

UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE INGENIERÍA MECÁNICA



**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
CONTROL POR LÓGICA DIFUSA DEL TIPO
TAKAGI-SUGENO APLICADO A UN PROTOTIPO
DE GRÚA-PUENTE**

INFORME DE SUFICIENCIA

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

Gustavo Adolfo Navarro Chávez

PROMOCIÓN 2001-II

LIMA - PERÚ

2011

A Irma.

TABLA DE CONTENIDOS

PRÓLOGO	1
CAPÍTULO 1.	
INTRODUCCIÓN	4
1.1. Antecedentes: Evolución del Control de las Grúas-Puente	6
1.2. Formulación del Problema	9
1.3. Objetivos Generales	10
CAPÍTULO 2.	
TEORÍA DE CONTROL POR LÓGICA DIFUSA	11
2.1. Introducción a la Teoría de la Lógica Difusa	11
2.1.1. Conjuntos Difusos	12
2.1.2. Reglas Difusas y Razonamiento Difuso	22
2.1.3. Sistemas de Inferencia Difusa: Modelo de Takagi-Sugeno	42
CAPÍTULO 3.	
MODELO MATEMÁTICO DEL SISTEMA GRÚA-PUENTE	45
3.1. Descripción del Sistema	45
3.2. Parámetros del Modelo	48
3.3. Ecuaciones de Movimiento	49
3.3.1. Subsistema Carro - Péndulo	49
3.3.2. Subsistema Motor-Polea	53
3.4. Representación en el Espacio de Estados del Sistema Grúa-Puente	55
3.5. Modelos de los Fenómenos No-Lineales	57
3.5.1. Zona Muerta del Motor	57
3.5.2. Fricción	59
3.5.3. Inclusión de los Fenómenos No-Lineales en el Modelo de la Grúa-Puente	61
3.6. Respuesta en Lazo Abierto del Sistema	62
3.7. Parámetros Ajustados mediante un Algoritmo Genético	64
CAPÍTULO 4.	
SISTEMA DE CONTROL Y SIMULACIONES	70
4.1. Diseño del Controlador por Lógica Difusa	71
4.1.1. Planeamiento de la Trayectoria	72

4.1.2.	Sistema de Inferencia Difusa del Tipo Takagi-Sugeno para el Control de la Posición	73
4.1.3.	Sistema de Inferencia Difusa del Tipo Takagi-Sugeno para la Amortiguación de la Oscilación	79
4.1.4.	Puesta en Paralelo de los Sistemas de Inferencia	81
4.2.	Compensador de la Zona Muerta	83
4.3.	Simulación del Controlador Continuo	84
4.3.1.	Planta sin Fenómenos No-lineales	85
4.3.2.	Planta con Fenómenos No-lineales Sin Compensar	85
4.3.3.	Planta con Fenómenos No-lineales Compensados	89
4.4.	Simulación del Controlador Discreto	91
4.4.1.	Discretización de los Sensores	92
4.4.2.	Influencia del Período de Muestreo	95
4.4.3.	Planta sin Fenómenos No-lineales	99
4.4.4.	Planta con Fenómenos No-lineales Sin Compensar	101
4.4.5.	Planta con Fenómenos No-lineales Compensados	102

CAPÍTULO 5.

RESULTADOS EXPERIMENTALES 105

5.1.	Resultados de las Pruebas Realizadas	105
5.1.1.	Sin Compensación de los Fenómenos No Lineales	105
5.1.2.	Con Compensación de los Fenómenos No Lineales	107
5.1.3.	Variaciones en los Parámetros de la Planta	107
5.1.4.	Variaciones en los Parámetros del Controlador	109

CONCLUSIONES 116

RECOMENDACIONES 118

BIBLIOGRAFÍA 120

APÉNDICE A.

IMPLEMENTACIÓN EXPERIMENTAL 123

A.1.	Prototipo de Grúa-Puente	125
A.2.	Actuador	127
A.2.1.	Generador de PWM	127
A.2.2.	Optoacopladores	128
A.2.3.	Amplificador de Potencia	129
A.2.4.	Motor-Reductor DC	131
A.3.	Sensores	131
A.3.1.	Sensor del Ángulo de Oscilación de la Carga	131
A.3.2.	Sensor de la Posición del Cabezal	132
A.4.	Controlador <i>DSP TMS320F240</i>	132
A.4.1.	Relojes de Propósito General	134
A.4.2.	Configuración del Módulo PWM	135
A.4.3.	Decodificador del Tren de Pulsos en Cuadratura	135

A.5. Programa del Controlador en Tiempo Real	137
--	-----

APÉNDICE B.

IDENTIFICACIÓN DE PARÁMETROS	139
B.1. Identificación del Motor-Reductor DC	140
B.2. Identificación de la Zona Muerta del Motor-Polea	143
B.3. Identificación de los Parámetros de Fricción del Cabezal	144
B.4. Identificación Fina de los Parámetros de la Planta por un Algoritmo Genético	148

APÉNDICE C.

PROGRAMAS EN MATLAB[®]	153
C.1. Programa de Simulación en Tiempo Continuo	154
C.2. Programa de Simulación en Tiempo Discreto	162
C.3. Programa para la Identificación del Motor-reductor	171
C.4. Programas para la Identificación por un Algoritmo Genético	172
C.5. Programa del Modelo del Prototipo de Grúa-Puente	178

APÉNDICE D.

PROGRAMAS EN C	180
D.1. Programa para la Identificación de Parámetros del Motor DC	181
D.2. Programa para la Determinación de la Zona Muerta	182
D.3. Programa del Algoritmo de Control	184
D.4. Cabeceras del Algoritmo de Control	193

APÉNDICE E.

PROGRAMAS EN ASSEMBLER	198
E.1. Configuración DSP: Controlador	199

APÉNDICE F.

NORMAS APLICABLES A LAS GRÚAS-PUENTE	203
F.1. Listado Principales Normas Americanas e Internacionales	204

LISTA DE FIGURAS

1.1. Vista Panorámica del Puerto de Singapur	4
1.2. Vista Parcial del Puerto de Singapur	5
1.3. Grúas-puente en operación el puerto de Savannah, Georgia (USA).	6
2.1. (a) $A = \text{"número de niños en una familia"}$; (b) $B = \text{"alrededor de 50 años de edad"}$	13
2.2. Núcleo, soporte y puntos de cruce del conjunto difuso "Adulto".	14
2.3. El concepto de $A \subseteq B$	15
2.4. Operaciones con conjuntos difusos: (a) dos conjuntos difusos A y B ; (b) \bar{A} ; (c) $A \cup B$; (d) $A \cap B$	17
2.5. Ejemplos de cuatro clases de MF parametrizadas: (a) $triangle(x; 20, 60, 80)$; (b) $trapezoid(x; 10, 20, 60, 95)$; (c) $gaussian(x; 50, 20)$; (d) $bell(x; 20, 4, 50)$	19
2.6. (a) Dos funciones sigmoideas y_1 y y_2 ; (b) una MF cerrada obtenida con $ y_1 - y_2 $; (c) Dos funciones sigmoideas y_1 y y_3 ; (d) una MF cerrada obtenida con $y_1 y_3$	21
2.7. (a) Dos funciones sigmoideas y_1 y y_2 ; (b) una MF cerrada obtenida con $ y_1 - y_2 $; (c) Dos funciones sigmoideas y_1 y y_3 ; (d) una MF cerrada obtenida con $y_1 y_3$	23
2.8. Principio de extensión sobre conjuntos difusos con universos continuos. El gráfico inferior representa al conjunto difuso A ; el superior izquierdo es la función $y = f(x)$; y el superior derecho es el conjunto difuso B inducido a través del principio de extensión.	25
2.9. Funciones de membresía típicas del conjunto de términos $T(edad)$	27
2.10. Primera fila: funciones de implicación difusa basadas en la interpretación "A asociado con B"; segunda fila: las relaciones difusas correspondientes.	31
2.11. Primera fila: funciones de implicación difusa basadas en la interpretación "A implica B"; segunda fila: las relaciones difusas correspondientes.	32
2.12. Primera fila: funciones de implicación difusa basadas en la interpretación "A implica B"; segunda fila: las relaciones difusas correspondientes.	33
2.13. Derivación de $y = b$ a partir de $x = a$ y $y = f(x)$: (a) a y b son puntos, $y = f(x)$ es una curva; (b) a y b son intervalos, $y = f(x)$ es una función evaluada en intervalos.	34

2.14. Regla Composicional de Inferencia.	35
2.15. Interpretación gráfica del GMP usando la implicación difusa de Mamdani y la composición max-min.	38
2.16. Razonamiento aproximado para múltiples antecedentes.	39
2.17. Razonamiento aproximado para múltiples reglas con múltiples antecedentes.	41
2.18. El modelo difuso de Takagi-Sugeno.	44
3.1. Vista esquemática de una grúa-puente.	46
3.2. Sistema general del prototipo de grúa-puente.	47
3.3. Esquema del subsistema carro-péndulo.	51
3.4. Representación del subsistema motor-polea.	54
3.5. Modelo de la zona muerta.	58
3.6. Modelos de fricción.	60
3.7. Diagrama del sistema con la inclusión de fenómenos no lineales.	62
3.8. Diagrama de bloques del sistema grúa-puente en lazo abierto.	63
3.9. Respuesta del Cabezal a una Entrada Escal'on.	64
3.10. Respuesta del Péndulo a una Entrada Escalón.	65
3.11. Comparación entre la respuesta en lazo abierto de la planta y el modelo con los parámetros ajustados por un algoritmo genético: Posición.	68
3.12. Comparación entre la respuesta en lazo abierto de la planta y el modelo con los parámetros ajustados por un algoritmo genético: Desviación angular.	68
3.13. Comparación entre la respuesta en lazo abierto de la planta y el modelo con los parámetros ajustados por un algoritmo genético: Señal de control empleada.	69
4.1. Diagrama de bloques del FLC propuesto.	71
4.2. Perfiles de aceleración y velocidad de la trayectoria.	73
4.3. Funciones de membresía para la posición.	74
4.4. Funciones de membresía para la velocidad lineal.	74
4.5. Funciones de membresía C_i para la defuzzificación.	75
4.6. Esquema de una partición típica de grupos de celdas.	76
4.7. Diagrama de fase del controlador de la posición.	77
4.8. Diagrama de fase y funciones de membresía.	78
4.9. Funciones de membresía para la desviación angular.	79
4.10. Funciones de membresía para la velocidad angular.	80
4.11. Diagrama de fase del compensador de la oscilación.	81
4.12. Diagrama de la puesta en paralelo de los dos sistemas de inferencia difusa.	82
4.13. Diagrama de bloques de la compensación de la zona muerta.	83
4.14. Función de compensación de la zona muerta.	84
4.15. Diagrama de bloques del sistema de control en lazo cerrado.	85
4.16. Planta sin fenómenos no lineales: Posición y desviación angular; $K_{prop} = 1.4, M_L = 2.35Kg$	86

4.17. Planta sin fenómenos no lineales: Señales de control; $K_{prop} = 1.4, M_L = 2.35Kg$.	86
4.18. Planta con fenómenos no lineales sin compensar: Posición y desviación angular; $K_{prop} = 1.4, M_L = 2.35Kg$.	87
4.19. Planta con fenómenos no lineales sin compensar: Señales de control; $K_{prop} = 1.4, M_L = 2.35Kg$.	87
4.20. Planta con fenómenos no lineales sin compensar: Posición y desviación angular; $K_{prop} = 2.5, M_L = 2.35Kg$.	88
4.21. Planta con fenómenos no lineales sin compensar: Señales de control; $K_{prop} = 2.5, M_L = 2.35Kg$.	88
4.22. Planta con fenómenos no lineales sin compensar: Posición y desviación angular; $K_{prop} = 2.5, M_L = 1.35Kg$.	89
4.23. Planta con fenómenos no lineales sin compensar: Señales de control; $K_{prop} = 2.5, M_L = 1.35Kg$.	89
4.24. Planta con compensación de la zona muerta: Posición y desviación angular; $K_{prop} = 1.4, M_L = 2.35Kg$.	90
4.25. Planta con compensación de la zona muerta: Señales de control; $K_{prop} = 1.4, M_L = 2.35Kg$.	90
4.26. Planta con compensación de la zona muerta: Posición y desviación angular; $K_{prop} = 1.4, M_L = 1.35Kg$.	91
4.27. Planta con compensación de la zona muerta: Señales de control; $K_{prop} = 1.4, M_L = 1.35Kg$.	91
4.28. Señales continuas simuladas.	94
4.29. Señales discretizadas a 100 Hz.	95
4.30. Señales discretizadas a 30 Hz.	96
4.31. Señales discretizadas a 15 Hz.	97
4.32. Posición y desviación angular del sistema controlado en tiempo discreto: frecuencia de muestreo de 60 Hz.	97
4.33. Componentes de la señal de control en tiempo discreto: frecuencia de muestreo de 60 Hz.	98
4.34. Posición y desviación angular del sistema controlado en tiempo discreto: frecuencia de muestreo de 30 Hz.	98
4.35. Componentes de la señal de control en tiempo discreto: frecuencia de muestreo de 30 Hz.	99
4.36. Posición y desviación angular del sistema controlado en tiempo discreto: frecuencia de muestreo de 15 Hz.	99
4.37. Componentes de la señal de control en tiempo discreto: frecuencia de muestreo de 15 Hz.	100
4.38. Planta sin fenómenos no lineales: Posición y desviación angular; $K_{prop} = 1.4, M_L = 2.35Kg$.	100
4.39. Planta sin fenómenos no lineales: Señales de control; $K_{prop} = 1.4, M_L = 2.35Kg$.	101
4.40. Planta con fenómenos no lineales sin compensar: Posición y desviación angular; $K_{prop} = 1.4, M_L = 2.35Kg$.	101

4.41. Planta con fenómenos no lineales sin compensar: Señales de control; $K_{prop} = 1.4, M_L = 2.35Kg$	102
4.42. Planta con compensación de la zona muerta: Posición y desviación angular; $K_{prop} = 1.4, M_L = 2.35Kg$	102
4.43. Planta con compensación de la zona muerta: Señales de control; $K_{prop} = 1.4, M_L = 2.35Kg$	103
4.44. Planta con compensación de la zona muerta: Posición y desviación angular; $K_{prop} = 1.4, M_L = 1.35Kg$	103
4.45. Planta con compensación de la zona muerta: Señales de control; $K_{prop} = 1.4, M_L = 1.35Kg$	104
5.1. Respuesta experimental: Posición y desviación angular sin compensación de la zona muerta.	106
5.2. Respuesta experimental: Señal de control sin compensación de la zona muerta.	106
5.3. Respuesta experimental: Posición y desviación angular con y sin compensación de la zona muerta.	107
5.4. Respuesta experimental: Señal de control con y sin compensación de la zona muerta.	108
5.5. Respuesta experimental: Posición y desviación angular con 1.35 kg de carga.	108
5.6. Respuesta experimental: Señal de control con 1.35 kg de carga.	109
5.7. Respuesta experimental: Posición y desviación angular con 2.35 kg de carga.	109
5.8. Respuesta experimental: Señal de control con 2.35 Kg de carga.	110
5.9. Respuesta experimental: Posición y desviación angular con $K_{prop} = 1.4$	111
5.10. Respuesta experimental: Señal de control con $K_{prop} = 1.4$	111
5.11. Respuesta experimental: Posición y desviación angular con $K_{prop} = 2.1$	112
5.12. Respuesta experimental: Señal de control con $K_{prop} = 2.1$	112
5.13. Respuesta experimental: Posición y desviación angular para un recorrido de 1.5m con un generador de trayectoria de dos oscilaciones.	113
5.14. Respuesta experimental: Señal de control para un recorrido de 1.5m con un generador de trayectoria de dos oscilaciones.	113
5.15. Respuesta experimental: Posición y desviación angular para un recorrido de 1.5m con un generador de trayectoria de una oscilación.	114
5.16. Respuesta experimental: Señal de control para un recorrido de 1.5m con un generador de trayectoria de una oscilación.	115
A.1. Organización de la implementación experimental.	124
A.2. Prototipo Grúa-Puente.	125
A.3. Dimensiones del prototipo.	126
A.4. Dimensiones generales de una grúa puente.	126
A.5. Cabezal del prototipo grúa-puente.	127
A.6. Componentes del Actuador.	128

A.7. Circuito del Optoacoplador 4N33.	129
A.8. Esquema del Amplificador de Potencia.	129
A.9. Amplificador de Potencia - Puente H.	130
A.10. Motor-reductor DC GM9236C534-R2.	131
A.11. Configuración del Sensor de Posición del Péndulo.	132
A.12. Configuración del Sensor de Posición del Carro.	133
A.13. Tarjeta de Control DSP - TMS320F240.	133
A.14. Generación de salidas PWM con <i>deadband</i>	136
A.15. Diagrama de Tiempos del Circuito QEP.	136
A.16. Diagrama de Flujo del Algoritmo de Control.	138
B.1. Curva de Velocidad del Motor-Reductor DC.	141
B.2. Gráfico donde se aprecia la zona muerta del subsistema motor-polea.	144
B.3. Cálculo de la fricción: Velocidad del cabezal.	145
B.4. Cálculo de la fricción: Posición del cabezal.	146
B.5. Fuerza de fricción en función de la velocidad.	147
B.6. Respuesta de la planta a una entrada compleja.	151
B.7. Evolución inicial de la población.	152
B.8. Simulación del modelo con la entrada registrada.	152

LISTA DE CUADROS

3.1. Variables del Sistema Grúa-Puente.	47
3.2. Parámetros del Subsistema Cabezal-Péndulo.	48
3.3. Parámetros del Subsistema Motor-Polea.	48
3.4. Otras variables del sistema.	49
3.5. Valores de los parámetros del subsistema cabezal-péndulo.	63
3.6. Valores de los parámetros del subsistema motor-polea.	64
3.7. Valores ajustados por el algoritmo genético.	66
3.8. Valores de los parámetros del subsistema cabezal-péndulo.	66
3.9. Valores de los parámetros del subsistema motor-polea.	67
4.1. Base de reglas del controlador de la posición.	75
4.2. Base de reglas del compensador de la oscilación.	80
A.1. Modos de Operación del Puente H.	131
B.1. Resultados de Ensayos en Laboratorio.	141
B.2. Coeficientes de Fricción μ	146
B.3. Identificación de la fricción: Valores obtenidos a partir de los ex- perimentos.	147
F.1. Listado de Normas Industriales.	204

PRÓLOGO

El presente Informe de Suficiencia busca describir completamente y en detalle el diseño y aplicación experimental de un controlador por Lógica Difusa del tipo Takagi-Sugeno, destinado a compensar las oscilaciones y controlar la posición de la carga en un prototipo a escala reducida de una grúa-puente de contenedores (*containers*).

El prototipo utilizado fue desarrollado como parte de un proyecto de investigación, financiado por el Instituto de Investigación de la Facultad de Ingeniería Mecánica [INIFIM] en el año 2004. La motivación de este proyecto era implementar sistemas de control inteligentes en una aplicación experimental que fuera de interés en el ámbito nacional; en este caso, la mejora sustancial de la eficiencia de nuestras operaciones portuarias. El proyecto fue ejecutado en su totalidad y fue puesto en marcha el mismo año, siendo el autor de este informe uno de los dos investigadores principales que llevaron a cabo el desarrollo del proyecto. Este informe de suficiencia corresponde a la investigación desarrollada durante este proyecto y tras la finalización del mismo.

Este Informe de Suficiencia consta de cinco capítulos y está organizado de la siguiente forma:

El **capítulo 1** abarca la introducción a este Informe de Suficiencia. Se revisa la literatura correspondiente con la automatización de las grúas y se indica la formulación del problema y los objetivos de este Informe de Suficiencia.

El **capítulo 2** es una introducción a la teoría de la Lógica Difusa. Se parte de los conceptos básicos de operaciones con conjuntos difusos y se van desarrollando los temas necesarios para presentar el modelo de inferencia difusa de Takagi-Sugeno.

En el **capítulo 3** se explica el desarrollo del modelo matemático basado en las ecuaciones dinámicas de Euler-Lagrange y su expresión en el espacio de estados. Se trata también la inclusión de los fenómenos no lineales principales: la fricción y la zona muerta del actuador. Finalmente se muestran los valores de los parámetros del modelo y una simulación de la respuesta en lazo abierto del modelo.

El **capítulo 4** expone el desarrollo del diseño del sistema de control, basado en un mecanismo doble de inferencia difusa del tipo Takagi-Sugeno, incluyendo un algoritmo compensador de la zona muerta para mejorar el desempeño del controlador. En este capítulo se presentan diversas simulaciones del sistema de control, las cuales se dividen en dos categorías: en tiempo continuo y en tiempo discreto, incluyendo la simulación de la discretización de los sensores, la influencia del período de muestreo y el efecto de la compensación de la zona muerta.

El **capítulo 5** muestra los resultados obtenidos durante la experimentación, en la que se realizaron diferentes pruebas para observar y mejorar el desempeño del controlador programado en el DSP. Las pruebas incluyen la experimentación con el compensador de la zona muerta, variaciones en la masa de la carga y en el controlador.

Tras la exposición por capítulos, se presentan las **conclusiones y recomendaciones** de esta investigación, donde se proponen temas para investigaciones futuras. En los **apéndices** se incluye información relevante acerca de la implementación experimental para las pruebas en tiempo real del sistema de control, describiendo el prototipo, los sistemas del actuador, de los sensores y la configuración de la tarjeta de control basada en el DSP TMS320F240. Se incluye también

en los apéndices el proceso de identificación de parámetros en detalle, con información adicional correspondiente a la optimización por algoritmos genéticos. Finalmente, se anexan a este informe de suficiencia los programas desarrollados en MATLAB[®], C y ASSEMBLER.

CAPÍTULO 1

INTRODUCCIÓN

Las grúas son ampliamente utilizadas para transportar cargas pesadas en diversas instalaciones industriales y comerciales; principalmente en puertos, astilleros, operaciones de construcción e instalaciones de industria pesada. En general, las grúas aceleran la producción de los procesos en los que intervienen y son indispensables en muchos de éstos. Particularmente, en los puertos marítimos, el uso de grúas-puente (o grúas pórtico) para la manipulación de contenedores aumenta la productividad y reduce el tiempo de permanencia de los buques.



Figura 1.1: Vista Panorámica del Puerto de Singapur

En las figuras 1.1 y 1.2 se muestra una fotografía panorámica y otra parcial del Puerto de Singapur (PSA Singapore Terminals), que en el 2009 fue el puerto de contenedores con mayor actividad a nivel mundial, con un volumen de 25'866 600 TEUs¹, [1]. En una terminal de contenedores, las actividades de carga y descarga son de vital importancia y deben ser realizadas con la máxima

¹ *Twenty-foot equivalent unit* por sus siglas en inglés, es una unidad de carga equivalente a la capacidad de un contenedor de 20 pies



Figura 1.2: Vista Parcial del Puerto de Singapur

eficiencia. El presente trabajo se orienta a la automatización de estas actividades, concentrándose en el control automático de la operación de traslado de la carga.

La operación de las grúas puede ser dividida en cinco pasos: agarre, izado, traslado de un punto a otro, descenso y liberación de la carga. Las operaciones de traslado de punto a punto y la manipulación fina de la carga son las tareas que mayor tiempo consumen, aproximadamente un 44 % y 35 % respectivamente, [2], y requieren de un operador muy hábil para que sean cumplidas segura y eficientemente. Por esto, una buena parte de la investigación actual se concentra en la implementación de métodos apropiados para facilitar el movimiento de la carga sin inducir grandes oscilaciones. En la figura 1.3 se muestra un muelle de carga y descarga de contenedores con grúas-puente en operación.

La automatización de la operación de traslado implica el control de la posición y la amortiguación de oscilaciones de la carga. Un aspecto fundamental a considerar es que las grúas son sistemas subactuados, vale decir, tienen más grados de libertad que entradas de control. Con el fin de validar el diseño, se prueba



Figura 1.3: Grúas-puente en operación el puerto de Savannah, Georgia (USA).

el sistema de control en un prototipo que presente efectos no-lineales similares a los de las grúas-puente industriales.

En este estudio se diseña e implementa un controlador basado en un modelo no lineal de las grúas puente, considerando el modelo específico del prototipo construido para la experimentación. Se incluye en este modelo el efecto de la fricción, con el fin de que las simulaciones se aproximen eficazmente al comportamiento del prototipo real. Para el control de la posición del cabezal y la amortiguación de la oscilación se emplea un controlador basado en lógica difusa con un compensador de la zona muerta; cuya salida es opuesta a la estimación del efecto de la zona muerta del motor-reductor DC.

1.1. Antecedentes: Evolución del Control de las Grúas-Puente

Hasta hace pocos años, las grúas han venido siendo operadas manualmente; pero, principalmente desde el comienzo del presente siglo, se han venido desarro-

llando diversos intentos para automatizar estas maquinarias en forma efectiva. Con las nuevas capacidades del hardware actual y el avance de la ingeniería de control, así como la necesidad cada vez mayor de su automatización, el control de las grúas-puente para tareas repetitivas de traslado de contenedores se ha vuelto una exigencia en el marco de la competitividad de los puertos, que buscan ofrecer menores tiempos de carga y descarga a sus clientes.

Este panorama ha impulsado el trabajo de diversos investigadores a nivel mundial, sin embargo la mayoría de los esquemas de control no han sido apropiados en la práctica, debido principalmente a que en un inicio los intentos se basaban en técnicas convencionales de control que requieren de modelos matemáticos detallados, pero las no linealidades de estos sistemas tienen efectos importantes y son difíciles de modelar. Lo anterior a dado lugar al empleo de técnicas de control avanzado e inteligente por parte de la comunidad de ingenieros involucrados. A continuación se revisarán los enfoques generales utilizados en este campo.

La automatización de las grúas puede dividirse en dos enfoques. En el primero, el operador es mantenido en el bucle de control y la dinámica de la carga es modificada para hacer su trabajo más fácil. Una forma consiste en añadir amortiguación al retroalimentar el ángulo de oscilación de la carga y su velocidad, o retroalimentando una versión retrasada del ángulo de oscilación, [3]. Esta retroalimentación añade una trayectoria extra a la generada por el operador, pero da lugar a un atraso en la acción de control que suele confundir al operador. Otra forma de modificar la dinámica para facilitar el trabajo se basa en añadir un sistema mecánico de absorción a la estructura de la grúa, [4]. Implementar este método requiere de una considerable potencia extra, lo que vuelve este método impráctico.

En el segundo enfoque, se excluye al operador del bucle de control y la operación es completamente automatizada. Esto se puede hacer mediante varias técnicas. La primera se basa en la generación de trayectorias para transferir la

carga a su destino con una mínima oscilación. La segunda técnica se basa en la retroalimentación de la posición y del ángulo de oscilación. La tercera se basa en la división del diseño del controlador en dos partes: un controlador para la amortiguación de las oscilaciones y un controlador de rastreo de la trayectoria. Cada uno se diseña separadamente y se combinan para asegurar el desempeño y estabilidad del sistema global.

Considerando que la oscilación de la carga es afectada por la aceleración del movimiento, muchos investigadores se han concentrado en la generación de trayectorias, que despachen la carga el menor tiempo posible y que a la vez minimicen la oscilación. Estas trayectorias son obtenidas generalmente utilizando técnicas de optimización. La función objetivo puede ser el tiempo de traslado, la acción de control, o el ángulo de oscilación. Otro método importante de generación de trayectorias es el desarrollo de la trayectoria a través de una secuencia de pulsos de aceleración y desaceleración. Estas secuencias se generan de modo tal que evite oscilaciones residuales al final de la operación de traslado, [5]. El controlador resultante es de lazo abierto, lo que hace que el sistema se vea afectado por las perturbaciones y variaciones en los parámetros; además la acción de control es discontinua y requiere que no existan oscilaciones al inicio del movimiento.

En general, los métodos de control en lazo cerrado son mucho más apropiados para el control de las grúas. Algunos investigadores ([6], [7]) han buscado controlar este sistema en forma global, aplicando estrategias de control de programación de ganancias (*gain scheduling*) y retroalimentación total de estados (*full-state feedback*). En Perú, se han llevado a cabo investigaciones al respecto empleando estrategias de control óptimo y por ubicación de polos, [8]. Además de las estrategias de control en lazo cerrado que buscan controlar la totalidad de los estados mediante un solo controlador, se puede emplear un sistema con dos controladores en paralelo. Estos sistemas tienen la ventaja de poder combinarse con un diseño conveniente de las trayectorias al destinar un controlador al se-

guimiento del recorrido deseado y otro para la compensación de las oscilaciones. Algunas investigaciones acerca de estos controladores dobles para el caso de grúas montadas en barcos se han realizado con buenos resultados, [3]. Dentro de esta categoría se encuentra el controlador que se propone en este informe, empleando controladores basados en sistemas de inferencia difusa del tipo Takagi-Sugeno.

1.2. Formulación del Problema

El problema que busca resolver este informe de suficiencia se enuncia a continuación: "Dado el sistema Grúa-Puente, se desea diseñar un sistema de control por lógica difusa, el cual deberá controlar simultáneamente la posición del cabezal para una referencia arbitraria y compensar las oscilaciones en la varilla sólo considerando al sistema en dos dimensiones, en donde la varilla oscila en el mismo plano en que se mueve el cabezal, con un mínimo sobreimpulso, un mínimo tiempo de establecimiento y un error en estado estable nulo".

El sistema Grúa-Puente está conformado por un cabezal que se traslada horizontalmente a lo largo de un riel, en este cabezal se encuentra una varilla con un extremo articulado en el centro de gravedad del cabezal y con el otro extremo acoplado a una carga. Este cabezal desarrolla su movimiento de traslación por la acción de dos poleas y una cadena. Una de las poleas está acoplada al eje de un motor-reductor DC. Este último gobierna el movimiento del cabezal.

La implementación del proceso consigna el uso de la tarjeta de control DSP Texas Instruments TMS320F240, que se encarga de manejar los datos de las entradas digitales de los sensores y dar la referencia de voltaje modulada en ancho de pulso al amplificador que alimenta al motor-reductor DC.

1.3. Objetivos Generales

El propósito de este informe de suficiencia es desarrollar un sistema de control por lógica difusa que resuelva el problema de control propuesto en la sección anterior. Los objetivos generales de esta investigación son:

- Diseñar una ley de control por lógica difusa que resuelva el problema de control propuesto.
- Obtener como respuesta del sistema controlado, un tiempo de establecimiento corto, un mínimo sobreimpulso, un error en estado estable muy cercano a cero. En el capítulo 4 se definirán los valores admisibles para los errores en estado estable, para el sobreimpulso y para el tiempo de establecimiento.
- Implementar la ley de control en un prototipo del sistema Grúa-Puente, utilizando la tarjeta de control DSP TMS320F240.

CAPÍTULO 2

TEORÍA DE CONTROL POR LÓGICA DIFUSA

2.1. Introducción a la Teoría de la Lógica Difusa

Los seres humanos podemos realizar diversas tareas altamente complejas sobre sistemas tan sofisticados como nuestros cuerpos o desconocidos y cambiantes como el medio ambiente que nos rodea, de manera precisa y eficiente. Controlamos la mayor parte de nuestros movimientos corporales de manera inconsciente a través del sistema nervioso, y realizamos otras actividades conscientemente a través de una evaluación aproximada basada en información inexacta. Sin embargo nuestro desempeño como sistemas de control es muy superior al de cualquier sistema de control digital desarrollado hasta ahora. Bajo esta perspectiva muchos investigadores han buscado desarrollar teorías matemáticas que describan o modelen la inteligencia humana y animal; este campo de estudio recibe el nombre de Inteligencia Artificial. Estas nuevas teorías buscan desarrollar algoritmos que tengan la capacidad de obtener información, tomar decisiones a través del razonamiento aproximado, *aprender* y evolucionar.

La toma de decisiones a través del razonamiento aproximado es una de las aplicaciones principales de la Lógica Difusa. Esta toma de decisiones, realizada en forma continua sobre un dominio, viene a ser una forma de control automático.

La teoría de la lógica difusa o teoría de conjuntos difusos provee un sistema de cálculo sistemático para para procesar lingüísticamente la información imprecisa

e incompleta que recibe nuestro cerebro a través de nuestros órganos sensoriales, y permite aplicar un procesamiento numérico de esta información mediante el uso de etiquetas lingüísticas estipuladas por una función de membresía.

2.1.1. Conjuntos Difusos

A diferencia de un conjunto clásico, un conjunto difuso no tiene límites nítidos. Vale decir que la transición entre pertenecer y no pertenecer a un conjunto es gradual, y esta transición suave está caracterizada por funciones de membresía que dan a los conjuntos difusos flexibilidad para modelar expresiones lingüísticas comúnmente usadas, tales como “El agua está caliente” o “la temperatura es alta”.

Definición 2.1 *Conjuntos difusos y funciones de membresía*

Si X es una colección de objetos denotados genéricamente por x , entonces un conjunto difuso A en X está definido por un conjunto de pares ordenados:

$$A = \{(x, \mu_A(x)) \mid x \in X\}, \quad (2.1)$$

*donde $\mu_A(x)$ es llamada la **función de membresía** (MF^1) para el conjunto difuso A . La MF hace un mapeo de cada elemento de X asignándole un nivel de pertenencia entre 0 y 1.*

Obviamente, la definición de un conjunto difuso es una simple extensión de la definición de un conjunto clásico en el que la función característica puede tener cualquier valor entre 0 y 1. Si el valor de la función de membresía $\mu_A(x)$ está restringido a ser 1 ó 0, entonces A se reduce a un conjunto clásico y $\mu_A(x)$ es la función característica de A . Para mayor claridad, se considerarán equivalentes las denominaciones de conjuntos clásicos, conjuntos ordinarios, conjuntos no-difusos, o simplemente conjuntos.

¹MF: Por sus siglas en inglés (*Membership Function*).

Usualmente, se trata a X como el “universo de discurso”, o simplemente como el “universo”, y puede consistir de objetos discretos o de un espacio continuo. Puede observarse gráficamente en la figura 2.1 dos tipos de conjuntos difusos, uno discreto y otro continuo.

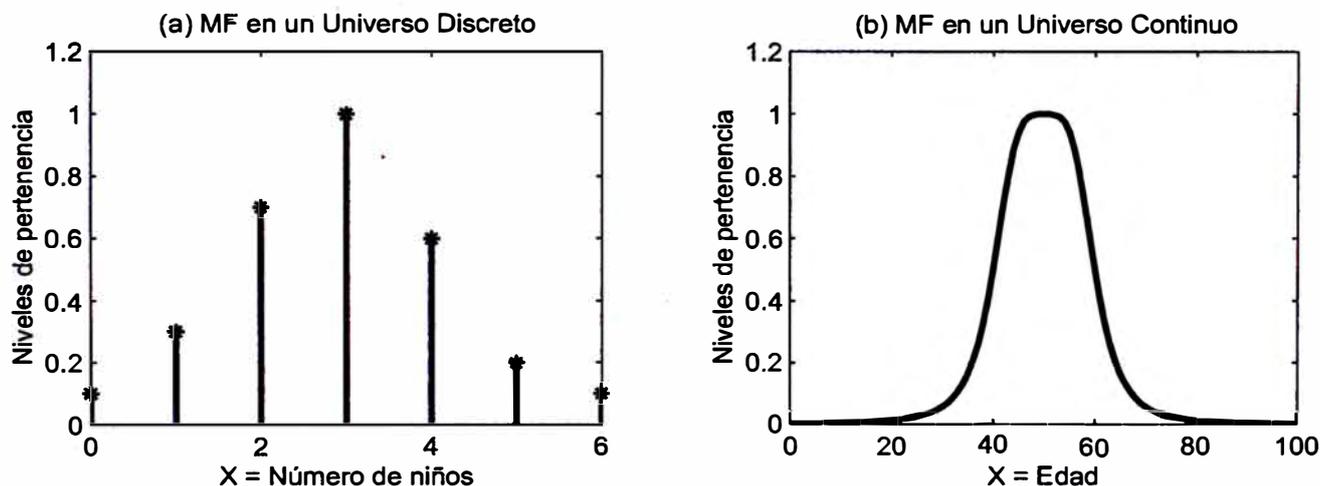


Figura 2.1: (a) A = “número de niños en una familia”; (b) B = “alrededor de 50 años de edad”.

Definición 2.2 Soporte

El soporte de un conjunto difuso A es el conjunto de todos los puntos x en X tales que $\mu_A(x) > 0$:

$$\text{support}(A) = \{x \mid \mu_A(x) > 0\}. \quad (2.2)$$

Definición 2.3 Núcleo

El núcleo de un conjunto difuso A es el conjunto de todos los puntos x en X tales que $\mu_A(x) = 1$:

$$\text{core}(A) = \{x \mid \mu_A(x) = 1\}. \quad (2.3)$$

Definición 2.4 Normalidad

Un conjunto difuso A es normal si su núcleo es no vacío. En otras palabras, se puede siempre encontrar un punto $x \in X$ tal que $\mu_A(x) = 1$.

Definición 2.5 Puntos de cruce

Un punto de cruce de un conjunto difuso A es un punto $x \in X$ en el que $\mu_A(x) = 0,5$:

$$\text{crossover}(A) = \{x \mid \mu_A(x) = 0,5\}. \quad (2.4)$$

Definición 2.6 Singleton Difuso

Un conjunto difuso cuyo soporte es un punto único en el que X con $\mu_A(x) = 1$ es llamado un singleton difuso.

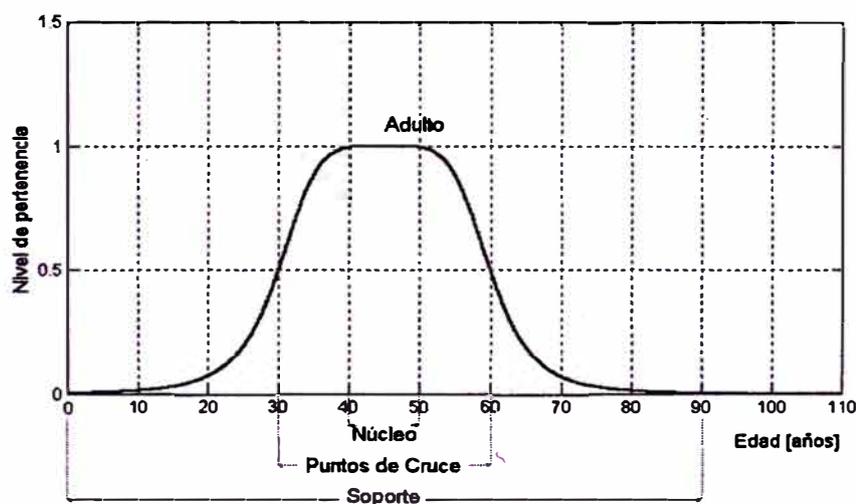


Figura 2.2: Núcleo, soporte y puntos de cruce del conjunto difuso "Adulto".

En la figura 2.2 se ilustran los núcleos, soportes y puntos de cruce de la función de membresía con forma de campana que representa al grupo "Adulto".

Operaciones con Conjuntos Difusos

En relación con las operaciones ordinarias de conjuntos, unión, intersección y complemento, los conjuntos difusos tienen operaciones similares, que fueron inicialmente definidas por Zadeh en [9]. Antes de introducir estas tres operaciones con conjuntos difusos, se definirá la noción de subconjunto, que tiene un rol fundamental tanto para los conjuntos ordinarios como para los difusos.

Definición 2.7 *Pertenencia o subconjunto*

Un conjunto difuso A está contenido en un conjunto difuso B (o, equivalentemente, A es un subconjunto de B , o A es menor o igual que B) si y sólo si $\mu_A(x) \leq \mu_B(x)$ para todo x . Simbólicamente:

$$A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x). \quad (2.5)$$

La figura 2.3 ilustra el concepto de $A \subseteq B$.

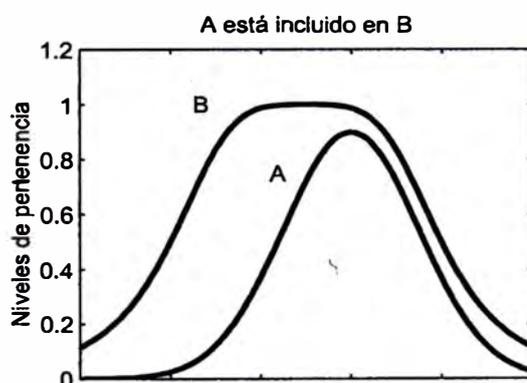


Figura 2.3: El concepto de $A \subseteq B$.

Definición 2.8 *Unión o Disyunción*

La unión de dos conjuntos difusos A y B es un conjunto difuso C , escrito como $C = A \cup B$ o $C = A$ OR B , cuya MF está relacionada con las de A y B por

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x). \quad (2.6)$$

Definición 2.9 *Intersección o Conjunción*

La intersección de dos conjuntos difusos A y B es un conjunto difuso C , escrito como $C = A \cap B$, o $C = A \text{ AND } B$, cuya MF está relacionada con las de A y B por

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x). \quad (2.7)$$

Definición 2.10 *Complemento*

El complemento de un conjunto difuso A , denotado por \bar{A} ($\neg A$, NOT A), se define como

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x). \quad (2.8)$$

La figura 2.4 muestra estas tres operaciones básicas: La figura 2.4 (a) ilustra dos conjuntos difusos A y B ; la figura 2.4 (b) es el complemento de A ; la figura 2.4 (c) es la unión de A y B ; la figura 2.4 (d) muestra la intersección de A y B .

Definición 2.11 *Producto Cartesiano*

Sean A y B conjuntos difusos en X y Y , respectivamente. El producto cartesiano de A y B , denotado por $A \times B$, es un conjunto difuso el espacio $X \times Y$ con la función de membresía

$$\mu_{A \times B}(x, y) = \min(\mu_A(x), \mu_B(y)). \quad (2.9)$$

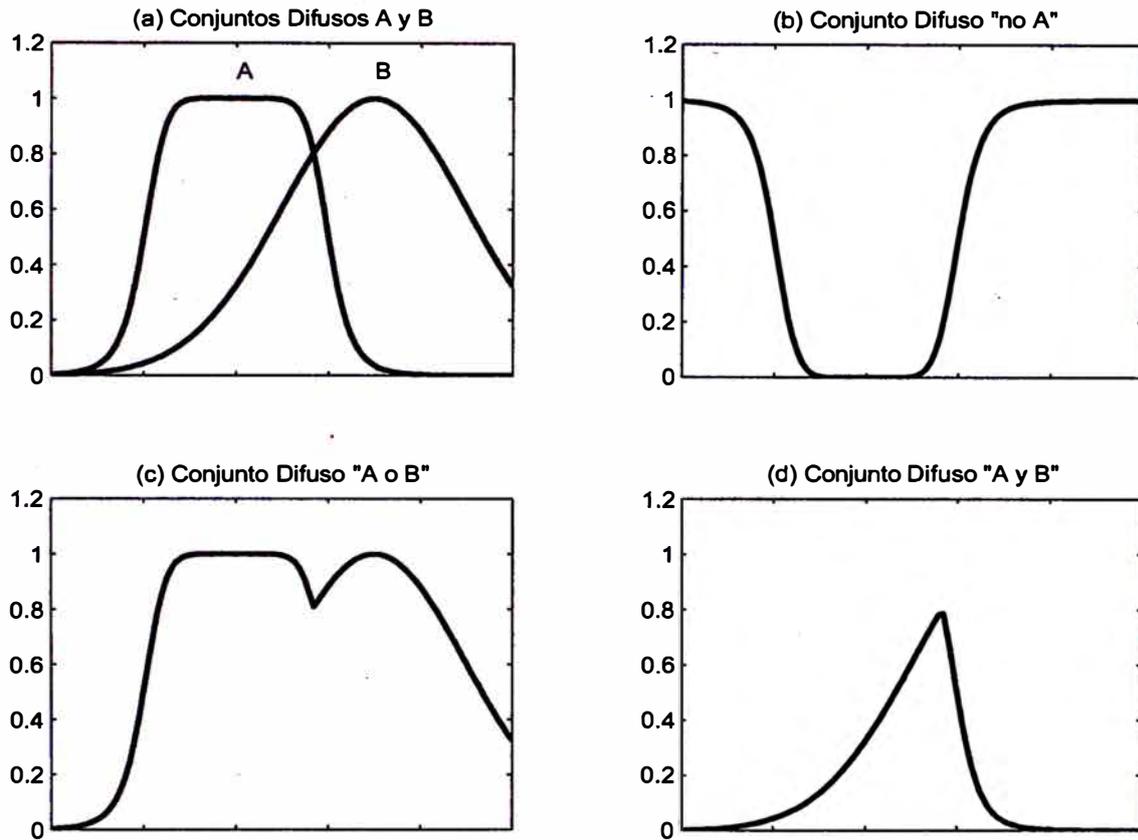


Figura 2.4: Operaciones con conjuntos difusos: (a) dos conjuntos difusos A y B ; (b) \bar{A} ; (c) $A \cup B$; (d) $A \cap B$.

Similarmente, el co-producto cartesiano $A + B$ es un conjunto difuso con la función de membresía

$$\mu_{A+B}(x, y) = \max(\mu_A(x), \mu_B(y)). \quad (2.10)$$

Formulación y Parametrización de las Funciones de Membresía

Como se mencionó anteriormente, un conjunto difuso queda completamente descrito por su función de membresía. Considerando el hecho de que la mayoría de conjuntos difusos que se emplean en la práctica tienen un universo de discurso en variable real, la definición de las funciones de membresía se hace en términos de fórmulas matemáticas. Una función de membresía puede tener un dominio

multidimensional, sin embargo, se mostrará tan sólo los tipos más comunes de funciones de una dimensión.

Definición 2.12 *Función de Membresía Triangular*

Una MF triangular queda definida por tres parámetros $\{a, b, c\}$ como sigue:

$$\text{triangle}(x; a, b, c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x. \end{cases} \quad (2.11)$$

Al usar \min y \max , se obtiene una expresión alternativa para la ecuación anterior:

$$\text{triangle}(x; a, b, c) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (2.12)$$

Los parámetros $\{a, b, c\}$ (con $a < b < c$) determinan las coordenadas x de los tres vértices de la MF triangular. En la figura 2.5 (a) se ilustra una MF triangular definida por $\text{triangle}(x; 20, 60, 80)$.

Definición 2.13 *Función de Membresía Trapezoidal*

Una MF triangular queda definida por cuatro parámetros $\{a, b, c, d\}$ como sigue:

$$\text{trapezoid}(x; a, b, c, d) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ 1, & b \leq x \leq c. \\ \frac{d-x}{d-c}, & c \leq x \leq d. \\ 0, & d \leq x. \end{cases} \quad (2.13)$$

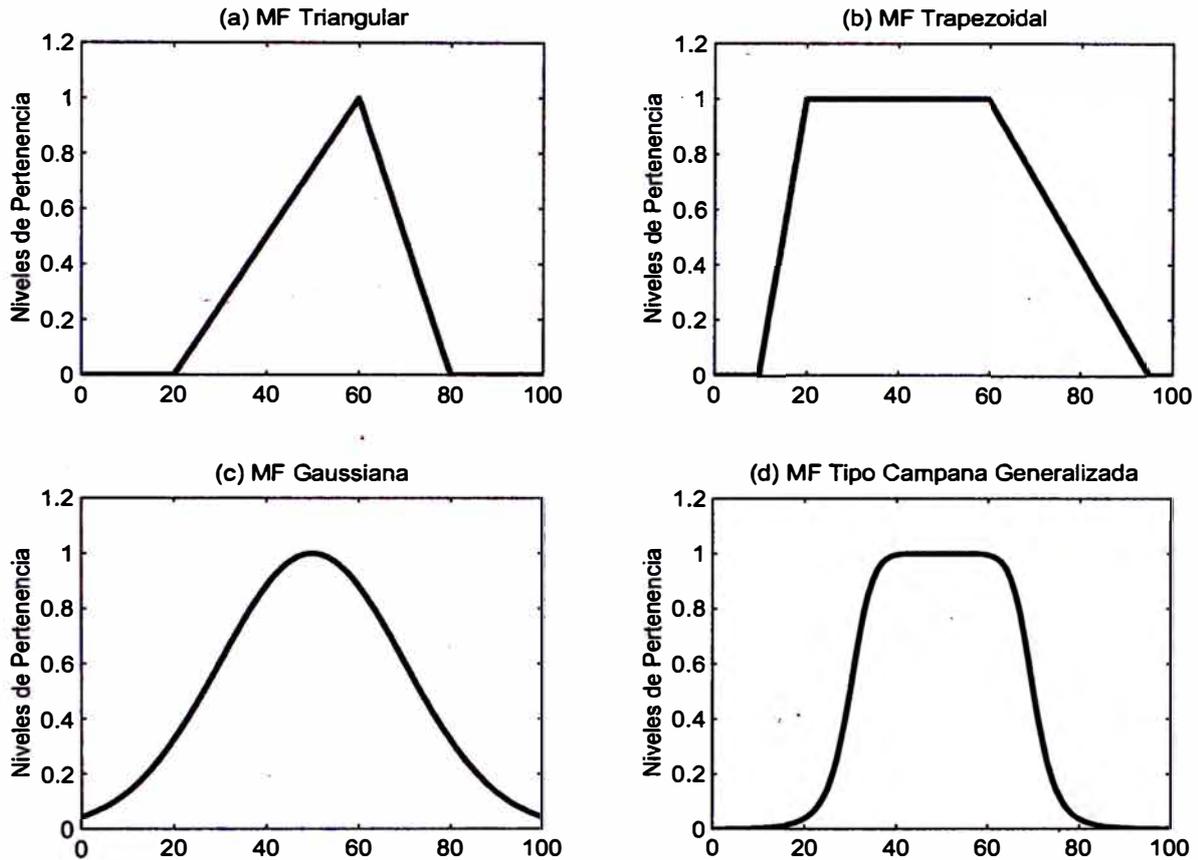


Figura 2.5: Ejemplos de cuatro clases de MF parametrizadas: (a) $triangle(x; 20, 60, 80)$; (b) $trapezoid(x; 10, 20, 60, 95)$; (c) $gaussian(x; 50, 20)$; (d) $bell(x; 20, 4, 50)$.

Una concisa forma alternativa, usando \min y \max , es

$$trapezoid(x; a, b, c, d) = \max \left(\min \left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right) \quad (2.14)$$

Los parámetros $\{a, b, c, d\}$ (con $a < b \leq c < d$) determinan las coordenadas x de los tres vértices de la MF trapezoidal. En la figura 2.5 (b) se ilustra una MF trapezoidal definida por $trapezoid(x; 10, 20, 60, 95)$. Se puede observar que una función trapezoidal se reduce a una triangular cuando b es igual a c .

Debido a su simple formulación y eficiencia computacional, tanto las MFs triangulares como trapezoidales han sido usadas extensamente, especialmente en implementaciones en tiempo real. sin embargo, debido a que están compuestas por segmentos rectos, no son suaves en los vértices especificados por los parámetros. Se muestran ahora otras funciones de membresía definidas por funciones suaves y no lineales.

Definición 2.14 *Función de Membresía Gaussiana*

Una MF gaussiana queda definida con dos parámetros $\{c, \sigma\}$:

$$\text{gaussian}(x; c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}. \quad (2.15)$$

Una MF gaussiana es determinada completamente por c y σ ; c representa el centro y σ el ancho de la MF. La figura 2.5 (c) muestra una MF gaussiana definida por $\text{gaussian}(x;50,20)$.

Definición 2.15 *Función de Membresía Tipo Campana Generalizada*

Una MF tipo campana generalizada (o tipo campana) queda definida por tres parámetros $\{a, b, c\}$:

$$\text{bell}(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}. \quad (2.16)$$

El parámetro b usualmente es positivo. Si b fuera negativo, la forma de esta MF sería la de una campana invertida. Esta MF es una generalización de la distribución de Cauchy, usada en la teoría probabilística, por lo que suele llamarse también como función de membresía de Cauchy.

La figura 2.5 (d) ilustra una MF tipo campana definida por $\text{bell}(x; 20, 4, 50)$. Una MF tipo campana particular puede ser obtenida mediante una selección

apropiada del conjunto de parámetros $\{a, b, c\}$. Específicamente, podemos ajustar c y a para variar el centro y el ancho de la MF, y luego usar b para controlar la pendiente de la transición en los puntos de cruce. La figura 2.6 muestra los efectos que produce cambiar cada parámetro.

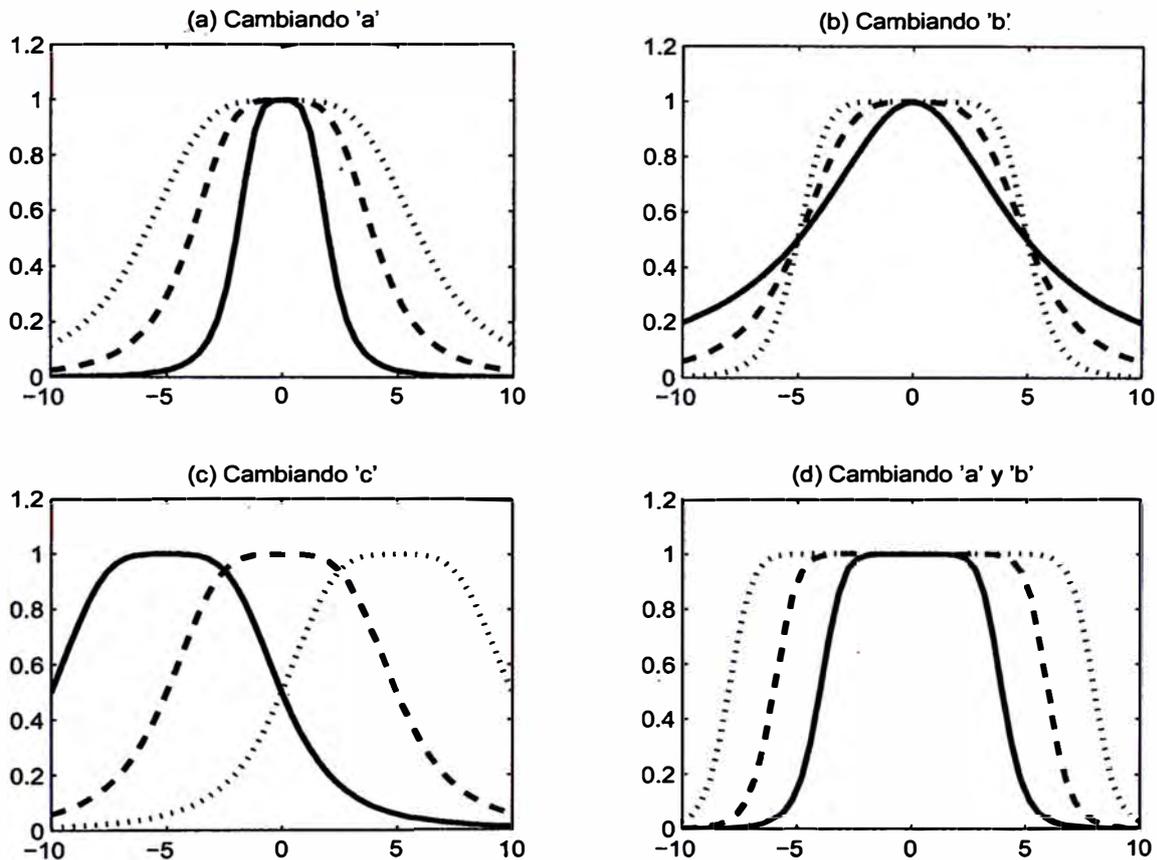


Figura 2.6: (a) Dos funciones sigmoideas y_1 y y_2 ; (b) una MF cerrada obtenida con $|y_1 - y_2|$; (c) Dos funciones sigmoideas y_1 y y_3 ; (d) una MF cerrada obtenida con $y_1 y_3$.

Debido a su suavidad y concisa notación, las MF gaussiana y tipo campana son muy utilizadas al definir conjuntos difusos. En este informe se emplean las MF gaussianas junto con las sigmoideas que se tratan en la siguiente definición. Las funciones gaussianas tienen propiedades muy útiles, tales como la invariancia ante la multiplicación y la transformada de Fourier. El producto de dos gaussianas es una gaussiana con un factor de escala y la transformada de Fourier de una gaussiana es también una gaussiana.

Aunque las MF gaussianas y tipo campana satisfacen el requisito de ser funciones suaves, son incapaces de definir funciones MFs asimétricas, que son importantes en muchas aplicaciones. Para esto, las funciones sigmoideas pueden resultar muy convenientes.

Definición 2.16 *Función de Membresía Sigmoideal*

Una MF sigmoideal queda definida por dos parámetros $\{a, c\}$:

$$\text{sig}(x; a, c) = \frac{1}{1 + e^{-a(x-c)}}, \quad (2.17)$$

donde a controla la pendiente en el punto de cruce $x = c$.

Dependiendo del signo del parámetro a , una MF sigmoideal es inherentemente abierta hacia la derecha o hacia la izquierda. Estas funciones son apropiadas para representar conceptos tales como “muy grande” o “muy negativo”. Las funciones sigmoideas pueden expresar funciones de membresía cerradas y asimétricas al realizar operaciones sobre éstas.

En la figura 2.7 se muestra como las funciones sigmoideas pueden dar lugar a MFs cerradas y asimétricas mediante el uso de la resta y la multiplicación.

2.1.2. Reglas Difusas y Razonamiento Difuso

Las reglas difusas y el razonamiento difuso son los elementos fundamentales de los sistemas de inferencia difusa, que son la herramienta de modelado más importante basada en la teoría de conjuntos difusos. Estos han sido exitosamente aplicados a muchas áreas, tales como el control automático, los sistemas expertos, el reconocimiento de patrones, la predicción de series cronológicas y la clasificación de datos.

Dentro de la teoría de conjuntos difusos, existe un concepto básico que provee un procedimiento general para extender los dominios *crisp* a los dominios difusos. Este concepto es el **principio de extensión**, que generaliza un común mapeo

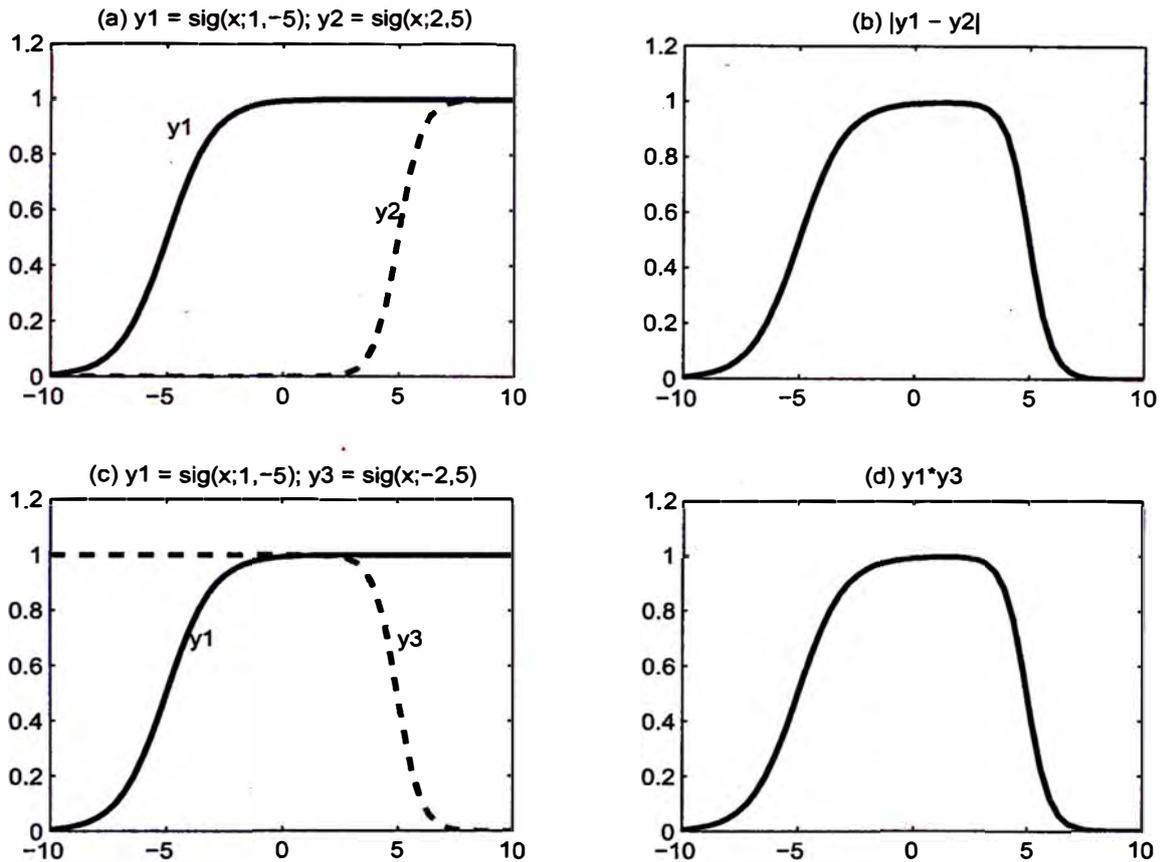


Figura 2.7: (a) Dos funciones sigmoidales y_1 y y_2 ; (b) una MF cerrada obtenida con $|y_1 - y_2|$; (c) Dos funciones sigmoidales y_1 y y_3 ; (d) una MF cerrada obtenida con $y_1 y_3$.

punto a punto de una función $F(\cdot)$ a un mapeo entre conjuntos difusos. Más específicamente, si suponemos que f es una función de X a Y y A es un conjunto difuso en X definido como

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n.$$

Entonces, el principio de extensión establece que la imagen de un conjunto difuso A bajo el mapeo $f(\cdot)$ puede ser expresado como un conjunto difuso B ,

$$B = f(A) = \mu_A(x_1)/y_1 + \mu_A(x_2)/y_2 + \dots + \mu_A(x_n)/y_n,$$

donde $y_i = f(x_i)$, $i = 1, \dots, n$. En otras palabras, el conjunto difuso B puede ser definido a través de los valores de $f(\cdot)$ en x_1, \dots, x_n . Si $f(\cdot)$ es un mapeo *MISO*, entonces existe $x_1, x_2 \in X$, $x_1 \neq x_2$, tal que $f(x_1) = f(x_2) = y^*$, $y^* \in Y$. En este caso, el nivel de pertenencia de B en $y = y^*$ puede obtenerse a partir tanto de $x = x_1$ o $x = x_2$. Más generalmente, tenemos

$$\mu_B(y) = \max_{x=f^{-1}(y)} \mu_A(x),$$

Formalmente, el principio de extensión se define como sigue:

Definición 2.17 *Principio de Extensión*

Sea la función f un mapeo desde el espacio n -dimensional de un producto cartesiano $X_1 \times X_2 \times \dots \times X_n$ a un universo unidimensional Y tal que $y = f(x_1, \dots, x_n)$, y sean A_1, \dots, A_n n conjuntos difusos en X_1, \dots, X_n respectivamente. Luego, el principio de extensión asevera que el conjunto difuso B inducido por el mapeo f queda definido por

$$\mu_B(y) = \begin{cases} \max_{(x_1, \dots, x_n), (x_1, \dots, x_n) = f^{-1}(y)} [\min_i \mu_{A_i}(x_i)], & \text{si } f^{-1}(y) \neq \emptyset. \\ 0, & \text{si } f^{-1}(y) = \emptyset. \end{cases} \quad (2.18)$$

En la figura 2.8 se aprecia una explicación gráfica del principio de extensión.

Relaciones Difusas

Las relaciones difusas binarias son conjuntos difusos en $X \times Y$ que asignan a cada elemento en $X \times Y$ un nivel de pertenencia entre 0 y 1. En particular, las relaciones difusas unitarias son conjuntos difusos con MFs unidimensionales, las relaciones difusas binarias son conjuntos difusos con MFs bidimensionales, y

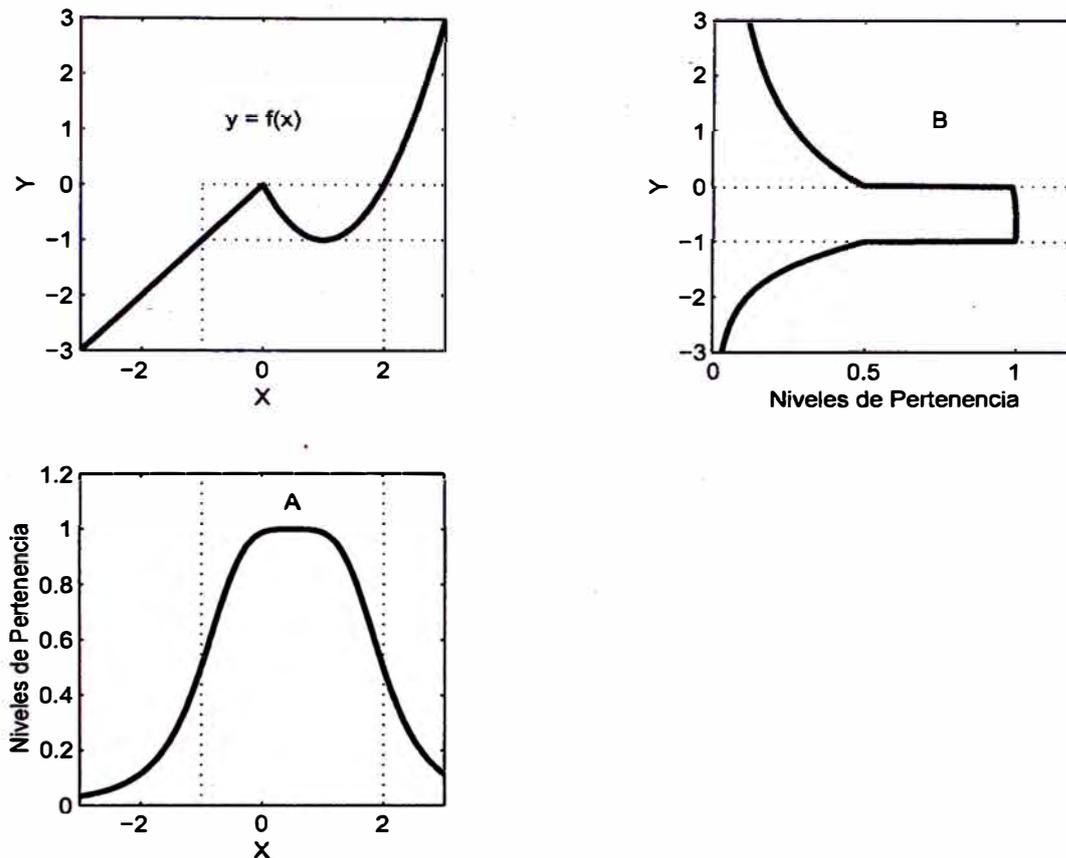


Figura 2.8: Principio de extensión sobre conjuntos difusos con universos continuos. El gráfico inferior representa al conjunto difuso A ; el superior izquierdo es la función $y = f(x)$; y el superior derecho es el conjunto difuso B inducido a través del principio de extensión.

así sucesivamente. El controlador propuesto en este trabajo basa sus sistemas de inferencia difusa en relaciones difusas binarias.

Definición 2.18 *Relación Difusa Binaria*

Sean X y Y dos universos de discurso. Luego,

$$\mathcal{R} = \{((x, y), \mu_{\mathcal{R}}(x, y)) \mid (x, y) \in X \times Y\} \quad (2.19)$$

es una relación difusa binaria en $X \times Y$.

Definición 2.19 *Composición max-min*

Sean \mathcal{R}_1 y \mathcal{R}_2 dos relaciones difusas definidas en $X \times Y$ y $Y \times z$, respec-

tivamente. La composición max-min de \mathcal{R}_1 y \mathcal{R}_2 es un conjunto difuso definido por

$$\mathcal{R}_1 \circ \mathcal{R}_2 = \{[(x, z), \max_y \min(\mu_{\mathcal{R}_1}(x, y), \mu_{\mathcal{R}_2}(y, z))] \mid x \in X, y \in Y, z \in Z\} \quad (2.20)$$

o, equivalentemente,

$$\begin{aligned} \mu_{\mathcal{R}_1 \circ \mathcal{R}_2}(x, z) &= \max_y \min[\mu_{\mathcal{R}_1}(x, y), \mu_{\mathcal{R}_2}(y, z)] \\ &= \bigvee_y [\mu_{\mathcal{R}_1}(x, y) \wedge \mu_{\mathcal{R}_2}(y, z)] \end{aligned} \quad (2.21)$$

considerando que \bigvee y \wedge representan \max y \min , respectivamente.

Definición 2.20 Composición max-producto

Asumiendo la misma notación usada en la definición de la composición max-min, puede definirse la composición max-producto como sigue:

$$\mu_{\mathcal{R}_1 \circ \mathcal{R}_2}(x, z) = \max_y [\mu_{\mathcal{R}_1}(x, y) \mu_{\mathcal{R}_2}(y, z)] \quad (2.22)$$

Variables Lingüísticas

El concepto de variable lingüística es importante en la medida que sirve para dar una aproximación al modelado matemático de cantidades expresadas en forma imprecisa. De acuerdo con Zadeh: “Conforme la complejidad de un sistema aumenta, nuestra habilidad para hacer declaraciones precisas y a la vez significativas disminuye, hasta que un nivel umbral es alcanzado, a partir del cual precisión y relevancia se vuelven características casi mutuamente exclusivas” [10]. La aproximación proporcionada por las variables lingüísticas sirve para resumir la información y expresarla en términos de conjuntos difusos, en vez de números *crisp*.

Definición 2.21 Variables Lingüísticas

Una variable lingüística queda definida por $(x, T(x), X, G, M)$ en el que x es el nombre de la variable; $T(x)$ es el conjunto de términos de x (conjunto de los valores lingüísticos o términos lingüísticos); X es el universo de discurso; G es una regla sintáctica que genera los términos en $T(x)$; y M es una regla semántica que asocia con cada valor lingüístico A su significado $M(A)$, donde $M(A)$ denota un conjunto difuso en X .

En la figura 2.9 se aprecia las funciones de membresía correspondientes con el conjunto de términos de la variable lingüística *edad*.

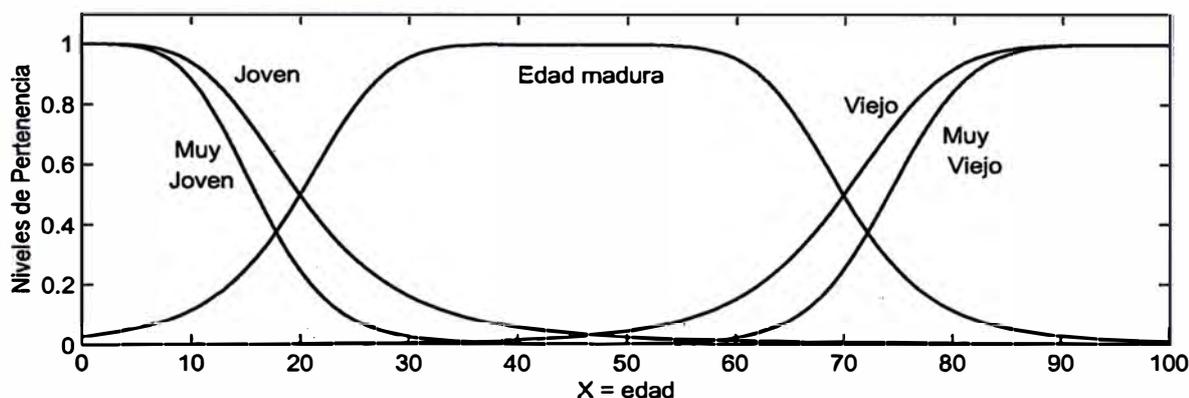


Figura 2.9: Funciones de membresía típicas del conjunto de términos $T(\text{edad})$.

Definición 2.22 Ortogonalidad

Un conjunto de términos $T = t_1, \dots, t_n$ de una variable lingüística x en el universo X es ortogonal si satisface la siguiente propiedad:

$$\sum_{i=1}^n \mu_{t_i}(x) = 1, \forall x \in X \quad (2.23)$$

donde los t_i 's son conjuntos difusos convexos y normales definidos sobre X y estos conjuntos difusos conforman el conjunto de términos T .

Reglas de Implicación Difusa

Una regla de implicación difusa (también conocida como regla difusa, implicación difusa, o sentencia condicional difusa) asume la forma

$$\text{Si } x \text{ es } A, \text{ entonces } y \text{ es } B \quad (2.24)$$

donde A y B son valores lingüísticos definidos por conjuntos difusos en los universos de discurso X y Y , respectivamente. Frecuentemente la proposición “ x es A ” se denomina como llamada el antecedente o la premisa, mientras que la proposición “ y es B ”, el consecuente o conclusión.

Es necesario formalizar el significado de la expresión “si x es A entonces y es B ”, abreviada como $A \rightarrow B$. En esencia, la expresión describe una relación difusa binaria \mathcal{R} entre las variables x y y , dentro del espacio $X \times Y$. Genéricamente, existen dos maneras de interpretar $A \rightarrow B$. Si se interpreta como A **asociado con** B , entonces

$$\mathcal{R} = A \rightarrow B = A \times B = \int_{X \times Y} \mu_A(x) \tilde{*} \mu_B(y) / (x, y), \quad (2.25)$$

donde $\tilde{*}$ es un operador T-norm. Por otro lado, si $A \rightarrow B$ se interpreta como A **implica** B , entonces esta expresión puede ser descrita por cuatro fórmulas diferentes:

- Implicación material:

$$\mathcal{R} = A \rightarrow B = \neg A \cup B. \quad (2.26)$$

- Cálculo proposicional:

$$\mathcal{R} = A \rightarrow B = \neg A \cup (A \cap B). \quad (2.27)$$

- Cálculo proposicional extendido:

$$\mathcal{R} = A \rightarrow B = (\neg A \cap \neg B) \cup B. \quad (2.28)$$

- Generalización del *modus ponens*:

$$\mu_{\mathcal{R}}(x, y) = \sup\{c \mid \mu_A(x) \tilde{*} c \leq \mu_B(y) \wedge 0 \leq c \leq 1\}, \quad (2.29)$$

donde $\mathcal{R} = A \rightarrow B$ y $\tilde{*}$ es un operador T-norm.

Las fórmulas anteriores se reducen a la identidad $A \rightarrow B \equiv \neg A \cup B$ cuando A y B son proposiciones en el sentido de la lógica tradicional.

Combinando las posibilidades de interpretación de esta relación con varios operadores T-norm y T-conorm, pueden formularse varios métodos calificados para calcular $\mathcal{R} = A \rightarrow B$. Considerando que \mathcal{R} puede ser vista como un conjunto difuso con una MF bidimensional

$$\mu_{\mathcal{R}}(x, y) = f(\mu_A(x), \mu_B(y)) = f(a, b),$$

con $a = \mu_A(x)$, $b = \mu_B(y)$, donde la función f , llamada la función de implicación difusa, desempeña la tarea de transformar los niveles de pertenencia de x en A y y en B en aquellos de (x, y) en $A \rightarrow B$.

Al adoptarse la primera interpretación, "A asociado con B", como el significado de $A \rightarrow B$. Entonces cuatro relaciones difusas diferentes $A \rightarrow B$ resultan al emplear cuatro de los operadores T-norm más usados.

- $\mathcal{R}_m = A \times B = \int_{X \times Y} \mu_A(x) \wedge \mu_B(y) / (x, y)$, o $f_c(a, b) = a \wedge b$. Esta relación, que fue propuesta por Mamdani, resulta de usar el operador mín para la conjunción.

- $\mathcal{R}_p = A \times B = \int_{X \times Y} \mu_A(x) \mu_B(y) / (x, y)$, o $f_p(a, b) = ab$. Propuesta por Larsen [11], esta relación está basada en el uso del producto algebraico para la conjunción.
- $\mathcal{R}_{bp} = A \times B = \int_{X \times Y} \mu_A(x) \odot \mu_B(y) / (x, y) = \int_{X \times Y} 0 \vee (\mu_A(x) + \mu_B(y) - 1) / (x, y)$, o $f_{bp}(a, b) = 0 \vee (a + b - 1)$. Esta fórmula emplea el producto acotado para la conjunción.
- $\mathcal{R}_{dp} = A \times B = \int_{X \times Y} \mu_A(x) \hat{\cdot} \mu_B(y) / (x, y)$, o

$$f(a, b) = a \hat{\cdot} b = \begin{cases} a & \text{si } b = 1. \\ b & \text{si } a = 1. \\ 0 & \text{otro caso.} \end{cases}$$

Esta fórmula utiliza el operador producto drástico para la conjunción.

La primera fila de la figura 2.10 muestra dos funciones de implicación difusa bajo el esquema “A asociado con B”, con $a = \mu_A(x)$ y $b = \mu_B(y)$; la segunda fila muestra las relaciones difusas correspondientes \mathcal{R}_m y \mathcal{R}_p cuando $\mu_A(x) = bell(x; 4, 3, 10)$ y $\mu_B(y) = bell(y; 4, 3, 10)$.

Cuando adoptamos la segunda interpretación, “A implica B”, como el significado de $A \rightarrow B$, existen también varias funciones de implicación difusa que son candidatos razonables. Las siguientes cuatro han sido propuestas en la literatura:

- $\mathcal{R}_a = \neg A \cup B = \int_{X \times Y} 1 \wedge (1 - \mu_A(x) + \mu_B(y)) / (x, y)$, o $f_a(a, b) = 1 \wedge (1 - a + b)$. Ésta es la regla aritmética propuesta por Zadeh, que proviene de la ecuación 2.26 al usar el operador de suma acotada para \cup .
- $\mathcal{R}_{mn} = \neg A \cup (A \cap B) = \int_{X \times Y} (1 - \mu_A(x)) \vee (\mu_A(x) \wedge \mu_B(y)) / (x, y)$, o $f_{mn}(a, b) = (1 - a) \vee (a \wedge b)$. Ésta es la regla max-min de Zadeh, que proviene de la ecuación 2.27 al usar mín por \cap y máx por \cup .

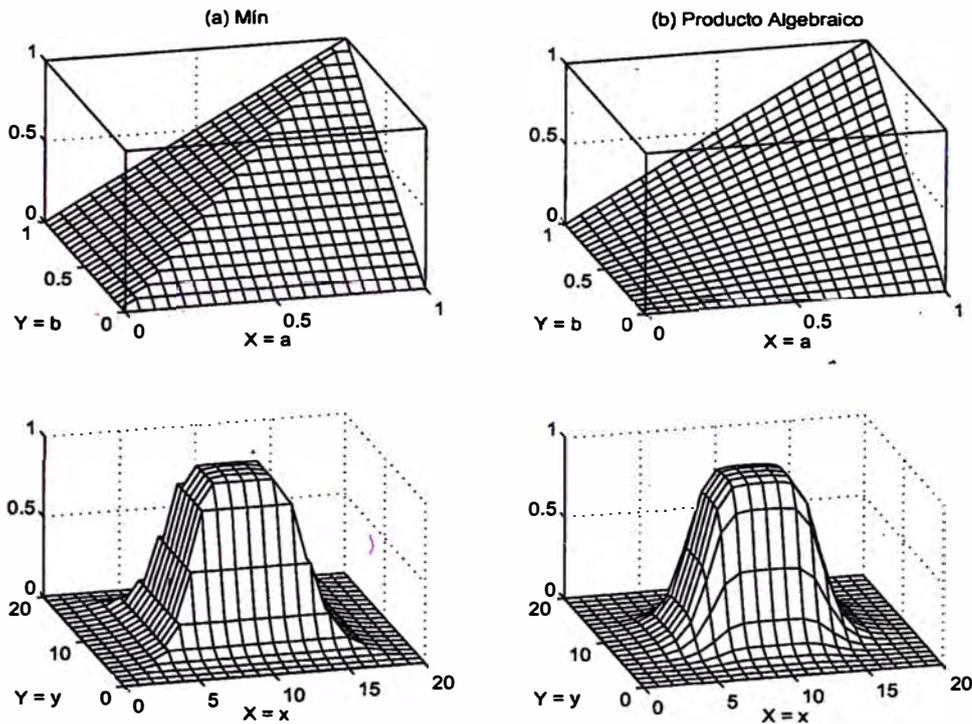


Figura 2.10: Primera fila: funciones de implicación difusa basadas en la interpretación “A asociado con B”; segunda fila: las relaciones difusas correspondientes.

- $\mathcal{R}_s = \neg A \cup B = \int_{X \times Y} (1 - \mu_A(x)) \vee \mu_B(y) / (x, y)$, o $f_s(a, b) = (1 - a) \vee b$.
Ésta es la implicación difusa booleana usando máx por \cup .
- $\mathcal{R}_\Delta = \int_{X \times Y} \mu_A(x) \tilde{<} \mu_B(y) / (x, y)$, donde

$$a \tilde{<} b = \begin{cases} 1 & \text{si } a \leq b. \\ b/a & \text{si } a > b. \end{cases}$$

Ésta es la implicación difusa de Goguen, que proviene de la ecuación 2.29 al usar el producto algebraico por el operador T-norm.

Las figuras 2.11 y 2.12 muestran las cuatro funciones de implicación difusa bajo el esquema “A implica B” mostradas anteriormente ($a = \mu_A(x)$ y $b = \mu_B(y)$); la segunda fila muestra las relaciones difusas resultantes \mathcal{R}_a , \mathcal{R}_{mn} , \mathcal{R}_s y \mathcal{R}_Δ cuando $\mu_A(x) = bell(x; 4, 3, 10)$ y $\mu_B(y) = bell(y; 4, 3, 10)$. Debe tenerse en cuenta que existen otras funciones de implicación difusa en la literatura (ver [12]).

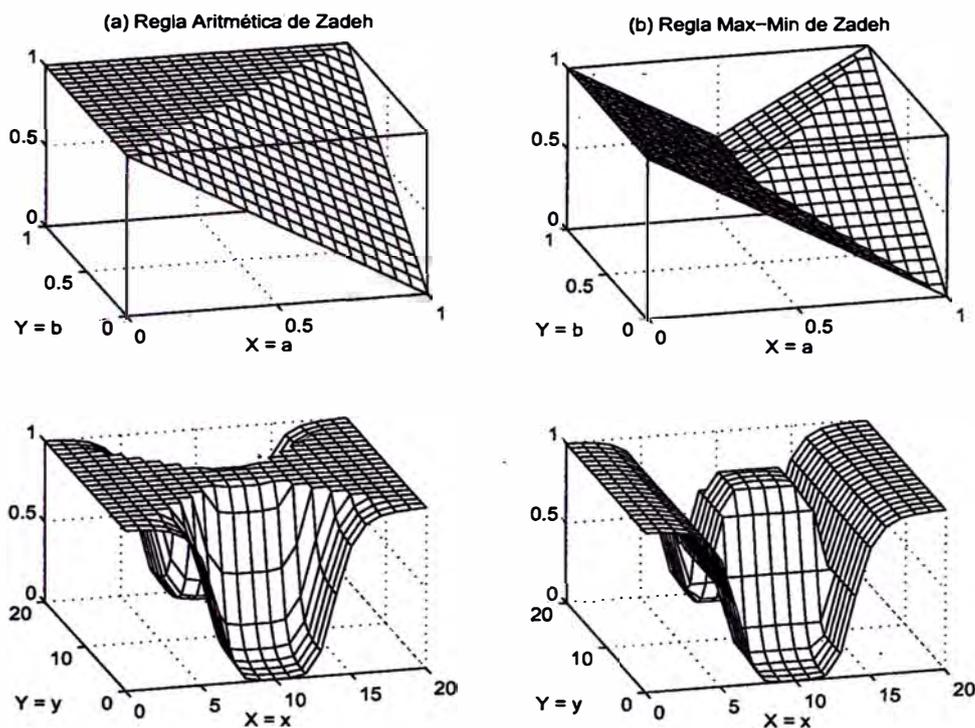


Figura 2.11: Primera fila: funciones de implicación difusa basadas en la interpretación “ A implica B ”; segunda fila: las relaciones difusas correspondientes.

Regla Composicional de Inferencia

La regla composicional de inferencia es una generalización del siguiente concepto. Supóngase que disponemos de una curva $y = f(x)$ que regula la relación entre x e y . Para un determinado $x = a$, entonces, a partir de $y = f(x)$, se puede inferir que $y = b = f(a)$; véase la figura 2.13(a). Una generalización del proceso anteriormente mencionado sería permitir que a sea un intervalo y $f(x)$ una función evaluada en intervalos, como se muestra en la figura 2.13(b). Para encontrar el intervalo resultante $y = b$ correspondiente al intervalo $x = a$, primero se construye una extensión cilíndrica de a y luego se encuentra su intersección I con la curva evaluada en intervalos. La proyección de I sobre el eje y da lugar al intervalo $y = b$.

Yendo un paso adelante en esta generalización, se asumirá que F es una relación difusa en $X \times Y$ y A es un conjunto difuso de X , como se muestra

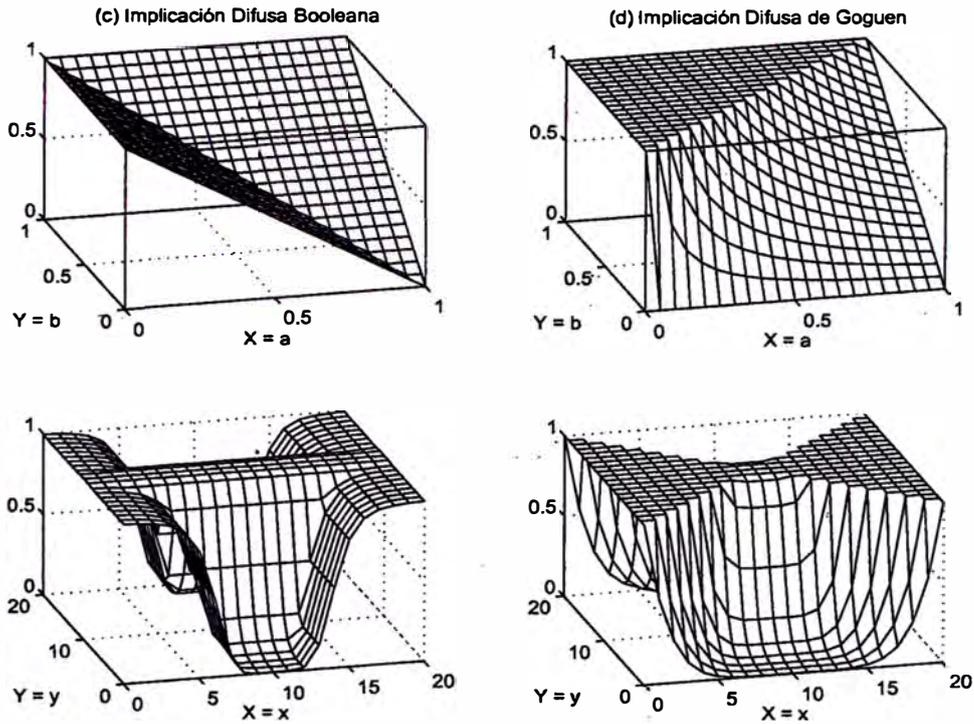


Figura 2.12: Primera fila: funciones de implicación difusa basadas en la interpretación “ A implica B ”; segunda fila: las relaciones difusas correspondientes.

en las figuras 2.14(a) y 2.14(b) . Para encontrar el conjunto difuso resultante B , nuevamente se construye una extensión cilíndrica $c(A)$ con base A . La intersección de $c(A)$ y F (ver Figura 2.14(c)), forma el análogo de la región de intersección I en la figura 2.13(b). Al proyectar $c(A) \cap F$ sobre el eje y , se infiere y como un conjunto difuso B sobre el eje y , como se muestra en la figura 2.14(d).

Específicamente, sean μ_A , $\mu_{c(A)}$, μ_B y μ_F las MFs de A , $c(A)$, B y F , respectivamente, donde $\mu_{c(A)}$ está relacionado con μ_A a través de:

$$\mu_{c(A)}(x, y) = \mu_A(x).$$

Entonces,

$$\begin{aligned} \mu_{c(A) \cap F}(x, y) &= \text{mín}[\mu_{c(A)}(x, y), \mu_F(x, y)]. \\ &= \text{mín}[\mu_A(x), \mu_F(x, y)]. \end{aligned}$$

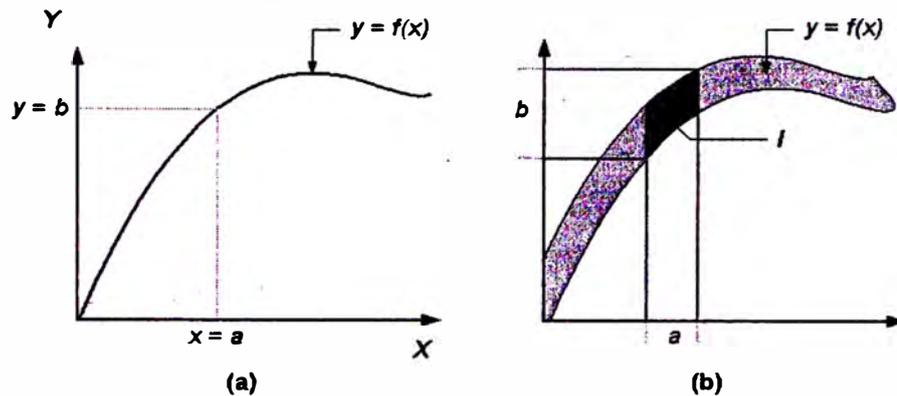


Figura 2.13: Derivación de $y = b$ a partir de $x = a$ y $y = f(x)$: (a) a y b son puntos, $y = f(x)$ es una curva; (b) a y b son intervalos, $y = f(x)$ es una función evaluada en intervalos.

Al proyectar $c(A) \cap F$ sobre el eje y , se tiene

$$\begin{aligned}\mu_B(y) &= \max_x \min[\mu_A(x), \mu_F(x, y)]. \\ &= \vee_x [\mu_A(x) \wedge \mu_F(x, y)].\end{aligned}$$

Esta fórmula se reduce a la composición max-min (ver la definición 2.19) de dos matrices de relación si ambas A (una relación difusa unitaria) y F (una relación difusa binaria) tienen universos de discurso finitos. Convencionalmente, B se representa como

$$B = A \circ F,$$

donde \circ denota al operador de composición.

Usando la regla composicional de inferencia, podemos formalizar un procedimiento de inferencia sobre un set de reglas difusas si-entonces. Este procedimiento de inferencia, es generalmente llamado razonamiento aproximado o razonamiento difuso.

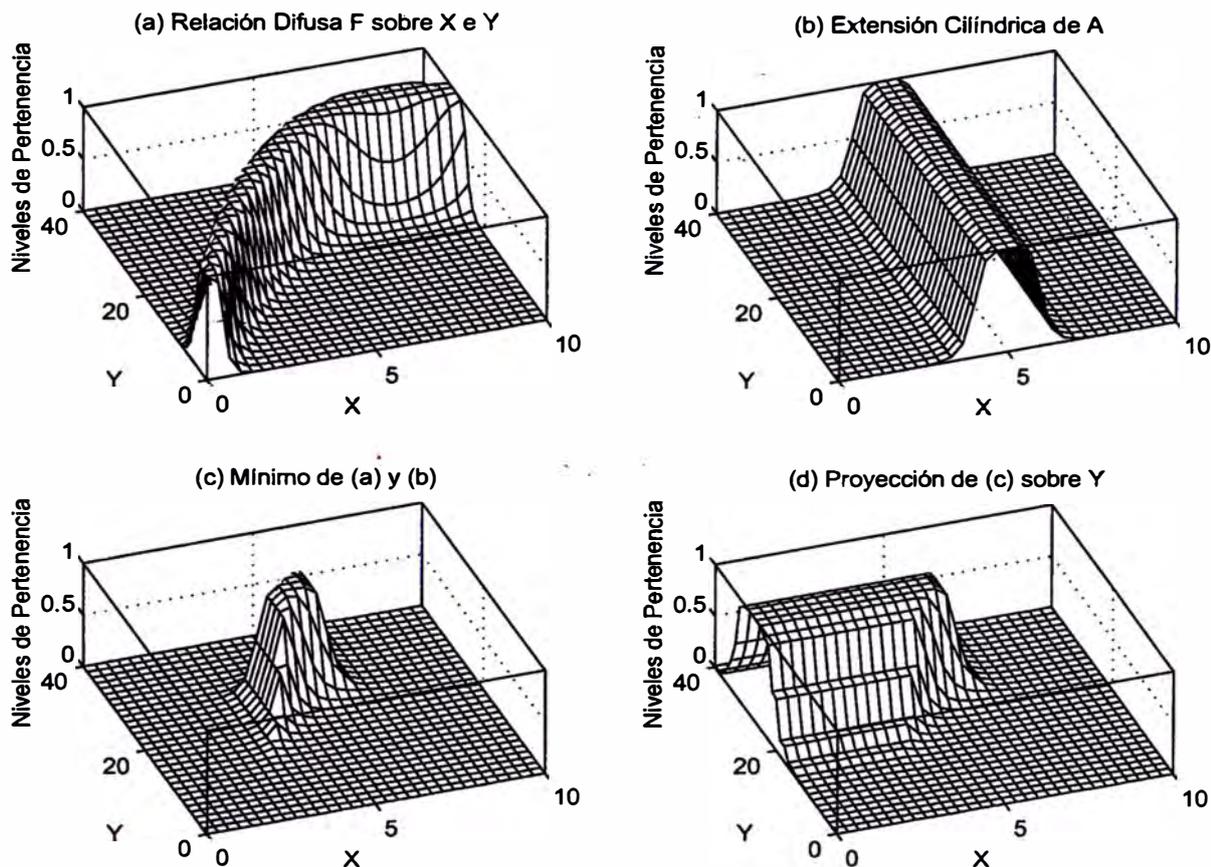


Figura 2.14: Regla Composicional de Inferencia.

Razonamiento Difuso

La regla básica de inferencia en la lógica tradicional es el *modus ponens*, de acuerdo con el cual se puede inferir la verdad de una proposición B a partir de la verdad de una proposición A y de la implicación $A \rightarrow B$. Este concepto se ilustra como sigue:

premisa 1 (hecho):	x es A ,
premisa 2 (regla):	si x es A entonces y es B ,
<hr/>	
consecuencia (conclusión):	y es B .

Sin embargo, generalmente, el razonamiento humano hace uso del *modus ponens* de manera aproximada; de modo que las proposiciones no se evalúan

como estrictamente verdaderas o falsas. Esto puede escribirse como:

$$\begin{array}{ll} \text{premisa 1 (hecho):} & x \text{ es } A', \\ \text{premisa 2 (regla):} & \text{si } x \text{ es } A \text{ entonces } y \text{ es } B, \\ \hline \text{consecuencia (conclusión):} & y \text{ es } B', \end{array}$$

donde A' es cercano a A y B' es cercano a B . Cuando A , B , A' y B' son conjuntos difusos de universos apropiados, el procedimiento de inferencia correspondiente es llamado **razonamiento aproximado** o **razonamiento difuso**; llamado también **modus ponens generalizado (GMP)**², ya que tiene al *modus ponens* como un caso especial.

Usando la regla composicional de inferencia se puede formular el procedimiento de inferencia del razonamiento difuso bajo la siguiente definición.

Definición 2.23 *Razonamiento aproximado (razonamiento difuso)*

Sean A , A' y B conjuntos difusos de X , X , y Y , respectivamente. Se asume que la implicación difusa $A \rightarrow B$ está expresada como una relación difusa \mathcal{R} en $X \times Y$. Entonces el conjunto difuso B inducido por "x es A' " y la regla difusa "si x es A entonces y es B " queda definido por

$$\begin{aligned} \mu_{B'}(y) &= \max_x \min[\mu_{A'}(x), \mu_{\mathcal{R}}(x, y)] \\ &= \bigvee_x [\mu_{A'}(x) \wedge \mu_{\mathcal{R}}(x, y)], \end{aligned} \tag{2.30}$$

o, equivalentemente,

$$B' = A' \circ \mathcal{R} = A' \circ (A \rightarrow B). \tag{2.31}$$

A partir de la definición anterior, puede usarse el procedimiento de inferencia del razonamiento aproximado para derivar conclusiones, siempre que la implicación difusa $A \rightarrow B$ sea definida como una relación difusa binaria apropiada.

²Por sus siglas en inglés

En lo que sigue, se discutirán los aspectos computacionales del razonamiento difuso introducido en la definición precedente, y luego se extenderá la discusión a las situaciones en las que múltiples reglas difusas con múltiples antecedentes se involucran en la descripción del comportamiento de un sistema. Con el fin de facilitar la expresión gráfica del tema, se emplearán las funciones de implicación difusa de Mamdani; aunque en el controlador se empleará la implicación difusa de Takagi-Sugeno³.

Una Regla con Un Antecedente

Éste es el caso más simple, y la fórmula proviene de la ecuación 2.30. Una siguiente simplificación da lugar a:

$$\begin{aligned}\mu_{B'}(y) &= [\forall_x(\mu_{A'}(x) \wedge \mu_A(x))] \wedge \mu_B(y) \\ &= w \wedge \mu_B(y).\end{aligned}\tag{2.32}$$

En otras palabras, primero se encuentra el nivel de semejanza w como el máximo de $\mu_{A'}(x) \wedge \mu_A(x)$ (el área sombreada en la parte del antecedente de la figura 2.15); luego la MF del resultante B' es igual a la MF de B limitado por w (véase el área sombreada en la parte del consecuente de la figura 2.15). Intuitivamente, w representa una medida del nivel de certeza para el antecedente de la regla; esta medida se propaga a través de las reglas de implicación, de modo que el nivel de certeza o MF para el consecuente (B' en la figura 2.15) debería ser igual o menor que w .

³Como se verá más adelante, estos procedimientos de implicación se diferencian principalmente por el uso de la composición *max-min* en lugar de la composición *max-prod*.

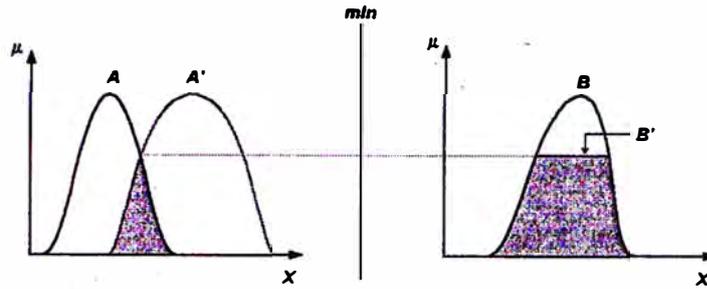


Figura 2.15: Interpretación gráfica del GMP usando la implicación difusa de Mamdani y la composición max-min.

Una Regla con Múltiples Antecedentes

Una regla de implicación difusa con dos antecedentes se escribe usualmente como “Si x es A y y es B , entonces z es C .” El problema correspondiente para el GMP se expresa como:

premisa 1 (hecho):	x es A' y y es B' ,
premisa 2 (regla):	si x es A y y es B , entonces z es C ,
consecuencia (conclusión):	z es C' ,

La regla difusa en la premisa 2 puede ser puesta en una forma más simple “ $A \times B \rightarrow C$ ”. Intuitivamente, esta regla difusa puede ser transformada en una relación difusa ternaria R_m basada en la función de implicación difusa de Mamdani, como sigue:

$$R_m(A, B, C) = (A \times B) \times C = \int_{X \times Y \times Z} \mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z) / (x, y, z).$$

El conjunto difuso resultante C' es expresado como

$$C' = (A' \times B') \circ (A \times B \rightarrow C).$$

Por lo tanto

$$\begin{aligned}
\mu_{C'}(z) &= \bigvee_{x,y} [\mu_{A'}(x) \wedge \mu_{B'}(y)] \wedge [\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)] \\
&= \bigvee_{x,y} \{[\mu_{A'}(x) \wedge \mu_{B'}(y) \wedge \mu_A(x) \wedge \mu_B(y)]\} \wedge \mu_C(z) \\
&= \underbrace{\{\bigvee_x [\mu_{A'}(x) \wedge \mu_A(x)]\}}_{w_1} \wedge \underbrace{\{\bigvee_y [\mu_{B'}(y) \wedge \mu_B(y)]\}}_{w_2} \wedge \mu_C(z) \\
&= (w_1 \wedge w_2) \wedge \mu_C(z),
\end{aligned} \tag{2.33}$$

donde w_1 y w_2 son los máximos de las MFs de $A \cap A'$ y $B \cap B'$, respectivamente. En general, w_1 denota el grado de compatibilidad entre A y A' , similarmente para w_2 . Desde que el antecedente de la regla difusa es construido por el conectivo “y”, $w_1 \wedge w_2$ es llamado **intensidad de disparo** de la regla difusa, que representa cuan cercano cercano a la verdad se encuentra el antecedente de la regla. Una representación gráfica se muestra en la figura 2.16, donde la MF del resultante C' es igual a la MF de C limitada por la intensidad de disparo w , $w = w_1 \wedge w_2$. La generalización a más de dos antecedentes es directa.

Una forma alternativa de calcular C' es explicada por el siguiente teorema.

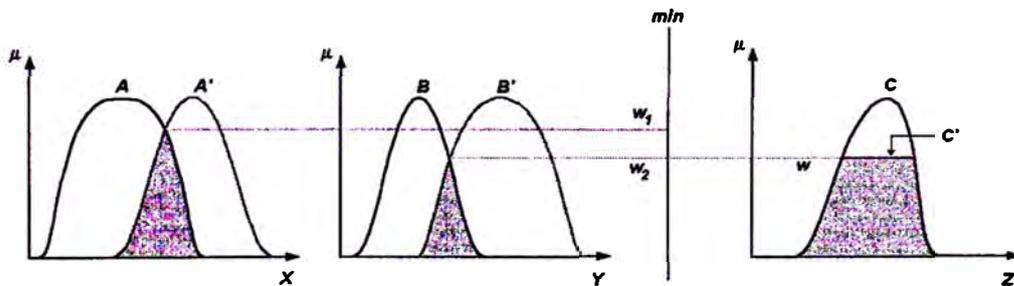


Figura 2.16: Razonamiento aproximado para múltiples antecedentes.

Teorema 2.1

$$\begin{aligned}
C' &= (A' \times B') \circ (A \times B \rightarrow C) \\
&= [A' \circ (A \rightarrow C)] \cap [B' \circ (B \rightarrow C)]
\end{aligned} \tag{2.34}$$

Prueba:

$$\begin{aligned}
 \mu_{C'}(z) &= \forall_{x,y} [\mu_{A'}(x) \wedge \mu_{B'}(y)] \wedge [\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)] \\
 &= \forall_x \{ \mu_{A'}(x) \wedge \mu_B(y) \wedge \mu_C(z) \wedge \forall_y [\mu_{B'}(y) \wedge \mu_B(y) \wedge \mu_C(z)] \} \\
 &= \mu_{A'}(x) \wedge \mu_B(y) \wedge \mu_C(z) \wedge \mu_{B' \circ (B \rightarrow C)}(z) \\
 &= \mu_{A' \circ (A \rightarrow C)}(z) \wedge \mu_{B' \circ (B \rightarrow C)}(z).
 \end{aligned} \tag{2.35}$$

El teorema precedente establece que el consecuente C' puede ser expresado como la intersección de $C'_1 = A' \circ (A \rightarrow C)$ y $C'_2 = B' \circ (B \rightarrow C)$, cada uno de los cuales corresponde al conjunto difuso inferido de un problema GMP para una regla difusa simple con un antecedente simple.

Múltiples Reglas con Múltiples Antecedentes

La interpretación de múltiples reglas es usualmente tomada como la unión de las relaciones difusas correspondientes a las reglas difusas. Por lo tanto, para un problema GMP escrito como

premisa 1 (hecho):	x es A' y y es B' ,
premisa 2 (regla 1):	si x es A_1 y y es B_1 , entonces z es C_1 ,
premisa 3 (regla 2):	si x es A_2 y y es B_2 , entonces z es C_2 ,
consecuencia (conclusión):	z es C' ,

se puede emplear el razonamiento difuso mostrado en la figura 2.17 como un procedimiento de inferencia para obtener el conjunto difuso de salida resultante C' .

Para verificar este procedimiento de inferencia, sean $\mathcal{R}_1 = A_1 \times B_1 \rightarrow C_1$ y $\mathcal{R}_2 = A_2 \times B_2 \rightarrow C_2$. Considerando que el operador \circ de composición max-min es distributivo sobre el operador \cup , se tiene que

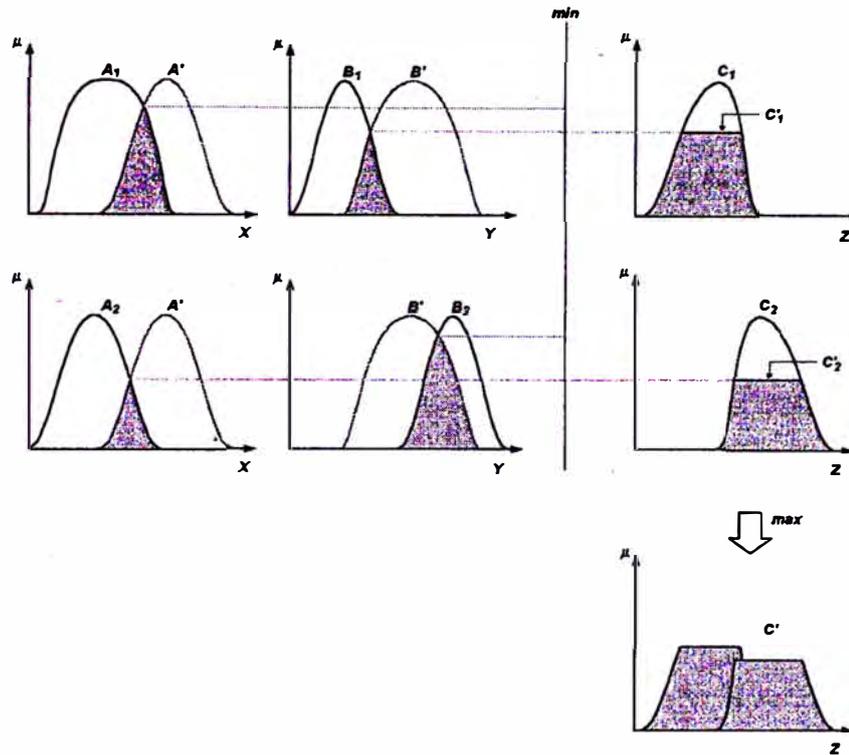


Figura 2.17: Razonamiento aproximado para múltiples reglas con múltiples antecedentes.

$$\begin{aligned}
 C' &= (A' \times B') \circ (\mathcal{R}_1 \cup \mathcal{R}_2) \\
 &= [(A' \times B') \circ \mathcal{R}_1] \cup [(A' \times B') \circ \mathcal{R}_2] \\
 &= C'_1 \cup C'_2,
 \end{aligned} \tag{2.36}$$

donde C'_1 y C'_2 son conjuntos difusos inferidos por las reglas 1 y 2, respectivamente. La figura 2.17 muestra gráficamente la operación de razonamiento difuso para múltiples reglas con múltiples antecedentes.

Cuando una regla difusa dada asume la forma “si x es A o y es B , entonces z es C ,” la intensidad de disparo es considerada como el máximo de los niveles de concordancia de los antecedentes para una condición dada. Esta regla difusa es equivalente a la unión de dos reglas difusas “si x es A , entonces z es C ” y “si y es B , entonces z es C .”

En resumen, el proceso del razonamiento difuso o razonamiento aproximado puede ser dividido en cuatro pasos:

1. Nivel de compatibilidad: Se comparan los hechos conocidos con los antecedentes de las reglas difusas para encontrar el grado de compatibilidad respecto de cada MF antecedente.
2. Intensidad de disparo: Se combina los niveles de compatibilidad respecto de las MFs antecedentes en una regla usando los operadores difusos AND u OR para definir una intensidad de disparo que indica que tan cercano a la verdad, o qué tanto se satisface la condición del antecedente de la regla.
3. Inducción de la MF del consecuente: Se aplica la intensidad de disparo a la MF del consecuente de una regla para generar la MF inducida, que representa como la intensidad de disparo es propagada y usada en una declaración de implicación difusa.
4. MF de la salida global: Se agregan todos las MF inducidas para obtener una MF de la salida global.

Estos cuatro pasos se emplean en los sistemas de inferencia difusa que se describen a continuación.

2.1.3. Sistemas de Inferencia Difusa: Modelo de Takagi-Sugeno

La estructura básica de un sistema de inferencia difusa (FIS)⁴ consiste de tres componentes conceptuales: una base de reglas, que contiene una selección de reglas difusas; una base de datos (o diccionario) que define las funciones de membresía usadas en las reglas difusas; y un mecanismo de razonamiento, que desempeña el procedimiento de inferencia (usualmente el razonamiento difuso

⁴*FIS: Fuzzy Inference System*

introducido en la subsección anterior) sobre las reglas definidas para derivar una conclusión.

Un FIS puede tomar tanto entradas difusas como entradas *crisp* (que son vistas como *singletons* difusos), pero las salidas que genera pertenecen al espacio de los conjuntos difusos. Para un controlador es necesario transformar la salida difusa en *crisp*; por lo tanto, se necesita un método de defuzzificación para extraer un valor *crisp* que represente al conjunto difuso de salida.

Con entradas y salidas *crisp*, un FIS implementa un mapeo no lineal desde su espacio de entrada a su espacio de salida. Este mapeo es llevado a cabo por una variedad de reglas difusas, cada una de las cuales describe el comportamiento local del mapeo. En particular, el antecedente de una regla define una región difusa en el espacio de entrada, mientras que el consecuente especifica la salida en la región difusa.

El modelo difuso de Takagi-Sugeno (conocido también como el modelo difuso TSK) fue propuesto por Takagi, Sugeno y Kang [13, 14] en un esfuerzo por desarrollar una aproximación sistemática para la generación de reglas a partir de un conjunto de datos de entrada y salida predefinidos. La forma típica de las reglas difusas bajo este modelo es como sigue:

$$\text{Si } x \text{ es } A \text{ e } y \text{ es } B \text{ entonces } z = f(x, y),$$

donde A y B son conjuntos difusos en el antecedente, mientras que $z = f(x, y)$ es un polinomio en las variables de entrada x e y , pero puede ser cualquier función que describa apropiadamente la salida del modelo en la región difusa definida por el antecedente de la regla. Cuando $f(x, y)$ es un polinomio de primer orden, se dice que se trata de un modelo difuso de Takagi-Sugeno de primer orden, [13, 14]. Cuando $f(x, y)$ es una constante se tiene un modelo difuso Takagi-Sugeno de orden cero, cuya salida es una función suave de sus variables de entrada

siempre que las MFs vecinas en el antecedente tengan suficiente traslape. Este modelo puede ser visto como un caso especial del modelo difuso de Mamdani, [15], donde el consecuente de cada regla quedaría definido por un *singleton*.

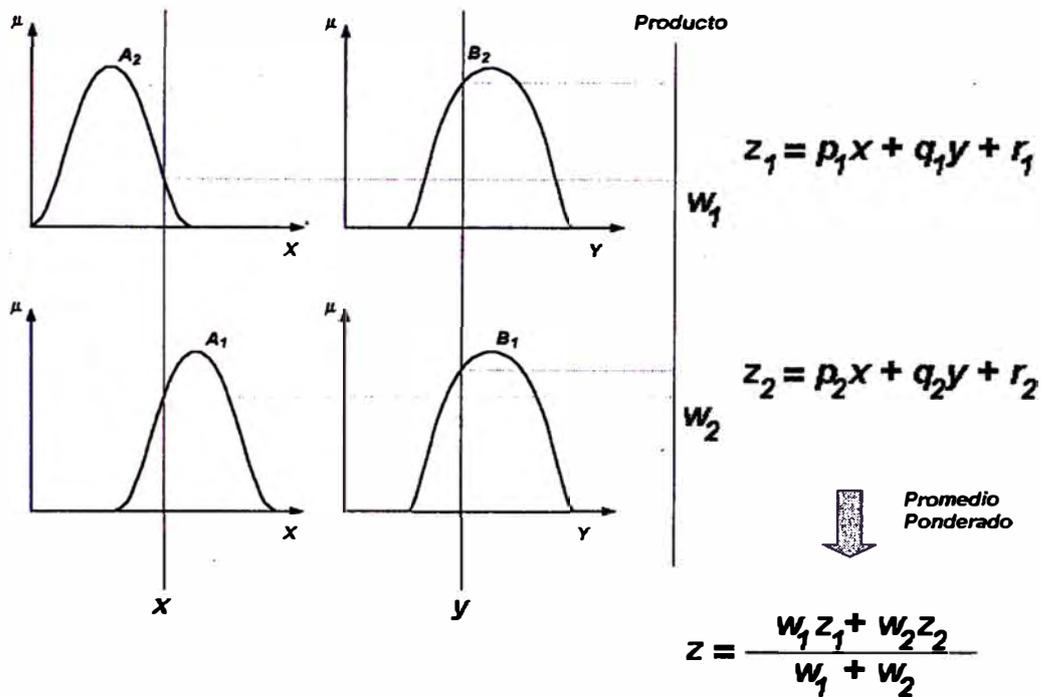


Figura 2.18: El modelo difuso de Takagi-Sugeno.

La figura 2.18 muestra el procedimiento de razonamiento para un modelo de primer orden. Desde que cada regla tiene una salida *crisp*, la salida global es obtenida mediante el promedio ponderado, que es un proceso de defuzzificación más rápidamente computable que el de centro de gravedad, requerido usualmente por el modelo de Mamdani.

CAPÍTULO 3

MODELO MATEMÁTICO DEL SISTEMA GRÚA-PUENTE

En este capítulo se da una descripción completa del modelo de la grúa-puente, una derivación de las ecuaciones de movimiento y el modelo en el espacio de estados correspondiente. Para obtener un conjunto de ecuaciones de movimiento que modelen la dinámica del sistema, se usará el enfoque de las ecuaciones de Euler-Lagrange; que son apropiadas para analizar el comportamiento de los sistemas dinámicos.

3.1. Descripción del Sistema

Una grúa-puente está compuesta por un carro que se mueve en línea recta sobre un puente o pescante horizontal. Un torno de izado en el carro da lugar al movimiento vertical de la carga, mediante unos cables de amarre que se enrollan y desenrollan en este torno. La carga colgante del cable es tratada como un péndulo simple que oscila con un grado de libertad.

Mediante un sistema de transmisión accionado por un motor eléctrico, el carro se desliza sobre los rieles que descansan en el puente. A su vez, el puente puede deslizarse sobre rieles instalados en el suelo. En la figura 3.1 se muestra el esquema de una grúa-puente de contenedores usada comúnmente en un puerto comercial. En esta investigación se analizará solamente el movimiento de la carga

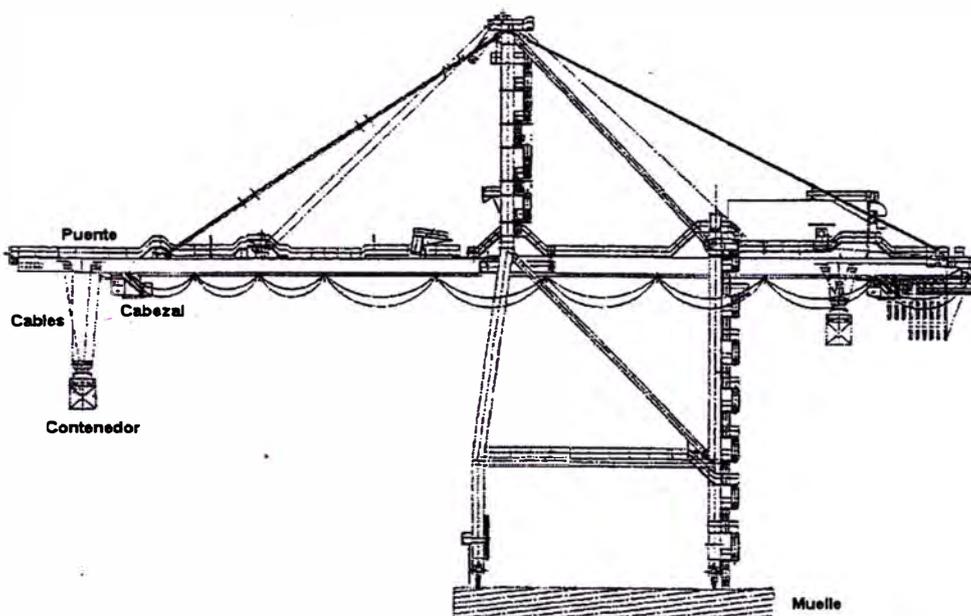


Figura 3.1: Vista esquemática de una grúa-puente.

en el plano formado por la recta de traslación del carro y la vertical. En el caso de una grúa puente con dos movimientos horizontales perpendiculares, el análisis es muy similar, debido a que los movimientos bidireccionales pueden ser divididos en dos movimientos unidireccionales desacoplados.

El prototipo que se usará en los experimentos difiere de una grúa-puente industrial en el uso de una varilla que reemplaza a los cables. Fuera de esta diferencia, el prototipo puede considerarse un modelo a escala reducida.

Como se muestra en la figura 3.2, el sistema general del prototipo de grúa-puente puede considerarse conformado por los siguientes elementos:

- (a) Dos columnas que sostienen la viga de la grúa.
- (b) Una base pesada para prevenir oscilaciones. (Las grúas-puente reales se mueven sobre rieles.)
- (c) Una viga que se apoya en las columnas.
- (d) Un carro que se desliza sobre los rieles de la grúa.

- (e) Una varilla que une la carga con el carro. En el caso real se usan cables de acero enrollados en uno o varios cabrestantes.
- (f) Propiamente, la carga no forma parte de la grúa, pero en el caso del prototipo se tiene una pesa unida a la varilla.

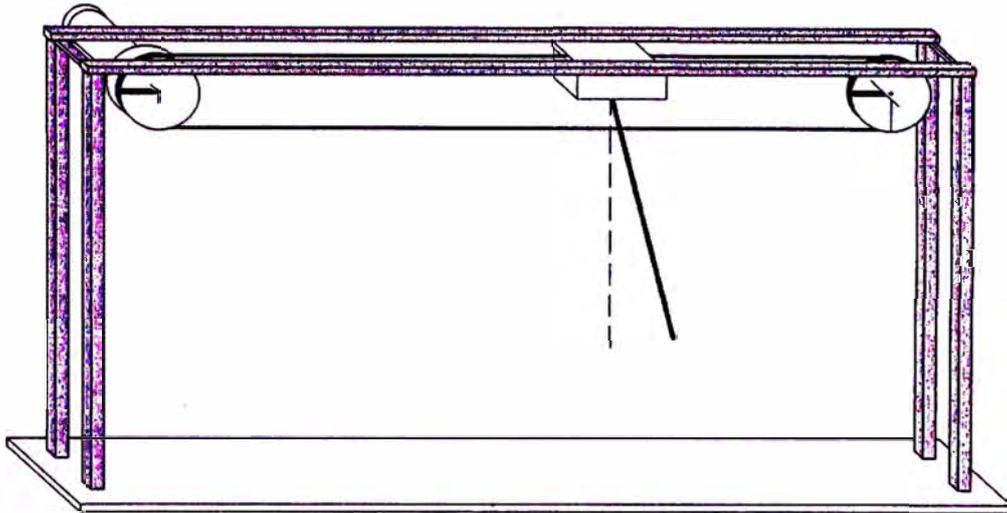


Figura 3.2: Sistema general del prototipo de grúa-puente.

El sistema Grúa-Puente es un sistema no lineal multivariable y subactuado. Para el análisis de este sistema se considerarán dos variables de salida y una variable de entrada, ver tabla 3.1. Este es un sistema subactuado desde que el número de grados de libertad es mayor que el de las señales de control, [16], [17]. Se considerará también que todo movimiento del cabezal y de la carga se encuentra en un mismo plano invariable en el tiempo.

Variable	Descripción	Unidad
x	Posición del Cabezal	m
θ	Posición Angular del Péndulo	rad
u	Señal de Control	V

Cuadro 3.1: Variables del Sistema Grúa-Puente.

3.2. Parámetros del Modelo

El sistema Grúa-Puente está compuesto por dos subsistemas: el subsistema cabezal-péndulo y el subsistema motor-polea. Los parámetros involucrados en el modelamiento de ambos subsistemas se encuentran detallados en las tablas 3.2 y 3.3. En la tabla 3.4 se muestran otras variables consideradas en el desarrollo de las ecuaciones dinámicas.

Variable	Descripción	Unidad
m_c	Masa del Cabezal	Kg
m_e	Masa de la Carga	Kg
m_v	Masa del péndulo	Kg
l_e	Distancia del punto P al C.G de la Carga	m
l_v	Distancia del punto P al C.G del Péndulo	m
μ_k	Coefficiente de Fricción Cinético del Cabezal	--
g	Gravedad	m/s^2

Cuadro 3.2: Parámetros del Subsistema Cabezal-Péndulo.

Variable	Descripción	Unidad
K_m	Constante de Torque del Motor	$N.m/A$
K_b	Constante de Fuerza Contraelectromotriz	$V/rad/s$
K_a	Constante de Ganancia del Amplificador	--
R_a	Resistencia de Armadura	Ω
J_m	Momento de Inercia del Motor	$Kg.m^2$
B_m	Constante de Fricción Viscosa del Motor	$N.m/rad/s$
J_n	Momento de Inercia de la Reducción	$Kg.m^2$
B_n	Constante de Fricción Viscosa de la Reducción	$N.m/rad/s$
r_p	Radio de la Polea	m
J_p	Momento de Inercia de la Polea	$Kg.m^2$
B_p	Constante de Fricción Viscosa de la Polea	$N.m/rad/s$
n	Relación de Reducción	--

Cuadro 3.3: Parámetros del Subsistema Motor-Polea.

En las siguientes secciones se obtendrán las ecuaciones de ambos subsistemas para formar las ecuaciones dinámicas del sistema Grúa-Puente a partir del desarrollo de las *Ecuaciones de Lagrange*.

Símbolo	Descripción	Unidad
F	Fuerza aplicada al carro	N
ϕ	Ángulo de giro del eje del motor-reductor	rad
x_L	Desplazamiento horizontal de la carga	m
\vec{r}_c	Vector de posición del carro (eje del péndulo)	m
\vec{r}_L	Vector de posición del centro de gravedad de la carga	m
i_a	Corriente de armadura	A

Cuadro 3.4: Otras variables del sistema.

3.3. Ecuaciones de Movimiento

A continuación se desarrollarán las ecuaciones diferenciales que describen la dinámica involucrada en el funcionamiento de un sistema grúa-puente. Se subdivide el análisis en dos partes: un subsistema carro-péndulo y un subsistema motor-polea. El primero tiene como entrada la tensión en el cable generada por el segundo, siendo ambos interdependientes; ya que la carga en el eje del motor depende de los estados del subsistema carro-péndulo.

Se emplearán las ecuaciones de Lagrange para el primer subsistema y un modelo convencional para describir la dinámica del motor.

3.3.1. Subsistema Carro - Péndulo

Las ecuaciones de Lagrange pueden usarse convenientemente para el análisis dinámico de los sistemas mecánicos, debido a que se basan en la relación que existe entre las energías potencial, cinética y disipada; las cuales son magnitudes que pueden expresarse en términos algebraicos de parámetros físicos escalares. Al prescindir de la mecánica vectorial se disminuye la complejidad del análisis, que de otro modo se presentaría con el uso de diagramas vectoriales.

Para comenzar el análisis mecánico por el método de Lagrange se debe definir un conjunto de coordenadas generales $q_i (i = 1, 2, \dots, r)$ para cada uno de los grados de libertad independientes del proceso. Luego, se deben formular las ecuaciones de energía cinética V , potencial U y disipada D , en términos de estas

coordenadas generales y sus derivadas. Después, a partir de estas ecuaciones, se define el lagrangiano \mathcal{L} del proceso:

$$\mathcal{L} = V(q_1, \dots, q_r, \dot{q}_1, \dots, \dot{q}_r) - U(q_1, \dots, q_r, \dot{q}_1, \dots, \dot{q}_r) \quad (3.1)$$

Una vez obtenido el lagrangiano, éste se integra en la trayectoria comprendida entre los instantes t_1 y t_2 , dando lugar a I .

$$I = \int_{t_1}^{t_2} \mathcal{L}(q_1, \dots, q_r, \dot{q}_1, \dots, \dot{q}_r) \quad (3.2)$$

La variación de I , de acuerdo con el Principio de Hamilton, es igual al trabajo realizado por todas las fuerzas externas en un sistema conservativo. De este modo se relacionan las fuerzas y torques que actúan en el proceso con la energía del sistema en sus tres formas. La ecuación 3.3 describe esta relación.

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} + \frac{\partial D}{\partial \dot{q}_i} = Q_i \quad (3.3)$$

Q_i representa la i -ésima fuerza o torque generalizado que es externo al proceso o que no es obtenible a partir de una función potencial escalar.

Con la ecuación anterior se puede describir matemáticamente el comportamiento del sistema, queda entonces definir las energías potencial, cinética y disipativa en función de los parámetros declarados en la sección 2.2.

El subsistema Cabezal-Péndulo está representado en la figura 3.3, el carro tiene un movimiento horizontal y el brazo (péndulo) gira libremente alrededor del C.G. del carro.

La ecuación de energía cinética del subsistema está dada por:

$$T = \frac{1}{2} m_c \bar{r}_c^2 + \frac{1}{2} m_L \bar{r}_L^2 \quad (3.4)$$

El desarrollo de la ecuación 3.4 se explica con la ayuda de la figura 3.3 donde

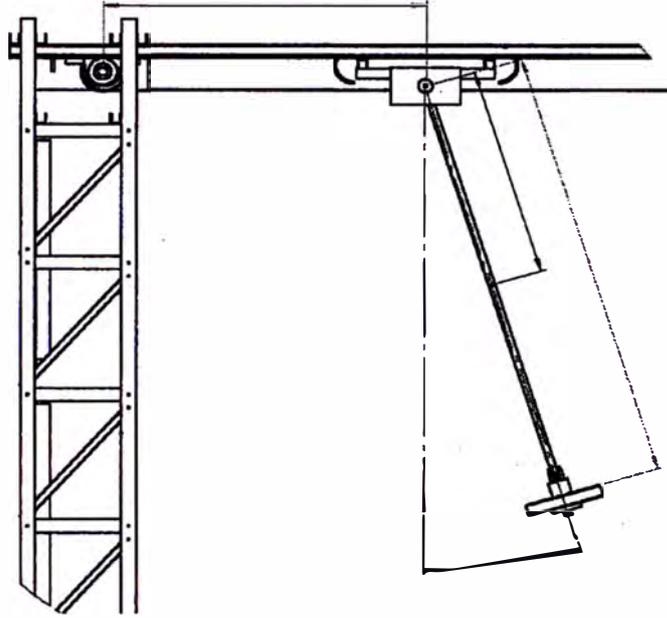


Figura 3.3: Esquema del subsistema carro-péndulo.

se obtienen las coordenadas (x, y) de los elementos del subsistema.

$$\bar{r}_c = [x, 0]^T \quad (3.5)$$

$$\bar{r}_L = [x + l_v \sin \theta, -l_v \cos \theta]^T \quad (3.6)$$

derivando las ecuaciones 3.5 y 3.6,

$$\dot{\bar{r}}_c = [\dot{x}, 0]^T \quad (3.7)$$

$$\dot{\bar{r}}_L = [\dot{x} + l_v \dot{\theta} \cos \theta, l_v \dot{\theta} \sin \theta]^T \quad (3.8)$$

elevando al cuadrado, obtenemos

$$\dot{\bar{r}}_c^2 = \dot{x}^2 \quad (3.9)$$

$$\bar{r}_L^2 = \dot{x}^2 + 2l_v \dot{x} \dot{\theta} \cos \theta + l_v^2 \dot{\theta}^2 \quad (3.10)$$

reemplazando las ecuaciones 3.9 y 3.10 en 3.4, tenemos la energía cinética del subsistema.

$$T = \frac{1}{2}(m_c + m_L)\dot{x}^2 + \cos \theta (m_L l_v) \dot{x} \dot{\theta} + \frac{1}{2}(m_L l_v^2) \dot{\theta}^2 \quad (3.11)$$

La energía potencial del subsistema está dada por,

$$U = -m_L l_v g \cos \theta \quad (3.12)$$

$$D = \frac{1}{2} F_f \dot{x}^2 + \frac{1}{2} C \dot{\alpha}^2 \quad (3.13)$$

Para obtener las ecuaciones que representan la dinámica del subsistema carro-péndulo, primero debemos obtener en términos de las coordenadas generales (x, θ) las Ecuaciones de Lagrange que pueden ser formuladas de la siguiente manera

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{x}} \right) - \frac{\partial T}{\partial x} + \frac{\partial U}{\partial x} + \frac{\partial D}{\partial \dot{x}} = F \quad (3.14)$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\theta}} \right) - \frac{\partial T}{\partial \theta} + \frac{\partial U}{\partial \alpha} + \frac{\partial D}{\partial \dot{\theta}} = 0 \quad (3.15)$$

desarrollando las derivadas parciales de la ecuación 3.14

$$\frac{\partial T}{\partial \dot{x}} = (m_c + m_L)\dot{x} + (m_L l_v) \cos(\theta) \dot{\theta}$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{x}} \right) = (m_c + m_L)\ddot{x} + (m_L l_v)(\cos(\theta)\ddot{\theta} - \sin(\theta)\dot{\theta}^2) \quad (3.16)$$

$$\frac{\partial T}{\partial x} = 0 \quad \frac{\partial U}{\partial x} = 0 \quad (3.17)$$

$$\frac{\partial D}{\partial \dot{x}} = F_f \dot{x} \quad (3.18)$$

desarrollando las derivadas parciales de la ecuación 3.15

$$\frac{\partial T}{\partial \dot{\alpha}} = (m_L l_v) \cos(\theta) \dot{x} + (m_L l_v^2) \dot{\theta}$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\theta}} \right) = (m_L l_v) (\ddot{x} \cos(\theta) - \dot{x} \sin(\theta) \dot{\theta}) + (m_L l_v^2) \ddot{\theta} \quad (3.19)$$

$$\frac{\partial T}{\partial \theta} = -(m_L l_v) \dot{x} \sin(\theta) \dot{\theta} \quad \frac{\partial U}{\partial \theta} = (m_L l_v) g \sin(\theta) \quad (3.20)$$

$$\frac{\partial D}{\partial \dot{\theta}} = C \dot{\theta} \quad (3.21)$$

de las ecuaciones 3.16 a 3.21 obtenemos las ecuaciones que describen la dinámica del subsistema Cabezal-Péndulo.

$$(m_c + m_v + m_e) \ddot{x} + (m_L l_v) (\cos(\theta) \ddot{\theta} - \sin(\theta) \dot{\theta}^2) + F_f \dot{x} = F \quad (3.22)$$

$$(m_L l_v^2) \ddot{\theta} + (m_L l_v + m_v l_v) (\ddot{x} \cos(\theta) + g \sin(\theta)) + C \dot{\theta} = 0 \quad (3.23)$$

3.3.2. Subsistema Motor-Polea

El subsistema Motor-Polea esta representado en la figura 3.4, consta de un motor-reductor DC acoplado a una polea. Utilizaremos el modelo de motor controlado por corriente de armadura (i_a).

Despreciando la inductancia L_a , el voltaje de armadura $K_a u$ aplicado a la armadura esta dado por

$$R_a i_a + V_b = K_a u \quad (3.24)$$

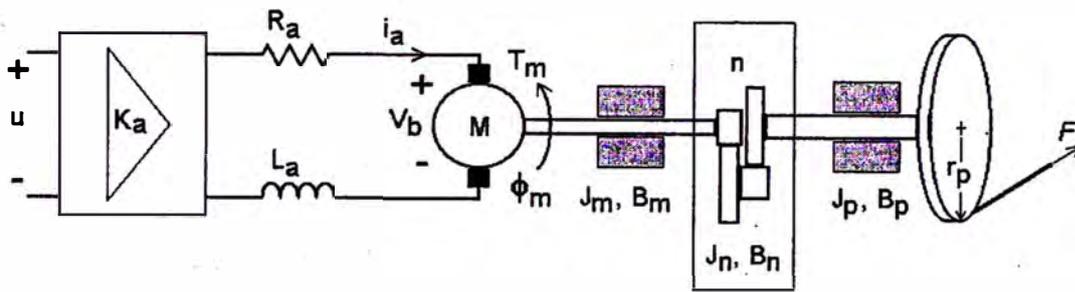


Figura 3.4: Representación del subsistema motor-polea.

donde voltaje de fuerza contra-electromotriz V_b es proporcional a la velocidad del motor, es decir

$$V_b = K_b \dot{\phi}_m \quad (3.25)$$

el torque del motor-reductor DC esta dado por

$$T_m = J_{eq} \ddot{\phi}_m + B_{eq} \dot{\phi}_m + nr_p F \quad (3.26)$$

donde J_{eq} es el momento de inercia equivalente y B_{eq} es la constante de fricción equivalente, dados por

$$J_{eq} = J_m + n^2(J_n + J_p) \quad (3.27)$$

$$B_{eq} = B_m + n^2(B_n + B_p) \quad (3.28)$$

donde η es la relación de reducción, además, J_m, J_n, J_p y B_m, B_n, B_p son los momentos de inercia y constantes de fricción del motor, reducción y de la polea, respectivamente.

También, el torque del motor-reductor DC es proporcional a la corriente de armadura i_a , es decir

$$T_m = K_m i_a \quad (3.29)$$

reemplazando 3.26 en 3.29, obtenemos

$$i_a = \frac{J_{eq}}{K_m} \ddot{\phi}_m + \frac{B_{eq}}{K_m} \dot{\phi}_m + \frac{nr_p}{K_m} F \quad (3.30)$$

reemplazando 3.30 en 3.24,

$$K_b \dot{\phi}_m + \frac{J_{eq} R_a}{K_m} \ddot{\phi}_m + \frac{B_{eq} R_a}{K_m} \dot{\phi}_m + \frac{nr_p R_a}{K_m} F = K_a u \quad (3.31)$$

además, el movimiento rotacional de la motor puede se transformado como el movimiento traslacional del carro de la siguiente manera

$$\phi_m = \frac{x}{nr_p} \quad (3.32)$$

luego, utilizando la ecuación 3.32, despejamos F

$$F = \left(\frac{K_m K_a}{nr_p R_a} \right) u - \left(\frac{J_{eq}}{n^2 r_p^2} \right) \ddot{x} - \left(\frac{B_{eq}}{n^2 r_p^2} + \frac{K_m K_b}{n^2 r_p^2 R_a} \right) \dot{x} \quad (3.33)$$

la ecuación 3.33 describe la dinámica del subsistema motor-polea.

3.4. Representación en el Espacio de Estados del Sistema Grúa-Puente

Las ecuaciones 3.34, 3.35 y 3.36 representan el modelo matemático del sistema Grúa-Puente controlado por corriente de armadura, tales ecuaciones pueden tomar la forma,

$$m_{11} \ddot{r} + p_{11} \dot{r} + m_{12} \cos(\theta) \ddot{\theta} + p_{12} \sin(\theta) \dot{\theta}^2 = u_1 \quad (3.34)$$

$$m_{22} \ddot{\theta} + p_{22} \dot{\theta} + m_{21} \cos(\theta) \ddot{r} + d_{21} \sin(\theta) = 0 \quad (3.35)$$

donde, las ecuaciones 3.34 y 3.35 se pueden expresar matricialmente,

$$M(x)\ddot{x} + P(x, \dot{x})\dot{x} + Q(x) = u \quad (3.36)$$

donde x y u son los estados,

$$x = \begin{bmatrix} r \\ \theta \end{bmatrix} \quad u = \begin{bmatrix} u_1 \\ 0 \end{bmatrix}$$

las matrices M , P y Q del modelo son de la siguiente forma,

$$M = \begin{bmatrix} m_{11} & m_{12} \cos \theta \\ m_{21} \cos \theta & m_{22} \end{bmatrix} \quad P = \begin{bmatrix} p_{11} & p_{12} \sin(\theta)\dot{\theta} \\ 0 & p_{22} \end{bmatrix} \quad Q = \begin{bmatrix} 0 \\ d_{21} \sin \theta \end{bmatrix}$$

donde los elementos de las matrices M , P y Q del sistema Grúa-Puente son,

$$m_{11} = \left(\frac{n r_p R_a}{K_m K_a} \right) (m_c + m_v + m_e + \frac{J_{eq}}{n^2 r_p^2}) \quad (3.37)$$

$$m_{12} = \left(\frac{n r_p R_a}{K_m K_a} \right) (m_e l_e + m_v l_v) \quad (3.38)$$

$$m_{21} = m_e l_e + m_v l_v \quad (3.39)$$

$$m_{22} = m_e l_e^2 + m_v l_v^2 \quad (3.40)$$

$$p_{11} = \left(\frac{n r_p R_a}{K_m K_a} \right) \left(F_f + \frac{B_{eq}}{n^2 r_p^2} + \frac{K_m K_b}{n^2 r_p^2 R_a} \right) \quad (3.41)$$

$$p_{12} = - \left(\frac{n r_p R_a}{K_m K_a} \right) (m_e l_e + m_v l_v) \quad (3.42)$$

$$p_{22} = C \quad (3.43)$$

$$d_{21} = (m_e l_e + m_v l_v) g \quad (3.44)$$

3.5. Modelos de los Fenómenos No-Lineales

Los fenómenos no lineales presentes en todo sistema mecánico tienen una influencia más o menos importante en el desempeño del sistema controlado. Su inclusión en el diseño de un sistema de control depende de la relación entre el grado de precisión que se desea alcanzar y el efecto relativo de los fenómenos no lineales en el comportamiento de la planta.

En esta investigación se ha considerado necesario modelar los fenómenos no lineales correspondientes a la zona muerta del motor-reductor DC acoplado a la polea y a la fricción del carro o cabezal. Incluir estos fenómenos en el modelo permite contar con una simulación del sistema controlado con una respuesta más cercana a la experimental, de modo que el proceso de ajuste de los parámetros del controlador en la fase experimental se reduzca sustancialmente.

3.5.1. Zona Muerta del Motor

Se llama “zona muerta” al fenómeno no lineal estático que describe la insensibilidad de un sistema a señales pequeñas, [18]. Aunque existen muchas aplicaciones donde las características de una zona muerta son altamente deseables, en la mayoría de las aplicaciones de lazo cerrado da lugar a efectos indeseables en la dinámica de la retroalimentación y en el desempeño del sistema de control.

El fenómeno de zona muerta tiene la relación estática mostrada en la figura 3.5. Un modelo matemático es dado por

$$\tau = D(u) = \begin{cases} m_-(u + d_-), & u \leq -d_-, \\ 0, & -d_- < u < -d_+, \\ m_+(u - d_+), & u \geq d_+. \end{cases} \quad (3.45)$$

En el caso de los motores DC, la zona muerta es causada por la fricción. En esta investigación se busca determinar los umbrales de la zona muerta para incluir un compensador de este fenómeno no lineal en el sistema de control. Para

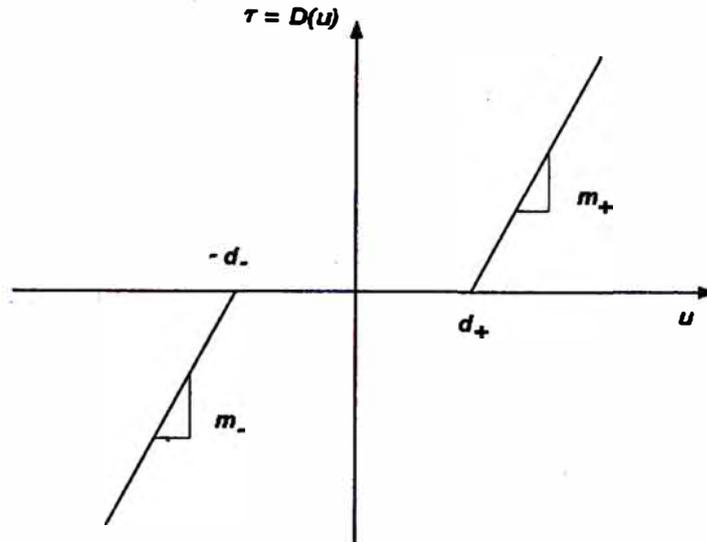


Figura 3.5: Modelo de la zona muerta.

este efecto, resulta útil considerar a la zona muerta como

$$\tau = D(u) = u - sat_d(u), \quad (3.46)$$

donde la función de saturación asimétrica es definida por

$$sat_d(u) = \begin{cases} -d_-, & u < -d_-, \\ u, & -d_- \leq u < -d_+, \\ d_+, & d_+ \leq u. \end{cases} \quad (3.47)$$

Esta descomposición resulta útil para proceder a la compensación, dejando de lado los parámetros m_- y m_+ que dependerán del modelo del sistema afectado por la zona muerta, que en el caso del motor-reductor DC se encuentra disponible independientemente.

En el apéndice B.2 se describe el procedimiento para determinar la zona muerta. De acuerdo con éste se ha determinado que se puede considerar aproximadamente simétrica y con un valor umbral aproximado d :

$$dz = 2.2V \quad (3.48)$$

3.5.2. Fricción

La fricción es la resistencia natural al movimiento relativo entre dos cuerpos en contacto y es esencial para la operación de los sistemas mecánicos comunes (tales como ruedas, frenos, etc). Pero en la mayoría de procesos industriales, representa también un problema; ya que es difícil de modelar y hace más difícil el diseño de los sistemas de control. En general, durante la manufactura se busca reducir la fricción; la fricción que no puede reducirse debe ser compensada por el sistema de control.

El modelo clásico de la fuerza de fricción que es proporcional a la carga, se opone al movimiento y es independiente del área de contacto fue conocido por Leonardo da Vinci. El modelo de Da Vinci fue redescubierto por Coulomb y es ampliamente usado actualmente. Este modelo queda descrito por

$$F(v) = a \cdot \text{sgn}(v), \quad (3.49)$$

donde v es la velocidad relativa y $F(v)$ es la correspondiente fuerza o torque. El parámetro a representa el valor constante de la fuerza de fricción. El modelo de fricción de Coulomb se muestra en la figura (a). Un modelo más detallado se muestra en la figura (b), que incluye la fricción viscosa, la cual es proporcional a la velocidad.

Los experimentos físicos han mostrado que en muchos casos la fuerza requerida para iniciar el movimiento relativo es mayor que la fuerza que se opone la movimiento una vez que éste inicia. Este efecto es conocido como fricción estática. La figura (c) grafica el modelo que incluye el efecto de la fricción estática.

Un modelo completo y apropiado para sistemas de control de dispositivos

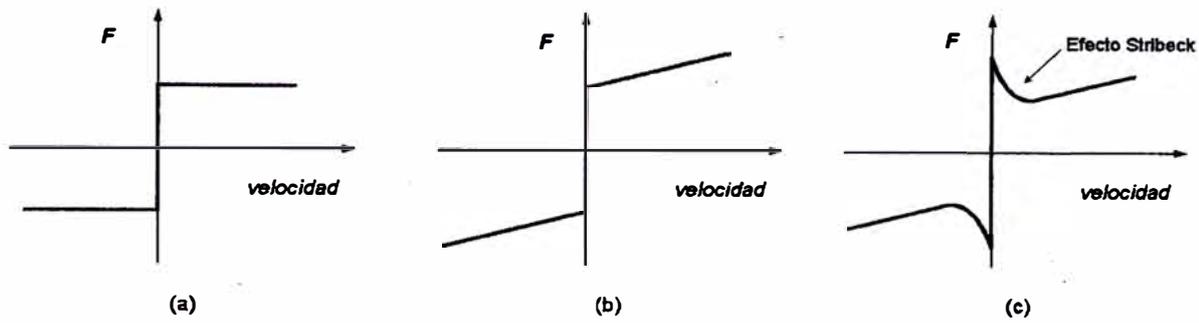


Figura 3.6: Modelos de fricción.

mecánicos de precisión fue dado por Canudas de Wit, [19]. Este modelo estático es el siguiente:

$$F(v) = [\alpha_0 + \alpha_1 e^{-\beta_1 |v|} + \alpha_2 (1 - e^{-\beta_2 |v|}) v] \text{sgn}(v), \quad (3.50)$$

donde la fricción de Coulomb está dada por α_0 , la fricción estática es $(\alpha_0 + \alpha_1)$, y α_2 representa al modelo de fricción viscoso. El efecto por el cual para pequeños valores de v la fuerza de fricción decrece con la velocidad es llamado *viscosidad negativa* o *efecto Stribeck*. El efecto Stribeck es modelado con un segundo término exponencial en el modelo 3.50. Este modelo de fricción captura todos los efectos mostrados en la figura (c).

Es importante resaltar que existen modelos dinámicos más completos para describir el comportamiento de la fricción que los modelos estáticos presentados. Entre estos el más frecuentemente encontrado en la literatura es el de Lund-Grenoble, [20].

Para esta investigación se ha considerado suficiente el modelo de Coulomb (3.49) ya que se incluye el efecto de la fricción viscosa en el modelo original de la planta. Esta fricción viscosa se modela en función de algunos parámetros conocidos y los desconocidos se ajustan mediante un algoritmo genético que aproxima el modelo a los resultados experimentales de una prueba en lazo abierto (ver B.4).

Además, la fricción estática se incluye en una medida considerable como parte de la zona muerta del subsistema motor-polea, que se vio en la sección anterior.

En B.3 se muestra el procedimiento para determinar el valor de la fricción de Coulomb a partir de 3.49 y

$$a = \mu_k N, \quad (3.51)$$

donde μ_k es una constante que se conoce como el coeficiente de fricción cinético.

El valor de μ_k , determinado en B.3 es:

$$\mu_k = 0.0569. \quad (3.52)$$

3.5.3. Inclusión de los Fenómenos No-Lineales en el Modelo de la Grúa-Puente

En la figura 3.7 se muestra el diagrama de bloques del sistema grúa-puente incluyendo el efecto de los fenómenos no lineales considerados: la zona muerta del actuador y la fricción del cabezal con el riel.

La entrada u pasa por el bloque correspondiente a la zona muerta, de modo que el sistema sólo percibe la señal de entrada cuando ésta supera el efecto de este fenómeno no lineal. El subsistema motor-polea es excitado por la entrada modificada y da lugar a la generación de un torque T_m que da lugar a la fuerza F a través de la acción de la polea. En el sistema carro péndulo, la fuerza de fricción F_f , que depende del sentido de movimiento afecta al movimiento del cabezal, oponiéndose a éste. Puede verse en el diagrama de bloques que la fricción depende de la derivada de la posición (\dot{x}) y actúa sobre el sistema con signo negativo, para representar su efecto de oposición al movimiento.

La expresión algebraica de esta inclusión de los efectos no lineales puede

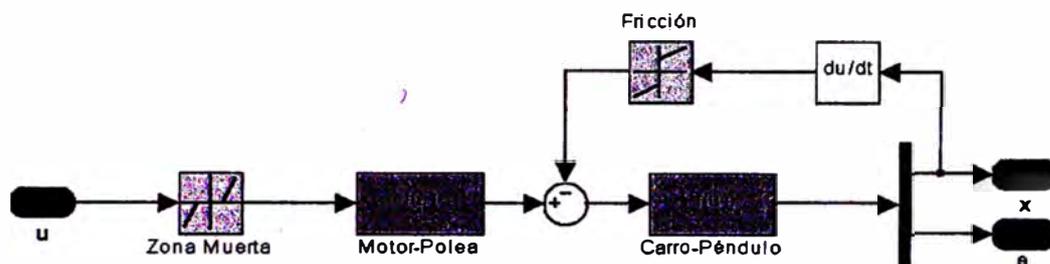


Figura 3.7: Diagrama del sistema con la inclusión de fenómenos no lineales.

apreciarse en el programa mostrado en el apéndice C.5.

El diagrama de bloques muestra a la fricción restándose de la fuerza proporcionada por el subsistema motor-polea. Debe tenerse en cuenta que la fuerza de fricción es una perturbación sobre el sistema global, ya que los subsistemas motor-polea y carro-péndulo están acoplados. El acoplamiento entre los subsistemas motor-polea y carro-péndulo puede observarse en la ecuación 3.36, donde los elementos de las matrices M y P relacionan a ambos subsistemas. La inercia del subsistema carro-péndulo afecta el movimiento del eje del motor, así como el movimiento del subsistema carro-péndulo se ve afectado por la inercia del subsistema motor-polea.

3.6. Respuesta en Lazo Abierto del Sistema

Se mostrará en esta sección la respuesta del modelo propuesto al escalón unitario.

Se considera el diagrama de bloques mostrado en la figura 3.8. Para poder observar una respuesta, se omitirá el efecto de la zona muerta, de modo que la

señal constante de 1 voltio produzca un efecto en el sistema. De incluirse el efecto de la zona muerta, el sistema sería insensible a esta entrada, como en efecto es.

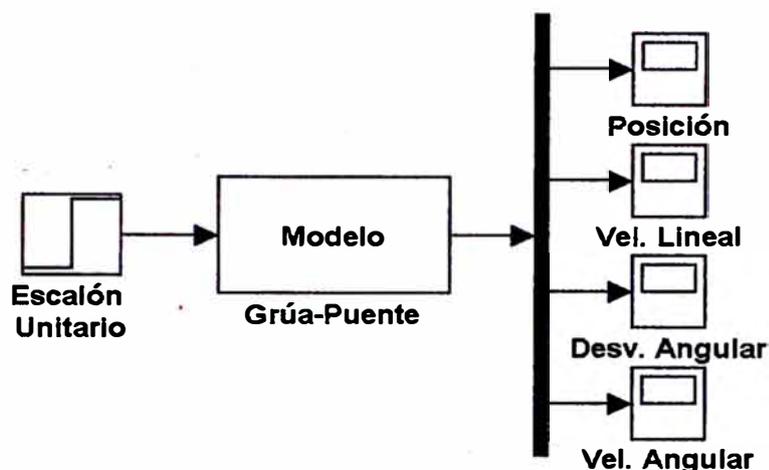


Figura 3.8: Diagrama de bloques del sistema grúa-puente en lazo abierto.

Los valores de los parámetros considerados para esta simulación son aquellos obtenidos a través de la documentación, mediciones y experimentos mencionados hasta este punto. Más adelante se verá la respuesta en lazo abierto del sistema con los parámetros ajustados por un algoritmo genético. En las tablas 3.5 y 3.6 se muestran los valores que dan lugar a los gráficos 3.9 y 3.10.

Variable	Descripción	Valor
m_c	Masa del Cabezal	0.67
m_e	Masa de la Carga	2.345
m_v	Masa del péndulo	0.1724
l_e	Distancia del punto P al C.G de la Carga	0.83
l_v	Distancia del punto P al C.G del P'endulo	0.415
μ_k	Coefficiente de Fricción Cinética del Cabezal	0.052
g	Gravedad	9,81

Cuadro 3.5: Valores de los parámetros del subsistema cabezal-péndulo.

Variable	Descripción	Valor
K_m	Constante de Torque del Motor	4.94×10^{-2}
K_b	Constante de Fuerza Contraelectromotriz	4.94×10^{-2}
K_a	Constante de Ganancia del Amplificador	1.00
R_a	Resistencia de Armadura	5.8
J_{eq}	Momento de Inercia Equivalente	3.056×10^{-5}
B_{eq}	Constante de Fricción Viscosa Equivalente	3.935×10^{-5}
r_p	Radio de la Polea	0.03785
J_p	Momento de Inercia de la Polea	4.298×10^{-4}
B_p	Constante de Fricción Viscosa de la Polea	9.90×10^{-3}
n	Relación de Reducción	5.9

Cuadro 3.6: Valores de los parámetros del subsistema motor-polea.

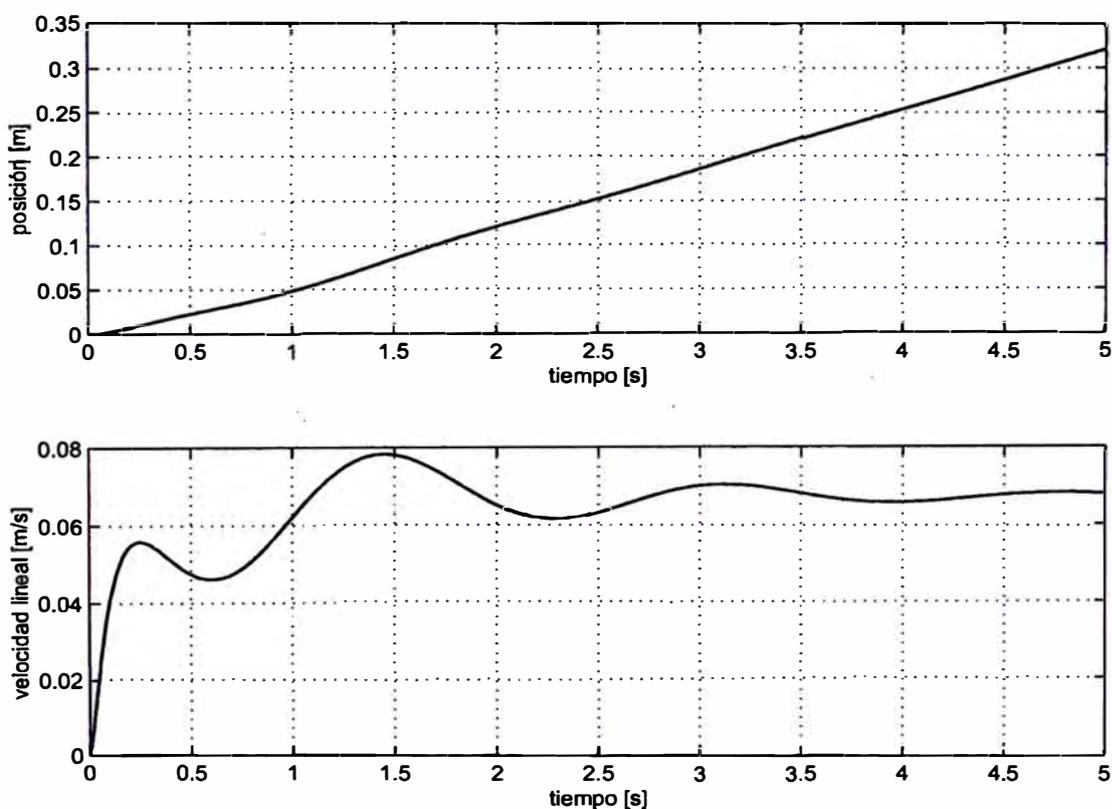


Figura 3.9: Respuesta del Cabezal a una Entrada Escal'on.

3.7. Parámetros Ajustados mediante un Algoritmo Genético

En las secciones anteriores de este capítulo, se ha buscado describir matemáticamente el comportamiento de la planta bajo un modelo dinámico basado en las ecuaciones de Euler-Lagrange, considerando el accionamiento del sistema

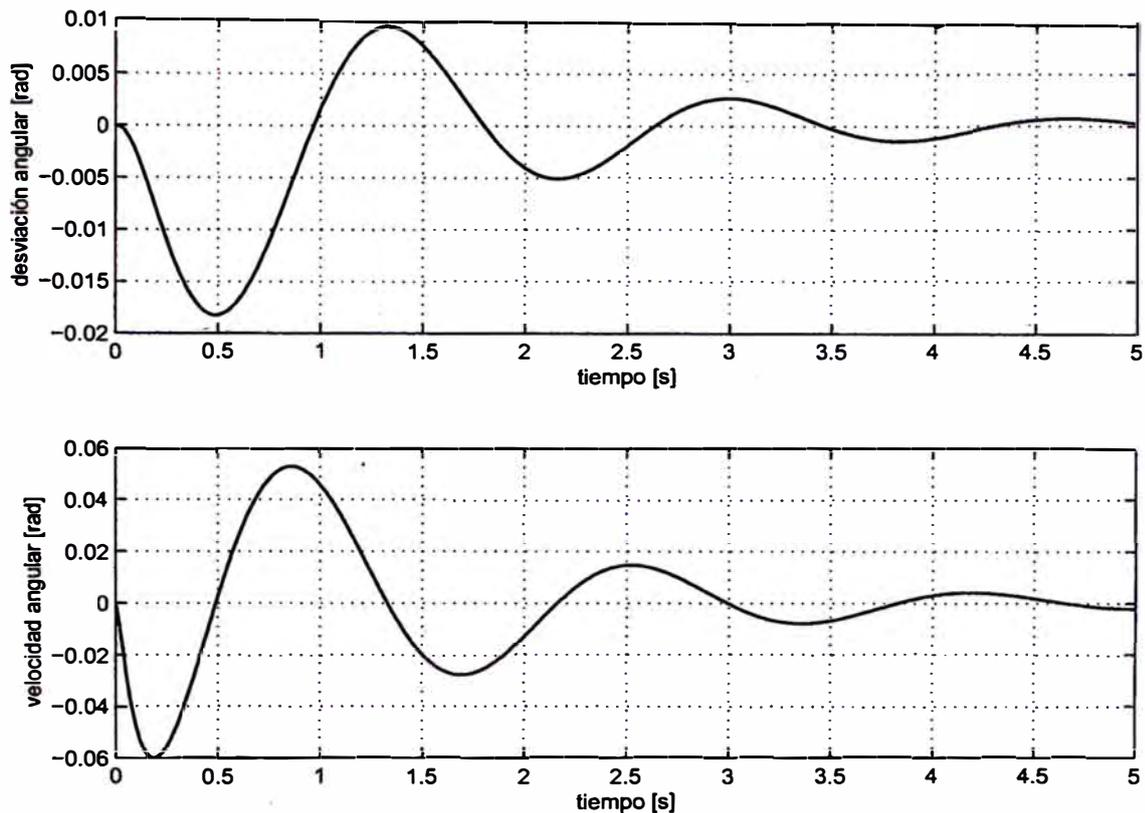


Figura 3.10: Respuesta del Péndulo a una Entrada Escalón.

mecánico a partir del modelo de segundo orden de un motor-reductor DC e incluyendo el efecto de los fenómenos no lineales correspondientes a la fricción y a la zona muerta. Cada subsistema tiene sus propios parámetros, y estos han sido medidos o calculados individualmente; sin embargo, sobre algunos de estos, los procedimientos para su medición dan lugar a una considerable imprecisión. Además, los pequeños errores individuales en la estimación de los parámetros de la planta dan lugar a una respuesta ligeramente diferente, pero apreciable, entre la respuesta de la simulación y la respuesta experimental. Se considera deseable determinar con mayor precisión los parámetros cuya estimación o medición no es muy precisa, mediante un procedimiento de optimización global, de modo que el ajuste de estos parámetros aproxime mejor la respuesta global del modelo matemático con la respuesta experimental de la planta.

Los algoritmos genéticos reciben en la actualidad una gran atención por

parte de la comunidad científica como un método de optimización versátil para problemas de optimización de sistemas continuos y discretos. En el caso de este informe, se empleará un algoritmo genético para aproximar la respuesta simulada a la experimental mediante la modificación iterativa de tres parámetros: B_p , μ_k y d_z , que son, respectivamente, la constante de fricción viscosa de las poleas, el coeficiente de fricción cinético del cabezal y el ancho de la zona muerta del motor-reductor.

El coeficiente B_p se determina en el apéndice B.1, pero con la polea sin carga; sucede lo mismo con μ_k ; y, en el caso de d_z se realiza un experimento en condiciones estáticas, que no reflejan el comportamiento dinámico de este fenómeno. Debido a lo anterior, se escogen estos tres parámetros para ser ajustados por el algoritmo genético que se detalla en el apéndice B.4.

A continuación se muestra una tabla con los parámetros ajustados:

Variable	Descripción	Valor
μ_k	Coefficiente de Fricción Cinética del Cabezal	0.052903
B_p	Constante de Fricción Viscosa de la Polea	9.194×10^{-3}
d_z	Zona Muerta del Motor-reductor	2.8387

Cuadro 3.7: Valores ajustados por el algoritmo genético.

De este modo, los parámetros finales del modelo se resumen en las tablas 3.8 y 3.9.

Variable	Descripción	Valor
m_c	Masa del Cabezal	0.67
m_e	Masa de la Carga	2.345
m_v	Masa del péndulo	0.1724
l_e	Distancia del punto P al C.G de la Carga	0.83
l_v	Distancia del punto P al C.G del Péndulo	0.415
μ_k	Coefficiente de Fricción Cinética del Cabezal	0.052903
g	Gravedad	9,81

Cuadro 3.8: Valores de los parámetros del subsistema cabezal-péndulo.

Variable	Descripción	Valor
K_m	Constante de Torque del Motor	4.94×10^{-2}
K_b	Constante de Fuerza Contraelectromotriz	$4,94 \times 10^{-2}$
K_a	Constante de Ganancia del Amplificador	1.00
R_a	Resistencia de Armadura	5.8
J_{eq}	Momento de Inercia Equivalente	3.056×10^{-5}
B_{eq}	Constante de Fricción Viscosa Equivalente	3.935×10^{-5}
r_p	Radio de la Polea	0.03785
J_p	Momento de Inercia de la Polea	4.298×10^{-4}
B_p	Constante de Fricción Viscosa de la Polea	9.194×10^{-3}
n	Relación de Reducción	5.9
dz	Amplitud de la Zona Muerta	2.84
n	Relación de Reducción	5.9

Cuadro 3.9: Valores de los parámetros del subsistema motor-polea.

En las figuras 3.11, 3.12 y 3.13 se muestra un contraste entre la respuesta del modelo optimizado y la respuesta del prototipo ante la misma excitación. Esta excitación es una entrada de voltaje registrada en un experimento con un controlador difuso. Esta entrada se empleó para el procedimiento de optimización, debido a que se trata de una señal compleja y que da lugar a una respuesta acotada. Es importante que se trate de una señal compleja para poder probar el sistema ante diferentes excitaciones en un mismo experimento y a la vez es necesario obtener una respuesta acotada, ya que el prototipo tiene una longitud definida y debe operarse dentro de ciertos límites o especificaciones técnicas.

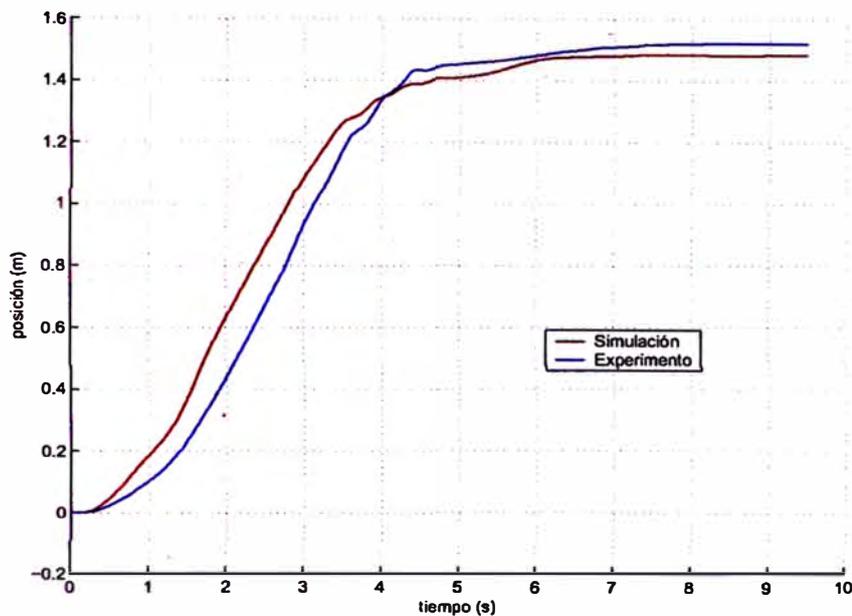


Figura 3.11: Comparación entre la respuesta en lazo abierto de la planta y el modelo con los parámetros ajustados por un algoritmo genético: Posición.

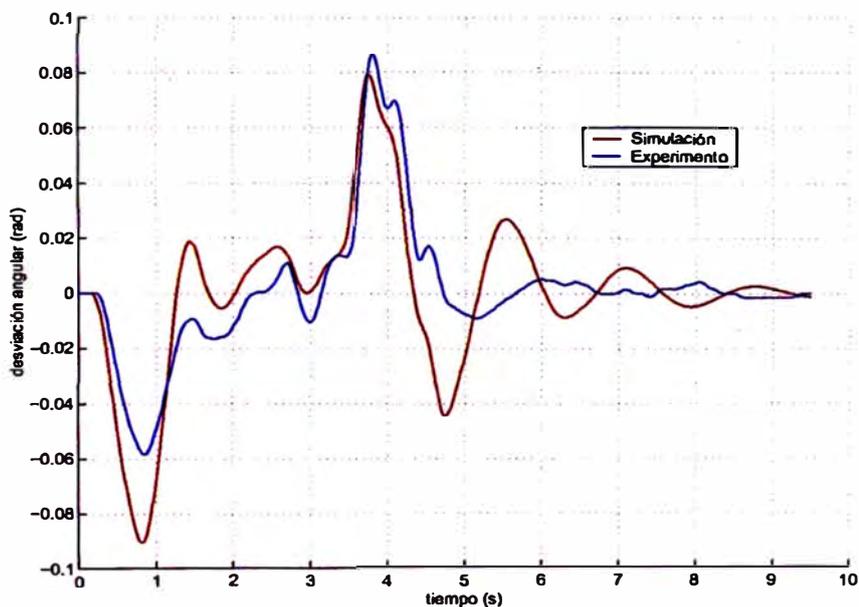


Figura 3.12: Comparación entre la respuesta en lazo abierto de la planta y el modelo con los parámetros ajustados por un algoritmo genético: Desviación angular.

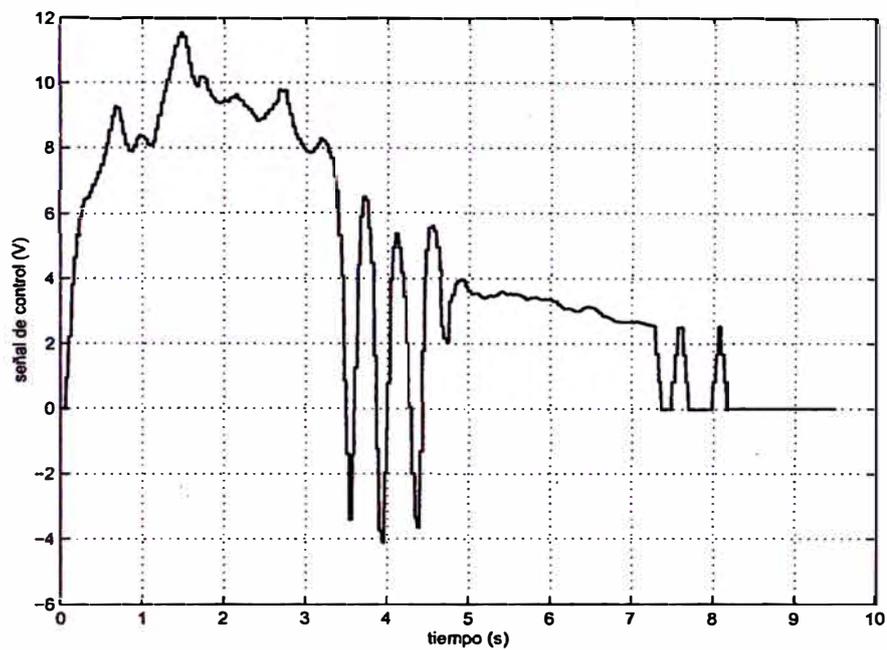


Figura 3.13: Comparación entre la respuesta en lazo abierto de la planta y el modelo con los parámetros ajustados por un algoritmo genético: Señal de control empleada.

CAPITULO 4

SISTEMA DE CONTROL Y SIMULACIONES

En este capítulo se probará el controlador diseñado mediante simulaciones. Dos categorías principales agrupan a los resultados simulados; en el primero se considera al sistema de control en tiempo continuo y en el segundo, en tiempo discreto. En la segunda sección se observa el efecto de la discretización en los sensores, así como la influencia de la elección del período de muestreo en el desempeño del sistema. En cada caso se probará el controlador modificando la masa de la carga, con el fin de analizar la robustez del controlador ante esta variación.

En esta investigación se considerará admisible un error de posicionamiento de la carga, debido a la oscilación, igual a $\pm 0.5\%$ de la longitud suspendida. Este porcentaje equivale a una amplitud angular de oscilación residual de ± 0.3 grados sexagesimales. El error admisible en la posición del cabezal se considerará como $\pm 4\text{mm.}$, para el prototipo de grúa-puente empleado en los experimentos. Asimismo, el sobreimpulso admisible será equivalente a $\pm 5\%$ del recorrido, y el tiempo de establecimiento no deberá ser mayor a 7 segundos, que corresponden a 4 segundos para la aproximación a la posición final y 3 segundos para el posicionamiento fino y estabilización.

4.1. Diseño del Controlador por Lógica Difusa

El controlador basado en lógica difusa que se propone consiste de dos sistemas de inferencia en paralelo, cuyas salidas se combinan linealmente. Siendo el primer FIS un seguidor de la trayectoria generada especialmente para este FLC, mientras que el segundo FIS compensa las oscilaciones que se presentan durante el traslado de la carga. En la figura 4.1 se muestra el diagrama de bloques de este FLC.

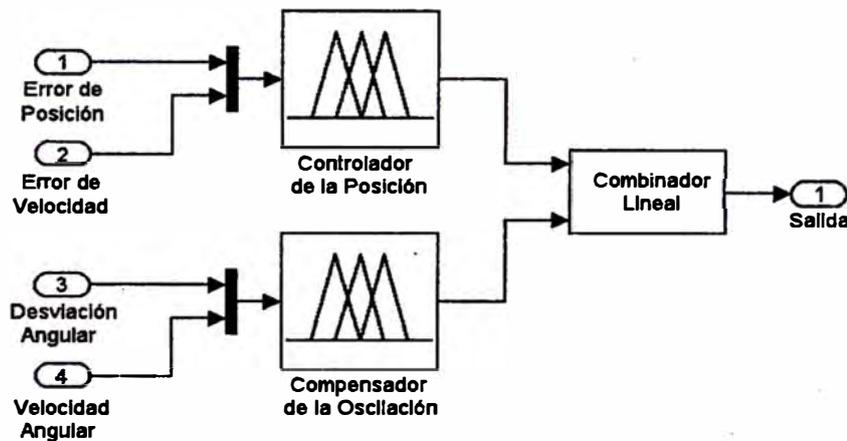


Figura 4.1: Diagrama de bloques del FLC propuesto.

El primer paso consiste en generar una trayectoria eficiente para el movimiento del cabezal, lo que se verá en la sección 3.2.1. Se diseñarán los dos controladores considerando la robustez ante la variación de parámetros. Se parte de dos fuentes para el diseño de los FLC: el conocimiento de ingeniería del diseñador y el uso del *algoritmo de asignación por análisis del plano de fase*, [21]. El controlador inicial es luego mejorado a través de la simulación y experimentación.

Este FLC se propone como un sistema paralelo compuesto por dos mecanismos de inferencia del tipo Takagi-Sugeno, para estos sistemas de inferencia se determinarán primero las funciones de membresía. Con el fin de definir las funciones de membresía es necesario determinar una división apropiada del rango de

operación del sistema. Después, se procederá a definir una base de reglas mediante el algoritmo de asignación por análisis del plano de fase. En una investigación previa se ha determinado que una defuzzificación de orden nulo es suficiente para un buen desempeño del sistema de control, además es más simple de procesar numéricamente en el DSP, lo que permite tener un menor tiempo de cálculo, [22].

Mediante un proceso de simulaciones y experimentación se determinará posteriormente el valor de la constante de combinación lineal (K_{comb}) entre los dos sistemas de inferencia. Del mismo modo se determinará el valor de la ganancia final (K_{prop}), que da un factor de escala a la salida del controlador.

4.1.1. Planeamiento de la Trayectoria

La trayectoria puede ser diseñada usando una técnica de formación de la señal de entrada en función del modelo de la grúa, bajo la aplicación de una simulación de un controlador óptimo. Sin embargo, en presencia de imprecisiones en el modelo y de perturbaciones externas, la trayectoria resultante requeriría un tiempo mayor para el traslado, [5]. Al considerar la inclusión de un controlador para compensar las oscilaciones se puede diseñar una trayectoria para obtener un tiempo de traslado óptimo. La solución típica para el movimiento de un cuerpo rígido sometido a restricciones de velocidad y aceleración consiste de un perfil escalonado de aceleración y desaceleración. Este perfil escalonado de aceleraciones, al integrarse, origina un perfil trapezoidal de velocidades. Para un tiempo de movimiento suficientemente largo, de modo que la velocidad máxima pueda ser alcanzada, los perfiles de aceleración y velocidad son los que se muestran en la Figura 4.2. El tiempo final para desplazarse una distancia Z se define como dos veces el período de oscilación natural del péndulo (T), y los tiempo de aceleración y desaceleración como $3T/8$. Esto da lugar a:

$$Z = V_{max} \cdot \frac{5T}{4} + V_{max} \cdot \frac{3T}{8} \quad (4.1)$$

Simplificando:

$$Z = V_{max} \cdot \frac{13T}{8} \quad (4.2)$$

La ecuación 4.2 permite calcular directamente el valor de V_{max} , con el que se define el perfil trapezoidal de la trayectoria. Un ejemplo de este perfil es el que se aprecia en la figura 4.2.

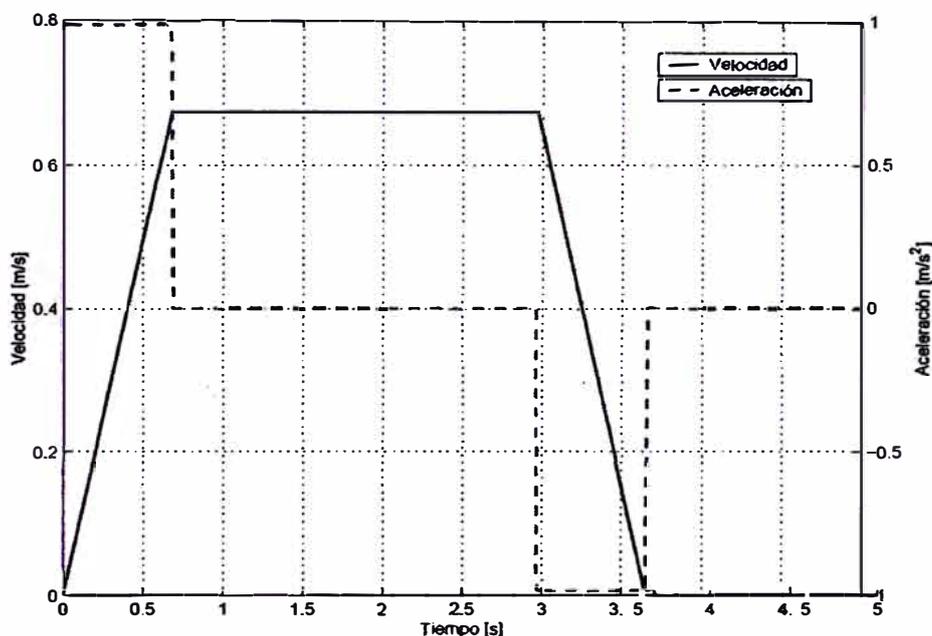


Figura 4.2: Perfiles de aceleración y velocidad de la trayectoria.

4.1.2. Sistema de Inferencia Difusa del Tipo Takagi-Sugeno para el Control de la Posición

Para el Sistema de Inferencia Difusa para el control de la posición se han considerado las funciones de membresía Gaussianas que se muestran en las figuras 4.3 y 4.4, que corresponden a la posición y a la velocidad, respectivamente.

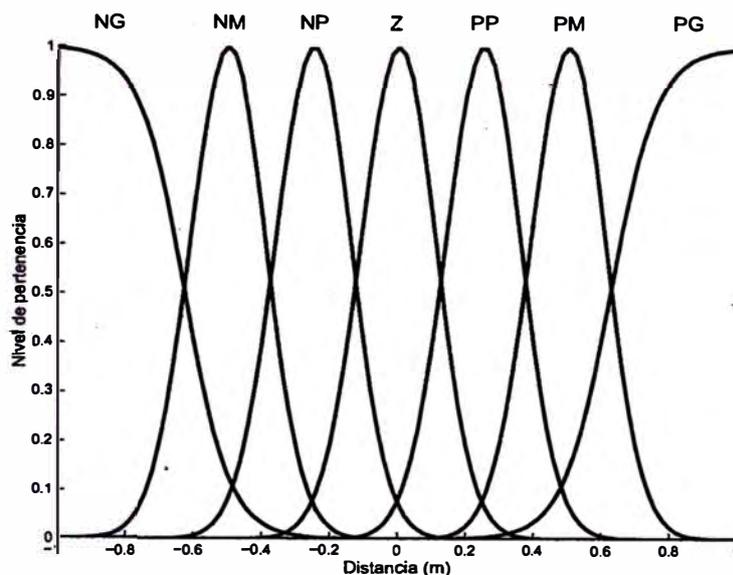


Figura 4.3: Funciones de membresía para la posición.

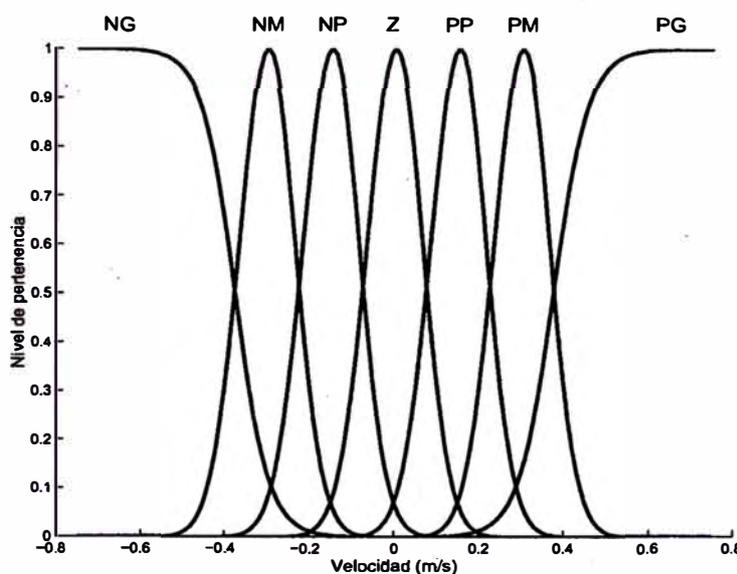


Figura 4.4: Funciones de membresía para la velocidad lineal.

La base de reglas para el sistema de inferencia difusa para el control de la posición del cabezal se ilustra en la tabla 4.1 y corresponde al modelo de inferencia de Mamdani, que busca el equilibrio de las acciones de control, [15].

Posteriormente, un mecanismo de inferencia, tal como la regla composicional de Zadeh, se usa para determinar la acción de control. El paso final involucra la

		Error de Velocidad Lineal						
		NG	NM	NP	Z	PP	PM	PG
Error de Posición	NG	NG	NG	NM	NM	NP	NP	Z
	NM	NG	NM	NM	NP	NP	Z	PP
	NP	NM	NM	NP	NP	Z	PP	PP
	Z	NM	NP	NP	Z	PP	PP	PM
	PP	NP	NP	Z	PP	PP	PM	PM
	PM	NP	Z	PP	PP	PM	PM	PG
	PG	Z	PP	PP	PM	PM	PG	PG

Cuadro 4.1: Base de reglas del controlador de la posición.

aplicación de un método de defuzzificación.

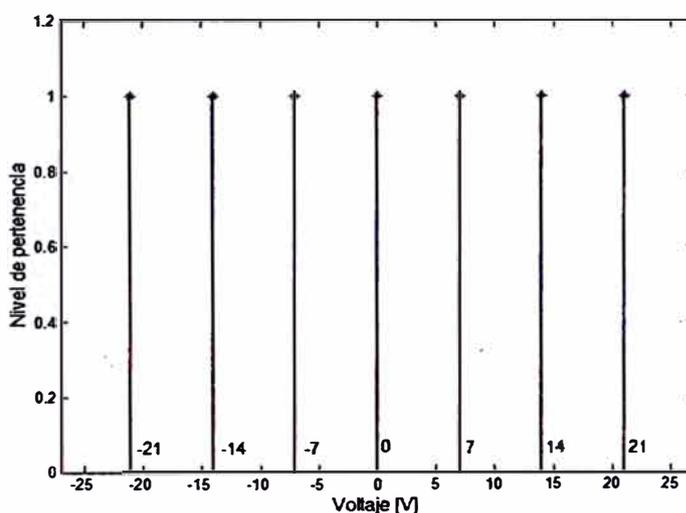


Figura 4.5: Funciones de membresía C_i para la defuzzificación.

La defuzzificación que se propone, dentro del esquema de los FIS tipo Takagi-Sugeno, es la de orden 0. Esta defuzzificación genera, como salida del FIS, el promedio ponderado de los niveles de pertenencia obtenidos a través del proceso de inferencia y los valores *crisp* correspondientes a las funciones de membresía de salida (C_i), que son implicadas por la base de reglas. En la figura 4.5 se muestran las funciones de membresía de salida. La ecuación 4.3 expresa el cálculo de la salida u_x del FIS de seguimiento, obtenida a partir del promedio ponderado o defuzzificación de orden 0. N_C es el número de funciones de pertenencia que

describen al conjunto difuso de salida “*Voltaje*”, conformado por las funciones C_i , y μ_{C_i} representa al nivel de pertenencia asociado con cada C_i .

$$u_x = \frac{\sum_{i=1}^{N_C} \mu_{C_i} \cdot z_{C_i}}{\sum_{i=1}^{N_C} \mu_{C_i}} \quad (4.3)$$

Con el fin de evaluar la estabilidad y convergencia del sistema de control con la base de reglas considerada, se realiza un análisis del diagrama de fase. Para esto, se construye una partición de grupos de celdas que se relaciona biunívocamente con las reglas de la base de reglas. La figura 4.6 muestra una partición de estados típica con la numeración de grupos de celdas empleada en el esquema propuesto.

L_{17}	L_{27}	L_{37}	L_{47}	L_{57}	L_{67}	L_{77}
L_{16}	L_{26}	L_{36}	L_{46}	L_{56}	L_{66}	L_{76}
L_{15}	L_{25}	L_{35}	L_{45}	L_{55}	L_{65}	L_{75}
L_{14}	L_{24}	L_{34}	L_{44}	L_{54}	L_{64}	L_{74}
L_{13}	L_{23}	L_{33}	L_{43}	L_{53}	L_{63}	L_{73}
L_{12}	L_{22}	L_{32}	L_{42}	L_{52}	L_{62}	L_{72}
L_{11}	L_{21}	L_{31}	L_{41}	L_{51}	L_{61}	L_{71}

Figura 4.6: Esquema de una partición típica de grupos de celdas.

En cada grupo de celdas, se considerará el punto central como el estado inicial para la simulación que construirá el diagrama de fase. Se aplicará al sistema

una entrada constante (u_{1i}) para cada (u_1) en el conjunto S_1 y se correrá una simulación hasta que el sistema entre en otro grupo de celdas del espacio de estados, o hasta que el tiempo de simulación se vuelva mayor que un valor fijado t_0 .

El proceso anterior da lugar finalmente a un diagrama de fase exhaustivo, en el que se prueban todas las condiciones iniciales comprendidas por la base de reglas completa. Siendo estos puntos iniciales los centros de las celdas, el diagrama múltiple de fase generado para el controlador de la posición se muestra en la figura 4.7. Con el fin de ilustrar mejor la relación del plano de fase con la base de reglas, en la figura 4.8 se muestra el diagrama de fase junto con las funciones de membresía.

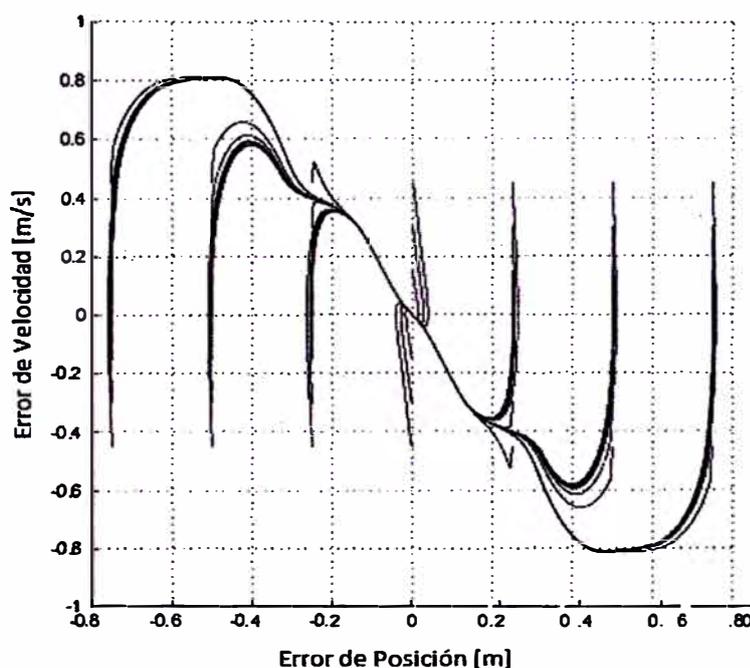


Figura 4.7: Diagrama de fase del controlador de la posición.

Al analizar el plano de fase, se observa que para cada celda, o punto central de cada celda, el controlador lleva el sistema al punto de equilibrio y permanece en él. A partir de esto se puede considerar que el controlador satisface el criterio de estabilidad asintótica, dentro del dominio considerado en el plano de fase. El

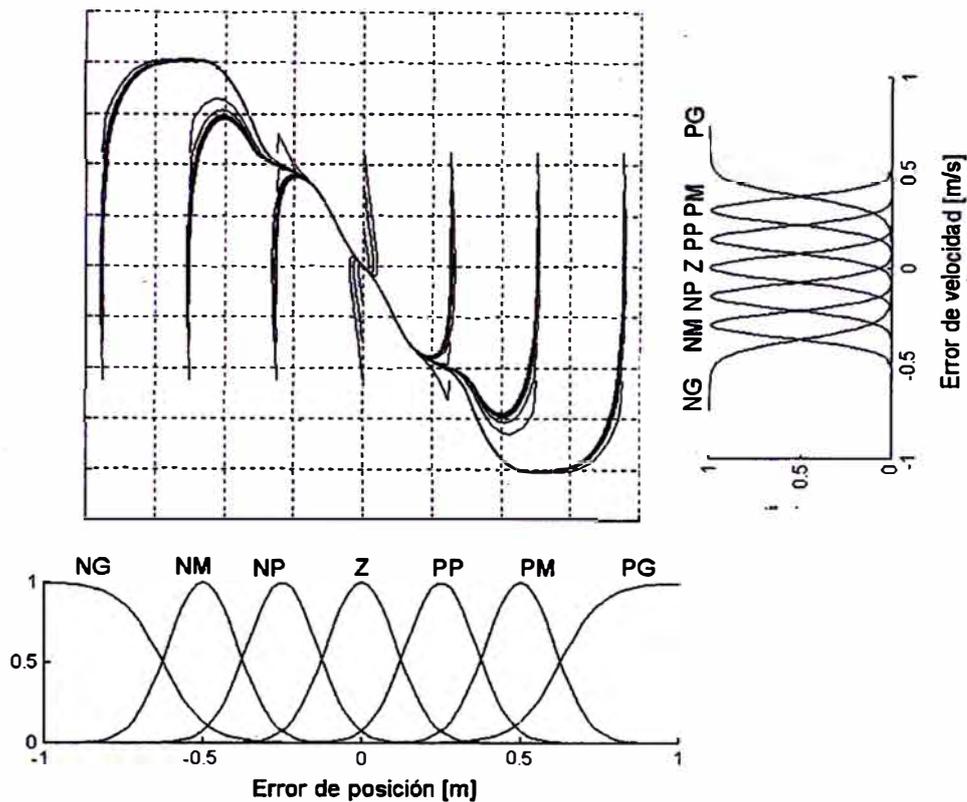


Figura 4.8: Diagrama de fase y funciones de membresía.

estado final se observa como un punto de equilibrio estable.

En [21], Vachtsevanos y Farinwata proponen métodos para el análisis de robustez y estabilidad. Sin embargo, en el caso del análisis de estabilidad, el método busca analizar la estabilidad en forma convencional al linealizar el sistema controlado alrededor del punto de equilibrio para los diferentes casos de aproximación del sistema hacia el punto de equilibrio; vale decir, sería necesario tomar todas las combinaciones que llevan al controlador al equilibrio en la vecindad de la celda objetivo y efectuar un análisis lineal de estabilidad para cada una de estas combinaciones. En esta investigación, se ha considerado suficiente el análisis por el plano de fase y a través de la simulación numérica y experimentación, más aún considerando que el sistema es inherentemente estable y a través del análisis del plano de fase y de las simulaciones y experimentos no se encuentra inestabilidad;

más aun, el controlador acelera drásticamente la estabilización del sistema. En cuanto al análisis de robustez propuesto en [21], el procedimiento es similar al del análisis de estabilidad. Debido a la naturaleza del controlador basado en FIS, en cuanto a su calidad de sistema de control de estructura variable y su complejidad, un análisis de robustez exhaustivo resulta poco práctico; por lo que ésta se analizará en función de la variación de la masa de la carga mediante simulaciones y experimentos.

4.1.3. Sistema de Inferencia Difusa del Tipo Takagi-Sugeno para la Amortiguación de la Oscilación

Para el Sistema de Inferencia Difusa para la amortiguación de la oscilación se han considerado las funciones de membresía Gaussianas que se muestran en las figuras 4.9 y 4.10, correspondientes al ángulo y velocidad angular de la oscilación de la carga.

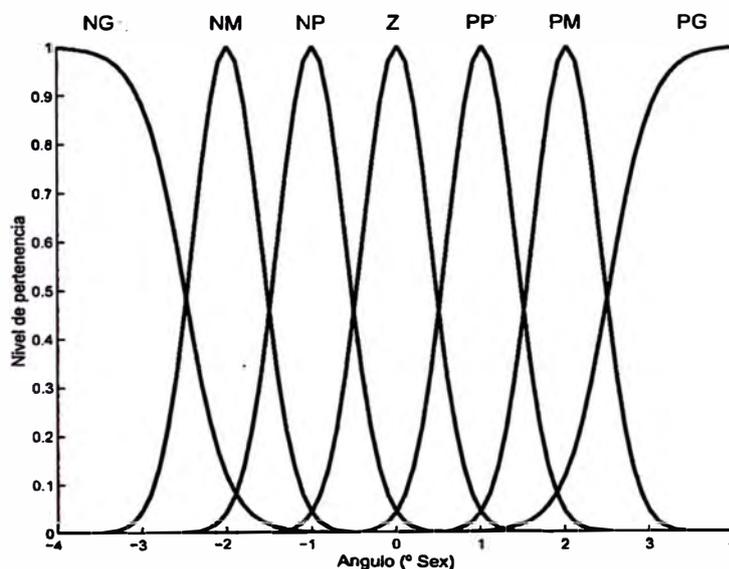


Figura 4.9: Funciones de membresía para la desviación angular.

Tal como se hizo con el control de la posición del cabezal, para la amortiguación de la oscilación se emplea una base de reglas de tipo Mamdani, que se

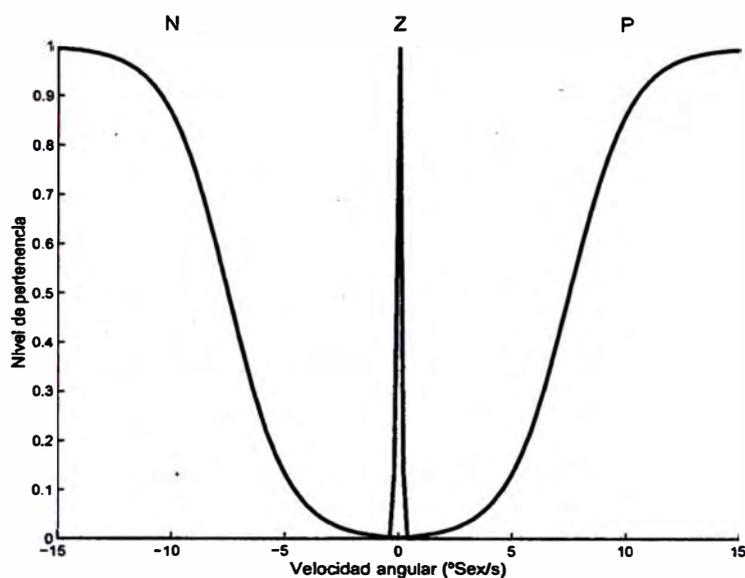


Figura 4.10: Funciones de membresía para la velocidad angular.

ilustra en la tabla 4.2

		Error de Velocidad Angular		
		N	Z	P
Error de Ángulo	NG	NG	Z	NG
	NM	NM	Z	NM
	NP	NP	Z	PP
	Z	Z	Z	Z
	PP	PP	Z	PP
	PM	PM	Z	PM
	PG	PG	Z	PG

Cuadro 4.2: Base de reglas del compensador de la oscilación.

Para este FIS se ha empleado el mismo procedimiento de defuzzificación definido por la ecuación 4.3, empleando el mismo conjunto difuso de salida, visto gráficamente en la figura 4.5.

El procedimiento de generación del diagrama de fase aplicado al controlador de la posición es empleado para obtener el diagrama múltiple de fase que se muestra en la figura 4.11.

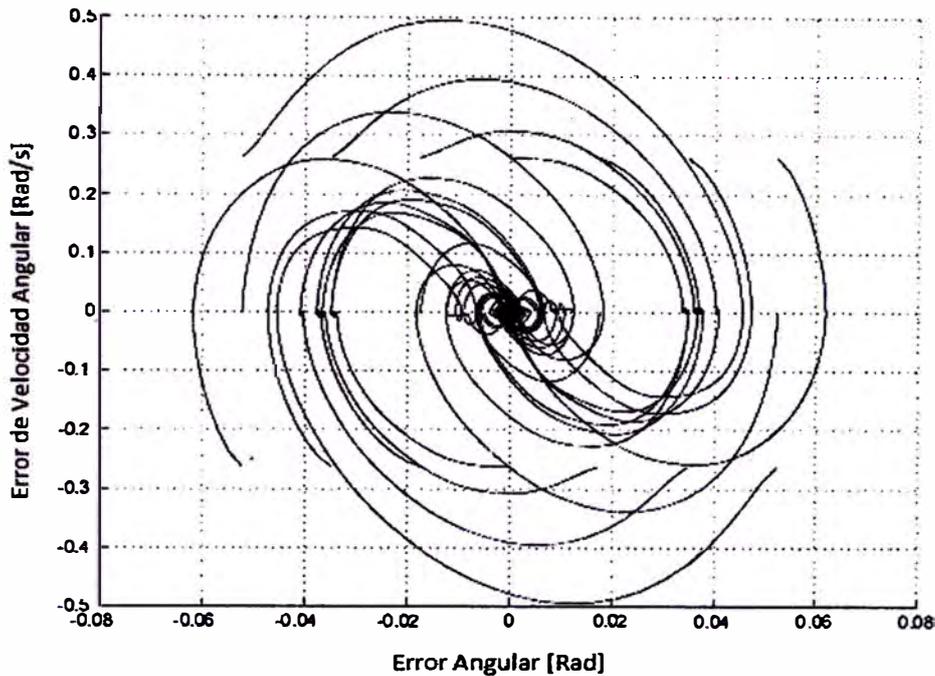


Figura 4.11: Diagrama de fase del compensador de la oscilación.

Al analizar el plano de fase, se observa que para cada celda, o punto central de cada celda, el controlador lleva el sistema a un foco de equilibrio y permanece en él. A partir de esto se puede considerar que el controlador satisface el criterio de estabilidad asintótica, dentro del dominio considerado en el plano de fase. Además de la compensación ejercida por el controlador, la fricción presente en el péndulo disipa la energía del sistema de modo que el sistema termina en el punto de equilibrio rápidamente; o más precisamente, la amplitud de la oscilación disminuye por debajo de lo medible con gran rapidez.

El análisis de robustez se basa en simulaciones y experimentos, tal como se indicó en la sección anterior para el FIS de control de la posición.

4.1.4. Puesta en Paralelo de los Sistemas de Inferencia

Los FIS obtenidos en las dos secciones anteriores son combinados linealmente a través del factor K_{comb} . La salida del combinador lineal pasa por una ganancia

K_{prop} que se ajusta para mejorar la respuesta del lazo de control. Las entradas del FIS de control de posición son multiplicadas por k_X y k_{Xd} que ajustan la escala global de las funciones de membresía de este FIS para tener una respuesta adecuada.

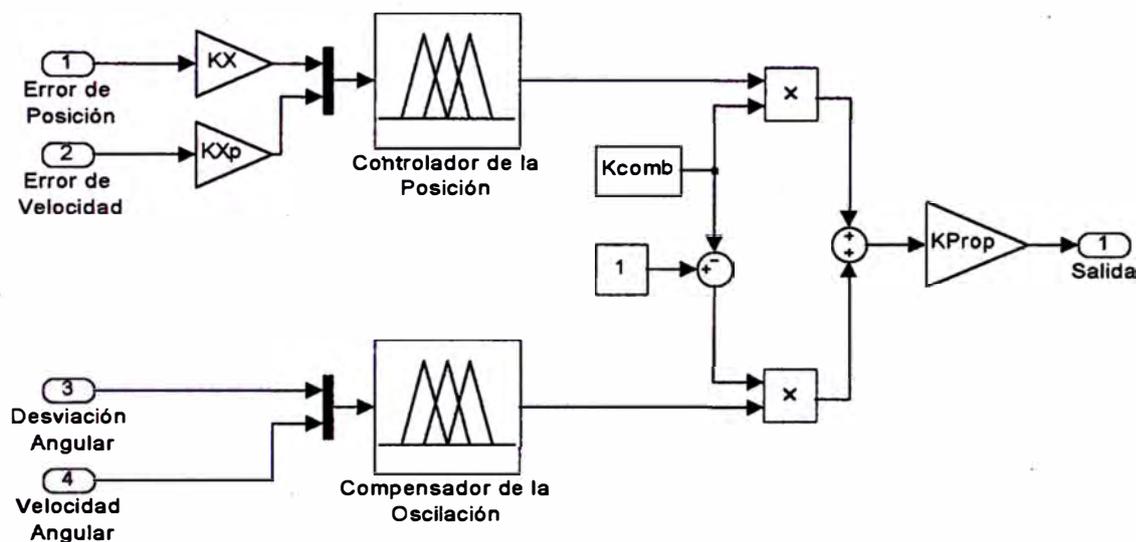


Figura 4.12: Diagrama de la puesta en paralelo de los dos sistemas de inferencia difusa.

La ecuación 4.4 y el diagrama 4.12 describen la puesta en paralelo de los dos FIS.

$$u = K_{prop} \left(K_{comb} \cdot g_x(k_X e_x, k_{Xd} \dot{e}_x) + (1 - K_{comb}) \cdot g_\theta(\theta, \dot{\theta}) \right), \quad (4.4)$$

donde g_x y g_θ representan a los FIS de control de la posición y amortiguación de la oscilación, respectivamente.

4.2. Compensador de la Zona Muerta

Una vez que se ha computado la salida del controlador basado en lógica difusa, y se ha escalado esta salida mediante la multiplicación por K_{prop} , se procede a compensar el efecto de la zona muerta.



Figura 4.13: Diagrama de bloques de la compensación de la zona muerta.

En las ecuaciones 3.46 y 3.47 se describe el modelo de la zona muerta del motor-reductor DC. El compensador de zona muerta propuesto y programado en el DSP sigue un comportamiento inverso al del modelo del fenómeno no lineal. El compensador considera una zona muerta simétrica con ancho igual a dz ; este valor es sumado o restado del valor de la señal de entrada siempre que sea mayor o menor que el ancho de la banda muerta db . La función de la banda muerta es filtrar los valores menores que db , de modo que no se ignoren las señales que corresponden a errores muy pequeños y admisibles; así se evita un consumo innecesario de energía y se esfuerza menos el actuador. El modelo del compensador es como sigue:

$$u_{dz} = \begin{cases} u - dz, & u < -db, \\ 0, & -db \leq u \leq db, \\ u + dz, & u > db. \end{cases} \quad (4.5)$$

En la figura 4.14 se muestra gráficamente la función definida en 4.5. El valor de dz se toma igual al indicado en el capítulo 2 en la sección 3.7, este valor es 2.84 V. El valor de db se ajusta en 0.2 V durante la simulación y experimentación.

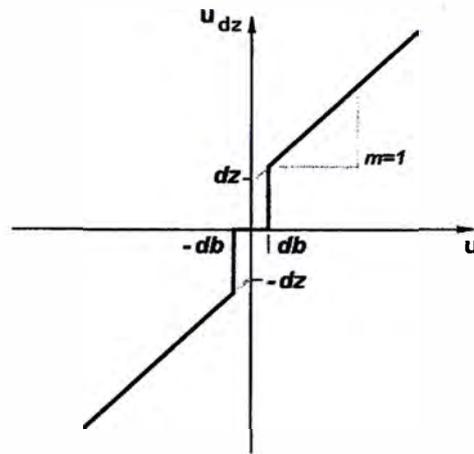


Figura 4.14: Función de compensación de la zona muerta.

Finalmente, en la figura 4.15 se muestra el diagrama de bloques del sistema de control en lazo cerrado, incluyendo el compensador de la zona muerta.

4.3. Simulación del Controlador Continuo

En tanto se considera al controlador continuo, se probará su desempeño ante la influencia de los fenómenos no lineales. En esta serie de simulaciones el tiempo se considera continuo para un período de muestreo global de 5 ms, equivalente a una frecuencia de muestreo de 200 Hz. La diferencia en la respuesta con períodos de muestreo de 1 ms, o inclusive menores, es despreciable.

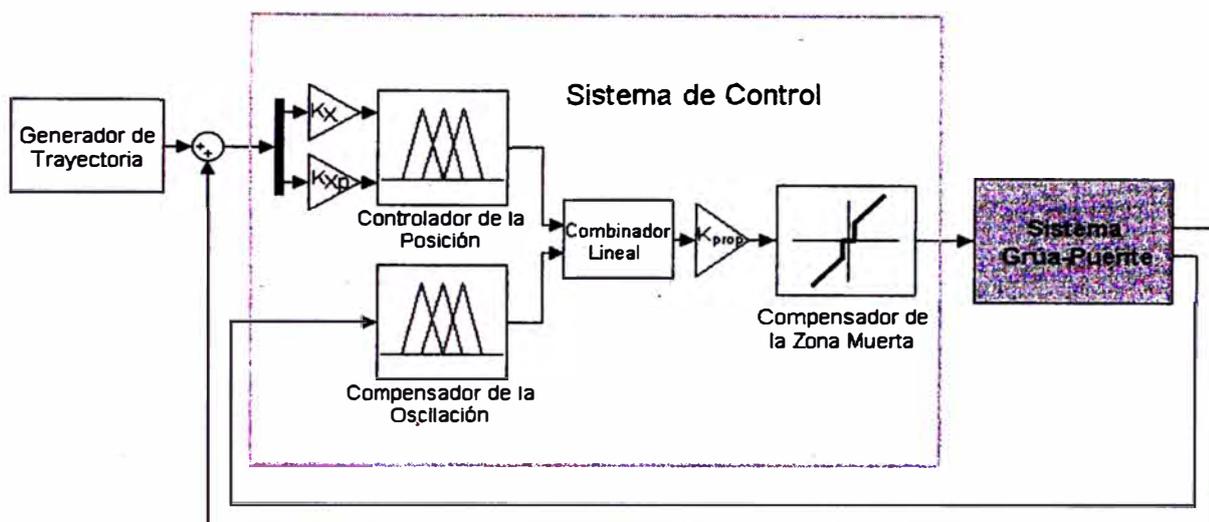


Figura 4.15: Diagrama de bloques del sistema de control en lazo cerrado.

4.3.1. Planta sin Fenómenos No-lineales

La primera simulación (figuras 4.16 y 4.17) que se presenta corresponde a la prueba sin incluir fenómenos no lineales en el modelo de la grúa-puente. Se observa que a los 5 segundos el sistema ya ha alcanzado la posición objetivo y la oscilación tiene una amplitud menor a ± 0.2 grados sexagesimales. En esta prueba se ha empleado una carga de 2.35 Kg y K_{prop} se toma igual a 1.4; éste valor se considerará como el valor normal de K_{prop} . Dentro de la tolerancia de oscilación, se puede afirmar que el sistema alcanza el estado estable (foco estable) en 5 segundos y que presenta un sobreimpulso menor al 2.5 %.

4.3.2. Planta con Fenómenos No-lineales Sin Compensar

En esta sección, se simulará la respuesta del sistema cuando la planta incluye los fenómenos de fricción de Coulomb y zona muerta en el motor-reductor, y el controlador no incorpora el compensador de zona muerta.

En las figuras 4.18 y 4.19 se muestra la respuesta simulada con una carga de 2.35 Kg y manteniendo el valor de K_{prop} en 1.4. Se observa que si bien el

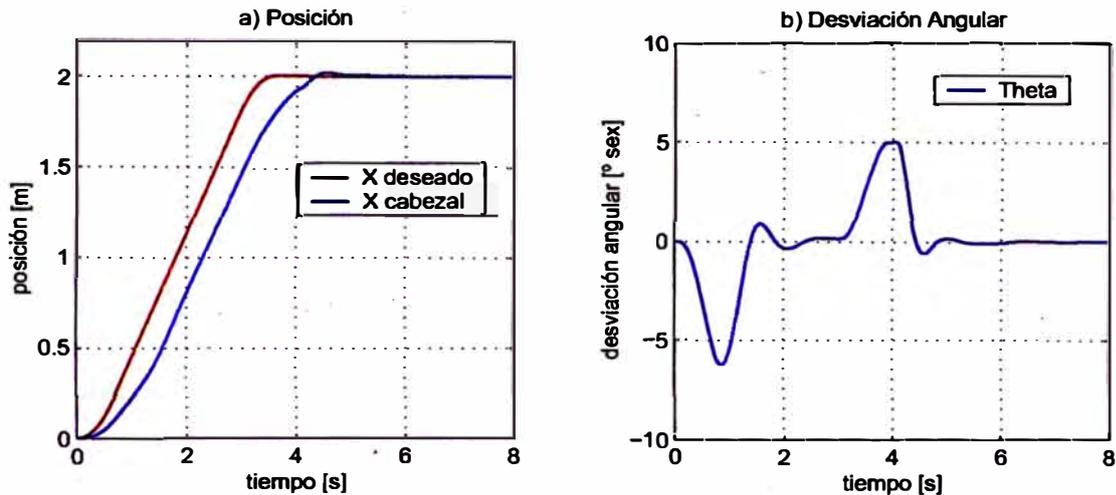


Figura 4.16: Planta sin fenómenos no lineales: Posición y desviación angular; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

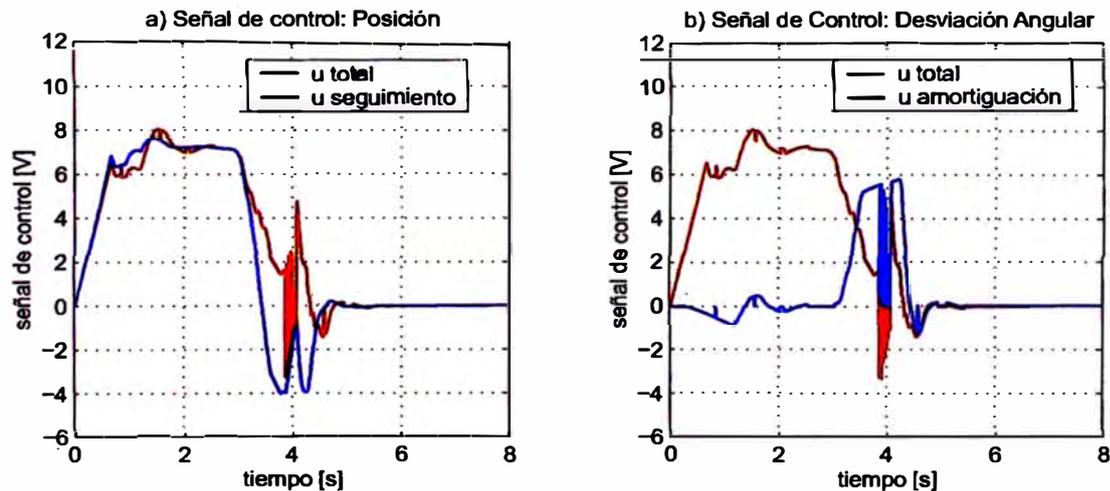


Figura 4.17: Planta sin fenómenos no lineales: Señales de control; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

controlador consigue amortiguar las oscilaciones, el error de posición del cabezal es completamente inadmisibles. En la figura 4.19 se ve como la señal de control busca llevar el sistema hacia el objetivo, pero no logra vencer la oposición de los efectos no lineales.

En las figuras 4.20 y 4.21, se observa la respuesta cuando K_{prop} se eleva a 2.5. En este caso se consigue mejorar la respuesta, pero es aún insatisfactoria. Se

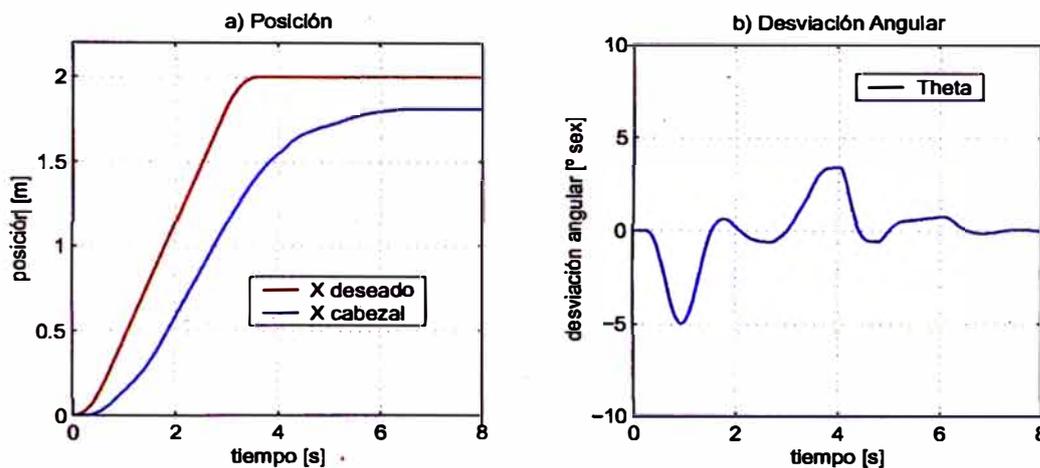


Figura 4.18: Planta con fenómenos no lineales sin compensar: Posición y desviación angular; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

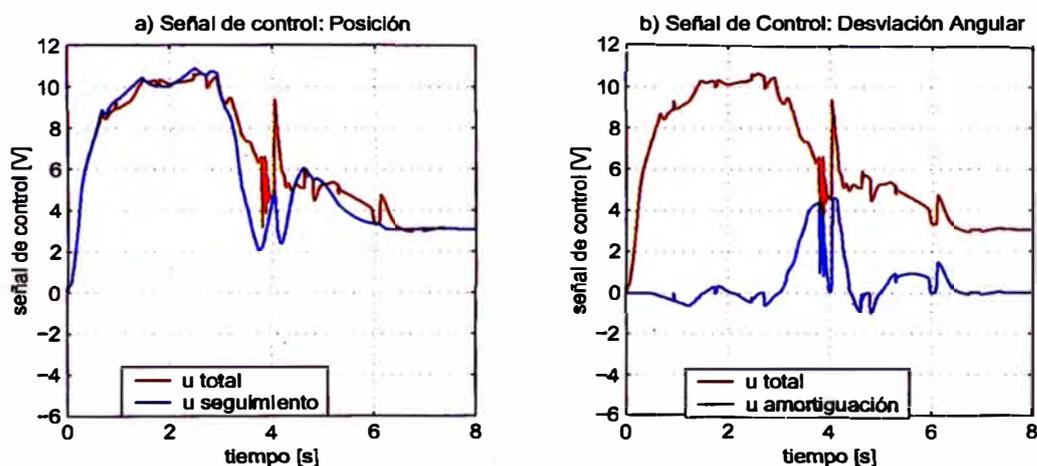


Figura 4.19: Planta con fenómenos no lineales sin compensar: Señales de control; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

muestra que hace falta el compensador de la zona muerta, o bien un integrador. Sin embargo un integrador da lugar a una acumulación considerable de error integral que es difícil de compensar. Esto se ha probado en el transcurso de la presente investigación sin obtener buenos resultados.

Se prueba la respuesta cambiando la masa de la carga a 1.35 Kg y con el mismo $K_{prop} = 2.5$. En este caso, al ser menor el efecto de la fricción, el sistema se desempeña mejor; sin embargo, aún no tiene una respuesta suficientemente

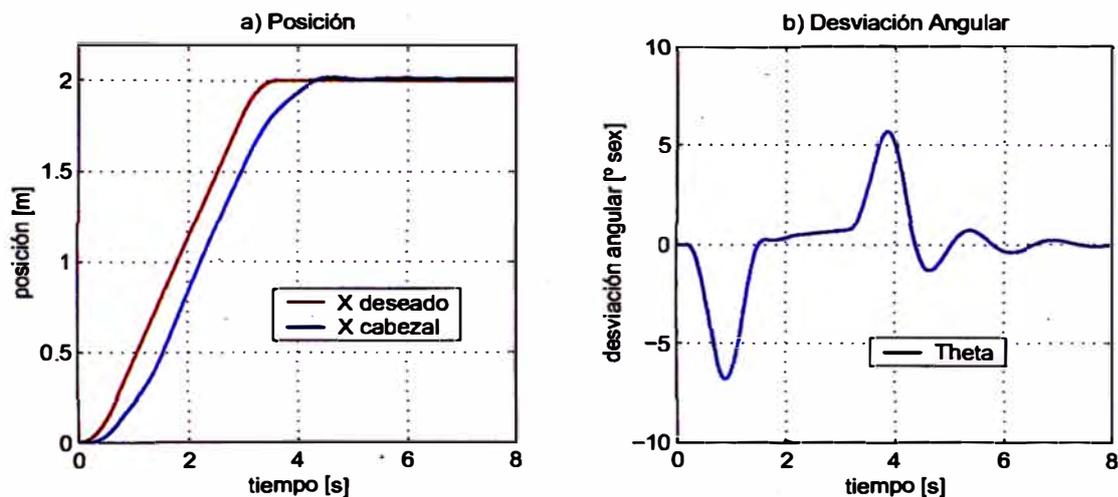


Figura 4.20: Planta con fenómenos no lineales sin compensar: Posición y desviación angular; $K_{prop} = 2.5$, $M_L = 2.35Kg$.

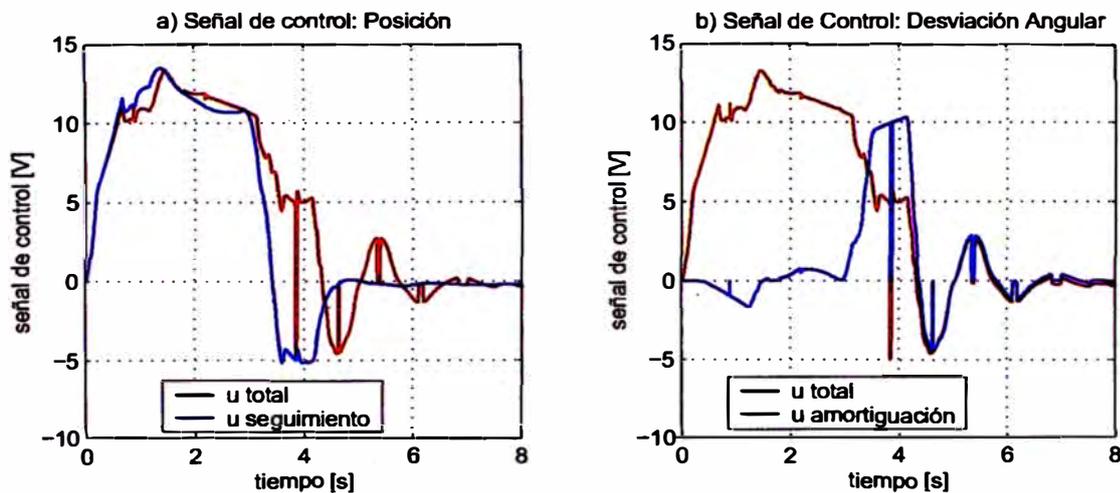


Figura 4.21: Planta con fenómenos no lineales sin compensar: Señales de control; $K_{prop} = 2.5$, $M_L = 2.35Kg$.

buena, aunque sería admisible en una operación de menor precisión (ver figuras 4.22 y 4.23). A los 6 segundos la oscilación residual es menor a 0.3 grados, pero el error del cabezal es de 15 mm.

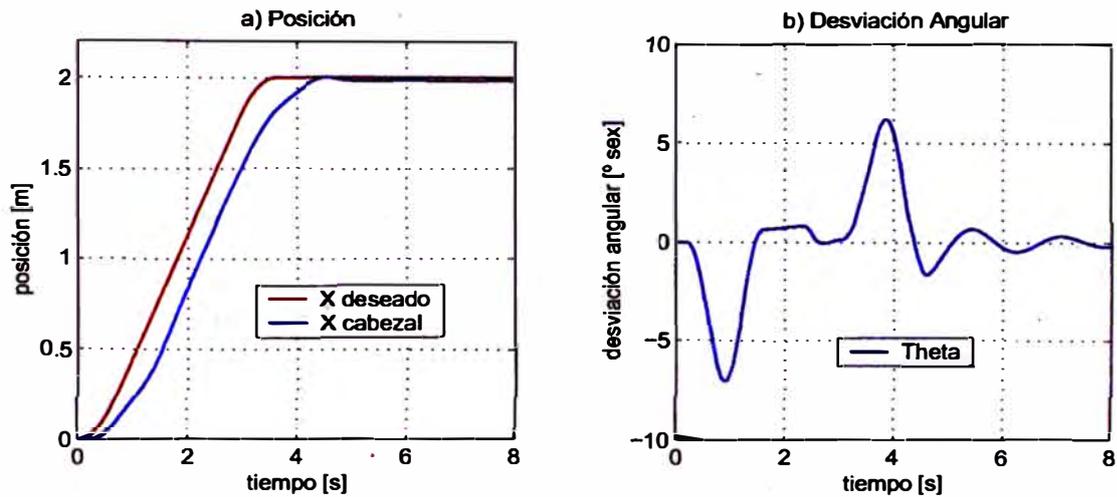


Figura 4.22: Planta con fenómenos no lineales sin compensar: Posición y desviación angular; $K_{prop} = 2.5$, $M_L = 1.35Kg$.

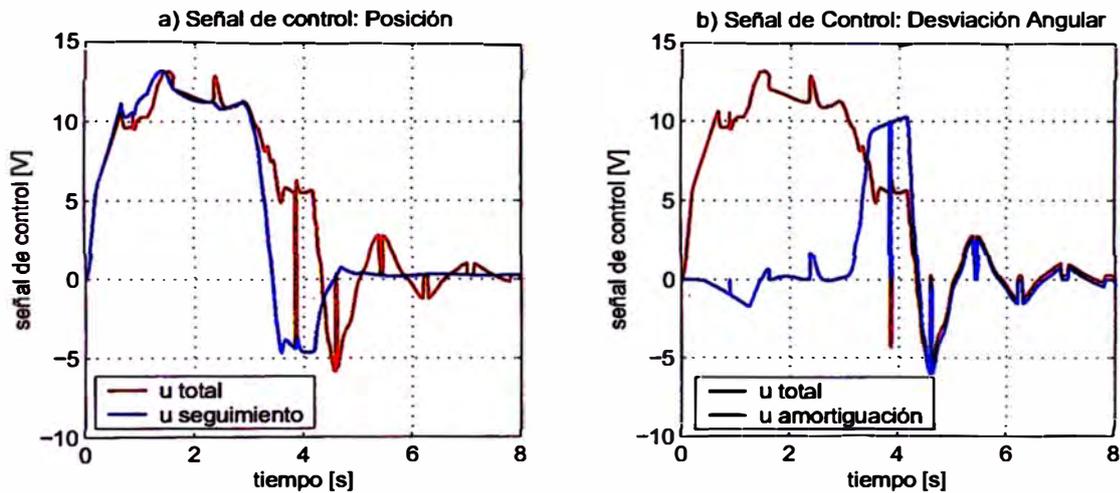


Figura 4.23: Planta con fenómenos no lineales sin compensar: Señales de control; $K_{prop} = 2.5$, $M_L = 1.35Kg$.

4.3.3. Planta con Fenómenos No-lineales Compensados

En esta sección se considerará el modelo de la planta incluyendo los efectos de fricción y zona muerta; el controlador incluirá al compensador de zona muerta.

Puede verse en las figuras 4.24 y 4.25 la respuesta de controlador con compensación de zona muerta para $K_{prop} = 1.4$, $M_L = 2.35Kg$. En este caso la respuesta del sistema sí es admisible y alcanza la estabilidad a los 5.5 s y presenta

un sobreimpulso de 2.5 %.

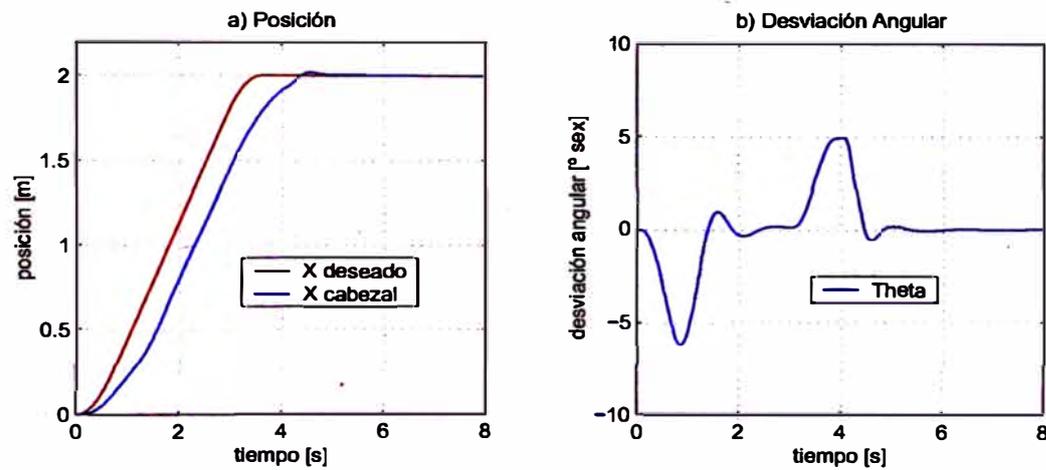


Figura 4.24: Planta con compensación de la zona muerta: Posición y desviación angular; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

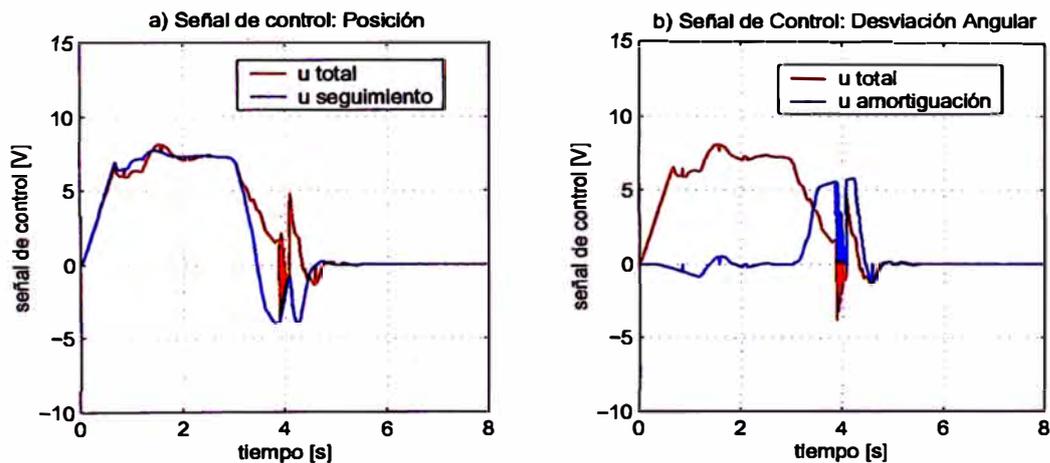


Figura 4.25: Planta con compensación de la zona muerta: Señales de control; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

En las figuras 4.26 y 4.27 se observa la respuesta del sistema controlado con la modificación en la masa de la carga, que ahora es de 1.35 Kg. En este caso el sistema queda ligeramente por debajo de la referencia en un principio y termina estabilizándose en 6.8 s consiguiendo el objetivo en forma satisfactoria. Se observa que la compensación angular es efectiva.

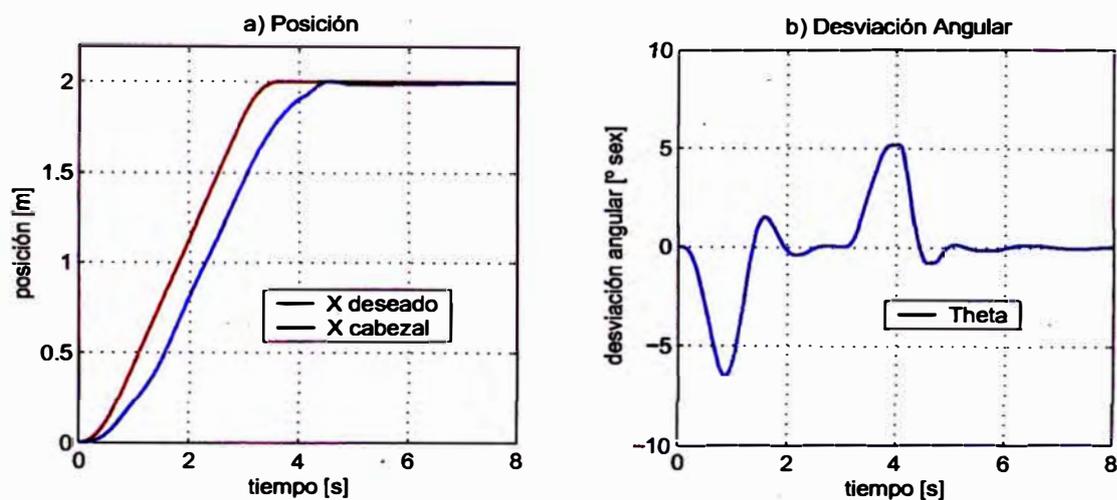


Figura 4.26: Planta con compensación de la zona muerta: Posición y desviación angular; $K_{prop} = 1.4$, $M_L = 1.35Kg$.

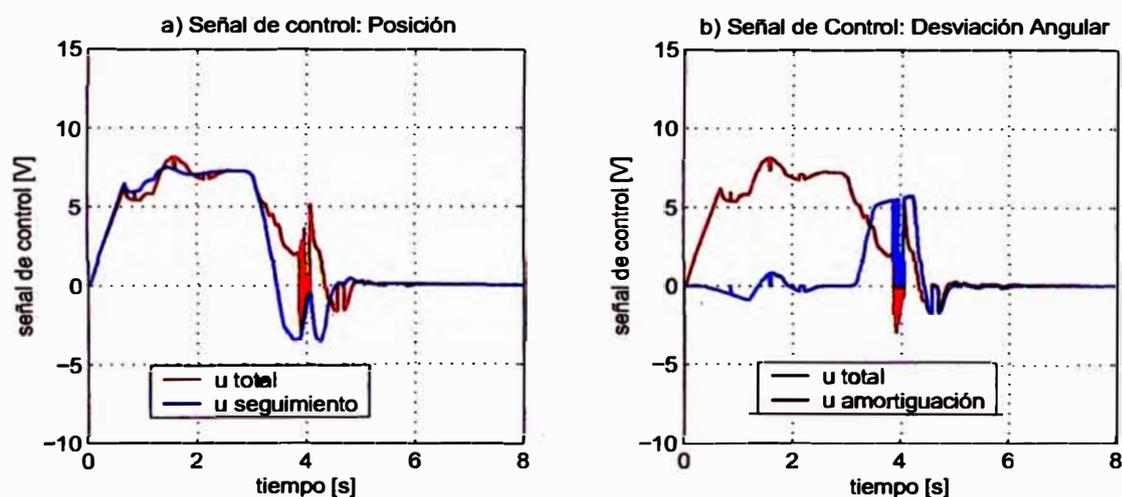


Figura 4.27: Planta con compensación de la zona muerta: Señales de control; $K_{prop} = 1.4$, $M_L = 1.35Kg$.

4.4. Simulación del Controlador Discreto

Previamente se presentaron las simulaciones del sistema controlado en tiempo continuo. En adelante, se presentan las simulaciones del sistema en tiempo discreto. En primer lugar se explican las consideraciones tomadas en la discretización de los sensores y en la elección del período de muestreo. Luego, se muestran los resultados al considerar la planta sin fenómenos no lineales y luego incorpo-

rando los fenómenos no lineales en el modelo; al incluir los fenómenos no lineales, se simulará el sistema con y sin la compensación de la zona muerta del motor-reductor.

La simulación del controlador discreto consigna el uso de de dos períodos de muestreo, uno para la planta y otro para el controlador. El período de muestreo de la planta es de 4 ms y el del controlador se escoge finalmente como 30Hz, tras evaluar las simulaciones mostradas en este capítulo 4.4.2.

Como referencia, en el apéndice C.2 se anexan los programas en MATLAB usados para estas simulaciones.

4.4.1. Discretización de los Sensores

Los sensores empleados son de naturaleza discreta. Los encoders permiten contar un número entero de fracciones de revolución, de este modo, al obtener directamente la posición a través del conteo de pulsos, se computa un valor que es proporcional al número de pulsos de la señal en cuadratura. La doble señal en cuadratura de los enconders empleados permite definir el desplazamiento angular no sólo en magnitud, sino también en sentido.

En Matlab, se ha empleado la siguiente secuencia para discretizar el ángulo medido por los enconders en función de la resolución:

```
npulse_encoder1=512;%numero de pulsos del encoder para la posicion
qep=1;%multiplicador del QEP
npulse_encoder2=1024;%numero de pulsos para el ángulo de la varilla
ls7184=4;%multiplicador del timer 2
dpolea=0.0757; %diámetro de la polea en m.
m2pulse=npulse_encoder1*qep*5.9/(pi*dpolea); %metros en pulsos.
gsex2pulse=npulse_encoder2*ls7184/360; %Convierte grados sexagesimales
%en pulsos.
```

```
%...
pp=floor(m2pulse*x(i,1)); %pp=posición en pulsos.
ap=floor(gsex2pulse*x(i,4)*57.2958); %ap=angulo en pulsos.
```

Para simular el efecto de la discretización o *cuantización* de los sensores, se emplea el listado anterior. Al observar este listado puede verse que mediante el uso del comando “floor” se produce una pérdida de información. Este efecto es más crítico en el caso de la medición de la velocidad mediante la diferenciación numérica de la posición medida mediante pulsos por un encoder. Mientras mayor sea la frecuencia de muestreo, menor, en términos generales, será la diferencia entre un valor anterior y un valor presente para una variable cualquiera. Un sensor que discretice estas pequeñas diferencias, perderá información en forma inversamente proporcional con su resolución. En la sección siguiente se verá el efecto del período de muestreo en el controlador; se procederá a buscar un período de muestreo que sea lo suficientemente corto para un cálculo oportuno de la señal de control y lo suficientemente largo para poder obtener la velocidad de las variables medidas de forma aceptable.

En la figura 4.28 se muestra las señales de posición y desviación angular a discretizar. Para todas las discretizaciones se considerará la resolución del encoder correspondiente (en el caso de la posición se toma en cuenta el efecto de la reducción). En la figura mencionada se consideran los rangos de las variables y sus derivadas para que correspondan con los rangos que se presentan en las simulaciones y experimentos, de modo que el análisis sea útil para la investigación actual.

Al ver la figura 4.29 puede apreciarse que la discretización con período de muestreo de 100 Hz da lugar a una notable pérdida de información en la medición de las velocidades. Particularmente, la velocidad angular limita su resolución a $7.25^\circ/\text{s}$; mientras que la velocidad lineal se ve afectada severamente por el ruido de

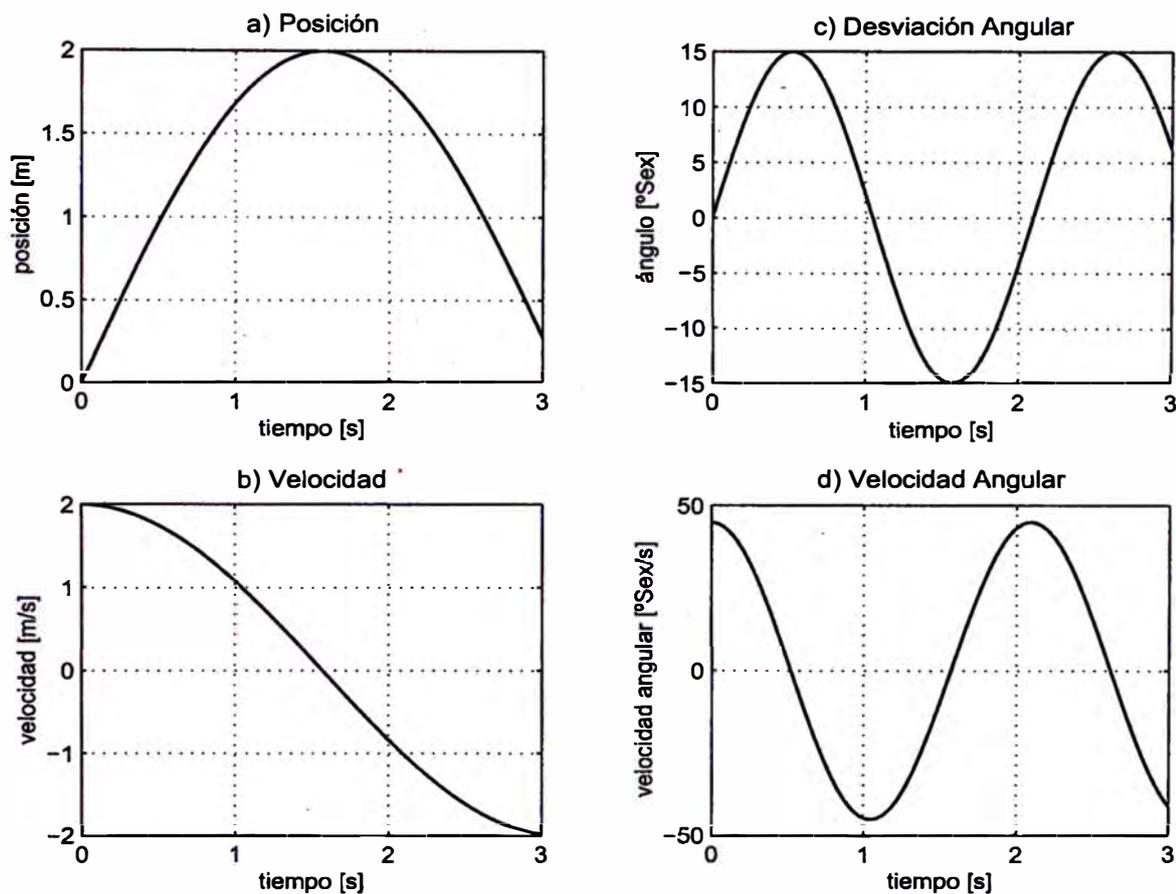


Figura 4.28: Señales continuas simuladas.

la cuantización. De este modo debe descartarse un período de muestreo superior o similar a 100 Hz, debido a su efecto sobre la medición de velocidad.

Se realiza una simulación con un período de muestreo de 30 Hz, que se aprecia en la figura 4.30, donde la resolución de la velocidad angular se puede considerar aceptable; con un rango de 37 valores posibles. La velocidad lineal presenta un ruido de cuantización bajo. En el caso de la medición de la posición, durante el desplazamiento a velocidad máxima, a 30 Hz se observan incrementos de hasta 18 mm por período de muestreo; si consideramos la velocidad de aproximación al objetivo, los intervalos están en el orden de los 3 mm. Lo anterior no deja de lado el hecho de que la resolución de posición es independiente del período de muestreo e igual a 0.0787 mm, mientras que la resolución de la desviación angular es de 0.0879 grados sexagesimales.

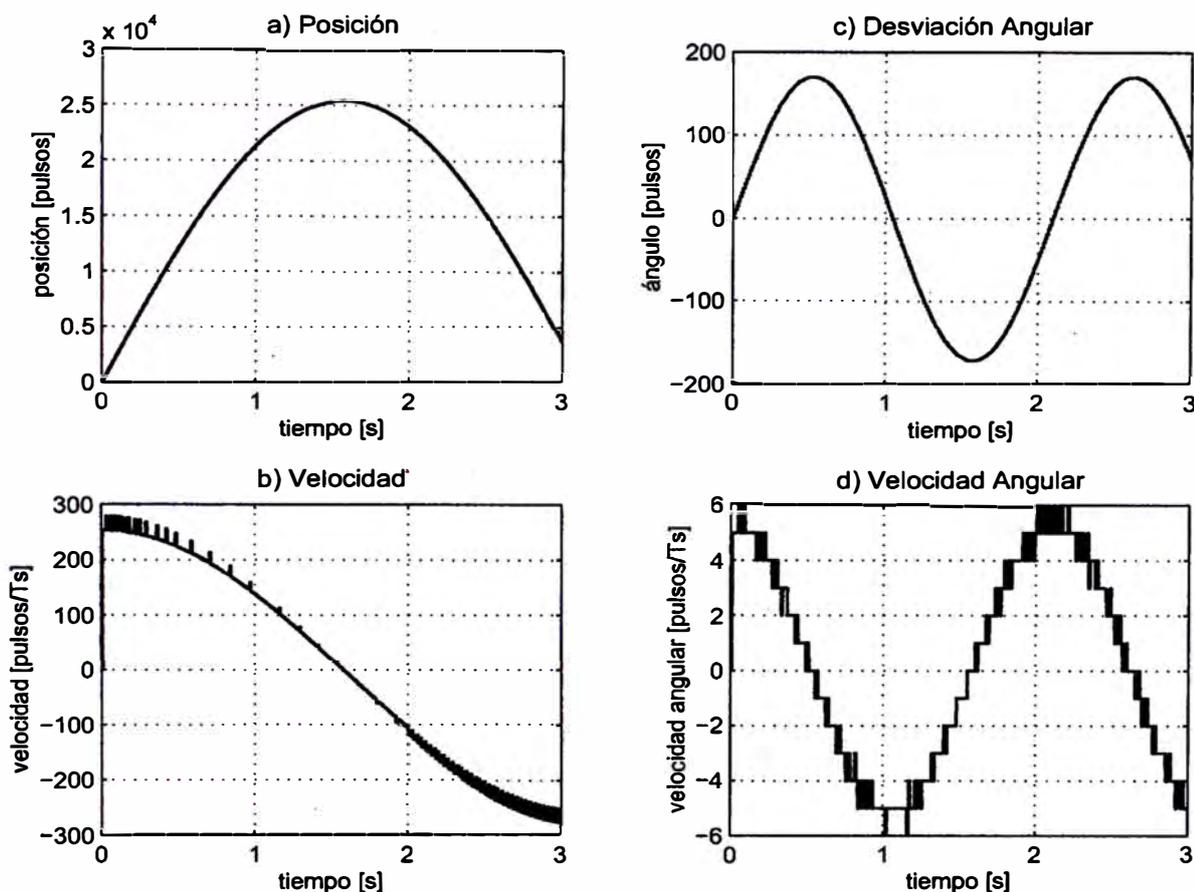


Figura 4.29: Señales discretizadas a 100 Hz.

En la figura 4.31 puede apreciarse que a 15 Hz, las señales se deterioran por el efecto de la baja frecuencia de muestreo, que afecta los intervalos de medición a velocidades altas y medias. Se verá en las simulaciones de la sección siguiente que a 15 Hz la controlabilidad del sistema se ve severamente afectada.

4.4.2. Influencia del Período de Muestreo

En esta sección se apreciará el efecto de la variación en el período de muestreo sobre el desempeño del controlador considerando la influencia de la discretización inherente a los encoders.

En las figuras 4.32 y 4.33 se muestra la respuesta simulada del sistema controlado en tiempo discreto con una frecuencia de muestreo de 60 Hz. En estas figuras se puede observar que se presenta un fuerte sobreimpulso y que

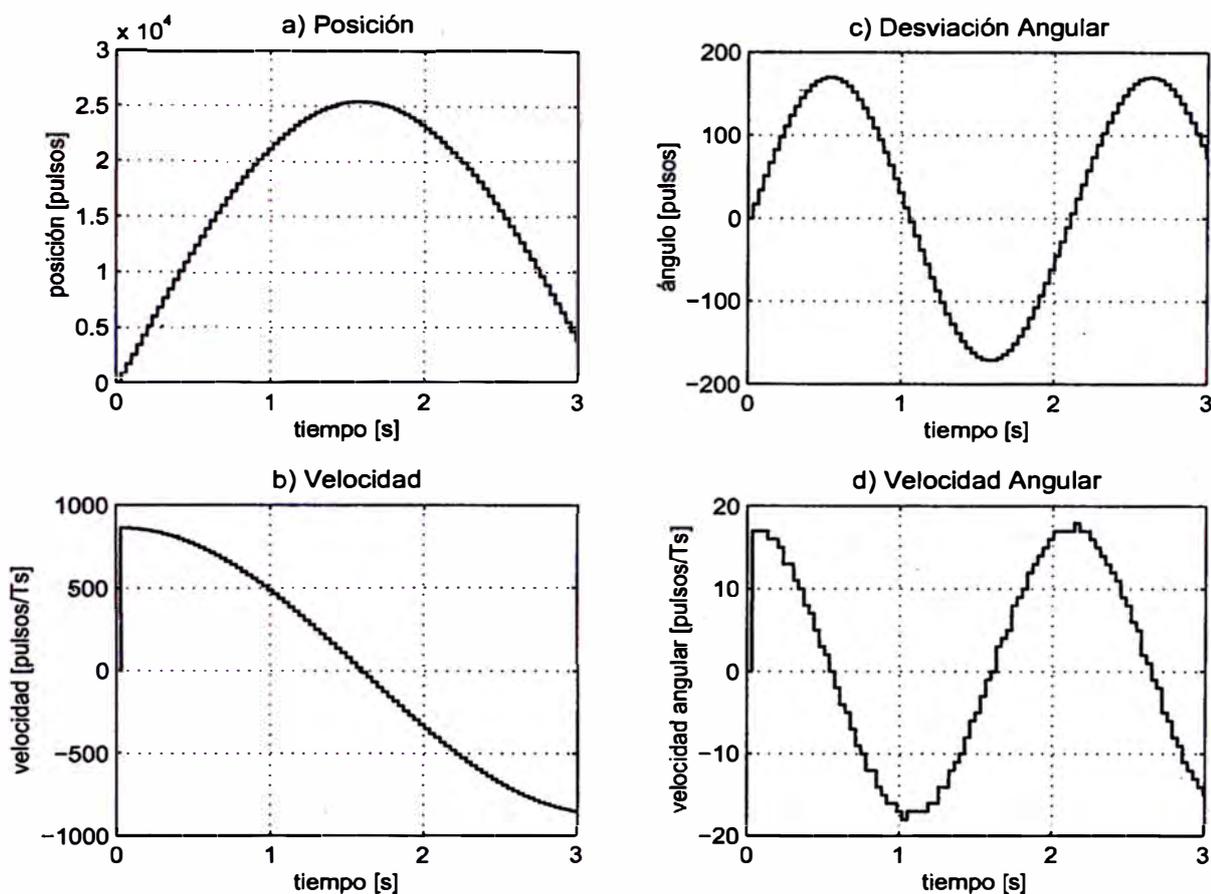


Figura 4.30: Señales discretizadas a 30 Hz.

la compensación de la oscilación pendular es deficiente. Esto se debe al ruido en la medición de la velocidad angular y lineal; éste ruido en la retroalimentación da lugar a que la señal de control también presente ruido. El ruido en la señal de control termina por deteriorar el comportamiento del sistema en lazo cerrado.

Si se escoge una frecuencia de muestreo de 30 Hz, como se aprecia en las figuras 4.34 y 4.35, se puede observar una buena respuesta del sistema en lazo cerrado. En este caso, el efecto del ruido en la medición de velocidad es menor que cuando el período de muestreo es de 60 Hz y no afecta severamente al controlador. Puede apreciarse la diferencia entre las señales de control a 60 Hz y a 30 Hz; la señal de control a 30 Hz es mucho más limpia.

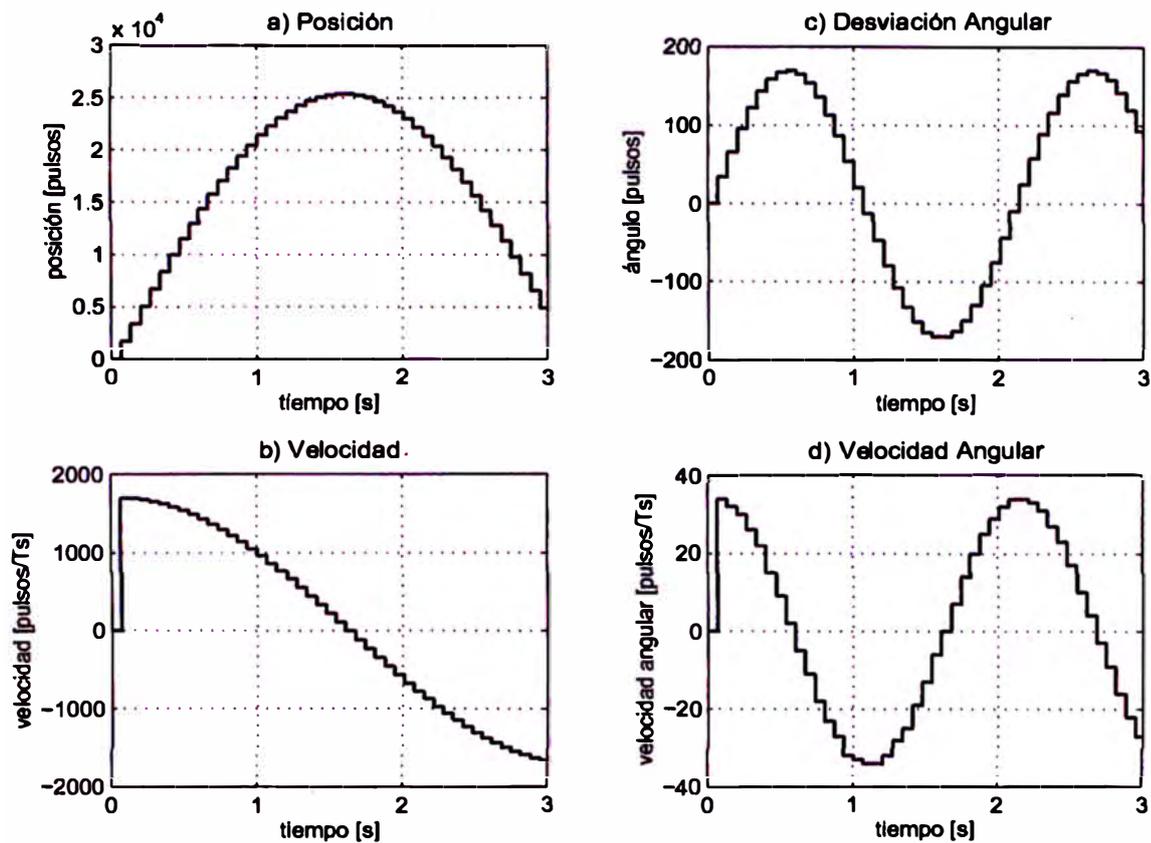


Figura 4.31: Señales discretizadas a 15 Hz.

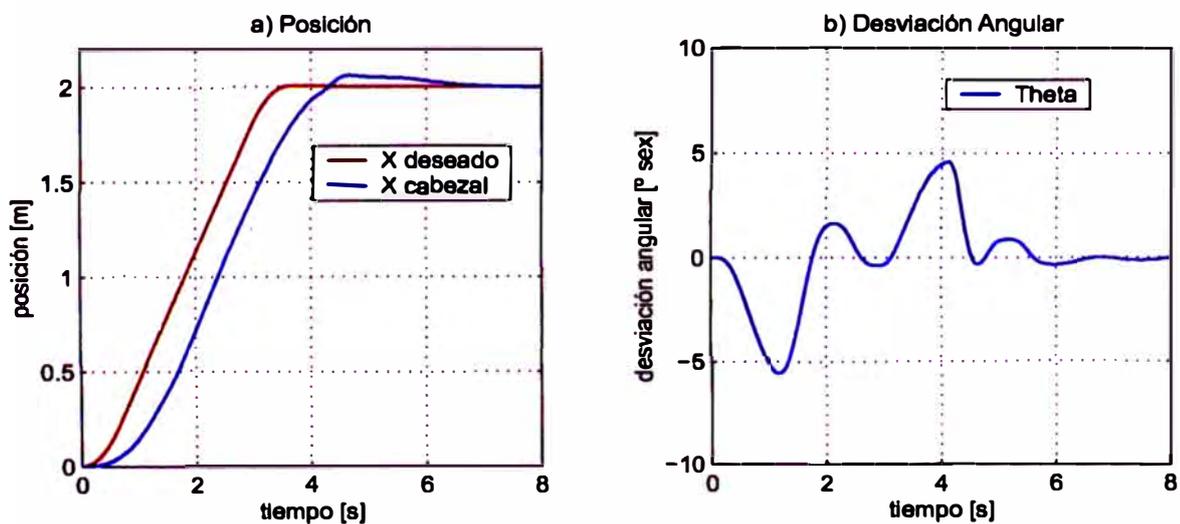


Figura 4.32: Posición y desviación angular del sistema controlado en tiempo discreto: frecuencia de muestreo de 60 Hz.

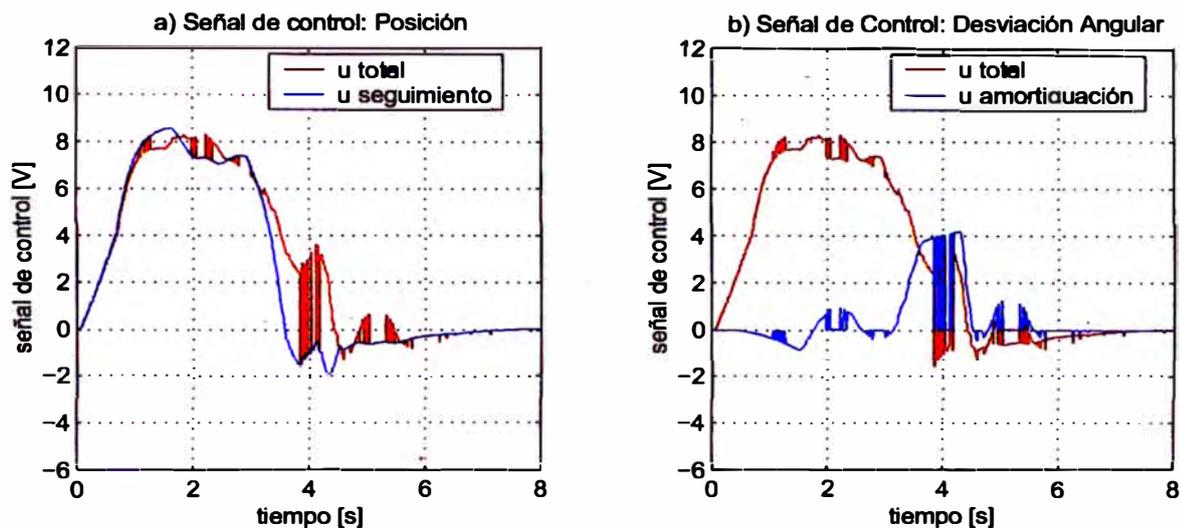


Figura 4.33: Componentes de la señal de control en tiempo discreto: frecuencia de muestreo de 60 Hz.

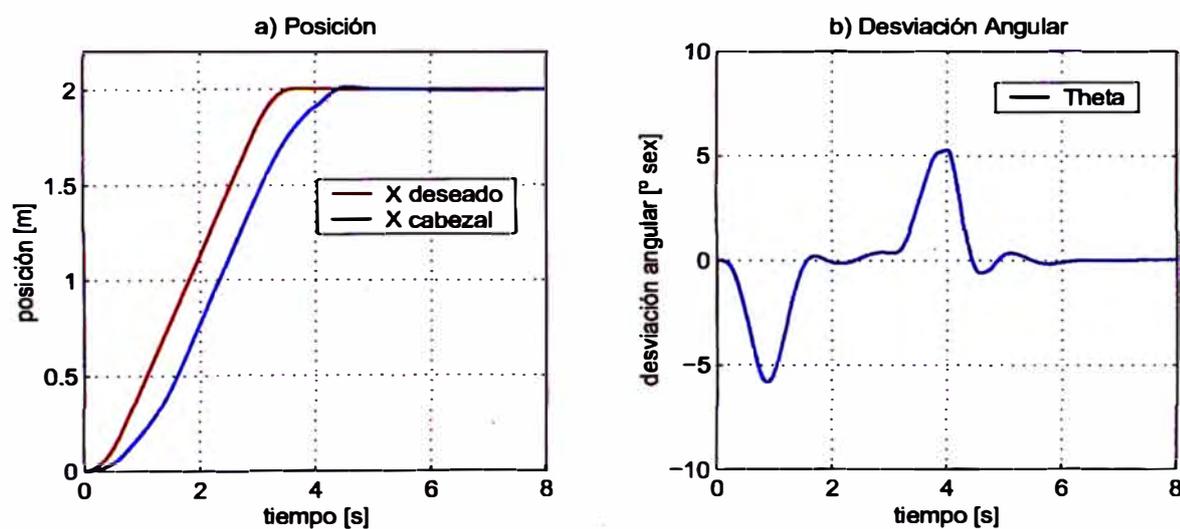


Figura 4.34: Posición y desviación angular del sistema controlado en tiempo discreto: frecuencia de muestreo de 30 Hz.

En las figuras 4.36 y 4.37 se puede observar la respuesta simulada cuando se toma como frecuencia de muestreo el valor de 15 Hz. La respuesta es bastante deficiente, debido en este caso a la pérdida o retardo en la información por causa de la baja frecuencia de muestreo.

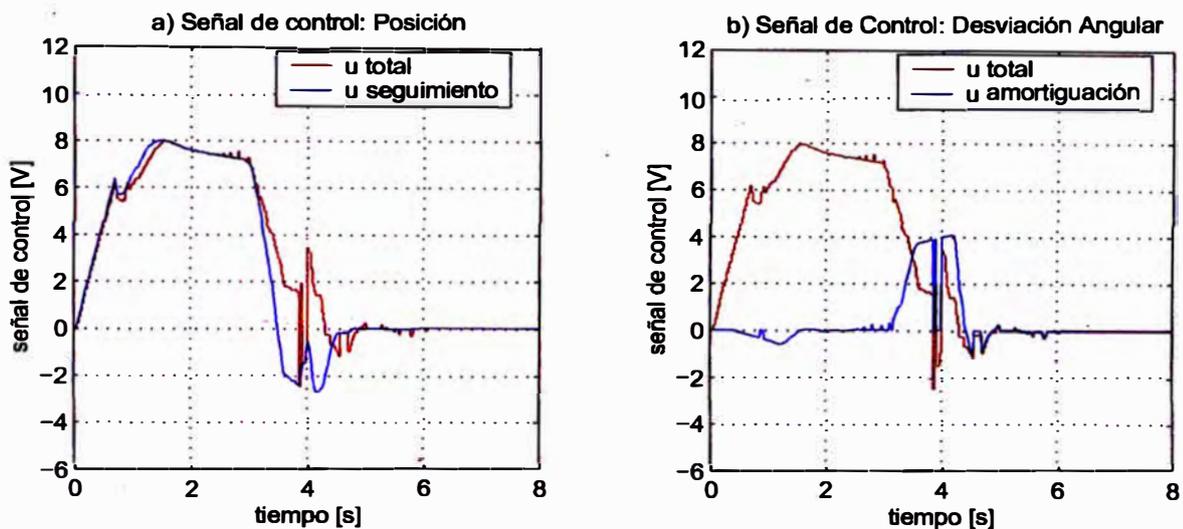


Figura 4.35: Componentes de la señal de control en tiempo discreto: frecuencia de muestreo de 30 Hz.

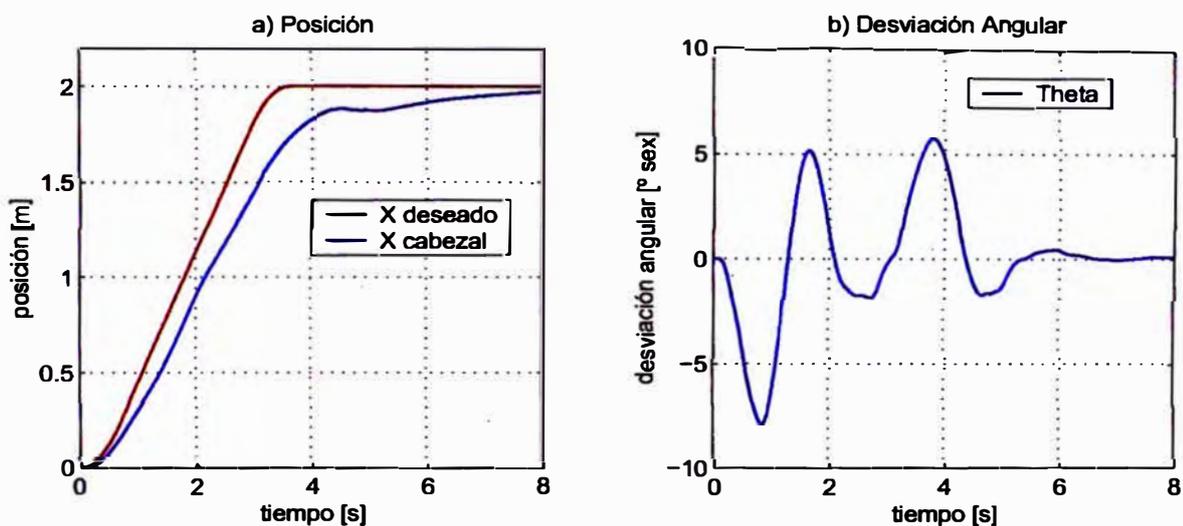


Figura 4.36: Posición y desviación angular del sistema controlado en tiempo discreto: frecuencia de muestreo de 15 Hz.

4.4.3. Planta sin Fenómenos No-lineales

Tras analizar los resultados de las simulaciones correspondientes a las dos secciones anteriores, se escoge una frecuencia de muestreo de 30 Hz.

Se procede ahora a simular la respuesta del sistema controlado cuando el modelo de la planta no incluye los efectos de la fricción y de la zona muerta; en

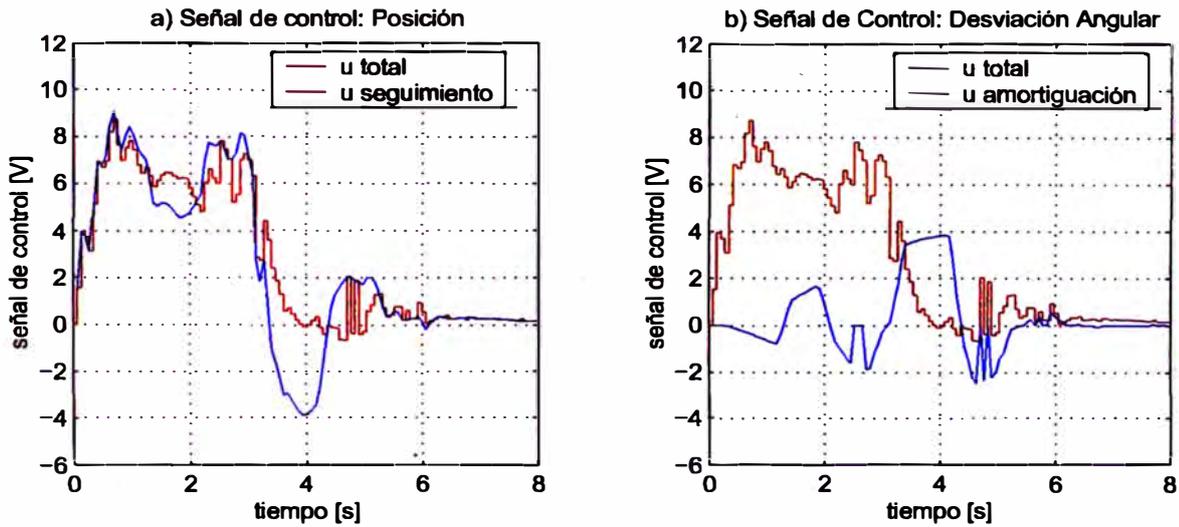


Figura 4.37: Componentes de la señal de control en tiempo discreto: frecuencia de muestreo de 15 Hz.

forma similar al análisis realizado en la sección 4.3.

Puede observarse en las figuras 4.38 y 4.39 que la respuesta del sistema es satisfactoria, y bastante similar a la respuesta en tiempo continuo. Sin embargo, cabe notar que no se presenta sobreimpulso y alcanza un estabilidad relativa a partir de los 6 segundos.

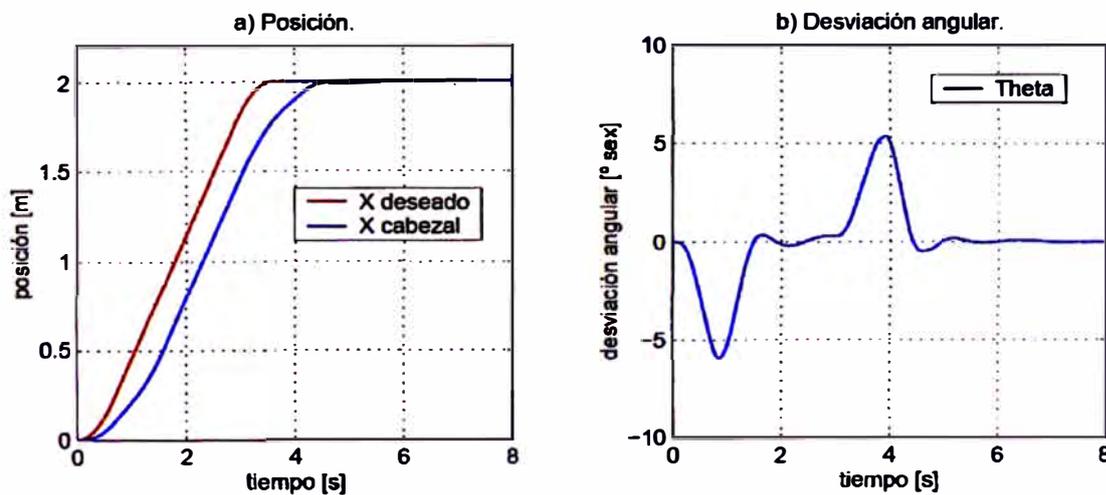


Figura 4.38: Planta sin fenómenos no lineales: Posición y desviación angular; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

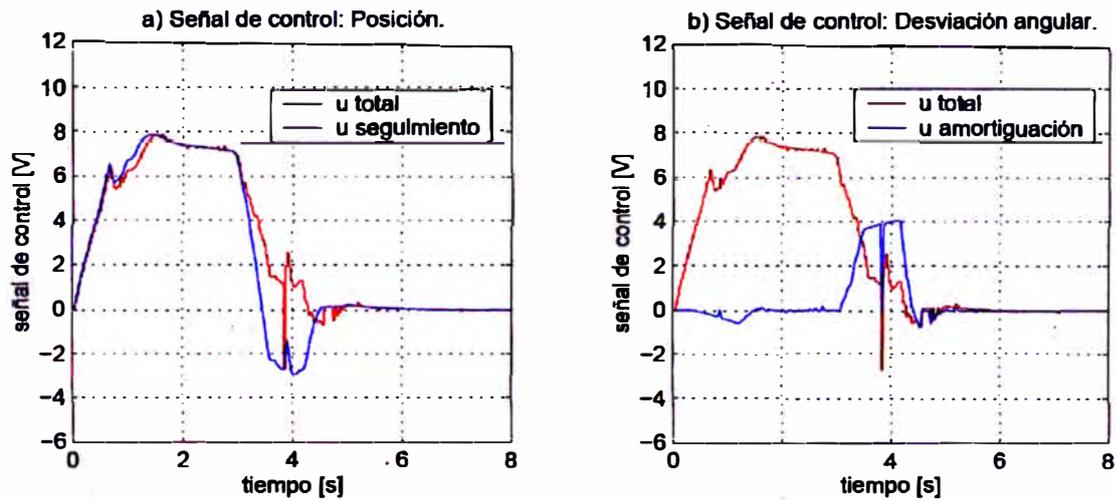


Figura 4.39: Planta sin fenómenos no lineales: Señales de control; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

4.4.4. Planta con Fenómenos No-lineales Sin Compensar

En este caso, tal como en tiempo continuo, la respuesta es muy deficiente si no se ajusta el valor de K_{prop} . Puede verse en las figuras 4.40 y 4.41 que el cabezal no alcanza el objetivo. Puede conseguirse una mejor respuesta ajustando K_{prop} , pero esta solución no es suficiente para garantizar una respuesta aceptable.

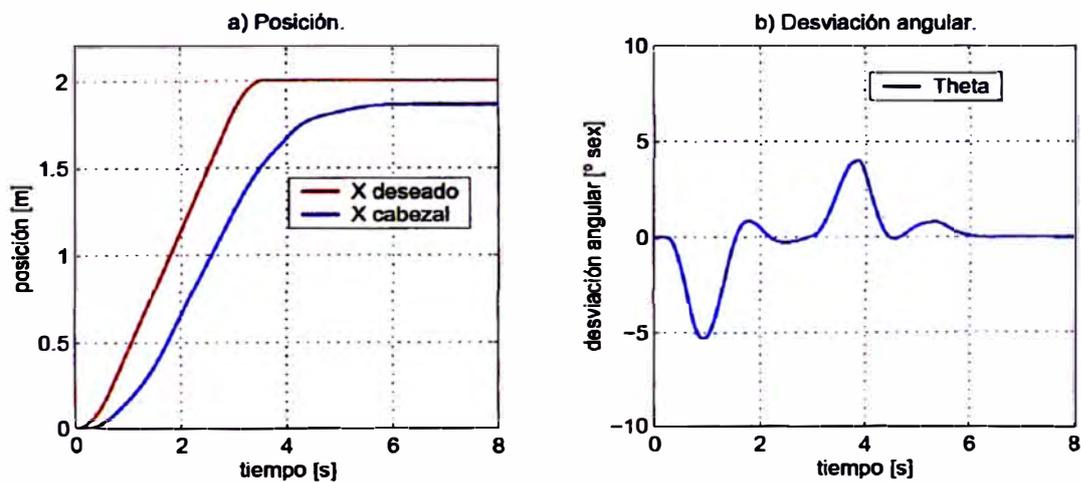


Figura 4.40: Planta con fenómenos no lineales sin compensar: Posición y desviación angular; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

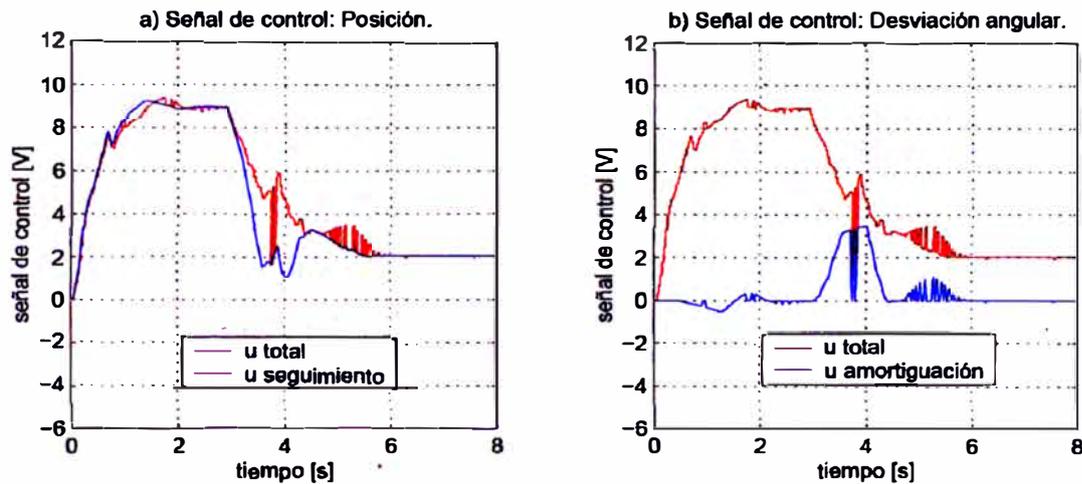


Figura 4.41: Planta con fenómenos no lineales sin compensar: Señales de control; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

4.4.5. Planta con Fenómenos No-lineales Compensados

Al compensar la zona muerta, el desempeño del sistema en tiempo discreto mejora notablemente. Se puede apreciar en las figuras 4.42 y 4.43 que a partir de los 6.8 segundos el sistema alcanza la estabilidad, aunque a los 6.0 segundos se disponía ya de un vector de estados aceptable.

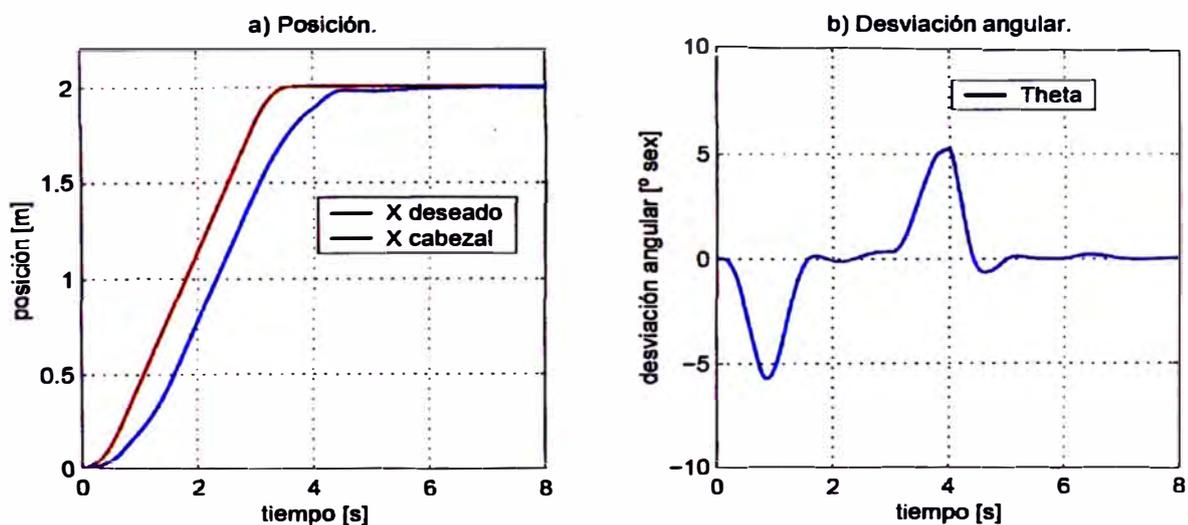


Figura 4.42: Planta con compensación de la zona muerta: Posición y desviación angular; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

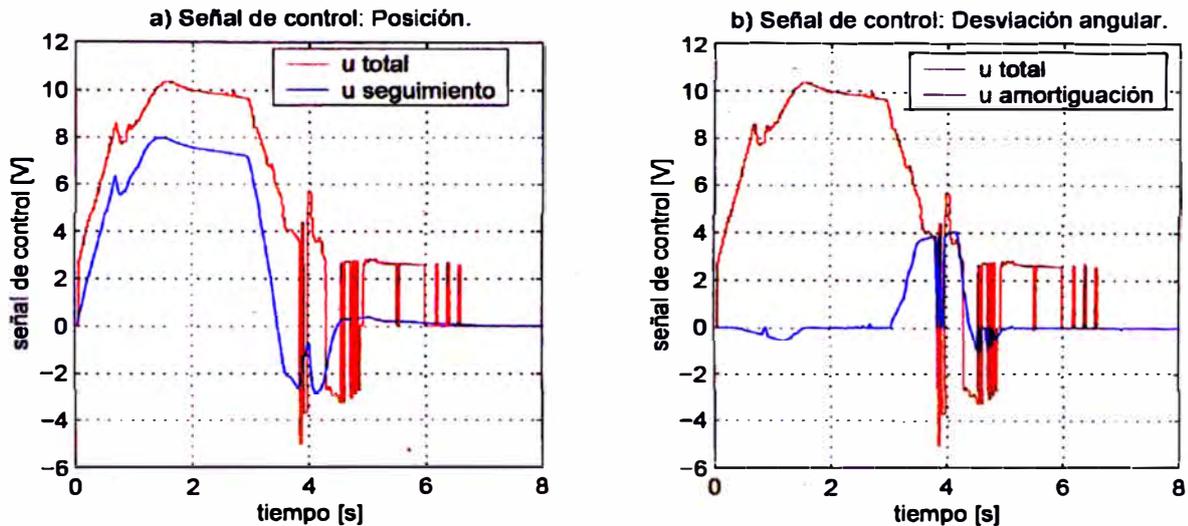


Figura 4.43: Planta con compensación de la zona muerta: Señales de control; $K_{prop} = 1.4$, $M_L = 2.35Kg$.

En las figuras 4.44 y 4.45 se muestra la prueba anterior, pero con 1.35 Kg de carga. Estas pruebas, en las que se varían los parámetros de la planta, nos permiten un análisis aproximado de la robustez del sistema de control. La respuesta del sistema es similar a la respuesta anterior con 2.35 Kg; se estabiliza en aproximadamente 6.8 segundos, no presenta sobreimpulso y la oscilación residual del péndulo es inferior a $\pm 0.25^\circ$.

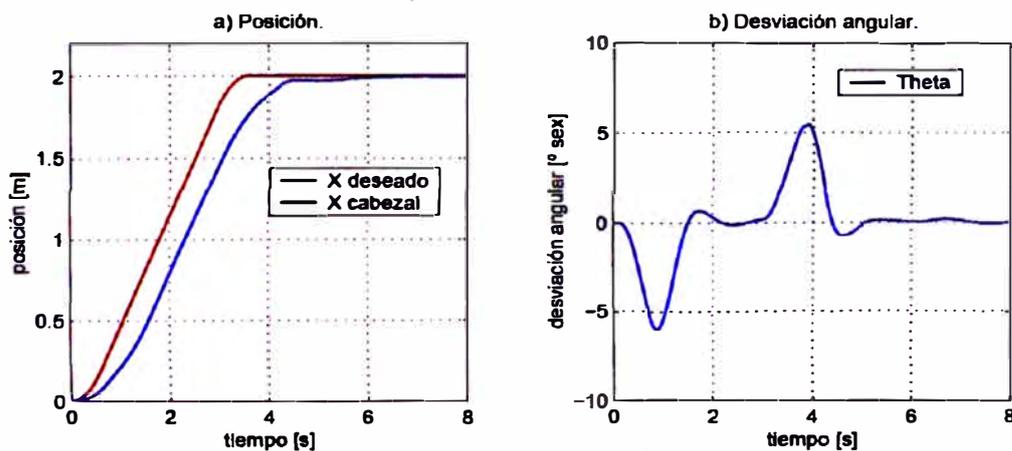


Figura 4.44: Planta con compensación de la zona muerta: Posición y desviación angular; $K_{prop} = 1.4$, $M_L = 1.35Kg$.

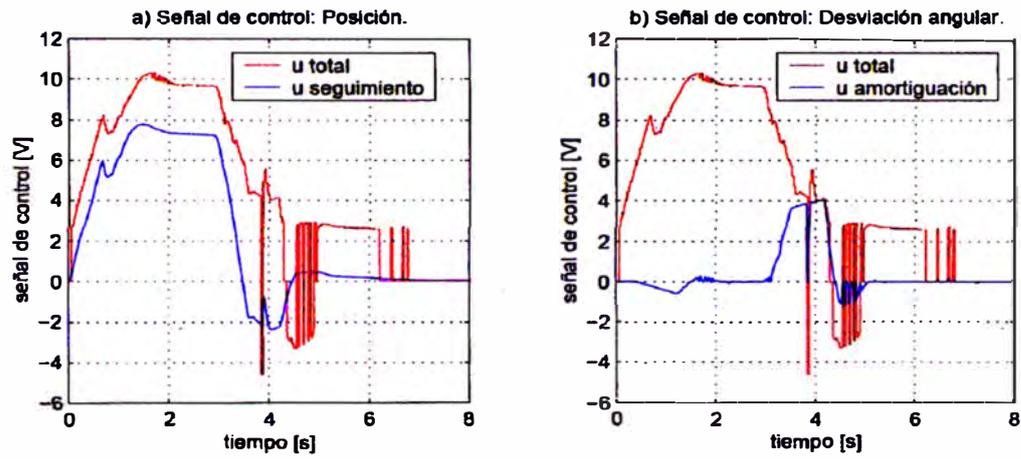


Figura 4.45: Planta con compensación de la zona muerta: Señales de control; $K_{prop} = 1.4$, $M_L = 1.35Kg$.

CAPÍTULO 5

RESULTADOS EXPERIMENTALES

Con el fin de verificar los resultados obtenidos en las simulaciones presentadas en el capítulo 4, se desarrollan experimentos similares con el prototipo construido para esta investigación y descrito en el apéndice A. Los parámetros experimentales son los mismos que los usados en la simulación.

Aunque en el capítulo 3 se buscó modelar los principales fenómenos no lineales, tales como la fricción y la zona muerta, estos modelos son aproximados y, particularmente en el caso de la fricción, tienen dinámicas complejas y variantes en el tiempo que no han sido consideradas. En cambio, se ha considerado una parte importante de su influencia sobre el sistema, de modo que, como se verá a continuación, los experimentos y las simulaciones mantienen una buena concordancia.

Durante esta investigación se desarrollaron diversos experimentos preliminares y numerosos experimentos para la prueba exhaustiva del sistema de control; este capítulo resume los resultados experimentales más representativos.

5.1. Resultados de las Pruebas Realizadas

5.1.1. Sin Compensación de los Fenómenos No Lineales

Tal como se vio en las simulaciones, la respuesta del controlador sin el compensador de la zona muerta es deficiente. Se observa que en el control de posición

del cabezal se tiene un error en estado estable de 60 mm aproximadamente, lo cual es inadmisibles. En este caso se probó con $K_{prop} = 1.4$, con valores mayores la respuesta mejora; pero si este valor es muy alto, se degenera. En cualquier caso, sin el compensador de la zona muerta no se obtiene un buen control. En el caso de la compensación de la oscilación sí se obtienen buenos resultados. Ver las figuras 5.1 y 5.2.

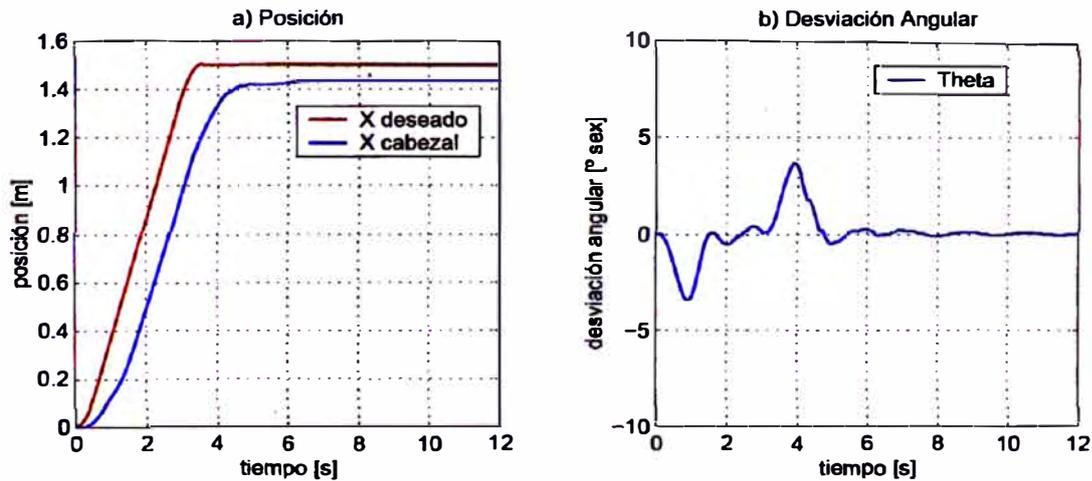


Figura 5.1: Respuesta experimental: Posición y desviación angular sin compensación de la zona muerta.

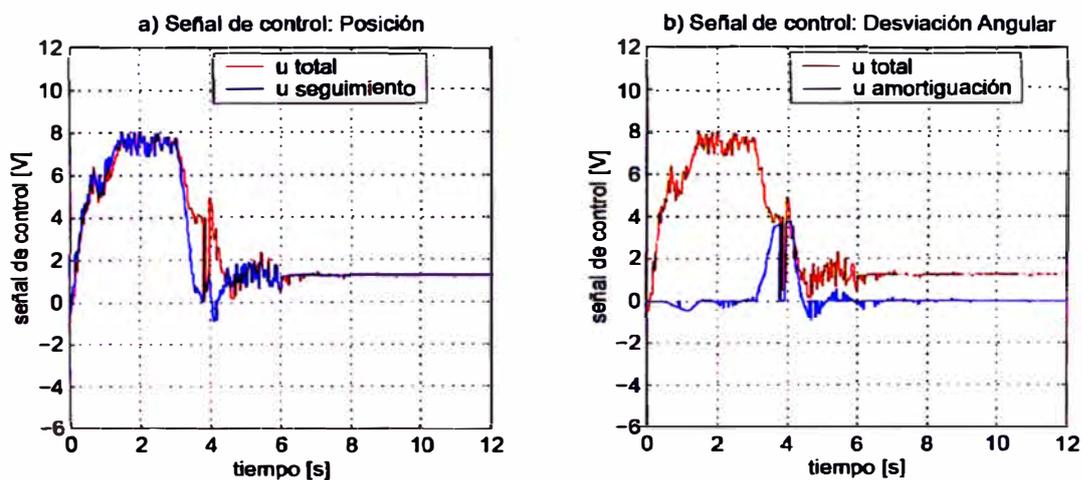


Figura 5.2: Respuesta experimental: Señal de control sin compensación de la zona muerta.

5.1.2. Con Compensación de los Fenómenos No Lineales

En las figuras 5.3 y 5.4 se muestra el contraste entre la respuesta del experimento anterior y la respuesta del experimento con el controlador que incluye al compensador de la zona muerta; los demás parámetros de control se conservan iguales. Puede apreciarse que la respuesta mejora notablemente y se vuelve satisfactoria, de modo que el error en estado estable de la posición del cabezal es menor a ± 0.2 mm. y la oscilación residual es menor a ± 0.2 grados. Se alcanza la estabilidad con error admisible a los 7 segundos y no presenta sobreimpulso.

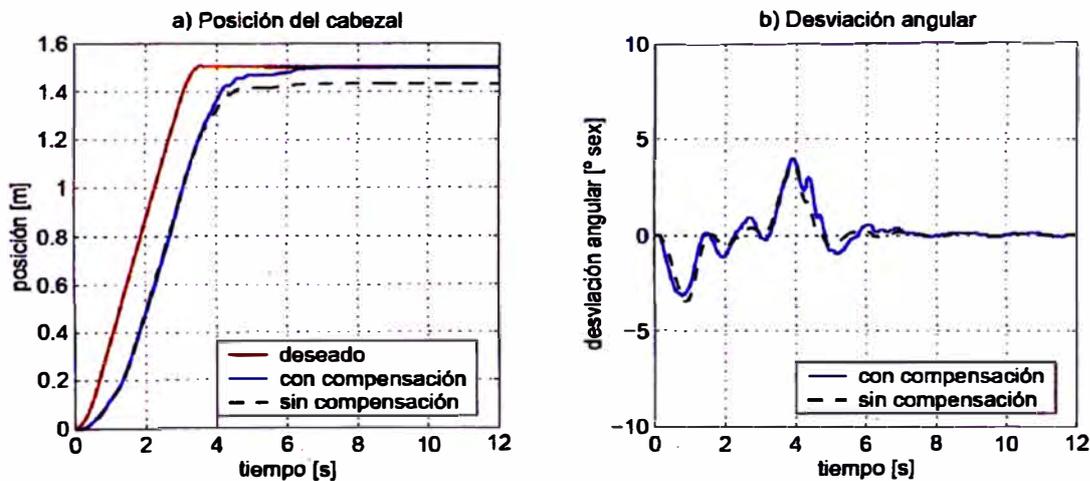


Figura 5.3: Respuesta experimental: Posición y desviación angular con y sin compensación de la zona muerta.

5.1.3. Variaciones en los Parámetros de la Planta

Se probará ahora la robustez del sistema de control ante la variación de la masa de la carga M_L . Se prueba el controlador cuando $M_L = 1.35$ kg (figuras 5.5 y 5.6) y cuando $M_L = 2.35$ kg (figuras 5.7 y 5.8); en ambos casos, la posición objetivo es 1.5 m.

Se observa que las respuestas son satisfactorias, la diferencia entre uno y otro radica en la energía de control empleada, que es mayor para la carga de 2.35 kg, y en el tiempo de elevación y establecimiento. Con una carga de 1.35

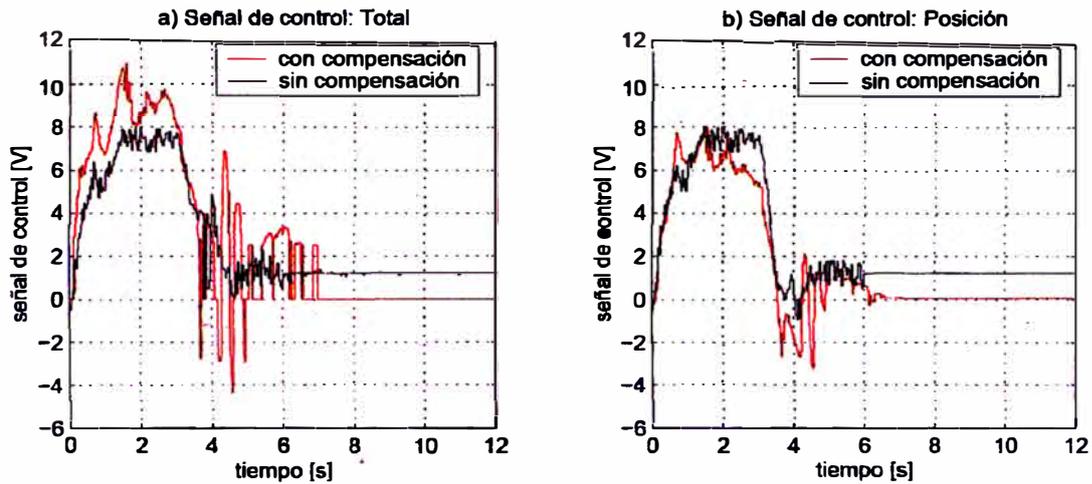


Figura 5.4: Respuesta experimental: Señal de control con y sin compensación de la zona muerta.

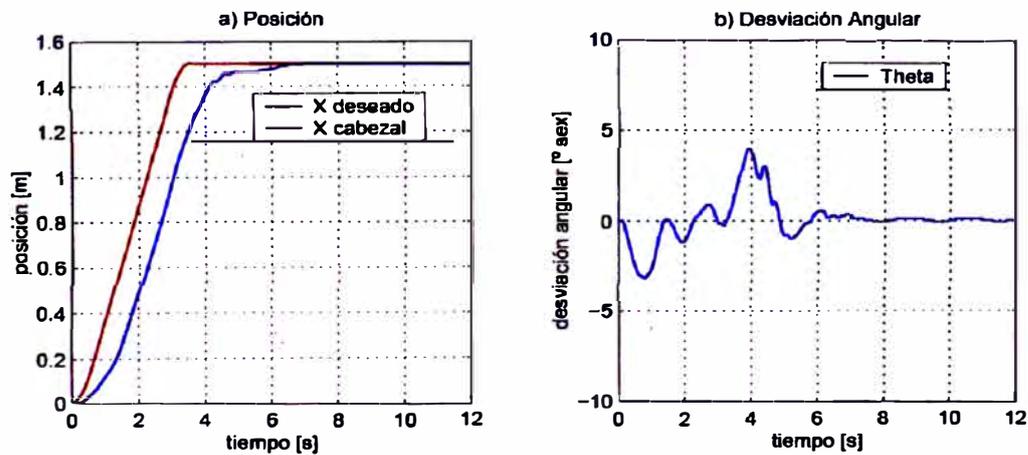


Figura 5.5: Respuesta experimental: Posición y desviación angular con 1.35 kg de carga.

kg el sistema alcanza la estabilidad en 7 s, mientras que con 2.35 kg toma 8 s. En todos los casos, el error de posición del cabezal en estado estable es inferior a 4 mm y la oscilación residual menor a ± 0.2 grados. En ningún caso se observa sobreimpulsos.

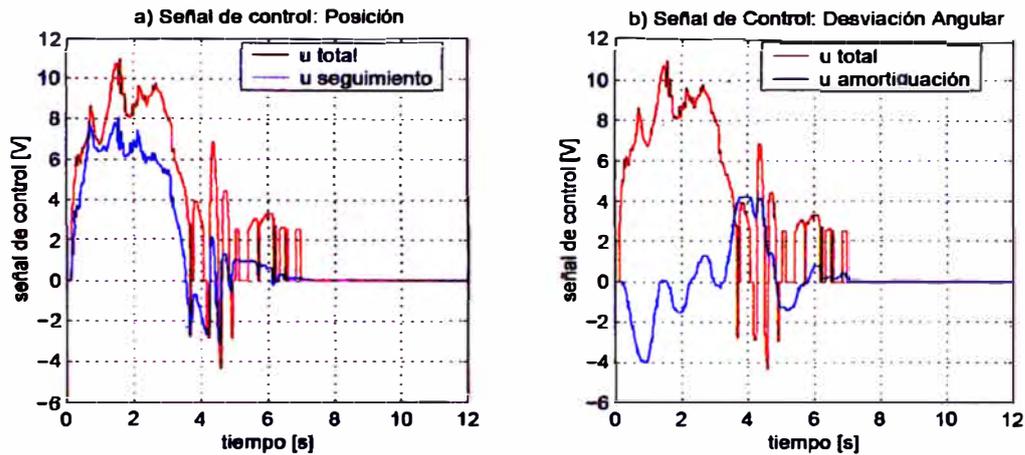


Figura 5.6: Respuesta experimental: Señal de control con 1.35 kg de carga.

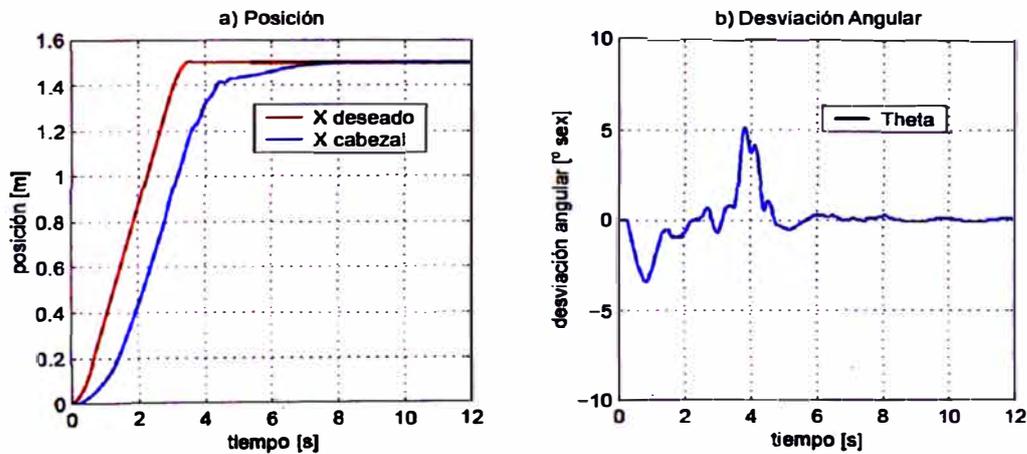


Figura 5.7: Respuesta experimental: Posición y desviación angular con 2.35 kg de carga.

5.1.4. Variaciones en los Parámetros del Controlador

Bajo este título, se contempla la modificación de dos parámetros importantes del controlador propuesto: la ganancia principal, K_{prop} , y el tiempo de traslado del generador de trayectoria, T_F . Si bien es cierto, que los dos sistemas de inferencia difusa que componen el controlador tienen varias decenas de parámetros, éstos parámetros sirven para definir la estructura del controlador; la cual se desea mantener. En cambio, los parámetros a modificar están en relación más directa con la aplicación del controlador a situaciones particulares. A priori se puede con-

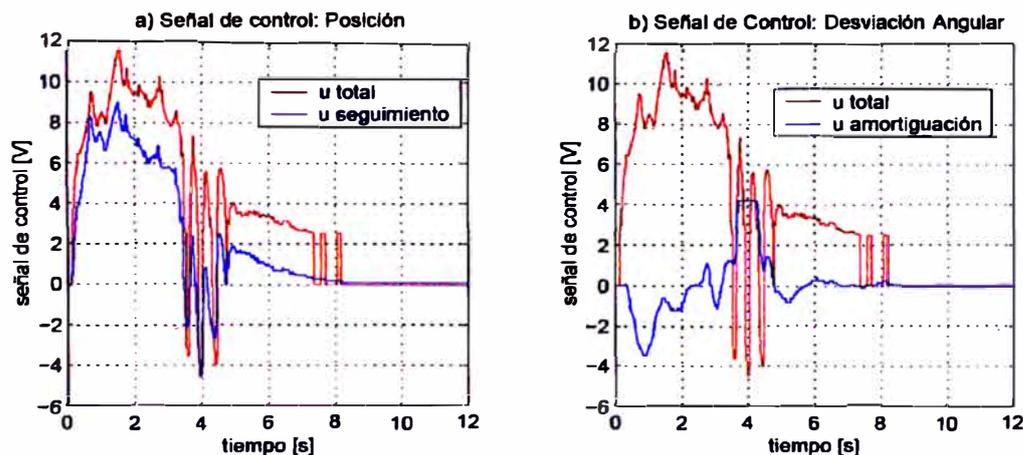


Figura 5.8: Respuesta experimental: Señal de control con 2.35 Kg de carga.

siderar que K_{prop} está relacionado con la velocidad de respuesta y la sensibilidad ante las perturbaciones; mientras que T_F , al ser la referencia para determinar el tiempo de traslado, podría modificarse en función de la distancia al objetivo.

En una serie preliminar de experimentos, considerando parámetros típicos de operación, se determinó que un valor apropiado para la ganancia proporcional (K_{prop}) es 1.4. Así también, de acuerdo con la literatura, se considera que el tiempo de traslado estándar corresponde a dos oscilaciones pendulares completas de la carga suspendida, [5].

Para experimentar con la modificación de la ganancia del controlador, ésta se incrementará en 50%. En las Figuras 5.9 y 5.10 se muestran las gráficas generadas durante la prueba con $K_{prop} = 1.4$; mientras que en las Figuras 5.11 y 5.12, K_{prop} toma el valor de 2.1.

Las principales diferencias se encuentran en el sobreimpulso y en el tiempo de levantamiento. Con un valor de K_{prop} más alto puede apreciarse un tiempo de elevación menor, aunque con la presencia de un sobreimpulso. El tiempo de establecimiento es prácticamente el mismo; sin embargo, puede observarse que la amplitud de la oscilación remanente crece ligeramente cuando el valor de K_{prop} es mayor ($K_{prop} = 2.1$). El incremento en la potencia suministrada al actuador

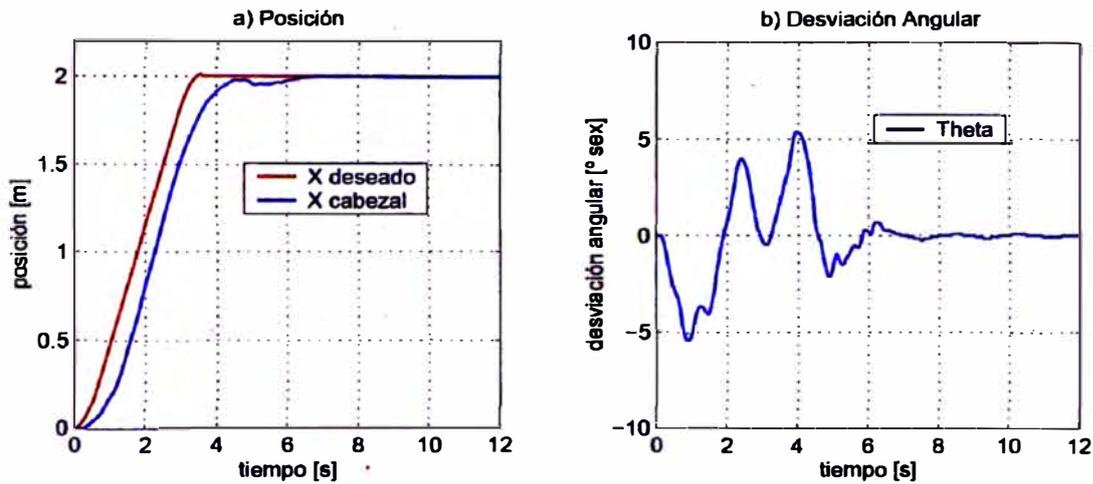


Figura 5.9: Respuesta experimental: Posición y desviación angular con $K_{prop} = 1.4$.

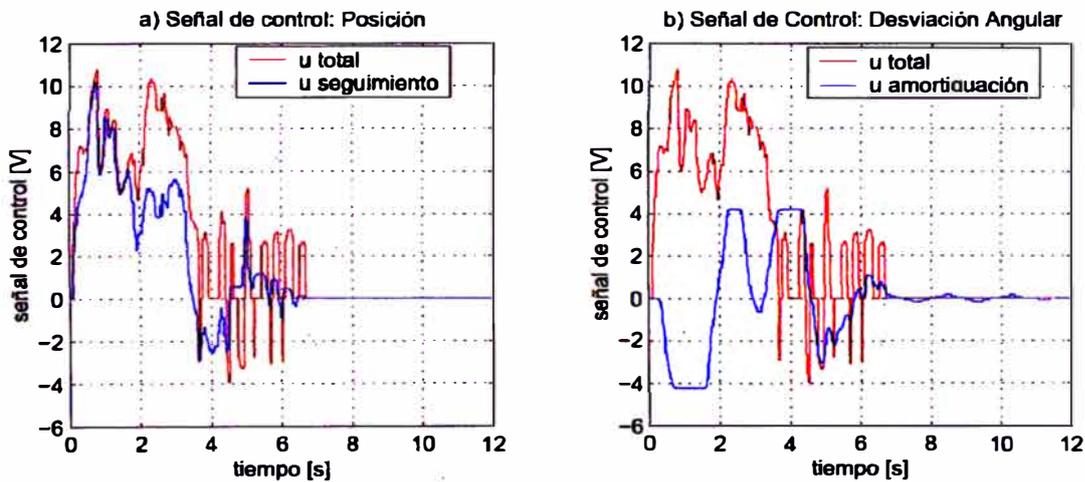


Figura 5.10: Respuesta experimental: Señal de control con $K_{prop} = 1.4$.

debido al incremento en K_{prop} ocasiona que los movimientos del cabezal sean más rápidos y den lugar a oscilaciones mayores en la carga; el valor máximo de la deflexión angular llega a 7.5° . Sin embargo, también se corrige el error con mayor rapidez y la respuesta es aceptable en términos de los errores admisibles en estado estable. Se observa que el movimiento del cabezal se aproxima más a la señal de referencia, pero con el sobreimpulso mencionado anteriormente (0.076 m), equivalente al 3.8% de la longitud recorrida.

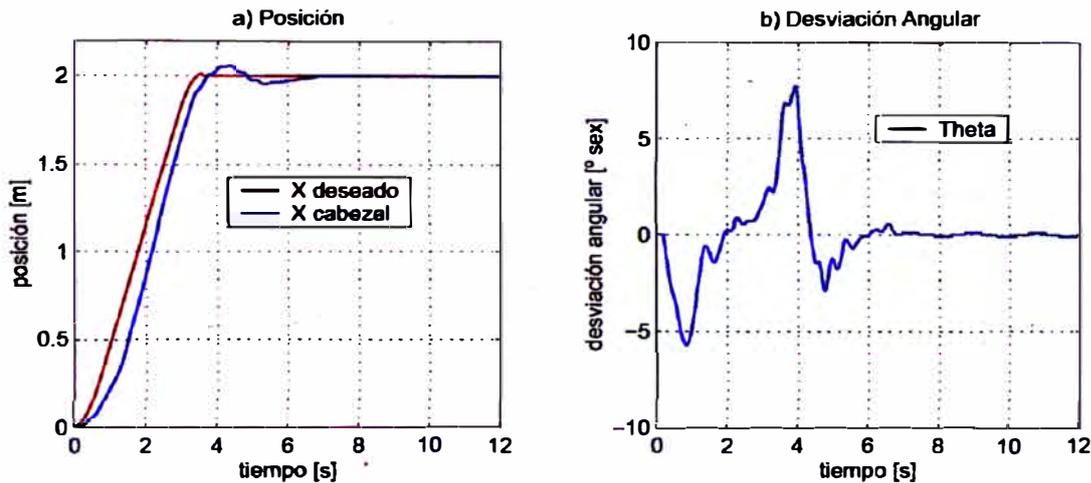


Figura 5.11: Respuesta experimental: Posición y desviación angular con $K_{prop} = 2.1$.

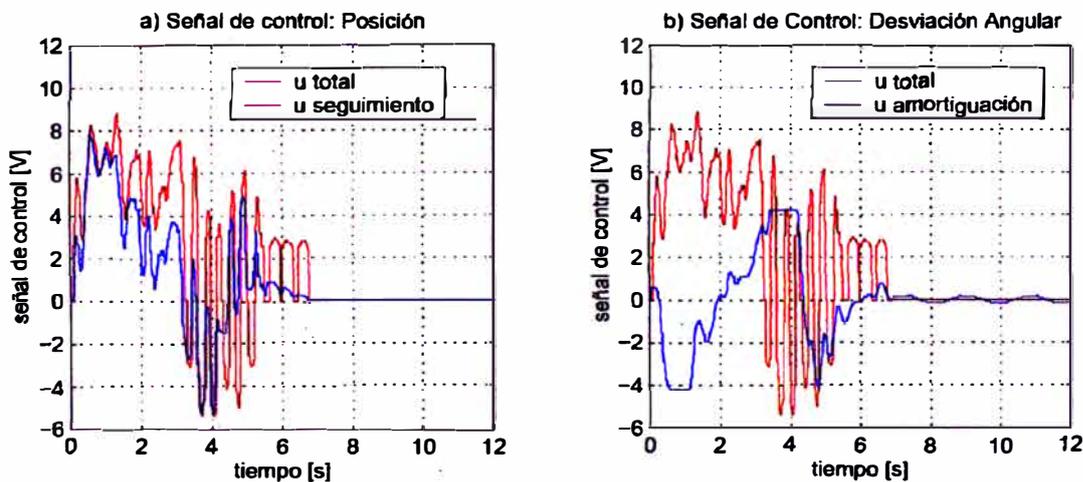


Figura 5.12: Respuesta experimental: Señal de control con $K_{prop} = 2.1$.

A continuación, se mostrará las respuestas del sistema cuando el valor de T_F es modificado. Dado que T_F es un valor relacionado con el período del péndulo formado por la carga suspendida (ver 4.1.1), para los experimentos siguientes se probará el sistema con el valor original y con la mitad de éste, vale decir, un período de oscilación de la carga.

En las Figuras 5.13 y 5.14 se muestra la respuesta del sistema controlado ante una referencia generada considerando el tiempo correspondiente a dos oscila-

ciones completas del péndulo para alcanzar la meta. Éste es el tiempo de ejecución deseado normalmente para el control del sistema, y los resultados obtenidos son similares a los mostrados en las Figuras 5.9 y 5.10, donde el sistema alcanza una posición de 2 m., en lugar de 1.5m. en el mismo tiempo. En 6.8 segundos alcanza la estabilidad satisfaciendo los objetivos de control y la respuesta no presenta sobreimpulso.

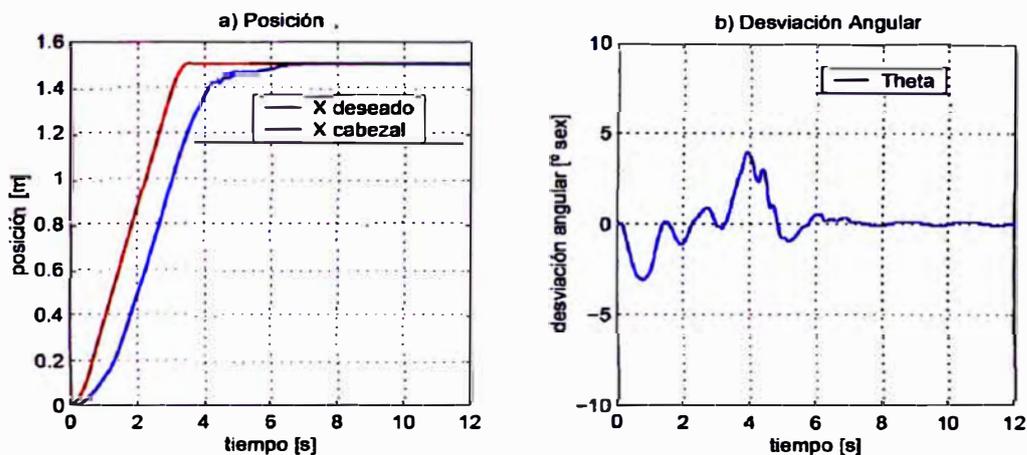


Figura 5.13: Respuesta experimental: Posición y desviación angular para un recorrido de 1.5m con un generador de trayectoria de dos oscilaciones.

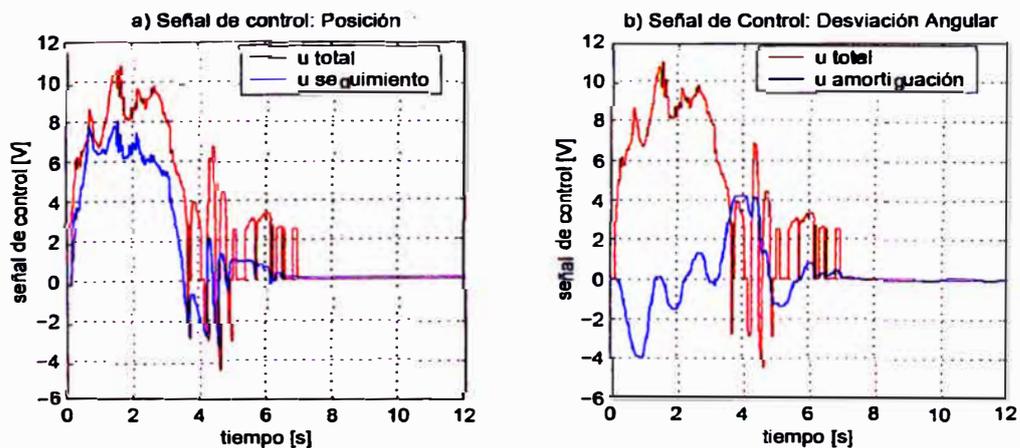


Figura 5.14: Respuesta experimental: Señal de control para un recorrido de 1.5m con un generador de trayectoria de dos oscilaciones.

En las Figuras 5.15 y 5.16, T_F toma como valor un período de oscilación pendular de la carga. En este caso, el tiempo para alcanzar el estado estable es ligeramente inferior, 6.5 segundos. Lo cual representa un 95.6 % del tiempo necesitado por el controlador cuando T_F vale el doble. Se podría esperar que el tiempo se redujera más; sin embargo cabe resaltar que la mayor aceleración inicial da lugar a mayores oscilaciones, cuya amortiguación toma un tiempo mayor. Sin embargo, se observa que el sistema controlado tiene errores menores que los admisibles. No se presenta sobreimpulso, pero la amplitud de la oscilación llega a 10° en valor absoluto. También, la señal de control alcanza una mayor amplitud, 18.0 V contra 11.0V en el caso de usar dos períodos de oscilación.

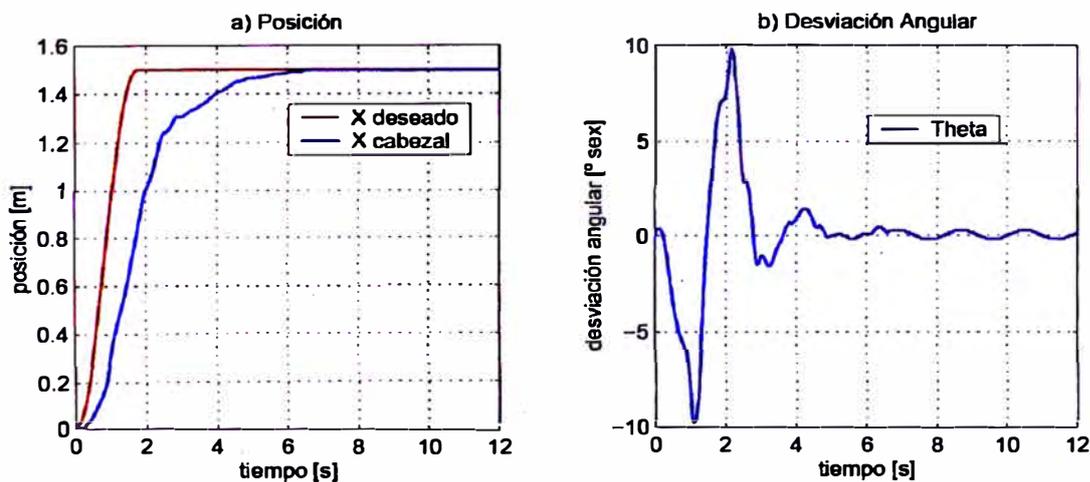


Figura 5.15: Respuesta experimental: Posición y desviación angular para un recorrido de 1.5m con un generador de trayectoria de una oscilación.

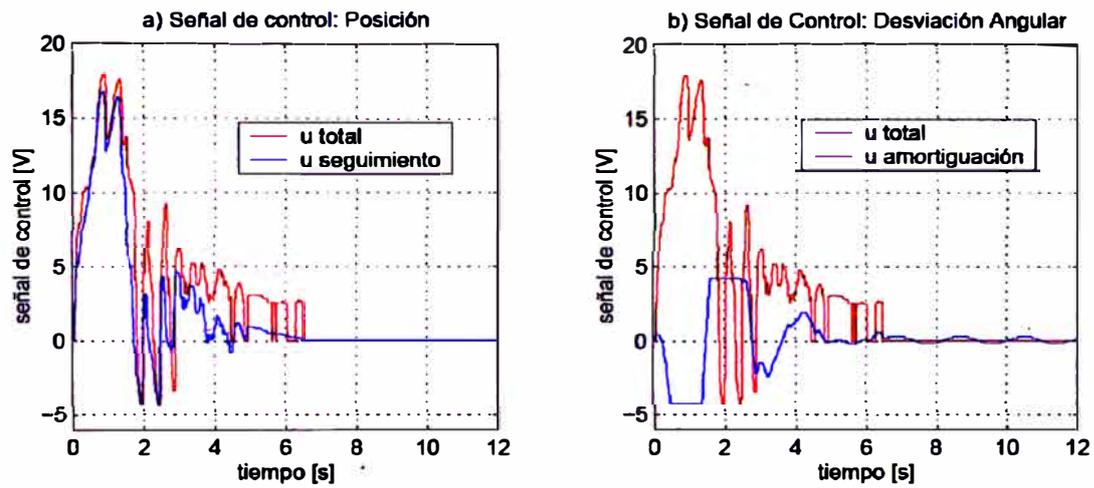


Figura 5.16: Respuesta experimental: Señal de control para un recorrido de $1.5m$ con un generador de trayectoria de una oscilación.

CONCLUSIONES

1. El sistema de control propuesto reduce efectivamente las oscilaciones de la carga, dentro del rango admisible, establecido en ± 0.3 grados sexagesimales. En los experimentos en los que se incluye la compensación del efecto de la zona muerta del motor, la oscilación residual es menor a ± 0.2 grados, que corresponden a $\pm 0.35\%$ de la longitud suspendida. En una aplicación portuaria, para una longitud suspendida de 20 metros, la amplitud de esta oscilación sería equivalente a ± 70 mm, que es suficientemente pequeña para descargar con facilidad un contenedor sobre un trailer o sobre la ubicación respectiva en el navío.

2. El sistema de control propuesto controla efectivamente la posición del cabezal, con un error de posición menor a $\pm 0,2$ mm en una longitud de traslado de 1.5 m para una carga de 1.35 kg. Este error corresponde a $\pm 0.013\%$ del recorrido; equivalente a ± 7 mm para una longitud de 50 m, que corresponde a la distancia típica entre el extremo más alejado del navío y el punto de descarga en el muelle. Un error de ± 7 mm se considera pequeño, dado que la tolerancia máxima para la ubicación de un contenedor sobre otro es de 100 mm [2].

3. La presente aplicación del control por inferencia difusa al sistema grúa-puente es exitosa, en la medida que reduce la oscilación, prácticamente anulándola, y posiciona automáticamente la carga. Esto permite el incremento de la relación de caja, aumentando el número de contenedores que pueden ser cargados o descargados por hora.

4. La compensación de la zona muerta del motor eléctrico es imprescindible para conseguir un desempeño aceptable. Si bien la zona muerta es una característica similar de los motores de corriente directa, otros fenómenos similares pueden encontrarse en motores asíncronos, que son los empleados en las grúas-puente industriales. Se concluye que los fenómenos fuertemente no-lineales y de efecto significativo deben ser compensados para conseguir un desempeño aceptable.

5. El uso de 2 períodos pendulares como referencia para la generación de la trayectoria es necesario para conseguir un desempeño aceptable. Los experimentos con 1 solo período pendular generan mayor oscilación inicial, lo que dificulta el posicionamiento fino y la amortiguación de las oscilaciones remanentes. Sin embargo, esto puede deberse a los límites de potencia del actuador y de rigidez de la estructura; este análisis no ha sido incluido en la presente investigación y se recomendará para futuros trabajos.

6. El sistema de control tiene un desempeño adecuado cuando la masa de la carga corresponde a la de diseño (1.35 Kg), al incrementarse la carga a 2.35 kg, la oscilación residual es aceptable (± 0.2 grados sexagesimales), sin embargo el error de posición del cabezal es ± 4 mm, que representa un error de $\pm 0.27\%$, o 133 mm para un recorrido de 50 m. Aunque la tolerancia máxima es 100 mm, puede proceder la descarga con un ajuste manual final. Cabe notar que esta carga de prueba supera en 74 % a la carga de diseño. Tal como en el caso anterior, esta diferencia en el desempeño puede deberse a las limitaciones del actuador, o más específicamente a la modificación del comportamiento de la zona muerta para una carga apreciablemente mayor.

RECOMENDACIONES

Se recomienda el desarrollo de las siguientes investigaciones o implementaciones:

1. Diseño e implementación de un nuevo modelo a escala reducida para experimentación, de mayores dimensiones y capacidad que el actual, tomando como referencia un modelo específico de grúa-puente industrial. Para este nuevo modelo, se propone el cambio de la varilla por un cable, con el fin de estudiar y controlar el efecto debido a la flexibilidad del cable. Este cambio involucraría el uso de un sensor apropiado para medir el ángulo o la posición de la carga.

2. Implementación industrial con una grúa puente en operación, usando un DSP o un PLC para dar la referencia de velocidad a un VFD. Se sugiere la propuesta de un convenio de asistencia técnica con la administración del Terminal Portuario del Callao (ENAPU) para la implementación industrial de este sistema de control como un prototipo para una de las grúas-puente actualmente en uso.

3. Desarrollar un enfoque alternativo, para dar asistencia al operador de grúas mediante la compensación de oscilaciones, en lugar del control totalmente automático. Este sistema de traslación asistida amortiguaría las oscilaciones, dejando el control de posición en manos del operador.

4. Se recomienda estudiar la incorporación de un algoritmo de compensación de la fricción y del juego mecánico (*backlash*). Se sugiere usar el modelo de

fricción de Lund-Grenoble, [20], [18].

5. Se propone el desarrollo de otras estrategias de control, las cuales se pueden implementar en el equipo usado en la presente investigación. Se sugiere considerar las estrategias de control por retroalimentación con retardo (*time delayed feedback*), control neuro-difuso (*neuro-fuzzy*) y control híbrido por modo deslizante y difuso (*fuzzy sliding mode*), [23].

6. Se propone el estudio de la automatización total del sistema de carga y descarga. Dicho sistema involucraría la determinación de la ubicación inicial y final de la carga de manera automática, mediante el empleo de seguimiento laser ("*laser painting*") u otras tecnologías para ubicar la ubicación del objeto a tomar; para luego izarlo, trasladarlo y descargarlo en su posición final.

BIBLIOGRAFÍA

- [1] American Association of Port Authorities. World port rankings. 2009.
- [2] Monica Louda. *On Load Sway Estimation and Control of Quay Cranes*. PhD dissertation, University of Sidney, Australian Centre for Field Robotics, School of Aerospace, Mechanical and Mechatronic Engineering, 2002.
- [3] Ryan J. Henry. Cargo pendulation reduction on ship mounted cranes. Master's thesis, Virginia Tech., 1999.
- [4] Lee Y.Y. & Fang C. Balachandran, B. A mechanical filter concept for control of non-linear crane-load oscillation. *Journal of Sound and Vibration*, pages 228(3):651–682, 1999.
- [5] Ong C.J. & Xu M. Teo, C.L. Pulse input sequences for residual vibration reduction. *Journal of Sound and Vibration*, pages 211(2):157–177, 1998.
- [6] Marttinen A. & Virkkunen J. Salminen, R. Adaptive pole placement control of a piloted crane. *Proceedings of the IFAC 11th Trienal World Congress, Tallinn Estonia, USSR*, pages Vol. 2:313–318, 1990.
- [7] R. Hurteau, R. & Desantis. Microprocessor-based adaptive control of a crane system. *Proceedings of the IEEE Conference on Decision and Control Including The Symposium on Adaptive Processes, New York*, pages Vol. 2:944–947, 1983.

- [8] A. Rojas Moreno. *Control Avanzado: Diseño y Aplicaciones en Tiempo Real*. Publicación Independiente, 2001.
- [9] L. A. Zadeh. Fuzzy sets. *Information and Control*, pages 8:338–353, 1965.
- [10] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics*, pages 3(1):28–44, Enero 1973.
- [11] P. M. Larsen. Industrial applications of fuzzy logic control. *International Journal of Man-Machine Studies*, pages 12(1):3–10, 1980.
- [12] Mizumoto M. & Tanaka K. Fukami, S. Some considerations on fuzzy conditional inference. *Fuzzy Sets and Systems*, pages 4:243–273, 1980.
- [13] M. Takagi, T. & Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, pages 15:116–132, 1985.
- [14] G. T. Sugeno, M. & Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, pages 28:15–33, 1998.
- [15] S. Mamdani, E.H. & Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, pages 7(1):1–13, 1975.
- [16] Weiping Slotine, Jean-Jacques E. & Li. *Applied Nonlinear Control*. Prentice-Hall, 1991.
- [17] Rogelio Fantoni, Isabelle & Lozano. *Non-linear Control for Underactuated Mechanical Systems*. Springer-Verlag, 2002.
- [18] Campos J. & Selmic R. Lewis, F. *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*. SIAM, 2002.

- [19] Noël P. Aubin A. & Brogliato B. Canudas de Wit, C. Adaptive friction compensation in robot manipulators: Low velocities. *International J. Robotics Res.*, page 10, 1991.
- [20] Olsson H. Åström K.J. & Lischinsky P. Canudas de Wit, C. A new model for control of systems with friction. *IEEE Trans. Automat. Control*, pages 40:419–425, 1995.
- [21] D. M. Patyra, M. J. & Mlynek. *Fuzzy Logic: Implementation and Applications*. Wiley-Teubner, 1996.
- [22] Patricio D. Navarro, Gustavo A. & San Martin. Compensación de oscilaciones y control de posición de una carga por lógica difusa para un prototipo de grúa-puente. *UNI-INIFIM: I+D Investigación y Desarrollo*, pages 31–38, Febrero 2004.
- [23] Cheng-Yuan Chang. Adaptive fuzzy controller of the overhead cranes with nonlinear disturbance. *IEEE Trans. Industrial Informatics*, pages Vol. 3 No. 2:164–172, 2007.
- [24] Muhammad H. Rashid. *Electrónica de Potencia Circuitos, Dispositivos y Aplicaciones*. Prentice-Hall, 1995.
- [25] Texas Instruments. *TMS320F240 DSP Controller Technical Data*. Texas Instruments Inc., 2002.
- [26] Texas Instruments. *TMS320F/C240 Controllers Reference Guide*. Texas Instruments Inc., 1999.
- [27] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

APÉNDICE A
IMPLEMENTACION EXPERIMENTAL

Con el fin de validar el diseño de los controladores y las simulaciones numéricas mediante pruebas experimentales, se construyó e instaló un prototipo de grúa-puente en las instalaciones del INIFIM¹. Este prototipo es un modelo a escala reducida de una grúa-puente empleada en la manipulación de containers. Los algoritmos de control se implementan en un microcontrolador *DSP*, el *TMS320F240* de Texas Instruments, empleando lenguajes *C* y *Assembler* para este dispositivo. El movimiento del cabezal de la grúa se consigue con un servomotor DC, cuyo encoder proporciona información de la posición del cabezal. Para obtener el valor del ángulo de desviación entre la varilla y la vertical se emplea también un encoder. Las salidas de los encoders se conectan con los contadores incorporados en el *DSP*. El voltaje que recibe el servomotor es suministrado por un amplificador basado en un puente H de transistores *MOSFET*², acoplado ópticamente con las salidas moduladas en ancho de pulso del microcontrolador. En la figura A.1 se muestra la organización de los dispositivos de control en conjunto con el prototipo.

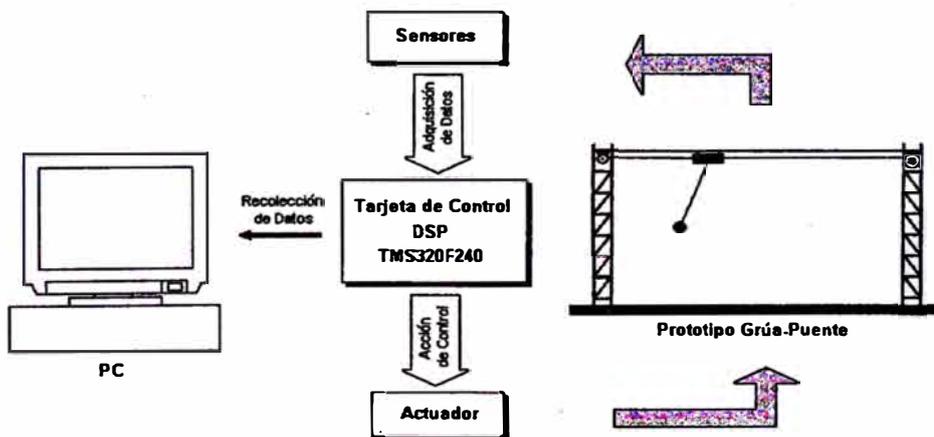


Figura A.1: Organización de la implementación experimental.

¹El Instituto de Investigación de la Facultad de Ingeniería Mecánica financió la construcción de este prototipo.

²Metal-Oxide-Semiconductor Field Effect Transistor: Transistor de Efecto de Campo Metal-Óxido-Semiconductor

A.1. Prototipo de Grúa-Puente

Este prototipo fue ideado originalmente como un modelo a escala de una grúa de pórtico real empleada para el traslado de contenedores. Como sus contraparte industrial, la estructura consiste de dos vigas con rieles sobre las que se se desliza un carro. La relación entre la altura útil y la envergadura es similar a la que se mantiene en los equipos industriales; así también la relación entre la masa de la carga y la masa de la grúa. Si bien se ha tenido cuidado en la precisión de mecanizado y manufactura en general de los componentes, ésta no llega a ser de alta precisión, dando lugar a que los fenómenos no lineales se presenten también en una magnitud significativa. Esta inclusión de “imperfecciones” en la maquinaria es voluntaria, y tiene como fin probar el desempeño de los controladores ante estas dificultades, que están siempre presentes en el mundo real y que son siempre de importancia. Puede observarse una fotografía del prototipo construido en la figura.

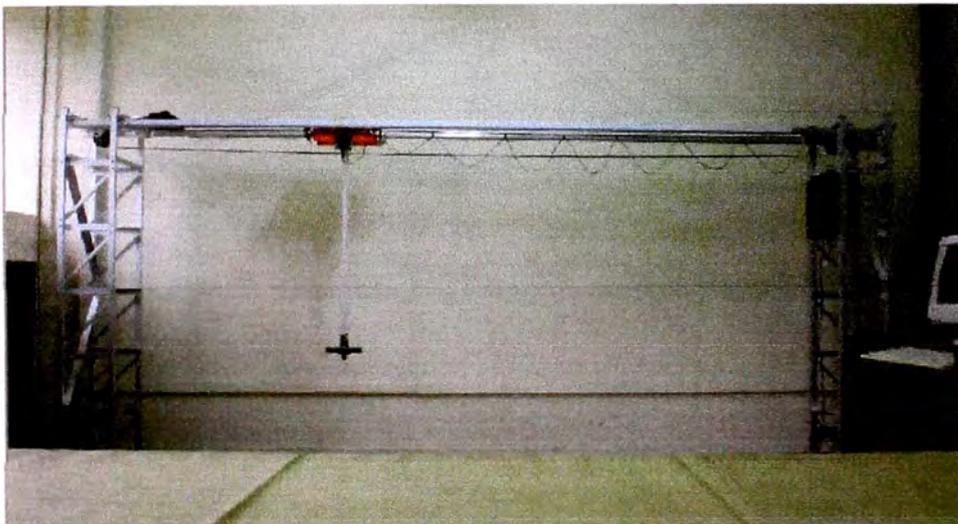


Figura A.2: Prototipo Grúa-Puente.

En la figura A.3 se muestran las dimensiones del prototipo empleado y en la figura A.4 se muestran las de una empleada en el puerto de .

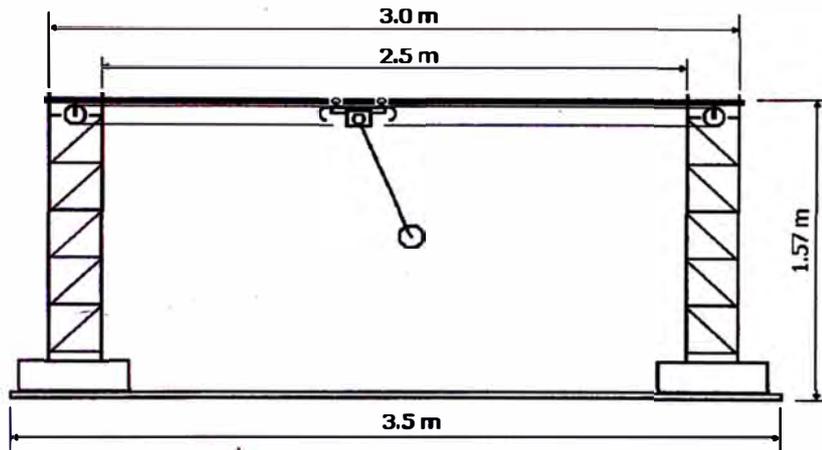


Figura A.3: Dimensiones del prototipo.

El riel tiene una longitud útil de 2.2 m. El cabezal tiene un peso de 670 g. La varilla tiene un peso de 170 g. y una longitud de 83 cm. La carga pesa 2 kg. y el soporte para acoplar la carga pesa 350 g. Como referencia, la Figura A.4 muestra las dimensiones de una grúa-puente de un puerto de contenedores.

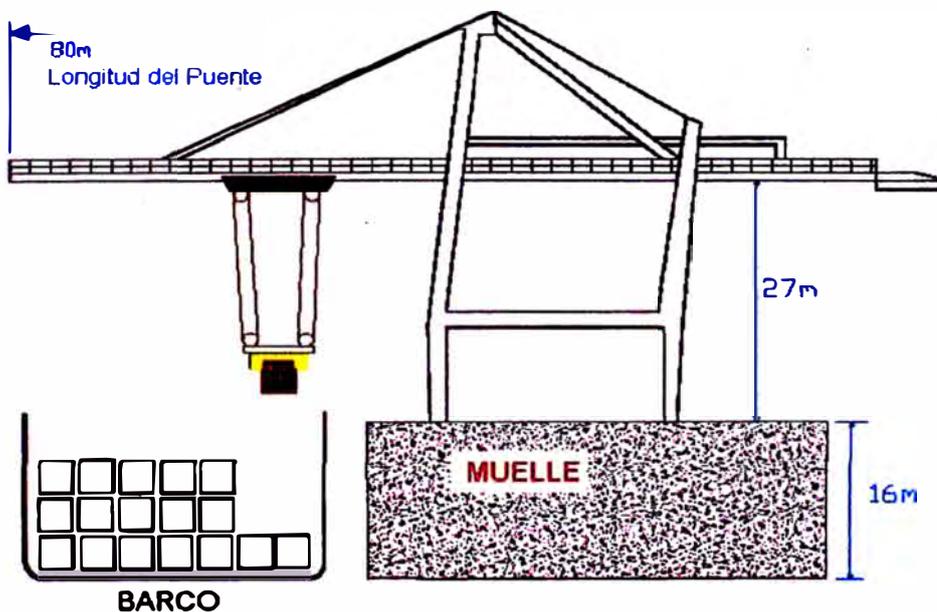


Figura A.4: Dimensiones generales de una grúa puente.

El sistema de tracción utiliza dos engranes de cadena de 7.5 cm. de diámetro

en ambos extremos del riel horizontal y una cadena que se encuentra acoplada al cabezal. En el cabezal (ver figura A.5) se encuentra un eje por donde el péndulo gira libremente. En este mismo eje se encuentra acoplado un encoder para obtener la posición angular del péndulo.

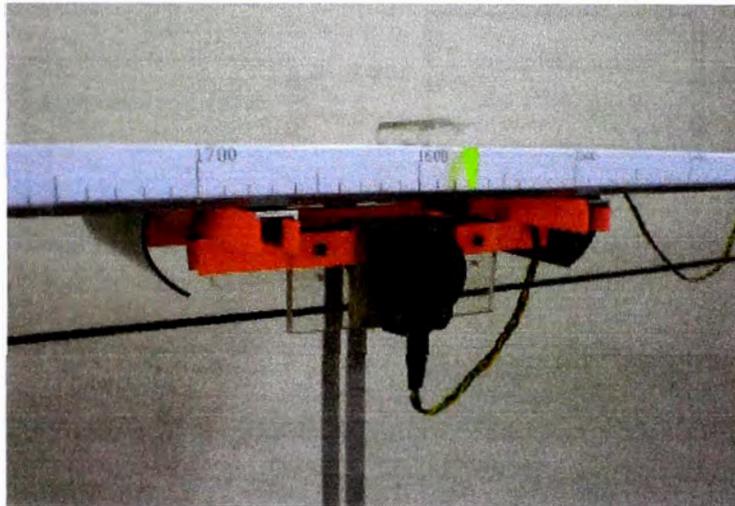


Figura A.5: Cabezal del prototipo grúa-puente.

A.2. Actuador

El actuador utilizado para el control del prototipo GP, está conformado por un generador de PWM, optoacopladores, un amplificador de potencia y un motor-reductor DC, tal como se muestra en la figura A.6.

A.2.1. Generador de PWM

En sistemas de control de motores, las señales PWM son utilizadas para controlar el tiempo de encendido y apagado de los dispositivos conmutadores que entregan la energía deseada al motor [24]. La forma y la frecuencia de las fases de la corriente y voltajes y la cantidad de energía entregada al motor controla la velocidad y torque requerido. El voltaje deseado de entrada a ser aplicada al motor es la señal de modulación. La frecuencia de modulación es generalmente

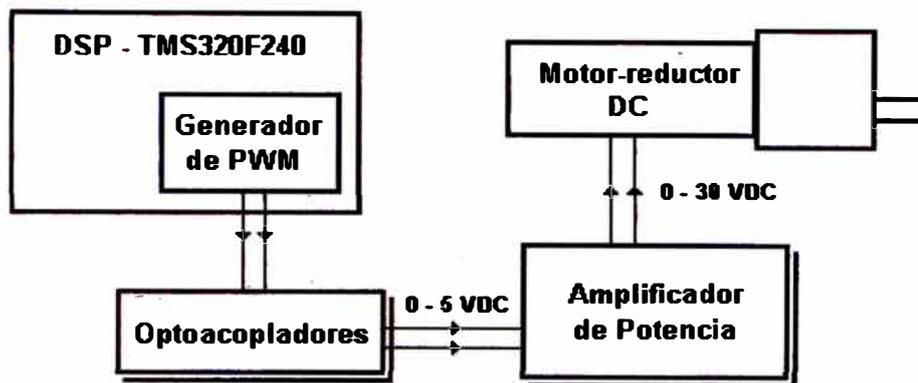


Figura A.6: Componentes del Actuador.

mucho menor que la frecuencia del PWM portador.

El generador de PWM utilizado se encuentra dentro del DSP TMS320F240, este se encarga de conmutar al motor mediante una serie de pulsos, logrando así, variar la velocidad de acuerdo al ancho de pulso. Más adelante, en A.4, se describe las características de este módulo interno del DSP.

A.2.2. Optoacopladores

El integrado 4N33 es un acoplador óptico compuesto por un led infrarrojo de galio-arsénico y un foto-transistor Darlington. Este optoacoplador es usado para aislar el circuito de control del circuito de potencia, con el propósito de proteger los componentes del circuito de control. Las características de este integrado son: alta tasa de transferencia de corriente (500 % mínimo) y alta resistencia de aislamiento ($10^{11}\Omega$). Este optoacoplador fue implementado tal como lo muestra la figura A.7 para ambas señales PWM provenientes del DSP, con esto logramos aislarlo del amplificador de potencia.

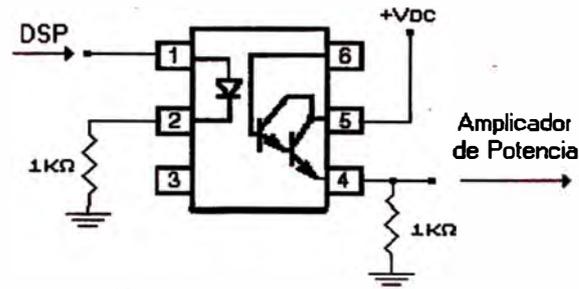


Figura A.7: Circuito del Optoacoplador 4N33.

A.2.3. Amplificador de Potencia

El amplificador de potencia utilizado para este motor-reductor DC es un puente-H, el esquema de su funcionamiento se muestra en la figura A.8, y está conformado por un circuito de control de disparo y un circuito de conmutación llamado puente-H. El circuito de control de disparo activa o desactiva a los conmutadores por los pares (A,C) y (B,D) para hacer girar al motor.

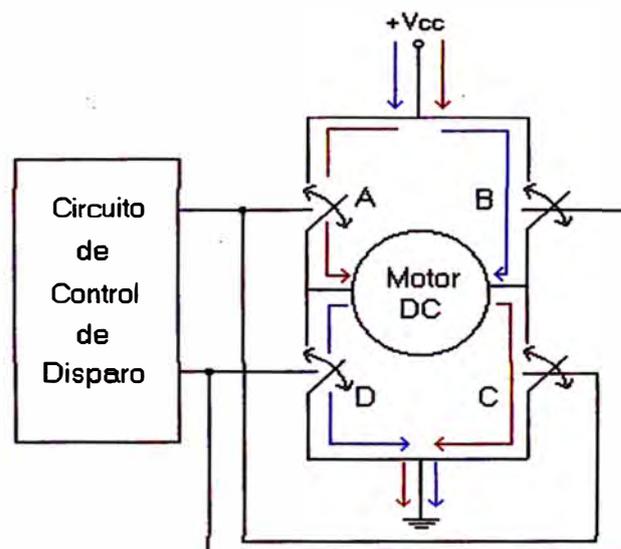


Figura A.8: Esquema del Amplificador de Potencia.

La razón de utilizar estos componentes, el generador de PWM y el amplificador de potencia es que como el motor-reductor DC no responde inmediatamente

a una tensión de entrada, es necesario usar un sistema que permita variar la velocidad entre un valor mínimo y un valor máximo, y que permita dar un cambio seguro en el sentido de rotación del eje del motor-reductor DC.

El puente-H utiliza transistores MOSFETs por tener mejor desempeño que otros tipos de transistores, mientras que el circuito de control de disparo utiliza transistores BJTs, en la figura A.9 se muestra este circuito.

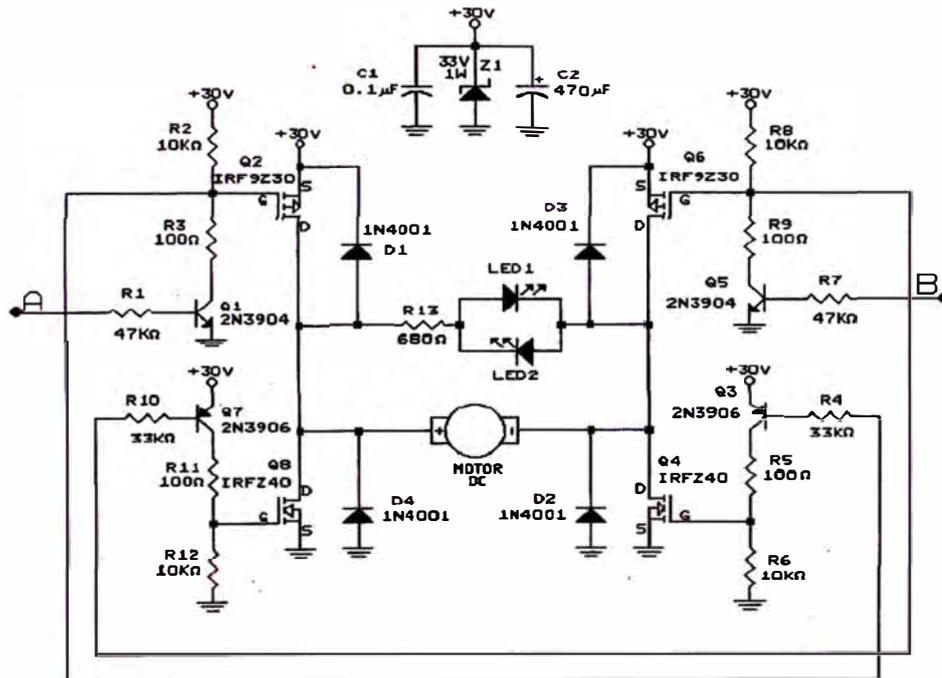


Figura A.9: Amplificador de Potencia - Puente H.

Los MOSFETs trabajan aplicándoles un voltaje en la compuerta. Cuando un voltaje positivo mayor que el voltaje umbral de la compuerta es aplicado, el MOSFET se activa (Q4 y Q8 - Canal N). Los de canal P trabajan en reversa (Q2 y Q6), es decir, cuando un voltaje negativo menor que el voltaje umbral de la compuerta es aplicado, el MOSFET se activa. Los diodos D1 a D4 conducen la fuerza contraelectromotriz desde el motor hasta la fuente de alimentación [24]. Los modos de operación se muestran en la tabla A.1.

	A	B
MODO PARADA	0	0
MODO AVANCE	1	0
MODO REVERSA	0	1
NO PERMITIDO	1	1

Cuadro A.1: Modos de Operación del Puesto H.

A.2.4. Motor-Reductor DC

El motor-reductor DC utilizado es un motor de marca *PITTMAN* modelo GM9236C534-R2 (ver figura A.10). Este motor-reductor es de imán permanente, con un estator de 2 polos, magnetos cerámicos, eje de acero inoxidable y escobillas de cobre-grafito.

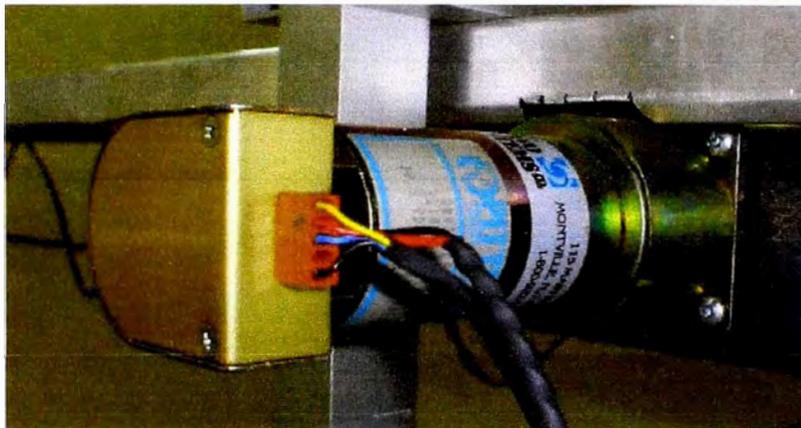


Figura A.10: Motor-reductor DC GM9236C534-R2.

Las principales características del motor-reductor DC son: voltaje de alimentación de +30 VDC, velocidad sin carga de 800 rpm, torque continuo de 0.34 N-m (max.), torque pico de 2.0 N-m (parado) y una reducción de 5.9 : 1.

A.3. Sensores

A.3.1. Sensor del Ángulo de Oscilación de la Carga

El sensor utilizado para la medición de la posición del péndulo se encuentra montado sobre el eje de giro del péndulo en el carro, su modulo de encoder tiene

1024 CPR, éste se encuentra conectado directamente a los pines QEP1 y QEP2 del DSP TMS320F240, tal como lo muestra la figura A.11.

Dentro del DSP, estas entradas se encuentran asociadas al módulo interno de detección de cuadratura, su factor de resolución es de X4 obteniendo 4096 PPR, y que a su vez utiliza un reloj interno configurado como contador para realizar el conteo y sentido de giro del encoder óptico incremental.

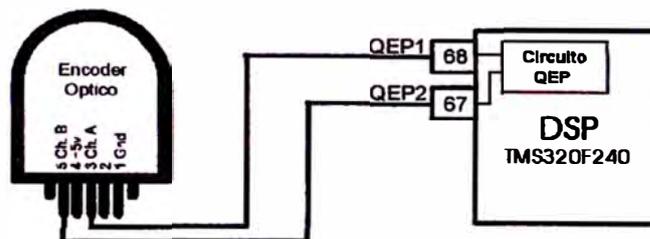


Figura A.11: Configuración del Sensor de Posición del Péndulo.

A.3.2. Sensor de la Posición del Cabezal

El sensor utilizado para la medición de la posición del carro se encuentra montado en el motor-reductor DC, su módulo óptico de 512 CPR, que debido a la reducción del motor $n = 5.9$ se incrementa a 3020 PPR, éste encoder se encuentra conectado al integrado LS7184 con la configuración en modo X1, con un ancho de pulso bajo de la salida de reloj de $3.0\mu\text{s}$ y conectado a los pines TMRCLK y TMRDIR del DSP TMS320F240, tal como lo muestra la figura A.12.

Dentro del DSP, estas entradas se encuentran asociadas a otro reloj interno configurado como contador para realizar el conteo y dirección del encoder óptico incremental.

A.4. Controlador *DSP TMS320F240*

Esta tarjeta de control utiliza un DSP marca *Texas Instrumets* modelo TMS320F240 (ver figura A.13). Este DSP utiliza una arquitectura de tipo Har-

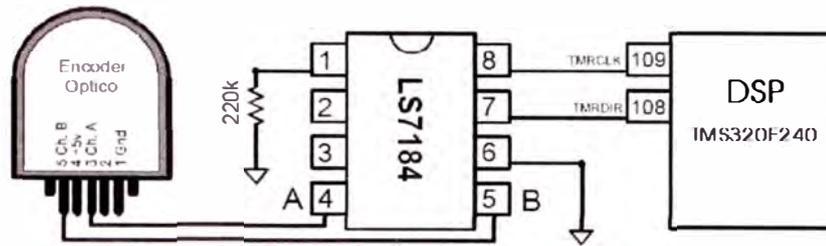


Figura A.12: Configuración del Sensor de Posición del Carro.

vard que maximiza la potencia de procesamiento al mantener dos buses de memoria, de datos y programa, para una ejecución a máxima velocidad. De este modo se pueden leer los datos y las instrucciones simultáneamente [25].

El microprocesador tiene implementado un conjunto de instrucciones que soporta tanto operaciones de procesamiento de señales numéricamente intensivas, como aplicaciones de propósito general, tal como control automático de alta velocidad.



Figura A.13: Tarjeta de Control DSP - TMS320F240.

Se dispone de cuatro modos de direccionamiento básicos: directo, indirecto,

inmediato y por registros auxiliares. La memoria est'a organizada en 4 espacios seleccionables; el de programa, el de datos local, el datos globales y el de entradas y salidas I/O.

Los periféricos I/O del DSP TMS320F240 son: 3 Relojes de Propósito-General de 16-bits (GP Timers), 2 Convertidores A/D simultáneos de 10 bits, 9 Unidades de Generación de PWM, 2 Sistemas de Comunicación Serial (SPI/SCI), 4 Unidades de captura de tiempos y dos en cuadratura-encoder (QEP), Entradas y Salidas I/O programables.

A.4.1. Relojes de Propósito General

Existen tres relojes de propósito general (GP Timers). Estos relojes son usados como bases de tiempo independientes para las diversas aplicaciones necesarias dentro del algoritmo de control [26].

El registro de las unidades de comparación asociadas a cada reloj puede ser usada para comparar y generar señales PWM. Se puede tener entradas de reloj internas o externas con en pre-escalador programable para cada reloj mediante su registro GPTCON.

- Reloj 1: Genera el periodo de muestreo del sistema de control y provee de una base de tiempo para las operaciones de comparación asociadas con la generación de salidas PWM.
- Reloj 2: Proveer de una base de tiempo para la operaciones del circuito cuadratura encoder pulso (QEP).
- Reloj 3: Proveer de una base de tiempo para la operaciones de los pines asignados al detector de cuadratura externo (TMRDIR/TMRCLK) del sensor de posición del cabezal.

A.4.2. Configuración del Módulo PWM

Utilizamos un par de salidas PWM1 y PWM2 generadas por medio de la unidad de comparación (*full-compare*) que tiene tiempo muerto (*deadband*) programable. La programación de esta unidad de generación de pulsos se realiza configurando los registros COMCON, ACTR y DBTCON. Para generar la señal PWM, se utiliza el reloj 1, necesario para repetir el periodo de conteo, que es el mismo que el periodo PWM.

En muchas aplicaciones en motores y electrónica de potencia, dos dispositivos de potencia (superior e inferior) son colocados en serie en una línea de conversión de potencia. Para evitar que ocurra un falla por corto entre los polos, los periodos de encendido de los dos dispositivos no deben traslaparse. Entonces, un par de salidas PWM *sin-traslape* es frecuentemente requerido para encender los dos dispositivos de potencia correctamente.

Un tiempo muerto (*deadband*) es generalmente insertado entre el apagado de un transistor y el encendido del otro transistor. Este retraso permite el completo desactivado del transistor antes de que el otro sea activado. El tiempo requerido para este retraso es especificado por las características de encendido y apagado del transistor utilizado y de las características de carga de la aplicación en donde será aplicado. En la figura A.14 se muestra la generación del par de salidas PWM con tiempo muerto.

A.4.3. Decodificador del Tren de Pulsos en Cuadratura

El circuito decodificador de pulsos de encoder en cuadratura (QEP) es activado con el registro CAPCON para detectar y contar los pulsos de entrada en cuadratura por los pines CAP1/QEP1 y CAP2/QEP2. Este circuito es usado como interfase con el encoder óptico para obtener la posición y velocidad angular del péndulo.

La base de tiempo para el circuito QEP se obtiene del reloj 2 de 16-bits. La

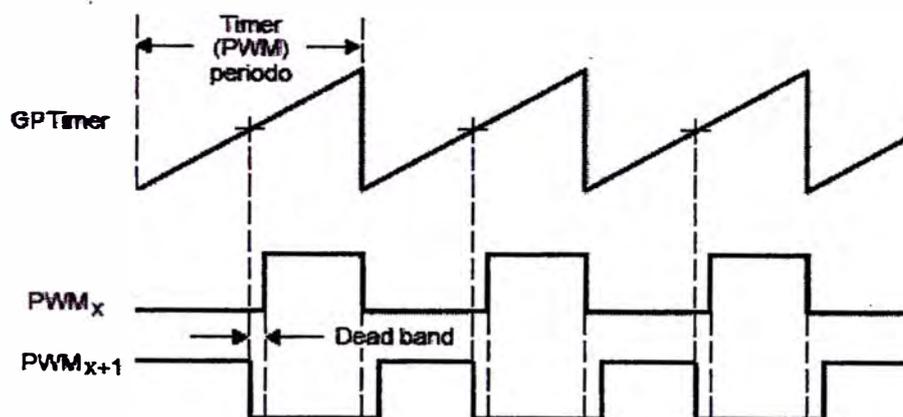


Figura A.14: Generación de salidas PWM con *deadband*.

selección se realiza configurando el registro T2CON. El reloj seleccionado debe ser configurado en el modo de conteo direccional up/down con el circuito QEP como fuente de reloj.

La lógica de detección de dirección del circuito QEP determina cual secuencia está en adelante. Esta genera una señal de dirección para la entrada de dirección del reloj seleccionado. El reloj seleccionado cuenta en forma ascendente o descendente si la entrada CAP1/QEP1 está en adelante o atraso de fase con respecto a la entrada CAP2/QEP2, respectivamente. Ambos flancos de las entradas en cuadratura son contados por el circuito QEP. Entonces, la frecuencia del reloj generado por el circuito QEP es cuatro veces la frecuencia de cada entrada, como lo muestra la figura A.15.

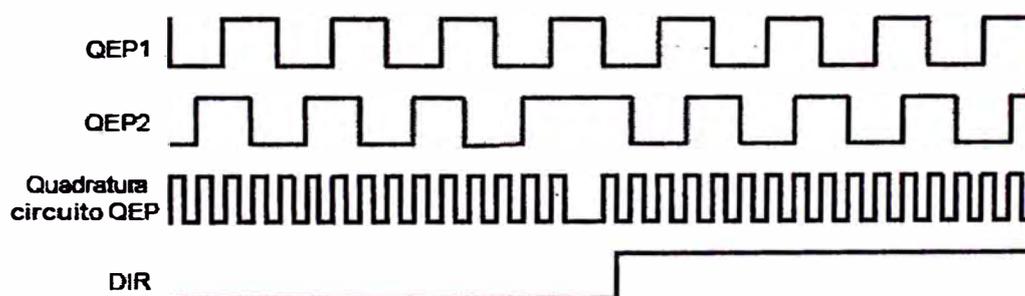


Figura A.15: Diagrama de Tiempos del Circuito QEP.

A.5. Programa del Controlador en Tiempo Real

En esta sección se muestra el diagrama de flujo del algoritmo de control (ver figura A.16) implementado en el DSP TMS320F240 (ver Apéndice D). A continuación se explican los principales pasos del algoritmo:

1. Inicialización de los registros y pines I/O de la tarjeta de control.
2. Seteo de variables de control, constantes y referencia deseada.
3. Inicio del bucle de control mientras que t menor que el tiempo de simulación.
4. Se realiza la adquisición de los datos provenientes de los sensores.
5. Se realiza el cálculo de la acción de control y el compensador de zona muerta del motor correspondiente a ese instante.
6. Se envía la acción de control y el compensador hacia el actuador. Luego de esto se cierra el bucle de control.

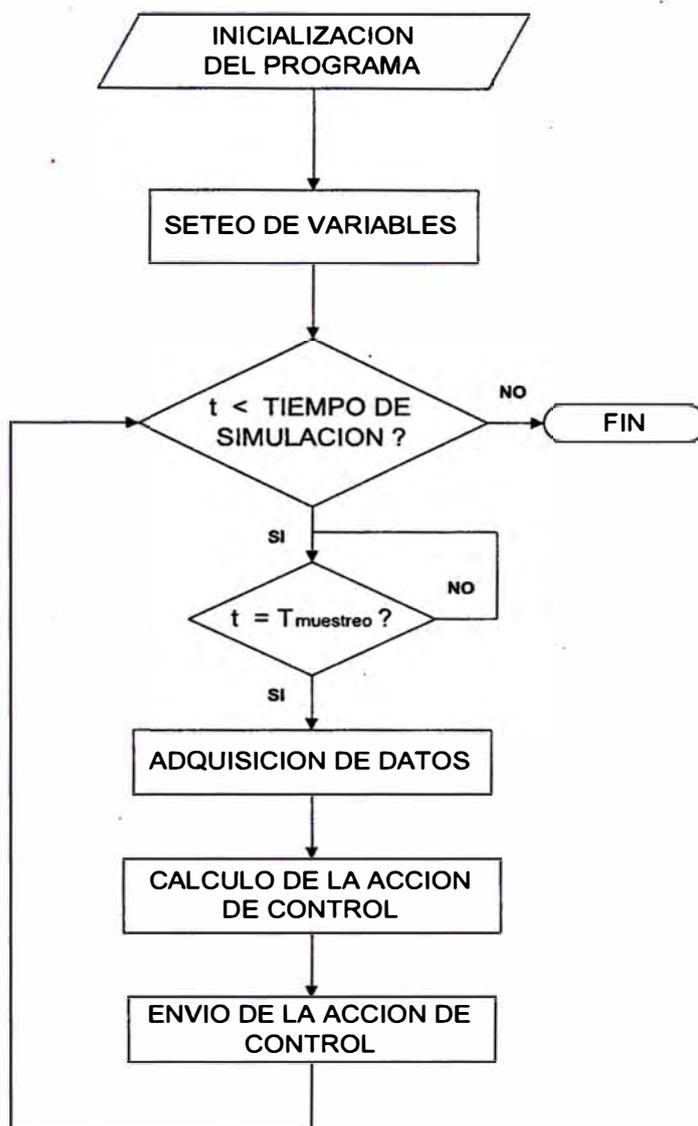


Figura A.16: Diagrama de Flujo del Algoritmo de Control.

APÉNDICE B
IDENTIFICACIÓN DE PARÁMETROS

B.1. Identificación del Motor-Reductor DC

Se empleará el método gráfico, asistido por computadora a través de Matlab, para identificar los parámetros del motor-reductor DC. Para este método se considera la aproximación inicial a un sistema de primer orden en el que se desprecia la inductancia L_a . Se tomará en cuenta los datos de laboratorio suministrados por el fabricante del motor-reductor para la identificación. El motor-reductor es fabricado *PennEngineering Motion Technologies* y pertenece a la serie *Pittman*, modelo GM9236C534-R2, con encoder incremental.

Para realizar la identificación se lleva a cabo un experimento en el que se excita el motor-reductor con una entrada escalón y se registra la velocidad angular. Se edita un programa en C++, con el respectivo programa de configuración en Assembler del DSP y en Matlab se edita el programa que analiza el gráfico para la identificación:

- `parDC.c`, contiene el programa principal para adquisición de la data de velocidad (ver apéndice D.1).
- `servomotordc.m`, grafica y analiza los datos registrados para identificar los parámetros del motor-reductor. (ver apéndice C.3).

En el gráfico se determinan primero el valor máximo de la velocidad y el punto de inflexión en el transitorio. El valor máximo de la velocidad se divide entre el valor nominal del voltaje para hallar K_{med} , que se utilizará más adelante para calcular otros parámetros. En el punto de inflexión, una recta tangente a la gráfica de velocidad se interseca con la recta horizontal máxima, de modo que la abscisa del punto de intersección define a T_{med} ; que se utilizará también para la identificación. En la figura B.1 se aprecia lo descrito anteriormente.

$$K_{med} = 15.9753 \quad T_{med} = 0.0574$$

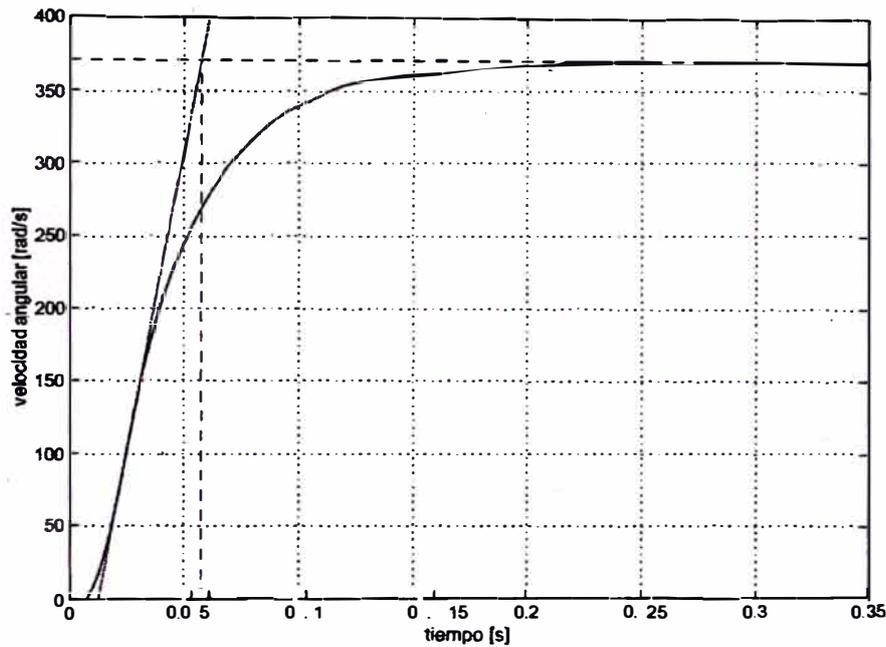


Figura B.1: Curva de Velocidad del Motor-Reductor DC.

También, disponemos de la información del proveedor del motor-reductor DC, ver tabla B.1. Las condiciones son: voltaje de alimentación igual a 30 Vdc y velocidad sin carga igual a 800 RPM.

Velocidad (rad/s)	Corriente (A)	Torque (N.m)
83.77	0.1	N.L.
67.02	1.04	0.2825
54.45	1.90	0.5649
41.89	2.85	0.8474
25.13	3.80	1.1298
STALL	4.85	1.4123

Cuadro B.1: Resultados de Ensayos en Laboratorio.

Siguiendo el mismo análisis realizado para el subsistema motor-polca de la sección 3.3.2, utilizaremos una función de transferencia de primer orden de la forma,

$$\frac{W(s)}{V_a(s)} = \frac{K_m}{R_a J_{eq} s + (R_a B_{eq} + K_m K_b)} = \frac{K_{med}}{T_{med} s + 1} \quad (B.1)$$

donde,

$$K_{med} = \frac{K_m}{R_a B_{eq} + K_m K_b} \quad (B.2)$$

y

$$T_{med} = \frac{R_a J_{eq}}{R_a B_{eq} + K_m K_b} \quad (B.3)$$

La resistencia de armadura R_a se obtiene de la siguiente relación:

$$R_a = \frac{V}{I_{a|STALL}} = \frac{23.2}{4.0} = 5.8 \text{ } [\Omega] \quad (B.4)$$

La constante contraelectromotriz K_b se obtiene de acuerdo a la tabla B.1 y considerando que la inductancia se desprecia $L_a = 0$.

$$K_b = \frac{V_a - R_a i_a}{\pi \omega} = \frac{23.2 - (5.8)(0.1)}{5.9(83.77)} = 0.0580 \text{ } [V/rad/s] \quad (B.5)$$

La constante de torque K_m también se obtiene de acuerdo con la tabla B.1,

$$K_m = \frac{T_{m|STALL}}{I_{a|STALL}} = \frac{0.2394}{4.85} = 0.0494 \text{ } [N.m/A] \quad (B.6)$$

Ahora, reemplazando los valores de K_{med} y T_{med} obtenidos de la gráfica B.1 y de los parámetros recién hallados R_a y K_m obtenemos la expresión para J_{eq} :

$$J_{eq} = \frac{T_{med} K_m}{K_{med} R_a} = \frac{0,0574(0,0494)}{15,973(5,8)} = 3,0562 \times 10^{-5} \text{ } [Kg.m^2] \quad (B.7)$$

La constante de fricción viscosa B_{eq} la obtenemos de la siguiente expresión:

$$B_{eq} = \frac{K_m(1 - K_{med} K_b)}{K_{med} R_a} = 3,9351 \times 10^{-5} \text{ } [N.m/rad/s] \quad (B.8)$$

Finalmente se puede determinar la ganancia del amplificador K_a por medi-

ción directa de la tensión en los terminales de armadura del motor-reductor, y considerando la tensión de entrada u de 23.2 voltios:

$$K_a = \frac{V_{a|MEDIDO}}{u} = 1 \quad (\text{B.9})$$

B.2. Identificación de la Zona Muerta del Motor-Polea

Se procede a identificar la zona muerta del subsistema motor-polea por medio del método gráfico. Debe tenerse en cuenta que este subsistema se considera conformado por el motor-reductor DC y la polea unida al cabezal; la polea está formada a su vez por los dos piñones¹ y la cadena. La cadena está tensada fuertemente con el fin de evitar holguras, dando lugar a una fuerza de fricción en los ejes. Debido a la configuración del mecanismo de posicionamiento del cabezal, la polea y el cabezal están siempre unidos; de modo que en la zona muerta de este subsistema también se incluye el efecto de la fricción estática del cabezal sin carga.

El procedimiento experimental para la identificación consiste en aplicar una entrada de voltaje tipo rampa al motor-polea y medir la velocidad angular de éste. El programa usado para la experiencia en tiempo real es:

- *dzone.c*: Es el programa principal para el experimento (ver apéndice D.2).

Los datos registrados durante el experimento sirven para generar un gráfico que muestra la relación entre el voltaje de alimentación y la velocidad angular del eje. Este gráfico se aprecia en la figura B.2. Puede observarse que la zona muerta en ambos sentidos de giro es aproximadamente de 2.2 V. Este valor es el parámetro fundamental del compensador de zona muerta implementado.

¹Conocidos como *sprockets* en inglés

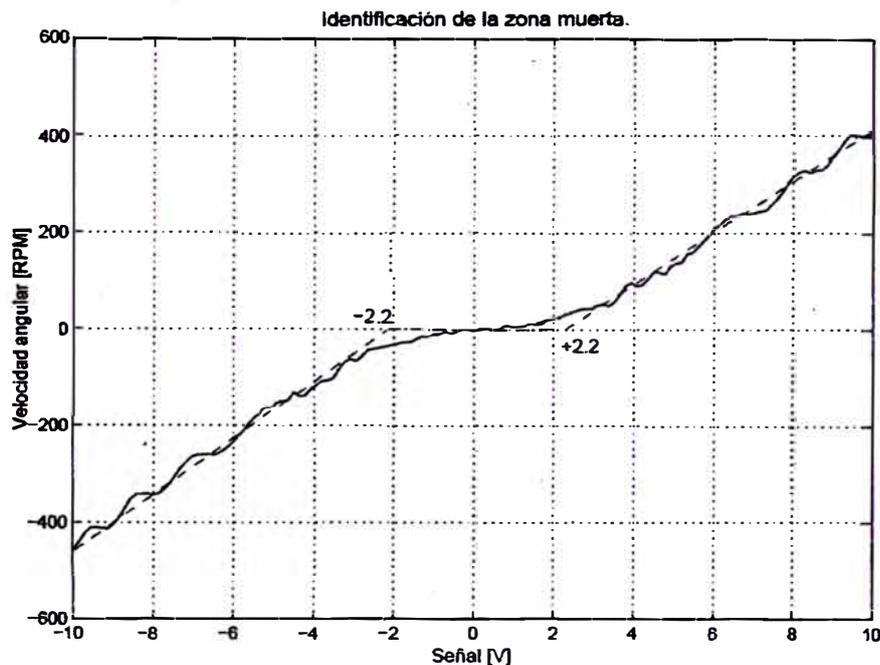


Figura B.2: Gráfico donde se aprecia la zona muerta del subsistema motor-polea.

B.3. Identificación de los Parámetros de Fricción del Cabezal

En el capítulo 3 se consideró la inclusión del efecto de la fricción en el modelo del sistema; la fuerza de fricción se regiría por el modelo de Coulomb expresado en la ecuación 3.49.

En este apéndice, se determina experimentalmente el coeficiente de fricción cinética mediante tres pruebas de la planta ante tres entradas tipo escalón; que tienen amplitudes de 5, 10 y 15 voltios. Mediante estas pruebas se determina la velocidad constante que alcanza el sistema en estado estacionario. Al alcanzar el estado estacionario y ser cero la aceleración, la fuerza presente en el sistema se debe a la fricción cinética y a la resistencia viscosa. Al conocer el modelo matemático aproximado del motor, se puede calcular fácilmente el valor de la fuerza efectiva proporcionada por el eje de salida del motor-reductor mediante la siguiente ecuación:

$$F_k(\dot{\theta}) = \frac{n[(u - K_b\dot{\theta})K_m - B_{eq}K_b\dot{\theta}R_a]}{r_p R_a} \quad (\text{B.10})$$

En la ecuación anterior, la simbología es la misma que la empleada en el capítulo 3. Debe tenerse en cuenta que:

$$\dot{\theta} = \frac{n\dot{x}}{r_p} \quad (\text{B.11})$$

Bajo el modelo considerado y los valores de fuerza de fricción calculados, se obtiene el valor de μ_k mediante la aproximación por el método de mínimos cuadrados de la relación entre la fuerza y la velocidad hallada experimentalmente. En las figuras B.3 y B.4 se muestran los resultados de los experimentos mencionados.

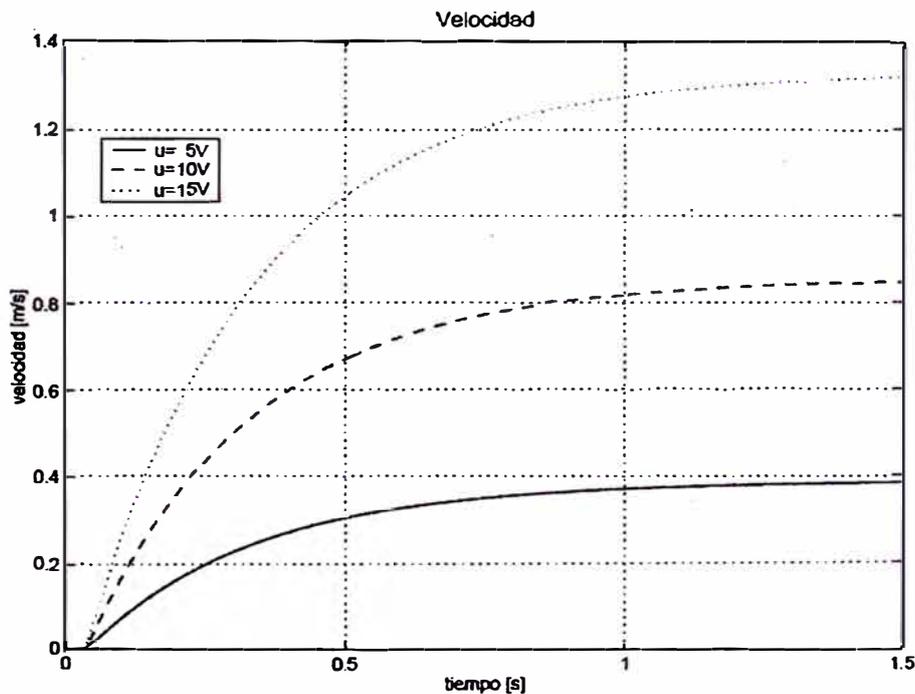


Figura B.3: Cálculo de la fricción: Velocidad del cabezal.

La tabla B.2 es considerada como una guía general. De acuerdo con esta tabla, puede considerarse que el valor del coeficiente de fricción debería ser menor a 0.3, que es el coeficiente de fricción cinética entre superficies metálicas engrasadas.

Los experimentos se efectuaron empleando una pesa de 1Kg sobre el carro

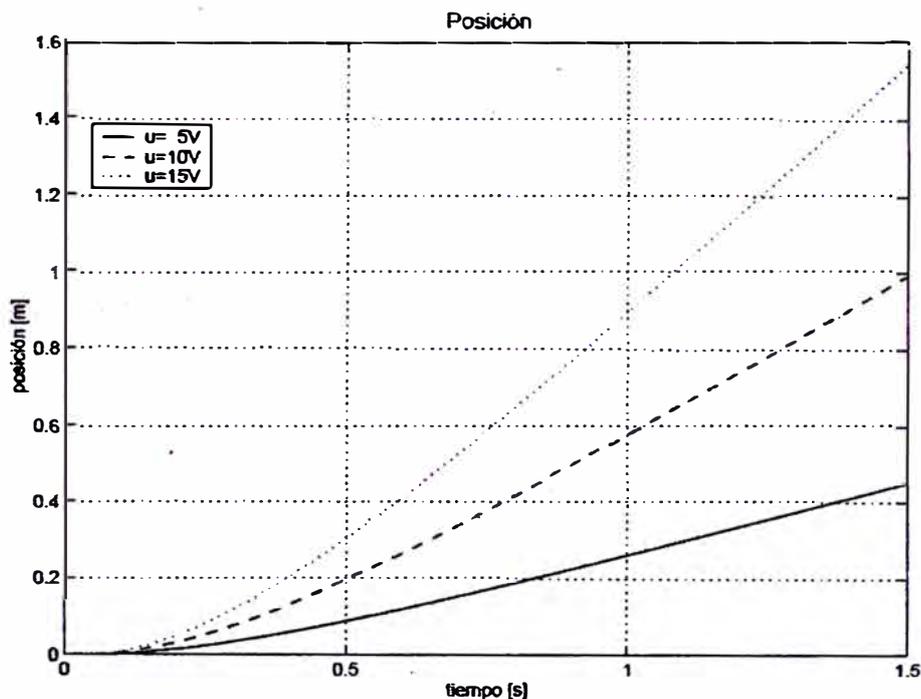


Figura B.4: Cálculo de la fricción: Posición del cabezal.

Material	Condición	Estática	Deslizante
Madera sobre madera	Seco	0.30 - 0.60	0.20 - 0.50
Madera sobre madera	Mojado	0.40 - 0.70	0.20 - 0.40
Madera sobre metal	Seco	0.20 - 0.60	0.30 - 0.50
Madera sobre metal	Mojado	0.50 - 0.70	0.20 - 0.40
Metal sobre metal	Seco	0.20 - 1.10	0.03 - 0.50
Metal sobre metal	Engrasado	0.10 - 0.50	0.03 - 0.30
Cuero sobre metal	Seco	0.30 - 0.50	0.30 - 0.50
Piedra sobre concreto	Seco	0.60 - 1.00	- -
Acero sobre hielo	Seco	0.02 - 0.03	0.01 - 0.02

Cuadro B.2: Coeficientes de Fricción μ .

y dejando la varilla cargada solamente con el soporte para pesas ($0.345Kg$). De este modo, la masa que se desplaza es $M = m_c + m_e + m_v = 2.19Kg$ (ver tabla 3.5). De acuerdo con lo anterior, se calcula la fuerza de fricción cinética en estado estacionario para cada entrada escalón. En la tabla siguiente (B.3) se muestran los valores numéricos derivados de los experimentos.

Con los valores anteriores, se halla la ecuación de la recta que aproxima mejor los datos mediante un ajuste por el método de mínimos cuadrados. Se

Escalón [V]	Fuerza [N]	Velocidad [m/s]
5.0	2.3310	0.3836
10.0	3.7657	0.8459
15.0	5.0884	1.3181

Cuadro B.3: Identificación de la fricción: Valores obtenidos a partir de los experimentos.

obtiene la siguiente ecuación:

$$F_k = 1.2229 + 2,9502 \dot{x}. \quad (\text{B.12})$$

En la figura B.5 se muestra gráficamente la relación entre fuerza y velocidad, mostrando la recta obtenida por el método de mínimos cuadrados.

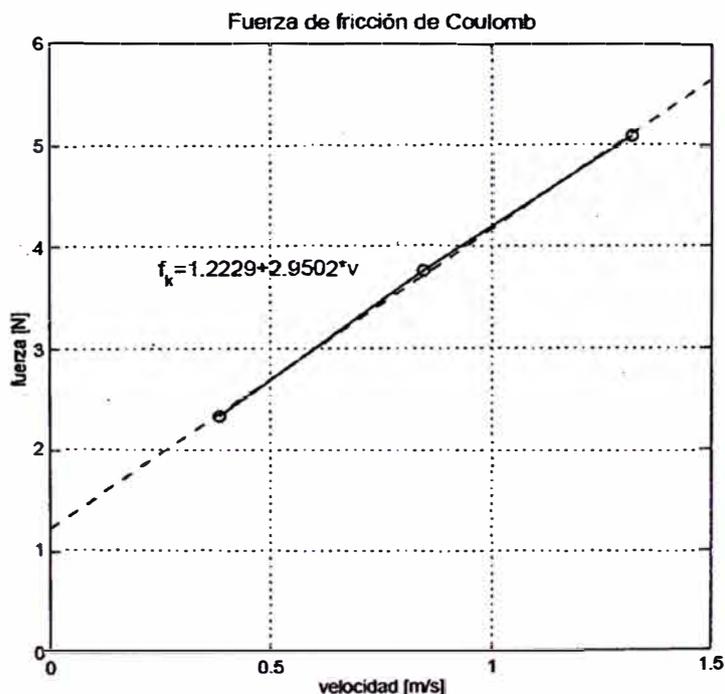


Figura B.5: Fuerza de fricción en función de la velocidad.

Al dividir la componente constante de la fuerza de fricción entre el peso, coincidente con la normal, de las masas que se desplazan linealmente, se obtiene el coeficiente de fricción cinética μ_k :

$$\mu_k = 0.0569. \quad (\text{B.13})$$

B.4. Identificación Fina de los Parámetros de la Planta por un Algoritmo Genético

Los algoritmos genéticos son métodos de optimización estocásticos que prescindan de la derivación de la función objetivo y están basados en los conceptos de la selección natural y de los procesos evolutivos. Estos algoritmos fueron propuestos e investigados inicialmente por John Holland en la Universidad de Michigan en 1975, [27].

Los algoritmos genéticos codifican cada punto de un espacio de parámetros en una cadena de bits binarios llamada *cromosoma*, y a cada punto se le asocia un valor de *aptitud* que, para el caso de la maximización, es usualmente igual a la función objetivo evaluada en ese punto. En lugar de un punto individual, los algoritmos genéticos usualmente toman un conjunto de puntos como una *población*, que evoluciona iterativamente hacia un mejor valor global de aptitud. En cada generación, el algoritmo genético construye una nueva población usando operadores genéticos, tales como el cruce y la mutación; los miembros con una mejor aptitud son más propensos a sobrevivir y participar en las operaciones de cruce. Después de cierto número de generaciones, la población contiene miembros con mejores valores de aptitud; esto es análogo al modelo de Darwin de la evolución por mutación aleatoria y selección natural.

Los principales componentes de los algoritmos genéticos son: los esquemas de codificación, la evaluación de la aptitud, la selección de progenitores, la operación de cruce y la operación de mutación.

Como se ha indicado en el capítulo 3, en 3.7, se ajustarán finamente tres parámetros: B_p , μ_k y d_z . El proceso de optimización consistirá en la reducción ite-

rativa de la diferencia entre la respuesta simulada y la real ante una misma señal de entrada. Experimentalmente se someterá la planta a la excitación debida a una señal de control registrada previamente. Esta señal de control se registró durante la fase de pruebas iniciales del controlador; y se emplea porque da lugar a un movimiento limitado del cabezal y porque se trata de una señal compleja que permite observar la respuesta de la planta con mayor prolijidad. Los componentes del algoritmo genético para el proceso actual de optimización se describen a continuación:

- **Esquemas de codificación:** Un esquema de codificación toma el valor de una variable de entrada y la expresa como una cadena binaria de bits, que viene a ser el cromosoma. La expresión en forma de cadena de bits facilita la manipulación de la información por parte del algoritmo. Para el proceso de optimización actual se formará el cromosoma a través de la concatenación de 3 cadenas de 6 bits cada una. Cada cadena representa a una variable de entrada, que ha sido transformada linealmente desde su dominio real original al rango, en números naturales, comprendido entre 0 y $2^6 - 1$.
- **Evaluación de la aptitud:** Es el procedimiento de evaluación matemática de los cromosomas, mediante una función que indica qué tan aptos son. En este caso se empleará el negativo del error mínimo cuadrático entre los resultados experimentales y los simulados. Al emplear el negativo, se trabajará como un problema de maximización.
- **Selección:** Tras la evaluación, se procede a seleccionar a los mejores individuos para que crucen entre sí, de modo que los descendientes de los mejores individuos reemplacen a los menos aptos. Se puede seleccionar además un grupo élite que se conservará sin modificación para la siguiente generación, de modo que se asegure la permanencia de una buena cepa; esto se conoce

como elitismo. El tamaño de la población será de 20 individuos para este caso, conservándose una élite de 2 unidades.

- **Cruce:** El cruce se realiza tomando una parte de la población para que se crucen partes de sus cromosomas, formando nuevos individuos. La parte de la población que se cruza es proporcional a la tasa de cruce, que en este caso se consigna como 50 %.
- **Mutación:** El procedimiento de cruce explota el potencial de los genes actuales, pero si la población no contiene toda la información codificada necesaria para resolver un problema particular, no podrá obtenerse un resultado satisfactorio. Para la optimización actual, se considera una tasa de mutación del 5 %, que determina qué probabilidad tendrá un bit de cambiar de valor.

El proceso de optimización se realizará mediante un programa de Matlab[®] (ver C.4) que empleará como entrada la respuesta del prototipo construido ante una entrada compleja (similar a las señales de control empleadas en las simulaciones). En la Figura B.6 se muestra las señales de entrada para el proceso de ajuste.

El programa ajustará los tres parámetros mencionados, consiguiendo en 10 generaciones resultados que ya no conseguirán un ajuste significativamente mejor con mayor procesamiento. A continuación se muestran los resultados:

```

Generation 1 : f(0.007097, 0.052097, 3.258065)=-4.204285
Generation 2 : f(0.007097, 0.052097, 3.258065)=-4.204285
Generation 3 : f(0.007097, 0.052097, 3.258065)=-4.204285
Generation 4 : f(0.007097, 0.052097, 3.258065)=-4.204285
Generation 5 : f(0.007097, 0.052097, 3.258065)=-4.204285
Generation 6 : f(0.006581, 0.055968, 3.258065)=-4.094636

```

Generation 7 : $f(0.008645, 0.061129, 2.806452)=-4.063382$

Generation 8 : $f(0.008645, 0.061129, 2.806452)=-4.063382$

Generation 9 : $f(0.008129, 0.052097, 3.032258)=-3.991786$

Generation 10: $f(0.007613, 0.061129, 3.032258)=-3.987285$

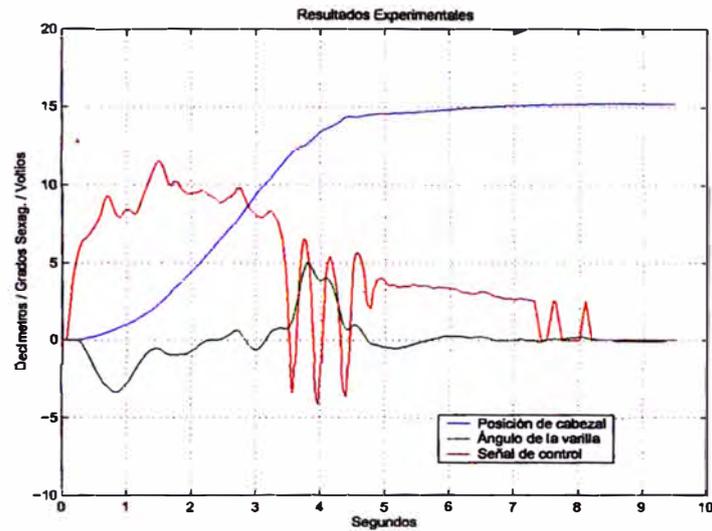


Figura B.6: Respuesta de la planta a una entrada compleja.

En la Figura B.7 se muestra la evolución del ajuste del sistema en función de la evolución de los parámetros, mientras que en la Figura B.8 se muestra la simulación del modelo matemático con los parámetros ajustados por este algoritmo genético.

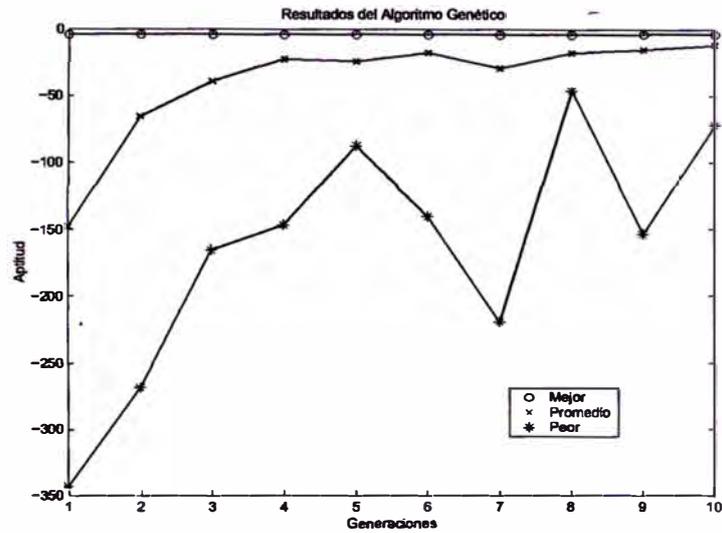


Figura B.7: Evolución inicial de la población.

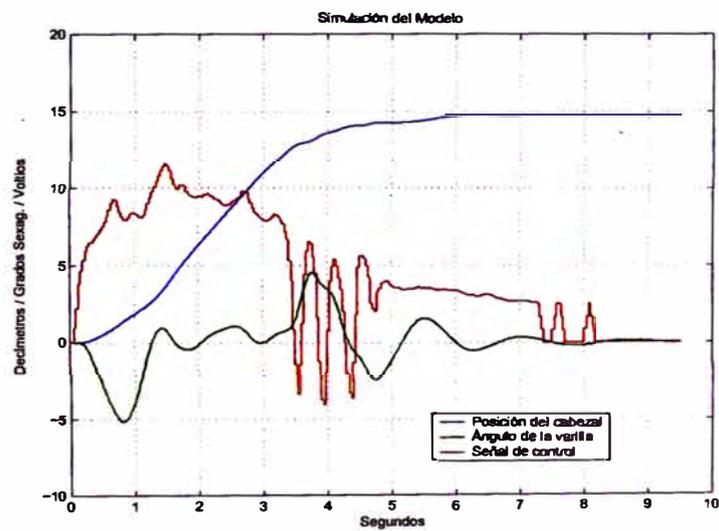


Figura B.8: Simulación del modelo con la entrada registrada.

APENDICE C
PROGRAMAS EN MATLAB®

Se muestran en este apéndice los programas desarrollados en MATLAB® para la simulación, análisis y diseño del sistema de control, así como para la identificación de la planta.

C.1. Programa de Simulación en Tiempo Continuo

A continuación se muestran los programas empleados para la simulación en el dominio continuo del tiempo:

1. `gantrycrane_2FLC_TS_cont_n12.m`: Es el programa principal de simulación.
2. `FLC_TS_d_cont.m`: Es el controlador de la posición.
3. `FLC_TS_a_cont.m`: Es el controlador de la oscilación.
4. `MF_TS_eval_cont.m`: Es el evaluador de funciones de pertenencia.

Programa:

■ `gantrycrane_2FLC_TS_cont_n12.m`

```
%Programa de simulación en tiempo continuo
%Sistema: Prototipo de Grúa-Puente
%Controlador: FLC tipo Takagi-Sugeno
%           Doble sistema de inferencia en paralelo
%           Defuzzificación de orden 0.

clear all; close all;
masa=input('Masa de la carga (Kg)= ');
%sc=30;
%sc=1/fsc;
ts=0.005;
simtime=0:ts:8;
nsamples=size(simtime); nsamples=nsamples(1,2);
Kprop=1.4;
Kcomb=0.80;
KX=0.60; KXd=0.30;
```

```

x(1,:)= [0 0 0 0 0 0];
xdes(1,1)=0;
intv_d=0;
inted=0;
longpend=0.8; Tpend=2.0061*sqrt(longpend);
pos_des=input('Ingrese la posición deseada = ');
Vmax=(pos_des/Tpend)*8/13;
timecontrol=0;u=0;hh=0;v_d=0;v_a=0;inted=0;pp=0;asex=0;ppant=0;asexant=0;
% Generador De Trayectoria:
for i=1:nsamples
    tt=(i-1)*ts;
        if (tt<(Tpend*3/8));
            vdes(i,1)=tt/(Tpend*3/8)*Vmax;
        end
        if ((tt>=(Tpend*3/8))&(tt<(Tpend*13/8)));
            vdes(i,1)=Vmax;
        end
        if (tt>=(Tpend*13/8))
            vdes(i,1)=Vmax*((2*Tpend-tt)/(Tpend*3/8));
        end
        if tt>=2*Tpend
            vdes(i,1)=0;
        end
        xdes(i+1,1)=xdes(i,1)+vdes(i,1)*ts;
% Controlador:
    timecontrol=tt;
    hh=hh+1;
    Kcombe=1+((Kcomb-1)/pos_des)*x(i,1);
    ppant2=ppant;
    ppant=pp;
    pp=(x(i,1));
    vp=(pp-ppant)/ts;
    asexant=asex;
    asex=(x(i,4)*57.2958);
    wsex=(x(i,5)*57.2958);wphh(hh,1)=wsex;
    ed(i,1)=KX*(xdes(i,1)-pp);
        ed_d(i,1)=KXd*(vdes(i,1)-vp);
        v_d=FLC_TS_d_cont(ed(i,1),ed_d(i,1));
    v_a=FLC_TS_a_cont(asex,wsex);
    inted=0;%nted+v_d*tsc;
    u_track(hh,1)=Kcombe*(Kprop*v_d);
    u_damp(hh,1)=Kprop*(1-Kcombe)*v_a;

```

```

        u=u_track(hh,1)+u_damp(hh,1);
    umat(hh,1)=u;
    xhh(hh,1)=x(i,1);
    thetahh(hh,1)=x(i,4);
    thh(hh,1)=tt;
% Compensador De La Zona Muerta:
    dzone=2.44;
    if u<-0.125
        u=u-dzone;
    elseif u>0.125
        u=u+dzone;
    else
        u=0;
    end
    v_dmat(i,1)=v_d; v_amat(i,1)=v_a; intedmat(i,1)=inted;
    dzone=2.44;
    if u<-dzone
        ud=u+dzone;
    elseif u>dzone
        ud=u-dzone;
    else
        ud=0;
    end
    uss(i,1)=u;
% Simulación De La Respuesta Del Sistema:
    xx=x(i,1);
    v=x(i,2);
    a=x(i,3);
    theta=x(i,4);
    w=x(i,5);
    ag=x(i,6);
% Inicio De La Rutina De Respuesta Del Modelo:
    h=ts;
    k1x=h*v;
    k1t=h*w;
    [m1xx,m1tx]=gantrycrane_model(xx,v,a,theta,w,ag,ud,masa);
    m1x=h*m1xx;
    m1t=h*m1tx;
    k2x=h*(v+m1x);
    k2t=h*(w+m1t);
    [m2x,m2t]=gantrycrane_model(xx+k1x,v+m1x,a,theta+k1t,w+m1t,ag,ud,masa);
    m2x=h*m2x;

```

```

    m2t=h*m2t;
    xx = xx + 0.5*(k1x+k2x);
    theta = theta + 0.5*(k1t+k2t);
    v = v + 0.5*(mlx+m2x);
    w = w + 0.5*(mlt+m2t);
    a = mlxx;
    ag = mltx;
    x(i+1,:)= [xx v a theta w ag];
end

figure(1)
subplot(121)
plot(simtime, xdes(1:nsamples,1), 'r', simtime, x(1:nsamples,1), 'b', ...
    'LineWidth',1); grid
legend('X_deseado', 'X_cabezal');
axis([0,8,0, pos_des*1.1]); axis square;
xlabel('tiempo_[s]'); ylabel('posición_[m]');
title('a)_Posición. ');
subplot(122)
plot(simtime, [x(1:nsamples,4)*57.2958], ...
    'LineWidth',1); grid
legend('Theta');
axis([0,8,-10,10]); axis square;
xlabel('tiempo_[s]'); ylabel('desviación angular_[°_sex]');
title('b)_Desviación angular. ');
figure(2)
subplot(121)
plot(simtime, uss, 'r', simtime, u_track, 'b', ...
    'LineWidth',1); grid
legend('u_total', 'u_segguimiento');
xlabel('tiempo_[s]'); ylabel('señal_de_control_[V]');
axis([0,8,-10,15]); axis square;
title('a)_Señal_de_control:_Posición. ');
subplot(122)
plot(simtime, uss, 'r', simtime, u_damp, 'b', ...
    'LineWidth',1); grid
legend('u_total', 'u_amortiguación');
axis([0,8,-10,15]); axis square;
xlabel('tiempo_[s]'); ylabel('señal_de_control_[V]');
title('b)_Señal_de_control:_Desviación angular. ');
% Animación:
if pos_des < 0

```

```

    sim=2;
else
    sim=1;
end
hei=1.6;
le=longpend;
offset=0.2;
ddx=0.03;
xc(1)=0; yc(1)=hei;
xe(1)=0; ye(1)=hei-le;
xc = x(:,1);
yc = hei*ones(nsamples+1,1);
xe = x(:,1) + le*sin(x(:,4));
ye = hei - le*cos(x(:,4));
if sim==1
    figure(4)
    for i=1:10:nsamples
        pause(0.005);
        plot([-offset,2.5],[hei,hei],'-k',[-offset,-offset],[0,hei],...
            '-k',[2.5,2.5],[hei,0],'-k','linewidth',5);
        grid;
        hold on;
        plot([xc(i),xe(i)],[yc(i),ye(i)],'-g','linewidth',2);
        plot([xc(i)-ddx,xc(i)+ddx],[hei,hei],'-r','linewidth',6);
        plot(xe(i),ye(i),'ob','linewidth',4);
        axis([-offset,2.5,-offset,2.5]);
        hold off;
    end
end
if sim==2
    figure(4)
    for i=1:10:nsamples
        pause(0.005);
        plot([offset,-2.5],[hei,hei],'-k',[offset,offset],[0,hei],...
            '-k',[-2.5,-2.5],[hei,0],'-k','linewidth',5);
        grid;
        hold on;
        plot([xc(i),xe(i)],[yc(i),ye(i)],'-g','linewidth',2);
        plot([xc(i)-ddx,xc(i)+ddx],[hei,hei],'-r','linewidth',6);
        plot(xe(i),ye(i),'ob','linewidth',4);
        axis([-2.5,offset,-offset,2.5]);
        hold off;
    end
end

```

end

end

Programa:

■ FLC_TS_d_cont.m

```
function u=flcts_d_1(x,x_d);
% Controlador De Posición Del Cabezal
% Funciones de Pertenencia

% Entradas:
% distance=Matriz de parámetros de las MSFs de Distance
distance=[ -15.00 -0.625 1;
           0.11 -0.500 0;
           0.11 -0.250 0;
           0.11 0.000 0;
           0.11 0.250 0;
           0.11 0.500 0;
           15.00 0.625 1];

sizerdistance=7;
% Distance=Matriz de parámetros de las MSFs de Distance
distance_d=[ -25.000 -0.375 1;
             0.065 -0.300 0;
             0.065 -0.150 0;
             0.065 0.000 0;
             0.065 0.150 0;
             0.065 0.300 0;
             25.000 0.375 1];

sizerdistance_d=7;
% Salida:
% Voltage=vector de voltajes
% MSFs      1  2  3  4  5  6  7
voltage=[-21 -14 -7 0 7 14 21];
% Base de Reglas
% distance-d
%NL NM NS  Z  PS PM PL
RB=...
[1  1  2  2  3  3  4; %NL
 1  2  2  3  3  4  5; %M
```

```

2 2 3 3 4 5 5; %NS
2 3 3 4 5 5 6; %Z distance
3 3 4 5 5 6 6; %PS
3 4 5 5 6 6 7; %PM
4 5 5 6 6 7 7]; %PL
mu_x=MF_TS_eval_cont(x,distance);
mu_x_d=MF_TS_eval_cont(x_d,distance_d);
% Evaluación de la base de reglas y agregación
% utilizando la disyunción probabilística
muv=[1 2 3 4 5 6 7;0 0 0 0 0 0 0]';
for i=1:7;
    for j=1:7;
        k=RB(mu_x(i,1),mu_x_d(j,1));
        and_mu=(mu_x(i,2))*(mu_x_d(j,2));
        muv(k,2)=max(muv(k,2),and_mu);
    end
end
% Defuzzificación singleton
u=0;
for i=1:7
    u=u+(voltage(1,i))*(muv(i,2));
end
u=u./sum(muv(:,2));

```

Programa:

■ FLC_TS_a_cont.m

```

function u=flcts_a_3at1time_max(a,a-d);
% Controlador De La Oscilación De La Carga
% Funciones de Pertenencia

% Entradas:
% angle=Matriz de parámetros de las MSFs de Angle
angle=[-4.0 -2.5 1;
        0.4 -2.0 0;
        0.4 -1.0 0;
        0.4 0.0 0;
        0.4 1.0 0;
        0.4 2.0 0;

```

```

    4.0  2.5  1];
sizerangle=7;
% angle_d=Matriz de parámetros de las MSFs de Angle_d
angle_d=[-0.75  -7.5  1;
         0.10   0.0  0;
         0.75   7.5  1];
sizerangle_d=3;
% Salida:
% voltage=vector de voltajes
% MSFs      1  2  3  4  5  6  7
voltage=[-21 -14 -7 0 7 14 21];
% Base de Reglas
% angle-d
%NL  Z  PL
RB=...
[1  4  1  ;  %NL
 2  4  2  ;  %NM
 3  4  3  ;  %NS
 4  4  4  ;  %Z           angle
 5  4  5  ;  %PS
 6  4  6  ;  %PM
 7  4  7  ]; %PL
mu_a=MF-TS_eval_cont(a, angle);
mu_a_d=MF-TS_eval_cont(a_d, angle_d);
% Evaluación de la base de reglas y agregación
% utilizando la disyunción probabilística
muv=[1 2 3 4 5 6 7;0 0 0 0 0 0 0]';
for i=1:7;
    for j=1:3;
        k=RB(mu_a(i,1), mu_a_d(j,1));
        and_mu=(mu_a(i,2))*(mu_a_d(j,2));
        muv(k,2)=max(muv(k,2), and_mu);
    end
end
% Defuzzificación singleton
u=0;
for i=1:7
    u=u+(voltage(1,i))*(muv(i,2));
end
u=u./sum(muv(:,2));

```

Programa:

- MF_TS_Eval_cont.m

```
function mu=msfts_eval(z,msf);
sizemsf=size(msf); sizemsf=sizemsf(1,1);
i=0;
for j=1:sizemsf;
    i=i+1;
    if msf(j,3)==0; mu(i,1)=j;
        mu(i,2)=exp((-0.5*((z-msf(j,2))/(msf(j,1))).^2));
    else mu(i,1)=j;
        mu(i,2)=1./(1+exp((-msf(j,1)*(z-msf(j,2)))));
    end
end
```

C.2. Programa de Simulación en Tiempo Discreto

A continuación se muestran los programas empleados para la simulación en el dominio discreto del tiempo:

1. gantrycrane_2FLC_TS_disc_n12.m: Es el programa principal de simulación.
2. FLC_TS_d_discrete.m: Es el controlador de la posición.
3. FLC_TS_a_discrete.m: Es el controlador de la oscilación.
4. MF_TS_eval.m: Es el evaluador de funciones de pertenencia.

Programa:

- gantrycrane_2FLC_TS_disc_n12.m

```
%Programa de simulación en tiempo discreto .
%Sistema: Prototipo de Grúa-Puente
%Controlador: FLC tipo Takagi-Sugeno
```

```

%           Doble sistema de inferencia en paralelo
%           Defuzzificación de orden 0.

clear all; close all;
masa=input('Masa de la carga (Kg)= ');
fsc=34;
tsc=1/fsc;
ts=0.004;
simtime=0:ts:8;
nsamples=size(simtime); nsamples=nsamples(1,2);
npulse_encoder1=512; %numero de pulsos del encoder de posición
qep=1; %multiplicador del QEP
npulse_encoder2=1024;
ls7184=4; %multiplicador del t2
dpolea=0.0757;
m2pulse=npulse_encoder1*qep*5.9/(pi*dpolea);
gsex2pulse=npulse_encoder2*ls7184/360;
v2pulse=npulse_encoder1*qep*5.9/(pi*dpolea)*tsc;
wsex2pulse=npulse_encoder2*ls7184/360*tsc;
Kprop=1.40;
Kcomb=0.80;
KX=0.60; KXd=0.30;
x(1,:)= [0 0 0 0 0 0];
xdes(1,1)=0;
intv_d=0;
inted=0;
longpend=0.8; Tpend=2.0061*sqrt(longpend);
pos_des=input('Ingreso la posición deseada = ');
Vmax=(pos_des/Tpend)*8/13;
timecontrol=0; u=0; hh=0; v_d=0; v_a=0; inted=0; pp=0; ap=0; ppant=0; apant=0;
for i=1:nsamples
    tt=(i-1)*ts;
% Generador De Trayectoria:
    if (tt < (Tpend*3/8));
        vdes(i,1)= tt / (Tpend*3/8)*Vmax;
    end
    if ((tt >= (Tpend*3/8)) & (tt < (Tpend*13/8)));
        vdes(i,1)=Vmax;
    end
    if (tt >= (Tpend*13/8))
        vdes(i,1)=Vmax*((2*Tpend-tt)/(Tpend*3/8));
    end
end

```

```

    if tt >= 2*Tpend
        vdes(i,1)=0;
    end
    xdes(i+1,1)=xdes(i,1)+vdes(i,1)*ts;
% Controlador :
    controlar=floor((tt-timecontrol)/tsc);
    if controlar==1;
        timecontrol=tt;
        hh=hh+1;
        controltime(hh)=tt;
        Kcombe=1+((Kcomb-1)/pos_des)*x(i,1);
        ppant2=ppant;
        ppant=pp;
        pp=floor(m2pulse*x(i,1));
        vp=(pp-ppant);
        apant2=apant;
        apant=ap;
        ap=floor(gsex2pulse*x(i,4)*57.2958);
        wp=(ap-apant); wphh(hh,1)=wp;
        ed(i,1)=KX*(xdes(i,1)*m2pulse-pp);
            ed_d(i,1)=KXd*(vdes(i,1)*v2pulse-vp);

            v_d=FLC_TS_d_discrete(ed(i,1),ed_d(i,1));
            v_a=FLC_TS_a_discrete(ap,wp);
        inted=0; %nted+v_d*tsc;
        u_track(hh,1)=Kcombe*(Kprop*v_d);
        u_damp(hh,1)=(1-Kcombe)*v_a;
            u=u_track(hh,1)+u_damp(hh,1);
        umat(hh,1)=u;
        xhh(hh,1)=x(i,1);
        thetahh(hh,1)=x(i,4);
        thh(hh,1)=tt;
% Compensador de la Zona Muerta :
        dzone=2.44;
            if u < -0.125
                u=u-dzone;
            elseif u > 0.125
                u=u+dzone;
            else
                u=0;
            end
    end
end

```

```

v_dmat(i,1)=v_d; v_amat(i,1)=v_a; intedmat(i,1)=inted;
dzone=2.44;
if u<-dzone
    ud=u+dzone;
elseif u>dzone
    ud=u-dzone;
else
    ud=0;
end
uss(i,1)=u;
% Simulación De La Respuesta Del Sistema:
xx=x(i,1);
v=x(i,2);
a=x(i,3);
theta=x(i,4);
w=x(i,5);
ag=x(i,6);
% Inicio De La Rutina De Respuesta Del Modelo:
h=ts;
k1x=h*v;
k1t=h*w;
[m1xx,m1tx]=gantrycrane_model(xx,v,a,theta,w,ag,ud,masa);
m1x=h*m1xx;
m1t=h*m1tx;
k2x=h*(v+m1x);
k2t=h*(w+m1t);
[m2x,m2t]=gantrycrane_model(xx+k1x,v+m1x,a,theta+k1t,w+m1t,ag,ud,masa);
m2x=h*m2x;
m2t=h*m2t;
xx = xx + 0.5*(k1x+k2x);
theta = theta + 0.5*(k1t+k2t);
v = v + 0.5*(m1x+m2x);
w = w + 0.5*(m1t+m2t);
a = m1xx;
ag = m1tx;
x(i+1,:)= [xx v a theta w ag];
end

figure(1)
subplot(121)
plot(simtime,xdes(1:nsamples,1),'r',simtime,x(1:nsamples,1),'b',...
'LineWidth',1); grid

```

```

legend('X_deseado','X_cabezal');
axis([0,8,0,pos_des*1.1]);axis square;
xlabel('tiempo_[s]');ylabel('posición_[m]');
title('a)_Posición. ');
subplot(122)
plot(simtime,[x(1:nsamples,4)*57.2958],...
      'LineWidth',1); grid
legend('Theta');
axis([0,8,-10,10]);axis square;
xlabel('tiempo_[s]');ylabel('desviación_angular_[°_sex]');
title('b)_Desviación_angular. ');
figure(2)
subplot(121)
plot(simtime,uss,'r',controltime,u_track,'b',...
      'LineWidth',0.5); grid
legend('u_total','u_segguimiento');
xlabel('tiempo_[s]');ylabel('señal_de_control_[V]');
axis([0,8,-6,12]);axis square;
title('a)_Señal_de_control:_Posición. ');
subplot(122)
plot(simtime,uss,'r',controltime,u_damp,'b'
      'LineWidth',0.5); grid
legend('u_total','u_amortiguación');
axis([0,8,-6,12]);axis square;
xlabel('tiempo_[s]');ylabel('señal_de_control_[V]');
title('b)_Señal_de_control:_Desviación_angular. ');
% Animación:
if pos_des<0
    sim=2;
else
    sim=1;
end
%Datos para animacion
hei=1.6;
le=longpend;
offset=0.2;
ddx=0.03;
xc(1)=0; yc(1)=hei;
xe(1)=0; ye(1)=hei-le;
xc = x(:,1);
yc = hei*ones(nsamples+1,1);
xe = x(:,1) + le*sin(x(:,4));

```

```

ye = hei - le*cos(x(:,4));
sim=3;
if sim==1
    figure(4)
    for i=1:10:nsamples
        pause(0.005);
        plot([-offset,2.5],[hei,hei],'-k',[-offset,-offset],[0,hei]
'-k',[2.5,2.5],[hei,0],'-k','linewidth',5);
        grid;
        hold on;
        plot([xc(i),xe(i)],[yc(i),ye(i)],'-g','linewidth',2);
        plot([xc(i)-ddx,xc(i)+ddx],[hei,hei],'-r','linewidth',6);
        plot(xe(i),ye(i),'ob','linewidth',4);
        axis([-offset,2.5,-offset,2.5]);
        hold off;
    end
end
if sim==2
    figure(4)
    for i=1:10:nsamples
        pause(0.005);
        plot([offset,-2.5],[hei,hei],'-k',[offset,offset],[0,hei],...
'-k',[-2.5,-2.5],[hei,0],'-k','linewidth',5);
        grid;
        hold on;
        plot([xc(i),xe(i)],[yc(i),ye(i)],'-g','linewidth',2);
        plot([xc(i)-ddx,xc(i)+ddx],[hei,hei],'-r','linewidth',6);
        plot(xe(i),ye(i),'ob','linewidth',4);
        axis([-2.5,offset,-offset,2.5]);
        hold off;
    end
end
end

```

Programa:

■ FLC_TS_d_discrete.m

```

function u=flcts_d-1(x,x-d);
% Controlador De Posición Del Cabezal
% Funciones de Pertenencia

```

```

% Entradas:
% distance=Matriz de parámetros de las MSFs de Distance
distance=[ -12,-7825,1;
           7,-6260, 0;
           7,-3130, 0;
           7,  0, 0;
           7, 3130, 0;
           7, 6260, 0;
           12, 7825, 1];

sizerdistance=7;
% Distance=Matriz de parámetros de las MSFs de Distance
distance_d=[ -599  -157    1;
             369  -125    0;
             369   -63    0;
             369    0    0;
             369   63    0;
             369  125    0;
             599  157    1];

sizerdistance_d=7;
% Salida:
% voltage=vector de voltajes
% MSFs      1  2  3  4  5  6  7
voltage=[-21 -14 -7 0 7 14 21];
% Base de Reglas
% distance-d
%NL NM NS  Z  PS PM PL
RB=...
[1  1  2  2  3  3  4; %NL
 1  2  2  3  3  4  5; %NM
 2  2  3  3  4  5  5; %NS
 2  3  3  4  5  5  6; %Z
 3  3  4  5  5  6  6; %PS
 3  4  5  5  6  6  7; %PM
 4  5  5  6  6  7  7]; %PL
mu_x=MF_TS.eval(x,distance);
mu_x_d=MF_TS.eval(x_d,distance_d);
% Evaluación de la base de reglas y agregación
% utilizando la disyunción probabilística
muv=[1 2 3 4 5 6 7;0 0 0 0 0 0 0]';
for i=1:7;
    for j=1:7;

```

```

    k=RB(mu_x(i,1), mu_x_d(j,1));
    and_mu=(mu_x(i,2))*(mu_x_d(j,2))/10000;
    muv(k,2)=max(muv(k,2), and_mu);
end
end
% Defuzzificación singleton
u=0;
for i=1:7
    u=u+(voltage(1,i))*(muv(i,2));
end
u=u./sum(muv(:,2));

```

Programa:

■ FLC_TS_a_discrete.m

```

function u=flcts_a_3at1time_max(a,a-d);
%CONTROLADOR DE LA OSCILACION DE LA CARGA
% Funciones de Pertenencia

% Entradas:
% angle=Matriz de parámetros de las MSFs de Angle
angle=[-3516, -28, 1;
        2197, -23, 0;
        -2197, -11, 0;
        2197, 0, 0;
        2197, 11, 0;
        2197, 23, 0;
        3516, 28, 1];
sizerangle=7;

% angle_d=Matriz de parámetros de las MSFs de Angle_d
angle_d=[-19775      -3      1;
         263672      0      0;
         19775       3      1];
sizerangle_d=3;

% Salida:
% voltage=vector de voltajes
% MSFs      1  2  3  4  5  6  7

```

```

voltage=[-21 -14 -7 0 7 14 21];
% Base de Reglas
% angle-d
%NL Z PL
RB=...
[1 4 1 ; %NL
 2 4 2 ; %M
 3 4 3 ; %NS
 4 4 4 ; %Z angle
 5 4 5 ; %PS
 6 4 6 ; %M
 7 4 7 ]; %PL
mu_a=MF_TS_eval(a, angle);
mu_a_d=MF_TS_eval(a_d, angle_d);
% Evaluación de la base de reglas y agregación
% utilizando la disyunción probabilística
muv=[1 2 3 4 5 6 7;0 0 0 0 0 0 0]';
for i=1:7;
    for j=1:3;
        k=RB(mu_a(i,1), mu_a_d(j,1));
        and_mu=(mu_a(i,2))*(mu_a_d(j,2))/10000;
        muv(k,2)=max(muv(k,2), and_mu);
    end
end
% Defuzzificación singleton
u=0;
for i=1:7
    u=u+(voltage(1,i))*(muv(i,2));
end
u=u./sum(muv(:,2));

```

Programa:

■ MF_TS_Eval.m

```

function mu=msfts_eval(z,msf);
sizemsf=size(msf); sizemsf=sizemsf(1,1);
i=0;
for j=1:sizemsf;
    i=i+1;

```

```

if msf(j,3)==0; mu(i,1)=j;
    mu(i,2)=10000*exp((-0.5*((z-msf(j,2))*(msf(j,1))).^2)/100000000);
else mu(i,1)=j;
    mu(i,2)=10000./(1+exp((-msf(j,1)*(z-msf(j,2)))/10000));
end
end
end

```

C.3. Programa para la Identificación del Motor-reductor

A continuación se muestra el programa desarrollado para el análisis gráfico de la respuesta del motor-reductor DC.

Programa:

■ servomotordc.m

```

%SERVOMOTORDC.M
% Identificación de parametros del motor DC
%-----
clear all; close all;
frec=100; %frecuencia de muestreo
% Carga de datos del DSP para la identificación del motor
L=load('vel2320.dat');
%Transformación de pulsos a rad/s (salida del motor-reductor)
cteradx=2*pi/(4*512)*frec; L=L.*cteradx;
% Muestra inicial de los datos
kk=60; Vx=L(1:kk,1);
% Definición del registro de tiempo
dt=1/frec;
time=0:dt:dt*(kk-1); time=time';
timex=0:dt/4:dt*(kk-1); timex=timex';
% Mejora digital de la muestra
Vxx=spline(time,Vx,timex);
% Hallando Pto. de Inflexion
n=size(Vxx,1); Vxx1=Vxx(1:n-2); Vxx2=Vxx(3:n);
Axx=[0; Vxx2-Vxx1]*2*frec;
% Punto donde la primera derivada de la velocidad (aceleración)
% llega al máximo, o la segunda derivada se hace cero.
[YY,ii]=max(Axx); % Hallando el indice
valuey=Vxx(ii,1);
valuex=timex(ii,1);

```

```

Kmed=max(Vxx); %Antes de dividir entre el voltaje)
%Se define la tangente en el punto de inflexión y
%se halla la abscisa temporal correspondiente a la intersección
%con la recta horizontal máxima.
tanginfl=(YY)*(timex-valuex)+valuey;
Tmed=interp1q(tanginfl,timex,Kmed); %Queda definido Tmed.
% Gráfica Velocidad vs. Tiempo
figure(1) plot(timex,Vxx,'k','LineWidth',1) ylabel('velocidad
angular_[rad/s]'); xlabel('tiempo_[s]'); title('Análisis
gráfico. '); grid hold on plot(timex,tanginfl,'k');
plot(timex,Kmed*ones(n,1),'—k');
plot([Tmed Tmed],[0 Kmed],'—k');
axis([0 0.35 0 400]);
Vnom=23.2;
Kmed=Kmed/Vnom %Kmed se divide entre el voltaje del escalón.
Tmed %Se muestra Tmed en pantalla.

```

C.4. Programas para la Identificación por un Algoritmo Genético

Para realizar el ajuste fino de parámetros en la identificación del sistema, se utilizaron los siguientes programas:

1. GA_optimization.m: Es el programa principal para la optimización de los parámetros.
2. evalpop.m: Evalúa la aptitud de la población.
3. evalmember.m: Evalúa un individuo de la población.
4. f_gcrane_test_obj.m: Función de evaluación de aptitud, que corre una simulación del modelo de la grúa-puente y compara la simulación con los parámetros a evaluar con los datos experimentales.
5. nextpop.m: Genera la siguiente población, mediante la mutación y cruce de los individuos de la población. Usa el elitismo para conservar los mejores cromosomas.

A continuación se encuentran los listados de los programas mencionados.

Programa:

■ GA_optimization.m

```
close all; clear all;
%PROGRAMA PARA LA OPTIMIZACIÓN DE PARÁMETROS MEDIANTE UN
%ALGORITMO GENÉTICO.

% Configuración del algoritmo
generation_n = 10; % Número de generaciones
popuSize = 20; % Tamaño de la población
xover_rate = 1.0; % Tasa de cruce
mutate_rate = 0.015; % Tasa de mutación
bit_n = 5; % Número de bits para cada variable
obj_fcn = 'f_gcrane_test_obj'; % Función objetivo
var_n = 3; % Número de variables de entrada
% Rangos de las variables de entrada
%      Bp      %nu.k      %lzone
range = [0.004, 0.02; 0.045, 0.065; 1, 8];
% Población inicial aleatoria, en binario
popu = rand(popuSize, bit_n*var_n) > 0.5;
upper = zeros(generation_n, 1); average = zeros(generation_n, 1);
lower = zeros(generation_n, 1);
% Bucle principal para el algoritmo genético
for i = 1:generation_n;
% Se borran los objetos innecesarios
    delete(findobj(0, 'tag', 'member'));
    delete(findobj(0, 'tag', 'individual'));
    delete(findobj(0, 'tag', 'count'));
% Evaluación de la función objetivo para los individuos
    fcn_value = evalpop(popu, bit_n, range, obj_fcn);
% Matrices de la evaluación de la función objetivo
    upper(i) = max(fcn_value);
    average(i) = mean(fcn_value);
    lower(i) = min(fcn_value);
% Muestra de resultados
    [best, index] = max(fcn_value);
    fprintf('Generation_%d:\n', i);
    %f strcmp(obj_fcn, 'peaksfcn'), % obj. function is 'peaksfcn'
    fprintf('f(%f,_%d,_%d)=%f\n', ...
        bit2num(popu(index, 1:bit_n), range(1,:)), ...
```

```

    bit2num(popu(index, bit_n+1:2*bit_n), range(2,:)), ...
    bit2num(popu(index, 2*bit_n+1:3*bit_n), range(3,:)), ...
        best);
    popu = nextpop(popu, fcn_value, xover_rate, mutate_rate);
end
figure; blackbg; x = (1:generation_n)'; plot(x, upper, 'o', x,
average, 'x', x, lower, '*'); hold on; plot(x, [upper average
lower]); hold off; legend('Mejor', 'Promedio', 'Peor');
xlabel('Generaciones'); ylabel('Aptitud');

```

Programa:

■ evalpop.m

```

function fitness = evalpop(population, bit_n, range, fcn)
% Evalpop: Evalúa la aptitud de la población
% population: Matriz binaria de tamaño [popu_n x string_leng]
% bit_n: Número de bits usados para representar una variable de entrada
% range: Rango de las variables de entrada, una matriz de tamaño [var_b x 2]
% fcn: Función objetivo (un string de Matlab)
global count
pop_n = size(population, 1);
fitness = zeros(pop_n, 1);
for count = 1:pop_n,
    fitness(count) = evalmember(population(count, :), bit_n, range, fcn);
end

```

Programa:

■ evalmember.m

```

function out = evalmember(string, bit_n, range, fcn)
% Evalmember: Evaluación de la aptitud de cada individuo
% bit_n: Número de bits de cada variable de entrada
% string: Representación en un string de bits de la variable
% range: Rango de las variables de entarda, una matriz de tamaño [var_n x 2]
% fcn: Función objetivo (un string de Matlab)
var_n = length(string)/bit_n; input = zeros(1, var_n); for i =
1:var_n,
    input(i) = bit2num(string((i-1)*bit_n+1:i*bit_n), range(i, :));
end out = feval(fcn, input);

```

Programa:

■ f_gcrane_test_obj.m

```

function ev=f_gcrane_test_obj(input);
% Función de evaluación para el GA
% Se cargan los datos del experimento
load ufilesave; load posfilesave; load angfilesave;
% Variables a optimizar
Bp=input(1,1); mu_k=input(1,2); dzone=input(1,3);
% Simulación del sistema con la señal de entrada del experimento
% y los parámetros a evaluar
fsc=30; % Frecuencia de muestreo del controlador
tsc=1/fsc;
ts=0.01; % Período de muestreo para la simulación
simtime=0:ts:9.5; % Vector temporal para la simulación completa
controltime=0:tsc:9.5; % Vector temporal para el controlador
nsamples=size(simtime); nsamples=nsamples(1,2);
ncontrol=size(controltime); ncontrol=ncontrol(1,2);
% Inicialización de variables
e_x_square=0; e_a_square=0; x(1,:)= [0 0 0 0 0 0]; longpend=0.83;
Tpend=2.0061*sqrt(longpend);
timecontrol=0; u=0; hh=1; v_d=0; v_a=0; inted=0; pp=0; ap=0; ppant=0; apant=0;
% Bucle de simulación
for i=1:nsamples
    tt=(i-1)*ts;
    if (simtime(i)>=controltime(hh)); % Permite la actualización de
        timecontrol=tt; % a señal de entrada almacenada
        hh=hh+1;
        u=ufilsave(hh); % Se carga la señal de entrada
        umat(hh,1)=u;
        xhh(hh,1)=x(i,1);
        thetahn(hh,1)=x(i,4);
        thh(hh,1)=tt;
        % Evaluación del error cuadrático
        e_x_square=e_x_square+(posfilesave(hh)-x(i,1))^2;
        e_a_square=e_a_square+(angfilesave(hh)-57.2958*x(i,4))^2;
    end
% Zona Muerta
if u<=-dzone
    ud=u+dzone;
elseif u>dzone

```

```

        ud=u-dzone;
    else
        ud=0;
    end
    uss(i,1)=u;
% Simulación de la Respuesta del Sistema
    xx=x(i,1);
    v=x(i,2);
    a=x(i,3);
    theta=x(i,4);
    w=x(i,5);
    ag=x(i,6);
    h=ts;
    k1x=h*v;
    k1t=h*w;
    [mlxx,mltx]=gantrycrane_model_obj(xx,v,a,theta,w,ag,ud,Bp,mu,k);
    mlx=h*mlxx;
    mlt=h*mltx;
    k2x=h*(v+mlx);
    k2t=h*(w+mlt);
    [m2x,m2t]=gantrycrane_model_obj(xx+k1x,v+mlx,a,theta+k1t,w+mlt,ag,ud,Bp,mu,k);
    m2x=h*m2x;
    m2t=h*m2t;
    xx = xx + 0.5*(k1x+k2x);
    theta = theta + 0.5*(k1t+k2t);
    v = v + 0.5*(mlx+m2x);
    w = w + 0.5*(mlt+m2t);
    a = mlxx;
    ag = mltx;
    x(i+1,:)= [xx v a theta w ag];
end
% La salida de la función es el negativo del error cuadrático
% Se define así para usar un algoritmo genético de maximización
ev=-1*(e_x_square+e_a_square/100);

```

Programa:

■ nextpop.m

```

function new_popu = nextpop(popu, fitness, xover_rate, mut_rate)
new_popu = popu; popu_s = size(popu, 1); string_leng = size(popu, 2);

```

```

% Elitismo: Se seleccionan los dos mejores y se conservan para la
% siguiente generación.
tmp_fitness = fitness;
[junk, index1] = max(tmp_fitness); % find the best
tmp_fitness(index1) = min(tmp_fitness);
[junk, index2] = max(tmp_fitness); % find the second best
new_popu([1 2], :) = popu([index1 index2], :);
% Reescalamiento de la aptitud.
fitness = fitness - min(fitness); % keep it positive
total = sum(fitness); if total == 0,
    fprintf('====Warning: converge to a single point====\n');
    fitness = ones(popu_s, 1)/popu_s; % sum is 1
else
    fitness = fitness/sum(fitness); % sum is 1
end cum_prob = cumsum(fitness);
% Selección y cruce
for i = 2:popu_s/2,
    % Se seleccionan dos padres basándose en su evaluación de aptitud.
    tmp = find(cum_prob - rand > 0);
    parent1 = popu(tmp(1), :);
    tmp = find(cum_prob - rand > 0);
    parent2 = popu(tmp(1), :);
    % Se hace el cruce
    if rand < xover_rate,
        % Desarrollo de la operación de cruce
        xover_point = ceil(rand*(string_leng - 1));
        new_popu(i*2-1, :) = ...
            [parent1(1:xover_point) parent2(xover_point+1:string_leng)];
        new_popu(i*2, :) = ...
            [parent2(1:xover_point) parent1(xover_point+1:string_leng)];
    end
end
% Mutación (Se excluyen los miembros de la élite)
mask = rand(popu_s, string_leng) < mut_rate; new_popu =
xor(new_popu, mask);
% Se restaura la élite
new_popu([1 2], :) = popu([index1 index2], :);

```

C.5. Programa del Modelo del Prototipo de Grúa-Puente

A continuación se muestra el código en Matlab para el modelo matemático del sistema Grúa-Puente:

Programa:

■ gantrycrane_model.m

```
function [xpp,thetapp]=gcrane-pendulum_EL(x, xp, xpp, theta, thetap, thetapp, u, masa);
% Modelo Dinámico del Modelo de Grúa-Puente
%-----
% Esta función entrega los valores para las aceleraciones
% de la posición y ángulo de oscilación.
% [xpp, thetapp]=gcrane-pendulum_EL(x, xp, xpp, theta, thetap, thetapp, u, motor)
% donde:
%
%           thetapp(k)=ag;
% thetap(k)=w;
% theta(k)=theta;
%
%           xpp(k)=a;
% xp(k)=v;
% x(k)=x;

% Parametros eléctricos del motor_polea
Ra=5.8; % 2.49; % + rand(1)-0.5; % Ohm
La=2.63; % uH
Ka=1;
Mm=0.644; % Kg (masa del motor)
n=1/5.9;
Km=0.0494; % N.m/A
Jm=3.0562e-5; % Kg.m^2
Kb=0.0494; % V/rad/s
Bm=3.9351e-5;

% Parámetros mecánicos del motor_polea:
Mc=0.67; % Kg (masa del carro)
Mcan=0.405; % Kg (masa de la cadena)
Mv=0.1724; % Kg (masa de las varillas)
Me=masa; % Kg (masa de la carga)
Mp=0.15; % Kg (masa de las poleas)
g=9.81; % m/s^2 (gravedad)
le=0.83; % m (longitud de la carga)
```

```

lv=le/2;          %n      (longitud de la varilla)
rp=0.075/2;      %n      (radio de la polea)
Mn=0.05;         %Kg      (masa de la reduccion)
rn=0.02;         %n
%Constantes Inerciales y coeficientes viscosos
Je=Me*le^2;      %De la carga
Be=0.6e-5;
Jv=Mv*lv^2;     %De la varilla
Bv=7.8e-3;
Jp=2*(Mp+Mcan)*rp^2; %De la polea+cadena
Bp=0.004381;
Jn=0.5*Mn*rn^2; %De la reduccion
Bn=3.7e-4;
Jeq = Jm + n^2*(Jp+Jn);
Beq = Bm + n^2*(Bp+Bn);
uc=0.002;       %Del carro
Fc=uc*g*(Mc+Me+Mv);

% Modelo matemático usando la formulación de Euler-Lagrange

xpp= (1/(n*n*rp*rp*(Mc+Me+Mv)+Jeq))*((Ka*Km*n*rp/Ra)*u +...
n*n*rp*rp*(Me*le+Mv*lv)*(sin(theta)*thetap*thetap -...
cos(theta)*thetapp) - (Beq+(Kb*Km/Ra)-n*n*rp*rp*Fc)*xp);

thetapp=(1/(Je+Jv))*((Me*le+Mv*lv)*(-cos(theta)*xpp -...
g*sin(theta) + sin(theta)*thetap*xp));

```

APÉNDICE D
PROGRAMAS EN C

En este apéndice se encuentran los listados de los programas usados para ejecutar los experimentos de la investigación. Estos programas han sido desarrollados en ANSI C mediante el software *Code Composer*[®] de *Texas Instruments*[®].

D.1. Programa para la Identificación de Parámetros del Motor DC

Con este programa se entrega un voltaje constante (función escalón) al motor DC y se registran los valores de posición y velocidad angular del eje. Estos valores son posteriormente analizados con MATLAB[®].

Programa:

■ parDC.c

```
#include "stdlib.h"
#include "registros.h"
#include "math.h"
int indiceAD=0;
int *temporal;      /* Variable puntero */
int vectorA;       /* Vector de interrupcion Nro.2 EVIVRA */
int pos_ac=0;
int pos_an=0;
int veloc=0;
int uint=0;
int t2conta=0;
int veloc1[1000];
extern void inicio(void);
void main()
{
  inicio(); /* ejecuta las instrucciones en asm como configuración */
  asm("espera:_____nop"); /* loop infinito */
  asm("_____nop");
  asm("_____nop");
  asm("_____b_espera");
}
void c_int2()      /* interrupción*/
{
  int o,p,valido;
  temporal=(int *)EVIVRA;
```

```

vectorA==temporal;
if (vectorA==39)
    {
        temporal=(int *)T2CNT; /*Dato: Encoder del servomotor*/
        t2conta==temporal;
        pos_an=pos_ac;
        pos_ac=t2conta;
        veloc=pos_ac-pos_an;
        if (indiceAD <100)
            {
                uint=0;
            };
        if ((indiceAD >=100)&(indiceAD <1100))
            {
                uint=3125;
                veloc1 [indiceAD -100]=veloc;
            };
        if (indiceAD >=1100)
            {
                uint=0;
            };
        indiceAD++;
    }
else if (vectorA==40)
    {
        temporal=(int *)CMP1;
        *temporal=uint;
    };
}

```

D.2. Programa para la Determinación de la Zona Muerta

Con este programa entrega una rampa de voltaje al motor y se registran los datos de posición y velocidad del subsistema motor-polea. Estos datos son posteriormente analizados con MATLAB®.

Para el registro de los datos se emplearon las siguientes matrices: posmatrix: Posición angular. velomatrix: Velocidad angular. voltmatrix: Voltaje.

Programa:

■ dzone.c

```

#include "stdlib.h" #include "registros.h" #include "math.h"
extern void inicio(void);
int vectorA; int *memo; int *temporal; int t3conta=0;
int indiceAD=0; int pos_ac=0; int pos_an=0; int veloc=0; int
uint=0; int comp_pos=0; int comp_neg=0;
int posmatrix[1000]; int velocmatrix[1000]; int voltmatrix[1000];
void main() { inicio(); /* ejecuta las instrucciones en asm */
asm("espera:_____nop"); /* loop infinito */ asm("_nop"); asm("
nop"); asm("_____b_espera"); }
void c_int2() /* interrupción*/ { int o,p,valido;
memo=(int *)EVIVRA; vectorA=*memo;
if (vectorA==39)
{
memo=(int *)T3CNT; /* QEPS (x4) */
t3conta=*memo;
pos_an=pos_ac;
pos_ac=t3conta;
veloc=pos_ac-pos_an;
/* Prueba para Deadzone */
if (indiceAD<1000)
{
uint=2360; /*indiceAD-589;*/
posmatrix[indiceAD]=pos_ac;
velocmatrix[indiceAD]=veloc;
voltmatrix[indiceAD]=uint;
}
else
{
uint=0;
};
indiceAD++;
}
else if (vectorA==40)
{
/* Single Compare FWM */
if (uint<0)
{
comp_pos=0;
comp_neg=-uint;
temporal=(int *)CMP1;

```

```

    *temporal=comp_pos;
    asm("-----nop");
    asm("-----nop");
    temporal=(int *)CMP2;
    *temporal=comp_neg;
};
if (uint>0)
{
    comp_neg=0;
    comp_pos=uint;
    temporal=(int *)CMP2;
    *temporal=comp_neg;
    asm("-----nop");
    asm("-----nop");
    temporal=(int *)CMP1;
    *temporal=comp_pos;
};
if (uint==0)
{
    comp_neg=0;
    comp_pos=0;
    temporal=(int *)CMP2;
    *temporal=comp_neg;
    asm("-----nop");
    asm("-----nop");
    temporal=(int *)CMP1;
    *temporal=comp_pos;
};
};
}

```

D.3. Programa del Algoritmo de Control

Programa:

- gpc1.c

```

#include "stdlib.h"
#include "registros.h"
#include "math.h"
#include "flcts.h"

```

```

#include "gauss.h"
#include "sigmf.h"

int kk=0; int indice=0; int imat_xdes[360];
int imat_epac[360]; int imat_evac[360];
int imat_aac[360]; int imat_wac[360];
int imat_uint[360]; int imat_x[360];
int imat_wacfil[360];
extern void inicio(void);
void main()
{
inicio (); /* ejecuta las instrucciones en asm como configuración */
asm("espera:_____nop"); /* loop infinito */
asm("_____nop");
asm("_____nop");
asm("_____b_espera");
}
/* Programa de control de posición fuzzy-pd para un servosistema encoder-motor */
void c_int2() /* interrupción*/
{
int o,p,valido; /* cualquier proceso que se necesite en cada interrupción */
temporal=(int *)EVIVRA;
vectorA=*temporal;
if (vectorA==39)
{
if (etapa==1)
{
/* ETAPA 1 */
/* Adquisición de Datos & Combinación */
temporal=(int *)T3CNT; /*Dato: Encoder del motor*/
t3conta=*temporal;
temporal=(int *)T2CNT; /*Dato: Encoder del pendulo*/
t2conta=*temporal;
/* Generador de Trayectoria */
Vmaxp_f=xref2vmax*(float)Xref/1000;
Vmaxp_i=(int)(Vmaxp_f+0.5);
vTpend1_f=Vmaxp_f/Tpend1;
vTpend1_f=vTpend1_f+0.5;
vTpend1_i=(int)vTpend1_f;
if (h<=(Tpend1r))
{
vdes=h*vTpend1_i;
xdes=xdes+vdes;
}
}
}
}

```

```

    };
    if ((h>Tpend1r)&(h<=Tpend2r))
    {
        vdes=(int)Vmaxp_i;
        xdes=xdes+vdes;
    };
    if ((h>Tpend2r)&(h<Tpend3r))
    {
        vdes=Vmaxp_i-(h-Tpend2r)*vTpend1_i;
        xdes=xdes+vdes;
    };
    if (h>=Tpend3r)
    {
        vdes=0;
        Xrefp=(float)Xref*12.9589; /*npulse_encoder1*qep*5.9/(pi*dpolea)*/
        xdes=(int)Xrefp;
    };
    if (indice < 360)
    {
        imat_xdes[indice]=xdes;
        imat_epac[indice]=epac_i;
        imat_evac[indice]=evac_i;
        imat_aac[indice]=aac_i;
        imat_wac[indice]=wac_i;
        imat_uint[indice]=uint;
        imat_x[indice]=x;
        imat_wacfil[indice]=wacfil_i;
    };
    indice++;
    h++;
    /* Formato de Datos */
    xant=x;
    x=t2conta; /* Valor actual (x) */
    th=t3conta; /* Valor actual (theta) */
    /*— carro —*/
    epan_i=epac_i;
    epac_i=(xdes-x);
    evac_i=(vdes-(x-xant));
    /*— pendulo —*/
    aan_i=aac_i;
    aac_i=th;
    wac_i=(aac_i-aan_i);

```

```

    if (indice > 5)
    {
        wacfil_i = wac_i + imat_wac[indice] + imat_wac[indice - 1]
                + imat_wac[indice - 2] + imat_wac[indice - 3]
                + imat_wac[indice - 4] + imat_wac[indice - 5]
                + imat_wac[indice - 6];
        wacfil_i = wacfil_i / 8;
    };
    if (indice <= 5)
    {
        wacfil_i = wac_i;
    };
/* Combinador Lineal */
    un_comb = kc * un_ep + (100 - kc) * un_ea; /* Combinacion Lineal */
    un_comb_i = (int)(un_comb / 100); /* kc = 0.8 (cte. angulo) */
};
/* ETAPA 2 */
    if (etapa == 2)
    {
/* Carro */
/* Evaluacion MSF (m_position) */
        for (k = 0; k < 7; k++)
        {
            if (m_position[k][2] == 0)
            {
                mu_ep[k][0] = k;
                z = (epac_i * 4 - m_position[k][1]) * m_position[k][0];
                if ((z >= -40000) & (z < 40000))
                {
                    af = ((float)z - (-40000)) * 0.0031875; /* 10000000 */
                    i = (int)af;
                    aof = gaussdsp01[0][i];
                };
                if (z < -40000) { aof = 0; };
                if (z >= 40000) { aof = 0; };
                mu_ep[k][1] = aof;
            }
            else
            {
                mu_ep[k][0] = k;
                w = (epac_i * 4 - m_position[k][1]) * m_position[k][0];
                if ((w >= -80000) & (w < 80000))

```

```

    {
        bf=((float)w-(-80000))*0.00159375; /* 20000000; */
        i=(int)bf;
        bof=sigmf DSP01 [0][ i ];
    };
    if (w<-80000){ bof=0;};
    if (w>=80000){ bof=10000;};
    mu_ep [k][1]= bof;
};
};

/* Evaluacion MSF (m_velocity) */
for (k=0;k<7;k++)
if (m_velocity [k][2]==0)
{
    mu_ev [k][0]=k;
    z=(evac_i*0.5-m_velocity [k][1])* m_velocity [k][0];
    if ((z>=-40000)&(z<40000))
    {
        af=((float)z-(-40000))*0.0031875;
        i=(int)af;
        aof=gauss DSP01 [0][ i ];
    };
    if (z<-40000){ aof=0;};
    if (z>=40000){ aof=0;};
    mu_ev [k][1]= aof;
}
else
{
    mu_ev [k][0]=k;
    w=(evac_i*0.5-m_velocity [k][1])* m_velocity [k][0];
    if ((w>=-80000)&(w<80000))
    {
        bf=((float)w-(-80000))*0.00159375;
        i=(int)bf;
        bof=sigmf DSP01 [0][ i ];
    };
    if (w<-80000){ bof=0;};
    if (w>=80000){ bof=10000;};
    mu_ev [k][1]= bof;
};
};

/* Implicacion & Agregacion */

```

```

    mov[0][1]=0;
    mov[1][1]=0;
    mov[2][1]=0;
    mov[3][1]=0;
    mov[4][1]=0;
    mov[5][1]=0;
    mov[6][1]=0;
        for (icont=0;icont<7;icont++)
        {
            for (jcont=0;jcont<7;jcont++)
            {
                rb1=mu_ep[icont][0];
                rb2=mu_ev[jcont][0];
            k=RB_P[rb1][rb2];
            and_mu=((long)mu_ep[icont][1])*((long)mu_ev[jcont][1])/10000;
            if (mov[k][1]<=and_mu)
            {
                mov[k][1]=and_mu;
            };
        };
    };
};

/* Defuzzificacion - Singleton */
    sum_muv=0;
    su=0;
        for (icont=0;icont<7;icont++)
        {
            su=su+((voltagepos[0][icont])*(mov[icont][1]));
            sum_muv=sum_muv+mov[icont][1];
        };
    un_ep=su/sum_muv; /* Accion de control - Posicion */
};

/* ETAPA 3 */
    if (etapa==3)
    {
        /* Pendulo */
        /* Evaluacion MSF (m_angle) */
        for (k=0;k<7;k++)
        {
            if (m_angle[k][2]==0)
            {
                mu_ea[k][0]=k;
                z=(long)((aac_i - m_angle[k][1])*m_angle[k][0]);
            }
        }
    }
};

```

```

if ((z>=-40000)&(z<40000))
{
af=((float)z-(-40000))*0.0031875;
i=(int)af;
aof=gaussdsp01[0][i];
};
if (z<-40000){ aof=0;};
if (z>=40000){ aof=0;};
mu_ea[k][1]=aof;
}
else
{
mu_ea[k][0]=k;
w=(long)((aac_i-m_angle[k][1])*m_angle[k][0]);
if ((w>=-80000)&(w<80000))
{
bf=((float)w-(-80000))*0.00159375;
i=(int)bf;
bof=sigmfdsp01[0][i];
};
if (w<-80000){ bof=0;};
if (w>=80000){ bof=10000;};
mu_ea[k][1]=bof;
};
};
/* Evaluacion MSF (m_velocity) */
for (k=0;k<3;k++)
{
if (m_velocity[k][2]==0)
{
mu_ew[k][0]=k;
z=(long)((wacfil_i - m_velocity[k][1])*m_velocity[k][0]);
if ((z>=-40000)&(z<40000))
{
af=((float)z-(-40000))*0.0031875;
i=(int)af;
aof=gaussdsp01[0][i];
};
if (z<-40000){ aof=0;};
if (z>=40000){ aof=0;};
mu_ew[k][1]=aof;
}
}
}

```

```

else
{
mu_ew[k][0]=k;
w=(long)(( wacfil_i - m_velocity[k][1])* m_velocity[k][0]);
if ((w>=-80000)&(w<80000))
{
bf=((float)w-(-80000))*0.00159375;
i=(int)bf;
bof=sigmf DSP01 [0][ i];
};
if (w<-80000){ bof=0;};
if (w>=80000){ bof=10000;};
mu_ew[k][1]=bof;
};
};

/* Implicacion & Agregacion */
muv[0][1]=0;
muv[1][1]=0;
muv[2][1]=0;
muv[3][1]=0;
muv[4][1]=0;
muv[5][1]=0;
muv[6][1]=0;
for (icont=0;icont<7;icont++)
{
for (jcont=0;jcont<3;jcont++)
{
rb1=mu_ea[icont][0];
rb2=mu_ew[jcont][0];
k=RB_A[rb1][rb2];
and_mu=((long)mu_ea[icont][1])*((long)mu_ew[jcont][1])/10000;
if (muv[k][1]<=and_mu)
{
muv[k][1]=and_mu;
};
};
};

/* Defuzzificacion - Singleton */
sum_muv=0;
su=0;
for (icont=0;icont<7;icont++)
{

```

```

        su=su+((voltageang[0][icont])*(muv[icont][1]));
        sum_muv=sum_muv+muv[icont][1];
    };
    un_ea=su/sum_muv;    /* Accion de control - Angulo */

};

/* ETAPA 4 */
if (etapa==4)
{
    if (un_comb.i >4)
    {
        un_compes=un_comb.i+160;
    };
    if (un_comb < -4)
    {
        un_compes=un_comb.i-160;
    };
    if ((un_comb.i >= -4) & (un_comb.i <= 4))
    {
        un_compes=0;
    };
    etapa=0;
};

etapa++;    /* Periodo de muestreo: 30 Hz */
}

else if (vectorA==40)
{
    /* Formato de variables de salida PWM */
    uint=(int)un_compes;
    /* Bounding outputs */
    if (uint >= 2500)
    {
        uint=2500;
    };
    if (uint <= -2500)
    {
        uint=-2500;
    };
    /* Single Compare PWM */
    if (uint < 0)
    {
        comp_pos=0;
        comp_neg=-uint;
    };
};

```

```

temporal=(int *)CMP1;
*temporal=comp_pos;
asm("-----nop");
asm("-----nop");
temporal=(int *)CMP2;
*temporal=comp_neg;
};
if (uint > 0)
{
comp_neg=0;
comp_pos=uint;
temporal=(int *)CMP2;
*temporal=comp_neg;
asm("-----nop");
asm("-----nop");
temporal=(int *)CMP1;
*temporal=comp_pos;
};
if (uint == 0)
{
comp_neg=0;
comp_pos=0;
temporal=(int *)CMP2;
*temporal=comp_neg;
asm("-----nop");
asm("-----nop");
temporal=(int *)CMP1;
*temporal=comp_pos;
};
}; /* bucle (vector==40) */
} /* main */

```

D.4. Cabeceras del Algoritmo de Control

1. `flcts.h`: Es la cabecera principal, con las funciones de pertenencia y el generador de trayectoria.
2. `gauss.h`: Es la representación en escala de enteros de 0 a 9999 de la función Gaussiana.

3. `sigmf.h`: Es la representación en escala de enteros de 0 a 9999 de la función Sigmoidea.

Programa:

■ `flcts.h`

```

/* HEADER GRUA-PUENTE - FLC (TAKAGI-SUGENO) */
/* Variables internas */
int   *temporal;      /* Variable puntero */
int   vectorA;       /* Vector de interrupcion Nro.2 EVIVRA */
int   t2conta;       /* Variable contador Nro.2 del pendulo */
int   t3conta;       /* Variable contador Nro.3 del carro */
int   k;              /* Contador en bucles */
int   m=0;
int   i;
int   j=0;
int   etapa=1;
float pi=3.14159;
float xref2vmax=146.5678; /*m2pulse*(8/19*(1/Tpend))/fm*/
/* Variables Adquisicion de Datos */
int   epan_i=0; /* Error Anterior posicion */
int   evac_i=0; /* Error Actual velocidad */
int   epac_i=0; /* Error Actual posicion */
int   aan_i=0; /* Error Anterior angulo */
int   wac_i=0; /* Error Actual velocidad angular */
int   aac_i=0; /* Error Actual angulo */
/* Variables Generador de Trayectoria */
int   Xref=-1500; /* Posicion en deseada (mm) */
int   h=1;
int   xdes=0; /* valor deseado */
int   vdes;
int   x=0; /* valor actual */
int   v=0;
int   xant=0;
int   th=0;
int   Vmaxp.i;
int   vTpend1.i;
int   Tpend1r=20; /*round(3/8*Tpend*fm);*/
int   Tpend2r=87; /*round(19/8*Tpend*fm);*/
int   Tpend3r=108; /*round(2*Tpend*fm);*/

```

```

float Tpend1=20.185; /*Tpend=2.0061*sqrt(longpend); Tpend1=3/8*Tpend*fm;*/
float Xrefp; /* en milímetros */
float Vmaxp-f;
float vTpend1-f;
/* Variables MSF (Posicion & Angulo) */
int m_position[7][3]=
{-12, -8013, 1,
 7, -6410, 0,
 7, -3205, 0,
 7, 0, 0,
 7, 3205, 0,
 7, 6410, 0,
12, 8013, 1};
int m_velocity[7][3]=
{ -585, -160, 1,
 360, -128, 0,
 360, -64, 0,
 360, 0, 0,
 360, 64, 0,
 360, 128, 0,
 585, 160, 1};
int m_angle[7][3]=
{-3516, -28, 1,
 2197, -23, 0,
 2197, -11, 0,
 2197, 0, 0,
 2197, 11, 0,
 2197, 23, 0,
 3516, 28, 1};
int m_welocity[3][3]=
{-19775, -3, 1,
 263672, 0, 0,
 19775, 3, 1};
long z,w;
int aof,bof;
long af,bf;
/* Variables Implicacion & Agregacion */
int rb1,rb2;
int icont=0;
int jcont=0;
int mu_ep[7][2];
int mu_ev[7][2];

```

```

int mu_ea[7][2];
int mu_ew[7][2];
int muv[7][2]={0,0,1,0,2,0,3,0,4,0,5,0,6,0};
long and_mu;
/* Defuzzificacion - Singleton */
int RB_P[7][7]={0,0,1,1,2,2,3,0,1,1,2,2,3,4,1,1,2,2,3,4,4,
                1,2,2,3,4,4,5,2,2,3,4,4,5,5,2,3,4,4,5,5,6,3,4,4,5,5,6,6};
int RB_A[7][3]={0,3,0,1,3,1,2,3,2,3,3,3,4,3,4,5,3,5,6,3,6};
int voltagepos[1][7]={-2065, -1377, -688, 0, 688, 1377, 2065};
int voltageang[1][7]={-2065, -1377, -688, 0, 688, 1377, 2065};
int un_ep=0; /*Accion de control (Posicion)*/
int un_ea=0; /*Accion de control (Angulo)*/
long su;
long sum_muv=0;
/* Variables Combinador Lineal */
int kc=85; /* Cte. combinador lineal */
long un_comb=0; /* Accion de Control (Total)*/
int un_comb_i=0;
/* Variables Compensadores No-lineales */
int un_compes=0; /* Accion de Control (Total)
/* Variables PWM */
int uint=0; /* Senal de Control al Servomotor (PWM) */
int comp_pos=0;
int comp_neg=0;

```

Programa:

■ gauss.h

```

/* Tablas: Funcion Gaussiana, Resolucion: 256 */
int gaussdsp01[1][256]=
{3, 4, 4, 5, 5, 6, 7, 8, 9, 10, 11, 13, 14, 16, 18, 20, 22, 25, 27, 30, 34,
38, 42, 46, 51, 57, 63, 69, 77, 84, 93, 102, 112, 123, 135, 148, 162, 178,
194, 212, 231, 252, 274, 298, 323, 351, 381, 412, 446, 482, 520, 561, 605,
651, 700, 753, 808, 866, 928, 993, 1062, 1135, 1211, 1291, 1375, 1463, 1555,
1651, 1751, 1856, 1965, 2078, 2196, 2318, 2445, 2576, 2711, 2851, 2994,
3142, 3294, 3450, 3610, 3774, 3941, 4111, 4285, 4461, 4640, 4822, 5006,
5191, 5378, 5567, 5756, 5946, 6137, 6327, 6516, 6705, 6892, 7078, 7261,
7442, 7620, 7795, 7965, 8132, 8293, 8450, 8601, 8746, 8885, 9017, 9142,
9260, 9370, 9472, 9566, 9651, 9727, 9794, 9852, 9901, 9940, 9969, 9989,
9999, 9999, 9989, 9969, 9940, 9901, 9852, 9794, 9727, 9651, 9566, 9472,
9370, 9260, 9142, 9017, 8885, 8746, 8601, 8450, 8293, 8132, 7965, 7795,

```

7620, 7442, 7261, 7078, 6892, 6705, 6516, 6327, 6137, 5946, 5756, 5567,
 5378, 5191, 5006, 4822, 4640, 4461, 4285, 4111, 3941, 3774, 3610, 3450,
 3294, 3142, 2994, 2851, 2711, 2576, 2445, 2318, 2196, 2078, 1965, 1856,
 1751, 1651, 1555, 1463, 1375, 1291, 1211, 1135, 1062, 993, 928, 866, 808,
 753, 700, 651, 605, 561, 520, 482, 446, 412, 381, 351, 323, 298, 274, 252,
 231, 212, 194, 178, 162, 148, 135, 123, 112, 102, 93, 84, 77, 69, 63, 57,
 51, 46, 42, 38, 34, 30, 27, 25, 22, 20, 18, 16, 14, 13, 11, 10, 9, 8, 7, 6,
 5, 5, 4, 4, 3};

Programa:

■ sigmf.h

/ Tabla: Funcion Sigmoide, Resolucion: 256 */*

int sigmfdsp01[1][256]=

{3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 11, 12, 13,
 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 25, 27, 28, 30, 32, 34, 36, 39,
 41, 44, 47, 50, 53, 56, 60, 64, 68, 72, 77, 82, 87, 92, 98, 105, 111,
 118, 126, 134, 143, 152, 161, 172, 183, 194, 207, 220, 234, 248, 264,
 281, 298, 317, 337, 358, 380, 404, 429, 455, 483, 513, 544, 577, 613,
 650, 689, 730, 774, 820, 868, 919, 973, 1030, 1089, 1151, 1217, 1286,
 1358, 1433, 1512, 1594, 1680, 1769, 1863, 1960, 2060, 2165, 2273, 2385,
 2501, 2621, 2744, 2870, 3001, 3134, 3270, 3410, 3552, 3697, 3845, 3994,
 4146, 4299, 4453, 4609, 4765, 4922, 5078, 5235, 5391, 5547, 5701, 5854,
 6006, 6155, 6303, 6448, 6590, 6730, 6866, 6999, 7130, 7256, 7379, 7499,
 7615, 7727, 7835, 7940, 8040, 8137, 8231, 8320, 8406, 8488, 8567, 8642,
 8714, 8783, 8849, 8911, 8970, 9027, 9081, 9132, 9180, 9226, 9270, 9311,
 9350, 9387, 9423, 9456, 9487, 9517, 9545, 9571, 9596, 9620, 9642, 9663,
 9683, 9702, 9719, 9736, 9752, 9766, 9780, 9793, 9806, 9817, 9828, 9839,
 9848, 9857, 9866, 9874, 9882, 9889, 9895, 9902, 9908, 9913, 9918, 9923,
 9928, 9932, 9936, 9940, 9944, 9947, 9950, 9953, 9956, 9959, 9961, 9964,
 9966, 9968, 9970, 9972, 9973, 9975, 9977, 9978, 9979, 9981, 9982, 9983,
 9984, 9985, 9986, 9987, 9987, 9988, 9989, 9990, 9990, 9991, 9991, 9992,
 9992, 9993, 9993, 9994, 9994, 9994, 9995, 9995, 9995, 9996, 9996, 9996,
 9996, 9997};

APÉNDICE E
PROGRAMAS EN ASSEMBLER

E.1. Configuración DSP: Controlador

En este apéndice se incluye el programa que sirve para configurar el DSP TMS320F240 en los experimentos realizados. Este programa han sido desarrollado en ASSEMBLER del DSP usando el software *Code Composer*[®] de *Texas Instruments*[®].

Con este programa se configura el reloj 3 para usar el encoder del motor, usando el circuito QEP (x4), el reloj 2 se configura para trabajar con el encoder del eje de la varilla (TMRDIR/TMRCLK) y el reloj 1 establece el período de muestreo y la generación de la salida PWM para el motor-reductor DC.

Programa:

■ asmgp1c.asm

```
.include "pathway.h"
.ref _main ;referencia al inicio del programa en C
.def temp
.bss temp,1,1
periodo .set 0a2ch ; valor del periodo (2604)
cmpXini .set 0000h ; valores de los comparadores CMP1/2/3
comparar .set 0068h ; valor del comparador (104)



---


; PROGRAMA PRINCIPAL EN ASSEMBLER


---



.text
.global _inicio
.def _inicio
_inicio:
setc intm
110
;*****
; Configurar Interrupciones
;*****
ldp #0e8h ; DP -> 0x7400 - 0x747f (Event Manager)
lacc EVIFRA ; Limpieza de banderas
sac1 EVIFRA ;
lacc EVIFRB ;
sac1 EVIFRB ;
```

```

lacc EVIFRC ;
sacl EVIFRC ;
splk #0000h, EVIMRC ; Ninguna int de C
splk #0000h, EVIMRB ; Ninguna int de B
splk #0000h, EVIMRA ; Interrupcion del Periodo del T1
; Habilitando interrupciones
ldp #00h ; DP -> 0x0000 - 0x007f
lacc IFR ; Limpieza de banderas del IFR principal
sacl IFR ;
splk #0010h, DMR ; Habilitar solo puerto serial a PC
;*****
; Configurar Pin TICMP/TIFWM
;*****
ldp #0e1h ; DP -> 0x7090 - 0x70ff
splk #0C800h, OCRA ; Permite la salida de TICMP y habilita
; a los pines TMRDIR y TMRCLK.
splk #0030h, OCRB ; Habilita los pines del QEP.
splk #0000h, PADATDIR ; Asegura que ADCIN0 sea entrada
ldp #0e8h ; DP -> 0x7400 - 0x747f (Event Manager)
splk #0000000101101010b, GPTCON ; Configuraci¶on de TICMP
;*****
; Configurar ACTR
;*****
splk #0000011010011001b, ACTR ; GP Timer control
splk #cmpXini, CMPR1
splk #cmpXini, CMPR2
splk #cmpXini, CMPR3
splk #0000001010000111b, COMCON
splk #1000001010000111b, COMCON
;*****
; Configurar GPTimer1
;*****
splk #periodo, T1PR ; setea GPTimer1 periodo
splk #comparar, TICMP ; setea GPTimer1 comparador
splk #00000h, T1CNT ; setea GPTimer1 contador
splk #110101101000010b, T1CON ; Setea Reloj Nro.1
splk #110101101100010b, T1CON ; INICIA Reloj Nro.1
;*****
111
; Configurar GPTimer2 - Encoder con TMRCLK & TMRDIR
;*****
splk #7fffh, T2PR ; setea GPTimer2 periodo

```

```

splk #0000h, T2CNT ; setea GPTimer2 contador
splk #110110000010000b, T2CON ; Setea Reloj Nro.2
splk #1101100001010000b, T2CON ; INICIA Reloj Nro.2
;*****
; Configurar GPTimer 3 - QEP y CAPCON
;*****
ldp #0e8h ; DP -> 0x7400 - 0x747f
splk #2000, T3PR ; set GPTimer3 periodo
splk #1000, T3CMP ; set GPTimer3 comparador
splk #0000h, T3CNT ; set GPTimer3 contador
splk #110110000110010b, T3CON ; Setea Reloj Nro.3
splk #1101100001110010b, T3CON ; INICIA Reloj Nro.3
splk #0110001000000000b, CAPCON
splk #1110001000000000b, CAPCON
;*****
; Configurar Interrupciones
;*****
ldp #0h
splk #0012h,IMR ;int2 asociado al EVIMRA T1INT
ldp #0e8h
splk #0180h, EVIMRA ; Interrupcion del T1
miprogrma:
lacl #27
mar *,arl
clrc intm
ret
PHANTOM
setc intm
ldp #0
splk #0010h,IMR
clrc intm
errorfin
nop
b errorfin
;*****
; PROGRAMAS DE INTERRUPTONES
;*****
PHANTOM1
setc intm
ldp #0
splk #0010h,IMR
ldp #0e0h

```

```
lacl SYSIVR
112
clrc intm
errorfin1
nop
b errorfin1
PHANTOM2 B PHANTOM2
PHANTOM3 B PHANTOM3
PHANTOM4 B PHANTOM4
PHANTOM5 B PHANTOM5
PHANTOM6 B PHANTOM6
PHANTOM7 B PHANTOM7
;*****
; TABLA DE INTERRUPCIONES
;*****
.sect ".vectors" ; Genera una seccion llamada .vectors
.ref _c_int0
.ref _c_int2
RESET B _c_int0 ; 00
INT1 B PHANTOM1 ; 02
INT2 B _c_int2 ; 04
INT3 B PHANTOM3 ; 06
INT4 B PHANTOM4 ; 08
INT5 B PHANTOM5 ; 0A
INT6 B PHANTOM6 ; 0C
.include "rtvecs.h"
```

APÉNDICE F
NORMAS APLICABLES A LAS GRÚAS-PUENTE

F.1. Listado Principales Normas Americanas e Internacionales

Institución	Número	Título	Tópico
OSHA	1910.179	Overhead and Gantry Cranes	Seguridad
ASME	B30.17:2006	Overhead and Gantry Cranes	Diseño-Operación
	B30.24:2008	Container Cranes	Diseño-Operación
	NUM-1-2009	Rules for Construction of Cranes, Monorails, and Hoists	Diseño-Operación Nuclear
ISO	4301-5:1991	Cranes - Classification - Part 5: Overhead travelling and portal bridge cranes	Clasificación
	8566-5:1992	Cranes - Cabins - Part 5: Overhead travelling and portal bridge cranes	Diseño
	9374-5:1991	Cranes - Information to be provided - Part 5: Overhead travelling cranes and portal bridge cranes	Diseño
	10972-5:2006	Cranes - Requirements for mechanisms - Part 5: Bridge and gantry cranes	Diseño
	11660-5:2001	Cranes - Access, guards and restraints - Part 5: Bridge and gantry cranes	Diseño

Cuadro F.1: Listado de Normas Industriales.