

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**Diseño e Implementación de un Sistema de
Control No Lineal Adaptivo Multivariable
Aplicado a un Manipulador Robótico
Translacional de 2DOF**

TESIS

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

Presentada por:

Percey Castillo Casanova

LIMA - PERU

2004

UNIVERSIDAD NACIONAL DE INGENIERIA
Facultad de Ingeniería Eléctrica y Electrónica

Diseño e Implementación de un Sistema de Control No Lineal Adaptivo
Multivariable Aplicado a un Manipulador Robótico Translacional de 2DOF

TESIS

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

Presentada por:

Percey Castillo Casanova

Lima - Perú

Extracto

La presente tesis trata sobre el control de posición y trayectoria de un manipulador robótico translacional multivariable de 2DOF (Two-Degrees-of-Freedoms), el cual consiste de un móvil montado sobre un par de rieles a lo largo del eje horizontal, y un brazo articulado a un servomotor fijo sobre el móvil. Este servomotor maneja el movimiento rotacional del brazo. El otro servomotor es articulado a una de las dos poleas. Estas poleas usa una faja para transmitir la fuerza para manejar el movimiento deslizante del móvil.

Este manipulador robótico no lineal será controlado mediante un Sistema de Control Adaptivo por Modelo Referencial. El objetivo de control es controlar simultáneamente la posición translacional del móvil y la posición angular del brazo, el cual es libre de rotar en cualquier dirección. Los objetivos impuesto en esta tesis son: modelado, simulación e implementación en tiempo real del Sistemas de Control Adaptivo por Modelo de Referencia (MRACS: Model Reference Adaptive Control System).

Un MRACS esta principalmente compuesta de cuatro partes: el modelo referencial, el controlador adaptivo, el proceso a ser controlado, y el mecanismo de adaptación. El modelo referencial, el cual es excitado por una entrada externa

$r(t)$, es un sistema dinámico auxiliar usado para especificar las respuestas deseadas del proceso. Tales respuestas deben ser alcanzadas por el MRACS a pesar de las restricciones generadas por el modelo inexacto de la estructura del modelo referencial así como del modelo del proceso. La ley de adaptación es la relación entre el error e , el cual es la diferencia entre las salidas del modelo referencial y la de los procesos, y su correspondiente vector de parámetros estimados $\hat{\theta}$. El mecanismo de adaptación es un conjunto de bloques interconectados empleados para implementar la ley de adaptación. De hecho, la ley de adaptación es un algoritmo de control usado para modificar los parámetros del controlador adaptivo de tal manera que MRACS permanece estable y el error de seguimiento converja a cero en presencia de cambios en los parámetros del proceso y disturbios externos.

Los resultados de simulación y experimental demuestran que el sistema MRACS diseñado, es capaz de hacer que las salidas controladas del proceso sigan trayectorias de referencia arbitrarias.

El manipulador robótico translacional usado en este trabajo de tesis fue concebido, diseñado y construido por un equipo de investigación en sistemas de control de la Escuela de Graduados de la Facultad de Ingeniería Eléctrica y Electrónica. Los miembros del equipo son el Profesor Arturo Rojas Moreno, y los estudiantes de postgrado Percey Castillo Casanova y Ricardo Rodríguez Bustinza.

Universidad Nacional de Ingeniería
Facultad de Ingeniería Eléctrica y Electrónica

Design and Implementation of a Nonlinear Adaptive Multivariable Control
System Applied to a Translational Robot Manipulator of 2DOF

THESIS

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

by:

Percey Castillo Casanova

Lima - Perú

Abstract

This thesis work deals with the position and trajectory control of a multivariable translational robotic manipulator of 2DOF (2-Degrees-of-Freedoms), which consists of a cart mounted on a pair of rails along the horizontal axes, and a link articulated to a servomotor fixed on the cart. Such a servomotor drives the rotary motion of the link. Another servomotor is articulated to one of two pulleys. This pulley uses a cable to transmit the force that drives the sliding motion of the cart.

Such a nonlinear robotic manipulator will be controlled by means of a model reference adaptive control system. The control objective is to control simultaneously the translational position of the cart and the angular position of the link, which is free of rotating in any direction. The goals imposed in this thesis are: modelling, simulation, and real-time implementation of the model reference adaptive control system (MRACS).

A MRACS is mainly composed of four parts: the reference model, the adaptive controller, the process to be controlled, and the adaptation mechanism. The reference model, which is excited by an external input $r(t)$, is an auxiliary dynamic system used to specify the desired process responses. Such responses should be

achievable for the MRACS despite the restrictions generated by inaccurate modelling of the structure of both, the reference model and the process model. The adaptation law is the relation between the error e , which is the difference between the outputs of the reference model and those of the process, and the corresponding estimated parameter vector $\hat{\theta}$. The adaptation mechanism is a set of interconnected blocks employed to implement the adaptation law. As a matter of fact, the adaptation law is the control algorithm used to modify the parameters of the adaptive controller such that the MRACS remains stable and the tracking error converges to zero in the presence of changing process parameters and external disturbances.

Simulation and experimental results demonstrate that the designed MRACS is capable of making the controlled outputs of the process track arbitrary reference trajectories.

The translational robotic manipulator used in this thesis work was conceived, designed and constructed by the research team for control systems of the Graduate School of the Faculty of Electrical and Electronic Engineering. The members of such a team are Professor Arturo Rojas Moreno, and graduate students Percey Castillo Casanova and Ricardo Rodríguez Bustinza.

Tabla de Contenido

Lista de Figuras	1
Lista de Tablas	3
Capítulo 1.	
Introducción	5
1.1. Formulación del Problema	6
1.2. Perspectiva Histórica	7
1.3. Organización de la Tesis	10
Capítulo 2.	
Modelado de la Planta	12
2.1. Descripción del Sistema	12
2.2. Modelo de Lagrange del Proceso MRTM	13
2.2.1. Ecuaciones de Energía para el Carro	13
2.2.2. Ecuaciones de Energía para el Brazo	14
2.3. Ecuaciones de Lagrange del Proceso MRTM	15
2.3.1. Modelo del Servomotor DC M_1 - Polea	17
2.3.2. Modelo del Servomotor DC M_2 - Brazo	20
2.3.3. Modelo Lagrangiano	21
Capítulo 3.	
Control Adaptivo	24
3.1. Configuración de un MRACS	24
3.2. Estabilidad por el método directo de Lyapunov	25
3.2.1. Conceptos de Estabilidad	26
3.2.2. Funciones de Lyapunov	27
3.2.3. Teoremas de la Estabilidad de Lyapunov	29
3.3. Control Adaptivo para Sistemas No Lineales	32
3.3.1. MRACS para Procesos SISO No Lineales)	33
3.3.2. Ley de Control	33
3.3.3. Ley de Adaptación	34
3.4. MRACS para Procesos MIMO No Lineales	35
3.5. Diseño del Control MRACS para el MRTM	38
3.5.1. Motivación	38
3.5.2. El Modelo de la Planta	39
3.5.3. Diseño del Controlador	40

3.5.4. La Ley de Control	41
3.5.5. La Ley de Adaptación	43
Capítulo 4.	
Implementación en Tiempo Real	44
4.1. El Actuador	44
4.2. El Sensado	51
4.3. Interfaz LABPC+	53
4.4. Software de Control	55
Capítulo 5.	
Simulaciones	57
5.1. Experimento 1: Referencia Constante	57
5.2. Experimento 2: Referencia Senoidal	60
5.3. Animación con la Interface Gráfica GUI	61
Capítulo 6.	
Resultados Experimentales	67
6.1. Experimento 1: Referencia Constante	68
6.2. Experimento 2: Referencia Senoidal	69
Capítulo 7.	
Conclusiones	73
7.1. Conclusiones	73
7.2. Recomendaciones	74
Apéndice A.	
Identificación de Parámetros	76
A.1. Identificación de Parámetros del Servomotor	76
A.2. Curva de la Velocidad para la Identificación	77
A.3. Coeficiente de Fricción del Carro	80
A.4. Coeficiente de Fricción Viscosa del Brazo	82
Apéndice B.	
Diagramas en SIMULINK	88
Apéndice C.	
Programa de Simulación	96
Apéndice D.	
Programas en C	104
Bibliografía.	
	115

Lista de Figuras

1.1.	Control adaptivo con ganancia programada.	8
1.2.	Control adaptivo con lazo cerrado.	9
1.3.	Control adaptivo con modelo referencial (MRACS).	9
1.4.	Control adaptivo con modelo de Identificación (MIAS).	10
2.1.	Manipulador Robótico Traslacional Multivariable (MRTM).	12
2.2.	Subsistema del brazo robótico.	14
2.3.	Subsistema motor - polea.	18
2.4.	Subsistema del torque del brazo.	21
3.1.	Construcción de un MRACS.	24
3.2.	Estabilidad en sistemas autónomos.	27
3.3.	Representación gráfica de la función de Lyapunov.	28
3.4.	la representación gráfica del teorema de inestabilidad de Cēteav.	30
4.1.	Diagramas de bloques de la implementación del sistema.	45
4.2.	Planta real MRTM de 2DOF.	45
4.3.	Implementación del sistema de control real.	46
4.4.	Circuito generador de PWM.	47
4.5.	Diagrama de tiempo del circuito generador de PWM.	49
4.6.	Esquema del circuito de disparo y conmutación.	50
4.7.	Circuito amplificador de potencia de PWM.	52
4.8.	Estructura del codificador óptico rotatorio.	53
4.9.	Sensado de posición.	54
4.10.	Diagrama de flujo del algoritmo de control.	56
5.1.	Posiciones controladas del carro y del brazo para el caso de referencia constante.	58
5.2.	Error de posición Móvil-Brazo para el caso de referencia constante.	58

5.3.	Error de velocidad Móvil–Brazo para el caso de referencia constante.	59
5.4.	Señales de control para el motor 1 y para el motor 2 para el caso de referencia constante.	59
5.5.	Identificación de parámetros para una referencia constante.	60
5.6.	Trayectorias controladas del carro y del brazo para el caso de referencia senoidal.	61
5.7.	Error de posición Móvil–Varilla para el caso de referencia senoidal.	61
5.8.	Error de velocidad Móvil— Varilla para el caso de referencia senoidal.	62
5.9.	Señales de control para el motor 1 y para el motor 2 para el caso de referencia senoidal.	62
5.10.	Identificación de parámetros para el caso de referencia senoidal.	63
5.11.	Panel de salida de la animación del MRTM de 2DOF.	64
6.1.	Posiciones controladas del carro y el brazo para el caso de referencia constante.	68
6.2.	Velocidades controladas del carro y el brazo para el caso de referencia constante.	69
6.3.	Señales de control para el motor 1 y para el motor 2 para el caso de referencia constante.	69
6.4.	Identificación de parámetros para el caso de referencia constante.	70
6.5.	Trayectorias controladas del carro y del brazo para el caso de trayectoria senoidal.	71
6.6.	Velocidades controladas del carro y del brazo para el caso de trayectoria senoidal.	71
6.7.	Señales de control para el carro y el brazo para el caso de trayectoria senoidal.	72
6.8.	Identificación de parámetros para el caso de trayectoria senoidal.	72
A.1.	Curva S.	78
A.2.	Dirección de movimiento.	81
A.3.	Modelo discreto del servomotor DC.	84
A.4.	Simulación del modelo simplificado $L = 0$	85
A.5.	Tensión de salida versus tensión de entrada.	86
A.6.	Experimento de la compensación de la fricción no lineal.	87

B.1. Bloque principal control adaptivo del MRTM de 2DOF.	88
B.2. Bloques de las matrices dinámicas del sistema.	89
B.3. Bloques de errores y referencias.	90
B.4. Bloque de la matriz 'Y'.	91
B.5. Bloque de la Ley de Control U1 y U2.	92
B.6. Bloque variables de estado.	93
B.7. Bloque de estimación de parámetros.	94
B.8. Gráficas de las posiciones del móvil y brazo.	94
B.9. Gráficas de las señales de control u1 y u2 para el móvil y brazo. . . .	95

Lista de Tablas

2.1. Coeficientes de las matrices M , P y d	23
2.2. Parámetros estimados del sistema.	23
4.1. Conmutación de los MOSFET's.	50
A.1. Especificaciones del servomotor DC.	78
A.2. Coeficiente de fricción f	83
A.3. Parámetros del sistema a simular.	85

Capítulo 1

Introducción

Muchos procesos de control son típicamente de naturaleza no lineal [3] y variantes en el tiempo; así mismo en una gran variedad de casos el modelamiento es muy difícil especialmente cuando los parámetros del proceso varían frecuentemente, esto es porque muchos métodos de teoría de control moderno requieren un conocimiento exacto del modelo del proceso y que no puede ser aplicado satisfactoriamente en el contexto industrial. En muchos casos las estrategias de control con parámetros constantes no pueden ser adaptados a los cambios en las condiciones de operación. Las no-linealidades y la naturaleza variante en el tiempo de los procesos es la razón principal para el desarrollo de las *técnicas del control adaptivo*[1].

En nuestro caso particular nos centraremos en desarrollar el Control Adaptivo Multivariable para plantas no lineales basado en un Modelo de Referencia. Nuestro proceso controla el seguimiento del móvil y brazo respecto a una trayectoria deseada; que tiene dos grados libertad. El manipulador robótico transportara pesos de diferentes tamaños; no podemos asumir que los parámetros inerciales de las cargas son conocidos antes que el manipulador los levante y los mueva. Si los controladores con ganancia constante (P, PI, PID, Optimal, ...) son usados y los parámetros con la nueva carga no son conocidos con exactitud, el movimiento del manipulador es inexacto o inestable. El *control adaptivo es la solución al problema* al permitir mover al manipulador cargas de parámetros desconocidos con alta velocidad y gran exactitud.

El control adaptivo no significa un cambio en la estrategia de control, pero si una adaptación en los parámetros del controlador para prevenir un decaimiento en la performance global del sistema en presencia de cambios en la dinámica del proceso. El control adaptivo con modelo referencial expuesto en esta tesis esta compuesto de un término dinámico feed forward y un término PD. El controlador usa la dinámica del sistema para ayudar a converger los parámetros, mientras que el término PD le da estabilidad.

1.1. Formulación del Problema

El problema que se estudia en esta tesis es el control simultaneo de la posición o trayectoria angular del brazo y la posición o trayectoria de traslación del carro, este problema de control es resuelto por medio de la técnica de control adaptivo. Para esto se requiere que el manipulador robótico realice: El posicionamiento o seguimiento de trayectorias de referencia arbitrarias empleando una ley de control acotada. Por consiguiente, el proceso a controlar posee dos entradas y dos salidas, es decir, es un proceso MIMO (Multiple-Input-Multiple-Output).

El manipulador robótico translacional de 2DOF(dos grados de libertad), es un proceso multivariable que posee un brazo articulado al eje del motor DC el cual desarrolla un movimiento rotacional; este motor esta montado sobre un móvil que desarrolla su movimiento de traslación por la acción de dos poleas. Una de las poleas está acoplada al eje de un segundo motor DC, que va a gobernar el movimiento del carro a través de una faja de transmisión.

El sistema manipulador translacional de 2DOF, ha sido construido y diseñado tomando como modelo el sistema de péndulo invertido. La finalidad de construir el sistema es para implementar las diferentes técnicas de control avanzado, en nuestro caso la implementación de un controlador adaptivo basado en un modelo referencial.

Fue necesario modelar la dinámica de la planta mediante las ecuaciones de energía de Lagrange. La exactitud del modelo es fundamental para la etapa de simulación; para la implementación de la estrategia solo basta la estructura de la ecuación dinámica del sistema. El modelo requiere de la identificación de los parámetros del servomotor DC cuyos resultados son presentados en el Apéndice A. La implementación del sistema consigna el uso de la tarjeta de control LABPC+, que se encarga de manejar los datos de entradas digitales de 16 bits de la lectura de los encoders y dos salidas analógicas que son entregadas a las tarjetas interfaces para generar la señal PWM (modulación por ancho de pulso) que se encargan de modular la señal que proviene de la ley de control. Además Se requiere de un amplificador de potencia (circuito H) para manejar los 02 motores DC, cuyo diseño tiene por finalidad manejar una lógica de control de switcheo de los mosfet para polarizar adecuadamente los motores de acuerdo a la señal de salida generada por PWM.

El experimento se ha realizado en el ambiente del laboratorio de proyectos de automatización de la Sección de Postgrado de la Universidad Nacional de Ingeniería, en donde está instalada el proceso a controlar. La parte experimental ha consistido en realizar pruebas para diferentes posiciones así como para trayectorias senoidales a diferentes amplitudes y frecuencias. Los resultados obtenidos de dicho experimento se han guardado en archivos de datos para manejarlos mediante el programa MATLAB para ser graficados y analizados.

1.2. Perspectiva Histórica

En los inicios de la década de los 50, hubo una gran investigación en control adaptivo en relación con el diseño de autopilotos para un alto desempeño en el vuelo de las aeronaves; tales aeronaves operaban bajo un gran rango de velocidad y altitud. Se conocía que los controladores lineales existentes trabajaban bien en un punto; pero no para un régimen de vuelo variado. Se necesitaban controladores mas avanzados. Después de muchos esfuerzos de investigación se encontró que variando la ganancia de los controladores era la solución para los sistemas de control de vuelo. El interés en control adaptivo parcialmente disminuyo porque los problemas de control adaptivo fueron también difíciles de manejar usando las técnicas que estuvieron disponible en ese entonces.

En los inicio de los años 60, hubo un mayor avance en la teoría de control para desarrollar control adaptivo. Las variables de estado y la teoría de estabilidad fueron introducidos. Hubo también un importante resultado en la teoría del control estocástico. La programación dinámica fue introducido por Bellman, incrementando el entendimiento del proceso adaptivo. La contribución fundamental fue hecha por Tsytkin, quien mostró que muchos esquemas para el aprendizaje del control adaptivo podrían ser descritos en un marco común. Hubieron también desarrollos en sistemas de identificación.

El renacimiento del control adaptivo ocurrió en 1970, cuando diferentes diseños de estimación fueron combinados con varios métodos de diseño. Muchas aplicaciones fueron reportados, pero los resultados teóricos fueron muy limitados. En los finales de los años 70 y comienzos de los años 80, se demostró estabilidad en los sistemas adaptivos que aparecían, bajo ciertas asunciones limitantes. Los esfuerzos para unir ideas del control robusto e identificación de sistemas son de

particular relevancia; se inicio una interesante investigación dentro de la robustez del control adaptivo.

En los finales de los años 80 e inicio de los años 90, las investigaciones realizadas en el campo del control adaptivo, dieron una nueva visión acerca de la robustez en la operación de tales sistemas. La teoría del control adaptivo robusto ha sido bien probada y comprendida en 1990. Esencialmente, el diseño del controlador adaptivo robusto involucra modificaciones apropiadas de las leyes convencionales adaptivas.

Como un breve resumen los controles adaptivos los podemos dividir en varias categorías o tipos. La categoría mas general lo divide en sistemas adaptivos de lazo abierto y de lazo cerrado. Los sistemas de lazo abierto son también llamados controladores de ganancia programada (ver figura 1.1). Estos sistemas de lazo abierto se basan en la idea de que los parámetros del controlador cambian en función de algunos valores medidos o estimados z o encontrados experimentalmente.

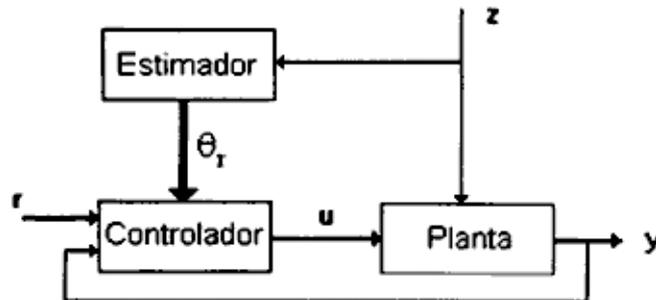


Figura 1.1: Control adaptivo con ganancia programada.

En la figura 1.2, se muestra un sistema de control adaptivo a lazo cerrado. Notar que aparte de lazo tradicional (I), existe otro lazo (II) para estimar los parámetros del sistema, y así poder calcular los parámetros del nuevo controlador.

La estructura MRACS es mostrada en la figura 1.3, su configuración esta basada en la comparación de la señal de salida del sistema y la señal de salida de un modelo. Un mecanismo de adaptación sintoniza en base a este error, los parámetros de este controlador harán que este error tienda a cero. El mecanismo de adaptación esta diseñado para mantener una estabilidad al sistema no lineal, las consideraciones de estabilidad estan basadas en la teoría de la estabilidad de

Lyapunov. MRACS está diseñado para cubrir sistemas que varían en el tiempo. MRACS ha sido usado en muchas situaciones y es uno de las dos "escuelas" en el campo del control adaptivo. Hay muchas otras configuraciones; en años recientes han sido mostrados que MRACS tiene una estrecha conexión con otros tipos de controladores adaptivos.

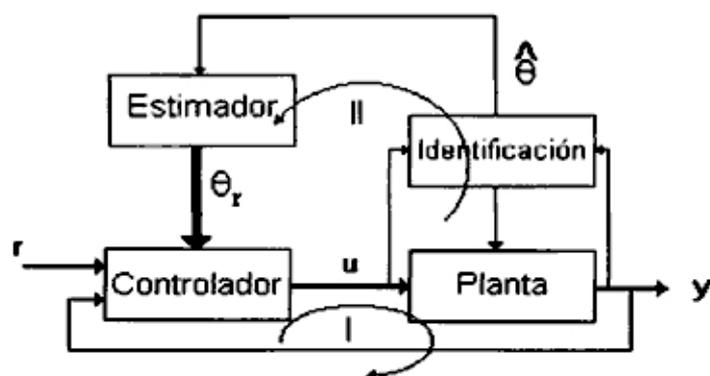


Figura 1.2: Control adaptivo con lazo cerrado.

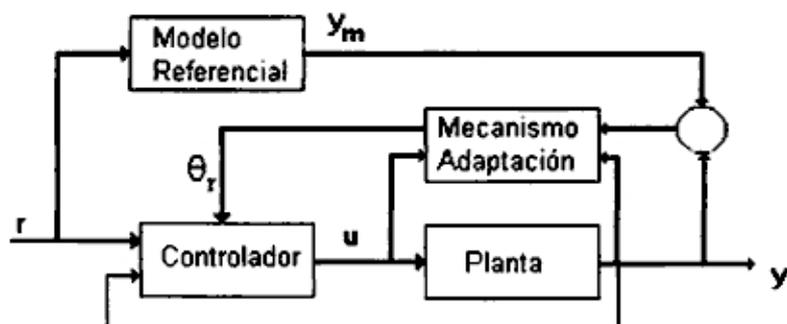


Figura 1.3: Control adaptivo con modelo referencial (MRACS).

El sistema adaptivo con modelo de identificación (MIAS) es el término usado para sistemas adaptivos de la otra escuela. La estructura MIAS se muestra en la figura 1.4, y consiste básicamente en un algoritmo recursivo de identificación que encuentra los parámetros del sistema y diseña un algoritmo de control en base a estos valores estimados para satisfacer la demanda del control. Este control puede

ser descrito en términos del modelo referencial. Este tipo de controladores son llamados "Self Tuning Controllers" (controladores auto sintonizantes) son basados en un marco estocástico y fue originalmente usado para sistemas de parámetros desconocidos pero constantes.

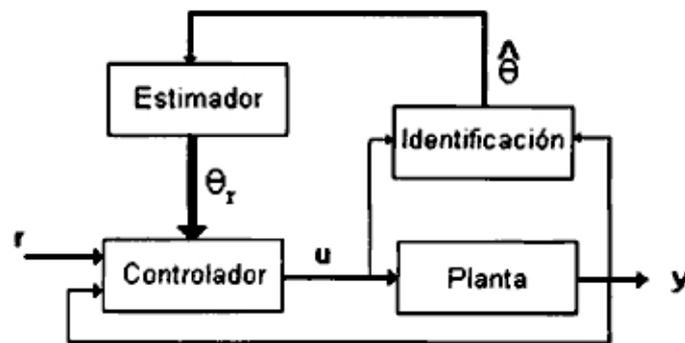


Figura 1.4: Control adaptivo con modelo de Identificación (MIAS).

1.3. Organización de la Tesis

La tesis esta distribuida en 7 Capítulos y 3 Apéndices. Se describe a continuación el contenido de cada uno de ellos:

El capítulo 2 trata sobre el modelado de la planta no lineal usando las ecuaciones de Lagrange, en términos de las energías cinéticas, potenciales y disipativas. En este capítulo también se construye la representación dinámica del proceso en términos de sus variables de estado, para un mejor análisis.

El capítulo 3 formula el problema de diseño del controlador para la planta no lineal. Se utiliza la estructura de la ecuación dinámica del manipulador, se linealiza paramétricamente la planta y se determina la ley de control en base a la dinámica de la planta basada en un modelo referencial. Utilizamos el segundo método de Lyapunov para establecer una superficie deslizante y garantizar la estabilidad y rendimiento del sistema de control no lineal.

El capítulo 4 presenta la descripción general de los equipos y tarjetas de interfaz utilizadas en la implementación del sistema. Los esquemas de los circuitos: generador de PWM, amplificador de potencia y codificador de cuadratura. Tam-

bién se presenta el diagrama de flujo para el software de control.

El capítulo 5 ilustra los resultados obtenidos de las simulaciones en MATLAB. Las simulaciones han sido realizadas para el control de posición y para trayectorias senoidales. Se realizó el diseño de una interfaz gráfica de usuario (GUI) con MATLAB.

El capítulo 6 ilustra los resultados obtenidos de las pruebas experimentales referidas al control de posición y seguimiento de trayectorias senoidales realizadas en el laboratorio, utilizando una tarjeta de adquisición de datos LABPC+. Se ilustran los resultados experimentales comparándolas con las simulaciones realizadas con MATLAB.

El capítulo 7 presenta las conclusiones derivadas de este trabajo y algunas recomendaciones a tomar en cuenta en investigaciones futuras.

Los apéndices contienen una descripción detallada de la identificación de parámetros del servomotor DC, el modelado de la fricción deslizante para el móvil, la fricción de Coulomb para el brazo, los diagramas de bloques del sistema en SIMULINK, el listado de programas en MATLAB y el listado de los programas de tiempo real en C.

Capítulo 2

Modelado de la Planta

Para aplicar la técnica de control adaptivo basado en un modelo referencial, es necesario presentar la dinámica del modelo del manipulador robótico translacional de 2DOF [11]; usando la ecuación dinámica de Lagrange, que nos va a proporcionar la exactitud del modelo, la misma que debe de responder a la exigencia del desarrollo del control que se le aplique al sistema en mención.

2.1. Descripción del Sistema

El sistema del Manipulador Robótico Translacional Multivariable de 2DOF que en adelante lo llamaremos (MRTM de 2DOF) es mostrado en la figura 2.1. M_1 es un servomotor de DC con un mecanismo de reducción y un encoder óptico articulado a una polea de radio R_p . Esta polea usa una faja para transmitir la fuerza F_1 que maneja el movimiento corredizo de un carro de masa M_c montado a un par de rieles a lo largo del eje de x . M_2 también es un servomotor DC con un encoder óptico empleado para manejar el movimiento rotatorio del brazo (libre de girar en ambos sentidos) del MRTM alrededor de un punto del pivote. En la

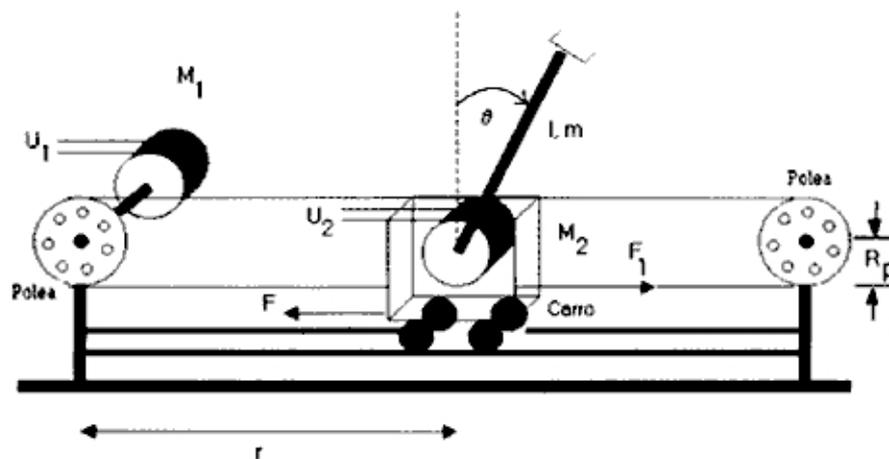


Figura 2.1: Manipulador Robótico Traslacional Multivariable (MRTM).

figura 2.1, θ es la posición angular del brazo de longitud l y masa m , r es la posición del carro, F es la fuerza de fricción opuesta al movimiento del carro.

El proceso de lazo cerrado para ser controlado representa un proceso con dos entradas; los voltajes de control $K_{A1}u_1$ y $K_{A2}u_2$ aplicados a los terminales de armadura de M_1 y M_2 respectivamente, y dos salidas: r y θ . Los parámetros K_{A1} y K_{A2} son las ganancias de amplificación.

2.2. Modelo de Lagrange del Proceso MRTM

Antes de generalizar la ecuación dinámica de Lagrange, se debe de construir las condiciones de energía necesarias para cada componente del sistema. El propósito de este modelo matemático es desarrollar y probar una ley de control que pueda hacer que la trayectoria del carro y la trayectoria del brazo sigan trayectorias de referencia arbitrarias. Nosotros asumimos que las dos componentes de la salida $y(r, \theta)$ son medidas y que la fuerza $u(u_1, u_2)$ es la entrada de control [8].

Nuestro primer objetivo es encontrar la relación dinámica entre $u(u_1, u_2)$ y $y(r, \theta)$. Para esto podemos usar las ecuaciones de Lagrange. Para expresar la energía en este sistema, introducimos las variables de la ecuación de Lagrange. El principio de Lagrange nos permite escribir en forma directa las ecuaciones de energía cinética T , energía potencial U y energía disipada D .

2.2.1. Ecuaciones de Energía para el Carro

El carro es restringido para un movimiento en la dirección horizontal, las ecuaciones dinámicas para el carro comprenden el desarrollo de las energías cinética (T_1), potencial (U_1) y de disipación (D_1) dadas por.

$$T_1 = \frac{1}{2}M_c\dot{r}^2 \quad (2.1)$$

$$U_1 = 0 \quad (2.2)$$

$$D_1 = \frac{1}{2}F_c\dot{r}^2 \quad (2.3)$$

2.2.2. Ecuaciones de Energía para el Brazo

Sea la energía cinética (T_2) del brazo dada por.

$$T_2 = \frac{1}{2}J\dot{\theta}^2 + \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) \quad (2.4)$$

donde J es el momento de inercia referido brazo de masa de m de la figura 2.1.

El desarrollo de la ecuación (2.4) se explica con ayuda de la figura 2.2 que muestra las coordenadas (x,y) en función de los parámetros del sistema del brazo robótico.

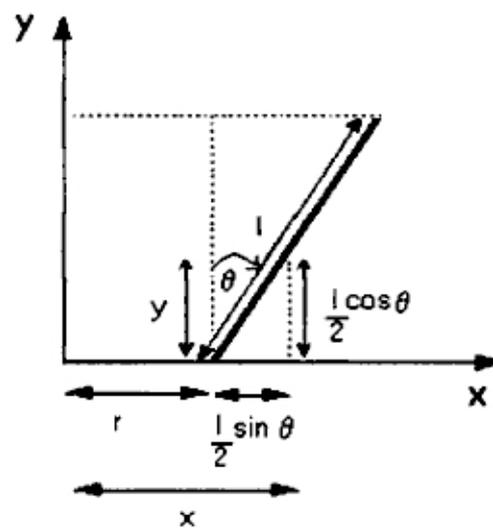


Figura 2.2: Subsistema del brazo robótico.

$$x = r + \frac{l}{2}\sin\theta \quad (2.5)$$

$$y = \frac{l}{2}\cos\theta \quad (2.6)$$

derivando las ecuaciones (2.5) y (2.6).

$$\dot{x} = \dot{r} + \frac{l}{2}\cos(\theta)\dot{\theta} \quad (2.7)$$

$$\dot{y} = -\frac{l}{2}\sin(\theta)\dot{\theta} \quad (2.8)$$

elevando al cuadrado.

$$\dot{x}^2 = \dot{r}^2 + r\dot{\theta}\cos(\theta) + \frac{l^2}{4}\cos^2(\theta)\dot{\theta}^2 \quad (2.9)$$

$$\dot{y}^2 = \frac{l^2}{4}\sin^2(\theta)\dot{\theta}^2 \quad (2.10)$$

sumando las ecuaciones (2.9) y (2.10).

$$\dot{x}^2 + \dot{y}^2 = \dot{r}^2 + r\dot{\theta}\cos(\theta) + \frac{l^2}{4}\dot{\theta}^2 \quad (2.11)$$

reemplazando (2.11) en (2.4).

$$T_2 = \frac{1}{2}J\dot{\theta}^2 + \frac{1}{2}m(\dot{r}^2 + r\dot{\theta}\cos(\theta) + \frac{l^2}{4}\dot{\theta}^2) \quad (2.12)$$

La energía potencial y la energía de disipación del brazo es dada por las siguientes ecuaciones.

$$U_2 = \frac{m\dot{l}g}{2}\cos\theta \quad (2.13)$$

$$D_2 = \frac{1}{2}C\dot{\theta}^2 \quad (2.14)$$

2.3. Ecuaciones de Lagrange del Proceso MRTM

La ecuación de Lagrange generalizada es.

$$\frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + \frac{\partial D}{\partial \dot{q}} = F_q \quad (2.15)$$

donde.

$$L = T - U$$

siendo.

- L : conjunto de energías del sistema.
- D : disipación de energía externa del sistema.
- q : coordenada de variable independiente del sistema.
- \dot{q} : derivada de variable independiente del sistema.

F_q : fuerza externa que actúa en la dirección de q .

Para el sistema MRTM de 2DOF, los términos de las energías individuales componen en su conjunto la totalidad energética del sistema.

Energía cinética del sistema

$$T = T_1 + T_2 = \frac{1}{2}M_c\dot{r}^2 + \frac{1}{2}J\dot{\theta}^2 + \frac{1}{2}m(\dot{r}^2 + \dot{r}l\cos(\theta)\dot{\theta} + \frac{l^2}{4}\dot{\theta}^2)$$

Energía potencial del sistema

$$U = U_1 + U_2 = \frac{mgl}{2}\cos\theta$$

Energía disipada del sistema

$$D = D_1 + D_2 = \frac{1}{2}F_c\dot{r}^2 + \frac{1}{2}C\dot{\theta}^2$$

Podemos generalizar la ecuación de Lagrange para el vector q en función de coordenadas generales (r, θ) del proceso MRTM de 2DOF es formulado de acuerdo con (2.15).

$$\frac{\partial}{\partial t}\left(\frac{\partial T}{\partial \dot{r}}\right) - \frac{\partial T}{\partial r} + \frac{\partial U}{\partial r} + \frac{\partial D}{\partial \dot{r}} = F_1 \quad (2.16)$$

$$\frac{\partial}{\partial t}\left(\frac{\partial T}{\partial \dot{\theta}}\right) - \frac{\partial T}{\partial \theta} + \frac{\partial U}{\partial \theta} + \frac{\partial D}{\partial \dot{\theta}} = T_L \quad (2.17)$$

Para el planteo de las ecuaciones de Lagrange, se observa que el término de energía potencial U no es una función del tiempo derivativo de r . Las ecuaciones dinámicas del Lagrange serán desarrolladas para las variables en donde interviene el sistema.

Desarrollamos las derivadas parciales que indica la ecuación (2.16) para determinar la fuerza F_1 .

$$\frac{\partial T}{\partial \dot{r}} = (M_c + m)\dot{r} + \frac{ml}{2}\cos(\theta)\dot{\theta} \quad (2.18)$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{r}}\right) = (M_c + m)\ddot{r} - \frac{ml}{2}\sin(\theta)\dot{\theta}^2 + \frac{ml}{2}\cos(\theta)\ddot{\theta} \quad (2.19)$$

$$\frac{\partial T}{\partial r} = 0; \quad \frac{\partial U}{\partial r} = 0; \quad \frac{\partial D}{\partial \dot{r}} = F_c\dot{r} \quad (2.20)$$

Obtenemos la primera ecuación parcial del sistema para F_1 .

$$F_1 = (M_c + m)\ddot{r} - \frac{ml}{2}\sin(\theta)\dot{\theta}^2 + \frac{ml}{2}\cos(\theta)\ddot{\theta} + F_c\dot{r} \quad (2.21)$$

Desarrollamos las derivadas parciales que indica la ecuación (2.17) para determinar el torque T_L .

$$\frac{\partial T}{\partial \dot{\theta}} = J\dot{\theta}_1 + \frac{ml}{2}\cos(\theta)\dot{r} + \frac{ml^2}{4}\dot{\theta} \quad (2.22)$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}}\right) = J\ddot{\theta} + \frac{ml}{2}\cos(\theta)\ddot{r} - \frac{ml}{2}\sin(\theta)\dot{\theta}\dot{r} + \frac{ml^2}{4}\ddot{\theta} \quad (2.23)$$

$$\frac{\partial T}{\partial \theta} = -\frac{ml}{2}\sin(\theta)\dot{\theta}\dot{r}; \quad \frac{\partial U}{\partial \theta} = -\frac{ml}{2}g\sin(\theta); \quad \frac{\partial D}{\partial \dot{\theta}_1} = C\dot{\theta} \quad (2.24)$$

Obtenemos la segunda ecuación parcial del sistema para T_L .

$$T_L = \frac{ml}{2}\cos(\theta)\dot{r} + \left(\frac{ml^2}{4} + J\right)\ddot{\theta} + C\dot{\theta} - \frac{mlg}{2}\sin(\theta) \quad (2.25)$$

El modelo Lagrangiano puede ser derivado usando las ecuaciones (2.21) y (2.25). Sin embargo si nosotros estamos interesados en tener como fuerzas de control a los voltajes de armadura u_1 y u_2 en lugar de F_1 y T_L , respectivamente, para esto es necesario modelar los subsistemas eléctricos del MRTM de 2DOF.

2.3.1. Modelo del Servomotor DC M_1 - Polea

La figura 2.3 muestra el servomotor de DC M_1 que es articulado a una polea. Despreciando la inductancia de la armadura L_a , el voltaje de entrada $K_A u_1$ aplicado a la armadura se da por.

$$R_a I_{a1} + V_{b1} = K_A u_1 \quad (2.26)$$

El voltaje de la fuerza electromotriz es proporcional a la velocidad de motor.

$$V_{b1} = K_b \dot{\theta}_{m1} \quad (2.27)$$

La ecuación del torque T_{m1} del servomotor es dado por.

$$T_{m1} = J_m \ddot{\theta}_{m1} + B_m \dot{\theta}_{m1} + T_{g1} \quad (2.28)$$

El torque T_{g2} requiere la transmisión de la polea, y el torque en la polea es directamente proporcional a la fuerza F_1 , luego formulamos.

$$T_{g2} = nT_{g1} = (J_g + J_p) \ddot{\theta}_L + (B_g + B_p) \dot{\theta}_L + F_1 R_p \quad (2.29)$$

donde J_m , J_g y J_p son los momentos de inercia de la armadura, mecanismo de reducción de engranaje y polea, respectivamente, mientras B_m , B_g y B_p son las constantes de fricción de la armadura, mecanismo de reducción de engranaje y polea respectivamente.

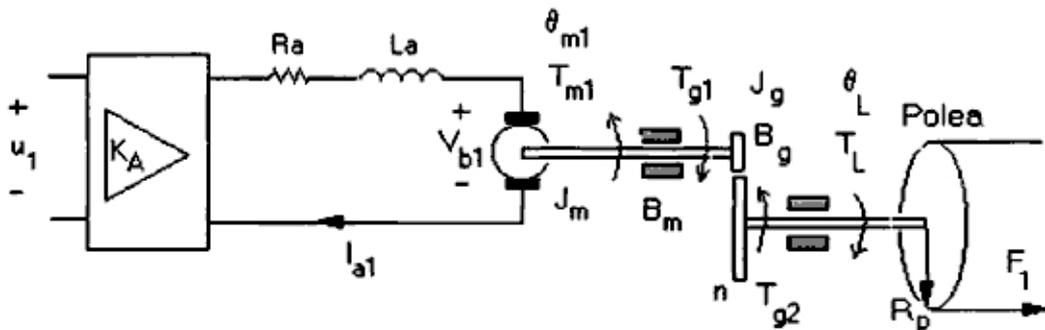


Figura 2.3: Subsistema motor - polea.

Nosotros también hemos usado la relación para la conservación de energía de engranajes $T_{g1} \dot{\theta}_{m1} = T_{g2} \dot{\theta}_L$ con $(n\dot{\theta}_L = \dot{\theta}_{m1})$, de este modo podemos expresar (2.28) como.

$$T_{m1} = J_m n \ddot{\theta}_L + B_m n \dot{\theta}_L + T_{g1} \quad (2.30)$$

despejamos T_{g1} de (2.29) y reemplazamos en (2.30), luego agrupamos términos.

$$nT_{m1} = J_{eq1}\ddot{\theta}_L + B_{eq1}\dot{\theta}_L + F_1R_p \quad (2.31)$$

donde,

$$J_{eq1} = n^2J_m + J_g + J_p$$

$$B_{eq1} = n^2B_m + B_g + B_p$$

el torque del servomotor T_{m1} es proporcional a la corriente I_{a1} .

$$T_{m1} = K_m I_{a1} \quad (2.32)$$

reemplazando (2.31) en (2.32), luego despejamos I_{a1} .

$$I_{a1} = \frac{J_{eq1}}{nK_m}\ddot{\theta}_L + \frac{B_{eq1}}{nK_m}\dot{\theta}_L + \frac{R_p F_1}{nK_m} \quad (2.33)$$

usando las ecuaciones (2.27), (2.33) en (2.26), luego despejamos F_1 .

$$F_1 = \frac{nK_m K_A}{R_a R_p} u_1 - \frac{J_{eq1}}{R_p} \ddot{\theta}_L - \left(\frac{B_{eq1}}{R_p} + \frac{n^2 K_m K_b}{R_a R_p} \right) \dot{\theta}_L \quad (2.34)$$

El movimiento rotacional de la polea puede ser transformado como un movimiento translacional del carro usando la siguiente relación (ver figura 2.3).

$$\theta_L = \frac{r}{R_p} \quad (2.35)$$

derivando (2.35) y sustituyendo en (2.34), obtenemos la ecuación general del subsistema del motor.

$$F_1 = \frac{nK_m K_A}{R_a R_p} u_1 - \frac{J_{eq1}}{R_p^2} \ddot{r} - \left(\frac{B_{eq1}}{R_p^2} + \frac{n^2 K_m K_b}{R_a R_p^2} \right) \dot{r} \quad (2.36)$$

igualando (2.36) con (2.21), se obtiene la primera ecuación dinámica de Lagrange del sistema.

$$u_1 = \frac{R_a R_p}{nK_m K_A} \left\{ (M_c + m + \frac{J_{eq1}}{R_p^2}) \ddot{r} + (F_c + \frac{B_{eq1}}{R_p^2} + \frac{n^2 K_m K_b}{R_a R_p^2}) \dot{r} + \frac{ml}{2} \cos(\theta) \ddot{\theta} + \dots \right. \\ \left. - \frac{ml}{2} \sin(\theta) \dot{\theta}^2 \right\} \quad (2.37)$$

2.3.2. Modelo del Servomotor DC M_2 - Brazo

La figura 2.4 muestra el servomotor de DC M_2 que articula a un punto del pivote localizado en el CG del carro. Teniendo en cuenta que M_1 y M_2 tiene los mismos parámetros y despreciando la inductancia de la armadura L_a , nosotros podemos formular.

$$R_a I_{a2} + V_{b2} = K_A u_2 \quad (2.38)$$

donde.

$$V_{b2} = K_b \dot{\theta}_{m2} \quad (2.39)$$

La relación de desplazamiento angular del eje del motor y eje de carga es.

$$\theta_{m2} = n\theta \quad (2.40)$$

La ecuación de torque T_{m2} del servomotor es (ver figura 2.4).

$$T_{m2} = J_m n \ddot{\theta} + B_m n \dot{\theta} + T_{g1} \quad (2.41)$$

El torque T_{g2} requiere manejar el brazo que se expresa.

$$T_{g2} = nT_{g1} = J_g \ddot{\theta} + B_g \dot{\theta} + T_L \quad (2.42)$$

reemplazamos (2.42) en (2.41), siendo T_L el torque de carga.

$$nT_{m2} = J_{eq2} \ddot{\theta} + B_{eq2} \dot{\theta} + T_L \quad (2.43)$$

donde.

$$J_{eq2} = n^2 J_m + J_g$$

$$B_{eq2} = n^2 B_m + B_g$$

El torque T_{m2} del servomotor es proporcional a la corriente I_{a2} .

$$T_{m2} = K_m I_{a2} \quad (2.44)$$

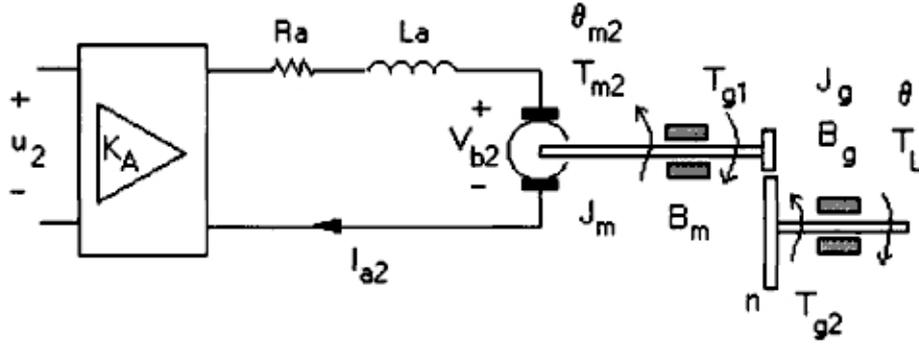


Figura 2.4: Subsistema del torque del brazo.

sustituyendo (2.40), (2.39) en (2.38) y despejando I_{a2} .

$$I_{a2} = \frac{K_A}{R_a} u_2 - \frac{nK_b}{R_a} \dot{\theta} \quad (2.45)$$

sustituyendo (2.45), (2.44) en (2.43) y despejando T_L .

$$T_L = -J_{eq2} \ddot{\theta} - (B_{eq2} + \frac{n^2 K_b K_m}{R_a}) \dot{\theta} + \frac{nK_m K_A}{R_a} u_2 \quad (2.46)$$

igualando (2.46) con (2.25), se obtiene la segunda ecuación dinámica de Lagrange del sistema.

$$u_2 = \frac{R_a}{nK_m K_A} \left\{ \frac{ml}{2} \cos(\theta) \ddot{r} + \left(\frac{ml^2}{4} + J + J_{eq2} \right) \ddot{\theta} + \left(C + B_{eq2} + \frac{n^2 K_m K_b}{R_a} \right) \dot{\theta} + \dots - \frac{mlg}{2} \sin(\theta) \right\} \quad (2.47)$$

2.3.3. Modelo Lagrangiano

La ecuación (2.21) con (2.37) y (2.25) con (2.47) conforman el proceso Lagrangiano del MRTM de 2DOF que toma la forma.

$$u_1 = m_{11} \ddot{r} + p_{11} \dot{r} + m_{12} \cos\theta \ddot{\theta} + p_{12} \sin\theta \dot{\theta}^2 \quad (2.48)$$

$$u_2 = m_{22}\ddot{\theta} + p_{22}\dot{\theta} + m_{21}\cos\theta\ddot{r} + d_{21}\sin\theta \quad (2.49)$$

Las ecuaciones (2.48) y (2.49) pueden ser transformadas en su representación de la matriz como sigue.

$$M(q)\ddot{q} + P(q, \dot{q})\dot{q} + d(q) = u \quad (2.50)$$

donde q y u son los estados:

$$q = \begin{bmatrix} r \\ \theta \end{bmatrix} \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

las matrices M , P y d del modelo del sistema son.

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

La tabla 2.1 muestra los coeficientes de las matrices M , P y d del sistema, mientras que la tabla 2.2 muestra los parámetros del servomotor DC. El método de identificación de los parámetros del motor y la obtención de las constantes de fricción del carro F_c y del brazo C se explican en el apéndice A. Se han considerado las siguientes equivalencias para la momento de inercia y la fricción viscosa $J_{eq} = J_{eq1} = J_{eq2}$ y $B_{eq} = B_{eq1} = B_{eq2}$, debido a que reflejan valores similares, tal como lo demuestran los resultados experimentales obtenidos en la identificación de los parámetros para los dos servomotores DC de igual modelo.

Cuadro 2.1: Coeficientes de las matrices M , P y d .

$m_{11} = (\frac{R_a R_p}{n K_A K_m})(M_c + m + \frac{J_{eq}}{R_p^2})$	$p_{11} = (\frac{R_a R_p}{n K_A K_m})(F_c + \frac{B_{eq}}{R_p^2} + \frac{n^2 K_m K_b}{R_a R_p^2})$
$m_{12} = (\frac{R_a R_p}{n K_A K_m}) \frac{ml}{2}$	$p_{12} = -(\frac{R_a R_p}{n K_A K_m}) \frac{ml}{2}$
$m_{21} = (\frac{R_a}{n K_A K_m}) \frac{ml}{2}$	$p_{21} = 0$
$m_{22} = (\frac{R_a}{n K_A K_m})(\frac{ml^2}{4} + J + J_{eq})$	$p_{22} = (\frac{R_a}{n K_A K_m})(C + B_{eq} + \frac{n^2 K_m K_b}{R_a})$
$d_{12} = 0$	$d_{21} = -(\frac{R_a}{n K_A K_m}) \frac{mlg}{2}$

Cuadro 2.2: Parámetros estimados del sistema.

Símbolo	Descripción	Valor	Unidades
M_c	masa del carro + masa de la polea	1.6492	Kg
m	masa del brazo	0.0519	Kg
l	longitud del brazo	0.30	m
J	momento de inercia del brazo robótico	$2,38e^{-3}$	Kgm^2
J_{eq}	momento de inercia equivalente	3.0551×10^{-4}	Kgm^2
B_{eq}	fricción viscosa equivalente	5.3848×10^{-4}	$Nm/rad/s$
C	coeficiente de fricción viscosa brazo	$9,5e^{-4}$	Kgm^2/seg
F_c	coeficiente de fricción del carro	2.81	Kg/seg
g	aceleración de la gravedad	9.81	$mseg^{-2}$
K_b	constante de fuerza contraelectromotriz	0.0565	$V/rad/s$
K_m	constante de torque de motor	0.0421	Nm/A
R_p	radio de la polea	0.048	m
R_a	resistencia de armadura	5.3	Ω
n	factor de reducción	12.5	
K_A	constante de ganancia de amplificación	5.5	

Capítulo 3

Control Adaptivo

3.1. Configuración de un MRACS

La figura 3.1, muestra la configuración de un sistema de control adaptivo con modelo de referencia (MRACS) empleado en una variedad de aplicaciones. El MRACS esta básicamente compuesta por 4 partes: *el modelo referencial, controlador, planta, y el mecanismo o ley de adaptación*. El esquema básico del MRACS en consideración es llamado un *MRACS paralelo* debido a la relativa ubicación del modelo referencial con respecto al proceso.

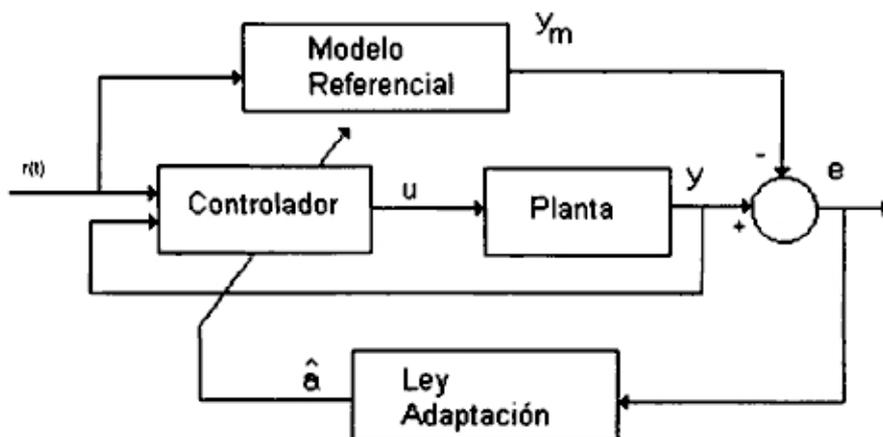


Figura 3.1: Construcción de un MRACS.

El modelo referencial, el cual es excitado por una entrada externa $r(t)$, es un sistema dinámico auxiliar usado para especificar las respuestas de proceso deseado, tales respuestas deben ser alcanzados por el MRACS a pesar de las restricciones generadas por el modelo inexacto de la estructura de ambos, el modelo referencial y el modelo del proceso.

La ley de adaptación es la relación entre el error e , el cual es la diferencia entre las salidas del modelo referencial y las del proceso y el correspondiente

vector $\hat{\mathbf{a}}$ de parámetros estimados. El mecanismo de adaptación es un conjunto de bloques interconectados empleados para implementar la ley de control. La ley de adaptación usada para ajustar los parámetros en la ley de control. El mecanismo de adaptación busca parámetros de tal manera que la respuesta de la planta sea igual al modelo. El objetivo de la adaptación es hacer que garantice que el control permanezca estable y el error de seguimiento (la señal controlada siga la trayectoria deseada) converga a cero según cambie los parámetros del proceso y los disturbios externos.

El proceso, se asume que la planta no lineal tiene una estructura de sus ecuaciones dinámicas conocidas, aunque sus parámetros no sean conocidos. De aquí su descripción podría ser imprecisa; esto es, el modelo dinámico del proceso podría presentar cualquier incertidumbre paramétrica en su estructura o en su representación de su dinámica no modelada. Desde que la descripción de los procesos contienen incertidumbres, el control adaptivo en general puede ser considerado una aproximación particular al control robusto.

El controlador es generalmente parametrizable por un número de parámetros ajustables; debe ser capaz de realizar un perfecto seguimiento en presencia de parámetros inciertos. Esto es, cuando los parámetros son conocidos o desconocidos, los parámetros del controlador necesitan ser ajustados por un mecanismo de adaptación en orden de hacer cero el error e . El *método directo de Lyapunov* es empleado para encontrar un diseño MRACS que garantice la convergencia global de seguimiento. Si la ley de control (algoritmo de control) posee una estructura lineal con parámetros ajustables, es conocido como linealmente parametrizable.

3.2. Estabilidad por el método directo de Lyapunov

A. M. Lyapunov contribuyó con la estabilidad de sistemas descritas por las ecuaciones dinámicas diferenciales usando dos métodos distintos. El **primer método** analiza el comportamiento estable de una solución explícita de un sistema y es solo aplicable a algunos casos. El **segundo método o método directo** es más general y potente porque aquello no requiere del conocimiento de la solución de la descripción del sistema.

3.2.1. Conceptos de Estabilidad

Un sistema no lineal de la forma:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \quad (3.1)$$

Se dice que es no-autónomo si \mathbf{f} depende del tiempo, es decir, si \mathbf{f} posee parámetros que varían en el tiempo. De aquí un sistema autónomo puede ser descrito por $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$. Trayectorias de estado para un sistema autónomo son independientes del tiempo inicial, mientras para un sistema no autónomo generalmente no lo son.

Un *estado de equilibrio* o punto de equilibrio \mathbf{x}_e (vector constante) de un sistema autónomo puede ser descrito como.

$$\mathbf{0} = \mathbf{f}(\mathbf{x}_e) \quad (3.2)$$

Puesto que trataremos con la estabilidad en el origen de los sistemas básicos.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \quad \mathbf{x}(\mathbf{0}) = \mathbf{0} \quad (3.3)$$

entonces necesitamos hacer una cuantas simples asunciones en relación con la figura 3.2, denotaremos por $B(R)$ la región esférica $\|\mathbf{x}\| < R$ y por $S(R)$ la esfera $\|\mathbf{x}\| = R$. La región anular esférica cerrada $r \leq \|\mathbf{x}\| \leq R$ será denotado por B_r^R . Nosotros asumiremos que en cierta región esférica $\Omega : \|\mathbf{x}\| < B(R)$ las derivadas parciales $\partial x_i / \partial x_j$ todas existen y son continuas en Ω . Nosotros enunciaremos que el origen es:

- *Estable* si cualquier trayectoria de estado empieza en $B(r)$ en un punto arbitrario \mathbf{x}_0 nunca alcanza el límite de la esfera $S(R)$ de $B(R)$.
- *Asintóticamente estable* siempre que es estable y además cada trayectoria de estado empieza en $B(R)$ en un punto arbitrario \mathbf{x}_0 , tiende al origen como el tiempo se incrementa indefinidamente.
- *Inestable* Siempre que para algún R y cualquier r , grande o pequeño, existe cualquier trayectoria de estado que empieza en $B(r)$ en un punto arbitrario \mathbf{x}_0 y alcance la esfera acotada $S(R)$. Se observa en la figura 3.2, para la trayectoria T el origen es inestable en el sentido de Lyapunov, a pesar de la convergencia de estado.

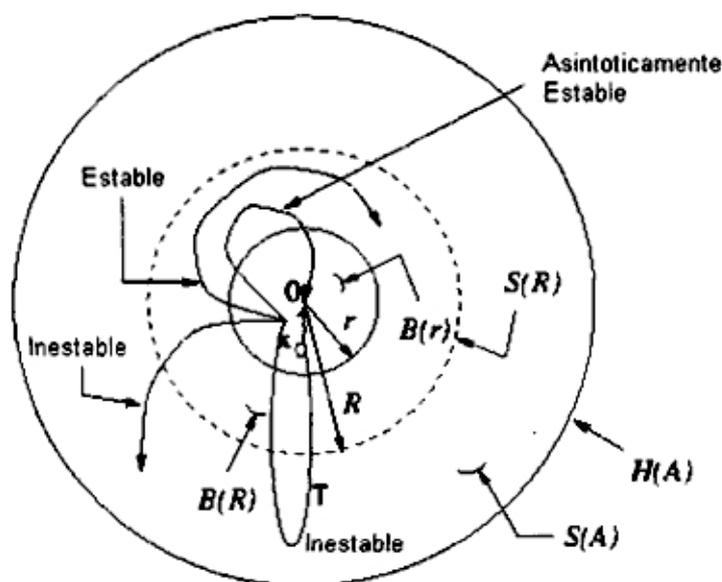


Figura 3.2: Estabilidad en sistemas autónomos.

3.2.2. Funciones de Lyapunov

Un tipo especial de función escalar $V(x)$, llamada función de Lyapunov, juega un papel muy importante en el análisis de estabilidad y diseños en sistemas de control. Una función de Lyapunov $V(x)$ verifica las siguientes propiedades.

- (a) $V(x)$ y sus primeras derivadas parciales

$$\frac{\partial V(x)}{\partial x} = \nabla V(x)$$

son continuas en cierta region abierta Ω alrededor del origen.

- (b) $V(0) = 0$
- (c) Fuera del origen, pero siempre en Ω , $V(x)$ es positiva. Asimismo, el origen es un mínimo de separación de $V(x)$.
- (d) $\dot{V}(x) = \nabla V(x)\dot{x} = \nabla V(x)f(x) \leq 0$ en Ω .

$V(x)$ se dice que es función definida positiva si cumple las propiedades (a)-(c). La figura 3.3, ilustra la función de Lyapunov para un sistema de segundo orden. Notar que $V(x_1, x_2)$ tiene el aspecto general de una parábola apuntando hacia

arriba. Si V fue definida negativa, la parábola estaría apuntando hacia abajo. Ahora, $V(x)$ es definida negativa si $-V(x)$ es definida positiva. También $V(x)$ es semi-definida positivo si $V(0) = 0$ y $V(x) \geq 0$ para $x \neq 0$; $V(x)$ es semi-definido negativo si $-V(x)$ es semi-definido positivo.

Además, una matriz cuadrada V $n \times n$ es definida positiva, es decir $x^T V x > 0$, si la función cuadrática $x^T V x$ es definida positiva para cualquier $x \neq 0$. Así mismo una matriz cuadrada V $n \times n$ es definida positiva si todos sus eigenvalores son mayores que cero. Similarmente una matriz cuadrada V $n \times n$ es semi-definida positiva, es decir $x^T V x \geq 0$, si la función cuadrática $x^T V x$ es semi-definida positiva para cualquier x .

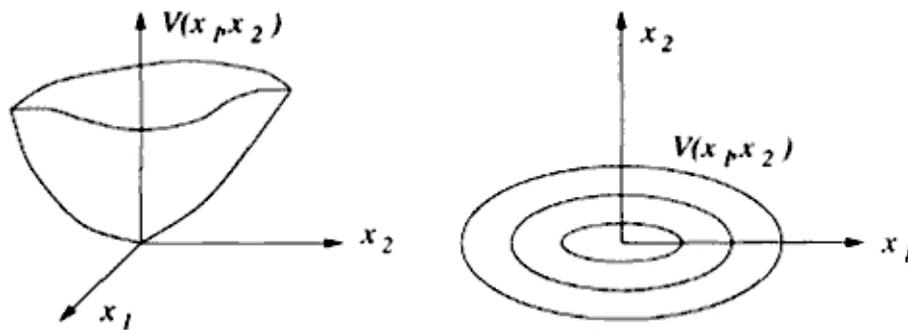


Figura 3.3: Representación gráfica de la función de Lyapunov.

Funciones Lyapunov para Sistemas No-Autónomos

Los sistemas no autónomos en consideración son descritos en una región Ω : $\|\mathbf{x}\| < A$ por.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \quad \mathbf{f}(\mathbf{0}, t) = \mathbf{0} \quad t \geq 0 \quad (3.4)$$

Tenemos que $W(\mathbf{x})$ es una función de Lyapunov en Ω , entonces $V(\mathbf{x}, t)$ será una función de Lyapunov si

- (a) $V(\mathbf{x}, t)$, es definida en $\Omega \quad \forall t \geq 0$
- (b) $V(\mathbf{0}, t) = 0 \quad \forall t \geq 0$
- (c) $V(\mathbf{x}, t) \geq W(\mathbf{x}) \quad \forall \mathbf{x}$ en $\Omega \quad \text{y} \quad \forall t \geq 0$

(d) $\dot{V}(\mathbf{x}, t) \leq 0 \quad \forall t \geq 0,$ donde

$$\dot{V} = \frac{dV}{dt} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} \dot{x} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} \mathbf{f}(\mathbf{x}, t)$$

Una función $V(\mathbf{x}, t)$ se dice que va ser función definida positiva si cumple las condiciones (a) – (c). También, $V(\mathbf{x}, t)$ es una función definida positiva si contiene una función definida positiva $W(\mathbf{x})$, es decir, si $V(\mathbf{x}, t) \geq W(\mathbf{x})$. Similarmente, $V(\mathbf{x}, t)$ es definido negativo si $-V(\mathbf{x}, t)$ es definido positivo; $V(\mathbf{x}, t)$ es semi-definido positivo si contiene una función semi-definida $W(\mathbf{x})$; $V(\mathbf{x}, t)$ es semi-definido negativo si $-V(\mathbf{x}, t)$ es semi-definido positivo.

3.2.3. Teoremas de la Estabilidad de Lyapunov

Los teoremas de estabilidad de Lyapunov generalizan la idea que cerca a un estado de equilibrio de un sistema físico la energía del sistema es siempre decreciente.

Teoremas de Estabilidad de Lyapunov para Sistemas Autónomos

Teorema 3.1 (Teorema de Estabilidad). *El equilibrio en el origen es estable si allí existe en alguna cercanía Ω del origen una función de Lyapunov $V(x)$.*

Teorema 3.2 (Teorema de Estabilidad Asintótica). *El equilibrio en el origen es asintóticamente estable si $-\dot{V}$ es definido positivo (i.e. $-\dot{V} > 0$) en Ω .*

Teorema 3.3 (Teorema de Estabilidad Completa (Global)). *Consideramos una función escalar $V(x)$ con las primeras derivadas parciales continuas para todo $x \neq 0$ tal que $V(x) > 0$, $\dot{V}(x) < 0$ y $V(x) \rightarrow \infty$ como $\|\mathbf{x}\| \rightarrow \infty$. Entonces el sistema autónomo es completamente (globalmente) estable asintóticamente.*

Teorema 3.4 (Teorema de Inestabilidad de Cęteav). *Asumamos que Ω es una vecindad del origen. Definimos una región Ω_1 contenida en una región Ω . Dada una función $V(\mathbf{x})$ en Ω . Entonces el equilibrio en el origen es inestable si*

- (a) $V(\mathbf{x})$ tiene las primeras derivadas parciales continuas en Ω_1 .
- (b) $V(\mathbf{x})$ y $\dot{V}(x)$ son definidas positivas en Ω_1 .
- (c) $V(\mathbf{x})=0$ en los puntos límites de Ω_1 dentro de Ω .

- (d) El origen es un punto límite de Ω_1 . La figura 3.4 ilustra la inestabilidad (Teorema de Cęteav)

El teorema de inestabilidad de Cęteav viene a ser el llamado primer teorema de inestabilidad si $\Omega = \Omega_1$, con Ω estando a cierta cercanía al origen. Si, además la condición $\Omega = \Omega_1$, la condición $\dot{V} > 0$ es remplazado por

$$\dot{V}(x) - \lambda V(x) \geq 0 \quad \forall t \geq 0 \quad \forall x \in \Omega$$

de manera que el teorema de inestabilidad de Cęteav llega a ser el segundo teorema de inestabilidad. Las demostraciones de los teoremas descritos anteriormente estan dadas en su forma geometrica en [12] [5].

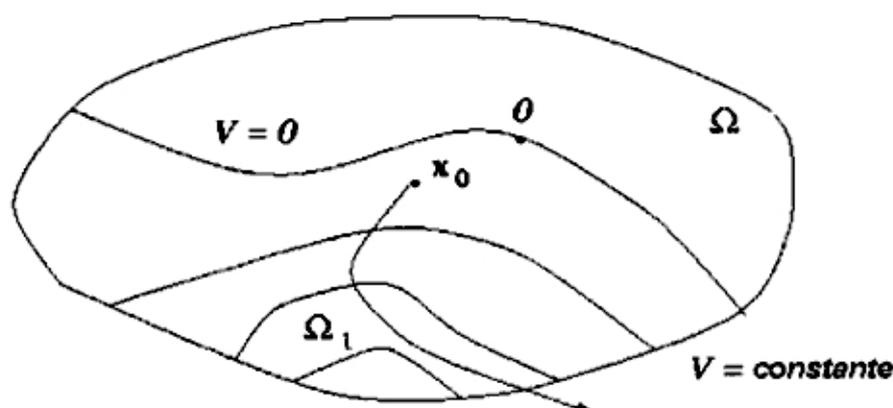


Figura 3.4: la representación grafica del teorema de inestabilidad de Cęteav.

Teoremas de Estabilidad de Lyapunov para Sistemas No Autonomos

Teorema 3.5 (Teorema de Estabilidad). *El equilibrio en el origen es estable si allı existe en alguna cercanıa Ω del origen una funcion de Lyapunov $V(\mathbf{x}, t)$.*

Teorema 3.6 (Teorema de Estabilidad Asintotica Uniforme). *El equilibrio en el origen es uniformemente y asintoticamente estable si una funcion definida positiva $V(\mathbf{x}, t)$ es decreciente (significa que V es contenido por una funcion definida positiva $W(\mathbf{x})$ para cualquier $t \geq 0$) y $-\dot{V}(\mathbf{x}, t)$ es definido positivo (i.e., $-\dot{V} > 0$) en Ω .*

Teorema 3.7 (Teorema de Inestabilidad). *Dejemos que Ω es una vecindad del origen. Dejemos que Ω_1 es una región en Ω . Dada una función $V(\mathbf{x}, t)$ en Ω . Entonces el equilibrio en el origen en el tiempo t_0 es inestable si:*

- (a) $V(\mathbf{x}, t)$ tiene las primeras derivadas parciales continuas en Ω_1 .
- (b) $V(\mathbf{x})$ y $\dot{V}(\mathbf{x})$ son definidas positivas en Ω_1 .
- (c) $V(\mathbf{x}) = 0$ para cualquier $t \geq 0$ en los puntos límites de Ω_1 dentro de Ω .
- (d) El origen es un punto límite de Ω_1 dentro de Ω , $V(\mathbf{x}, t) = 0$ para todo $t \geq t_0$.

El teorema de inestabilidad llega a ser el primer teorema de inestabilidad si $\Omega = \Omega_1$, con Ω estando en cierta cercanía al origen. Si, además la condición $\Omega = \Omega_1$, la condición $\dot{V} > 0$ es reemplazado por

$$\dot{V}(\mathbf{x}, t) - \lambda V(\mathbf{x}, t) \geq 0 \quad \forall t \geq t_0 \quad \forall x \in \Omega$$

entonces el teorema de inestabilidad llega a ser el llamado segundo teorema de inestabilidad.

Teoremas para Conjuntos Invariantes

En aplicaciones de sistemas de control, la estabilidad asintótica es mas importante que la estabilidad. Ciertamente, pequeñas desviaciones de la salida controlada con respecto a la referencia deseada puede ser cancelada tanto como el resultado de la operación de un sistema estable asintóticamente. Sin embargo, aplicando el segundo teorema de estabilidad, aquello a menudo sucede que $-\dot{V} \geq 0$ en lugar de la condición requerida $-\dot{V} > 0$ para una estabilidad asintótica. Pare tales casos, podria ser mas útil emplear el **conjunto de teoremas invariantes** atribuidos a La Salle en orden de obtener algunas conclusiones en estabilidad asintótica.

Conjunto invariante.- Un conjunto G se dice que es un conjunto invariante para un sistema dinámico si cada trayectoria de estado empieza de un punto x_0 en G y permanece en G . Por lo tanto, una trayectoria de estado cerrada en G es un conjunto invariante.

Teorema 3.8 (Teorema del conjunto invariante). *Consideremos el sistema dado por la ecuación 3.1. Sea $V(\mathbf{x})$ una función escalar con sus primeras*

derivadas parciales continuas. Permitiremos Ω_l ser una región acotada definida por $V(\mathbf{x}) < l > 0$. Asumimos que

$$\dot{V}(\mathbf{x}) \leq 0 \quad x \in \forall \Omega_l \quad (3.5)$$

sea R un conjunto de todos los puntos dentro de Ω_l donde $V(x) = 0$, y dejaremos M ser un conjunto invariante grande en R . Entonces cada solución $\mathbf{x}(t)$ en Ω_l tiende a M como $t \rightarrow \infty$. Si la condición (3.5) es sustituida por

$$\dot{V}(x) < 0 \quad \forall \quad x \neq 0 \quad \text{en} \quad \Omega_l \quad (3.6)$$

y el origen esta en Ω_l , entonces como el origen es asintóticamente estable, y cada solución en Ω_l tiende a el origen como $t \rightarrow \infty$.

Teorema 3.9 (Teorema del conjunto invariante global). *Consideraremos el sistema dado por la ecuación 3.1. Sea $V(x)$ una función escalar con sus primeras derivadas parciales continuas. Supondremos que $V(x) > 0$ para todo $x \neq 0$ y $\dot{V} \leq 0$. Permitiremos R ser el conjunto de todos los puntos donde $\dot{V} = 0$, y M ser un conjunto invariante grande en R . Entonces todas las soluciones converge completamente (globalmente) y Asintóticamente a M como $t \rightarrow \infty$.*

3.3. Control Adaptivo para Sistemas No Lineales

Control adaptivo ha sido satisfactoriamente desarrollado para algunas clases importantes de problemas de control no lineal. Tales problemas generalmente satisfacen las siguientes condiciones:

1. La dinámica de la planta no lineal puede ser linealmente parametrizable
2. Vector de estado tiene que ser medible.
3. Las no linealidades pueden ser establemente canceladas por la entrada de control si los parámetros son conocidos.

3.3.1. MRACS para Procesos SISO No Lineales)

Consideremos un sistema SISO no lineal de n-ésimo orden

$$y^{(n)} + \sum_{i=1}^n \alpha_i f_i(\mathbf{x}, t) = bu \quad \frac{d^n y}{dt^n} \triangleq y^{(n)} \quad (3.7)$$

donde $x = [y \dot{y}, \dots, y^{(n-1)}]^T$ es el vector de estado, f_i son funciones no lineales conocidas y los parámetros α_i y b son constantes desconocidas. Nosotros asumiremos que los estados son medidos, y que el signo de b es conocido. El objetivo del diseño del control adaptivo es hacer que la salida del sistema siga a la salida deseada $y_d(t)$ a pesar de sus parámetros inciertos. Para facilitar la derivación del controlador adaptivo, escribiremos la ecuación anterior como

$$hy^{(n)} + \sum_{i=1}^n a_i f_i(\mathbf{x}, t) = u \quad (3.8)$$

Dividiendo a ambos lados por una constante desconocida b , donde $h = 1/b$ y $a_i = \alpha_i/b$

3.3.2. Ley de Control

Definimos la salida del error de seguimiento como

$$e = y - y_d$$

tal que el error concatenado s puede ser expreado como.

$$s = e^{(n-1)} + \lambda_{n-2}e^{(n-2)} + \dots + \lambda_0 e = \Delta_{(p)}e = y^{(n-1)} - y_r^{(n-1)} \quad (3.9)$$

donde

$$\Delta_{(p)} = p^{n-1} + \lambda_{n-2}p^{n-2} + \dots + \lambda_0 = (p - p_1)\dots(p - p_n)$$

$$y_r^{(n-1)} = y_d^{(n-1)} - y_{n-2}e^{(n-2)} - \dots - \lambda_0 e$$

en la cual p es el operador de Laplace y Δ_p es un polinomio estable Hurwitz, significa que todos sus raíces complejas $p_i = \sigma_i + j\omega_i, i = 1, \dots, n$ de $\Delta_p = 0$ verifican la condición $\sigma_i < 0$.

Consideremos la ley de control.

$$u = hy_r^{(n)} - ks + \sum_{i=1}^n a_i f_i(\mathbf{x}, t) \quad (3.10)$$

donde k es una constante del mismo signo como h , y la variable y_r^n denotada como la referencia es la derivada de $y_r^{(n-1)}$, es decir

$$y_r^{(n)} = y_d^{(n)} - \lambda_{n-1}e^{(n-1)} - \dots - \lambda_0 e \quad y_r^{(n)} = \frac{dy_r^{(n-1)}}{dt}$$

sustituyendo la ecuación (3.10) dentro de la ecuación (3.8) obtenemos que la dinámica del error de seguimiento es

$$hs + ks = 0 \quad (3.11)$$

Y por tanto da una convergencia exponencial de s . El cual a su vez, garantiza la convergencia del error de seguimiento e .

3.3.3. Ley de Adaptación

La ley de control para el sistema MRACS esta definida por

$$u = \hat{h}y_r^{(n)} - ks + \sum_{i=1}^n \hat{a}_i f_i(\mathbf{x}, t) \quad (3.12)$$

donde h y el a_i han sido reemplazado por sus valores estimados. El error de seguimiento desde esta ley de control puede ser fácilmente mostrado

$$hs + ks = \tilde{h}y_r^{(n)} + \sum_{i=1}^n \tilde{a}_i f_i(\mathbf{x}, t) \quad (3.13)$$

donde $\tilde{h} = \hat{h} - h$ y $\tilde{a}_i = \hat{a}_i - a_i$

Esto puede ser reescrito como

$$s = \frac{\frac{1}{h}}{p + \frac{k}{h}} [\tilde{h}y_r^{(n)} + \sum_{i=1}^n \tilde{a}_i f_i(\mathbf{x}, t)] \quad (3.14)$$

con la función de transferencia estrictamente positiva y real, escogemos la siguiente ley de adaptación.

$$\dot{\hat{h}} = -\gamma \text{sgn}(h) s y_r^n \quad \dot{\hat{a}}_i = -\gamma \text{sgn}(h) s f_i \quad (3.15)$$

donde γ es la ganancia de adaptación.

Para determinar si el diseño MRACS garantiza una convergencia global de seguimiento, consideraremos la siguiente función candidata de Lyapunov

$$V = |h|s^2 + \gamma^{-1}[\tilde{h}^2 + \sum_{i=1}^n \tilde{a}_i^2] \quad (3.16)$$

Usando el factor $h \operatorname{sign}(h) \equiv |h|$ y $k \operatorname{sign}(k) \equiv |k|$, se puede facilmente demostrar que

$$\dot{V} = -2|k|s^2 < 0 \quad (3.17)$$

el cual garantiza una convergencia global de seguimiento MRACS método directo en el sentido de Lyapunov, ahora consideremos la ley de adaptación

$$\dot{\tilde{h}} = -\gamma_h \operatorname{sign}(h) s y_r^n \quad \dot{\tilde{a}}_i = -\gamma_a \operatorname{sign}(h) s f_i \quad (3.18)$$

donde las ganancias de la adaptación γ_h y γ_k son diferentes para cada parámetro desconocido. Seleccionamos la siguiente función candidata de Lyapunov.

$$V = |h|s^2 + \gamma_h^{-1} \tilde{h}^2 + \gamma_k^{-1} \sum_{i=1}^n \tilde{a}_i^2 \quad (3.19)$$

Puede facilmente demostrar que \dot{V} mencionado en la ecuación (3.17), garantiza una convergencia global de seguimiento del modelo MRACS.

3.4. MRACS para Procesos MIMO No Lineales

Conocemos que la dinámica de la gran mayoría de procesos electro-mecánicos, tales como manipuladores robóticos es fuertemente no lineal, y puede ser escrito en forma general

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{P}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{d}(\mathbf{q}) = \mathbf{u} \quad \mathbf{q} = \begin{bmatrix} q_1 \\ \vdots \\ q_m \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \quad (3.20)$$

donde las matrices \mathbf{M} , \mathbf{P} y \mathbf{d} representan la inercia del proceso, torques centrípetos y de coriolis, y el torque gravitacional respectivamente. También \mathbf{q} es el vector de coordenadas generalizadas y \mathbf{u} el vector de control.

El objetivo de control es diseñar una ley de control \mathbf{u} capaz de hacer que la salida del proceso $\mathbf{q}(t)$ siga la trayectoria deseada $\mathbf{q}_d(t)$ con suficiente velocidad en presencia de parámetros inciertos.

Asumiremos que todos los términos de la ecuación (3.20) dependen linealmente de un vector de parámetros \mathbf{a} de elementos desconocidos, es decir

$$\mathbf{Y}\mathbf{a} = \mathbf{u} \quad (3.21)$$

donde \mathbf{Y} es una matriz conocida. También, consideramos la siguiente ley de control

$$\mathbf{u} = \mathbf{Y}\hat{\mathbf{a}} - \mathbf{K}_D\mathbf{s} \quad (3.22)$$

donde $\mathbf{K}_D\mathbf{s}$ es el término derivativo, $\mathbf{Y}\hat{\mathbf{a}}$ es el término "feed forward", \mathbf{K}_D (la ganancia derivativa) es una matriz constante simétrica definida positiva, y \mathbf{s} es un vector de la superficie sliding cuyos elementos $s_i, i = 1, \dots, m$ son definidos por la ecuación escalar $s_i(\mathbf{q}, t) = 0$, tal que

$$s_i = \left(\frac{d}{dt} + \lambda_i\right)^{n-1} \tilde{q}_i = (p + \lambda_i)^{n-1} \tilde{q}_i \quad (3.23)$$

donde λ_i es una constante positiva (ancho de banda), p es el operador laplaceano y $\tilde{q}_i = q_i - q_{di}$ es el error de seguimiento. Por ejemplo la forma del vector para la ecuación (3.23), donde $n = 2$ tiene la siguiente velocidad mas un error de seguimiento proporcional

$$\mathbf{s} = \dot{\tilde{\mathbf{q}}} + \Lambda\tilde{\mathbf{q}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r \quad \tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d \quad \dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \Lambda\tilde{\mathbf{q}} \quad (3.24)$$

donde

$$\mathbf{s} = [s_1 \dots s_m]^T \quad \dot{\tilde{\mathbf{q}}}_1 = [\dot{\tilde{q}}_1 \dots \dot{\tilde{q}}_m]^T \quad \tilde{\dot{\mathbf{q}}}_r = [\dot{\tilde{q}}_{r1} \dots \dot{\tilde{q}}_{rm}]^T$$

$$\mathbf{q}_d = [q_{d1} \dots q_{dm}]^T \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \lambda_m \end{bmatrix}$$

La conservación de la energía requiere que

$$\frac{1}{2} \frac{d}{dt} [\dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}}] = \dot{\mathbf{q}}^T (\mathbf{u} - \mathbf{d}) \quad (3.25)$$

donde $\dot{q}^T M \dot{q}$ es la energía cinética del proceso y $\dot{q}^T (u - d)$ es la entrada de potencia generada por el actuador. Diferenciando la parte izquierda de la ecuación (3.25) obtenemos

$$\dot{q}^T M \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M} \dot{q} = \dot{q}^T (u - d)$$

de la ecuación (3.20) obtenemos $M \ddot{q} = u - d - P \dot{q}$ y sustituyendo este término dentro de la última ecuación se obtiene

$$\dot{q}^T (\dot{M} - 2P) \dot{q} = 0 \quad (3.26)$$

Se ha establecido que $(\dot{M} - 2P) = J$ es una matriz anti-simetría. Por lo tanto

$$\dot{M} - 2P = J \quad (3.27)$$

Consideramos la siguiente función candidata de Lyapunov

$$V(t) = \frac{1}{2} [s^T M s + \hat{a}^T \Gamma^{-1} \tilde{a}] \quad (3.28)$$

Donde Γ es una matriz simétrica definida positiva y $\tilde{a} = \hat{a} - a$ es el error de estimación de parámetros. Diferenciando la ecuación (3.28) se obtiene

$$\dot{V}(t) = s^T (M \ddot{q} + M \ddot{q}_r) \frac{1}{2} s^T \dot{M} s + \dot{\hat{a}}^T \Gamma^{-1} \tilde{a}$$

donde hemos usado la relación $\dot{s} = \ddot{q} - \ddot{q}_r$ de la ecuación (3.24). Sustituyendo la ecuación (3.27) en la ecuación (3.22), y $M \ddot{q}$ de la ecuación (3.20) en \dot{V} se obtiene

$$\dot{V}(t) = s^T (Y \hat{a} - K_D s - M \ddot{q}_r - P \dot{q}_r - d) + \dot{\hat{a}}^T \Gamma^{-1} \tilde{a} \quad (3.29)$$

En la cual usamos el factor $s^T J s = 0$. La actualización de los Parámetros estimados \hat{a} pueden ser formulado como.

$$\dot{\hat{a}} = -\Gamma Y^T s \quad (3.30)$$

Mientras la propiedad de linealización paramétrica en la ecuación (3.29) establece que

$$Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r) a = M(q) \ddot{q}_r + P(q, \dot{q}) \dot{q}_r + d(q) \quad (3.31)$$

reemplazando la ecuación (3.31) y (3.30) en la ecuación (3.29) nos lleva

$$\dot{V}(t) = -s^T K_D s \leq 0 \quad (3.32)$$

Puesto que $s = \dot{\tilde{q}} + \Lambda \tilde{q}$ (ver ecuación 3.24), entonces en la ecuación (3.32) garantiza que el error de seguimiento $\tilde{\mathbf{q}}$ y $\dot{\tilde{\mathbf{q}}}$ tienden a $\mathbf{0}$ como t tiende al infinito. En otras palabras, el error de seguimiento converge a la superficie $\mathbf{s} = \mathbf{0}$.

Para nuestro caso, de las relaciones (3.21), (3.22) y (3.31) toman la siguiente forma

$$Y a = T \quad (3.33)$$

$$T = Y \hat{a} - K_D s \quad (3.34)$$

$$T = Y(q, \dot{q}, \ddot{q}_r, \ddot{q}_r) a = H(q) \ddot{q}_r + C(q, \dot{q}) \dot{q}_r + d(q) \quad (3.35)$$

La actualización de la estimación de parámetros \hat{a} en la ecuación (3.30) podría también ser empleado para este caso.

3.5. Diseño del Control MRACS para el MRTM

3.5.1. Motivación

El objetivo principal del control adaptivo es mantener consistente el rendimiento del sistema en presencia de incertidumbres, o variaciones no conocidas en los parámetros de la planta. Control Adaptivo puede también mantener buen error de seguimiento en presencia de cambios físicos característicos tales como la fricción y diferentes cambios en la masa. El controlador adaptivo requiere poco del conocimiento a priori de los actuales parámetros físicos del sistema y utiliza el misma cantidad de control como un simple controlador PD. Además el controlador adaptivo tiene como beneficios adicionales que puede en muchos casos ser usado para estimación de parámetros. El control adaptivo es desarrollado para algunas importantes clases de problemas de control no lineal. Tales problemas usualmente satisfacen las siguientes condiciones

- La dinámica de la planta no lineal puede ser linealmente parametrizada.
- El estado total es medible.
- Las no linealidades pueden ser canceladas firmemente por la entrada de control si los parámetros son conocidos.

Las aplicaciones son múltiples y los podemos encontrar en robots manipuladores, control aircraft, control de procesos, etc

3.5.2. El Modelo de la Planta

Nuestro sistema es un simple manipulador no lineal que esta expresada por la siguiente ecuación dinámica

$$M(\mathbf{q})\ddot{\mathbf{q}} + P(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + d(\mathbf{q}) = \mathbf{u} \quad (3.36)$$

$$M \begin{bmatrix} \ddot{r} \\ \ddot{\theta}_1 \end{bmatrix} + P \begin{bmatrix} \dot{r} \\ \dot{\theta}_1 \end{bmatrix} + d = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.37)$$

despejando obtenemos:

$$\begin{bmatrix} \ddot{r} \\ \ddot{\theta}_1 \end{bmatrix} = -M^{-1}(P \begin{bmatrix} \dot{r} \\ \dot{\theta}_1 \end{bmatrix} + d) + M^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.38)$$

Las matrices columnas

$$\begin{bmatrix} r \\ \theta_1 \end{bmatrix}; \quad \begin{bmatrix} \dot{r} \\ \dot{\theta}_1 \end{bmatrix}; \quad \begin{bmatrix} \ddot{r} \\ \ddot{\theta}_1 \end{bmatrix} \quad (3.39)$$

representan la posición , velocidad y aceleración respectivamente.

Representación de Matrices y Lista de Variables del Sistema

Las matrices **M**, **P** y **d**, son de la forma:

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \quad P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \quad d = \begin{bmatrix} d_{11} \\ d_{21} \end{bmatrix}$$

donde:

$$\begin{array}{ll} m_{11} = \left(\frac{R_a R_p}{nk_a k_m}\right) \left(M_c + m + \frac{J_{eq1}}{R_p^2}\right) & m_{12} = \left(\frac{R_a R_p}{nk_a k_m}\right) \frac{ml}{2} \\ m_{21} = \left(\frac{R_a}{nk_a k_m}\right) \frac{ml}{2} & m_{22} = \left(\frac{R_a}{nk_a k_m}\right) \left(\frac{ml^2}{4} + J + J_{eq2}\right) \\ p_{11} = \left(\frac{R_a R_p}{nk_a k_m}\right) \left(F + \frac{B_{eq1}}{R_p^2} + \frac{n^2 k_m k_b}{R_a R_p^2}\right) & p_{12} = -\left(\frac{R_a R_p}{nk_a k_m}\right) \frac{ml}{2} \\ p_{21} = 0 & p_{22} = \left(\frac{R_a}{nk_a k_m}\right) \left(C + B_{eq2} + \frac{n^2 k_m k_b}{R_a}\right) \\ d_{11} = 0 & d_{21} = -\left(\frac{R_a}{nk_a k_m}\right) \frac{mlg}{2} \end{array}$$

Agrupamos los parámetros del sistema quedando de la siguiente manera:

$$\begin{aligned}
\text{alfa} &= \frac{R_a r_p}{n k_m k_a} \\
\text{beta} &= \frac{R_a}{n k_a k_m} \\
a_1 &= \text{alfa} \left(M_c + m_1 + \frac{J_{eq}}{r_p^2} \right) \\
a_2 &= \text{alfa} \left(m_1 \frac{L}{2} \right) \\
a_3 &= \text{beta} \left(m_1 \frac{L}{2} \right) \\
a_4 &= \text{beta} \left(m_1 \frac{L^2}{4} + J_1 + J_{eq} \right) \\
a_5 &= \text{alfa} \left(f + \frac{B_{eq}}{r_p^2} + \frac{n^2 k_m k_h}{R_a r_p^2} \right) \\
a_6 &= \text{beta} \left(c_1 + B_{eq} + \frac{k_m k_h n^2}{R_a} \right)
\end{aligned}$$

Definimos las matrices M , P y d en función de sus parámetros a_1, a_2, a_3, a_4, a_5 y

a_6

obtenemos:

$m_{11} = a_1$	$p_{11} = a_5$	$d_{11} = 0$
$m_{12} = a_2 \cos(x_2)$	$p_{12} = -a_2 \sin(x_2) x_4$	$d_{21} = -a_3 g \cdot \sin(x_2)$
$m_{21} = a_3 \cos(x_2)$	$p_{21} = 0$	
$m_{22} = a_4$	$p_{22} = a_6$	

Las variables de estado a usarse

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{bmatrix}$$

3.5.3. Diseño del Controlador

[7] Dada una trayectoria deseada $q_d(t)$, con algunos o todos sus parámetros desconocidos; el problema del diseño del controlador es derivar una ley de control y una ley de adaptación (estimación) con parámetros desconocidos para que la salida del sistema $q(t)$ converge con la trayectoria deseada. A este efecto se define $\tilde{a} = \hat{a} - a$, como el parámetro de estimación de error, con a siendo un vector constante de los parámetros desconocidos, que describe las propiedades de la masas del sistema y \hat{a} es el estimado. Definimos la matriz conocida:

$$\mathbf{Y} = \mathbf{Y}(q, \dot{q}, \ddot{q}, \ddot{r})$$

tal que

$$\mathbf{M}(q)\ddot{q} + \mathbf{P}(q, \dot{q})\dot{q} + \mathbf{d}(q) = \mathbf{Y}(q, \dot{q}, \ddot{q}_r, \ddot{q}_r)\mathbf{a} \quad (3.40)$$

donde

$$\mathbf{Y} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} & Y_{16} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} & Y_{25} & Y_{26} \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix}$$

Realizando el cambio:

$$\dot{r} = d\frac{qr}{dt} = dq_1 \quad \dot{\theta} = d\frac{\theta}{dt} = dq_2 \quad \ddot{r} = d\frac{dq_1}{dt} = ddq_1 \quad \ddot{\theta} = d\frac{dq_2}{dt} = ddq_2$$

y reemplazando en la ecuación (3.40), se obtiene:

$$Y_{11a} = ddq_1$$

$$Y_{12a} = \cos(x_2)ddq_2 - \sin(x_2)x_4dq_2$$

$$Y_{13a} = 0$$

$$Y_{14a} = 0$$

$$Y_{15a} = dq_1$$

$$Y_{16a} = 0$$

$$Y_{21b} = 0$$

$$Y_{22b} = 0$$

$$Y_{23b} = \cos(x_2)ddq_1 - g\sin(x_2)$$

$$Y_{24b} = ddq_2$$

$$Y_{25b} = 0$$

$$Y_{26b} = -dq_2$$

3.5.4. La Ley de Control

La ley de control viene expresado por:

$$u = Y\hat{a} - K_D s \quad (3.41)$$

donde u es la señal de control, \mathbf{Y} es una matriz de términos conocidos, \mathbf{a} es un vector de parámetros desconocidos, K_D es un ganancia derivativa y \mathbf{s} la superficie sliding.

Definimos las variables utilizadas en la ley de control de la siguiente manera:

$$s = \dot{\tilde{q}} + \Lambda \tilde{q} = \dot{q} - \dot{q}_r$$

$$\dot{q}_r = \dot{q}_d - \Lambda \tilde{q}$$

$$\tilde{q} = q - q_d$$

donde:

\dot{q}_r : velocidad de referencia

Λ : ancho de banda)

\tilde{q} : error de seguimiento

q : posición

q_d : trayectoria deseada

Para el manipulador robótico, el cual tiene 02 entradas y 02 salidas

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} Y_{11a} & Y_{12a} & Y_{13a} & Y_{14a} & Y_{15a} & Y_{16a} \\ Y_{21b} & Y_{22b} & Y_{23b} & Y_{24b} & Y_{25b} & Y_{26b} \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{a}_4 \\ \hat{a}_5 \\ \hat{a}_6 \end{bmatrix} - K_D \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \quad (3.42)$$

desarrollando la matriz anterior obtenemos el valor de u_1 y u_2

$$u_1 = Y_{11a}\hat{a}_1 + Y_{12a}\hat{a}_2 + Y_{13a}\hat{a}_3 + Y_{14a}\hat{a}_4 + Y_{15a}\hat{a}_5 + Y_{16a}\hat{a}_6 - K_D s_1 \quad (3.43)$$

$$u_2 = Y_{21b}\hat{a}_1 + Y_{22b}\hat{a}_2 + Y_{23b}\hat{a}_3 + Y_{24b}\hat{a}_4 + Y_{25b}\hat{a}_5 + Y_{26b}\hat{a}_6 - K_D s_2 \quad (3.44)$$

Para nuestro caso la ley de control difiere ligeramente de la ecuacion (3.41) de la siguiente forma:

$$u = Y\hat{a} - K_D \dot{\tilde{q}} - K_p \tilde{q} \quad (3.45)$$

comprendiendo que la ganancia K_p este sintonizado tan alto que $K_D \Lambda$ alcance el mejor rendimiento en presencia de altas frecuencias de la dinámica no modelada.

Redefiniendo nuestra ley de control quedará de la siguiente forma:

$$u_1 = Y_{11a}\hat{a}_1 + Y_{12a}\hat{a}_2 + Y_{13a}\hat{a}_3 + Y_{14a}\hat{a}_4 + Y_{15a}\hat{a}_5 + Y_{16a}\hat{a}_6 - K_{D1}\dot{\tilde{q}} - K_{p1}\tilde{q} \quad (3.46)$$

$$u_2 = Y_{21b}\hat{a}_1 + Y_{22b}\hat{a}_2 + Y_{23b}\hat{a}_3 + Y_{24b}\hat{a}_4 + Y_{25b}\hat{a}_5 + Y_{26b}\hat{a}_6 - K_{D2}\dot{\tilde{q}} - K_{p2}\tilde{q} \quad (3.47)$$

3.5.5. La Ley de Adaptación

La estimación de parámetros es la parte final del Control Adaptivo el cual actualiza los parámetros basados en la medición del error de seguimiento s . La Ley de Adaptación es seleccionado tal que garantice la estabilidad en el sentido de Lyapunov y viene expresado por:

$$\dot{\hat{a}} = -\Gamma Y^T s \quad (3.48)$$

donde Γ : Matriz de ganancia simétrica

$$\Gamma = \text{diag} \left[k_1 \quad k_2 \quad k_3 \quad k_4 \quad k_5 \quad k_6 \right]$$

Empleando las ecuaciones anteriores, puede demostrarse que usando el método de Lyapunov, el sistema es estable y que el error de seguimiento debería, en casos ideales, converger a cero. Para trayectorias suficientemente "excitadas", los parámetros deberían converger a su valores actuales.

Capítulo 4

Implementación en Tiempo Real

La implementación del sistema MRTM de 2DOF se muestra en la figura 4.1. Los componentes físicos (hardware) del sistema de control son: dos servomotores DC (modelos iguales) con encoder óptico incremental de 1250 cuentas por vuelta con salida a dos canales en cuadratura. Los actuadores están compuesto por tarjetas similares, como son el generador de *PWM* que modula las señales de control para luego sean desfasadas en 90° y el amplificador de potencia que encargado de amplificar las señales que provienen del generador de *PWM* de modo que su salida genere el control de inversión del giro del servomotor DC. La tarjeta de adquisición de datos LABPC+ encargada de manejar los datos de entrada/salida del sistema y una computadora personal con microprocesador Pentium de 100 MHz. El sensado tiene el decodificador de cuadratura que se encarga de leer los datos del encoder óptico para emitir salidas *UP/DOWN* al circuito contador, que a su vez tiene una salida de 16 bits que son entradas digitales a la tarjeta LABPC+.

El sistema MRTM de 2DOF real se muestra en la figura 4.2 y la implementación del sistema de control real se observa en la figura 4.3. Estos sistemas han sido implementados en el laboratorio de proyectos de investigación en automatización y control de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Ingeniería.

4.1. El Actuador

El actuador está conformado por dos servomotores DC de campo magnético permanente y escobillas conmutadas para el control del MRTM de 2DOF, dos generadores de *PWM* y dos amplificadores de potencia como se muestra en la figura 4.1. Como el servomotor no responde inmediatamente a una tensión de entrada, le toma un tiempo para alcanzar la máxima velocidad, es necesario usar un sistema que permita variar la velocidad entre un valor mínimo y un valor

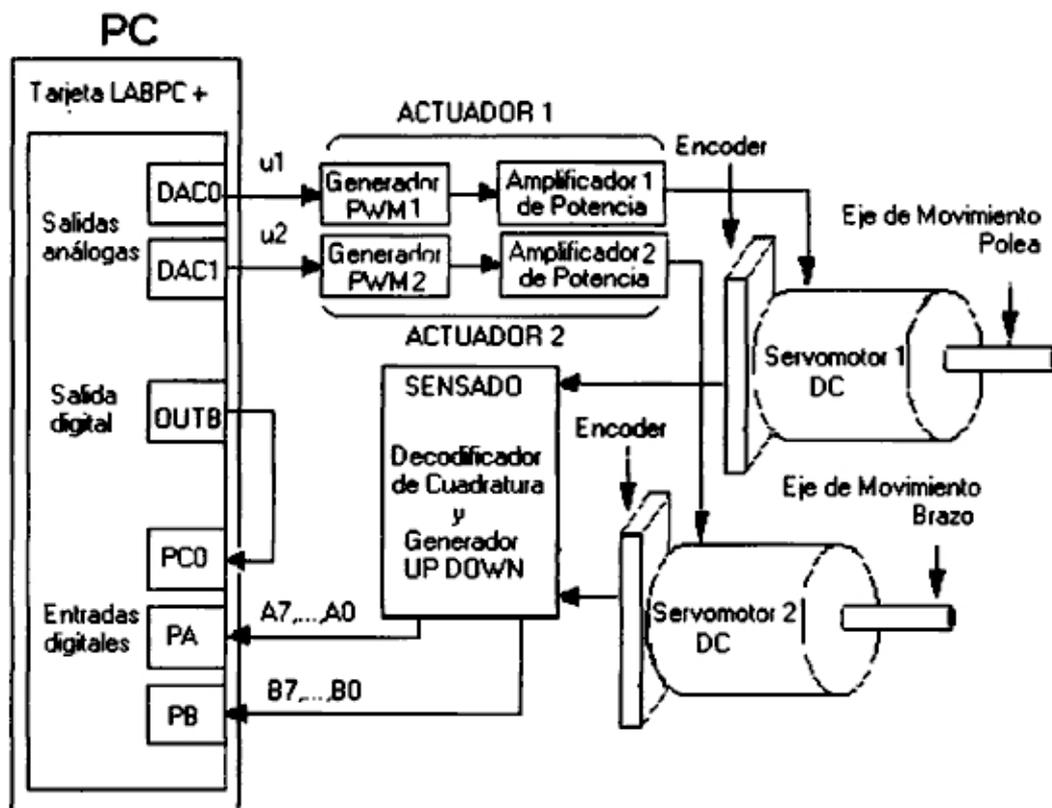


Figura 4.1: Diagramas de bloques de la implementación del sistema.



Figura 4.2: Planta real MRTM de 2DOF.

máximo, de modo que permita dar un cambio seguro en el sentido de rotación del servomotor DC. Este sistema está constituido por un generador *PWM* y el circuito amplificador de potencia que describiremos a continuación.



Figura 4.3: Implementación del sistema de control real.

El Generador de PWM

El generador de *PWM* que se muestra en la figura 4.4, se encarga de conmutar al motor mediante una serie de pulsos, logrando así, variar la velocidad de acuerdo al ancho de pulso.

El *LM3524* genera una modulación por ancho de pulso de decenas de KHz, esta señal será amplificada por el driver antes de alimentar al motor con zona muerta. El *LM3524* es un generador de *PWM*, el comparador *COMP* (pin 9) recibe la señal de control de la tarjeta de adquisición de datos LABPC+, el rango de tensión en el *COMP* puede variar entre 0.8 a 3.7 voltios, un valor superior o inferior satura al *LM3524*; la base de tiempo se controla por la resistencia R_T (pin 6) y el condensador C_T del (pin 7) trabajan aproximadamente a una frecuencia de 15.4Khz; en *CA* y *CB* (pin 12 y pin 13) respectivamente, se forman dos señales pulso desfasadas 180 grados, con un duty cycle limitado a variar de 0 a 50%; la suma de estas da lugar a una señal *PWM* con un duty cycle variable de 0 a 100%; el efecto de la resistencia de valor $R=330\Omega$ es sumar las señales de *CA* y *CB*; los condensadores de valores 0.1uF se utilizan para filtrar el ruido. Los circuitos osciladores 74LS123 realizan la función de sincronización para la generación del tren de pulsos en cuadratura, cuyas salidas son las señales *PWM*

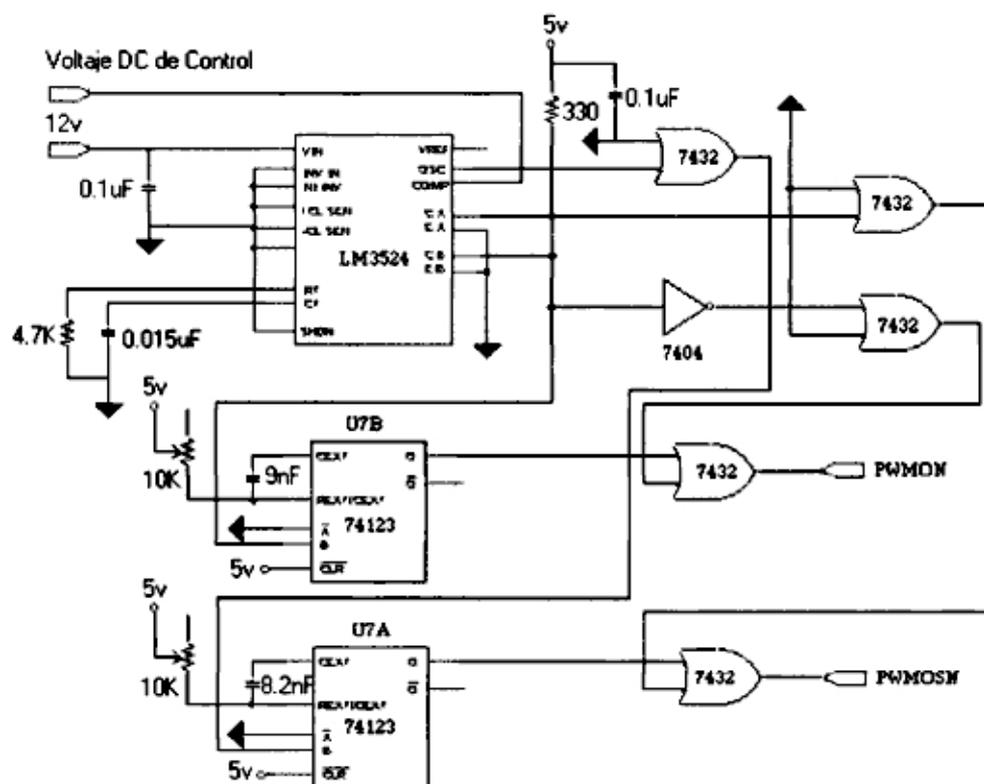


Figura 4.4: Circuito generador de PWM.

que serán entradas al circuito amplificador [10].

En la figura 4.5 se observa el diagrama de tiempos de la generación de PWM y \overline{PWM} con zona muerta, se están considerando los retardos de las compuertas lógicas AND y NOT . la señal PWM es mostrada en (a); la señal \overline{PWM} en (b) que se deriva de la compuerta lógica NOT que tiene un retardo T_a ; la compuerta AND que proviene de la salida OSC del $LM3524$ es utilizada como seguidor, así aumenta el $FANOUT$ del OSC de (c); los multivibradores $U7B$ y $U7A$ son disparados con los flancos de subida de PWM y OSC respectivamente; T_d es el retardo del multivibrador $U7A$, el ancho del pulso de la salida Q del $U7A$ (d) está controlado por el potenciómetro de $10k\Omega$ y un condensador de valor $C=9nF$; similarmente, T_b es el retardo del multivibrador $U7B$, el ancho del pulso de la salida Q del $U7B$ (g) es controlado por el potenciómetro de $10k\Omega$ y un condensador de valor $C=8.2nF$. El tiempo muerto entre el flanco de bajada del PWM y el flanco de subida del \overline{PWM} está dado por la salida Q del $U7A$ (d) el cual es adicionado a PWM generando la nueva señal de control $PWMOSN=PWM+(U7A-13)$ (f). El tiempo muerto para el siguiente flanco es dado por la salida Q del $U7B$ este se adiciona a \overline{PWM} generado la nueva señal de control $PWMON=\overline{PWM}+(U7B-5)$ (g) en esta señal se observa un retardo T_c que no es visible al conmutador porque tiene un retardo superior a T_c que es igual al retardo de una compuerta lógica (en el orden de los $20nseg$).

El Amplificador de Potencia

En la figura 4.6 se muestra el esquema del funcionamiento del amplificador de potencia conformado por un sistema de disparo y dos pares de conmutadores $A_{1,4}$ y $A_{2,3}$. El objetivo del sistema de disparo es realizar la conmutación de cada par $A_{1,4}$ y $A_{2,3}$, de tal forma que se eviten cortocircuitos durante el tiempo de conmutación de $A_{1,4}$ a $A_{2,3}$. Esto se manifiesta cuando el sistema de disparo es inadecuado, en el caso de una señal $PWMON$ y $PWMOSN$, las cuales controlan compuertas de conmutadores diferentes que deberían estar complementados en todo instante, el tiempo en que estos conmutadores están cerrados simultáneamente, cortocircuitan la fuente, lo que produce respuestas transitorias subamortiguadas durante el tiempo de conmutación, lo que se puede atenuar con un condensador adecuado, el efecto perjudicial se produce cuando en el tiempo en que los dos

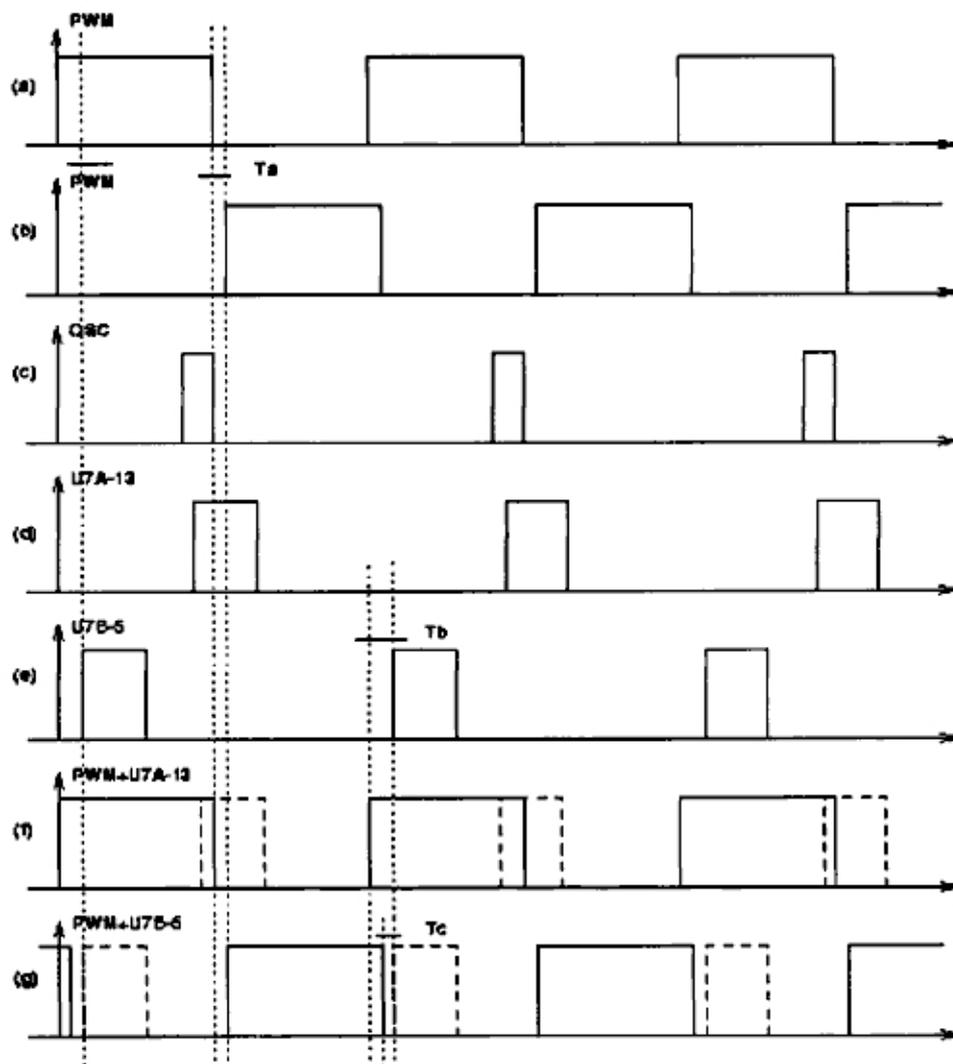


Figura 4.5: Diagrama de tiempo del circuito generador de PWM.

conmutadores se encuentran cerrados circula una corriente significativa que va deteriorando al dispositivo conmutador.

En síntesis, cuando el sistema de disparo cierra al conmutador $A_{1,4}$ y se abre $A_{2,3}$, el sentido de la corriente es la línea con puntos induciendo de esta forma una tensión $+V_{cc}$ en el servomotor. Luego si el sistema de disparo abre al conmutador $A_{1,4}$ y cierra $A_{2,3}$ el sentido de la corriente es la línea continua, induciendo así una tensión $-V_{cc}$ en el servomotor. Por lo tanto el servomotor ve en sus bornes una onda de voltaje cuadrada, variando entre $\pm V_{cc}$ y la corriente que puede absorber dependerá de los conmutadores.

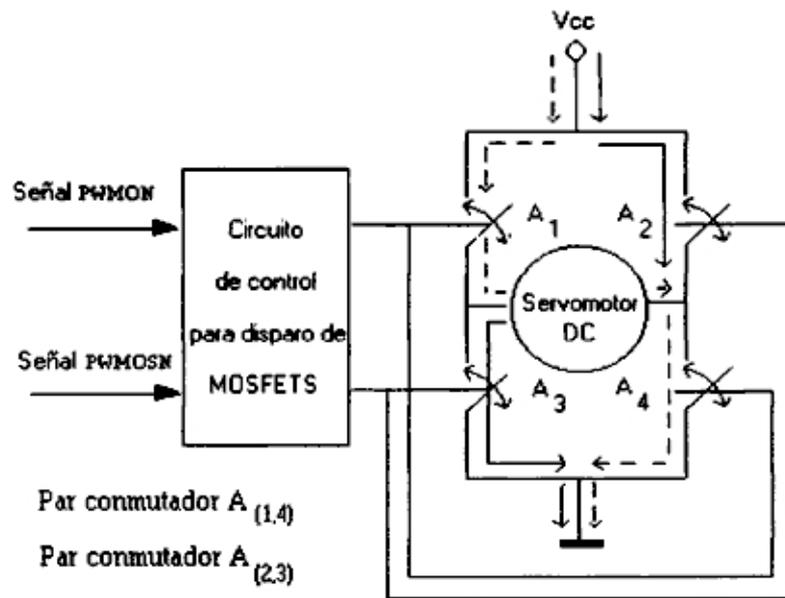


Figura 4.6: Esquema del circuito de disparo y conmutación.

Se ha previsto la modificación de la lógica de disparo mediante la construcción de un circuito en base en compuertas lógicas de tecnología TTL (Transistor Transistor Logic) conformado por integrados 7404, 7486, 7408 y dos resistencias de 500Ω . Este circuito gobierna los disparos de los MOSFET's de manera que las señales *PWM* que ingresan al circuito amplificador de potencia que tengan igual amplitud estarán siempre en nivel bajo (*OFF*), tal como se muestra en la tabla 4.1.

Cuadro 4.1: Conmutación de los MOSFET's.

<i>PWMON</i>	<i>PWMOSN</i>	<i>A₁</i>	<i>A₂</i>	<i>A₃</i>	<i>A₄</i>
0	0	OFF	OFF	OFF	OFF
0	1	ON	OFF	ON	OFF
1	0	OFF	ON	OFF	ON
1	1	OFF	OFF	OFF	OFF

Implementación del Amplificador de Potencia

Un circuito conmutador puede ser construido de varias formas, una cambiando la frecuencia de conmutación o el duty cycle y otras cambiando ambas. El

efecto buscado en un método u otro es que el promedio de voltaje de salida sea proporcional a la tensión de comando. El método más común de un conmutador amplificador es la modulación por ancho de pulso (*PWM*). En nuestro caso, los dispositivos conmutadores MOSFET's, conmutan a una frecuencia constante, dando como resultado la variación tensión de salida entre dos valores extremos. Por variación del ancho de pulso o el duty cycle, el valor promedio de voltaje de salida puede ser cambiado en forma proporcional a la tensión de comando.

El conmutador amplificador implementado es de tipo "H" su diagrama es visto en la figura 4.6. La principal ventaja del tipo "H" es que solo es necesario una sola fuente y son requeridos cuando se necesita una alta tensión; los dispositivos conmutadores son conmutados en pares generando una tensión bipolar a la salida. El efecto de tiempo muerto, el cual es explicado en la etapa de disparo es dejar al motor en el "aire", como el motor tiene una inductancia trata de mantener la corriente aumentando la tensión. Por consiguiente, hay que tener cuidado en la elección de la duración del tiempo muerto, minimizando su efecto, siendo el típico entre 1 y 3 mseg. El amplificador tipo "H" es implementado con dispositivos conmutadores MOSFET's IRFZ34 a una corriente de 30A y una fuente de 24 voltios a 8A; para disparar los MOSFET's es necesario una circuitería adicional para que genere la tensión de disparo en cada conmutador.

En la figura 4.7 se muestra el circuito de disparo de Q_{14} consiste de dos transistores Q_1 y Q_3 los cuales forman un amplificador que trabaja en clase B, este amplificador es manejado por un circuito buffer de colector abierto que dependiendo del fin de Q_3 absorba las corrientes de fuga de Q_{14} . El circuito de disparo de Q_5 también es un amplificador que trabaja en clase B el cual esta formado por los transistores Q_2 y Q_4 ; Q_4 sirve para absorber las corrientes de fuga de Q_5 . El circuito de disparo de Q_6 y Q_7 son idénticos al funcionamiento de los MOSFET's Q_5 y Q_{14} , respectivamente.

4.2. El Sensado

El sensor de posición esta conformado por un codificador óptico rotatorio (interno al servomotor), un decodificador de cuadratura y un contador UP/DOWN. El codificador óptico rotatorio de la figura 4.8 está conformado por:

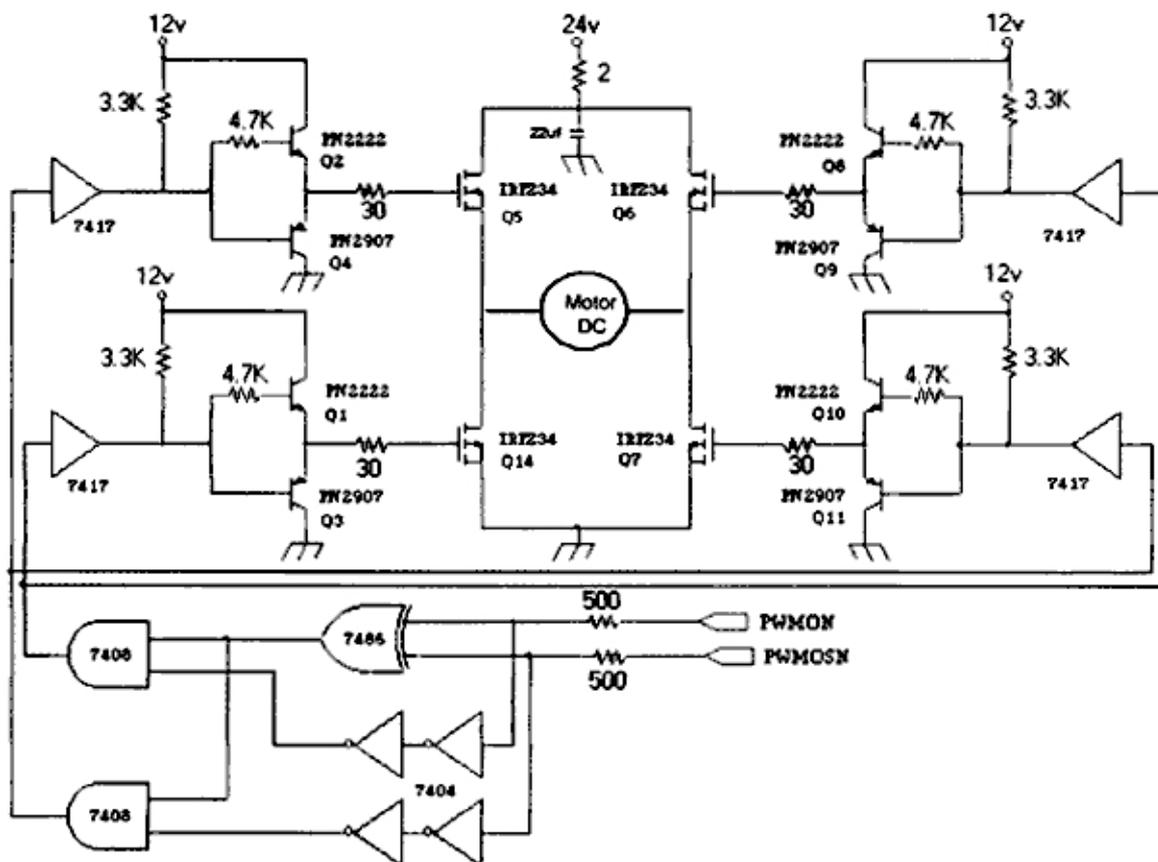


Figura 4.7: Circuito amplificador de potencia de PWM.

- Una fuente de luz que puede ser un LED infrarrojo o láser.
- Un disco ranurado acoplado al eje del servomotor, que se mueve entre una fuente de luz permanente habilitada a un detector. A medida que gira el rotor, el paso de la luz hacia el detector es habilitado e inhabilitado por un patrón de áreas oscuras y transparentes impreso en el disco, produciéndose una señal digital que puede ser fácilmente interpretada por el controlador de movimiento del sistema.
- Un sensor de luz, usualmente es un fototransistor o un fotodiodo.
- Un conformador de pulsos que convierte la señal sinusoidal proveniente de la salida del fotodetector en señal de onda cuadrada.

El codificador usado en esta implementación, como la mayoría de los casos, incluye dos pistas de marcas o ranuras en cuadratura (canales A y B), cada una

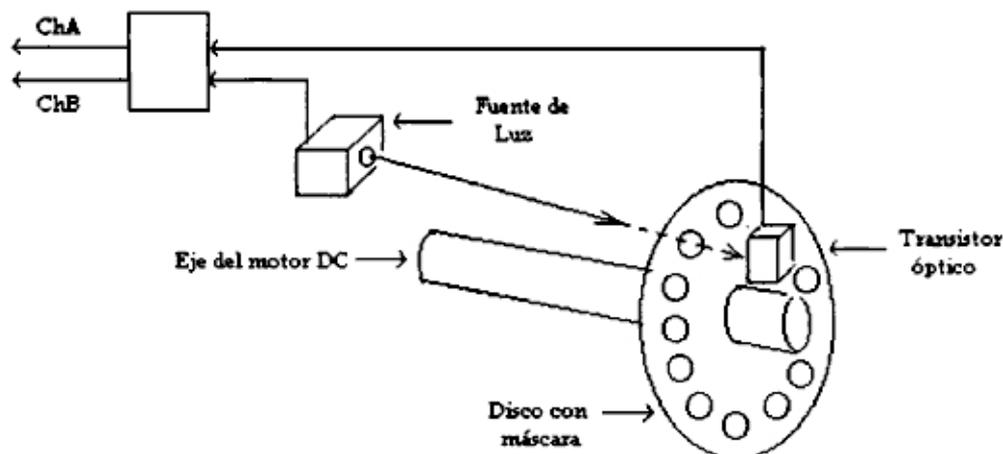


Figura 4.8: Estructura del codificador óptico rotatorio.

asociada a una fuente de luz y un fotodetector independientes. Los canales A y B producen dos ondas cuadradas desfasadas en 90 grados entre sí que se utilizan para determinar la dirección del movimiento. Por ejemplo si el canal B adelanta al canal A, el eje estará girando en el sentido antihorario y viceversa. Si no se utiliza una detección de cuadratura, la vibración puede causar el envío de pulsos con error al contador. El codificador usado tiene una resolución de 100 ppr (pulsos por revolución).

La figura 4.9 muestra al sistema de posición que conforman el decodificador de cuadratura es el CI LS7083, que activará la cuenta ascendente o descendente del contador UP/DOWN de 16 bits, dependiendo del sentido de giro del servomotor, captado por el desfase entre los trenes de pulsos de los canales A y B. El contador de configuración en cascada UP/DOWN los conforman cuatro CI 74HC193; dos de ellos entregan un grupo de 8 bits de datos (bus A) a la tarjeta LABPC+; en forma similar el otro grupo de igual bits conforman el (bus B). $X_4/\overline{X_1}$ son las entradas de resolución de multiplicación. Las salidas digitales PA y PB de 8 bits del bloque contador son datos a procesar por la tarjeta LABPC+.

4.3. Interfaz LABPC+

La tarjeta de adquisición de datos LABPC+ que se muestra en la figura 4.1, es la interfaz empleada para el envío de la señal de control y para la adquisición

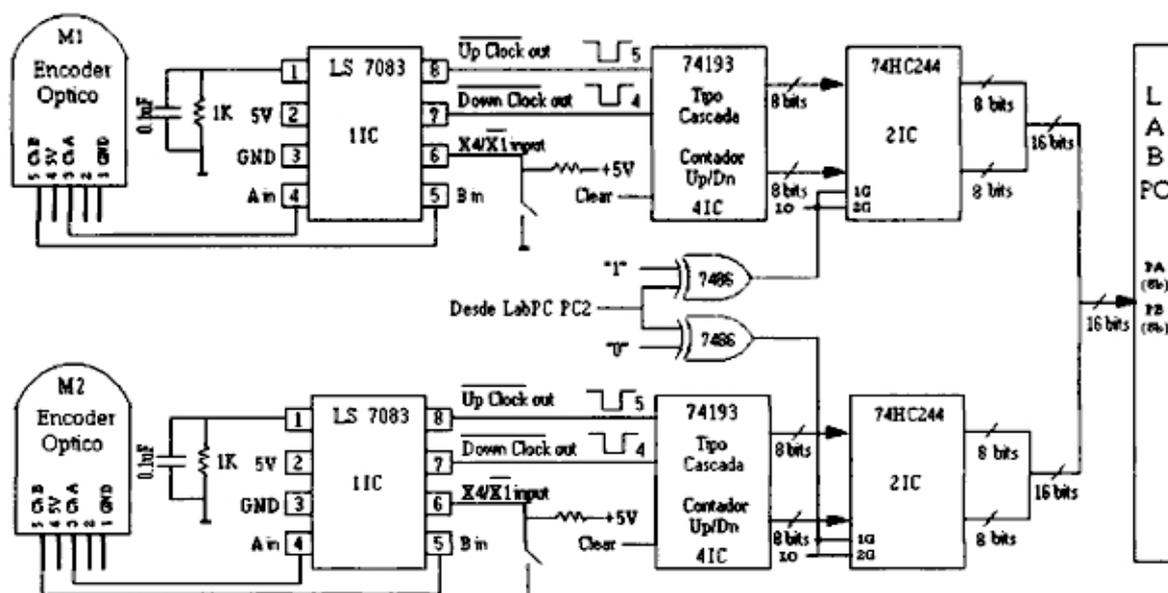


Figura 4.9: Sensado de posición.

de datos con el amplificador y los sensores se realiza en la forma siguiente:

1. Dos salidas análogas (DAC0 y DAC1), estas señales de control envían por los terminales de los CI LM3524 (generador *PWM*) con una resolución de 12 bits para generar la señal analógica.
2. Dos puertos digitales de entrada (PA y PB) de 8 bits cada uno. Estos puertos reciben los datos provenientes del contador UP/DOWN.
3. El terminal OUTB0 que se utiliza como salida de una señal de reloj para el funcionamiento del sistema de control.
4. El terminal PC0 (bit 0 del puerto C) que se utiliza como entrada de la señal de reloj proveniente del OUTB0.

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

4.4. Software de Control

Se utilizó una PC con procesador Pentium de 100MHz. El software fué desarrollado en lenguaje C, el cual consiste en un programa de control adaptivo por modelo referencial. La simulaciones han sido realizadas en MATLAB y luego se tradujeron a lenguaje C para aumentar la velocidad de simulación en el control. El diagrama de flujo se muestra en la figura 4.10.

El diagrama de flujo describe los siguientes pasos:

1. Inicialización de la tarjeta de adquisición de datos LABPC+ y de las variables utilizadas en el programa.
2. Se detecta el flanco de subida del reloj (clock) y se lee el primer dato de la posición inicial, inicializa el algoritmo.
3. Se comprueba que la variable k sea menor que el número de puntos n . La variable n junto con la frecuencia de muestreo determinan el tiempo de duración del experimento.
4. Se detecta un flanco de subida del reloj, se inicia un periodo para la adquisición de datos y control. Se lee la posición y y se aplica restricciones, luego se estima la velocidad.
5. Se calcula la señal de control u , se compensa las tensiones de las fricciones estáticas y de Coulomb v , se aplican restricciones en la señal de control, luego se envía la señal de control $u = u + v + \text{offset}$ por el DAC0 y DAC1 a los generadores de *PWM*.
6. Se actualizan las variables y se repite el lazo de control hasta que k alcance a n .

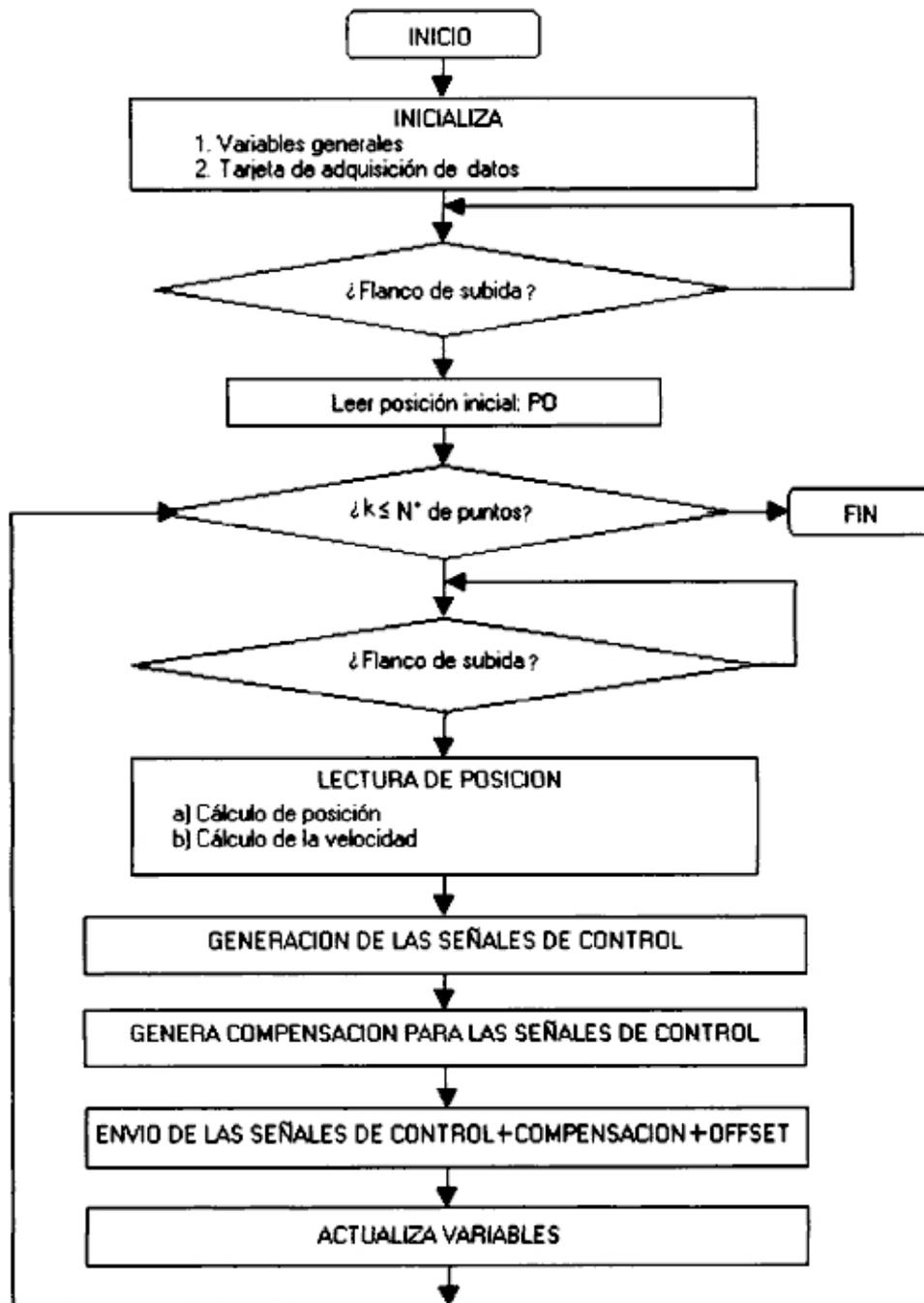


Figura 4.10: Diagrama de flujo del algoritmo de control.

Capítulo 5

Simulaciones

En esta sección se muestra las simulaciones del control adaptivo para el MRTM de 2DOF. Las simulaciones se obtienen utilizando el programa MATLAB. El desarrollo de las simulaciones se ha dividido en dos partes, la primera parte presenta simulaciones para la posición del móvil y el brazo, la segunda parte presenta simulaciones para trayectorias senoidales.

Los resultados de las simulaciones, son mostrados en las figuras (5.2) hasta (5.10). Los parámetros de sintonización que presentamos cumplen con el objetivo del control; se probaron muchos valores y la gran mayoría cumplía; pero sin embargo en el siguiente capítulo experimental presentamos los valores de sintonización mas precisos. Los valores resultantes fueron $Kd1=10 \cdot I$, $Kp1=160 \cdot I$ para la ley de control $u1$ y $Kd2 = 3 \cdot I$ y $Kp2=37 \cdot I$. para la ley de control $u2$; las figuras que se muestran fueron generalizadas con estos valores. Para el ancho de banda usamos un valor tal que elimine los componentes indeseados de alta frecuencia $\Lambda = 2 \cdot I$. Para la ganancia de adaptación obtuvimos $k1 = 0,05$, $k2 = 0,1$, $k3 = 0,001$, $k4 = 0,040$, $k5 = 15,8$ y $k6 = 0,03$, de tal manera que los parámetros estimados converjan en un tiempo muy corto.

5.1. Experimento 1: Referencia Constante

El tiempo de simulación utilizado es de 15 seg. La figura (5.1) muestra las salidas controladas para la posición del carro y posición angular del brazo, luego de un tiempo de 0.6 seg. para ambos, siguen a la trayectoria deseada. Los valores de consigna para la posición del móvil es de $0,1m$ y para la posición de la varilla es de $\pi/4$ radianes. La figura (5.2) muestra la respuesta del error de posición para el móvil y el brazo, con magnitudes muy pequeñas. La figura (5.3) muestra la respuesta del error de velocidad para el móvil y el brazo, con magnitudes muy pequeñas.

Esta simulación se realizó con el programa adaptivo.m escrito en MATLAB y cuyo listado se muestra en el Apéndice C y se encuentra en el CD adjunto. Para este caso, la referencia usada en el programa, es constante.

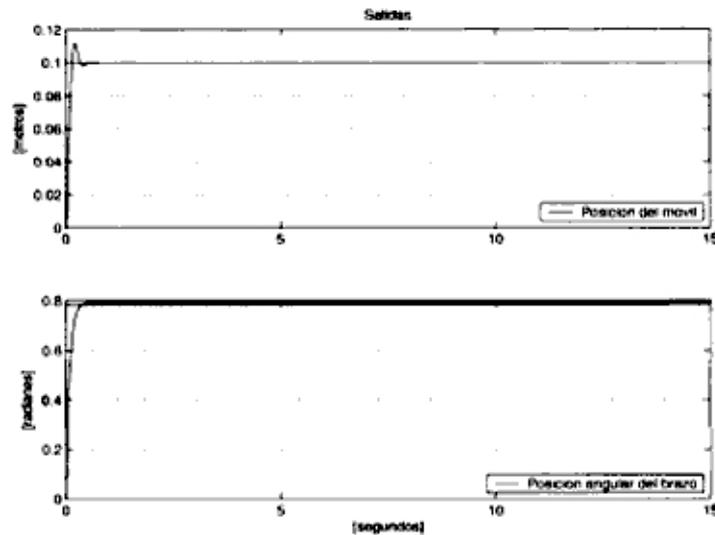


Figura 5.1: Posiciones controladas del carro y del brazo para el caso de referencia constante.

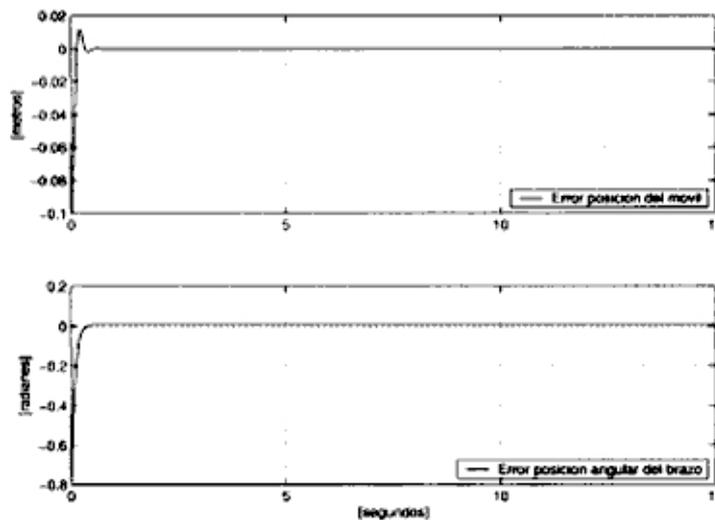


Figura 5.2: Error de posición Móvil-Brazo para el caso de referencia constante.

La figura 5.4 muestra la ley de control que presenta una magnitud grande al inicio del control; la cual es limitada por el algoritmo de control.

Ley de control es susceptible a los parámetros de sintonización.

La figura 5.5 muestra la identificación de parámetros para el caso de referencia constante. Los valores de la matriz de ganancia para la adaptación se escogieron

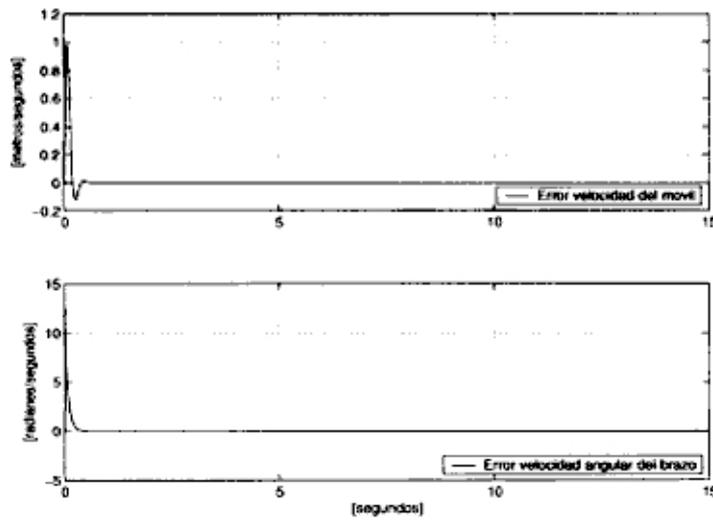


Figura 5.3: Error de velocidad Móvil–Brazo para el caso de referencia constante.

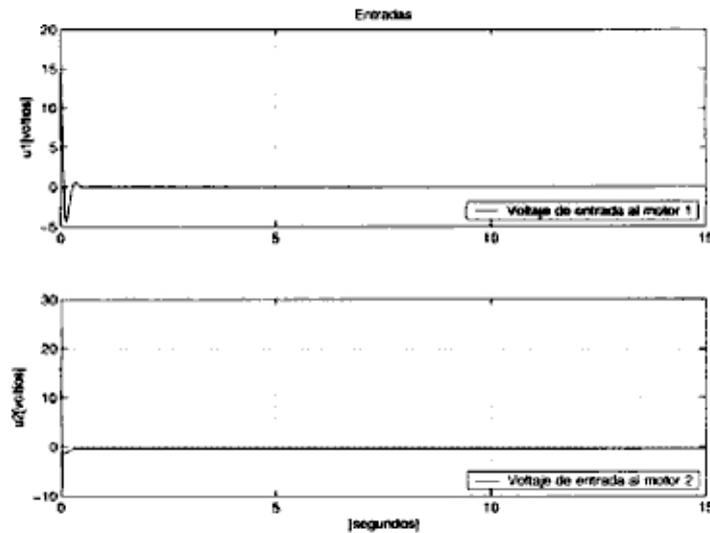


Figura 5.4: Señales de control para el motor 1 y para el motor 2 para el caso de referencia constante.

de tal manera que los parámetros converjan en un mismo rate; estas convergencias no necesariamente tienen que converger al valor real del modelo; pero si encontrar una estabilidad en el sistema a pesar de incertidumbres.

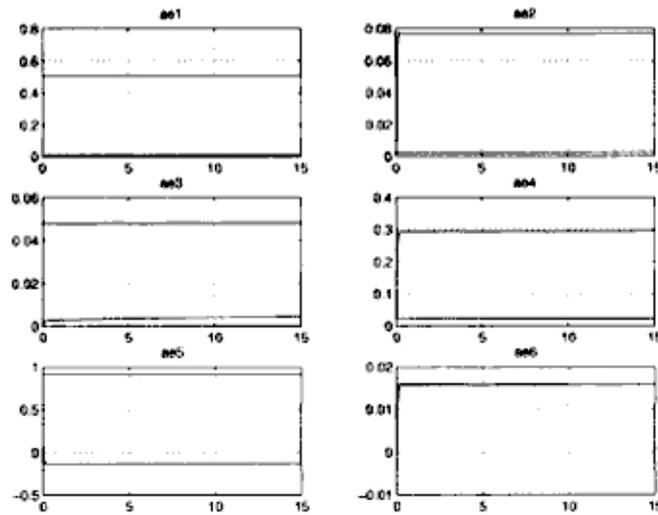


Figura 5.5: Identificación de parámetros para una referencia constante.

5.2. Experimento 2: Referencia Senoidal

Consideramos una trayectoria senoidal con las condiciones iniciales iguales a los casos de posición. La trayectoria deseada para el carro es de $x_{d1} = 0,2\cos(2\pi kT)$ m y la trayectoria deseada para el brazo es $x_{d2} = \pi/4\cos(2\pi kT)$ rad. El tiempo de simulación utilizado es de 15 seg. La figura (5.6) muestra las salidas controladas para la trayectoria del carro y brazo. La figura (5.7) muestra la respuesta del error de trayectoria para el móvil y el brazo, con magnitudes muy pequeñas. La figura (5.8) muestra la respuesta del error de velocidad para el móvil y el brazo, con magnitudes muy pequeñas.

Esta simulación se realizó con el programa `adaptivo.m` escrito en MATLAB y cuyo listado se muestra en el Apéndice C y se encuentra en el CD adjunto. Para este caso, la referencia usada en el programa, es senoidal. La figura 5.9 muestra la ley de control para ambos motores; las cuales se encuentran sobre un rango aceptable en magnitud (± 8 Voltios para el móvil y ± 2 voltios para el brazo). Ley de control es susceptible a los parámetros de sintonización. La figura (5.10) muestra la identificación de Parámetros para el caso de referencia senoidal. Los valores de la matriz de ganancia para la adaptación se escogieron de tal manera que los parámetros converjan en un mismo rate; estas convergencias no necesariamente tienen que converger al valor real del modelo; pero si encontrar una estabilidad en el sistema a pesar de incertidumbres.

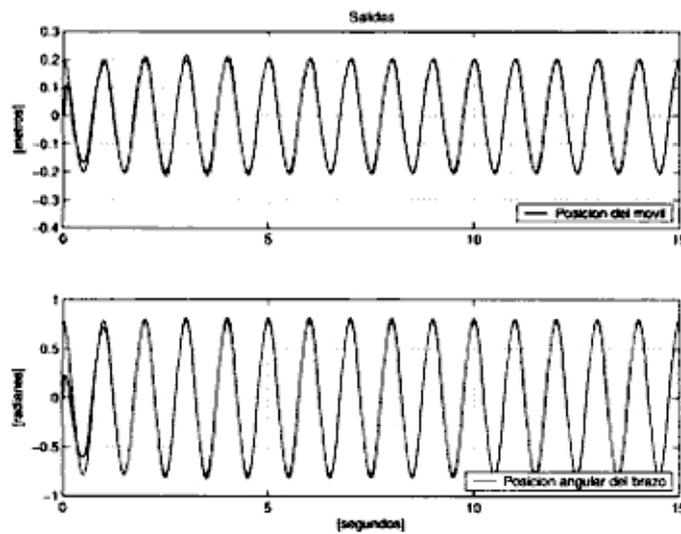


Figura 5.6: Trayectorias controladas del carro y del brazo para el caso de referencia senoidal.

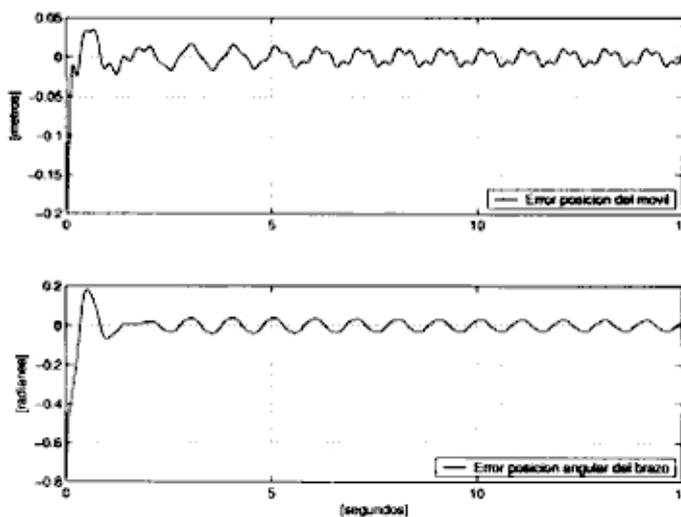


Figura 5.7: Error de posición Móvil-Varilla para el caso de referencia senoidal.

5.3. Animación con la Interface Gráfica GUI

Los programas de simulación anteriormente expuestos, han servido para llevar a cabo los cálculos requeridos y producir el ploteo para trayectorias constantes y senoidales. En esta sección se presenta el diseño de la interfaz gráfica para el usuario (GUI: Graphic User Interface) [9], la misma que combina todas las estrategias desarrolladas en esta sección y permite visualizar tales resultados en un panel de control.

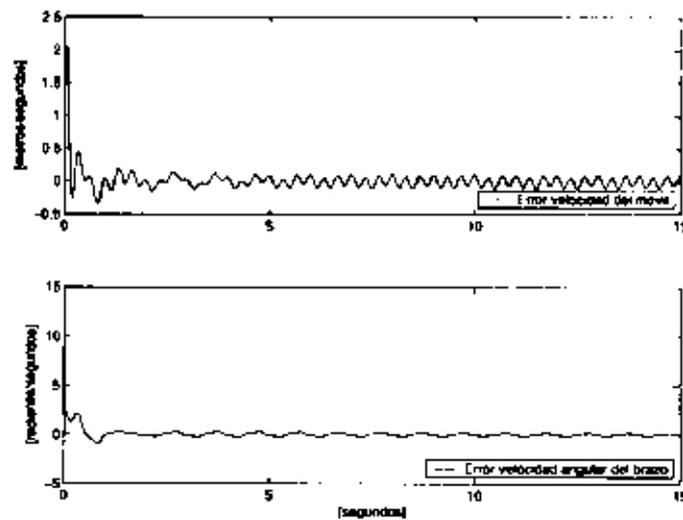


Figura 5.8: Error de velocidad Móvil - Varilla para el caso de referencia senoidal.

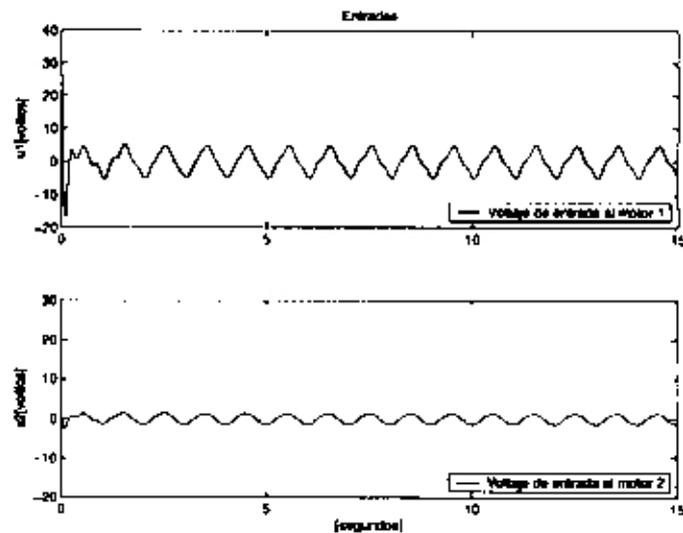


Figura 5.9: Señales de control para el motor 1 y para el motor 2 para el caso de referencia senoidal.

Específicamente, en esta sección se emplea la GUI con dos estrategias de control; la primera estrategia es el control Adaptivo, la cual corresponde a este trabajo de tesis, y la segunda estrategia es el control Sliding (empleado en otra tesis). Ambas estrategias realizan la tarea de controlar al MRTM de 2DOF.

El propósito del uso de la Interfaz del Usuario Gráfica (GUI) , es una ayuda visual de la animación del sistema MRTM de 2DOF para el usuario. Los gráficos generados por la animación son los estados del sistema que involucren la posición y la velocidad, es decir el usuario puede observar la correlación entre el ploteo y

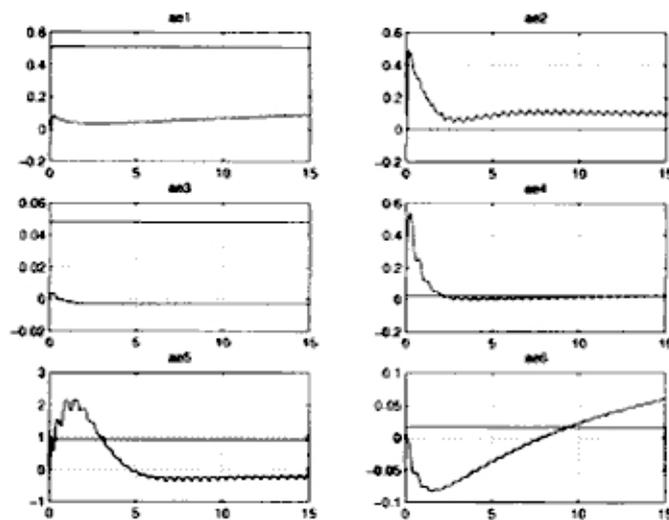


Figura 5.10: Identificación de parámetros para el caso de referencia senoidal.

el sistema físico.

El método del diseño de la interfaz gráfica permite sintonizar los parámetros desde el panel visual y modificarlos en la medida que el controlador lo requiera. En el panel visual de control es posible modificar las condiciones iniciales del sistema y direccionar las consignas en forma arbitraria para la traslación del carro y para la posición angular del brazo que es libre de girar en ambas direcciones.

Para ejecutar la GUI se necesitan 3 archivos desarrollados en MATLAB 6.5 (los productos finales también se pueden ejecutar con versiones anteriores de MATLAB). Los archivos son copiados en un directorio de MATLAB y se ejecutan de acuerdo al siguiente orden:

- El programa *anim_carro.m*, es ejecutado en primer lugar, ninguno de los otros dos archivos necesita ser corrido sin previo inicio del panel visual de control. El programa contiene la interfaz y los controles para el sistema, siendo de gran ayuda para que el usuario pueda manipularlos.
- El programa *update_AnimObjects.m* contiene la GUI de llamada de función para animar objetos. El programa utiliza las propiedades gráficas de los objetos.
- *Adaptive_ctrl.m* - contiene el archivo de simulación para el sistema no lineal. El programa direcciona las datas de los parámetros de sintonía, variables

de estado y la ley de control que serán evaluadas por *anim_carro.m* para generar el movimiento del manipulador robótico.

Uso del Panel de Control

El panel de control se muestra en la figura 5.11. Como se puede observar, el diseño de la interfaz es de fácil uso y accesible para realizar modificaciones desde la ventana principal del computador.

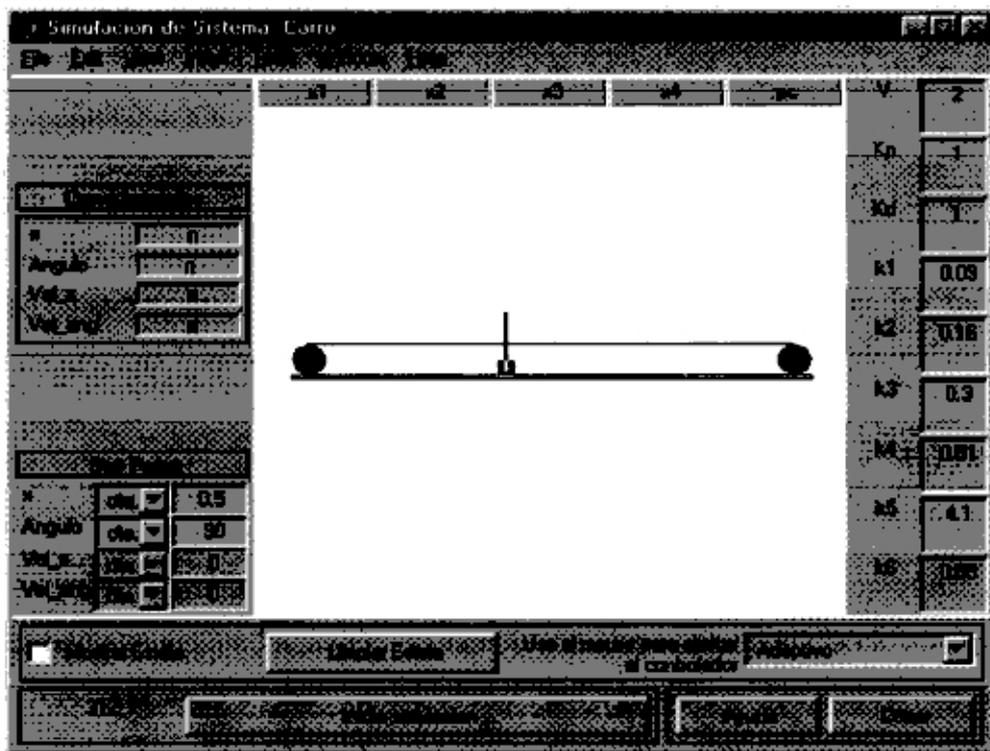


Figura 5.11: Panel de salida de la animación del MRTM de 2DOF.

Los pasos a seguir para el empleo de esta interfaz gráfica son:

1. En la ventana de trabajo de MATLAB emplear el comando *path* para ubicar la carpeta que contiene los archivos de animación. Luego, digitar el nombre del programa *anima_carro*. El programa presenta el panel de control del sistema, tal como se muestra en la figura 5.11.
2. En la parte superior izquierda de dicha figura se encuentran las condiciones iniciales de la posición del carro (x) y de la posición del ángulo del brazo

(*angulo*). Las condiciones iniciales pueden ser cambiadas a voluntad externamente por el usuario.

3. En la parte inferior izquierda de la figura 5.11, se encuentra ubicado el *set point*, para especificar las trayectorias de referencia constante y arbitrarias de las variables controladas x y *angulo*.
4. En las partes superior e inferior del lado derecho, se ubican los parámetros de sintonización del sistema. Las variables que pueden ser cambiadas a voluntad con fines de sintonización del sistema de control son: V , Kp , Kd , $k1$, $k2$, $k3$, $k4$, $k5$ y $k6$, las cuales son parte del algoritmo del controlador Adaptivo que se exponen en este trabajo de tesis.
5. En la parte central está ubicado el esquema del MRTM de 2DOF. En la parte superior de esta ventana se ubican los estados del sistema y los parámetros de sintonización, a saber: la posición del carro x_1 , la posición angular del brazo x_2 , la velocidad del carro x_3 , la velocidad angular del brazo x_4 y pc (empleado para la identificación de parámetros). Todas las variables de estado y parámetros de sintonización son graficadas por el programa.
6. En el extremo inferior izquierdo se ubica el bloque *Iniciar Estela* empleado para visualizar un barrido del sistema de control en movimiento. Este barrido se activa haciendo un check con el mouse en el check box que se encuentra al lado izquierdo del bloque *Iniciar Estela*. Este movimiento puede desactivarse en cualquier momento mediante haciendo click en el bloque *Limpiar Estela*.
7. El bloque *Ayuda* ubicado en la parte inferior derecha invoca al help de MATLAB.
8. El bloque *Iniciar Simulación* localizado en la parte inferior central ejecuta la corrida del sistema controlado.
9. El bloque *Ajuste del Controlador* ubicado en la parte inferior derecha, muestra dos posibilidades de operación del controlador del sistema. Por defecto, aparece la estrategia de control *Adaptivo*. El cursor ubicado a la derecha permite cambiar la estrategia de control a voluntad. La segunda estrategia de control *Sliding*, es la que corresponde al trabajo de otra tesis.

10. El bloque *Cerrar* localizado en la parte inferior derecha permite cerrar el panel de control descrito.

Todos los programas empleados para animar los sistemas de control diseñados se encuentran en el CD-ROM adjunto a este trabajo de tesis.

Capítulo 6

Resultados Experimentales

Los resultados experimentales para el sistema MRTM de 2DOF, son mostrados en las figuras (6.1) – (6.8). Para implementar el controlador adaptivo, necesitamos especificar los valores del ancho de banda Λ y de la matriz de ganancia de la adaptación Γ . El ancho de banda debe ser sintonizado para que los elementos de frecuencia alta no sean considerados por el controlador. Para seleccionar Γ , se desea que cada uno de los parámetros converja aproximadamente en el mismo rate. Los componentes *PD* fueron sintonizados tan alto sea posible con mínimas oscilaciones en el seguimiento de la trayectoria. Todos los valores de sintonización fueron hallados por el método de prueba y error hasta conseguir la mejor performance al sistema; sin embargo en las conclusiones se muestra algunos métodos que nos permiten acercarnos a estos valores analíticamente. Los valores resultantes fueron $Kd1=100 \cdot I$, $Kp1=1600 \cdot I$ para la ley de control $u1$ y $Kd2 = 4 \cdot I$ y $Kp2=37 \cdot I$. para la ley de control $u2$; las figuras que se muestran fueron generalizadas con estos valores. Para el ancho de banda usamos un valor tal que elimine los componentes indeseados de alta frecuencia $\Lambda = 2 \cdot I$. Para la ganancia de adaptación obtuvimos 02 matrices para la posición y 01 matriz para trayectoria en la cual obtuvimos resultados muy satisfactorios. Matriz de ganancia para la posición (ángulos del brazo menor que π). $k1 = 0,05$, $k2 = 0,1$, $k3 = 0,001$, $k4 = 0,040$, $k5 = 15,8$ y $k6 = 0,03$. Matriz de ganancia para la posición (ángulos del brazo mayor que π). $k1 = 0,05$, $k2 = 0,1$, $k3 = 0,0001$, $k4 = 0,040$, $k5 = 15,8$ y $k6 = 0,03$. Matriz de ganancia para trayectorias. $k1 = 0,05$, $k2 = 0,1$, $k3 = 0,01$, $k4 = 0,040$, $k5 = 15,8$ y $k6 = 0,03$. El desarrollo de los experimentos se estructuran en 02 experiencias la primera para la posición y la segunda para trayectoria.

6.1. Experimento 1: Referencia Constante

El software de control para este experimento es el programa `Adaptivo.cpp`, escrito en C++ de Borland (versión 3.2), cuyo listado se muestra en el Apéndice D y se encuentra en el CD adjunto a esta tesis. Para el caso que nos ocupa, en este programa se considera una referencia constante. El tiempo de experimentación utilizado es de 1 seg. La figura (6.1) muestra las salidas controladas para la posición del carro y posición angular brazo, luego de un tiempo de 0.4 seg. y 0.5 seg. respectivamente, siguen a la trayectoria deseada. Los valores de consigna para la posición del móvil es de 10 cm y para la posición de la varilla es de 45 grados.

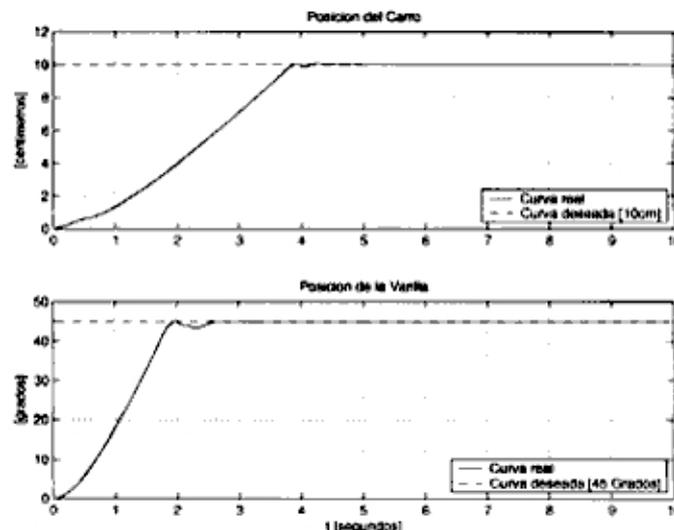


Figura 6.1: Posiciones controladas del carro y el brazo para el caso de referencia constante.

La figura (6.2) muestra la respuesta del sistema para la velocidad del carro y velocidad angular del brazo, el tiempo de convergencia para el móvil es de 0.5 seg y para la varilla es de 0.35 seg. La figura (6.3) muestra la ley de control que presenta una magnitud grande al inicio del control, la cual es limitada por el algoritmo de control. La figura 6.3 muestra las señales de control para el motor 1 y para el motor 2 para el caso de referencia constante que nos ocupa La figura 6.4 muestra la identificación de parámetros para el caso de referencia constante. Los valores de la matriz de ganancia para la adaptación se escogieron de tal manera que los parámetros converjan en un mismo rate; estas convergencias no necesariamente tienen que converger al valor real del modelo; pero si encontrar

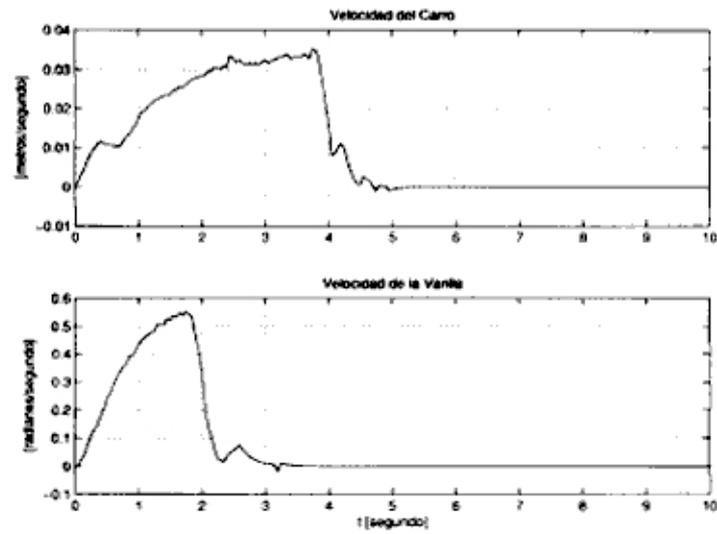


Figura 6.2: Velocidades controladas del carro y el brazo para el caso de referencia constante.

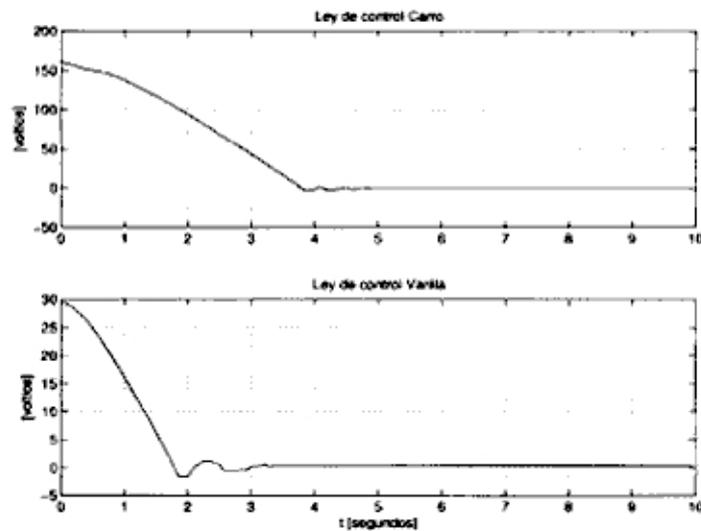


Figura 6.3: Señales de control para el motor 1 y para el motor 2 para el caso de referencia constante.

una estabilidad en el sistema a pesar de incertidumbres.

6.2. Experimento 2: Referencia Senoidal

El software de control para este experimento es el programa `Adaptivo.cpp`, escrito en C++ de Borland (versión 3.2), cuyo listado se muestra en el Apéndice D y se encuentra en el CD adjunto a esta tesis. Para el caso que nos ocu-

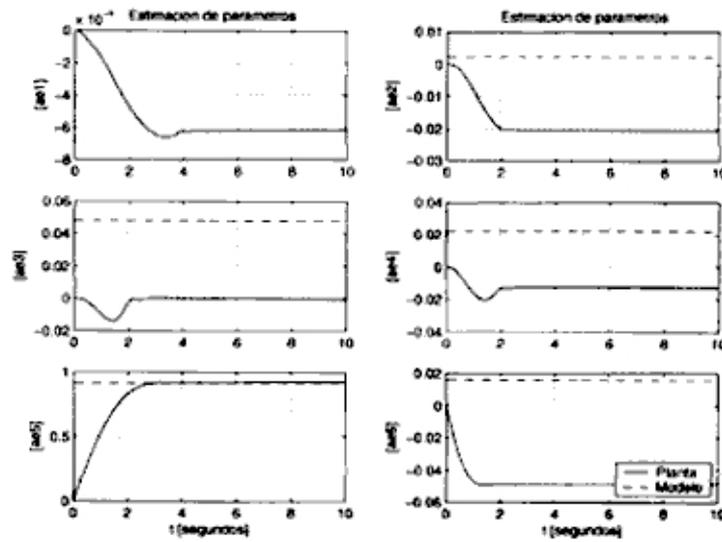


Figura 6.4: Identificación de parámetros para el caso de referencia constante.

pa, en este programa se considera una trayectoria senoidal con las condiciones iniciales iguales a los casos de posición. La trayectoria deseada para el carro es de $x_{d1} = 0,2\cos(2\pi kT/45)\text{m}$ y la trayectoria deseada para el brazo es $x_{d2} = \pi/4\cos(2\pi kT/45)\text{rad}$. La frecuencia se dividió entre 45 para evitar que el seguimiento sea demasiado rápido no dando tiempo de respuesta al sistema mecánico (Manipulador).

La figura (6.5 muestra la salida controlada realizando un seguimiento a la señal deseada. Las señales se han fijado a una frecuencia de $f = \frac{100}{45}\text{Hz}$ de ciclo senoidal. Las amplitudes de las señales senoidales indican el movimiento carro con un barrido horizontal sobre la riel haciendo un recorrido de 40cm y para el brazo un barrido vertical izquierda derecha con un barrido de 180 grados. La figura (6.6) muestra la respuesta del sistema para la velocidad del carro y velocidad angular del brazo. Se observa que las amplitudes para la velocidad son pequeñas, estas amplitudes son sensibles a los parámetros que sintonizan al controlador. La figura (6.7) muestra las señales de control para el carro y el brazo para el caso de trayectoria senoidal. La figura 6.8 muestra la identificación de parámetros para el caso de trayectoria senoidal.

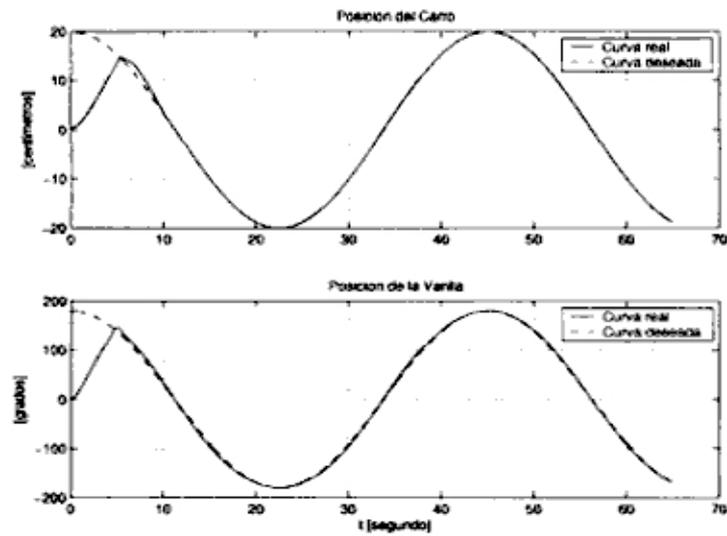


Figura 6.5: Trayectorias controladas del carro y del brazo para el caso de trayectoria senoidal.

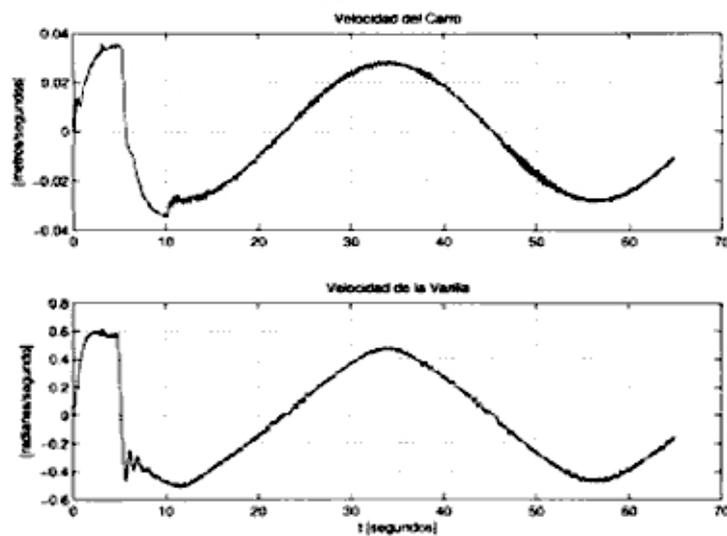


Figura 6.6: Velocidades controladas del carro y del brazo para el caso de trayectoria senoidal.

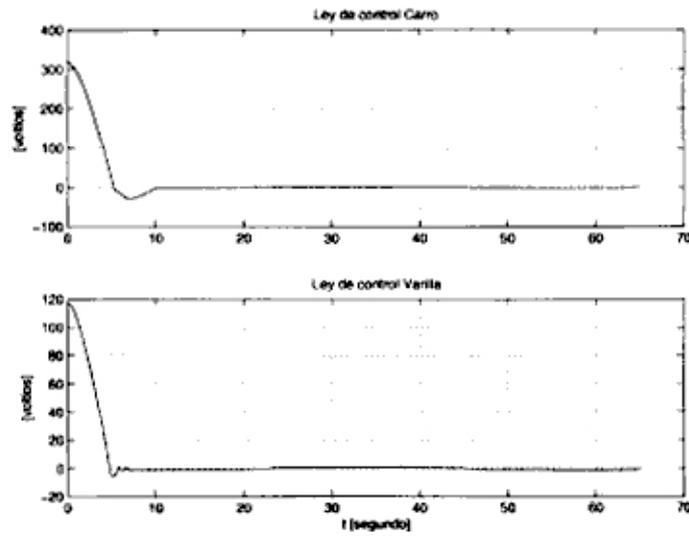


Figura 6.7: Señales de control para el carro y el brazo para el caso de trayectoria senoidal.

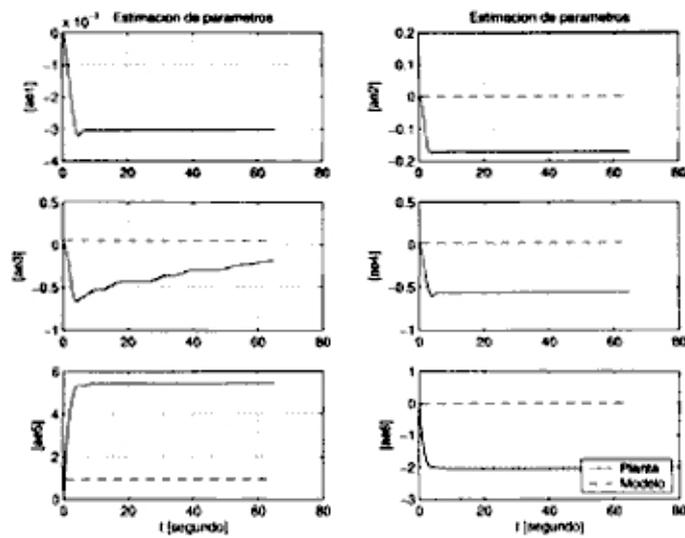


Figura 6.8: Identificación de parámetros para el caso de trayectoria senoidal.

Capítulo 7

Conclusiones

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

7.1. Conclusiones

1. El proceso manipulador translacional ha sido construido con el propósito de implementar los experimentos relacionados con el sistema de control adaptivo con modelo referencial. En su construcción se ha optado por utilizar materiales de aluminio en la fabricación del móvil, poleas y brazo. En lo que respecta a la parte mecánica, se ha tomado en cuenta el buen diseño de las bocinas para evitar posibles desajustes que puedan ocasionar errores en la experiencia. El proceso construido se encuentra disponible en la SPGSE-FIEE para futuros experimentos.
2. Ya es conocido que no es posible obtener un modelo matemático exacto del proceso. En nuestro caso, se ha determinado un modelo dinámico no lineal multivariable correspondiente al manipulador robótico translacional. Según los resultados obtenidos en la simulación, nos indica que el modelo empleado ha sido satisfactorio.
3. El problema planteado fue el de controlar, el móvil y el brazo usando control adaptivo con modelo de referencia. La entrada al sistema se basa en un control por realimentación en adelanto mas los términos proporcional y derivativo, de tal manera que la estabilidad está asegurada de acuerdo a la Teoría de Lyapunov. Por otro lado las simulaciones efectuadas en el Capítulo 5 validan el método de diseño empleado. Para obtener resultados experimentales exitosos mostrados en el Capítulo 6, ha sido necesario sintonizar los parámetros para obtener respuestas de las variables contro-

ladas que cumplan las especificaciones de diseño impuestas. Los valores resultantes de sintonización para las ganancias de la ley de adaptación (ver ecuación (3.48)) son: $k_1=0.05$, $k_2=0.1$, $k_3=0.0001(+180)$, $k_3=0.001(-180)$, $k_4=0.040$, $k_5=15.8$, $k_6=0.03$. Los parámetros óptimos del ancho de banda (ver ecuación (3.41)) son: $I=1$, $V=2*I$, mientras que las ganancias derivativa y proporcionales resultaron: $Kd1=100.*I$, $Kp1=1600.*I$, $Kd2 = 1.*I$, y $Kp2=20.*I$, respectivamente (ver (3.41)). Estos parámetros hacen que la ley de control sea capaz de estabilizar las salidas del proceso con respecto a las trayectorias deseadas.

4. La implementación del sistema de control diseñado, también necesitó que se perfeccionarán los diseños de las tarjetas electrónicas siguientes: el generador de PWM y el amplificador de potencia. Los circuitos correspondientes a los diseños finales se muestran en el Capítulo 4. Los experimentos realizados han demostrado el buen desempeño de estas tarjetas.

7.2. Recomendaciones

Existe una manera analítica para determinar los parámetros de sintonización que no fueron utilizados en esta tesis; sin embargo, los cito para que puedan ser usados en trabajos posteriores. Algunas guías básicas para seleccionar el ancho de banda son:

$$\Lambda \leq \Lambda_R \approx \frac{2\pi}{3} \Lambda_R$$

$$\Lambda \leq \Lambda_A \approx \frac{1}{3T_A}$$

$$\Lambda \leq \Lambda_S \approx \frac{1}{5} \Lambda_{sampling}$$

Donde Λ es seleccionado para ser el mínimo de los tres limites propuestos anteriormente. Estos limites se relacionan con la estructura no modelada, retardos de tiempo de la dinámica del actuador y otros retardos, limitación en el tiempo de muestreo. Encontramos también el método para seleccionar la base para los valores iniciales de la matriz Γ desde las cuales podemos sintonizar una mejor performance.

$$\Gamma \propto \text{diag} \left(\int_0^T Y^T Y dt \right)^{-1}$$

donde Y es la matriz de valores conocidos. Esta relación puede ser vista como una regla identificación de mínimos cuadrados

Algunas de las áreas que necesitan investigarse, partiendo del proceso construido (el manipulador robótico translacional) se listan a continuación.

- Utilizar la planta para resolver nuevas estrategias de control avanzado (predictivo no lineal, adaptivo-predictivo, robusto, entre otros) o controladores basados en inteligencia artificial (fuzzy logic, redes neuronales, sistemas experto).
- Adicionar un grado de libertad a la planta, para obtener una planta MIMO de 3 entradas y 3 salidas y agregar un efector final (una pinza de dos grados de libertad, por ejemplo). En esta forma, los diseños de los sistemas de control correspondientes pueden ya ser orientados a aplicaciones industriales.
- También se requiere investigar el caso cuando el manipulador robótico translacional presenta en su construcción eslabones y juntas flexibles.
- Utilizar un DSP (procesador digital de señales), u otro dispositivo programable, para que el sistema de control sea parte del manipulador robótico.

Apéndice A

Identificación de Parámetros

Por identificación se entiende la formulación de un modelo que permita representar al sistema con todas sus propiedades y características relevantes. En nuestro caso determinaremos los parámetros de los servomotores para el sistema MRTM de 2DOF, estos parámetros hallados, nos permitirán validar los resultados experimentales. En la sección desarrollaremos el modelo matemático del sistema para realizar la identificación, basándonos en el análisis del servomotor DC expuesto en la sección anterior. Un modelo matemático es un conjunto de ecuaciones que envuelven las variables del sistema que proporcionan una representación óptima al sistema. En general, si realizamos un experimento en que se aplica ciertos valores a las entradas del sistema, obtendremos ciertas respuestas en sus salidas; de ese modo dando los mismos valores que en el experimento a las variables independientes de nuestro modelo, podemos esperar que después de resolver el conjunto de ecuaciones la respuesta del modelo deberá tener respuestas similares a aquellas salidas del sistema físico.

A.1. Identificación de Parámetros del Servomotor

Para identificar los parámetros de un servomotor DC (sin carga) es necesario tener la edición de un programa en C para la generación de un archivo de datos de la velocidad del servomotor en lazo abierto. Los archivos usados para la experiencia en tiempo real son:

- *adqs.cpp*, contiene el programa principal para adquisición de la data de la velocidad. Este programa es desarrollado en lenguaje C.
- *adq_d1.cpp*, contiene el programa de la configuración de la tarjeta LABPC+ que trabaja con la librería *Nidaqbc.lib*.

- *adq_d1.h*, contiene las funciones para el manejo de la tarjeta LABPC+.
- *Nidaqbc.lib* (contiene las librerías de funciones para el manejo de la tarjeta LABPC+.

La creación del proyecto se realiza por medio del archivo *adqs.prj*, incluye los archivos: {*adqs.cpp*, *adq1.cpp* y *Nidaqbc.lib*}.

La ejecución del archivo *prj* (ejecución en tiempo real), generan el archivo de datos *adqs24.dat* para el servomotor. El archivo de datos es utilizado para obtener la curva de velocidad que se muestra en la figura A.1. El uso de un método de identificación para determinar los parámetros del servomotor nos conduce a hallar la resistencia de armadura R_a , constante de torque de servomotor K_m , constante de fuerza contraelectromotriz K_b , fricción viscosa B y momento de inercia J .

A.2. Curva de la Velocidad para la Identificación

Usaremos el método gráfico de aproximación de la respuesta del proceso a un sistema de primer orden, cuando es despreciada la inductancia de armadura L_a . Los servomotores que utilizaremos son de fabricación Matsushita modelo GMX 6MPO13A con encoder incremental, disponemos de la información (ver tabla A.1) del servomotor Matsushita con encoder incremental. Para obtener los datos necesarios como son la amplitud K_{med} y a constante de tiempo T_{med} de la curva de velocidad, trazamos una recta tangente en el punto de inflexión, tal como se muestra en la figura A.1, posteriormente trazamos las perpendiculares hacia los ejes coordenados que cortaran a estos en dos puntos, cuyos valores son: $K_{med} = 9,1\text{rad/seg.}$ y $T_{med} = 0,35\text{seg.}$

Disponemos de la información del servomotor Matsushita Modelo GMX-6MPO13A con encoder incremental; Alimentación 24VDC/0.2A; Velocidad sin carga 310RPM; Reducción 12.5:1. La tabla A.1 muestra los valores que corresponden a las pruebas de laboratorio a 24VDC.

Siguiendo el mismo análisis realizado para los subsistemas de los modelos de los servomotores M_1 y M_2 , y tratándose de servomotores de la mismo modelo de fabricación, se realizó la experiencia en tiempo real para el servomotor M_2 que está acoplado al carro, es decir, los parámetros para el servomotor M_2 serán los mismos para el servomotor M_1 .

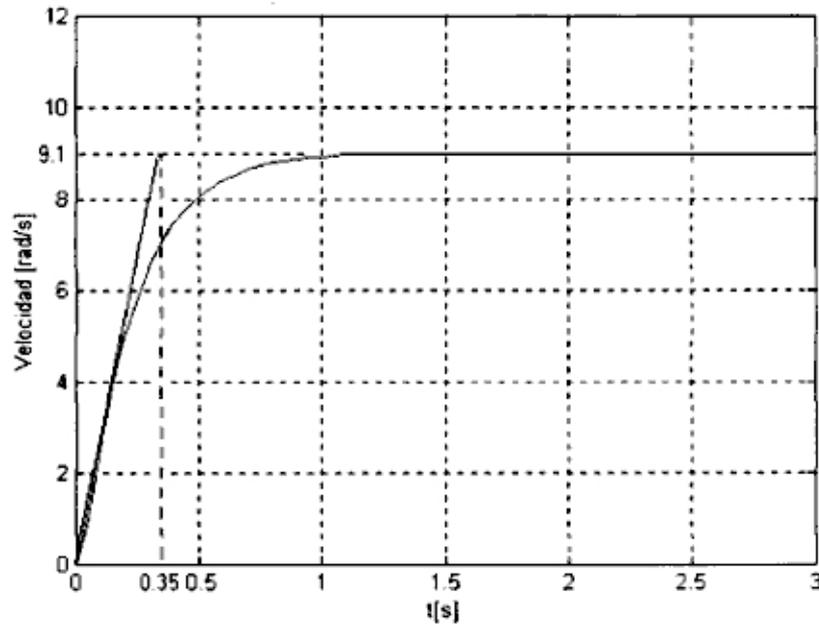


Figura A.1: Curva S.

Cuadro A.1: Especificaciones del servomotor DC.

Velocidad (W) [rad/seg]	Torque (nT_m) [Nm]	Corriente [A]
32.46	N.L.	0.2
29.84	0.2118	0.5
27.22	0.5507	1.0
24.39	0.8826	1.5
21.46	1.2710	2.0
19.37	1.5534	2.5
15.70	1.8712	3.0
10.47	2.0336	3.5
6.28	2.1466	4.0
STALL	2.3725	4.5

Usualmente un modelo lineal de segundo orden para un servomotor DC se expresa mediante la función de transferencia de la velocidad de un servomotor DC dada por.

$$\frac{W(s)}{V_a(s)} = \frac{K_m}{J_{eq}L_a s^2 + (L_a B_{eq} + R_a J_{eq})s + (R_a B_{eq} + K_m K_b)} \quad (A.1)$$

Si omitimos la inductancia, es decir $L_a = 0$, entonces (A.1) se convierte es una

función de transferencia de primer orden de la forma.

$$\frac{W(s)}{V_a(s)} = \frac{K_m}{R_a J_{eq} s + (R_a B_{eq} + K_m K_b)} = \frac{K_{med}}{T_{med} s + 1} \quad (A.2)$$

donde.

$$K_{med} = \frac{K_m}{R_a B_{eq} + K_m K_b} \quad (A.3)$$

y

$$T_{med} = \frac{R_a J_{eq}}{R_a B_{eq} + K_m K_b} \quad (A.4)$$

La resistencia R_a se obtiene de la siguiente relación.

$$R_a = \frac{V}{I_{aSTALL}} = \frac{24V}{4,5A} = 5,3\Omega \quad (A.5)$$

La constante contra electromotriz K_b se obtienen de acuerdo con la tabla A.1 y considerando la inductancia $L_a = 0$.

$$K_b = \frac{V_a - R_a I_a}{nW} = \frac{24V - 5,3\Omega \times 0,2A}{12,5 \times 32,46 rad/s} = 0,0565 V/rad/s \quad (A.6)$$

La constante de torque K_m se obtienen de acuerdo con la tabla A.1;

$$nT_{m|STALL} = 2,3725 Nm$$

El factor de reducción es 12.5, entonces.

$$T_{m|STALL} = 0,1898 Nm$$

La constante del motor se determina a partir de la relación.

$$K_m = \frac{T_{mSTALL}}{I_{aSTALL}} = \frac{0,1898 Nm}{4,5A} = 0,0421 Nm/A \quad (A.7)$$

Reemplazamos los valores $K_{med} = 9,1$ y $T_{med} = 0,35$ que fueron obtenidos a partir de las pruebas de laboratorio para la velocidad en lazo abierto, utilizando la tarjeta de adquisición de datos LABPC+, usando los resultados de R_a y K_m de las ecuaciones (A.5) y (A.7) y dividiendo (A.4) de (A.3) obtenemos la expresión para J_{eq} .

$$J_{eq} = \frac{T_{med} K_m}{K_{med} R_a} = \frac{0,35 \times 0,0421}{9,1 \times 5,3} = 3,0551 \times 10^{-4} Kgm^2 \quad (A.8)$$

La constante de fricción viscosa obtenemos reemplazando los valores de K_{med} , K_m y R_a en la ecuación (A.3), luego despejamos B_{eq} .

$$B_{eq} = \frac{K_m/K_{med} - K_m^2}{R_a} = \frac{0,0421/9,1 - 0,0421^2}{5,3} = 5,3848 \times 10^{-4} Nm/rad/s \quad (A.9)$$

Finalmente se puede determinar la ganancia del amplificador K_A , por medición directa de la tensión en los terminales de armadura del servomotor (salida del amplificador), y considerando la tensión de entrada u de 1.4 voltios.

$$K_A = \frac{V_{aMEDIDO}}{u} = \frac{9,1V}{1,4V} = 5,5 \quad (A.10)$$

A.3. Coeficiente de Fricción del Carro

El concepto de fricción estática puede entenderse mejor considerando el diagrama mostrado en figura A.2. La figura A.2 representa el carro de peso W que descansan en un plano horizontal (el peso W considera los pesos del carro, servomotor y brazo). Una fuerza F_1 que tiende a producir el movimiento hacia la derecha. El movimiento se opone a una fuerza F conocida como fuerza de fricción del carro. F es siempre paralela al plano y opuesta al movimiento. El peso es soportado por la fuerza normal N (el peso de N es siempre perpendicular a la superficie de contacto). Todas las fuerzas son mostradas mediante vectores simples y el objetivo es el equilibrio.

F es una fuerza pasiva Es decir, aumenta F_1 hasta un máximo F_m conocido como la fuerza friccional máxima. F_m es determinado por el peso del objeto y la naturaleza de las superficies en el contacto. Si F_1 es mayor que F_m , el cuerpo se moverá. F_m es igual a la fuerza mínima, mientras se omiten las fuerzas inerciales, necesarias para que el objeto empiece a moverse.

Fricción Deslizante

La fricción deslizante es la fuerza requerida para mantener un movimiento relativo entre dos cuerpos. La figura A.2 puede ser usada para representar el bloque

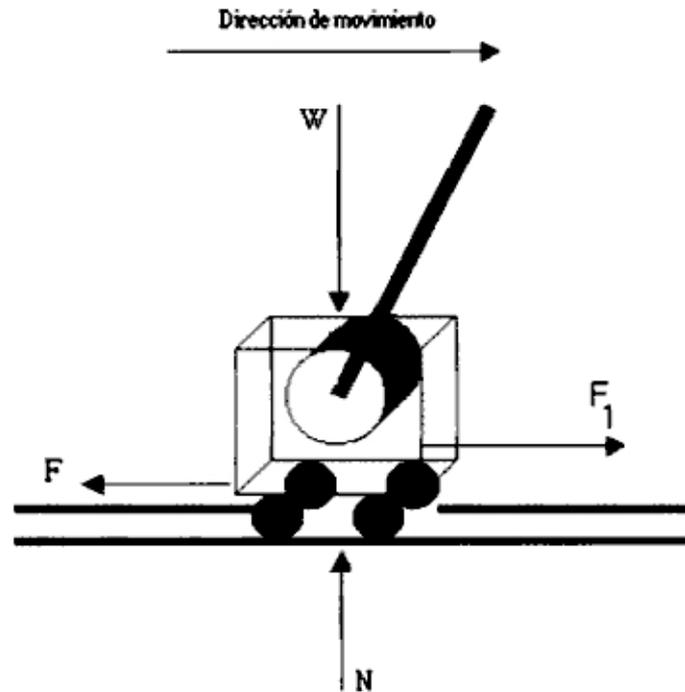


Figura A.2: Dirección de movimiento.

deslizante con una velocidad constante en el plano horizontal. A velocidades bajas, la fricción es independiente de la velocidad deslizante. A velocidades más altas, la fricción disminuye con el aumento de velocidad. La fricción deslizante normalmente es menor que la fricción estática; por consiguiente, los coeficientes de fricción deslizante son normalmente más pequeños que los coeficientes de fricción estática.

Valor Experimental del Coeficiente de Fricción

El coeficiente de fricción es determinado experimentalmente y puede variar dependiendo de las superficies de contacto y las condiciones particulares bajo la cual se han medido. La tabla A.2 es considerada como una guía general [4].

Sean las ecuaciones de movimiento independientes.

$$\begin{aligned}
 F_1 &= F_m \\
 F_m &= fN
 \end{aligned}
 \tag{A.11}$$

la ecuación en el equilibrio.

$$\sum F_y = 0 = N - W \quad (\text{A.12})$$

la ecuación de la fuerza friccional máxima queda expresada en función de la constante de fricción F_c y la velocidad v .

$$vF_c = fN \quad (\text{A.13})$$

luego, la constante de fricción queda expresada.

$$F_c = \frac{fN}{v} \quad (\text{A.14})$$

Los valores de W y v se asignan de acuerdo a las mediciones realizadas experimentalmente, tal es el caso del peso $W = M_c + m = 1,7011Kg$ (ver tabla 2.2) y para calcular la velocidad, tenemos la longitud que recorre el carro de $0,9m$ (longitud máxima de la riel) que recorre en un tiempo de $2,8seg$, de este modo tenemos la velocidad promedio de $0,32m/seg$. Para el coeficiente de fricción, se considera el caso de material metal sobre metal con una condición seca y para un tipo de fricción deslizante, el valor asignado es $f = 0,054$ (ver tabla A.2). Con estos datos reemplazamos en (A.14) para obtener el coeficiente de fricción del carro F_c .

$$F_c = \frac{0,054(1,7011)(9,8)}{0,32} = 2,81Kg/seg$$

A.4. Coeficiente de Fricción Viscosa del Brazo

La obtención del coeficiente de fricción viscosa del brazo, tiene que ver con el modelamiento y compensación de la fricción de Coulomb. La fricción de Coulomb es el torque requerido para mantener al servomotor en movimiento y es una fricción que se opone al movimiento. Se ha comprobado experimentalmente que la fricción de Coulomb puede ser considerada constante. El efecto de la fricción de Coulomb es modelado como un torque constante proporcional al signo de la velocidad, es decir.

Cuadro A.2: Coeficiente de fricción f .

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

$$T_{fc} = \begin{cases} -T_c & \dot{q} < 0 \\ T_c & \dot{q} > 0 \end{cases} \quad (\text{A.15})$$

donde T_{fc} es el torque de fricción de Coulomb, \dot{q} es la velocidad del eje del servomotor.

El modelo discreto para el servomotor DC ($L = 0$) considerando la fricción de Coulomb se deducen a partir de las siguientes ecuaciones.

$$T_m = J_{eq}\dot{w} + B_{eq}w + f^*(nw) \quad (\text{A.16})$$

$$T_m = K_m I_a \quad (\text{A.17})$$

$$I_a = \frac{V_b K_A}{R_a} - \frac{nK_b w}{R_a} \quad (\text{A.18})$$

Reemplazamos (A.18) en (A.17), luego igualamos con (A.16).

$$J_{eq}\dot{w} = \frac{V_b K_A K_m}{R_a} - (B_{eq} + \frac{nK_b w}{R_a}) - f^*(nw) \quad (\text{A.19})$$

Las señales en tiempo discreto surgen si el sistema involucra la operación de muestreo de señales continuas. Sea w_k la velocidad angular muestreada en el k -ésimo instante de tiempo. Usamos la aproximación de Euler $T_s \dot{w} = w_{k+1} - w_k$ para discretizar la ecuación (A.19), obteniendo:

$$w_{k+1} = [1 - \frac{T_s}{J_{eq}}(B_{eq} + \frac{nK_m K_b}{R_a})]w_k + \frac{T_s K_m K_A}{J_{eq} R_a} [V_{bk} - \frac{R f^*(n w_k)}{K_m K_A}] \quad (\text{A.20})$$

donde $f^*(nw_k)$ modela el torque de fricción de Coulomb; y podemos asumir que la aproximación es dada por $f^*(nw_k) = V_{zm} \text{sgn}(nw)$ (V_{zm} es la mitad del ancho de la zona muerta que será determinada más adelante). Si la velocidad es distinta de cero, el torque tomará dos valores constantes dependiendo del sentido de giro del servomotor.

La ecuación (A.20) se puede expresar como:

$$w_{k+1} = \left[1 - \frac{T_s}{J_{eq}} \left(B_{eq} + \frac{nK_m K_b}{R_a}\right)\right] w_k + \frac{T_s K_m K_A}{J_{eq} R_a} \left[V_{bk} - \frac{RC \text{sgn}(nw_k)}{K_m K_A}\right] \quad (\text{A.21})$$

El diagrama de bloques de la figura A.3 muestra al modelo del servomotor discreto. El diagrama involucra la compensación de la fricción de Coulomb determinada por el parámetro G_3 , que será objeto de análisis para determinar el coeficiente de fricción viscosa C .

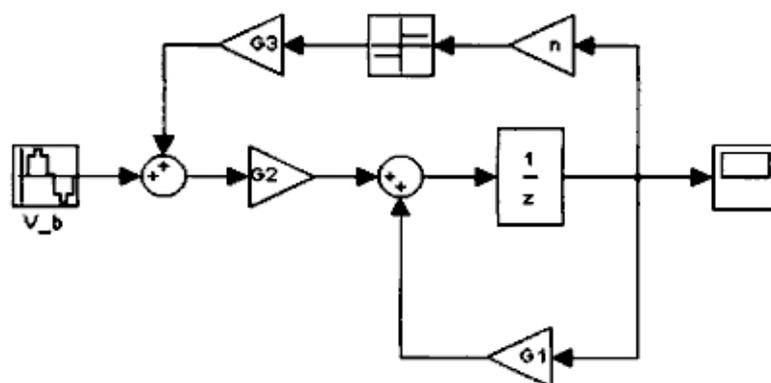


Figura A.3: Modelo discreto del servomotor DC.

Los parámetros para el sistema sin compensar se describen en la tabla A.3, donde G_i , $i = 1, 2, 3, 4$.

Las simulaciones han sido desarrolladas en SIMULINK, cuya entrada V_b es una entrada senoidal de amplitud $0,5V$ y frecuencia de $3rad/seg$. El resultado de la simulación para la velocidad angular w es mostrada en la figura A.4

La gráfica de la velocidad angular en estado estacionario del servomotor DC versus el voltaje aplicado al amplificador de potencia, se muestra en la figura A.5, nos permite apreciar una zona muerta a la velocidad cero. Esta zona muerta es debida a la fricción estática del servomotor. Note que el efecto no lineal en las cercanías de la zona muerta.

Cuadro A.3: Parámetros del sistema a simular.

Parámetro	Descripción	Valor
G_1	$1 - (T_s/J_{eq})(B_{eq} + nK_mK_b/R_a)$	0.9799
G_2	$K_mK_A T_s/(R_a J_{eq})$	0.1430
G_3	$-CR_a/(K_mK_A)$	-0.2174
G_4	n	12.5
T_s		0.001

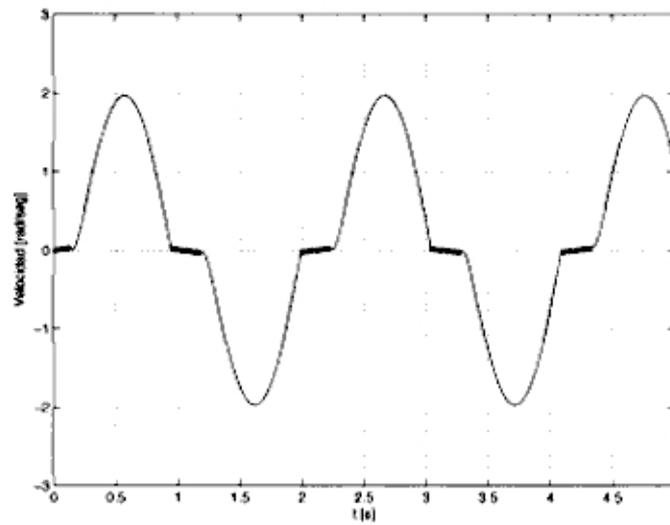


Figura A.4: Simulación del modelo simplificado $L = 0$.

Para linealizar la ecuación (A.19) se debe de adicionar una tensión de entrada a la entrada del amplificador tal que compense la fricción de Coulomb cuando la velocidad del servomotor $w_M \geq w_{th}$ (w_{th} velocidad umbral). Un esquema general es el que presenta la compensación de las fricciones estática y de Coulomb.

$$u = v + V_c \quad (\text{A.22})$$

siendo u la ley de control, v el voltaje de compensación de la fricción estática y V_c el voltaje de compensación de Coulomb. El objetivo del primer término (dependiente de la dirección del voltaje de armadura) es sacar al servomotor de la zona donde la fricción estática tiene efecto. El objetivo del segundo término es compensar el torque de fricción de Coulomb presente cuando el servomotor está en movimiento a velocidades mayores al umbral w_{th} .

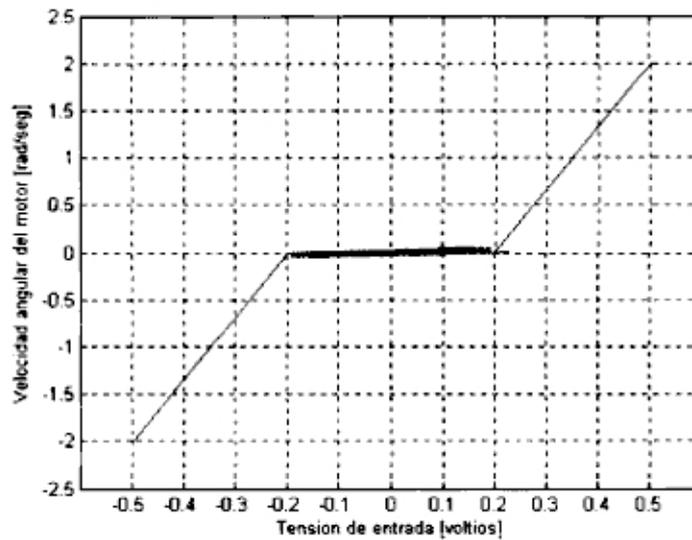


Figura A.5: Tensión de salida versus tensión de entrada.

El voltaje de compensación de Coulomb es dado por la relación

$$V_c = \frac{RC}{K_m K_A} \text{sgn}(V_b) \quad \text{si } w_M \geq w_{th} \quad (\text{A.23})$$

El coeficiente de fricción viscosa C es pequeño para servomotores DC, siendo difícil su identificación en presencia de ruido. Usamos el método de prueba y error para el cálculo de V_c , según la figura A.5, el valor inicial de V_{zm} es 0.2, este valor fue disminuido hasta encontrar el valor apropiado de $V_{zm} = 0,0217$.

Despejamos el coeficiente de fricción C de la ecuación (A.23) para reemplazar los parámetros previamente identificados, como son: $R_a = 5,3\Omega$, $K_A = 5,5$ y $K_m = 0,0421Nm/A$, luego.

$$C = \frac{V_{zm} K_m K_A}{R_a} = \frac{0,0217(0,0421)(5,5)}{5,3} = 9,5e^{-4} Kgm^2/seg \quad (\text{A.24})$$

El valor $V_{zm} = 0,0217$ permite compensar la fricción de Coulomb resultando el coeficiente de fricción de valor $C = 9,5e^{-4}$. Los efectos de compensación son mostrados en la figura A.6.

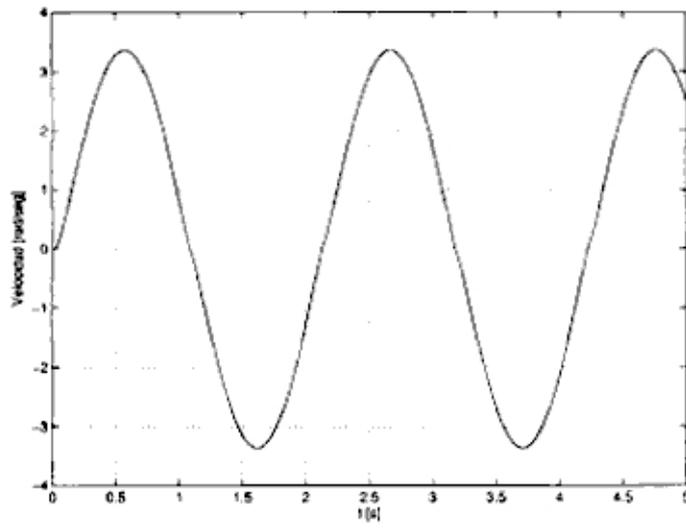


Figura A.6: Experimento de la compensación de la fricción no lineal.

Apéndice B

Diagramas en SIMULINK

Bloque principal

Los resultados de las simulaciones del capítulo 3 son también realizados mediante el programa de SIMULINK [2]. En la figura B.1 se presenta el bloque principal del sistema; como referencias consideramos las funciones escalón para la posición del móvil y del brazo, cuya magnitud es de 2 para cada uno cuyas unidades están expresadas en metros y radianes respectivamente; los parámetros de sintonización son los mismos considerados en el programa de simulación de matlab (Archivo *m*). Los valores de sintonización no son valores fijos, estos dependen principalmente de la trayectoria a controlar.

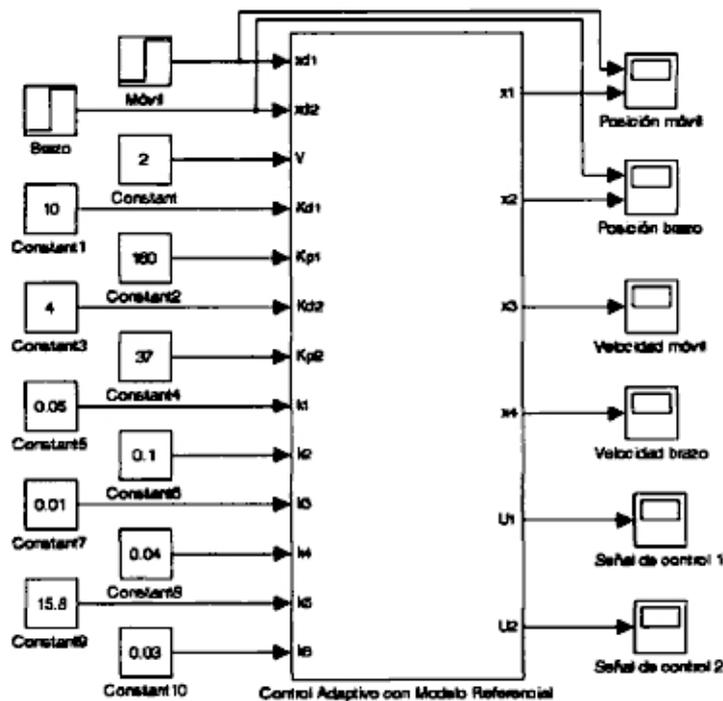


Figura B.1: Bloque principal control adaptivo del MRTM de 2DOF.

Bloque matrices dinámicas del sistema

En la figura B.2 se presentan las matrices de inercia, torques centrípetos coriolis y torques gravitacionales; mediante las ecuaciones m , p y q respectivamente. Los valores de la agrupación de parámetros (a_1 , a_2 , a_3 , a_4 , a_5 y a_6) están presentadas en el apéndice B.

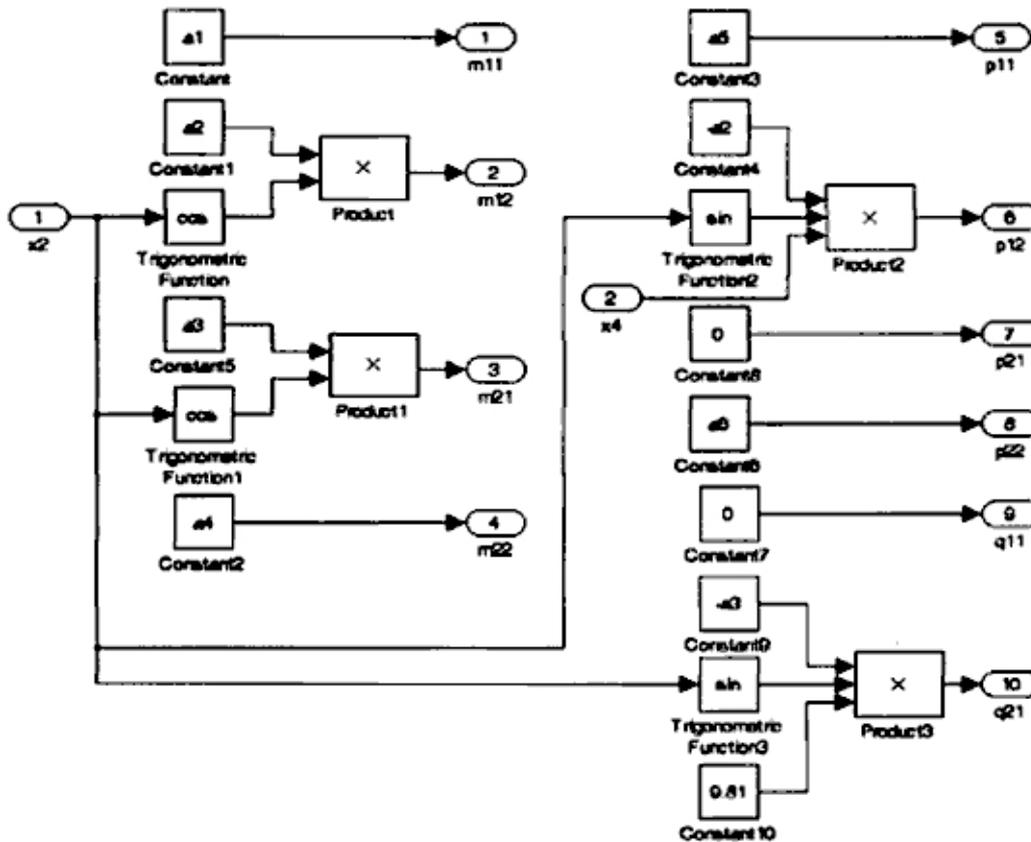


Figura B.2: Bloques de las matrices dinámicas del sistema.

Bloque de errores de seguimiento

En la figura B.3 se presentan los errores de Posición ex_1 y ex_2 , errores de velocidad dex_1 y dex_2 , calculo de ' s ' para medir la performance del error de seguimiento, ancho de banda del control ' V ' para eliminar los componentes de alta frecuencia indeseados, la performance de la velocidad (dqr_1 y dqr_2) y la aceleración ($ddqr_1$ y $ddqr_2$) de referencia.

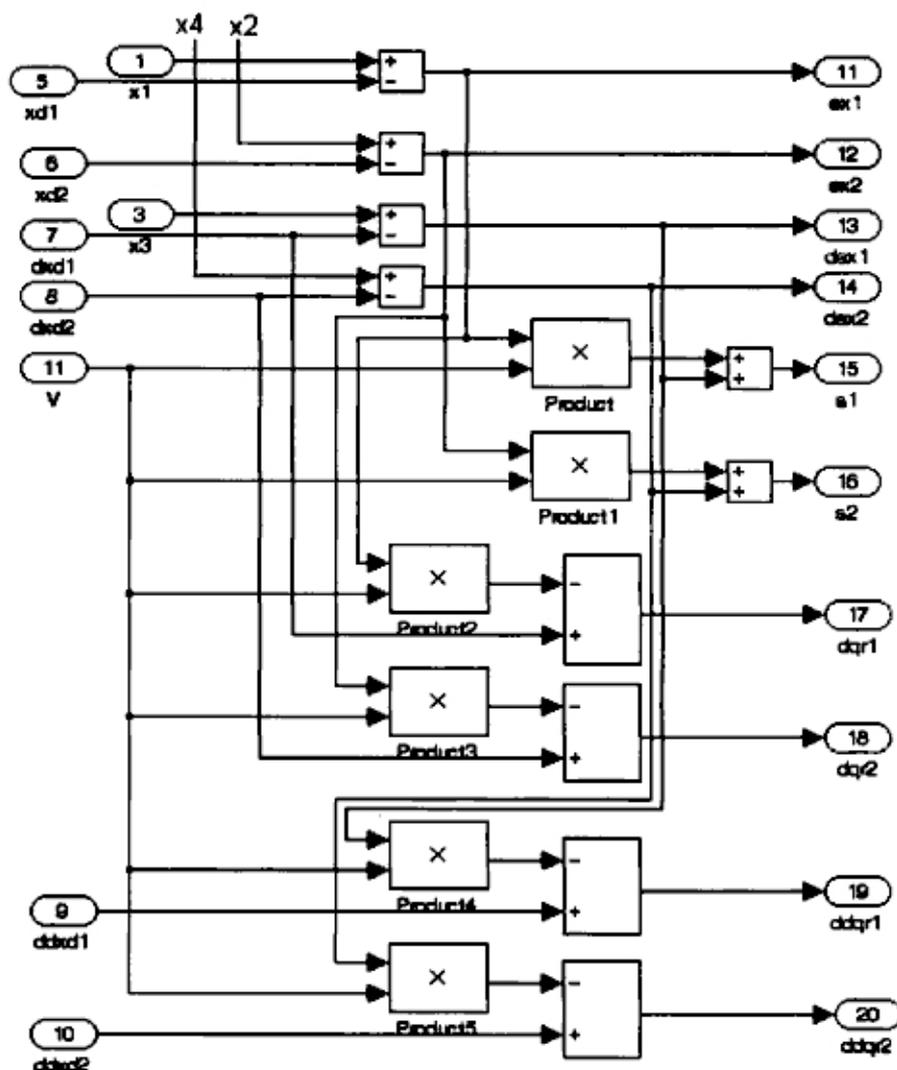


Figura B.3: Bloques de errores y referencias.

Bloque construcción de la matriz Y

En la figura B.4 se presentan las ecuaciones de la matriz Y , de términos conocidos, es posible la representación de esta matriz ya que el sistema es linealmente parametrizable, se representa las ecuaciones Y_{11} , Y_{12} , Y_{15} , Y_{23} , Y_{24} y Y_{26} , los términos que no aparecen tiene valor cero.

Bloque ley de control Y

En la figura B.5 se presentan las ecuaciones para obtener la ley de control u_1 y u_2 para el móvil y el brazo respectivamente. El controlador usado está princi-

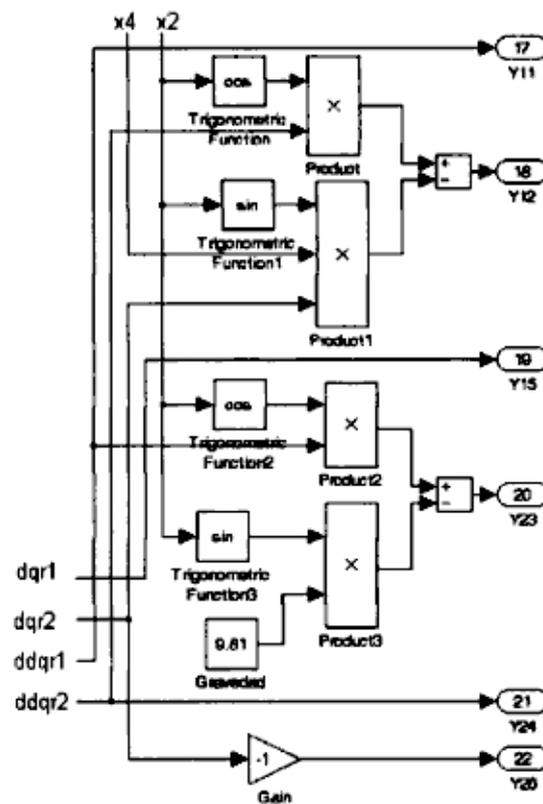


Figura B.4: Bloque de la matriz 'Y'.

palmente compuesto de un término dinámico (lazo en adelante - "feed forward") y un término **PD** (proporcional + derivativo). El controlador intenta cancelar la dinámica del sistema y el término **PD** le agrega estabilidad.

Bloque variables de estado x_1 , x_2 , x_3 y x_4

En la figura B.6 se presentan las ecuaciones discretizadas para obtener el modelo matemático de nuestro sistema, en ella obtenemos las variables x_1 , x_2 , x_3 y x_4 ; que corresponden a las posiciones (móvil - brazo) y las velocidades (móvil - brazo) respectivamente.

Bloque estimación de parámetros ae_1 , ae_2 , ae_3 , ae_4 , ae_5 y ae_6

En la figura B.7 se presentan las ecuaciones para obtener la estimación de parámetros (valores desconocidos) de nuestro sistema.

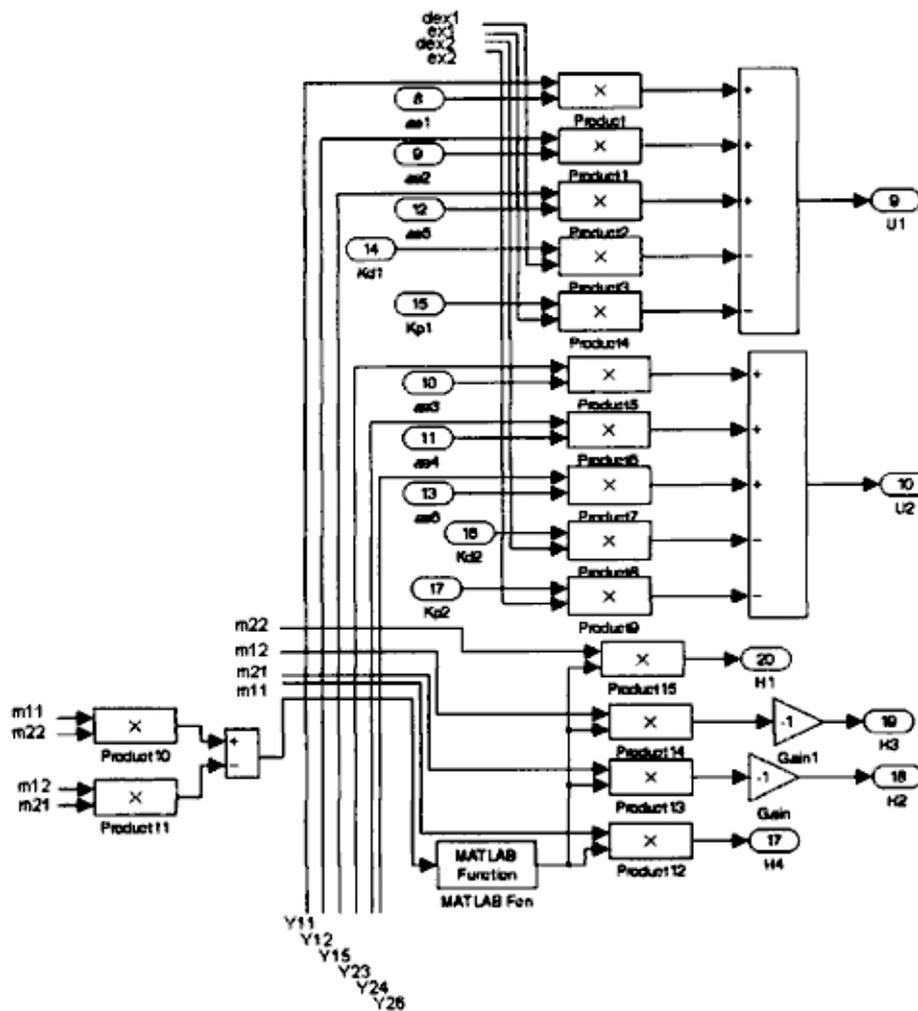


Figura B.5: Bloque de la Ley de Control U1 y U2.

Gráficas de la posición del móvil y brazo

En la figura B.8 se presentan las gráficas de las posiciones del móvil y el brazo. Consideramos un valor de referencia de 2 metros para el móvil y 2 radianes para el brazo.

Gráficas de las señales de control u1 y u2 para el móvil y brazo

En la figura B.9 se presentan las gráficas de control del móvil y el brazo. Las magnitudes de estas señales de control están expresadas en voltios y están dentro del rango permitido del amplificador.

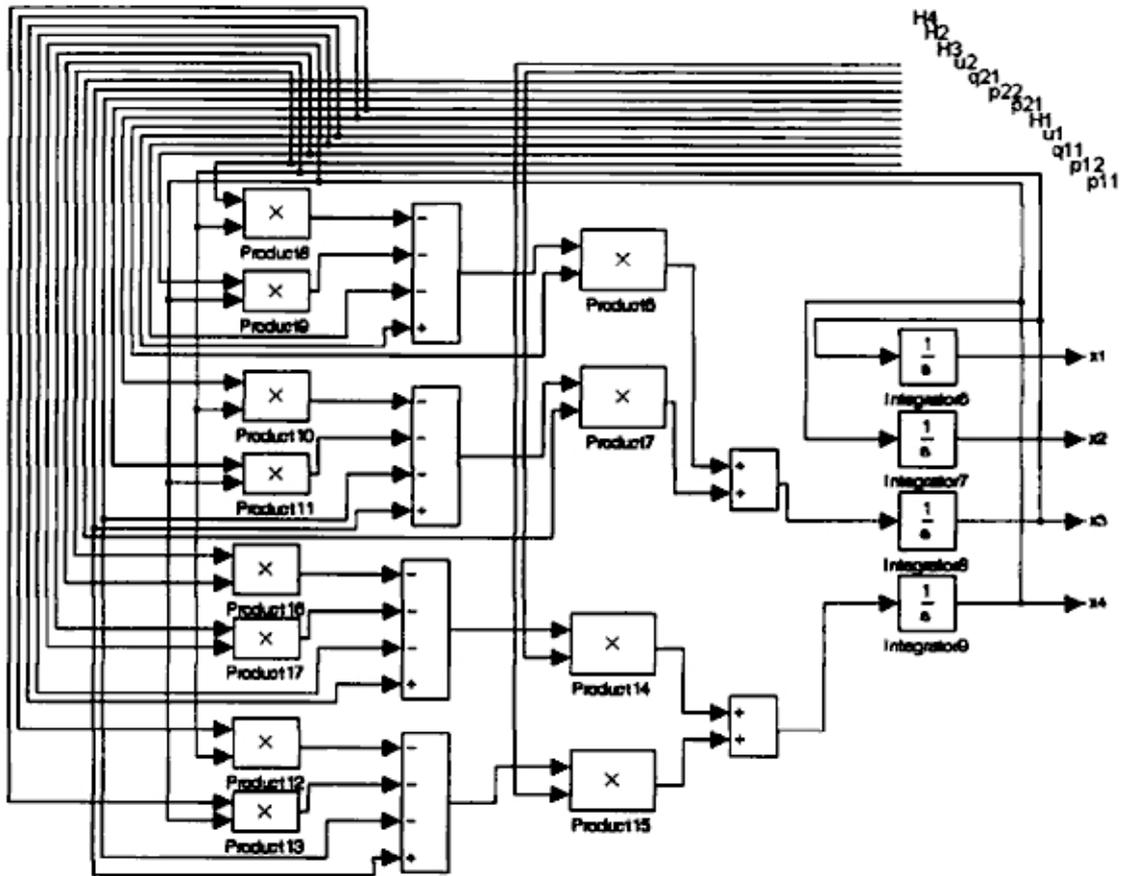


Figura B.6: Bloque variables de estado.

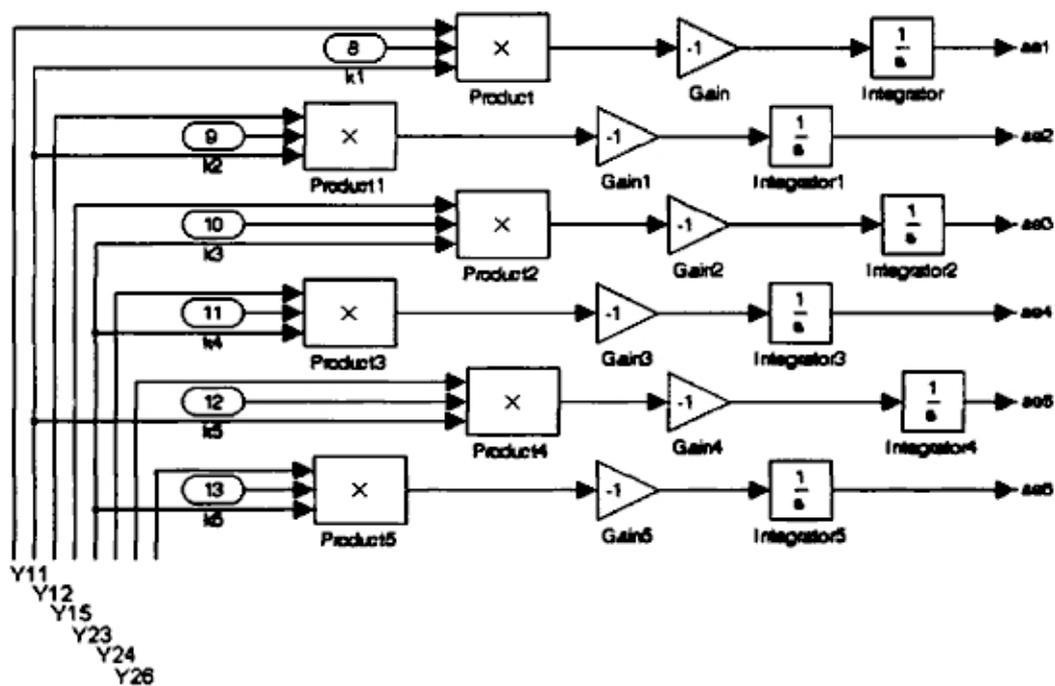


Figura B.7: Bloque de estimación de parámetros.

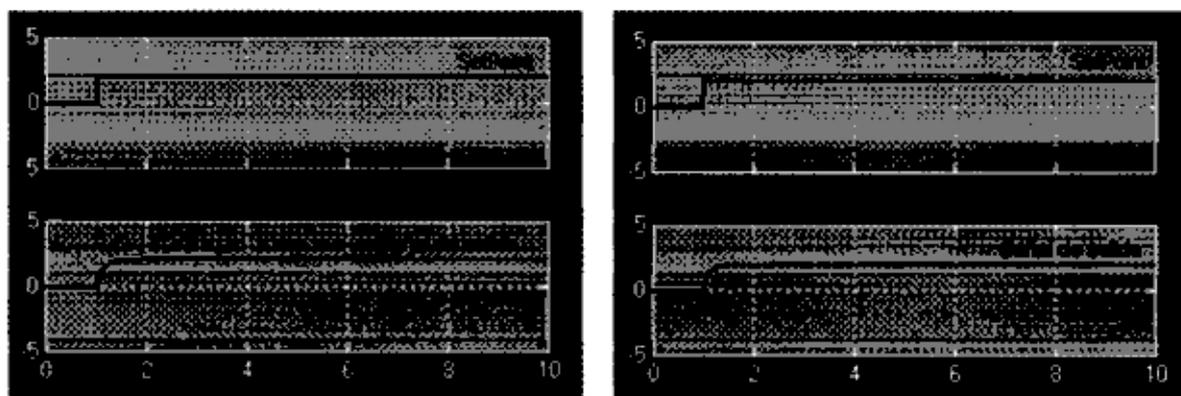


Figura B.8: Gráficas de las posiciones del móvil y brazo.

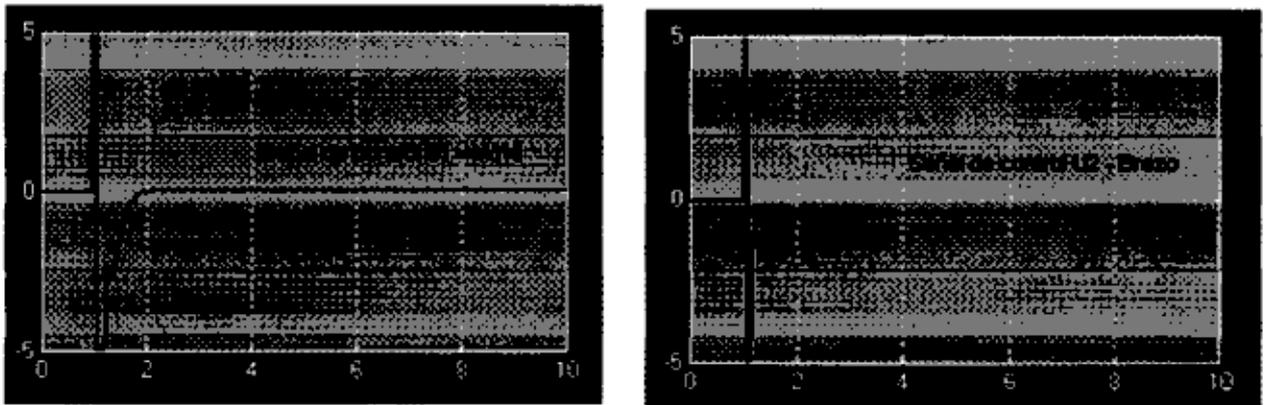


Figura B.9: Gráficas de las señales de control u_1 y u_2 para el móvil y brazo.

Apéndice C

Programa de Simulación

A continuación se muestra el programa principal de simulación adaptivo[2] [6].m escrito en Matlab 6.0. Este programa se encuentra en el CD adjunto con el nombre de adaptivo.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                %%
%%      adaptivo.m Manipulador robotico Multivariable 2DOF          %%
%%      Control Adaptivo no lineal con modelo referencial          %%
%%                                %%
%%                                %%
%%                                %%
%%                                %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Este programa es el control un manipulador robotico multiva-
% riable.Posee 02 motores; el primero en la polea y el segundo
% sobre el movil. La tecnica de control es ADAPTIVO NO LINEAL
% CON MODELO REFERENCIAL.
% Fecha: 07 de Abril 2003

clear all
close all
clc

% Parametros del sistema.
% -----

Mc=(0.802+0.7895+0.0577); % Masa de movil + masa motor(Polea) + seguro [Kg]
m1=(0.0519);           % Masa brazo. [Kg]
l1=0.15;              % Longitud desde el movil al centro de gravedad del
                    % brazo inferior. [m]
J1=2.386e-3;         % Masa del momento de inercia del brazo. [Kg.m^2]
L=0.30;             % Longitud total del brazo. [m]
g=9.81;            % Aceleracion de la gravedad. [m/s^2]
c1= 1.92e-3;       % Friccion viscosa rotacional de la juntura del Brazo
                    % [Kg.m^2/s]
f = 2.81;         % Coeficiente de friccion del movil [Kg/s]

% Parametros del motor (Motor de la polea / Motor sobre el movil)
% -----
Ra=5.3;           % Resistencia del motor [ohm].
n=12.5;         % Factor de reduccion de velocidad.
ka=14.9;        % Ganancia del amplificador.
kb=0.0046;      % Constante de fuerza contra-electromotriz.
```

```

km=0.0046;      % Constante del torque del motor. [N.m/A]
rp=0.048;      % Radio de la poea. [m]
Jeq=1.1822e-5; % Momento de inercia equivalente.  $J_m+n^2(J_o+J_p)$ 
Beq=2.75514e-5; % Friccion viscosa equivalente.  $B_m+n^2B_o$ 
La=0.0;        % Inductancia del motor (se desprecio).

% Valores del estado inicial del sistema
% -----

x1(1)=0; x2(1)=0; x3(1)=0; x4(1)=0; x5(1)=0; x6(1)=0;

% Valores iniciales de la estimacion de parametros.
% -----

ae1(1)=0; ae2(1)=0; ae3(1) =0; ae4(1) =0; ae5(1)=0; ae6(1)=0;

% Agrupacion de los parametros del sistema.
% -----

alfa = (Ra*rp)/(n*km*ka)
beta = Ra/(n*ka*km)

a1 = alfa*(Mc + m1 + (Jeq/rp^2))
a2 = alfa*m1*L/2
a3 = beta*m1*L/2
a4 = beta*(m1*L^2/4+J1+Jeq)
a5 = alfa*(f+ Beq/rp^2 + n^2*km*kb/(Ra*rp^2))
a6 = beta*(c1+Beq+(km*kb*n^2/Ra))

% Parametros de sintonizacion del control
% -----

I = 1;
V = 2*I;
Kd1 = 10.0*I;
Kp1 = 160.0*I;

Kd2 = 3.0*I;
Kp2 = 37.0*I;

% Parametros de simulacion.
% -----

T=0.01;      % Tiempo de muestreo. [s]
Tf=1500;     % Numero de iteraciones
             % Tiempo de simulacion. (T*M)

```

```

% Lazo de control (Simulacion)
% -----

for k=1:Tf

    aa1(k)=a1; aa2(k)=a2; aa3(k)=a3; aa4(k)=a4; aa5(k)=a5; aa6(k)=a6;

% Trayectoria deseada (Set-Point)
% -----

% Posicion
% -----
    xd1(k) = 0.2*cos(2*pi*k*T);    % Posicion del movil.
    %xd2(k) = pi/4;
    xd2(k) = (pi/4)*cos(2*pi*k*T);    % Posicion angular brazo .

% Velocidad
% -----
    dxd1(k) = -(2*0.2*pi)*sin(2*pi*k*T);    % Velocidad del movil.
    %dxd2(k) = 0;
    dxd2(k) = -(pi^2/2)*sin(2*pi*k*T);    % Velocidad angular brazo.

% Aceleracion
% -----
    ddx1(k) = -(2*2*0.2*pi*pi)*cos(2*pi*k*T);    % Aceleracion del movil.
    %ddxd2(k) = 0;
    ddx2(k) = -pi^3*cos(2*pi*k*T);    % Aceleracion angular brazo.

% Error tracking
% -----

% Error de posicion
% -----
    ex1(k) = x1(k) - xd1(k);
    ex2(k) = x2(k) - xd2(k);

% Error de velocidad
% -----
    dex1(k) = x3(k) - dxd1(k);
    dex2(k) = x4(k) - dxd2(k);

% Calculo de 's' para medir la performance del tracking
% Ancho de banda del sistema de contro ('V')
% -----
    s1(k) = dex1(k) + V*ex1(k);
    s2(k) = dex2(k) + V*ex2(k);

% Velocidad y aceleracion de referencia
% -----

% Velocidad

```

```

% -----
dqr1(k) = dxd1(k) - V*ex1(k);
dqr2(k) = dxd2(k) - V*ex2(k);

% Aceleracion
% -----
ddqr1(k) = ddx1(k) - V*dex1(k);
ddqr2(k) = ddx2(k) - V*dex2(k);

% Matrices dinamicas del sistema
% -----

% Matriz de Inercia
%-----
m11(k) = a1;
m12(k) = a2*cos(x2(k));
m21(k) = a3*cos(x2(k));
m22(k) = a4;

% Matriz de torques centripetos y coriolis
% -----
p11(k) = a5;
p12(k) = -a2*sin(x2(k))*x4(k);
p21(k) = 0;
p22(k) = a6;

% Matriz de los torques gravitacionales
% -----
q11(k) = 0;
q21(k) = -a3*g*sin(x2(k));

% Construccion de la matriz 'Y'; terminos conocidos
% -----
Y11a(k) = ddqr1(k);
Y12a(k) = cos(x2(k))*ddqr2(k) - sin(x2(k))*x4(k)*dqr2(k);
Y13a(k) = 0;
Y14a(k) = 0;
Y15a(k) = dqr1(k);
Y16a(k) = 0;

Y21b(k) = 0;
Y22b(k) = 0;
Y23b(k) = cos(x2(k))*ddqr1(k) - g*sin(x2(k));
Y24b(k) = ddqr2(k);
Y25b(k) = 0;
Y26b(k) = -dqr2(k);

% Ley de control
% (Feed forward dynamical + termino PD)
% -----
u1(k) = Y11a(k)*ae1(k)+Y12a(k)*ae2(k)+Y15a(k)*ae5(k)-Kd1*dex1(k)-Kp1*ex1(k);

```

```
u2(k) = Y23b(k)*ae3(k)+Y24b(k)*ae4(k)+Y26b(k)*ae6(k)-Kd2*dex2(k)-Kp2*ex2(k);
```

```
% Ecuaciones del modelo del sistema
```

```
% -----
M = [m11(k) m12(k); m21(k) m22(k)];
H = inv(M);
P = [p11(k) p12(k); p21(k) p22(k)];
```

```
% Discretizacion
```

```
% -----
x1(k+1) = x1(k) + T*x3(k);
x2(k+1) = x2(k) + T*x4(k);
x3(k+1) = x3(k) + T*(H(1)*(-p11(k)*x3(k)-p12(k)*x4(k)-q11(k)+u1(k))+...
                    H(3)*(-p21(k)*x3(k)-p22(k)*x4(k)-q21(k)+u2(k)));
x4(k+1) = x4(k) + T*(H(2)*(-p11(k)*x3(k)-p12(k)*x4(k)-q11(k)+u1(k))+...
                    H(4)*(-p21(k)*x3(k)-p22(k)*x4(k)-q21(k)+u2(k)));
```

```
% Estimacion de parametros (Valores desconocidos)
```

```
% -----
k1=0.05;
ae1(k+1) = ae1(k) - T*(k1*Y11a(k)*s1(k) + k1*Y21b(k)*s2(k));
k2=0.10;
ae2(k+1) = ae2(k) - T*(k2*Y12a(k)*s1(k) + k2*Y22b(k)*s2(k));
k3=0.001;
ae3(k+1) = ae3(k) - T*(k3*Y13a(k)*s1(k) + k3*Y23b(k)*s2(k));
k4=0.040;
ae4(k+1) = ae4(k) - T*(k4*Y14a(k)*s1(k) + k4*Y24b(k)*s2(k));
k5=15.8;
ae5(k+1) = ae5(k) - T*(k5*Y15a(k)*s1(k) + k5*Y25b(k)*s2(k));
k6=0.03;
ae6(k+1) = ae6(k) - T*(k6*Y16a(k)*s1(k) + k6*Y26b(k)*s2(k));
```

```
end
```

```
ejex=linspace(0,Tf*T,Tf);
```

```
figure(1)
subplot(211)
plot(ejex,ex1(1:Tf))
uaxis=axis;
ux1=uaxis(1,1);
ux2=uaxis(1,2);
uy1=uaxis(1,3);
uy2=uaxis(1,4);
utx=ux1+0.4*(ux2-ux1);
uty=uy1+0.2*(uy2-uy1);
text(utx,uty,['Error posicion del movil'])
ylabel(' [m] ')
grid
subplot(212)
plot(ejex,ex2(1:Tf))
uaxis=axis;
ux1=uaxis(1,1);
```

```

ux2=uaxis(1,2);
uy1=uaxis(1,3);
uy2=uaxis(1,4);
utx=ux1+0.4*(ux2-ux1);
uty=uy1+0.2*(uy2-uy1);
text(utx,uty,['Error posicion angular del brazo'])
ylabel('[rad]')
xlabel('[s]')
grid
print -f -deps simat1

```

```

figure(2)
subplot(211)
plot(ejex,dex1(1:Tf))
uaxis=axis;
ux1=uaxis(1,1);
ux2=uaxis(1,2);
uy1=uaxis(1,3);
uy2=uaxis(1,4);
utx=ux1+0.4*(ux2-ux1);
uty=uy1+0.2*(uy2-uy1);
text(utx,uty,['Error velocidad del movil'])
ylabel('[m/s]')
grid
subplot(212)
plot(ejex,dex2(1:Tf))
uaxis=axis;
ux1=uaxis(1,1);
ux2=uaxis(1,2);
uy1=uaxis(1,3);
uy2=uaxis(1,4);
utx=ux1+0.4*(ux2-ux1);
uty=uy1+0.2*(uy2-uy1);
text(utx,uty,['Error velocidad angular del brazo '])
ylabel('[rad/s]')
xlabel('[s]')
grid
print -f -deps simat2

```

```

figure(3)
subplot(211)
plot(ejex,xd1,ejex,x1(1:Tf)), grid on
title('Salidas')
uaxis=axis;
ux1=uaxis(1,1);
ux2=uaxis(1,2);
uy1=uaxis(1,3);
uy2=uaxis(1,4);
utx=ux1+0.4*(ux2-ux1);
uty=uy1+0.2*(uy2-uy1);
text(utx,uty,['Posicion del movil'])
ylabel('[m]')
subplot(212)
plot(ejex,xd2,ejex,x2(1:Tf)), grid on

```

```

uaxis=axis;
ux1=uaxis(1,1);
ux2=uaxis(1,2);
uy1=uaxis(1,3);
uy2=uaxis(1,4);
utx=ux1+0.4*(ux2-ux1);
uty=uy1+0.2*(uy2-uy1);
text(utx,uty,['Posicion angular del brazo'])
ylabel(' [rad] ')
xlabel(' [s] ')
print -f -deps simat3

figure(4)
subplot(211)
plot(ejex,u1(1:Tf)),grid on
title('Entradas')
uaxis=axis;
ux1=uaxis(1,1);
ux2=uaxis(1,2);
uy1=uaxis(1,3);
uy2=uaxis(1,4);
utx=ux1+0.3*(ux2-ux1);
uty=uy1+0.2*(uy2-uy1);
text(utx,uty,['Voltaje de entrada al motor 1'])
ylabel('u1[volt]')
subplot(212)
plot(ejex,u2(1:Tf)),grid on
uaxis=axis;
ux1=uaxis(1,1);
ux2=uaxis(1,2);
uy1=uaxis(1,3);
uy2=uaxis(1,4);
utx=ux1+0.4*(ux2-ux1);
uty=uy1+0.2*(uy2-uy1);
text(utx,uty,['Voltaje de entrada al motor 2'])
ylabel('u2[volt]')
xlabel(' [s] ')
print -f -deps simat4

figure(5)
subplot(321)
title('Identificacion de parametros')
plot(ejex,aa1,ejex,ae1(1:Tf)), grid on
title('ae1')
subplot(322)
plot(ejex,aa2,ejex,ae2(1:Tf)), grid on
title('ae2')
subplot(323)
plot(ejex,aa3,ejex,ae3(1:Tf)), grid on
title('ae3')
subplot(324)
plot(ejex,aa4,ejex,ae4(1:Tf)), grid on
title('ae4')
subplot(325)

```

```
plot(ejex,aa5,ejex,ae5(1:Tf)), grid on
title('ae5')
subplot(326)
plot(ejex,aa6,ejex,ae6(1:Tf)), grid on
title('ae6')
print -f -deps simat5
```

Apéndice D

Programas en C

Los programas siguientes han sido escritos en C++ de Borland, versión 3.2 y han sido desarrollados en una PC Pentium de 100 MHz, que tiene instalada una tarjeta de adquisición de datos LABPC+ que maneja los datos de entradas digitales y salidas analógicas para realizar el control en tiempo real.

Los programas en tiempo real para realizar el control del MRTM de 2DOF son los siguientes: el programa principal *Adaptivo.cpp*, utilizado para el control de posición y control de trayectoria, el programa de configuración de hardware de la tarjeta LABPC+ *func_pen.h*, y un archivo de librería *Nidaq.lib*. Los datos experimentales se capturan con los programas: *adqs.cpp* (programa principal), *adq_d1.cpp* (para configurar la tarjeta LABPC+) y *adq_d1.h* (que contiene el manejo de las funciones de la tarjeta LABPC+ y el archivo librería *Nidaq.lib*.)

⇒ *Adaptivo.cpp*

```
#include "func_pen.h"
#include <stdio.h>
#include <math.h>
#include <assert.h>
#include <alloc.h>
#include <conio.h>
#define IZQ 0 /* Sensor izquierdo del motor DC 1 (Movil)*/
#define DER 1 /* Sensor derecho del motor DC 2 (Varilla)*/
#define MOTOR1 IZQ
#define MOTOR2 DER
#define VRAD1 2.*M_PI/100.
#define VRAD2 2.*M_PI/100.
#define NN1 1/12.5
#define NN2 1/12.5
#define Rp 0.0485
#define Offset1 2.17
#define Offset2 2.17
#define Fs 100.
#define Ts (1./Fs)
#define tsimul 65.
#define DeadZone1 0.1.
#define DeadZone2 0.1.
#define g 9.8

/* Declaración de las funciones propias del programa*/
```

```

void Mide_Salidas(float *y1, float *vel1, float *y2, float *vel2);
/* Compensa la fricción estática y de coulomb*/
const float Wth=.1;
const float VcFE=.2,VcFC=.15;
float CompensaFriccion1(float u1, float vel1)
{
    float volt=0.;
    if(fabs(vel1)<=Wth)
    {
        /*compensa fricción estática*/
        if(u1<0.) volt=-VcFE;
        if(u1>0.) volt= VcFE;
    }
    else
    {
        /*compensa fricción de coulomb*/
        if (u1<0.) volt=-VcFC;
        if (u1>0.) volt= VcFC;
    }
    return (u1+volt);
}
float CompensaFriccion2(float u2, float vel2)
{
    float volt=0.;
    if(fabs(vel2)<=Wth)
    {
        /*compensa fricción estática*/
        if(u2<0.) volt=-VcFE;
        if(u2>0.) volt= VcFE;
    }
    else
    {
        /*compensa fricción de coulomb*/
        if (u2<0.) volt=-VcFC;
        if (u2>0.) volt= VcFC;
    }
    return (u2+volt);
}
void Mide_Salidas(float *y1, float *vel1, float *y2, float *vel2)
{
    static float y1ant=0;
    static float y2ant=0;
    static unsigned conM1_ant=0;
    static unsigned conM2_ant=0;
    unsigned conM1_act, conM2_act;
    /* Lee posicion del motor 1*/
    conM1_act=LeerPosicion(MOTOR1);
    if(conM1_act<conM1_ant)
    {
        if((conM1_ant-conM1_act)>32768)
            *y1=y1ant+(65536-conM1_ant+conM1_act)*VRAD1*NN1*Rp;
        else
            *y1=y1ant-(conM1_ant-conM1_act)*VRAD1*NN1*Rp;
    }
}

```

```

    }
else
{
    if((conM1_act-conM1_ant)>32768)
        *y1=y1ant-(65536-conM1_act+conM1_ant)*VRAD1*NN1*Rp;
    else
        *y1=y1ant+(conM1_act-conM1_ant)*VRAD1*NN1*Rp;
}
*vel1=(5.*(*y1-y1ant)+*vel1)/(1.+5.*Ts);
conM1ant=conM1act;
y1ant=*y1;
/* Lee posicion del motor 2*/
conM2_act=-LeerPosicion(MOTOR2);
if(conM2_act<conM2_ant)
{
    if((conM2_ant-conM2_act)>32768)
        *y2=y2ant+(65536-conM2_ant+conM2_act)*2.*M_PI/1250.;
    else
        *y2=y2ant-(conM2_ant-conM2_act)*2.*M_PI/1250.;
}
else
{
    if((conM2_act-conM2_ant)>32768)
        *y2=y2ant-(65536-conM2_act+conM2_ant)*2.*M_PI/1250.;
    else
        *y2=y2ant+(conM2_act-conM2_ant)*2.*M_PI/1250.;
}
*vel2=(5.*(*y2-y2ant)+*vel2)/(1.+5.*Ts);
conM2_ant=conM2_act;
y2ant=*y2;
}
void main()
{
    int ant=0, act=0, k, n;
    float t=0., vel1=0., vel2=0.;
    FILE *out;
    float *yb1, *ub1, *velb1, y1, u1, u11, y1ant;
    float *yb2, *ub2, *velb2, y2, u2, u22, y2ant;
    float *aeb1, *aeb2, *aeb3, *aeb4, *aeb5, *aeb6;
    float ae1=0, ae2=0, ae3=0, ae4=0, ae5=0, ae6=0;
    float ace1, ace2, ace3, ace4, ace5, ace6;
    float xd1, dxd1, ddx1, ex1, dex1, s1, dqr1, ddqr1;
    float xd2, dxd2, ddx2, ex2, dex2, s2, dqr2, ddqr2;
    float Y11a, Y12a, Y13a, Y14a, Y15a, Y16a;
    float Y21b, Y22b, Y23b, Y24b, Y25b, Y26b;
    float k1, k2, k3, k4, k5, k6;
    float I, V, Kd1, Kd2, Kp1, Kp2;
    float A,B,TT1,TT2;
    char tecla;
    ConfigurarHardware(Fs);
    EnviarVoltaje1(Offset1);
    EnviarVoltaje2(Offset2);
    clrscr();
}

```

```

printf("1) Colocar el movil: posición inicial\n

4) Colocar la varilla en posición inicial presionar F1\n");
while(((tecla=getch())!=59)&&(tecla!=27));
if(tecla==27) goto FIN;
ResetContadores();
clrscr();
assert(tsimul<150);
n= (int)tsimul/Ts;
/* Parámetros de sintonía*/
I = 1.0;
V = 2.0*I;
Kd1= 100.0*I;
Kp1= 1600.0*I;
Kd2= 4.0*I;
Kp2= 37.0*I;
A = 0.3;
B = M_PI;
TT1 = 50;
TT2 = 100;
assert(NULL!=(ub1 = (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(yb1 = (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(velb1= (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(ub2 = (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(yb2 = (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(velb2= (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(aeb1 = (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(aeb2 = (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(aeb3 = (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(aeb4 = (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(aeb5 = (float *)calloc(n/5,sizeof(float))));
assert(NULL!=(aeb6 = (float *)calloc(n/5,sizeof(float))));
for(k=0; k<n;)
{
    ant = 0;
    act = NivelClock();
    if ((ant==0)&&(act==1))
    {
        MideSalidas(&y1, &vel1, &y2, &vel2);
/* Trayectoria deseada (Set-Point)*/
        xd1 = A*cos(2*M_PI*k*T/TT1);
        xd2 = B*cos(2*M_PI*k*T/TT2);
/* Trayectoria deseada (Set-Point) Velocidad */
        dxd1 = -A*(2*M_PI/TT1)*sin(2*M_PI*k*T/TT1);
        dxd2 = -B*(2*M_PI/TT2)*sin(2*M_PI*k*T/TT2);
/* Trayectoria deseada (Set-Point) aceleracion */
        ddx1 = -A*(4*M_PI*M_PI/(TT1*TT1))*cos(2*M_PI*k*T/TT1);
        ddx2 = -B*(4*M_PI*M_PI/(TT2*TT2))*cos(2*M_PI*k*T/TT2);
/* Posición deseada (Set-Point) */
        xd1 = 0.5;
        xd2 = 180;
        dxd1 = 0;

```

```

ddxd1 = 0;
dxd2 = 0;
ddxd2 = 0;
/* Error de posición */
ex1 = y1 - xd1/100;
ex2 = y2 - xd2*M_PI/180;
ex1 = y1 - xd1;
ex2 = y2 - xd2;
/* Error de velocidad */
dex1 = vel1 - dxd1;
dex2 = vel2 - dxd2;
/* Calculo de 'S' para medir la performance */
/* Ancho de banda del sistema de control */
s1 = dex1 + V*ex1;
s2 = dex2 + V*ex2;
/* Velocidad */
dqr1 = dxd1 - V*ex1;
dqr2 = dxd2 - V*ex2;
/* Aceleracion */
ddqr1 = ddxd1 - V*dex1;
ddqr2 = ddxd2 - V*dex2;
/* Construccion de la matriz 'Y'; terminos conocidos */
Y11a = ddqr1;
Y12a = cos(y2)*ddqr2 - sin(y2)*vel2*dqr2;
Y13a = 0;
Y14a = 0;
Y15a = dqr1;
Y16a = 0;
Y21b = 0;
Y22b = 0;
Y23b = cos(y2)*ddqr1 - g*sin(y2);
Y24b = ddqr2;
Y25b = 0;
Y26b = -dqr2;
/* Leyes de control (u1 y u2) */
/* Feed forward dynamical + término PD */
u1 = Y11a*ae1+Y12a*ae2+Y15a*ae5-Kd1*dex1-Kp1*ex1;
u11= CompensaFriccion1(u1, vel1);
/* Compensa zona muerta */
if(u11 < 0) u11=u11-DeadZone1;
if(u11 > 0) u11=u11+DeadZone1;
EnviarVoltaje1(u11+Offset1);
u2 = Y23b*ae3+Y24b*ae4+Y26b*ae6-Kd2*dex2-Kp2*ex2;
u22= CompensaFriccion2(u2, vel2);
/* Compensa zona muerta */
if(u22 < 0) u22=u22-DeadZone2;
if(u22 > 0) u22=u22+DeadZone2;
EnviarVoltaje2(u22+Offset2);
/* Parámetros de sintonización para trayectoria */
k1=0.05, k2=0.1, k3=0.01, k4=0.040, k5=15.8 y k6=0.03;
I=1.; V=2*I; Kd1=100.*I; Kp1=1600.*I; Kd2 = 1.*I;
Kp2=20.*I; A=0.2; B=M_PI/2; TT1=45; TT2=45;
/* Parámetros de sintonización para posición*/

```

```

k1=0.05, k2=0.1, k3=0.0001(+180),k3=0.001(-180);
k4=0.040, k5=15.8 y k6=0.03;
I=1.; V=2*I; Kd1=100.*I; Kp1=1600.*I; Kd2 = 1.*I;
Kp2=20.*I; A=0.2; B=M_PI/2; TT1=45; TT2=45;
/* Estimación de parámetros (Valores desconocidos)*/
k1= 0.05;
aee1 = ae1 - Ts*(k1*Y11a*s1 + k1*Y21b*s2);
k2= 0.1;
aee2 = ae2 - Ts*(k2*Y12a*s1 + k2*Y22b*s2);
k3= 0.01;
aee3 = ae3 - Ts*(k3*Y13a*s1 + k3*Y23b*s2);
k4= 0.040;
aee4 = ae4 - Ts*(k4*Y14a*s1 + k4*Y24b*s2);
k5= 15.8;
aee5 = ae5 - Ts*(k5*Y15a*s1 + k5*Y25b*s2);
k6= 0.03;
aee6 = ae6 - Ts*(k6*Y16a*s1 + k6*Y26b*s2);
ae1=aee1;
ae2=aee2;
ae3=aee3;
ae4=aee4;
ae5=aee5;
ae6=aee6;
if(!(k %5))
{
    yb1[k/5]=y1*100;
    ub1[k/5]=u11;
    velb1[k/5]=vel1;
    yb2[k/5]=y2*180/M_PI;
    ub2[k/5]=u22;
    velb2[k/5]=vel2;
    aeb1[k/5]=ae1;
    aeb2[k/5]=ae2;
    aeb3[k/5]=ae3;
    aeb4[k/5]=ae4;
    aeb5[k/5]=ae5;
    aeb6[k/5]=ae6;
} ;
k++;
t+=Ts ;
gotoxy(15,1); printf("*****");
gotoxy(18,3); printf("VARIABLES DE PROCESO");
gotoxy(15,5); printf("*****");
gotoxy(21,7); printf("tiempo :t= %+5.3f",t);
gotoxy(21,9); printf("señal control1(Comp.) :u1= %+5.3f",u11);
gotoxy(21,11);printf("pos.del movil :y1= %+5.3f",y1*100);
gotoxy(21,13);printf("vel.del movil :vel1= %+5.3f",vel1);
gotoxy(21,15);printf("señal control2(Comp.) :u2= %+5.3f",u22);
gotoxy(21,17);printf("pos.angular2(Grados):y2= %+5.3f",y2*180/M_PI);
gotoxy(21,19);printf("vel.angular2(Rad) :vel2= %+5.3f",vel2);
gotoxy(21,19);printf("vel.angular2(Rad) :vel2= %+5.3f",vel2);
gotoxy(15,21);printf("*****");
if(kbhit())

```



```

    if(v>3.7) v=3.7;
    salidaDigital = (unsigned) (409.6*v+0.5);
    output(DAC1,salidaDigital); /* ESCRIBE salidaDigital en DAC1 */
}
unsigned int LeerPosicion(int sensor)
{
    unsigned char byteH,byteL;
    if(sensor) outputb(PORTC,inportb(PORTC)&251);
    else outputb(PORTC,inportb(PORTC)|4);
    /* PC2=0 selecciona contador derecho, PC2=1 selecciona contador izquierdo */
    byteL = inportb(PORTA); /* Lee PORT A */
    byteH = inportb(PORTB); /* Lee PORT B */
    return (256*byteH+byteL);
}
int NivelClock()
{
    /* unsigned char bytePoll */;
    /* bytePoll = inportb (PORTC); /* lee PORTC
    /* return (bytePoll>127); /* valida solo PC7 (clk de la tarjeta-OUTB0)
    return(1);
}
void ResetContadores()
{
    outputb(PORTC,inportb(PORTC)-8);
    for(int i=0;i<10000;i++);
    outputb(PORTC,inportb(PORTC)&(255-8));
}

```

⇒ adqs.cpp

```

#include "adq_d1.h"
#include <stdio.h>
#include <math.h>
#include <assert.h>
#include <alloc.h>
#include <conio.h>
#include sensor 1
const float ESCALA= 2.*M_PI/(100.*12.5);
/* Declaración de las funciones propias del programa */
void Inicializa_Sistema();
void Mide_Variables(float *yk, float *vel, float t);
float Aplica_Control(float u, float ykp);
/* Declaración de las variables globales */
float N, y, yant, T, t, tsimul, Fs;
int p0, pk;
float VcFC, VcFE, Wth, Ofst;
/* Funciones*/
float Aplica_Control(float u, float ykp)
{
    float v;
    if(fabs(ykp)<=Wth)
    {

```

```

    if(u<0.) v = -VcFE;
    if(u>0.) v = VcFE;
}
else
{
    if(u<0.) v = -VcFC;
    if(u>0.) v = VcFC;
}
    if(u+v<-1.4) u= -1.4;
    else if(u+v>1.4) u= 1.4;
    EnviarVoltaje(u+v+Ofst);
    return u;
}
void Inicializa_Sistema()
{
    VcFE = .2; VcFC = .15;
    Wth = .1; Ofst = 2.17; //servosystem
    Fs = 100.; T = 1./Fs;
    tsimul = 20.;
    ConfigurarHardware(Fs);
    EnviarVoltaje(Ofst);
    p0 = LeerPosicion(sensor);
    /* p0 = LeerPosicion() */
}
void Mide_Variables(float *y, float *vel)
{
    /* Ahora lee posicion */
    pk = LeerPosicion(sensor);
    *y = yant + ((float)(abs(pk-p0)>8000?0:pk-p0))*ESCALA;
    p0 = pk;
    /* Y la velocidad angular */
    /* vel = 0.*/
    *vel = (5.*(*y-yant)+*vel)/(1.+5.*T);
    /* vel= (*y-yant)/(T);*/
    /* Actualizamos los estados*/
    yant=*y;
}
void main()
{
    int ant=0, act=0, k, n;
    FILE *out;
    float *yb, *ub, *velb, vel, y, u;
    float yant, w, f;
    Inicializa_Sistema();
    clrscr();
    assert(tsimul<120);
    N = tsimul/T;
    n = (int)N;
    t = vel= 0.;
    /* resetContadores();*/
    assert(NULL! =(ub= (float *)calloc(n/5,sizeof(float))));
    assert(NULL! =(yb= (float *)calloc(n/5,sizeof(float))));
    assert(NULL! =(velb= (float *)calloc(n/5,sizeof(float))));
}

```

```

for(k=0; k<n;
{
    act=NivelClock();
    if ((ant==0)&&(act==1))
    {
        Mide_Variables(&y, &vel);
/* Señal de Control */
        f = 0.8;
/* w = 2*M_PI*f;*/
/* u = 0.5*sin(w*t);*/
        u = 1;
        u = Aplica_Control(u, vel);
        if(!(k%5))
        {
            yb[k/5]=y*180./M_PI; ub[k/5]=u; velb[k/5]=vel;
        }
/* ResetContadores;*/
        k++;
        t+ = T;
        gotoxy(15,5); printf("*****");
        gotoxy(18,7); printf("Identificacion de Motor D.C.");
        gotoxy(15,9); printf("*****");
        gotoxy(21,11);printf("tiempo : t= %+5.3f",t);
        gotoxy(21,17);printf("señal de control : u= %+5.3f",u);
        gotoxy(21,19);printf("posición angular : y= %+5.3f",y*180./M_PI);
        gotoxy(21,21);printf("velocidad angular : vel= %+5.3f",vel);
        gotoxy(15,25);printf("*****");
        if(kbhit()) if(getch()==27) break;
/* ResetContadores;*/
    }
    ant=act;
}
EnviarVoltaje(Ofst);
out=fopen("adqs24.dat", "wt");
for (int i=0; i<n/5; i++)
    fprintf(out, "%10f%10f%10f%10f \n",i*5*T, yb[i], ub[i], velb[i]);
delete yb, ub, velb;
fclose(out);
}

```

⇒ adq_d1.cpp

```

#include "adq_d1.h"
#include "nidaq.h"
#include <dos.h>
#include DAC1 0x266
#include PORTA 0x270
#include PORTB 0x271
#include PORTC 0x272
#include void ConfigurarHardware (float frec)
{
    int periodoTimer, deviceCode;
    periodoTimer = (int)(2000000/frec);
}

```

```

USE_LAB();
Init_DA_Brds(1,&deviceCode);
ICTR_Setup(1,0,3,periodoTimer,1);
outportb(0x273, 0x9A);
}
void EnviarVoltaje (float u)
{
  unsigned salidaDigital;
  if(u<0.) u=0.8;
  if(u>3.7) u=3.7;
  salidaDigital = (unsigned) (409.6*u+0.5);
  outport(DAC1,salidaDigital);
}
unsigned int LeerPosicion(int sensor)
{
  unsigned char byteH,byteL;
  if(sensor) outportb(PORTC,inportb(PORTC)&251);
  else outportb(PORTC,inportb(PORTC)|4);
  byteL = inportb(PORTA);
  byteH = inportb(PORTB);
  return (256*byteH+byteL);
}
int NivelClock()
{
  unsigned char bytePoll;
  bytePoll = inportb (PORTC);
  return (bytePoll>127);
}
void ResetContadores()
{
  outportb(PORTC,inportb(PORTC)|8);
  for(int i=0;i<10000;i++);
  outportb(PORTC,inportb(PORTC)&(255-8));
}

```

⇒ adq_d1.h

```

#if !defined (_ADQ_D1_H)
#define _ADQ_D1_H
  void ConfigurarHardware (float);
  void EnviarVoltaje (float);
  unsigned int LeerPosicion(int);
  int NivelClock ();
#endif

```

Bibliografía

- [1] Astrom, J.K. and Witternmark B. *Adaptive Control*. Addison - Wesley, 1989.
- [2] Bishop, Robert H. *Modern Control System Analysis and Design Using Matlab*. Addison Wesley, 1997.
- [3] Hassan and K. Khalil. *Nonlinear Systems*. Prentice Hall, 1996.
- [4] Jackson, Jhon H. and Harold G Wirtz. *Statics and Strength of Materials*. Mc Grawhill, 1983.
- [5] La Salle, Joseph and Salomon Lefschetz. *Stability by Liapunov's Direct Method*. Academic Press - London, 1967.
- [6] Nakamura, Shoichiro. *Análisis Numérico y Visualización Gráfica*. Prentice Hall, 1997.
- [7] of Technology, Massachusetts Institute. "Adaptive Control for the two degree of freedom Fast Eye Gimbal," <URL://exodus.mit.edu/jesse/papers/abstracts/2152-abs.html> (1994).
- [8] Ogata, Katsuhiko. *Dinámica de sistemas*. Prentice Hall, 1987.
- [9] Part Enander, Eva and Anders Sjoberg. *The Matlab 5 Handbook*. Addison Wesley, 1999.
- [10] Rojas Moreno, Arturo. *Control Avanzado Diseño y Aplicaciones en Tiempo Real*. Publicación Independiente, 2001.
- [11] Rojas Moreno, Arturo. *Applied MIMO Nonlinear Control Real Time Implementations, en construcción*. Publicación Independiente, 2003.

[12] Slotine, E. and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.