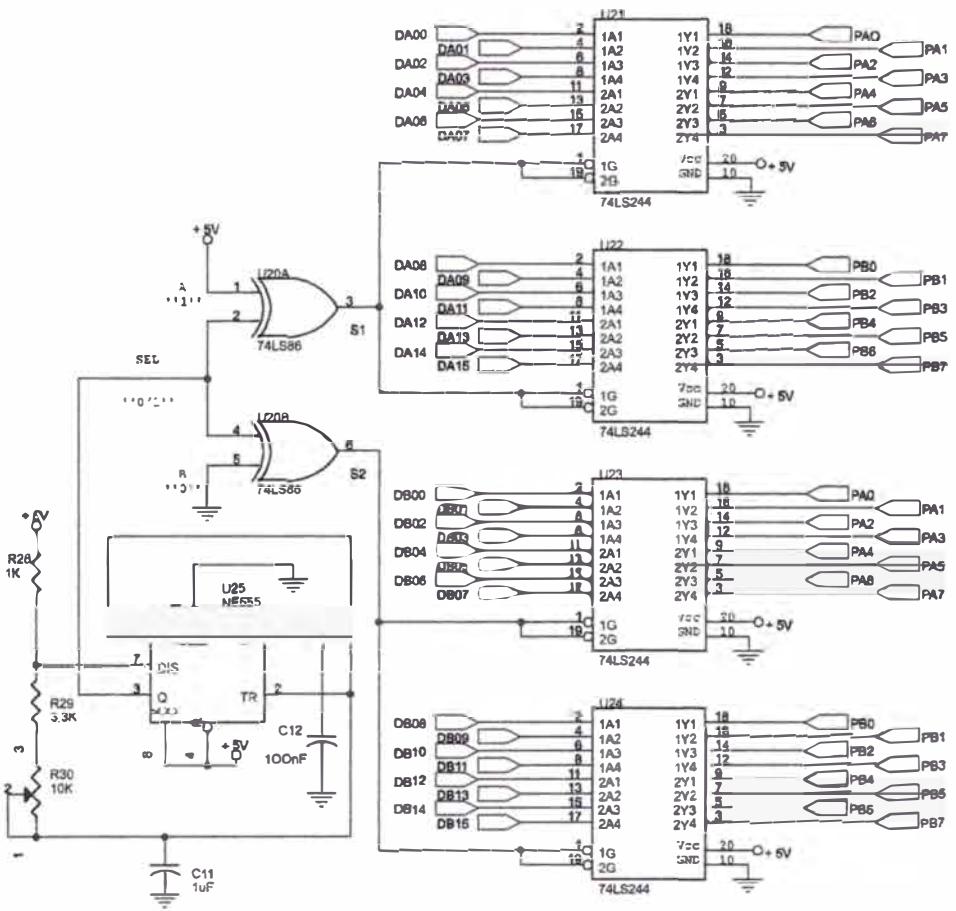


Figura B.5: Circuito Decodificador - Contador 3
(Sección de Multiplexaje de datos)

BIBLIOGRAFÍA

- [1] Ollero Baturone, Aníbal. *Robótica, Manipuladores y robots móviles*. Alfaomega, 2001.
- [2] Fu K.S., R.C. González y C.S.G. Lee. *Robotics: control, sensing, visión and intelligence*. McGraw-Hill, 1987.
- [3] Schilling, Robert J. *Fundamentals of Robotics: analysis and control*. Prentice Hall, 1999.
- [4] Angulo Usategui, José M. *Robótica Práctica, Tecnología y Aplicaciones*. Paraninfo, 2000.
- [5] José M. Angulo Usategui, Susana Romero Yesa, Ignacio Angulo Martínez. *Microbótica, Tecnología, Aplicaciones y Montaje Práctico*. Paraninfo, 2001.
- [6] JMikell P. , Mitchell Weiss, Roger N. Nagel y Nichols G. Odrey. *Robótica Industrial, Tecnología, Programación y Aplicaciones*. McGraw-Hill, 1990.
- [7] Antonio Barrientos, Luis Felipe Peñín, Carlos Balaguer, Rafael Aracil. *Fundamentos de Robótica*. McGraw-Hill, 1997.
- [8] De Prada, César. *Introduction to Modelling and Identification*. PADI2, International Conference on Automatic Control, october 1998, Piura, Perú.
- [9] Lewis, C.T. Abdallah y D.M. Dawson. *Control of Robot Manipulators*. Macmillan Publishing Co., 1993.
- [10] Gray, J.O. and Caldwell, D.G. *DC Motors, Speed Controls, Servo Systems*. The Institution of Electrical Enginners, 1996.
- [11] Dare A. Wells. *Teoría y Problemas de Dinámica de Lagrange*. McGraw-Hill, 1972.

- [12] Rojas Moreno, Arturo. *Control Avanzado-Diseño y Aplicaciones en Tiempo Real*. Publicación Independiente, 2001.
- [13] Rojas Moreno, Arturo. *Applied Multivariable Nonlinear Control-Design and Real-Time Applications*. En progreso, 2002.
- [14] Eronini-Umez-Eronini. *Dinámica de Sistemas y Control*. Thomson Learning, 2001.
- [15] Ogata, Katsuhiko. *Ingeniería de Control Moderna*. Prentice Hall Hispanoamericana, 1993.
- [16] Ogata, Katsuhiko. *Sistemas de Control en Tiempo Discreto*. Prentice Hall Hispanoamericana, 1996.
- [17] Ogata, Katsuhiko. *Ingeniería de Control utilizando MATLAB*. Prentice Hall Iberia, 1999.
- [18] National Semiconductor Corporation. *Lab-PC+ User Manual*. National Semiconductor Corporation, 1994.
- [19] The Electro-Craft Engineering Handbook by Reliance Motion Control, Inc. *DC Motors, Speed Controls, Servo Systems*.
- [20] The Math Works, Inc. *Control System Toolbox User's Guide*. The Math Works, Inc., 1992.
- [21] Benites Saravia, N. Raúl. *Diseño e Implementación de un Sistema Control de Posición Digital Predictivo para un Motor DC Sujeto a Cargas No Lineales*. Tecnia, Vol. 11 N° 2, págs. 67-75, 2001, Universidad Nacional de Ingeniería, Lima, Perú.