



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de la Laguna

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“Control basado en aproximación adaptable de
un sistema tipo péndulo invertido montado
sobre una esfera”

POR:

Ing. Rubén Hernández Rodríguez

TESIS

PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL
GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA

DIRECTOR DE TESIS

Dr. Francisco Jurado Zamarripa

ISSN: 0188 – 9060



RIITEC: (08)-TMCIE-2019

Torreón, Coahuila de Zaragoza, México

Julio 2019



SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de La Laguna

"2019, Año del Caudillo del Sur, Emiliano Zapata"

Torreón, Coah., 18/Junio/2019
Dependencia: DEPI/CPCIE
Oficio: DEPI/CPCIE/092/2019
Asunto: Autorización de
impresión de tesis.

C. ING. RUBÉN HERNÁNDEZ RODRÍGUEZ
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA.
PRESENTE

Después de haber sometido a revisión su trabajo de tesis titulado:

"Control basado en aproximación adaptable de un sistema tipo péndulo invertido montado sobre una esfera"

Habiendo cumplido con todas las indicaciones que el jurado revisor de tesis hizo, se le comunica que se le concede la autorización con número de registro **RIITEC: (08)-TMCIE-2019**, para que proceda a la impresión del mismo.

ATENTAMENTE
EDUCACIÓN TECNOLÓGICA FUENTE DE INNOVACIÓN

DR. JOSÉ IRVING HERNÁNDEZ JACQUEZ
Jefe de la División de Estudios de Posgrado e Investigación
del Instituto Tecnológico de La Laguna



SECRETARÍA DE
EDUCACIÓN PÚBLICA
INSTITUTO TECNOLÓGICO
de la laguna
División de Estudios de Posgrado
e Investigación



Carretera a la Laguna, s/n. Av. Revolución y Av. Instituto Tecnológico de La Laguna, Col. Centro. C.P. 27000. Torreón, Coah.

Tel. 01 (871) 7051313 e-mail: webmaster@correo.itlalaguna.edu.mx

www.tecnm.mx | http://lalaguna.tecnm.mx





"2019, Año del Caudillo del Sur, Emiliano Zapata"

Torreón, Coah., 17/Junio/2019

DR. JOSÉ IRVING HERNÁNDEZ JACQUEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Por medio de la presente, hacemos de su conocimiento que después de haber sometido a revisión el trabajo de tesis titulado:

"Control basado en aproximación adaptable de un sistema tipo péndulo invertido montado sobre una esfera"

Desarrollado por el **C. Rubén Hernández Rodríguez**, con número de control **M1713011** y habiendo cumplido con todas las correcciones que se le indicaron, estamos de acuerdo que se le conceda la autorización de la fecha de examen de grado para que proceda a la impresión de la misma.

ATENTAMENTE

EDUCACIÓN TECNOLÓGICA FUENTE DE INNOVACIÓN

Dr. Francisco Jurado Zamarripa
Asesor/Director de Tesis

Dr. Ramón García Hernández
Comité Tutorial

Dr. Ricardo Emmanuel Campa Cocom
Comité Tutorial

Dr. Edmundo Javier Ollervides Vázquez
Comité Tutorial



El comité tutorial de este trabajo de tesis está formado por:

Asesor: Dr. Francisco Juarado Zamarripa

Secretario: Dr. Ramón García Hernández

Vocal: Dr. Ricardo Emmanuel Campa Cocom

Vocal suplente: M.C. Edmundo Javier Ollervides Vázquez

Dedicatoria

*A ti Dios por darme el regalo de la existencia. Toda la vida no me alcanzará para
agradecerte todo lo que haces por mí....*

*A mis padres Rubén y Manuela, con todo mi cariño y amor para ustedes que hicieron todo
en la vida para que yo pudiera lograr mis sueños, por animarme y darme su apoyo cuando
lo necesitaba. A Javier, María y Missael por todo su apoyo. A mis hijos Jesé, Cecilia, Saúl
y Romina por ser una gran bendición y regalo de Dios en mi vida y a ti Edna por tu apoyo.*

A todos y a cada uno de ustedes mi corazón y mi gratitud.

Agradecimientos

Agradezco al M.E. Jesús Quistián por su amistad y apoyo incondicional para permitirme vivir este sueño. A la Junta Directiva del Instituto Tecnológico Superior de San Pedro de las Colonias por la beca-comisión otorgada a través del acuerdo SO42.08.181016 para realizar mis estudios de maestría, los cuales culmino con la presentación de esta Tesis.

Al Dr. Francisco Jurado Zamarripa por su amistad, su confianza, su paciencia y consejos durante estos dos años. Al Dr. Leonardo Herrera por motivarme al estudio del filtro de Kalman y contagiarme de su entusiasmo por la investigación. Agradecimiento especial a mi comité tutorial: Dr. Ramón García, M.C. Javier Ollervides y al Dr. Ricardo Campa por su amistad y enseñanzas a lo largo de varios años. A cada uno de ellos, gracias por sus invaluable comentarios para la mejora de este trabajo de tesis.

Agradezco a todos mis profesores del posgrado, Dr. Alfonso Pámanes, Dr. José Luis Meza, Dr. Víctor Santibañez, Dr. Alejandro Dzul y Dr. Miguel Ángel Llama, gracias por haberme compartido sus experiencias y conocimientos.

A mis compañeros y amigos del posgrado, de los cuales me siento bendecido por haberlos conocido en esta etapa de mi vida; *M.C. Óscar González, M.C. Amaury Meza, M.C. Roberto Franco, M.C. Jorge Villalobos, M.C. Andrés Pizarro, M.C. Omar Martínez, M.C. Ricardo Ovalle, M.C. Isaac Gandarilla, M.C. Sergio López, M.C. Jorge Montoya, M.C. Óscar Murillo, M.C. José Cruz, M.C. Rubén Ontiveros, M.C. Romeo Falcón, M.C. Alfredo Delgado y M.C. Ricardo Falcón.* A todos y cada uno de ellos mi amistad, admiración y agradecimiento para toda mi vida.

A las personas cercanas y lejanas que a lo largo de casi dieciocho años han sido fuente de inspiración y motivación para ser mejor persona: *M.C. José Navarrete, Dr. Javier Moreno, Dr. Rafael Kelly, Ing. Ulises Rosas, M.C. Omar Elio Torres, Lic. Yucef Mendoza, Ing. Luis Alberto Ruiz, Méd. Sergio González, Ing. Héctor Manuel González, MSC. Pablo Ulises González, Ing. Eduardo Valdívía, Ing. Enrique Martínez, Ing. José Luis Mota, Ing. Adriana Pineda, Dra. Evangelina Herrán, Dr. Jaime Herrera, Dr. Fernando Herrera, Dr. Juan Cruz Reyes y a tantas más.*

A todos, muchas gracias.

Control basado en aproximación adaptable de un sistema tipo péndulo invertido montado sobre una esfera

por RUBÉN HERNÁNDEZ RODRÍGUEZ

Resumen

En este trabajo de tesis se aborda el problema de control para la estabilización de un sistema subactuado del tipo péndulo invertido montado sobre una esfera conocido como *ballbot*. Dos controladores por modos deslizantes desacoplados basados en aproximación difuso adaptable y aproximación neuronal adaptable son propuestos. Se construyó el robot prototipo con LEGO MINDSTORMS EV3 para validar los esquemas de control propuestos a través de pruebas experimentales.

Palabras clave: Control difuso, control neuronal, *ballbot*, control adaptable, control por modos deslizantes, sistemas subactuados.

Control based on adaptive approximation of an inverted pendulum system mounted on a ball

by RUBÉN HERNÁNDEZ RODRÍGUEZ

Abstract

This thesis addresses the control problem for the stabilization of an inverted pendulum system mounted on a ball known as ballbot robot. Two controllers by decoupled sliding modes based on adaptive fuzzy approximation and adaptive neural approximation are proposed. The prototype robot was built with LEGO MINDSTORMS EV3 to validate the proposed control schemes through experimental tests.

Keywords: Fuzzy control, neural control, ballbot, adaptive control, sliding mode control, underactuated systems.

Índice general

| | |
|--|-----------|
| 1. Introducción | 1 |
| 1.1. Antecedentes | 1 |
| 1.1.1. Sistemas subactuados | 1 |
| 1.1.2. Control y dinámica no lineal | 3 |
| 1.2. El <i>ballbot</i> | 7 |
| 1.2.1. Motivación | 7 |
| 1.2.2. Antecedentes | 7 |
| 1.3. Objetivo | 10 |
| 1.3.1. Objetivo general | 10 |
| 1.3.2. Objetivos específicos | 10 |
| 1.4. Contribución | 10 |
| 1.5. Organización de la tesis | 11 |
| 2. Modelo dinámico | 12 |
| 2.1. Consideraciones | 12 |
| 2.2. Descripción del modelo | 12 |
| 2.2.1. Ecuaciones de Euler-Lagrange | 15 |
| 2.2.2. Parámetros del modelo | 19 |
| 2.3. Representación en espacio de estado | 19 |
| 3. Preliminares | 21 |
| 3.1. Fundamentos de sistemas no lineales | 21 |
| 3.2. Propiedades fundamentales | 23 |
| 3.3. Conceptos de estabilidad | 23 |
| 3.4. Lema de Barbalat | 25 |
| 3.4.1. Espacios de funciones | 25 |
| 3.4.2. Sistemas variantes en el tiempo | 26 |
| 4. Control por modo deslizante | 28 |

| | | |
|-----------|--|-----------|
| 4.1. | Introducción | 28 |
| 4.2. | Superficies deslizantes | 29 |
| 4.2.1. | Construcción de Filippov de la dinámica equivalente | 34 |
| 4.2.2. | Perfecto desempeño a un alto precio | 36 |
| 4.2.3. | Aplicaciones de las leyes de control de conmutación discontinuas | 41 |
| 4.3. | Aproximaciones continuas | 42 |
| 4.4. | Comentarios | 45 |
| 5. | Controlador difuso adaptable | 47 |
| 5.1. | Lógica difusa | 47 |
| 5.1.1. | Conjunto difuso | 47 |
| 5.1.2. | Funciones de pertenencia | 48 |
| 5.2. | Control basado en lógica difusa | 48 |
| 5.2.1. | Características | 49 |
| 5.2.2. | Estructura | 50 |
| 5.2.3. | Aproximador universal | 52 |
| 5.3. | Control difuso adaptable | 53 |
| 5.4. | Control por modos deslizantes | 54 |
| 5.4.1. | Diseño del controlador por modos deslizantes | 55 |
| 5.5. | Diseño del sistema difuso | 57 |
| 5.6. | Diseño de la ley de adaptación | 58 |
| 5.6.1. | Función candidata de Lyapunov | 58 |
| 5.6.2. | Derivada de la función candidata de Lyapunov | 59 |
| 5.7. | Análisis de estabilidad | 60 |
| 5.8. | Resultados de simulación | 62 |
| 5.8.1. | Parámetros de diseño | 63 |
| 5.8.2. | Resultados | 65 |
| 5.9. | Comentarios | 66 |
| 6. | Controlador neuronal adaptable | 68 |
| 6.1. | Control Neuronal | 68 |
| 6.1.1. | ¿Por qué el control neuronal? | 68 |
| 6.1.2. | Revisión del control por redes neuronales | 70 |
| 6.2. | Redes neuronales | 71 |
| 6.2.1. | Funciones de base radial (RBF) | 74 |
| 6.2.2. | Redes neuronales <i>wavelet</i> (ondaleta) | 77 |
| 6.3. | Diseño del controlador | 80 |
| 6.3.1. | Descripción del sistema subactuado | 80 |
| 6.3.2. | Diseño del controlador | 80 |
| 6.4. | Resultados de simulación | 86 |
| 6.4.1. | Parámetros de diseño | 86 |
| 6.4.2. | Resultados | 88 |

| | |
|--|------------|
| 6.5. Comentarios | 89 |
| 7. Resultados experimentales | 90 |
| 7.1. Control basado en RBFNN | 91 |
| 7.2. Control basado en ondaleta NN | 93 |
| 7.3. Control LQR | 95 |
| 7.4. Desempeño de los controladores | 97 |
| 7.5. Comentarios | 97 |
| 8. Conclusiones | 99 |
| 8.1. Problemas abiertos y trabajo a futuro | 100 |
| 8.2. Producción académica | 101 |
| Apéndice A. Matlab-Simulink-LEGO EV3 | 102 |
| A.1. Plataforma experimental | 102 |
| A.1.1. Ejecución en modo externo | 104 |
| A.1.2. Descripción de la planta | 105 |
| Apéndice B. Controlador LQR | 106 |
| B.1. Control en espacio de estado | 106 |
| B.1.1. Regulador cuadrático lineal | 107 |
| B.1.2. Modelo lineal en espacio de estado | 108 |

Índice de figuras

| | | |
|------|--|----|
| 1.1. | Péndulo de Furuta; q_1 y q_2 son ángulos de desplazamiento de las articulaciones y τ es el par de entrada [Moreno-Valenzuela and Aguilar-Avelar, 2018]. . . . | 3 |
| 1.2. | Péndulo con rueda inercial; q_1 y q_2 son ángulos de desplazamiento de las articulaciones y τ es el par de entrada [Moreno-Valenzuela and Aguilar-Avelar, 2018] | 4 |
| 1.3. | Sistemas mecánicos subactuados [Moreno-Valenzuela and Aguilar-Avelar, 2018] | 6 |
| 1.4. | Primeros <i>ballbots</i> | 8 |
| 1.5. | <i>Ballbots</i> : prototipo LEGO NXT y vehículo para transporte de personas | 9 |
| 2.1. | Modelo simplificado del <i>ballbot</i> en un plano. | 13 |
| 2.2. | Legó <i>Ballbot</i> , mostrando sistema de coordenadas y estados medibles. | 14 |
| 2.3. | Diagrama eléctrico simplificado del motor. | 18 |
| 4.1. | Cálculo de cotas sobre \tilde{x} | 31 |
| 4.2. | La condición deslizante | 32 |
| 4.3. | Interpretación gráfica de las ecuaciones (4.3) y (4.5) ($n = 2$) | 34 |
| 4.4. | Chattering como un resultado de control imperfecto. | 35 |
| 4.5. | Construcción de Filippov de la dinámica equivalente en modo deslizante. . . . | 36 |
| 4.6. | La capa límite | 43 |
| 4.7. | Control de interpolación en la capa límite | 43 |
| 4.8. | Entrada de control conmutada y resultado del desempeño del error de seguimiento | 44 |
| 4.9. | Entrada de control suave y desempeño del error de seguimiento resultante . . | 45 |
| 5.1. | Estructura general de un controlador difuso | 49 |
| 5.2. | Conjuntos difusos en el bloque de fusificación | 50 |
| 5.3. | Representación gráfica de un defusificador centro promedio. | 52 |
| 5.4. | Configuración general de un sistema difuso adaptable | 54 |
| 5.5. | Diagrama de bloques del controlador difuso | 57 |
| 5.6. | Diagrama de control por modos deslizantes con aproximación difuso adaptable | 60 |

| | |
|--|-----|
| 5.7. Funciones de pertenencia de entrada: (a) Caso I y (b) Caso II | 63 |
| 5.8. Funciones de pertenencia de salida | 65 |
| 5.9. Simulación de control por modos deslizantes basado en aproximación difuso adaptable | 66 |
| 6.1. Primer modelo para una neurona artificial | 72 |
| 6.2. Neurona artificial con polarizador como peso | 73 |
| 6.3. Perceptrón | 73 |
| 6.4. Red neuronal de dos capas | 74 |
| 6.5. Ondaleta Morlet | 79 |
| 6.6. Estructura de la red neuronal de RBF | 82 |
| 6.7. Diagrama de bloques del control por modo deslizante basado en aproximación neuronal adaptable | 84 |
| 6.8. Simulación control por modos deslizantes basado en aproximación neuronal adaptable | 88 |
| 7.1. Resultados experimentales del controlador basado en RBFNN: a) desplazamiento angular q_{1x} del cuerpo; b) desplazamiento angular q_{1y} del cuerpo; c) posición en el plano $x-y$ | 91 |
| 7.2. Señales de control experimentales por cada plano para control basado en RBFNN | 92 |
| 7.3. Evolución de los pesos adaptables y superficie deslizante s_1 por cada plano para control basado en RBFNN | 92 |
| 7.4. Resultados experimentales del controlador basado en funciones ondaleta: a) desplazamiento angular q_{1x} del cuerpo; b) desplazamiento angular q_{1y} del cuerpo; c) posición en el plano $x-y$ | 94 |
| 7.5. Señales de control experimentales por cada plano para control basado en red neuronal con funciones ondaleta | 94 |
| 7.6. Evolución de los pesos adaptables y superficie deslizante s_1 por cada plano para control basado en red neuronal con funciones ondaleta | 95 |
| 7.7. Resultados experimentales del controlador lineal LQR: a) desplazamiento angular q_{1x} del cuerpo; b) desplazamiento angular q_{1y} del cuerpo; c) posición en el plano $x-y$ | 96 |
| 7.8. Señales de control experimentales por cada plano para el control LQR | 96 |
| A.1. “Ladrillo” inteligente, sensores y actuadores del kit LEGO MINDSTORMS EV3 | 103 |
| A.2. Bloques de Simulink para programar el bloque LEGO MINDSTORMS EV3 | 103 |
| A.3. Diagrama general en Simulink de controlador y planta LEGO EV3 | 104 |
| A.4. Configuración de parámetros de Simulink | 105 |
| A.5. Diagrama de bloque de Simulink de la planta LEGO EV3 | 105 |
| B.1. Diagrama de bloques de un controlador en espacio de estado | 107 |

B.2. Bloques en Simulink de controlador LQR en los planos $x - y$ 110
B.3. Controlador LQR en Simulink en el plano x 110

Índice de tablas

| | |
|--|----|
| 2.1. Relaciones de las coordenadas generalizadas de la bola | 14 |
| 2.2. Relaciones de las coordenadas generalizadas del cuerpo | 15 |
| 2.3. Parámetros del <i>ballbot</i> | 19 |
| 5.1. Tabla de búsqueda del sistema difuso adaptable | 65 |
| 6.1. Parámetros de simulación para los controladores neuronales basados en RBF y ondaleta | 87 |
| 7.1. Parámetros experimentales para control neuronal con RBF. | 90 |
| 7.2. Parámetros experimentales para control neuronal con funciones ondaleta . . | 93 |
| 7.3. Norma \mathcal{L}_2 para resultados experimentales | 97 |

Lista de símbolos

| | |
|---------------------|--------------------------------------|
| $\frac{d}{dt}$ | Derivada temporal |
| $\ \cdot \ $ | Norma Euclidiana |
| \mathcal{U} | Energía potencial |
| \mathcal{K} | Energía cinética |
| \equiv | Igual por definición o equivalente |
| \in | “pertenece a” |
| \exists | “existe” |
| \forall | “para todo” |
| \mathbb{R} | Conjunto de los números reales |
| $x^{(n)}$ | Derivada n -ésima |
| $(\cdot \cdot)$ | Función dependiente de dos variables |
| $\hat{}$ | Aproximado |
| $*$ | Ideal u óptimo |
| \cup | Unión |
| \cap | Intersección |
| \prod | Producto |

También se usarán las siguientes locuciones latinas:

i.e. *id est* “esto es”

e.g. *exampli gratia* “por ejemplo”

Introducción

En el presente capítulo se presenta un panorama en lo referente al estado del arte sobre el control de sistemas dinámicos no lineales subactuados, así como los trabajos reportados para el control de un robot móvil subactuado llamado *ballbot*. Finalmente, se presenta el objetivo y organización de este trabajo de tesis.

1.1. Antecedentes

1.1.1. Sistemas subactuados

La condición de subactuación en sistemas mecánicos, robótica, mecatrónica, o sistemas dinámicos se refiere a un sistema con más grados de libertad a ser controlado (número de variables independientes que definen la configuración del sistema) que actuadores (variables de entrada) [Spong, 1994]. Esta restricción implica que algunas de las variables de configuración del sistema no pueden ser controladas directamente lo que complica enormemente el diseño de algoritmos de control. Los sistemas subactuados se encuentran comúnmente en varias áreas y aplicaciones, como robótica, sistemas aeroespaciales, sistemas marinos, sistemas flexibles, sistemas móviles y sistemas de locomotoras. Las razones de la subactuación se resumen en [Olfati-Saber, 2001] de la siguiente manera:

- Dinámica del sistema por naturaleza, e.g., avión, nave espacial, helicóptero, vehículo submarino, grúa y robot móvil con ruedas.
- Por diseño para reducción de costo o algún otro propósito, e.g., satélites con dos propulsores, robots con eslabones flexibles;
- Fallo del actuador, e.g., aeronaves de superficie, despegue y aterrizaje vertical (VTOL).

- Impuesta artificialmente para crear sistemas no lineales complejos de bajo orden con el fin de comprender mejor el control de los sistemas subactuados de alto orden, e.g., el sistema carro-péndulo, el sistema de barra-bola, el péndulo de Furuta, el péndulo con rueda inercial (IWP), el sistema de oscilación traslacional con actuador rotacional (TORA), el Acrobot, el Pendubot, entre otros.

Hoy en día, los sistemas subactuados son un caso de estudio interesante para la comunidad de control automático, y algunos de ellos se han convertido en puntos de referencia, y se utilizan para probar diferentes técnicas de control; ver [Rudra et al., 2014], [Mehdi et al., 2014], [El-Hawwary et al., 2006], [Xu and Hu, 2013], [Liu and Yu, 2013], [Chen and Huang, 2012], [She et al., 2012] y [El-Hawwary et al., 2006], por citar algunos.

El péndulo invertido giratorio, mejor conocido como el péndulo de Furuta en honor a su inventor, es un sistema subactuado bien conocido con 2 gdl (grados de libertad). Este mecanismo consiste en un brazo rotatorio en el plano horizontal con un péndulo unido a la punta del brazo y girando en el plano vertical. El péndulo de Furuta se ilustra en la figura 1.1. El único par de entrada al sistema es aplicado directamente al eje de rotación del brazo horizontal. Luego, dos variables de configuración (desplazamientos del brazo y del péndulo marcadas como q_1 y q_2 en la figura) revelan el estado subactuado del sistema. La motivación para el diseño de este péndulo invertido proviene del sistema clásico del carro-péndulo, ya que con este diseño rotatorio se evita la limitación del movimiento del carro.

El péndulo con rueda inercial (IWP¹), fue presentado por primera vez en [Spong et al., 2001]; es un sistema mecánico subactuado especial, el cual consiste de un péndulo con un disco simétrico unido en su punta. El disco o rueda y el péndulo giran en planos paralelos. El IWP se ilustra en la figura 1.2. El eje de rotación del péndulo es pasivo, i.e., no tiene una actuación directa sobre el desplazamiento angular del péndulo, mientras que un motor aplica directamente un par a la rueda, la cual gira libremente sobre su eje de rotación. Así pues, el péndulo no está actuado y el sistema tiene que ser controlado mediante el momento de rotación de la rueda. Debido a su matriz de inercia constante, el IWP es el primer ejemplo de un sistema mecánico subactuado plano con 2 g.d.l. y un solo actuador [Olfati-Saber, 2001].

En la literatura existen una variedad de sistemas subactuados interesantes. En particular, el Acrobot, mostrado en la figura 1.3a, fue recientemente estudiado en [Donaire et al., 2017]. El control de swing-up y balanceo del Pendubot de la figura 1.3b fue tema de estudio en [Van Kien et al., 2016], el control del sistema híbrido del Pendubot y carro-péndulo (ilustrado en la figura 1.3c) fue presentado en [Flaßkamp et al., 2017]; un controlador PID difuso autosintonizable en [Azeloglu et al., 2016] y un control predictivo generalizado basado en la optimización de enjambres de partículas [Smoczek and Szpytko, 2017] fueron aplicados

¹Inertia Wheel Pendulum - por sus siglas en inglés

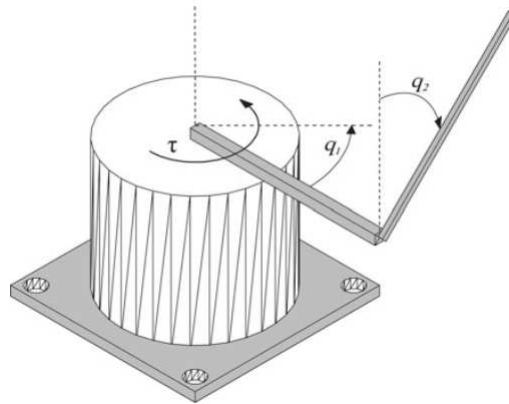


Figura 1.1: Péndulo de Furuta; q_1 y q_2 son ángulos de desplazamiento de las articulaciones y τ es el par de entrada [Moreno-Valenzuela and Aguilar-Avelar, 2018].

al sistema carro-grúa de la figura 1.3d; la estabilización del sistema TORA² de la figura 1.3e ha sido estudiado en [Chang et al., 2016] mediante el uso de esquemas de control no lineales complejos; el control del sistema barra-bola de la figura 1.3f es abordado en [Ramírez-Neria et al., 2016]. Otros sistemas subactuados interesantes son el sistema de levitación magnética [Zhang et al., 2016], robots planares subactuados [Lai et al., 2016], el Capsu-bot [Huda and Yu, 2015], el avión VTOL [Chwa, 2015], robots manipuladores en plataformas móviles [Nguyen and Dankowicz, 2015], vehículos aéreos no tripulados estudiados en trabajos como el de [Pérez-Alcocer et al., 2016], vehículos acuáticos submarinos [Kim et al., 2016], así como satélites y naves espaciales subactuadas [Petersen et al., 2017], entre otros.

1.1.2. Control y dinámica no lineal

Los fenómenos no lineales pueden encontrarse fácilmente en la naturaleza, ya que la mayoría de los sistemas reales son inherentemente no lineales. De hecho, es posible decir que el comportamiento lineal no es natural en el mundo real, porque la dinámica de los sistemas lineales no es lo suficientemente rica como para describir muchos fenómenos comúnmente observados [Sastrey, 1999]. Un sistema no lineal es un sistema cuya dinámica de entrada-salida no satisface los dos principios de linealidad: *superposición* y *homogeneidad*. En ingeniería, la dinámica no lineal se puede encontrar en circuitos eléctricos y electrónicos, sistemas de energía, sistemas mecánicos y reacciones químicas, entre otros; y la descripción de tales dinámicas no lineales utilizando modelos lineales fue el primer enfoque humano para comprenderlas.

Debido a las potentes herramientas desarrolladas en el siglo pasado para los sistemas lineales, el primer paso para analizar un sistema no lineal es, por lo general, linealizarlo sobre algún

²Translational Oscillator with Rotational Actuator - por sus siglas en inglés

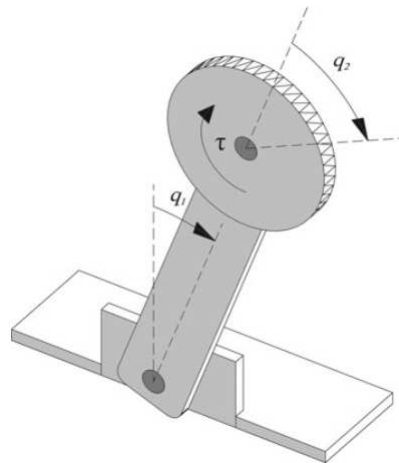


Figura 1.2: Péndulo con rueda inercial; q_1 y q_2 son ángulos de desplazamiento de las articulaciones y τ es el par de entrada [Moreno-Valenzuela and Aguilar-Avelar, 2018]

punto de operación nominal y analizar el modelo lineal resultante. No hay duda de que la linealización es útil para aprender lo más que se pueda sobre el comportamiento de un sistema no lineal. Sin embargo, hay dos limitaciones básicas de la linealización. Primero, dado que la linealización es una aproximación en la vecindad de un punto de operación, sólo puede predecir el comportamiento “local” de las trayectorias del sistema no lineal en la vecindad de ese punto. No puede predecir el comportamiento “no local” lejos del punto de operación y ciertamente no predice el comportamiento “global” en todo el espacio de estado. Segundo, como se mencionó anteriormente, la dinámica de un sistema no lineal es mucho más rica que la dinámica de un sistema lineal. Existen “fenómenos esencialmente no lineales”, como el *tiempo de escape finito*, *los equilibrios múltiples*, *los ciclos límite*, entre otros, que pueden tener lugar solo en presencia de una no linealidad; por lo tanto, no pueden ser descritos o predichos por modelos lineales [Khalil, 2002].

En años recientes, se han introducido una amplia gama de herramientas de análisis y metodologías de control para sistemas no lineales. Por ejemplo, el método de Lyapunov para el análisis de estabilidad de sistemas no lineales y las técnicas de control de realimentación no lineales como la linealización por realimentación, el control por modos deslizantes, el rediseño de Lyapunov, el backstepping y el control basado en pasividad. La linealización por realimentación es una técnica de control utilizada en sistemas no lineales con ciertas propiedades estructurales. La idea de esta técnica es la cancelación de las no linealidades a través de una ley de control de realimentación que depende de las variables de estado, transformando el sistema no lineal en un sistema lineal equivalente. El desarrollo de esta metodología se remonta a principios de los 70 [Krener, 1973] y es bien explicada en [Khalil, 2002] e [Isidori, 1985]. El sistema completo de lazo cerrado en esta metodología se debe encontrar utilizando una transformación de coordenadas que conduzca a la llamada dinámica externa e interna. Una posible desventaja de este enfoque es que, dependiendo de la selección de la función de salida para ca-

da sistema, la dinámica cero puede ser inestable. La idea de utilizar esta técnica para controlar sistemas mecánicos subactuados no es nueva, pero la mayor parte del trabajo se ha dedicado sólo al problema de la estabilización. Por ejemplo en [Acosta and López-Martínez, 2005], se presenta una metodología constructiva basada en la linealización por realimentación y el diseño de Lyapunov para sistemas mecánicos subactuados con 2 gdl. Específicamente, la metodología propuesta se probó con tres sistemas mecánicos subactuados: el péndulo de la rueda de inercial, el carro-péndulo y el péndulo de Furuta.

Por otro lado, se puede diseñar un controlador empleando consideraciones de energía. En esta aproximación, se propone una función definida positiva similar a la energía y la acción de control está diseñada para dar forma a la energía del sistema, tal que la derivada en el tiempo de la función de energía propuesta sea definida negativa. Enfoques como el control basado en pasividad y el rediseño de Lyapunov también se basan en esta idea. Sin embargo, la mayor parte de los trabajos en el diseño de algoritmos basados en energía para el control de sistemas subactuados con 2 gdl se han dedicado a los problemas de estabilización y balanceo.

El control adaptable es una metodología para controlar un sistema sin requerir un conocimiento exacto de su modelo. Esta estrategia de control permite mantener un nivel deseado de rendimiento durante la tarea de control cuando se desconocen los parámetros del modelo dinámico; sin embargo, se requiere que el modelo dinámico sea parametrizable linealmente, lo cual es una propiedad estructural que no se cumple en todos los casos. En años recientes se han aplicado esquemas de control adaptable a algunos sistemas subactuados ([Chwa, 2015], [Nguyen and Dankowicz, 2015] y [Chen and Huang, 2014]). Por otra parte, se han aprovechado las características del control adaptable para combinarse con las propiedades robustas del control por modos deslizantes ([Azar and Serrano, 2015], [Tao et al., 2010], [Yue et al., 2014], por citar algunos) mostrando la ventaja de esta técnica de control con respecto a esquemas lineales y algoritmos basados en el modelo.

Finalmente, las redes neuronales adaptables han llamado la atención de muchos investigadores y profesionales del control, ya que la propiedad de aproximación universal permite aproximar funciones lineales y no lineales desconocidas [Lewis et al., 1998], [Haykin, 1998]. La propiedad de aproximación universal consiste en la expansión de una función no lineal suave mediante funciones sigmoideas (o de otro tipo) ponderadas, más un error de aproximación. Por lo tanto, se puede proponer un controlador adaptable basado en redes neuronales, obviando la necesidad de conocer la estructura del sistema y sus parámetros. En otras palabras, la red neuronal en el controlador compensa el modelo del sistema desconocido y se puede usar en el control de sistemas desconocidos sin requisitos de linealidad en los parámetros del sistema.

En este trabajo de tesis se emplean algunas de las técnicas de control no lineal descritas an-

teriormente; *i.e.*, control por modos deslizantes se combina con las ventajas que presentan las redes neuronales adaptables y los sistemas difusos para resolver el problema de estabilización de un sistema macrótrónico subactuado de 2 gdl (por plano): el péndulo invertido montado sobre una esfera, mejor conocido como *ballbot*.

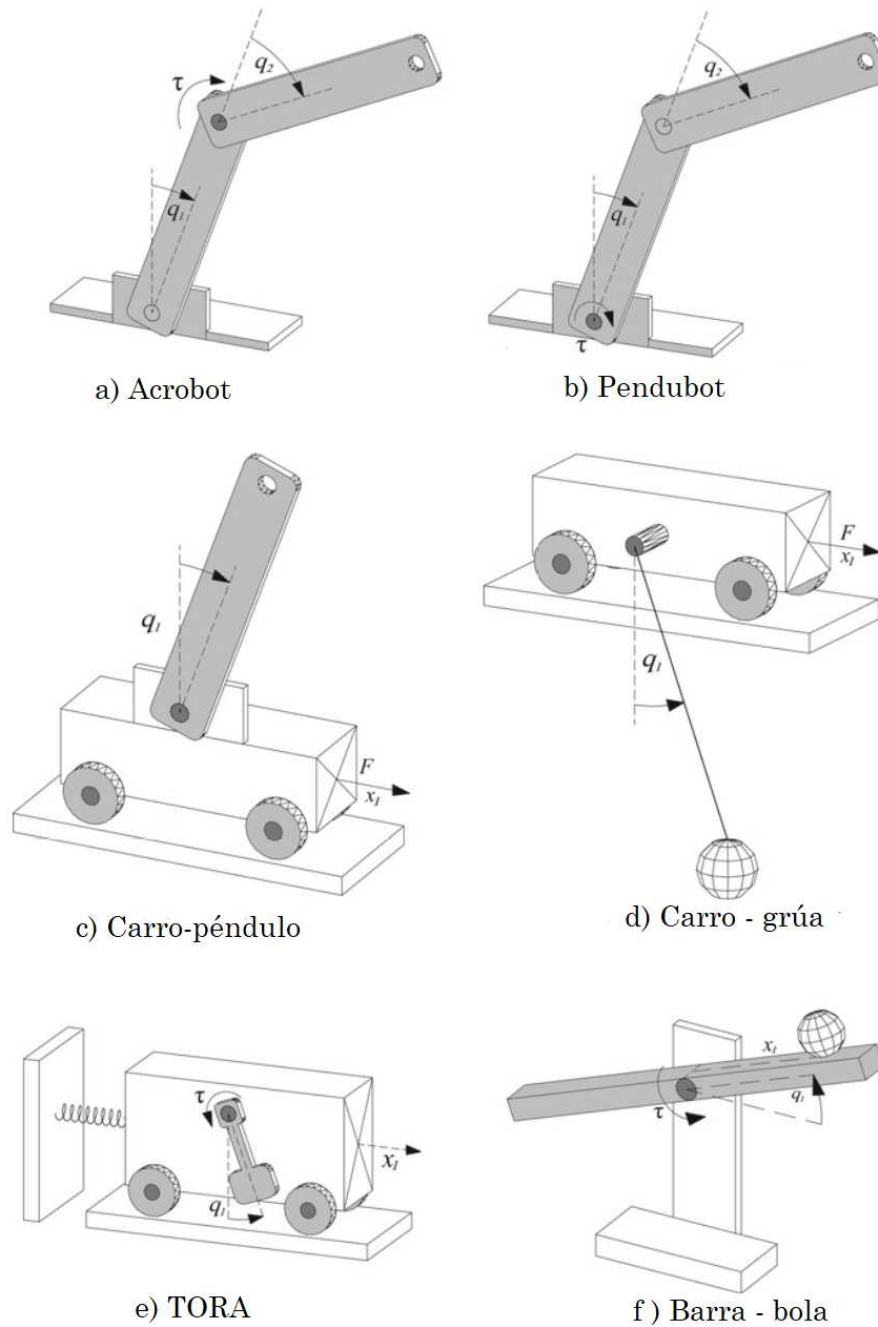


Figura 1.3: Sistemas mecánicos subactuados [Moreno-Valenzuela and Aguilar-Avelar, 2018]

1.2. El *ballbot*

El concepto del *ballbot* es simple: es un robot, el cual se balancea sobre una bola. Por lo tanto, la bola actúa como una rueda esférica, lo que permite que el robot se desplace en cualquier dirección. A diferencia de los robots tradicionales, que dependen de un centro de masa bajo y de una gran base de distancia entre ejes para mantenerse erguidos, el *ballbot* debe equilibrarse activamente, lo que da como resultado una *estabilidad dinámica*.

1.2.1. Motivación

El hombre siempre ha soñado con asistentes personales robóticos, creados para realizar todas sus tareas. La idea de la automatización atrae no sólo a aquellos que quizás son demasiado perezosos para completar una tarea en sí, sino a aquellos que sueñan con aumentar la productividad o la eficiencia dentro de sus propias vidas. Un requisito para tal robot es que sea de tamaño y forma similar a la de un ser humano. Para mantener una forma similar a la humana, la distancia entre ejes de dicho robot debe ser significativamente menor que la altura. Como resultado, en los robots de ruedas múltiples tradicionales, solo un pequeño cambio en la posición del centro de gravedad puede hacer que el robot se vuelva inestable. Esto se puede corregir bajando el centro de gravedad del robot para reducir el brazo de palanca, sin embargo, esto generalmente tiene el costo de un peso muerto significativo que se agrega al robot. Además, la velocidad a la que puede viajar el robot generalmente está limitada por temor a que el impulso del robot haga que se vuelque.

Una solución plausible a este problema, y la desarrollada en proyectos del *ballbot*, es la *estabilidad dinámica*. Es decir, la teoría de control se utiliza para garantizar que el robot se mantenga en posición vertical, sin la necesidad de depender de la estabilidad estática. La estabilidad dinámica se ha utilizado principalmente en la unidad de transporte de una sola persona, el péndulo invertido sobre ruedas. Este sistema tiene dos ruedas alineadas axialmente, y utiliza la solución al problema de control del ‘péndulo invertido’. Este vehículo, sin embargo, se basa en la estabilidad estática en su dirección transversal, una restricción superada por el concepto de *ballbot* que tiene como objetivo producir un robot que sea dinámicamente estable en todas las direcciones.

1.2.2. Antecedentes

Como ya se mencionó, el concepto de un *ballbot* es bastante simple. Consiste en una esfera y un robot que se balancea sobre la misma. El robot puede equilibrarse conduciendo la esfera, haciendo que su base se mueva. Además, el robot también puede desplazarse sobre una superficie, inclinándose y provocando el movimiento de la esfera de manera que no se caiga, sino que se mueva.

(a) *Ballbot* de [Schearer, 2006](b) *Ballbot* de [Kumagai and Ochiai, 2008]Figura 1.4: Primeros *ballbots*

Existen algunos trabajos publicados con sistemas de esta naturaleza. El primero de estos robots, ERROSphere, fue presentado en [Havasi, 2005]. Sin embargo ERROSphere no tiene proporciones similares a las humanas, y por lo tanto este proyecto no se consideró significativo. El primero de este tipo de robots fue construido y probado en la Universidad Carnegie Mellon (CMU, por sus siglas en inglés) en [Schearer, 2006] (figura 1.4a). Este robot recibió el nombre de “*ballbot*” que, en el contexto de este documento, ha sido el nombre de todos estos tipos de robots. El segundo robot fue desarrollado en la Universidad Tohoku Gakuin (TGU) reportado en [Kumagai and Ochiai, 2008] (figura 1.4b). Este *ballbot* es similar, pero utiliza un mecanismo de manejo más complejo, y presenta un solución más elegante. Estos dos *ballbots* son capaces de equilibrarse y moverse de punto a punto, con evidencia de un mayor desarrollo en el *ballbot* CMU incluyendo un soporte retráctil. Recientemente se ha desarrollado un *ballbot* capaz de transportar a una persona (figura 1.5b); el robot es de un tamaño apropiado y tiene una estructura similar a un automóvil montado en un solo neumático esférico con un asiento que un humano puede montar [Ba et al., 2016]. Por otra parte, debido al alto costo de actuadores, sensores, microcontroladores y dispositivos de hardware necesarios para construir un *ballbot* de gran escala, algunos investigadores han construido un sistema *ballbot* empleando el kit LEGO MINDSTORMS. El primer trabajo que abordó la construcción de un proyecto *ballbot* con la tecnología LEGO MINDSTORMS NXT, ha sido reportado por Yorihiisa Yamamoto [Yamamoto, 2009] (figura 1.5a).

El kit LEGO MINDSTORMS NXT es ampliamente usado en algunas áreas de la educación e investigación, cubriendo un cierto número de aplicaciones como plataforma experimental de proyectos de investigación a nivel posgrado, laboratorios y en proyectos de educación superior. Aunque el kit fue inicialmente diseñado como un juguete para niños, su uso en cursos universi-

(a) *Ballbot* de [Yamamoto, 2009](b) *Ballbot* de [Ba et al., 2016]Figura 1.5: *Ballbots*: prototipo LEGO NXT y vehículo para transporte de personas

tarios se ha venido incrementando considerablemente. El kit ha atraído el interés de gente que trabaja en algunas áreas en las que se incluyen inteligencia artificial, sistemas embebidos, sistemas de control, robótica y sistemas operativos [Bobtsov et al., 2009]-[Bobtsov et al., 2011].

Aunque la mayoría de las publicaciones sobre robótica usando LEGO MINDSTORMS se enfocan al control de sistemas de eventos discretos. En [Prieto et al., 2012] se reporta un experimento de control en tiempo continuo sobre un robot *ballbot*, basado en el kit LEGO MINDSTORMS NXT. Un regulador cuadrático lineal (LQR, por sus siglas en inglés) con realimentación completa de estados es utilizado para estabilizar el cuerpo a la vertical. Recientemente se han realizado trabajos sobre el *ballbot* basados en LEGO MINDSTORMS NXT. En [García-García and Arias-Montiel, 2016], se propone un controlador lineal para un prototipo *ballbot* NXT con variaciones paramétricas bajo el enfoque de desigualdades lineales matriciales (LMIs, por sus siglas en inglés), cuyo desempeño es validado a través de resultados experimentales. Por otra parte, en [Jurado and Ontiveros-Guerrero, 2017] se ha propuesto un control adaptable por modelo de referencia (MRAC, por sus siglas en inglés) directo descentralizado para el control de seguimiento para la bola y la estabilización vertical del cuerpo. El desempeño del MRAC descentralizado ha sido validado mediante resultados en simulación considerando presencia de ruido en las mediciones, perturbaciones externas y variaciones paramétricas.

1.3. Objetivo

1.3.1. Objetivo general

El objetivo de este trabajo de tesis es el de replicar la plataforma experimental *ballbot* basándose en el kit LEGO MINDSTORMS EV3; el cual es una versión más moderna en contraste con el kit LEGO MINDSTORMS NXT. Se persigue además, diseñar un control basado en aproximación adaptable (difuso y neuronal) para un sistema tipo *ballbot* y la implementación del controlador en cuestión sobre la plataforma física, para la evaluación experimental del desempeño del mismo.

1.3.2. Objetivos específicos

- Construir una réplica de un robot *ballbot* basándose en la tecnología LEGO MINDSTORMS EV3.
- Estudiar los modelos matemáticos existentes del *ballbot* y pronunciarse por la elección del más conveniente.
- Desarrollar una interfaz de comunicación basada en Simulink de MATLAB para controlar el *ballbot* en tiempo real.
- Estudiar y proponer un controlador basado en alguna estrategia de aproximación adaptable para el control de regulación del *ballbot*.
- Validar el algoritmo de control propuesto en el prototipo experimental del *ballbot*.

1.4. Contribución

La motivación que resulta de los antecedentes ya mencionados, permite que en este trabajo de tesis se replique una plataforma experimental del *ballbot* elaborada con la tecnología LEGO MINDSTORMS EV3, que hasta el momento no se ha reportado. Así también, se proponen dos controladores no lineales basados en una técnica de desacoplamiento por modos deslizantes y basados en una aproximación difuso adaptable y neuronal adaptable, garantizando el acotamiento de las señales en lazo cerrado y la estabilidad del sistema mediante la teoría de Lyapunov. Los resultados en simulación y experimentales validan los algoritmos propuestos.

1.5. Organización de la tesis

El documento de tesis está organizado en ocho capítulos. En el capítulo 2 se describe el modelo dinámico del *ballbot*. En el capítulo 3 se exponen algunos preliminares matemáticos básicos sobre sistemas no lineales y teoría de Lyapunov. En el capítulo 4 se presenta la teoría básica sobre control por modos deslizantes, la cual es la base para los esquemas de control propuestos. En el capítulo 5, se presenta un controlador por modos deslizantes basado en aproximación difuso adaptable así como el análisis que garantiza el acotamiento de las señales en lazo cerrado. En el capítulo 6, se presenta un controlador por modos deslizantes basado en aproximación neuronal adaptable y el correspondiente análisis que asegura el acotamiento de las señales. El capítulo 7, muestra los resultados experimentales del controlador basado en aproximación neuronal adaptable. Por último en el capítulo 8 se presentan las conclusiones y trabajo futuro de este trabajo de tesis.

Modelo dinámico del *ballbot*

2.1. Consideraciones

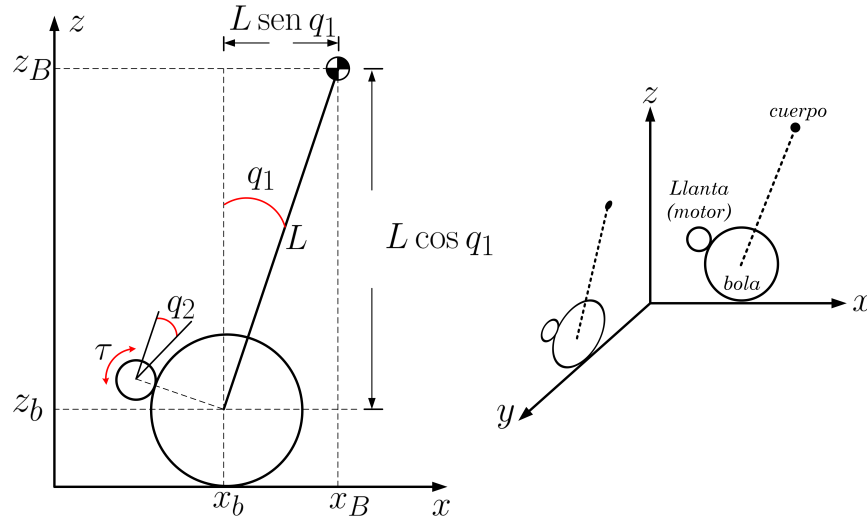
La dinámica del *ballbot* es compleja, por esta razón, es difícil de manejar a menos que se hagan suposiciones para simplificar su modelado. Un concepto simplificado, consiste en considerar un cuerpo rígido balanceándose sobre una esfera rígida. Para este modelo puede suponerse que el movimiento en dos planos –alabeo y cabeceo– está desacoplado, y que las ecuaciones de movimiento son idénticas en ambos planos [Lauwers et al., 2005]. Otras suposiciones incluyen modelado de la fricción del sistema solamente como fricción viscosa, mientras se desprecian sus efectos no lineales y estáticos [Lauwers et al., 2005]-[Yamamoto, 2009]. Otros modelos más complejos proponen modelar toda la dinámica 3D del *ballbot* con la metodología de Newton-Euler y los ángulos de Tait-Bryan para describir el modelo en términos de los parámetros físicos del sistema sin tener que recurrir a una solución numérica [Bonci, 2016].

2.2. Descripción del modelo

La dinámica del modelo simplificado del *ballbot* puede obtenerse utilizando el enfoque lagrangiano y ha sido reportada en [Lauwers et al., 2005] y [Liao et al., 2008].

La elección de coordenadas \mathbf{q} es arbitraria, con la restricción de que deben definir completamente el sistema. Sin embargo, es ventajoso utilizar coordenadas que se relacionen directamente con cantidades medibles, como el ángulo del cuerpo y el ángulo cuerpo-motor [Scheerer, 2006].

Para facilitar la obtención de las ecuaciones de movimiento del *ballbot*, se consideran las siguientes suposiciones:


 Figura 2.1: Modelo simplificado del *ballbot* en un plano.

- El *ballbot* se compone de dos partes: un cuerpo rígido sobre una bola rígida.
- Los pares de control son aplicados entre el cuerpo y la bola a través de una llanta.
- El movimiento se desacopla en los planos de alabeo y balanceo, y las ecuaciones de movimiento son idénticas en estos planos.
- Sólo la fricción viscosa está presente.
- No existe deslizamiento.

Las coordenadas generalizadas se eligen para que coincidan con las cantidades anticipadas que se pueden medir directamente; ángulo del cuerpo y ángulo del eje del motor. Esto resulta en el modelo simplificado del *ballbot* en un plano; el plano x - z , se muestra en la figura 2.1 y un modelo idéntico existe en el plano y - z .

Las variables y parámetros definidos para la obtención del modelo¹ son: q_1 : ángulo de inclinación del cuerpo con respecto a su vertical; q_2 : ángulo del eje del motor (actuador); R_b : radio de la bola; R_w : radio de la llanta; L : distancia del centro de masa del cuerpo al centro de la bola; m_B : masa del cuerpo; m_b : masa de la bola; I_b : momento de inercia de la bola; I_B : momento de inercia del cuerpo; I_M : momento de inercia del motor (incluyendo las llantas) sobre su eje de rotación; δ : relación de engranes, motor a bola, incluyendo dirección $\delta = -\frac{R_w}{R_b}$; f_{B_b} : coeficiente de fricción viscosa entre cuerpo y la bola; f_{B_g} : coeficiente de fricción viscosa entre el cuerpo y el suelo; k_b : constante de fuerza electromotriz de los motores; k_t : constante de par de los motores; R_m : resistencia de los motores; \mathbf{g} : aceleración de la gravedad.

¹Estos parámetros son por cada plano.

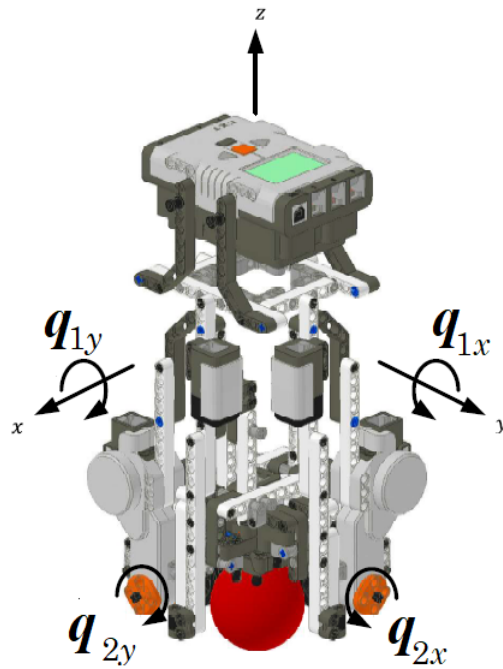


Figura 2.2: Lego *Ballbot*, mostrando sistema de coordenadas y estados medibles.

El prototipo experimental es mostrado en la figura 2.2, donde se ilustra el marco de coordenadas y variables medibles.

Relaciones de las coordenadas generalizadas de la bola

Para la bola:

Tabla 2.1: Relaciones de las coordenadas generalizadas de la bola

| posición | | velocidad | |
|---------------------------|-------------------------|--------------------------------|-------------------------------------|
| ángulo | $q_1 + \delta q_2$ | velocidad angular | $\dot{q}_1 + \delta \dot{q}_2$ |
| posición en x , (x_b) | $R_b(q_1 + \delta q_2)$ | velocidad en x , \dot{x}_b | $R_b(\dot{q}_1 + \delta \dot{q}_2)$ |
| posición en z , (z_b) | 0 | velocidad en z , \dot{z}_b | 0 |

La energía cinética lineal de la bola es:

$$\mathcal{K}_{\text{lin}_b} = \frac{1}{2} m_b R_b^2 (\dot{q}_1 + \delta \dot{q}_2)^2 \quad (2.1)$$

La energía cinética rotacional de la bola es:

$$\mathcal{K}_{\text{rot}_b} = \frac{1}{2} I_b (\dot{q}_1 + \delta \dot{q}_2)^2 \quad (2.2)$$

La energía potencial de la bola es: $\mathcal{U}_b = 0$

Relaciones de las coordenadas generalizadas del cuerpo

Tabla 2.2: Relaciones de las coordenadas generalizadas del cuerpo

| Posición | | Velocidad | |
|-----------------------------|---------------------------|--------------------------------|------------------------------------|
| ángulo | q_1 | velocidad angular | \dot{q}_1 |
| posición en x , (x_B) | $x_b + L \text{sen}(q_1)$ | velocidad en x , \dot{x}_B | $\dot{x}_b + L \cos(q_1)\dot{q}_1$ |
| posición en z , (z_B) | $L \cos(q_1)$ | velocidad en z , \dot{z}_B | $-L \text{sen}(q_1)\dot{q}_1$ |

La energía cinética lineal del cuerpo $\mathcal{K}_{\text{lin}_B}$ es:

$$\begin{aligned} \mathcal{K}_{\text{lin}_B} &= \frac{1}{2}m_B (\dot{x}_B^2 + \dot{z}_B^2) \\ &= \frac{1}{2}m_B [R_b^2 \dot{q}_1^2 + 2\delta R_b^2 \dot{q}_2 \dot{q}_1 + \delta^2 R_b^2 \dot{q}_2^2 + 2 \cos q_1 R_b L \dot{q}_1^2 + 2\delta L R_b \dot{q}_2 \dot{q}_1 \cos q_1 + L^2 \dot{q}_1^2] \end{aligned}$$

La energía cinética rotacional del cuerpo es:

$$\mathcal{K}_{\text{rot}_B} = \frac{1}{2}I_B \dot{q}_1^2 \quad (2.3)$$

La energía potencial del cuerpo es:

$$\mathcal{U}_B = m_B g L \cos(q_1) \quad (2.4)$$

Relaciones de las coordenadas generalizadas del motor

La energía cinética rotacional del motor es:

$$\mathcal{K}_{\text{rot}_m} = \frac{1}{2}I_M (\dot{q}_2 + \dot{q}_1)^2 \quad (2.5)$$

2.2.1. Ecuaciones de Euler-Lagrange

El lagrangiano se define como la diferencia entre la energía cinética total \mathcal{K} y la energía potencial total \mathcal{U} :

$$\begin{aligned} \mathcal{L} &= \mathcal{K} - \mathcal{U} \\ \mathcal{L} &= \mathcal{K}_{\text{lin}_b} + \mathcal{K}_{\text{lin}_B} + \mathcal{K}_{\text{rot}_b} + \mathcal{K}_{\text{rot}_B} + \mathcal{K}_{\text{rot}_m} - \mathcal{U}_b - \mathcal{U}_B \end{aligned} \quad (2.6)$$

Las correspondientes ecuaciones de movimiento son obtenidas utilizando las ecuaciones de Lagrange

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}_i} \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} = \mathbf{F} \quad (2.7)$$

donde

$$\mathbf{q} = [q_1 \quad q_2]^\top$$

representa las variables generalizadas una por cada grado de libertad del sistema, y

$$\mathbf{F} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \tau_{\text{ext}} = \begin{bmatrix} 0 \\ \tau \end{bmatrix}$$

denota las fuerzas externas aplicadas al sistema. Una vez especificado esto, se tiene que:

$$\begin{aligned} \left(\frac{\partial \mathcal{L}}{\partial q_1} \right) &= -\text{sen}(q_1)m_B R_b L \dot{q}_1^2 - m_B \delta L R_b \text{sen}(q_1) \dot{q}_2 \dot{q}_1 + m_B \mathbf{g} L \text{sen}(q_1) \\ \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right) &= m_b R_b^2 \ddot{q}_1 + m_b R_b^2 \delta \ddot{q}_2 + m_B R_b^2 \ddot{q}_1 + m_B \delta R_b^2 \dot{q}_2 + m_B L^2 \dot{q}_1 + 2m_B \cos(q_1) L R_b \dot{q}_1 \\ &\quad + \cos(q_1) \delta R_b \dot{q}_2 + I_b \dot{q}_1 + I_b \delta \dot{q}_2 + I_B \dot{q}_1 + I_M \dot{q}_2 + I_M \dot{q}_1 \\ \left(\frac{\partial \mathcal{L}}{\partial q_2} \right) &= 0 \\ \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \right) &= m_b R_b^2 (\dot{q}_1 + \delta \dot{q}_2) \delta + m_B \delta R_b^2 \dot{q}_1 + m_B \delta^2 R_b^2 \dot{q}_2 + m_B \cos(q_1) \delta L R_b \dot{q}_2 + I_b (\dot{q}_1 + \delta \dot{q}_2) \delta \\ &\quad + I_M (\dot{q}_1 + \dot{q}_2) \end{aligned}$$

Ahora se tiene que las derivadas con respecto al tiempo son:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right) &= m_b R_b^2 \ddot{q}_1 + m_b R_b^2 \delta \ddot{q}_2 + m_B R_b^2 \ddot{q}_1 + m_B \delta R_b^2 \ddot{q}_2 + m_B L^2 \ddot{q}_1 \\ &\quad + 2m_B L R_b [\cos(q_1) \ddot{q}_1 + \dot{q}_1^2 (-\text{sen}(q_1))] + m_B \delta R_b [\cos(q_1) \ddot{q}_2 + \dot{q}_2 (-\text{sen}(q_1)) \dot{q}_1] \\ &\quad + I_b \ddot{q}_1 + I_b \delta \ddot{q}_2 + I_B \ddot{q}_1 + I_M \ddot{q}_2 + I_M \ddot{q}_1 \\ \frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \right) &= m_b R_b^2 \delta \ddot{q}_1 + m_b R_b^2 \delta^2 \ddot{q}_2 + m_B \delta R_b^2 \ddot{q}_1 + m_B \delta^2 R_b^2 \ddot{q}_2 \\ &\quad + m_B \delta L R_b [\cos(q_1) \ddot{q}_1 + \dot{q}_1^2 (-\text{sen}(q_1))] + I_b \delta \ddot{q}_1 + \delta^2 I_b \ddot{q}_2 + I_M \ddot{q}_2 + I_M \ddot{q}_1 \end{aligned}$$

De la ecuación (2.7) se obtiene la primera ecuación del movimiento de Lagrange

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{q}_1} \right] - \frac{\partial \mathcal{L}}{\partial q_1} = F_1$$

resultando en

$$m_{11}\ddot{q}_1 + m_{12}\ddot{q}_2 - m_B L R_b \sin(q_1) \dot{q}_1^2 - m_B g L \sin(q_1) = F_1 \quad (2.8)$$

suponiendo el hecho de que no existe deslizamiento entre la bola y el suelo, entonces $F_1 = 0$, donde:

$$\begin{aligned} m_{11} &= I_B + I_M + I_b + L^2 m_B + m_B R_b^2 + m_b R_b^2 + 2L m_B R_b \cos(q_1) \\ m_{12} &= I_M + I_b \delta + m_B \delta R_b^2 + m_b \delta R_b^2 + L m_B \delta R_b \cos(q_1) \end{aligned}$$

La segunda ecuación del movimiento de Lagrange es

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{q}_2} \right] - \frac{\partial \mathcal{L}}{\partial q_2} = F_2$$

resultando en:

$$m_{21}\ddot{q}_1 + m_{22}\ddot{q}_2 - L m_B \delta R_b \dot{q}_1^2 = F_2 \quad (2.9)$$

siendo:

$$\begin{aligned} m_{21} &= I_M + I_b \delta + m_B \delta R_b^2 + m_b \delta R_b^2 + L m_B \delta R_b \cos(q_1) \\ m_{22} &= I_M + I_b \delta^2 + m_B \delta^2 R_b^2 + m_b \delta^2 R_b^2 \end{aligned}$$

El sistema puede escribirse en su forma matricial de segundo orden:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\dot{\mathbf{q}}) = \boldsymbol{\tau}_{\text{ext}} \quad (2.10)$$

La matriz de inercias $M(\mathbf{q})$ está dada por:

$$M(\mathbf{q}) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \quad (2.11)$$

La matriz de fuerzas centrífugas y de Coriolis se define como:

$$C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -L m_B R_b \sin(q_1) \dot{q}_1 & 0 \\ -L m_B \delta R_b \sin(q_1) \dot{q}_1 & 0 \end{bmatrix} \quad (2.12)$$

El vector de fuerzas gravitacionales está dado por:

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} -g L m_B \sin(q_1) \\ 0 \end{bmatrix} \quad (2.13)$$

El vector de fuerzas de fricción es:

$$\mathbf{f}(\dot{\mathbf{q}}) = \begin{bmatrix} f_{11} \\ f_{21} \end{bmatrix} = \begin{bmatrix} f_{B_g} \dot{q}_1 \\ f_{B_b} \dot{q}_2 \end{bmatrix} \quad (2.14)$$

siendo f_{B_g} y f_{B_b} los coeficientes de fricción entre el cuerpo-suelo y cuerpo-esfera, respectivamente. Finalmente el vector de pares externos se define como:

$$\boldsymbol{\tau}_{\text{ext}} = [0 \quad \tau]^\top \quad (2.15)$$

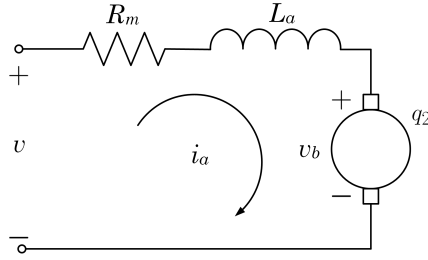


Figura 2.3: Diagrama eléctrico simplificado del motor.

Modelo eléctrico del actuador

La mayor limitación del resultado anterior es que las fuerzas generalizadas τ_{ext} son modeladas simplemente como pares, y no consideran la dinámica eléctrica del actuador. Un modelo el cual incluye el efecto de la dinámica del motor se usó en [Yamamoto, 2009]. En la figura 2.3 se describe el circuito simplificado del motor de cd (actuador). Si la fricción dentro del motor es despreciable, la ecuación de voltaje de malla es:

$$L_a \frac{d(i_a)}{dt} = v - R_m i_a - v_b \quad (2.16)$$

donde v es el voltaje de entrada, R_m es la resistencia del motor, L_a es la inductancia del motor, v_b es la fuerza electromotriz (EMF, por sus siglas en inglés), definida como $v_b = k_b \frac{d(q_2)}{dt}$. Con fines de simplificar el análisis, se considera que la inductancia del motor es despreciable, *i.e.*, $L_a \approx 0$. De esta manera

$$i_a = \frac{v - k_b \dot{q}_2}{R_m} \quad (2.17)$$

Como la fricción dentro del motor se desprecia, el par del motor puede expresarse en términos de la corriente del motor como:

$$\tau = k_t i_a \quad (2.18)$$

donde k_t es la constante de par del motor. Por lo tanto, reemplazando la dinámica de la corriente (2.17) del motor de cd en la ecuación del par (2.18) resulta que el par del motor considerando el modelo eléctrico del actuador es:

$$\tau = \frac{k_t (v - k_b \dot{q}_2)}{R_m} \quad (2.19)$$

Finalmente, de (2.15) el vector de pares externos se define como:

$$\tau_{\text{ext}} = \begin{bmatrix} 0 \\ \frac{k_t (v - k_b \dot{q}_2)}{R_m} \end{bmatrix} \quad (2.20)$$

2.2.2. Parámetros del modelo

Los parámetros relacionados a las dimensiones físicas tales como masa del cuerpo (m_B), masa de la bola (m_b), radio de la esfera (R_b) y radio de la llanta (R_w), se midieron experimentalmente. El valor de la distancia al centro de masa del cuerpo (L) fue tomado como un valor aproximado de [Fong et al., 2009], mientras que los valores correspondientes al momento de inercia del cuerpo (I_B), bola (I_b) se calcularon de acuerdo a las relaciones reportadas en [Yamamoto, 2009] y que se muestran en la Tabla 2.3. Finalmente los parámetros eléctricos de los motores EV3, fueron tomados de [Akmal et al., 2017]. En la tabla 2.3, se muestran los parámetros empleados.

Tabla 2.3: Parámetros del *ballbot*

| Párametro | Símbolo | Valor | Unidades |
|--------------------------------|-----------|------------------------|-------------------|
| Masa del cuerpo | m_B | 0.736 | kg |
| Altura del centro de masa | L | 0.132 | m |
| Masa de la esfera | m_b | 0.0126 | kg |
| Radio de la esfera | R_b | 0.026 | m |
| Radio de la llanta | R_w | 0.0186 | m |
| Momento de inercia del cuerpo | I_B | $\frac{m_B L^2}{3}$ | kg m ² |
| Momento de inercia de la bola | I_b | $\frac{2m_b R_b^2}{3}$ | kg m ² |
| Relación de engranes | δ | $-\frac{R_w}{R_b}$ | – |
| Momento de inercia del motor | I_M | 1×10^{-5} | kg m ² |
| Coef de fricción cuerpo-esfera | f_{B_b} | 0.0022 | Nms/rad |
| Coef de fricción cuerpo-suelo | f_{B_g} | 0 | Nms/rad |
| Constante de par del motor | k_t | 0.317 | Nm/A |
| Constante de EMF del motor | k_b | 0.468 | V s/rad |
| Resistencia del motor | R_m | 6.69 | Ω |
| Aceleración de la gravedad | g | 9.81 | m/s ² |

2.3. Representación en espacio de estado

Sean las variables de estado $x_1 = q_1, x_2 = \dot{q}_1, x_3 = q_2$ y $x_4 = \dot{q}_2$. El sistema (2.10) puede escribirse en la forma:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= f_1(\mathbf{x}) + g_1(\mathbf{x})u \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= f_2(\mathbf{x}) + g_2(\mathbf{x})u
 \end{aligned} \tag{2.21}$$

cuyas funciones $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, $g_1(\mathbf{x})$ y $g_2(\mathbf{x})$, son funciones no lineales. Suponiendo que se conocen todos los parámetros del sistema, los valores de estas funciones en términos de los parámetros del *ballbot*, por cada plano, son:

$$\begin{aligned} f_1(\mathbf{x}) &= \frac{1}{|M|} [-m_{22}c_{11}x_2 - m_{22}\mathbf{g}_{21} + m_{12}c_{21}x_2 + m_{12}f_{21}] \\ g_1(\mathbf{x}) &= -\frac{1}{|M|}m_{12} \\ f_2(\mathbf{x}) &= \frac{1}{|M|} [-m_{21}c_{11}x_2 - m_{21}\mathbf{g}_{11} + m_{11}c_{21}x_2 - m_{11}f_{21}] \\ g_2(\mathbf{x}) &= -\frac{1}{|M|}m_{11} \end{aligned}$$

siendo M la matriz de inercias; \mathbf{g}_{21} , \mathbf{g}_{11} , son los elementos del vector de fuerzas gravitacionales y f_{21} el segundo elemento del vector de fuerzas de fricción $\mathbf{f}(\dot{\mathbf{q}})$.

Capítulo 3

Preliminares

Este capítulo introduce algunos conceptos teóricos preliminares que serán útiles en el presente trabajo. El determinante de una matriz A es indicado como $|A|$, mientras la notación $|x|$ indica el valor absoluto de una variable real $x \in \mathbb{R}$. La norma de un vector $\mathbf{x} \in \mathbb{R}^n$ se define como $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}}$.

3.1. Fundamentos de sistemas no lineales

Un sistema dinámico no lineal puede representarse mediante un número finito de n ecuaciones diferenciales no lineales en la llamada forma de *espacio de estado*:

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2, \dots, x_n), \\ \dot{x}_2 &= f_2(x_1, x_2, \dots, x_n), \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, \dots, x_n),\end{aligned}$$

donde \dot{x}_i , representa la derivada de x_i con respecto al tiempo. Las variables x_1, x_2, \dots, x_n son los estados del sistema, los cuales son las variables que proporcionan una descripción completa del sistema en un momento determinado, y f_1, f_2, \dots, f_n son funciones no lineales que describen la dinámica completa del sistema. Bajo esta notación, el sistema en espacio de estado puede ser escrito en forma de vector como

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{3.1}$$

donde $\mathbf{f} \in \mathbb{R}^n$ es una función vectorial no lineal y $\mathbf{x} \in \mathbb{R}^n$, es el vector de estado. A un valor particular del vector del espacio de estado se le llama un *punto*, debido a que

corresponde a un punto del espacio de estado o a una configuración particular del sistema. El número de estados n es llamado *orden* del sistema. La solución $\mathbf{x}(t)$ de (3.1), la cual usualmente corresponde a una curva en el espacio de estados conforme el tiempo t varía de cero a infinito, es generalmente conocida como una *trayectoria de estado* o *trayectoria del sistema* [Slotine and Li, 1991]. Lo más importante en la ingeniería de control de sistemas es el comportamiento de las trayectorias del sistema $\mathbf{x}(t)$ alrededor de algún punto de operación o movimiento nominal deseado, el cual introduce la noción de *estabilidad*.

Si una trayectoria del sistema corresponde a un solo punto a lo largo de su evolución temporal completa, entonces tal punto es un equilibrio. En otras palabras, un valor particular \mathbf{x}^* del vector de estados es un *punto de equilibrio* del sistema (3.1) si, para la condición inicial $\mathbf{x}(0) = \mathbf{x}^*$, las trayectorias del sistema $\mathbf{x}(t)$ permanecen iguales a \mathbf{x}^* para todo tiempo t mayor que cero. Esto también significa que el vector constante \mathbf{x}^* , el cual es un punto de equilibrio, satisface $\mathbf{f}(\mathbf{x}^*) = \mathbf{0}$. El concepto de equilibrio es muy importante para el estudio de estabilidad, ya que muchos problemas de estabilidad se formulan naturalmente con respecto a los puntos de equilibrio [Slotine and Li, 1991].

El sistema no lineal (3.1) se dice que es *autónomo*, dado que \mathbf{f} no depende explícitamente del tiempo t . Por el otro lado, el sistema es llamado *no-autónomo* si el vector de función no lineal \mathbf{f} depende explícitamente del tiempo t , tal que

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}). \quad (3.2)$$

Una propiedad análoga para sistemas lineales depende de si la matriz de coeficientes A varía con el tiempo o no, clasificándose como variantes en el tiempo o invariantes en el tiempo, respectivamente.

Además, es importante tener en cuenta que las ecuaciones de estado (3.1) y (3.2) se describen sin la presencia explícita de una variable de entrada \mathbf{u} , es decir, la así llamada ecuación de estado es *no forzada*. Existen dos razones para trabajar con una ecuación de estado no forzada: (i) se considera que la entrada del sistema es cero; (ii) la ecuación de estado representa la dinámica de lazo cerrado de un sistema realimentado compuesto por el controlador y la planta, con la entrada de control \mathbf{u} como una función del estado \mathbf{x} , una función del tiempo t , o ambas, y por lo tanto desapareciendo en la dinámica de lazo cerrado ([Slotine and Li, 1991],[Khalil, 2002]).

Una representación más general de un sistema no lineal en lazo abierto, es decir, la planta del sistema en cuestión, es la ecuación diferencial vectorial de primer orden n -dimensional

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}) \quad (3.3)$$

con el vector de funciones no lineales $\mathbf{f} \in \mathbb{R}^n$ dependiente de la variable de tiempo $t \in \mathbb{R}^+$, el vector de estados $\mathbf{x} \in \mathbb{R}^n$, y el vector de entrada $\mathbf{u} \in \mathbb{R}^p$ definido como una función del estado \mathbf{x} y el tiempo t .

3.2. Propiedades fundamentales

Una función vectorial generalizada $\mathbf{f}(t, \mathbf{x})$ se dice que es Lipschitz si satisface la *condición Lipschitz*, la cual es dada por la desigualdad

$$\|\mathbf{f}(t, \mathbf{x}) - \mathbf{f}(t, \mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad (3.4)$$

para todo (t, \mathbf{x}) y (t, \mathbf{y}) en alguna vecindad (t_0, \mathbf{x}_0) , con $\mathbf{x}(t_0) = \mathbf{x}_0$. Para el caso de que $\mathbf{f}(t, \mathbf{x})$, sea una función autónoma y escalar la condición Lipschitz puede ser escrita como

$$|f(x) - f(y)| \leq L|x - y|, \quad (3.5)$$

Entonces se dice que f es una *función Lipschitz*, lo cual implica que sobre un plano de $f(x)$ contra x , una línea recta que une dos puntos de $f(x)$ no puede tener una pendiente cuyo valor absoluto sea mayor que la llamada *constante de Lipschitz*, L [Khalil, 2002].

La existencia y unicidad de la solución de $\mathbf{x}(t)$ para un sistema de ecuaciones diferenciales de la forma (3.1) ó (3.2), para una condición inicial dada del estado $\mathbf{x}(t_0) = \mathbf{x}_0$ con $t > t_0$, se puede garantizar si el lado derecho de (3.1) ó (3.2) satisfacen la condición Lipschitz en (3.4) [Khalil, 2002].

3.3. Conceptos de estabilidad

Como se mencionó anteriormente, la noción intuitiva de estabilidad de los sistemas dinámicos es una especie de buen comportamiento de las trayectorias del sistema alrededor de un punto de operación deseado (típicamente un equilibrio) además de que las trayectorias del sistema permanecerán cercanas al equilibrio si el sistema es ligeramente perturbado. Por conveniencia se considera que el punto bajo análisis es el origen del espacio de estado, esto es, $\mathbf{x}^* = \mathbf{0}$, dado que cualquier punto de equilibrio puede cambiarse al origen mediante un cambio de coordenadas. Por lo tanto, todas las definiciones de estabilidad serán dadas como un problema de estabilidad alrededor del origen del espacio de estado ([Slotine and Li, 1991],[Khalil, 2002]).

Definición 3.1 El punto de equilibrio $\mathbf{x} = \mathbf{0}$ del sistema (3.1) es estable si, para cada $\varepsilon > 0$, existe $\delta = \delta(\varepsilon) > 0$ tal que

$$\|\mathbf{x}(0)\| < \delta \Rightarrow \|\mathbf{x}(t)\| < \varepsilon, \forall t \geq 0. \quad (3.6)$$

De otra manera, es inestable.

Esta definición de estabilidad (también llamada *estabilidad en el sentido de Lyapunov*) requiere la existencia de $\delta > 0$ para cada $\varepsilon > 0$. Esto es, por cada valor de ε , debe existir un valor δ , posiblemente dependiente de ε , de modo que una trayectoria que comience en una vecindad δ del origen nunca dejará la vecindad ε [Khalil, 2002].

Definición 3.2 El punto de equilibrio $\mathbf{x} = \mathbf{0}$ del sistema (3.1) es asintóticamente estable si es estable y δ puede elegirse tal que

$$\|\mathbf{x}(0)\| < \delta \Rightarrow \lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{0}. \quad (3.7)$$

La estabilidad asintótica implica que el punto de equilibrio es estable, y que, además, las trayectorias del sistema que inician cerca del origen en realidad convergen a éste conforme el tiempo t se aproxima a infinito. El conjunto de todos los puntos tal que las trayectorias iniciadas en estos puntos eventualmente convergen al origen es llamado *dominio de atracción*. Un punto de equilibrio el cual es estable en el sentido de Lyapunov pero no asintóticamente estable es llamado *marginalmente estable* [Slotine and Li, 1991].

Definición 3.3 El punto de equilibrio $\mathbf{x} = \mathbf{0}$ del sistema (3.1) es exponencialmente estable si existe $\alpha > 0$ y $\lambda > 0$ tal que

$$\|\mathbf{x}(t)\| \leq \alpha \|\mathbf{x}(0)\| e^{-\lambda t}, \quad \forall t \geq 0. \quad (3.8)$$

Esto significa que las trayectorias del sistema que inician en el dominio de atracción de un equilibrio exponencialmente estable convergen a él más rápido que una función exponencial. De hecho, al número positivo λ se le conoce como *velocidad de convergencia*. Nótese que estabilidad exponencial implica estabilidad asintótica, pero lo contrario es falso [Slotine and Li, 1991].

Definición 3.4 Si el equilibrio $\mathbf{x} = \mathbf{0}$ del sistema (3.1) satisface la definición de estabilidad en la Definición 3.1, estabilidad asintótica en la Definición 3.2, o estabilidad exponencial en la Definición 3.3 para cualquier condición inicial $\mathbf{x}(t)$, entonces se dice que el punto de equilibrio es *globalmente* (asintóticamente o exponencialmente) estable.

De lo contrario, el comportamiento del sistema se caracteriza sólo *localmente*. Esto es, sólo se sabe como evolucionan las trayectorias del sistema después de comenzar cerca del punto de equilibrio, pero no se tiene información sobre cómo se comportará el sistema cuando las condiciones iniciales estén a cierta distancia del equilibrio.

El método directo de Lyapunov se puede utilizar para estudiar la estabilidad de los sistemas de forma local o global. El análisis local se ocupa de la estabilidad en una vecindad del punto de equilibrio, mientras que el análisis global describe las propiedades de estabilidad en todo el espacio de estado. Este análisis usualmente involucra una *función definida positiva* de forma local o global, *i.e.*, una función estrictamente positiva, la cual es nula únicamente en el origen.

Teorema 3.1 (*Estabilidad*) Si, en una bola B , la cual contiene el origen, existe una función escalar $V(\mathbf{x})$ con primeras derivadas parciales continuas a lo largo de las trayectorias (3.1) tal que

- $V(\mathbf{x})$ es definida positiva localmente en B .

- $\dot{V}(\mathbf{x})$ es semidefinida negativa localmente en B

entonces el origen de (3.1) es estable. Si la bola B , puede ser expandida para abarcar a todo el espacio de estado, entonces el origen es globalmente estable [Slotine and Li, 1991].

Teorema 3.2 (Estabilidad asintótica) Si en una bola B , la cual contiene el origen, existe una función escalar $V(\mathbf{x})$ con primeras derivadas parciales continuas a lo largo de las trayectorias (3.1) tal que

- $V(\mathbf{x})$ es definida positiva localmente en B .
- $\dot{V}(\mathbf{x})$ es definida negativa localmente en B

entonces el origen de (3.1) es asintóticamente estable. Si la bola B , puede ser expandida para abarcar a todo el espacio de estado, y

- $V(\mathbf{x}) \rightarrow \infty$ conforme $\|\mathbf{x}\| \rightarrow \infty$,

entonces el equilibrio en el origen es globalmente asintóticamente estable [Slotine and Li, 1991].

Para estabilidad asintótica global, una condición adicional que la función V tiene que satisfacer es que $V(\mathbf{x})$ debe ser *radialmente desacetada*, lo cual significa que $V(x)$ tiende a infinito conforme \mathbf{x} se aproxima a infinito en cualquier dirección, tal como lo indica la última condición de este Teorema.

3.4. Lema de Barbalat

Para sistemas autónomos es posible demostrar que el límite de una función se aproxima a cero haciendo uso de ciertas propiedades particulares obtenidas del análisis funcional.

3.4.1. Espacios de funciones

Una clase particular de espacios vectoriales son los denominados espacios L_p^n . Los elementos o vectores de los espacios L_p^n son funciones con propiedades particulares.

Los espacios lineales denominados L_2^n y L_∞^n que se definirán a continuación, son particularmente empleados en el análisis de sistemas dinámicos interconectados en el enfoque entrada-salida.

Una función $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ es una *función continua* si:

$$\lim_{\mathbf{x} \rightarrow \mathbf{x}_0} \mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_0) \quad \mathbf{x}_0 \in \mathbb{R}^n$$

Una condición necesaria para que una función sea continua es que esté definida en todo punto $\mathbf{x} \in \mathbb{R}^2$. También se deduce que no es necesario que una función continua tenga derivada en

todo su dominio de definición. Tal es el caso, por ejemplo de la función continua $f(x) = |x|$, cuya derivada con respecto a x en el punto $x = 0 \in \mathbb{R}$ no está definida. Sin embargo, si una función tiene derivada en todo punto, entonces la función es continua.

El espacio L_2^n consiste del conjunto de todas las funciones continuas $\mathbf{f} : \mathbb{R}_+ \rightarrow \mathbb{R}^n$ tales que:

$$\int_0^\infty \mathbf{f}(t)^T \mathbf{f}(t) dt = \int_0^\infty \|\mathbf{f}(t)\|^2 dt < \infty$$

Dicho de otro modo, una función \mathbf{f} pertenece al espacio L_2^n ($\mathbf{f} \in L_2^n$) si la integral del cuadrado de su norma euclidiana está acotada superiormente.

El espacio L_∞^n consiste en el conjunto de todas las funciones continuas $\mathbf{f} : \mathbb{R}_+ \rightarrow \mathbb{R}^n$ tales que sus normas euclidianas sean acotadas, es decir,

$$\sup_{t \geq 0} \|\mathbf{f}(t)\| < \infty$$

Los símbolos L_2 y L_∞ sirven para denotar a los espacios L_2^1 y L_∞^1 respectivamente.

A continuación, se presenta un resultado obtenido en la bibliografía de análisis funcional que proporciona condiciones suficientes sobre funciones pertenecientes al espacio L_2^n para que éstas tiendan asintóticamente a cero.

Lema 3.1 *Considérese una función continua $\mathbf{f} : \mathbb{R}_+ \rightarrow \mathbb{R}^n$. Supóngase que la función \mathbf{f} satisface las siguientes hipótesis [Kelly and Santibañez, 2003]:*

- $\mathbf{f}, \dot{\mathbf{f}} = \frac{d}{dt} \mathbf{f} \in L_\infty^n$.
- $\mathbf{f} \in L_2^n$.

Entonces, necesariamente, se tiene que $\lim_{t \rightarrow \infty} \mathbf{f}(t) = \mathbf{0} \in \mathbb{R}^n$.

3.4.2. Sistemas variantes en el tiempo

Para sistemas variantes en el tiempo, usualmente es muy difícil encontrar funciones de Lyapunov con derivada definida negativa. Por lo tanto, afirmar la estabilidad asintótica para tales sistemas no es una tarea fácil. El lema de Barbalat es una herramienta muy útil que ayuda a garantizar estabilidad asintótica para sistemas no autónomos con una derivada semi-definida negativa de la función de Lyapunov.

Lema 3.2 (Barbalat) *Si la función diferenciable $f(t)$ tiene un límite finito conforme $t \rightarrow \infty$, y si \dot{f} es uniformemente continua, entonces $\dot{f} \rightarrow 0$ conforme $t \rightarrow \infty$ [Slotine and Li, 1991].*

Para aplicar el lema de Barbalat al control de sistemas dinámicos, el siguiente lemma del “tipo-Lyapunov” puede ser usado:

Lema 3.3 Si una función escalar $V(t, \mathbf{x})$ satisface las siguientes condiciones:

- $V(t, \mathbf{x})$ es acotada inferiormente
- $\dot{V}(t, \mathbf{x})$ es semidefinida negativa
- $\dot{V}(t, \mathbf{x})$ es uniformemente continua en el tiempo

entonces $\dot{V}(t, \mathbf{x}) \rightarrow 0$ conforme $t \rightarrow \infty$ [Slotine and Li, 1991].

Una condición suficiente para que una función diferenciable sea *uniformemente continua* es que su derivada sea acotada [Slotine and Li, 1991].

Control por modo deslizante

4.1. Introducción

El contenido de este capítulo es tomado de [Slotine and Li, 1991]. Así pues, en este capítulo, se considera el control de sistemas no lineales de la forma general estudiada en el capítulo anterior, pero ahora se considera que el modelo del sistema es impreciso. La imprecisión de un modelo puede provenir de incertidumbres sobre la planta (*e.g.* parámetros desconocidos de la planta), o de la elección intencional de una representación *simplificada* de la dinámica del sistema (*e.g.* como el modelado de la fricción, o los modos estructurales en un sistema mecánico razonablemente rígido). Desde un punto de vista de control, las inexactitudes de modelado se pueden clasificar en dos tipos principales:

- incertidumbres *estructuradas* (o *paramétricas*)
- incertidumbres *no estructuradas* (o *dinámicas no modeladas*)

El primer tipo corresponde a inexactitudes en los términos incluidos en el modelo, mientras que el segundo tipo corresponde a inexactitudes sobre (*i.e.* no considerar dinámicas presentes) el orden del sistema.

La falta de precisiones en el modelado puede tener fuertes efectos adversos en el control de un sistema no lineal. Por lo tanto, cualquier diseño práctico debería abordar las imprecisiones explícitamente. Dos enfoques principales y complementarios para tratar con la incertidumbre del modelo son el *control robusto* y el *control adaptable*. La estructura típica de un controlador robusto está compuesta de una parte nominal, similar a una realimentación por linealización o ley de control inversa, y de términos adicionales encaminados a lidiar con la incertidumbre del modelo. Por otra parte, la estructura de un controlador adaptable es similar, pero además, el modelo se actualiza *durante la operación*, según el desempeño medido.

Una simple aproximación al control robusto, y el principal tópico de este capítulo, es la bien conocida metodología del *control por modos deslizantes*, intuitivamente, se basa en la observación de que es mucho más fácil controlar sistemas de primer orden. (*i.e.* sistemas descritos por ecuaciones diferenciales de primer orden), sean no lineales o inciertos, que controlar sistemas generales de orden n (*i.e.*, sistemas descritos por ecuaciones diferenciales de orden n). En consecuencia, se introduce una notación simplificada, la cual, en efecto, permite que problemas de orden n puedan ser reemplazados por problemas equivalentes de primer orden. Entonces es fácil de mostrar que, para problemas transformados, en principio, se puede lograr un rendimiento “perfecto” en presencia de inexactitudes de parámetros arbitrarios. Tal desempeño, sin embargo, se obtiene al precio de una actividad de control extremadamente alta. Esto suele estar en desacuerdo con la otra fuente de modelado de incertidumbres, a saber la presencia de dinámicas no consideradas, las cuales pueden ser excitadas por la alta actividad de control. Esto lleva a una modificación de las leyes de control que, dada la actividad de control admisible, tiene como objetivo lograr una compensación efectiva entre el rendimiento del seguimiento y la incertidumbre paramétrica. Además, en algunas aplicaciones específicas, particularmente en aquellas que involucran el control de motores eléctricos, las leyes de control no modificadas pueden ser usadas directamente [Slotine and Li, 1991].

Para la clase de sistemas a los que se aplica, el diseño de controladores por modo deslizante proporciona un enfoque sistemático al problema de mantener la estabilidad y el rendimiento consistente frente a las imprecisiones de modelado. El control por modo deslizante ha sido aplicado de forma exitosa a robots manipuladores, vehículos submarinos, vehículos aéreos, transmisiones automotrices, motores eléctricos de alto desempeño y sistemas de potencia.

Los conceptos que aquí se presentan son para sistemas con una sola entrada de control, lo cual permite desarrollar intuición sobre los aspectos básicos del diseño de controladores no lineales. La sección 4.2 introduce los principales conceptos y notaciones del control por modo deslizante e ilustra el diseño de controladores básicos asociados y la sección 4.3 describe las modificaciones de las leyes de control destinadas a eliminar la excesiva actividad de control.

4.2. Superficies deslizantes

Considere un sistema dinámico en lazo abierto de una sola entrada:

$$\dot{x}^{(n)} = f(\mathbf{x}) + b(\mathbf{x})u \quad (4.1)$$

donde la variable x es la salida de interés (por ejemplo, la posición de un sistema mecánico), la variable u es la entrada de control (por ejemplo, el par de un motor), y $\mathbf{x} = [x \ \dot{x} \ \dots \ x^{(n-1)}]^\top$ es el vector de estados. En la ecuación (4.1) la función $f(\mathbf{x})$ (en general no lineal) no es exactamente conocida, pero la *extensión de la imprecisión sobre $f(\mathbf{x})$ está acotada superiormente*

por una función continua conocida de \mathbf{x} ; de manera similar, la función no lineal $b(\mathbf{x})$ no es exactamente conocida, pero su signo es conocido y está acotado por funciones conocidas de \mathbf{x} . Por ejemplo, típicamente, la inercia de un sistema mecánico, es solamente conocida con cierta precisión, y los modelos de fricción solamente describen parte de las fuerzas de fricción reales. El problema de control es hacer que el estado \mathbf{x} siga un estado especificado variante en el tiempo $\mathbf{x}_d = [x_d \ \dot{x}_d \ \dots \ x_d^{(n-1)}]^\top$ en presencia de impresiones modeladas sobre $f(\mathbf{x})$ y $b(\mathbf{x})$ ([Slotine and Li, 1991]).

Para que la tarea de seguimiento sea alcanzable usando un control finito u , el estado inicial deseado $\mathbf{x}_d(0)$ debe ser tal que

$$\mathbf{x}_d(0) = \mathbf{x}(0) \quad (4.2)$$

En un sistema de segundo orden, por ejemplo, la posición o velocidad no pueden “brincar” de manera instantánea, de modo que cualquier trayectoria deseada factible desde el tiempo $t = 0$ necesariamente comienza con la misma posición y velocidad que las de la planta; de lo contrario, el seguimiento sólo se puede lograr después de un transitorio ([Slotine and Li, 1991]).

Una notación simplificada

Sea $\tilde{x} = x - x_d$ el error de seguimiento en la variable x y sea

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d = [\tilde{x} \ \dot{\tilde{x}} \ \dots \ \tilde{x}^{(n-1)}]^\top$$

el vector de error de seguimiento. Más aún, defínase una superficie variante en el tiempo $S(t)$ en el espacio de estado \mathbb{R}^n dad por la ecuación escalar $s(\mathbf{x}; t) = 0$, donde

$$s(\mathbf{x}; t) = \left(\frac{d}{dt} + c \right)^{(n-1)} \tilde{x} \quad (4.3)$$

y c es una constante estrictamente positiva, cuya elección será interpretada posteriormente.

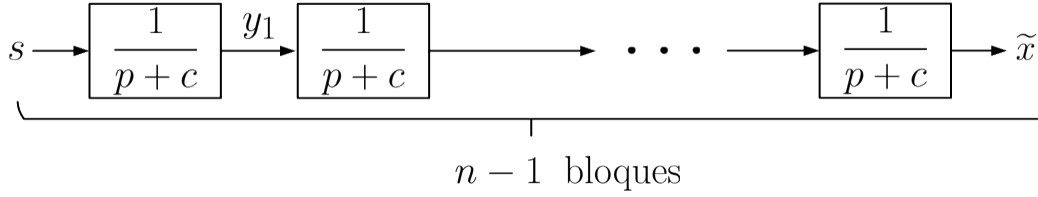
Por ejemplo, si $n = 2$,

$$s = \dot{\tilde{x}} + c\tilde{x}$$

i.e., s es simplemente una suma ponderada de el error de posición y error de velocidad: si $n = 3$,

$$s = \ddot{\tilde{x}} + 2c\dot{\tilde{x}} + c^2\tilde{x}$$

Dada la condición inicial (4.2), el problema de seguimiento $\mathbf{x} \equiv \mathbf{x}_d$ es equivalente al de permanecer sobre la superficie $S(t)$ para todo $t > 0$; ciertamente $s \equiv 0$ representa una ecuación diferencial lineal cuya única solución es $\tilde{\mathbf{x}} \equiv 0$, dadas las condiciones iniciales (4.2). Así, el problema de seguimiento de un vector n -dimensional \mathbf{x}_d se puede reducir al de mantener la cantidad escalar s en cero.


 Figura 4.1: Cálculo de cotas sobre \tilde{x}

Más precisamente, el problema de seguimiento de un vector n -dimensional \mathbf{x}_d (*i.e.*, el problema de seguimiento original de orden n en x) en efecto puede ser reemplazado por el establecimiento de un problema de *primer orden* en s . Ciertamente, ya que a partir de (4.3) la expresión de s contiene $\tilde{x}^{(n-1)}$, únicamente se necesita derivar s *una vez* para que la entrada u aparezca.

Más aún, las cotas sobre s se pueden traducir directamente a cotas en el vector de error de seguimiento $\tilde{\mathbf{x}}$, y así pues el escalar s representa una medición real del desempeño del seguimiento. Específicamente, suponiendo que $\tilde{\mathbf{x}}(0) = \mathbf{0}$ (el efecto de una condición inicial distinta de cero en $\tilde{\mathbf{x}}$ se puede agregar por separado) y teniendo que

$$\forall t \geq 0, |s(t)| \leq \Phi \quad \Rightarrow \quad \forall t \geq 0, |\tilde{x}^{(i)}(t)| \leq (2c)^i \varepsilon \quad i = 0, \dots, n-1 \quad (4.4)$$

donde $\varepsilon = \Phi/c^{n-1}$. Ciertamente, por la definición (4.3) el error de seguimiento \tilde{x} se obtiene de s a través de una secuencia de filtros pasa-bajas de primer orden (ver figura 4.1, donde $p = (d/dt)$). Sea y_1 la salida del primer filtro. Se tiene

$$y_1(t) = \int_0^t e^{-c(t-T)} s(T) dT$$

De $|s| \leq \Phi$, se obtiene

$$|y_1(t)| \leq \Phi \int_0^t e^{-c(t-T)} s(T) dT = \left(\frac{\Phi}{c}\right) (1 - e^{-ct}) \leq \frac{\Phi}{c}$$

Se puede aplicar el mismo razonamiento al segundo filtro, y así sucesivamente, todo el camino hasta $y_{n-1} = \tilde{x}$. Obteniendo entonces

$$|\tilde{x}| \leq \Phi/c^{n-1} = \varepsilon$$

De manera similar, se puede considerar que $\tilde{x}^{(i)}$ se obtiene a través de la secuencia de la figura 4.1. Del resultado anterior, se tiene $|z_i| \leq \Phi/c^{n-1-i}$, donde z_i es la salida del filtro $n-i-1$. Más aún, nótese que

$$\frac{p}{p+c} = \frac{p+c-c}{p+c} = 1 - \frac{c}{p+c}$$

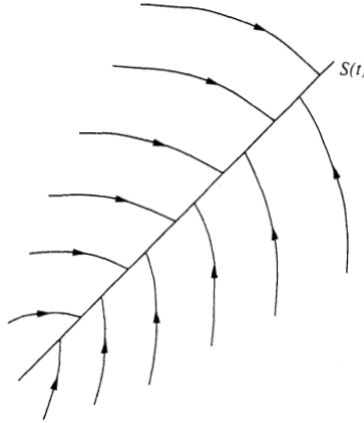


Figura 4.2: La condición deslizante

observando que la secuencia de la figura 4.1 implica que

$$|\tilde{x}^{(i)}| \leq \left(\frac{\Phi}{c^{n-1-i}} \right) \left(1 + \frac{c}{c} \right)^i = (2c)^i \varepsilon$$

i.e., las cotas en (4.4). Finalmente, en el caso de que $\tilde{\mathbf{x}}(0) \neq \mathbf{0}$, las cotas en (4.4) son obtenidas asintóticamente, *i.e.*, dentro de un corto tiempo constante $(n - 1)/c$.

Así, en efecto, se ha sustituido un problema de seguimiento de orden n por un problema de estabilización de primer orden, y se han cuantificado con (4.4) las correspondientes transformaciones de las medidas de desempeño.

El problema simplificado de primer orden de mantener el escalar s en cero ahora se puede lograr al elegir la ley de control u de (4.1) tal que fuera de $S(t)$

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s| \tag{4.5}$$

η es una constante estrictamente positiva. Esencialmente, (4.5) indica que la distancia cuadrada a la superficie, medida por s^2 , disminuye a lo largo de todas las trayectorias del sistema. Por lo tanto, limita las trayectorias para apuntar hacia la superficie $S(t)$ como se observa en la figura 4.2. En particular, una vez sobre la superficie, las trayectorias del sistema permanecen sobre la superficie. En otras palabras, satisfaciendo la condición (4.5), o *condición de deslizamiento*, hace de la superficie un *conjunto invariante*. Más aún, como ya se verá, (4.5) también implica que algunas perturbaciones o incertidumbres dinámicas pueden ser toleradas mientras se mantiene la superficie en un conjunto invariante. Gráficamente, esto corresponde al hecho de que en la figura 4.2 las trayectorias de la superficie pueden moverse sin dejar de apuntar hacia la superficie. La superficie $S(t)$ en el cumplimiento de

(4.5) se conoce como una *superficie deslizante*, y el comportamiento del sistema una vez en la superficie se denomina *régimen deslizante* o *modo deslizante*.

Otro aspecto interesante del conjunto invariante $S(t)$ es que una vez sobre el, las trayectorias del sistema están definidas por la ecuación del propio conjunto, es decir:

$$\left(\frac{d}{dt} + c\right)^{n-1} \tilde{x} = 0$$

En otras palabras, la superficie $S(t)$ es ambas cosas, un conjunto y una dinámica. Este hecho es simplemente la interpretación geométrica de la observación previa de que la definición (4.1) permite, en efecto, reemplazar un problema de orden n por uno de primer orden.

Finalmente, satisfaciendo (4.5) se garantiza que si la condición (4.2) no se verifica exactamente, *i.e.*, si $\mathbf{x}(t=0)$ está realmente fuera de $\mathbf{x}_d(t=0)$, la superficie $S(t)$ no obstante se alcanzará en un *tiempo finito* menor que $|s(t=0)|/\eta$. Ciertamente, supóngase que $s(t=0) > 0$, y sea $t_{\text{alcançe}}$ el tiempo requerido para alcanzar la superficie $s=0$. Integrando (4.5) entre $t=0$ y $t=t_{\text{alcançe}}$ lleva a

$$0 - s(t=0) = s(t=t_{\text{alcançe}}) - s(t=0) \leq -\eta(t_{\text{alcançe}} - 0)$$

lo cual implica que

$$t_{\text{alcançe}} \leq \frac{s(t=0)}{\eta}$$

Se obtendría un resultado similar iniciando con $s(t=0) < 0$.

Más aún, la definición (4.3) implica que una vez sobre la superficie, el error de seguimiento tiende exponencialmente a cero, con una constante de tiempo $(n-1)/c$ (de la secuencia de $(n-1)$ filtros de constantes de tiempo iguales a $1/c$).

El comportamiento típico del sistema que implica el cumplimiento de la condición de deslizamiento (4.5) se ilustra en la figura 4.3 para $n=2$. La superficie deslizante es una línea en el plano de fase, de pendiente $-c$ y contiene el punto (variante en el tiempo) $\mathbf{x}_d = [x_d \ \dot{x}_d]^\top$. Iniciando desde cualquier condición inicial, las trayectorias de los estados alcanzan la superficie variante en el tiempo en un tiempo menor a $|s(t=0)|/\eta$, y entonces se deslizan a lo largo de la superficie hacia \mathbf{x}_d exponencialmente con una constante de tiempo igual a $1/c$.

En resumen, la idea detrás de las ecuaciones (4.3) y (4.5) es seleccionar una función de buen comportamiento del error de seguimiento, de acuerdo con (4.3), y luego seleccionar la ley de control de retroalimentación u en (4.1) de manera que s^2 siga siendo una función del

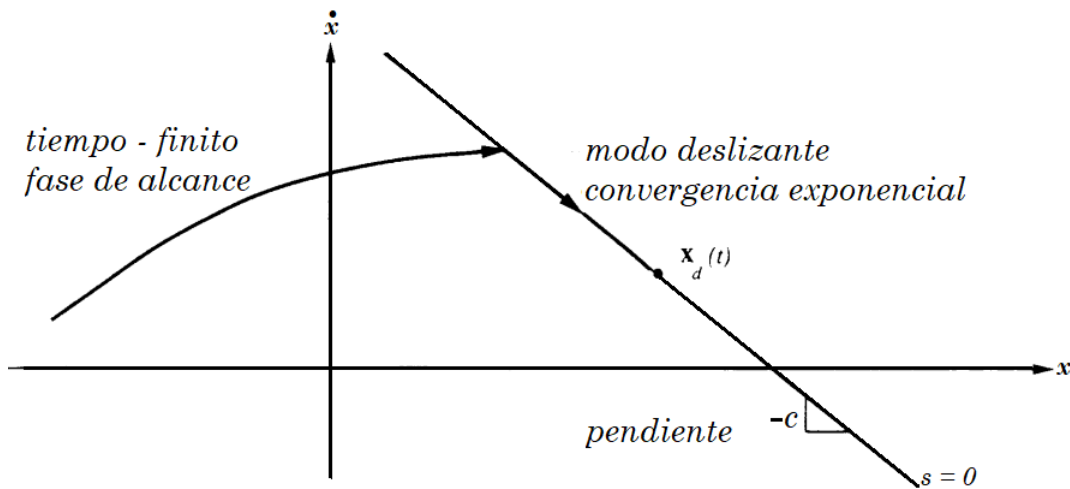


Figura 4.3: Interpretación gráfica de las ecuaciones (4.3) y (4.5) ($n = 2$)

tipo Lyapunov en el sistema de lazo cerrado, a pesar de la presencia de imprecisiones en el modelo y de perturbaciones. El procedimiento de diseño del controlador consta de dos pasos. Primero, como se ilustrará más adelante, se selecciona una ley de control de realimentación u para verificar la condición de deslizamiento (4.5); sin embargo, para tener en cuenta la presencia de imprecisiones en el modelado y de perturbaciones, la ley de control tiene que ser *discontinua* en $S(t)$. Ya que la implementación de las conmutaciones de control asociadas es necesariamente imperfecta (por ejemplo, en la práctica la conmutación no es instantánea, y el valor de s no se conoce con una precisión infinita), esto lleva a un *chattering* (figura 4.4). Ahora bien, el chattering no es deseable en la práctica, dado que involucra una alta actividad de control y puede excitar dinámicas de alta frecuencia ignoradas durante el modelado (tales como modos estructurales no modelados, retardos de tiempo despreciados, entre otras). Así pues, en un segundo paso, la *ley de control discontinua* u es *adecuadamente suavizada* para lograr una *negociación óptima entre control de ancho de banda y precisión de seguimiento*: mientras el primer paso da cuenta de incertidumbres paramétricas, el segundo paso logra robustez a dinámicas no modeladas de alto orden.

4.2.1. Construcción de Filippov de la dinámica equivalente

El movimiento del sistema en la superficie de deslizamiento puede recibir una interesante interpretación geométrica, como un “promedio” de la dinámica del sistema en ambos lados de la superficie.

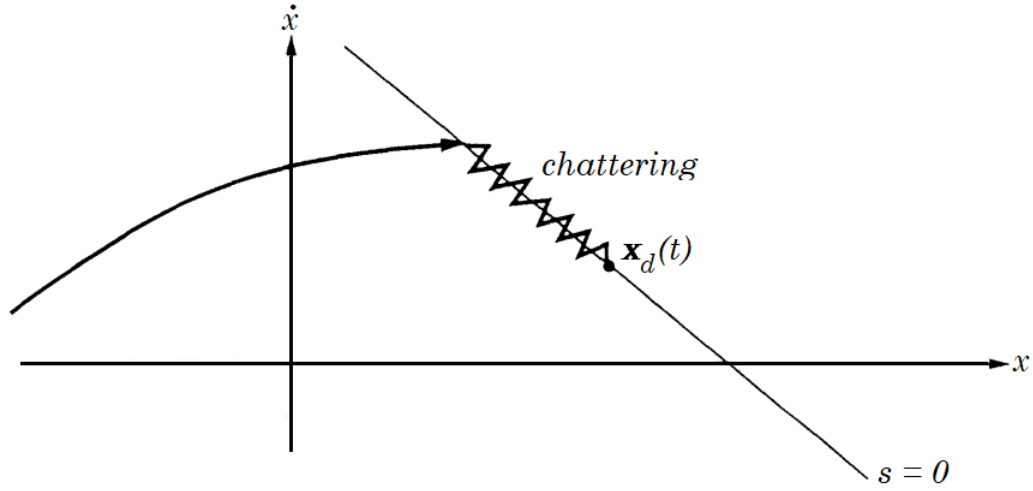


Figura 4.4: Chattering como un resultado de control imperfecto.

La dinámica en modo deslizante se puede escribir como

$$\dot{s} = 0 \tag{4.6}$$

Al resolver “formalmente” la ecuación anterior para la entrada de control, se obtiene una expresión para u llamada *ley de control equivalente*, u_{eq} , que puede interpretarse como la ley de control continuo que mantendrá a $\dot{s} = 0$ si la dinámica fuera exactamente conocida. Por ejemplo, para un sistema de la forma

$$\ddot{x} = f + u$$

se tiene

$$u_{eq} = -f + \ddot{x}_d - c\dot{\tilde{x}}$$

y la dinámica del sistema en modo deslizante es, por supuesto

$$\ddot{x} = f + u_{eq} = \ddot{x}_d - c\dot{\tilde{x}}$$

Geoméricamente, el control equivalente puede construirse como

$$u_{eq} = \alpha u_+ + (1 - \alpha)u_- \tag{4.7}$$

i.e., como una *combinación convexa* de valores de u sobre ambos lados de la superficie $S(t)$.

El valor de α puede obtenerse formalmente de (4.6) lo que corresponde a requerir que las trayectorias del sistema sean tangentes a la superficie. Esta construcción intuitiva es resumida en la figura 4.5, donde $\mathbf{f}_+ = [\dot{x} \quad f + u_+]^\top$, y de manera similar $\mathbf{f}_- = [\dot{x} \quad f + u_-]^\top$ y $\mathbf{f}_{eq} = [\dot{x} \quad f + u_{eq}]^\top$. Su justificación formal fue hecha a principios de los años 1960 por el matemático ruso A. F. Filippov.

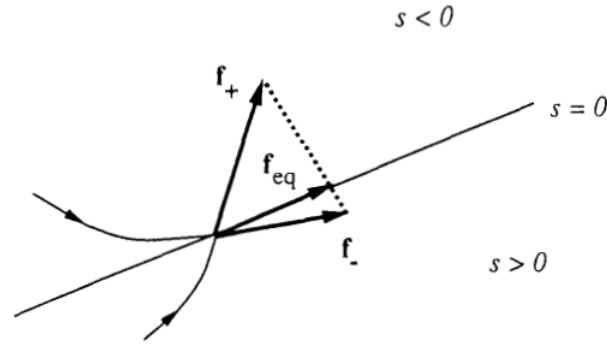


Figura 4.5: Construcción de Filippov de la dinámica equivalente en modo deslizante.

Recordando que el movimiento deslizante en la superficie corresponde a un comportamiento limitante ya que la conmutación de control ocurre infinitamente rápido, la solución formal de α de (4.6) y (4.7) se puede interpretar como el tiempo de “residencia promedio” de la trayectoria en el lado $s > 0$.

4.2.2. Perfecto desempeño a un alto precio

Dadas las cotas de la incertidumbre sobre $f(\mathbf{x})$ y $b(\mathbf{x})$, construir una ley de control para verificar la condición de deslizamiento (4.5) es sencillo, como se ilustra a continuación.

Un ejemplo básico

Considere el sistema de segundo orden

$$\ddot{x} = f + u \tag{4.8}$$

donde u es la entrada de control, x (escalar) es la salida de interés, y la dinámica f (posiblemente no lineal o variante en el tiempo) no es exactamente conocida, pero es estimada como \hat{f} . Se supone que el error de estimación de f es *acotado* por alguna función conocida $F = F(x, \dot{x})$:

$$|\hat{f} - f| \leq F \tag{4.9}$$

Por ejemplo, dado el sistema

$$\ddot{x} + a(t)\dot{x}^2 \cos 3x = u \tag{4.10}$$

donde $a(t)$ es desconocida pero se encuentra entre

$$1 \leq a(t) \leq 2$$

expresando el sistema en la forma (4.8)

$$\ddot{x} = \underbrace{-a(t)\dot{x}^2 \cos 3x}_f + u$$

se propone \hat{f} como un estimado de f dentro de los límites de $a(t)$, teniendo que

$$\hat{f} = -1.5\dot{x}^2 \cos 3x$$

mientras que de (4.9), al tomar la cota mayor de $a(t)$, se tiene

$$F = \left| \underbrace{-1.5\dot{x}^2 \cos 3x}_{\hat{f}} - \underbrace{(-2\dot{x}^2 \cos 3x)}_f \right|$$

$$F = 0.5\dot{x}^2 |\cos 3x|$$

Con el fin de tener un seguimiento del sistema $x(t) \equiv x_d(t)$, se define una superficie deslizante $s = 0$ de acuerdo a (4.3), es decir:

$$s = \left(\frac{d}{dt} + c \right) \tilde{x} = \dot{\tilde{x}} + c\tilde{x} \quad (4.11)$$

Al derivar una vez la superficie deslizante, se tiene:

$$\dot{s} = \ddot{x} - \ddot{x}_d + c\dot{\tilde{x}} = \underbrace{f + u}_{\ddot{x}} - \ddot{x}_d + c\dot{\tilde{x}} \quad (4.12)$$

La mejor aproximación \hat{u} de una ley de control continua que logrará que $\dot{s} = 0$ es entonces

$$\hat{u} = -\hat{f} + \ddot{x}_d - c\dot{\tilde{x}} \quad (4.13)$$

Nótese que \hat{u} puede ser interpretada como la mejor aproximación del control equivalente. Con el fin de satisfacer la condición (4.5) a pesar de la incertidumbre sobre la dinámica de f , se añade a \hat{u} un término *discontinuo* a través de la superficie $s = 0$:

$$u = \hat{u} - k \operatorname{sgn}(s) \quad (4.14)$$

donde la función signo es:

$$\operatorname{sgn}(s) = \begin{cases} +1 & \text{si } s > 0 \\ -1 & \text{si } s < 0 \end{cases}$$

Escogiendo $k = k(x, \dot{x})$ lo suficientemente grande en (4.14), se puede garantizar que (4.5) se satisface. Ciertamente se tiene, de (4.12)-(4.14)

$$\begin{aligned}
 \frac{1}{2} \frac{d}{dt} s^2 &= \dot{s} \cdot s \\
 &= \underbrace{[f + u - \ddot{x}_d + c\dot{x}]_s}_s \\
 &= [f + \underbrace{\hat{u} - k \operatorname{sgn}(s)}_u - \ddot{x}_d + c\dot{x}]s \\
 &= [f - \underbrace{\hat{f} + \ddot{x}_d - c\dot{x}}_{\hat{u}} - k \operatorname{sgn}(s) - \ddot{x}_d + c\dot{x}]s \\
 &= [f - \hat{f} - k \operatorname{sgn}(s)]s \\
 &= (f - \hat{f})s - k|s|
 \end{aligned}$$

de modo que, dejando

$$k = F + \eta \quad (4.15)$$

se obtiene de (4.9)

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta|s|$$

tal como se desea. Se observa de (4.15) que la discontinuidad de control k a través de la superficie $s = 0$ aumenta con el grado de incertidumbre paramétrica. También, nótese que \hat{f} y F no necesitan depender solamente de x ó \dot{x} . En general, pueden ser funciones de cualquier variable medida externa al sistema (4.8) y también pueden depender explícitamente del tiempo.

En este ejemplo básico se puede ver una de las principales ventajas de transformar el problema de control de seguimiento original en un simple problema de estabilización de *primer orden* en s . A saber, la estrategia de control de retroalimentación intuitiva “si el error es negativo, empuja lo suficientemente fuerte en la dirección positiva (y viceversa)” *en realidad funciona para sistemas de primer orden*. No para sistemas de orden superior.

MÁRGENES DE GANANCIA

Supóngase que ahora (4.8) se reemplaza por

$$\ddot{x} = f + bu \quad (4.16)$$

donde la ganancia de control b (posiblemente variante en el tiempo o dependiente de los estados) es desconocida pero con cotas conocidas (siendo posiblemente variantes en el tiempo o dependientes de los estados)

$$0 < b_{min} \leq b \leq b_{max} \quad (4.17)$$

Dado que la entrada de control entra multiplicativamente en la dinámica, es natural elegir el estimado \hat{b} de la ganancia b como la media geométrica de las cotas anteriores:

$$\hat{b} = (b_{min} b_{max})^{1/2} \quad (4.18)$$

(4.17) puede escribirse en la forma

$$\beta^{-1} \leq \frac{\hat{b}}{b} \leq \beta \quad (4.19)$$

donde

$$\beta = \left(\frac{b_{max}}{b_{min}} \right)^{1/2}$$

Dado que la ley de control será diseñada para ser robusta a la incertidumbre multiplicativa acotada (4.19), se le llamará a β el *margen de ganancia* del diseño, por la analogía usada en control lineal. Nótese que β puede ser variante en el tiempo o dependiente de los estados, y que también se tiene

$$\beta^{-1} \leq \frac{b}{\hat{b}} \leq \beta$$

También obsérvese que la incertidumbre sobre b puede venir directamente en la forma (4.19), *e.g.*, si la acción de control u es generada por un sistema dinámico lineal.

Con s y \hat{u} definidas anteriormente, la ley de control es

$$u = \hat{b}^{-1}[\hat{u} - k \operatorname{sgn}(s)] \quad (4.20)$$

con

$$k \geq \beta(F + \eta) + (\beta - 1)|\hat{u}| \quad (4.21)$$

satisfaciendo la condición deslizante. Ciertamente, usando (4.20) en la expresión de \dot{s} se tiene

$$\dot{s} = (f - b\hat{b}^{-1}\hat{f}) + (1 - b\hat{b}^{-1})(-\ddot{x}_d + \dot{c}\tilde{x}) - b\hat{b}^{-1}k \operatorname{sgn}(s)$$

por lo que k debe cumplir que

$$k \geq |\hat{b}\hat{b}^{-1}f - \hat{f} + (\hat{b}\hat{b}^{-1} - 1)(-\ddot{x}_d + \dot{c}\tilde{x})| + \eta\hat{b}\hat{b}^{-1} \quad (4.22)$$

Dado que $f = \hat{f} + (f - \hat{f})$, donde $|f - \hat{f}| \leq F$, esto a su vez conduce a

$$k \geq \hat{b}\hat{b}^{-1}F + \eta\hat{b}\hat{b}^{-1} + |\hat{b}\hat{b}^{-1} - 1| \cdot |\hat{f} - \ddot{x}_d + \dot{c}\tilde{x}|$$

y así a (4.21). Téngase en cuenta que la discontinuidad del control se ha incrementado para tener en cuenta la incertidumbre sobre la ganancia de control b .

Ejemplo: Un modelo simplificado del movimiento de un vehículo subacuático puede ser escrito como

$$m\ddot{x} + c\dot{x}|x| = u \quad (4.23)$$

donde x define la posición, u es la entrada de control (la fuerza proporcionada por la hélice), m es la masa del vehículo (incluyendo la llamada masa agregada, asociada con el movimiento en un fluido) y c es un coeficiente de arrastre. En la práctica, m y c no son conocidos con precisión, porque solo describen los complejos efectos hidrodinámicos que gobiernan el movimiento del vehículo.

Definiendo s como $s = \dot{\tilde{x}} + c\tilde{x}$, calculando \dot{s} explícitamente, y procediendo como se hizo anteriormente, puede obtenerse una ley de control que satisface la condición de deslizamiento, resultando en

$$u = \hat{m}(\ddot{x}_d - \dot{c}\tilde{x}) + \hat{c}\dot{x}|x| - k \operatorname{sgn}(s) \quad (4.24)$$

con

$$k = (F + \beta\eta) + \hat{m}(\beta - 1)|\ddot{x}_d - \dot{c}\tilde{x}| \quad (4.25)$$

Se observa que la expresión (4.25) es más estricta que la forma general (4.21), reflejando la estructura más simple de la incertidumbre paramétrica; intuitivamente, u puede compensar $c\dot{x}|x|$ directamente, a pesar de la incertidumbre sobre m . En general, para un problema dado, es una buena idea reobtener rápidamente una ley de control que satisfaga la condición de deslizamiento, en lugar de aplicar una fórmula pre-establecida.

□

Es útil hacer una pausa en este punto y preguntarse si una acción de control diferente obtenida por algún otro método, podría lograr la misma tarea. La respuesta a esta interrogante es que, dada una trayectoria deseada factible, hay una función *única* de control de *tiempo* suave que la sigue exactamente, es decir:

$$u(t) = \frac{1}{b(\mathbf{x}_d)}[\ddot{x}_d - f(\mathbf{x}_d)] \quad (4.26)$$

Por lo tanto, cualquiera que sea el método, la función de tiempo de control resultante será la misma, y, por lo tanto, el uso de este enfoque en particular simplemente proporciona una forma sencilla de modificar esa función de tiempo. Dado que se requiere un seguimiento perfecto incluso en presencia de incertidumbres paramétricas, esta función de tiempo se construye a través de un proceso *promedio* de interrupciones discontinuas infinitamente rápidas, en lo que se conoce como el control equivalente, que es precisamente (4.26).

Las leyes de control que satisfacen la condición de deslizamiento (4.5) y conducen a un seguimiento “perfecto” ante la incertidumbre del modelo, son discontinuas a lo largo de la superficie $S(t)$, llevando así en la práctica el control al chattering. En general, el chattering

es altamente indeseable, ya que implica una actividad de control extremadamente alta, y más aún puede excitar dinámicas de alta frecuencias ignoradas durante el modelado. Es posible mostrar cómo modificar las leyes de control de conmutación derivadas anteriormente para eliminar el chattering.

En aplicaciones específicas, sin embargo, el control con chattering es aceptable, y las leyes de control de conmutación obtenidas anteriormente pueden producir un rendimiento extremadamente alto.

4.2.3. Aplicaciones de las leyes de control de conmutación discontinuas

Las principales aplicaciones de los controladores de conmutación anteriores incluyen el control de los motores eléctricos y el uso de compensación artificial para reducir los efectos de la adhesión (fricción).

Control de conmutación en lugar de modulación de ancho de pulso

En los motores eléctricos modulados por ancho de pulso (PWM), la entrada de control u es una tensión eléctrica en lugar de una fuerza mecánica o aceleración. El control por chattering puede entonces ser aceptable siempre que esté fuera del rango de frecuencia de la dinámica relevante no modelada. Siempre que los cálculos necesarios (incluyendo la ley de control y estimación de estados) se puedan manejar en línea a una velocidad suficientemente alta, o sean implementados mediante circuitos analógicos, el control por modo deslizante puro mediante leyes de control conmutado puede ser una opción viable y de alto rendimiento.

Control de conmutación con observador lineal

La dificultad para obtener mediciones de estado significativas a una tasa de muestreo muy alta se puede cambiar utilizando observadores de estado. Para sistemas *lineales*, el diseño de tales observadores es bien conocido y sistemático. El principio del enfoque para diseñar un controlador de conmutación utilizando un observador es muy simple. En lugar de seguir la superficie $s = 0$, el sistema está diseñado para seguir la superficie $s_e = 0$, donde s_e se obtiene al reemplazar el estado \mathbf{x} por su estimado \mathbf{x}_e en la expresión de s . Esto puede lograrse calculando un término de compensación dinámico \hat{u}_e basado en las estimaciones de estado disponibles, y empleando términos de conmutación de la forma $-k(\mathbf{x}_e) \text{sgn}(s_e)$ donde $k\mathbf{x}_e$ se elige lo suficientemente grande para compensar tanto incertidumbres paramétricas como inexactitudes del observador. Esto produce que $s_e \rightarrow 0$ (conforme $t \rightarrow \infty$). Entonces, si el observador ha sido diseñado adecuadamente para que converja a pesar de las incertidumbres

en el modelado (lo que, de nuevo, es fácil de conseguir en el caso lineal), se tiene también que $s \rightarrow s_e$. Por lo tanto, $s \rightarrow 0$, y el estado actual converge hacia el estado deseado. Más aún, el modo deslizante y sus propiedades robustas son mantenidos sobre la superficie $s_e \equiv 0$, las cuales tienden hacia la superficie deslizante deseada conforme el observador converge.

Control de conmutación en lugar de señales de alta frecuencia

Cuando la incertidumbre consiste en efectos pequeños en magnitud pero difíciles de modelar, tales como el amarre del actuador, la conmutación (switcheo) en s se puede usar ventajosamente en lugar de una señal de interpolación más estándar. Idealmente, la frecuencia de conmutación debe ser elegida mucho *más allá* de los modos de vibración estructural significativos (en sistemas mecánicos), manteniéndose por *debajo* del ancho de banda de los actuadores. Esto supone nuevamente que se pueden proporcionar estimaciones de estado significativas a la frecuencia de conmutación seleccionada. Este enfoque puede mejorar particularmente la calidad del comportamiento a baja velocidad, de lo contrario es extremadamente sensible a la fricción.

Los ejemplos anteriores representan las pocas aplicaciones específicas en las que se puede tolerar y explotar el chattering. Sin embargo, en el caso general, la pregunta es cómo obtener leyes de control que mantengan el sistema cerca de la superficie $s = 0$, evitando al mismo tiempo las vibraciones. Esto se tratará a continuación.

4.3. Aproximaciones continuas de la ley de control de conmutación

En general, se debe eliminar el chattering para que el controlador funcione correctamente. Esto se puede lograr suavizando la discontinuidad del control en una *capa límite* delgada adyacente a la superficie de conmutación

$$B(t) = \{\mathbf{x}, |s(\mathbf{x}, t)| \leq \Phi\} \quad \Phi > 0 \quad (4.27)$$

donde Φ es el *espesor* de la capa límite y $\varepsilon = \Phi/c^{n-1}$ es el *ancho* de la capa límite, como lo ilustra la figura 4.6 para $n = 2$. En otras palabras, fuera de $B(t)$ se elige una ley de control u como se hizo anteriormente (*i.e.*, satisfaciendo la condición (4.5)), lo cual garantiza que la capa límite es atractiva, por lo tanto invariante: todas las trayectorias que inicien en $B(t = 0)$ permanecerán dentro de $B(t)$ para todo $t \geq 0$; y entonces se interpola u dentro de $B(t)$ –por ejemplo, reemplazando en la expresión de u el término $\text{sgn}(s)$ por s/Φ , dentro de $B(t)$, como se ilustra en la figura 4.7.

Dados los resultados anteriormente vistos, esto conduce al *seguimiento dentro de una precisión garantizada* ε (más que un seguimiento “perfecto”) y de manera más general garantiza que

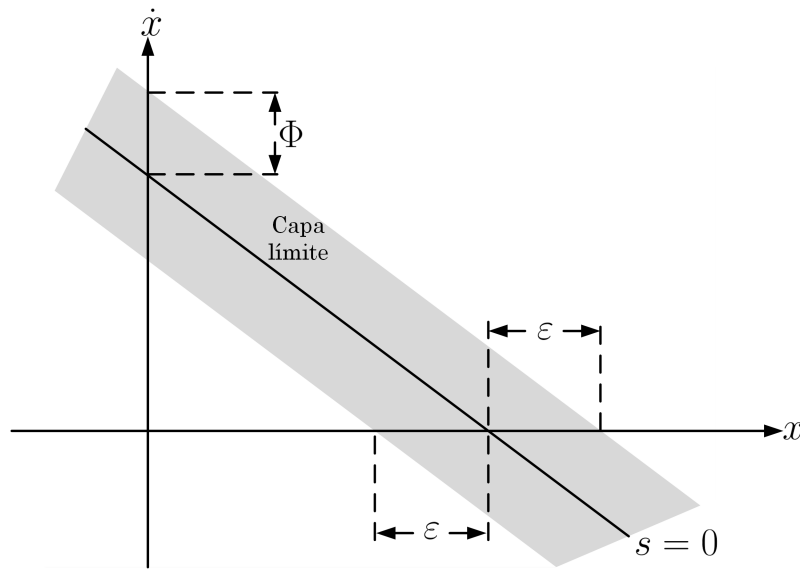


Figura 4.6: La capa límite

para todas las trayectorias que comienzan dentro de $B(t=0)$

$$\forall t \geq 0, |\tilde{x}^i(t)| \leq (2c)^i \epsilon \quad i = 0, \dots, n-1$$

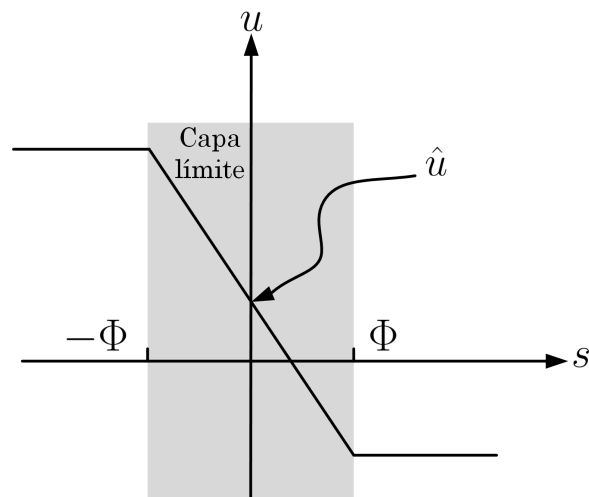


Figura 4.7: Control de interpolación en la capa límite

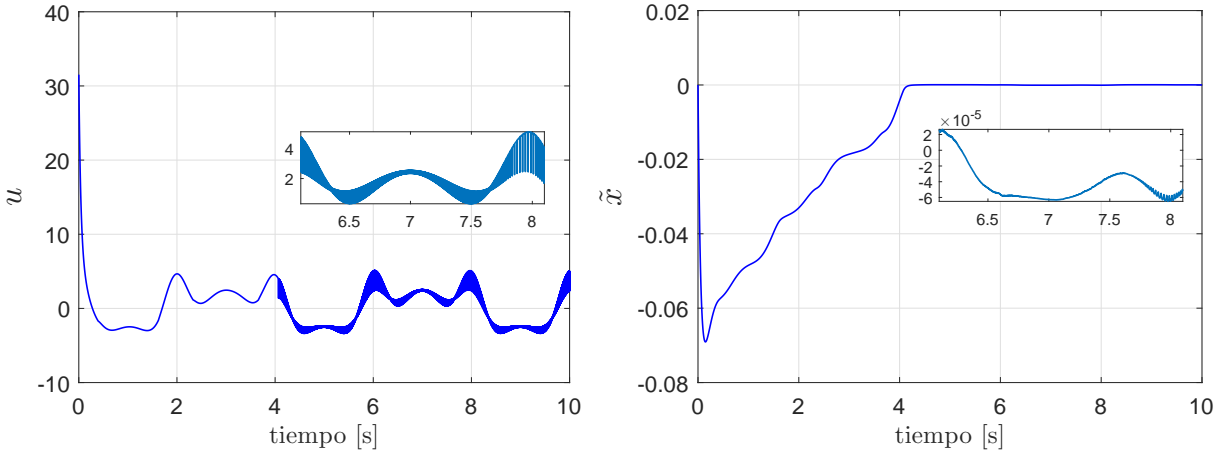


Figura 4.8: Entrada de control conmutada y resultado del desempeño del error de seguimiento

Ejemplo: Considere el sistema descrito en (4.10)

$$\ddot{x} = -a(t)\dot{x}^2 \cos 3x + u$$

El valor de $a(t)$ es $a(t) = |\sin t| + 1$, (el cual concuerda con la cota supuesta sobre $a(t)$) y considérese que la trayectoria deseada es $x_d = \sin(\frac{\pi}{2}t)$.

El error de seguimiento se define como: $\tilde{x} = x - x_d$, mientras la ley de control discontinua de acuerdo a (4.13) y (4.14) es

$$u = \underbrace{-\hat{f} + \ddot{x}_d - c\tilde{x}}_{\hat{u}} - k \operatorname{sgn}(s)$$

$$u = 1.5\dot{x}^2 \cos 3x + \ddot{x}_d - c\dot{\tilde{x}}_d - \underbrace{(F + \eta)}_k \operatorname{sgn}(s)$$

$$u = 1.5\dot{x}^2 \cos 3x + \ddot{x}_d - c\dot{\tilde{x}}_d - (0.5\dot{x}^2 |\cos 3x| + \eta) \operatorname{sgn}(\tilde{x} + c\tilde{x})$$

En la figura 4.8 se muestra el desempeño del error de seguimiento y de la ley de control utilizando una ley de control conmutada (con $c = 20$, $\eta = 0.1$) a una frecuencia de muestreo de 1 kHz.

Se observa que el desempeño del seguimiento es excelente, pero se obtiene a un precio de control de alto chattering.

Supóngase ahora que se interpola la entrada de control anterior en una capa límite delgada

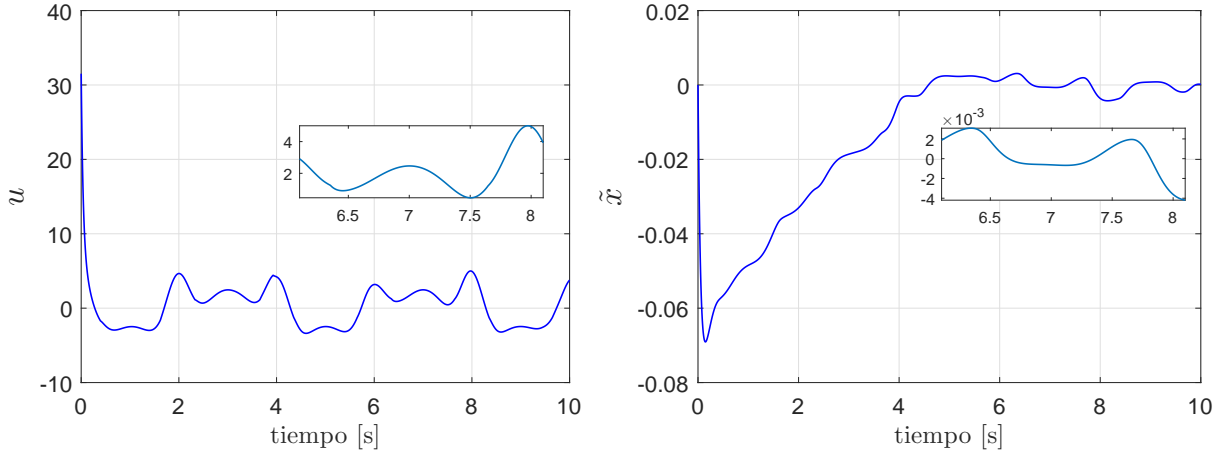


Figura 4.9: Entrada de control suave y desempeño del error de seguimiento resultante

de espesor 0.1

$$u = \underbrace{-\hat{f} + \ddot{x}_d - c\tilde{x}}_{\hat{u}} - k \text{sat}(s/\Phi)$$

$$u = 1.5\dot{x}^2 \cos 3x + \ddot{x}_d - \dot{c}\tilde{x}_d - (0.5\dot{x}^2 |\cos 3x| + \eta) \text{sat}\left(\frac{\tilde{x} + c\tilde{x}}{\Phi}\right)$$

Como se ilustra en la figura 4.9, el desempeño del seguimiento, si bien no es tan perfecto como el anterior, sigue siendo muy bueno y ahora se logra usando una ley de control uniforme (sin chattering). Nótese que las cotas del error de seguimiento son consistentes con (4.27).

□

4.4. Comentarios

- (i) En este capítulo se introdujo la teoría fundamental sobre el control por modo deslizante para sistemas no lineales, el cual es una alternativa de solución del control robusto al problema de tratar con sistemas que presenten incertidumbres paramétricas y dinámicas no modeladas, manteniendo su estabilidad y buen desempeño.
- (ii) La trayectoria deseada \mathbf{x}_d debe elegirse lo suficientemente suave como para no excitar la dinámica no modelada de alta frecuencia.
- (iii) Un argumento similar al de la discusión anterior muestra que la elección de la dinámica (4.3) utilizada para definir superficies deslizantes es la “mejor condicionada” entre

las dinámicas lineales, en el sentido de que garantiza el mejor rendimiento de seguimiento dado el ancho de banda de control deseado y el grado de incertidumbre de los parámetros.

- (iv) Se ha demostrado que es posible eliminar el chattering suavizando la discontinuidad del control en una capa límite delgada adyacente a la superficie de conmutación.

Control por modos deslizantes basado en aproximación difuso adaptable

En este capítulo se presenta un resumen de los sistemas de control basados en lógica difusa, así también, se mencionan las propiedades de los sistemas difusos como aproximadores universales. Finalmente, aprovechando estas propiedades, se combinan con el control por modos deslizantes para formular una estrategia de control que permita estabilizar a un sistema no lineal subactuado conocido como *ballbot*.

5.1. Lógica difusa

El concepto de lógica difusa fue introducido por primera vez en 1965 por Lotfi A. Zadeh como una alternativa para describir conjuntos que no pueden definirse con facilidad, debido a que existe vaguedad o incertidumbre en ellos, ni tampoco es posible definir el grado de pertenencia de ciertos elementos a esos conjuntos [Zadeh, 1965].

5.1.1. Conjunto difuso

Por definición un conjunto consiste en una colección de objetos que pueden ser tratados como una sola entidad y cumplen con ciertas características y condiciones.

En los conjuntos clásicos se determina si un elemento pertenece o no a un conjunto. Sea U el universo de discurso, es decir, todos los posibles elementos o valores en un determinado contexto; sea A un conjunto dado; los elementos x que pertenecen al conjunto A se expresan como

$$x \in A$$

mientras que los elementos que no pertenecen al conjunto se expresan como

$$x \notin A$$

Otra forma de definir un conjunto es mediante una función de pertenencia $\mu_A(x)$, también llamada de membresía, la cual asigna un valor de pertenencia en A para cada elemento x en el intervalo $[0, 1]$. En un conjunto clásico la función de pertenencia solo puede tomar los valores $\{0, 1\}$ y puede caracterizarse como

$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

A diferencia de los conjuntos clásicos, en un conjunto difuso la función de pertenencia puede tomar cualquier valor en el intervalo $[0, 1]$, y este valor es conocido también como grado de pertenencia. Entonces, un conjunto difuso A en U se define como el conjunto de pares ordenados, un elemento cualquiera x y su correspondiente valor de pertenencia como

$$A = \{(x, \mu_A(x)) | x \in U\}$$

También puede decirse que un conjunto difuso A es la gráfica de su función de pertenencia $\mu_A(x)$.

5.1.2. Funciones de pertenencia

Como ya se mencionó, una función de pertenencia denotada por $\mu_A(x)$ relaciona para cada x un grado de pertenencia al conjunto A en el intervalo $[0, 1]$. Existen diversas funciones de pertenencia utilizadas en sistemas difusos, pero las más comunes son: impulsivas (en inglés singleton), triangulares, pseudo-trapezoidales y gaussianas ([T. Nguyen et al., 2002]). Su utilización dependerá de las nociones sobre el sistema que se pretende describir y de su aplicación particular (ver [Jantzen, 2007]).

5.2. Control basado en lógica difusa

Los sistemas de control basados en lógica difusa constituyen una de las principales aplicaciones de la lógica difusa. Los controladores difusos se construyen a partir de un conjunto de reglas lingüísticas basadas en el conocimiento previo del sistema de control y en la experiencia de operadores expertos. Una regla lingüística tiene la forma

$$\text{SI } x \text{ es } A \text{ ENTONCES } y \text{ es } B$$

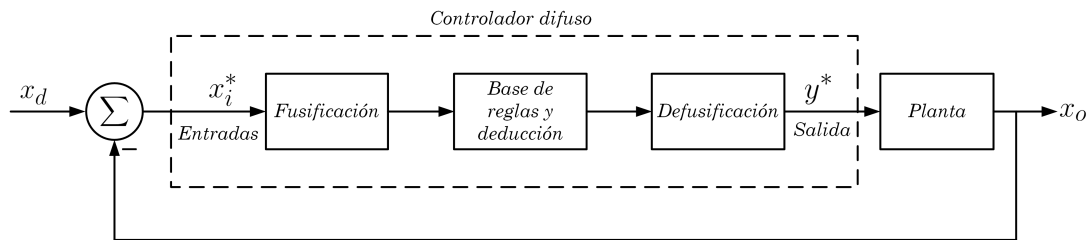


Figura 5.1: Estructura general de un controlador difuso

Por ello se dice que el control difuso es un control con reglas. A la primera parte de las reglas lingüísticas se le conoce como *antecedente*, mientras que la segunda parte de las reglas se denomina *consecuente*, conclusión o acción. Los controladores basados en lógica difusa permiten resolver complicados problemas de control en situaciones donde el modelado de la planta es muy complejo o incompleto, a partir de heurística o conocimientos previos, en lugar de controladores basados en ecuaciones diferenciales del modelado dinámico de la planta.

5.2.1. Características

Los sistemas de control basados en lógica difusa (figura 5.1) poseen las siguientes características:

- Ofrecen una salida de manipulación de acuerdo a las variables de entrada, que pueden ser variables de la planta o errores del sistema en lazo cerrado.
- Son tolerantes a la imprecisión de datos y parámetros no modelados.
- Pueden emplearse para modelar sistemas no lineales; y algunas veces la solución de control para sistemas no lineales es mucho más sencilla que otras estrategias al no requerir ecuaciones diferenciales.
- La calibración de los controladores es sencilla, ya que se realiza a partir de un lenguaje similar al de los humanos.
- Se pueden combinar con otras técnicas de control clásicas o no lineales, de manera que pueden coexistir en un mismo proceso.
- Una de sus desventajas es que requieren mayor tiempo de cálculo, por lo que debe tomarse en cuenta la capacidad y velocidad de cómputo de los sistemas en que se implementan.
- Por lo general, los controladores basados en lógica difusa poseen numerosos parámetros de configuración, por lo que la sintonización puede complicarse en algunos casos.

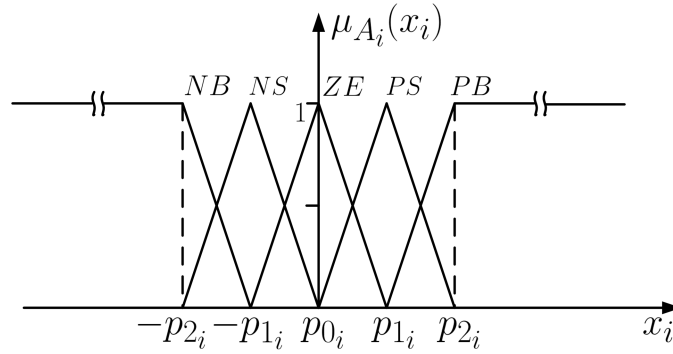


Figura 5.2: Conjuntos difusos en el bloque de fusificación

5.2.2. Estructura

Un controlador basado en lógica difusa se compone de tres módulos principales:

- **Módulo de fusificación.** Las entradas al controlador son convertidas a un grado de pertenencia para cada conjunto difuso de entrada.
- **Base de reglas y módulo de deducción.** Los valores obtenidos se relacionan y deducen un grado de pertenencia en los conjuntos difusos de salida.
- **Módulo de defusificación.** Se calcula un solo valor de salida, que se convierte en la salida del controlador.

Módulo de fusificación

El módulo de fusificación recibe las n variables de entrada x_i ($i = 1, 2, \dots, n$) y les asigna a cada uno de los N_i conjuntos difusos $A_i^{j_i}$, un valor de pertenencia $\mu_{A_i^{j_i}}(x_i)$ siendo j_i el número que denota a cada conjunto difuso de la entrada i . Generalmente se utiliza un número *impar* de N_i conjuntos difusos, distribuidos simétricamente alrededor del origen del *universo de discurso* (dominio de la variable de entrada), por lo que: $j_i = -\frac{N_i-1}{2}, \dots, -1, 0, 1, \dots, \frac{N_i-1}{2}$. En la figura 5.2 se muestra un ejemplo de la distribución de los conjuntos difusos para una entrada dada. A cada conjunto difuso se le asigna una variable lingüística; por ejemplo, el conjunto A_i^{-2} , es el conjunto NB (negativo grande) para la entrada x_i , y al mismo tiempo es una función de pertenencia de tipo pseudo-trapezoidal izquierda. La fusificación de las n entradas x_i produce $N_1 \cdot N_2 \cdot \dots \cdot N_n$ valores de pertenencia, uno por cada conjunto difuso o función de pertenencia. En la mayoría de los casos se emplea el fusificador *impulsivo*, que equivale a evaluar directamente la entrada en las funciones de pertenencia; sin embargo pueden usarse fusificadores con funciones de pertenencia triangulares o gaussianos, aunque esto complica los cálculos de los valores de pertenencia.

Base de reglas y módulo de deducción

Este módulo tiene dos etapas. La primera, la base de reglas, constituye la relación de los valores de pertenencia de las entradas basándose en el conocimiento y experiencia previos, asignando a la salida un conjunto difuso o función de pertenencia de salida. La estructura de la regla es

Regla (j_1, \dots, j_n) : SI x_1 es $A_1^{j_1}$ Y ... Y x_n es $A_n^{j_n}$ ENTONCES y es $B^{j_1 \dots j_n}$

El operador Y (AND o mínimo) es el más utilizado como conector de premisas. En forma lingüística una regla podría ser:

Regla (j_1, j_2) : SI x_1 es NS Y x_2 es ZE ENTONCES y es PS

El módulo de deducción tiene como función inferir el valor de pertenencia para cada función de pertenencia de salida, de acuerdo con la base de reglas. Existen varias posibilidades para inferir este valor, como son la inferencia max-min, que utiliza un operador AND entre las premisas de la regla, y la inferencia max-prod que utiliza el operador producto entre las premisas de las reglas y que es más utilizado por ser más rápido su cálculo. El valor que produce la operación entre premisas es el valor que corta (en inferencia max-min) o escala (en inferencia max-prod) a la función de pertenencia de salida de cada regla. Los valores de pertenencia de las funciones de pertenencia de salida que se han producido son evaluados con el operador máximo o la unión (OR) y el resultado se conoce como *conclusión* y representa la salida del módulo de deducción.

Módulo de defusificación

El módulo de defusificación tiene como tarea producir un valor escalar y a partir de la conclusión de las funciones de pertenencia de salida que entrega el módulo de deducción. Este valor corresponde al centro de gravedad de la curva generada por las funciones de pertenencia de salida. Existen varias alternativas para obtener el valor de y que corresponde a la salida de todo el sistema de control difuso; entre ellas se encuentran el defusificador centro de gravedad, centro promedio y máximo. El que ofrece el valor más exacto es el centro de gravedad, pero su cálculo es muy costoso; el defusificador máximo es el más rápido pero no posee gran exactitud. El defusificador centro promedio realiza los cálculos de integración de forma discreta por lo que no consume muchos cálculos y se aproxima muy bien al centro de gravedad, por lo que se constituye como la mejor alternativa. Este último puede calcularse mediante la siguiente expresión:

$$y^* = \frac{\sum_{i=1}^M \bar{y}^i w_i}{\sum_{i=1}^M w_i} \quad (5.1)$$

donde y^* es el valor real (crisp) de la salida, M es el número de conjuntos difusos de salida, \bar{y}^i representa el centro de una función de pertenencia de salida y w_i es su valor máximo (altura). La figura 5.3 ilustra gráficamente la operación para un simple ejemplo con $M = 2$.

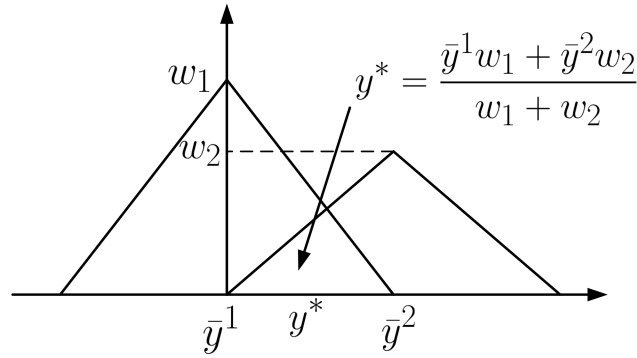


Figura 5.3: Representación gráfica de un defusificador centro promedio.

5.2.3. Aproximador universal

Una propiedad importante de los sistemas basados en lógica difusa es que estos son aproximadores universales. Esto significa que los sistemas difusos tienen la propiedad de aproximar funciones no lineales con una precisión arbitraria.

En este trabajo se combinan sistemas difusos con fusificador impulsivo, defusificador centro promedio e inferencia max-prod. Este tipo de sistema difuso puede escribirse en forma compacta con la siguiente fórmula no lineal

$$y^* = \frac{\sum_{l=1}^M \bar{y}^l (\prod_{i=1}^n \mu_{A_i^l}(x_i^*))}{\sum_{l=1}^M (\prod_{i=1}^n \mu_{A_i^l}(x_i^*))} \quad (5.2)$$

Esta fórmula compacta facilita el cálculo de la salida del sistema difuso, ya sea para aproximar una función o como controlador. Los siguientes teoremas son obtenidos de [Wang, 1997], los cuales son de particular interés desde el punto de vista de aproximación de funciones.

Teorema 5.1 *Supóngase U como el universo de discurso y un conjunto compacto en \mathbb{R}^n . Entonces, para una función continua real $g(\mathbf{x})$ en U y un $\epsilon > 0$ arbitrario, existe un sistema difuso $f(\mathbf{x})$ en la forma (5.2) tal que*

$$\sup_{\mathbf{x} \in U} |f(\mathbf{x}) - g(\mathbf{x})| < \epsilon \quad (5.3)$$

△

Teorema 5.2 *Para una función continua real $g(\mathbf{x})$ en U y un sistema difuso $f(\mathbf{x})$ en la forma (5.2) se tiene que*

$$\|g(\mathbf{x}) - f(\mathbf{x})\|_\infty \leq \left\| \frac{\partial g(\mathbf{x})}{\partial x_1} \right\|_\infty h_1 + \left\| \frac{\partial g(\mathbf{x})}{\partial x_2} \right\|_\infty h_2 + \dots + \left\| \frac{\partial g(\mathbf{x})}{\partial x_n} \right\|_\infty h_n \quad (5.4)$$

$\|d(x)\|_\infty = \sup_{x \in U} |d(x)|$ es la norma infinito, $h_i = \max_{1 \leq N_i - 1} |c_i^{j+1} - c_i^j|$, con c_i^k como el centro de la función de pertenencia $k = j$ o $k = j + 1$ (según sea el caso) de la entrada i , donde $i = 1, 2, \dots, n$, y n es el número de entradas al sistema difuso.

△

El teorema 5.1 indica que es posible encontrar un sistema difuso de la forma (5.2) que puede aproximarse a una función matemática con un determinado error. De acuerdo al teorema 5.2, este error está acotado y depende de las variables de entrada al sistema difuso con las que se realiza la aproximación de la distancia entre las funciones de pertenencia de entrada, en el módulo de fusificación. Esto significa que, a mayor número de funciones de pertenencia el error de aproximación será menor, aunque esto generará un mayor número de reglas.

Por lo anterior, es posible utilizar un sistema difuso y sus características para aproximar funciones no lineales. La construcción de un sistema de control basado en lógica difusa y un aproximador de funciones es similar, por lo que las consideraciones hechas en este capítulo son válidas para el aproximador.

5.3. Control difuso adaptable

Una de las características del control basado en lógica difusa es la de poder trabajar con plantas que tienen incertidumbre, parámetros desconocidos o variantes, ya que el controlador difuso no requiere de un amplio modelado de la planta sino información heurística sobre su comportamiento. Por otra parte, como es sabido, el controlador adaptable está definido como un sistema de control con parámetros ajustables y un mecanismo para ajustar estos parámetros.

En los últimos años se han incorporado a los controladores adaptables las técnicas de control inteligente, dando lugar a controladores como los neuroadaptables, difuso adaptables y neurodifuso adaptables. En este trabajo de tesis, se estudia el controlador difuso adaptable propuesto en [Wang, 1997], el cual es combinado con una estrategia basada en modos deslizantes reportada en [Azimi et al., 2014]. La figura 5.4 ilustra la configuración general de un sistema de control difuso adaptable.

El controlador difuso adaptable es un controlador híbrido, donde el mecanismo de adaptación proporciona parámetros de ajuste al controlador difuso, los cuales corresponden específicamente a los centros de las funciones de pertenencia de salida.

Los controladores difusos adaptables pueden clasificarse en dos tipos:

- **Directo.** El controlador se construye a partir del conocimiento de la acción control, es decir, cuando se conoce qué acción de control requiere la planta para ciertas condiciones; por ejemplo: “si la temperatura es baja, entonces aumentar el flujo de combustible a la caldera”.
- **Indirecto.** El controlador se construye a partir del conocimiento de la planta, es decir, cuando por experiencia se sabe cómo afecta la señal de control a la planta; por ejemplo: “si aumenta el flujo de combustible a la caldera, entonces la temperatura aumentará”.

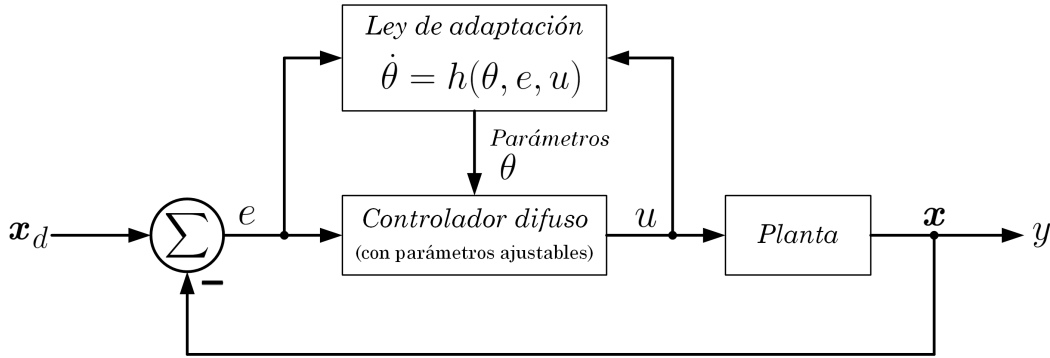


Figura 5.4: Configuración general de un sistema difuso adaptable

- **Combinación directo/indirecto.** El controlador se construye a partir del conocimiento tanto de la acción de control como de la planta.

El controlador difuso adaptable que se implementa en esta tesis es un controlador de tipo indirecto.

5.4. Control por modos deslizantes desacoplado

Considérese una clase de sistema no lineal subactuado, con incertidumbres paramétricas y perturbaciones externas descrito por:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) + \mathbf{f}(\dot{\mathbf{q}}) = \boldsymbol{\tau}_{\text{ext}} \quad (5.5)$$

donde

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad \mathbf{f}(\dot{\mathbf{q}}) = \begin{bmatrix} f_1(\dot{q}_1) \\ f_2(\dot{q}_2) \end{bmatrix}, \quad \boldsymbol{\tau}_{\text{ext}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

son el vector de coordenadas generalizadas, vector de fuerzas de fricción y vector de fuerzas externas (con la señal de control), respectivamente; mientras

$$M(\mathbf{q}) = \begin{bmatrix} m_{11}(\mathbf{q}) & m_{12}(\mathbf{q}) \\ m_{21}(\mathbf{q}) & m_{22}(\mathbf{q}) \end{bmatrix}, \quad C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} c_{11}(\mathbf{q}, \dot{\mathbf{q}}) & c_{12}(\mathbf{q}, \dot{\mathbf{q}}) \\ c_{21}(\mathbf{q}, \dot{\mathbf{q}}) & c_{22}(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix}, \quad g(\mathbf{q}) = \begin{bmatrix} g_1(q_1) \\ g_2(q_2) \end{bmatrix}$$

son la matriz de inercia, matriz de coriolis y fuerzas centrífugas y el vector de fuerzas de gravedad, respectivamente. Sean las variables de estado $x_1 = q_1$, $x_2 = \dot{q}_1$, $x_3 = q_2$ y $x_4 = \dot{q}_2$.

El sistema puede escribirse en presencia de incertidumbres y perturbaciones externas como:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_1(\mathbf{x}) + g_1(\mathbf{x})u + d_1(t) \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= f_2(\mathbf{x}) + g_2(\mathbf{x})u + d_2(t) \end{aligned} \tag{5.6}$$

donde $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^\top$ es el vector de estados, $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, $g_1(\mathbf{x})$ y $g_2(\mathbf{x})$ son funciones no lineales desconocidas y $d_1(t)$ y $d_2(t)$ son perturbaciones externas acotadas, i.e. $|d_1(t)| \leq D_1(t)$, $|d_2(t)| \leq D_2(t)$. Como puede verse de (5.6), la misma entrada de control u aparece en las dos dinámicas, así que el sistema completo es un sistema subactuado, [Huang et al., 2015]. Sea el vector de salida $\mathbf{y} = [x_1 \ x_3]^\top$. Por lo tanto, se puede considerar el sistema (5.5) como un sistema subactuado de una sola entrada y múltiples salidas¹. Así pues, el diseño de un controlador por modos deslizantes para el sistema (5.6) no es directo. Más aún, los grados de libertad actuados y no actuados están acoplados, y esto debe ser tomado en cuenta en el diseño del controlador. El problema de control es diseñar un controlador por modos deslizantes empleando una aproximación difuso adaptable para el sistema subactuado (5.6) tal que todas las señales del sistema en lazo cerrado permanezcan acotadas y la estabilidad pueda garantizarse.

5.4.1. Diseño del controlador por modos deslizantes

La estrategia de control que se presenta en este apartado ha sido probada con éxito en simulación en los bien conocidos sistemas carro-péndulo y TORA, [Azimi et al., 2014]. Con el fin de diseñar un controlador por modos deslizantes con aproximación difuso adaptable, se definen dos superficies deslizantes de la siguiente forma:

$$s_1 = c_1 x_1 + x_2 \tag{5.7}$$

$$s_2 = c_2 x_3 + x_4 \tag{5.8}$$

con $c_1, c_2 > 0$. Considérese además el uso de la función saturación, con el fin de evitar conmutación de alta frecuencia (fenómeno de chattering) del controlador al estar cerca de la superficie deslizante debido a la función signo involucrada, sea entonces

$$\text{sat}(\sigma) = \begin{cases} \text{sgn}(\sigma) & \text{si } |\sigma| \geq 1 \\ \sigma & \text{si } |\sigma| < 1 \end{cases} \tag{5.9}$$

Por otra parte, como se vio en el capítulo anterior, una típica entrada de control por modos deslizantes puede ser seleccionada como:

$$u_1 = \hat{u}_1 - k_1 \text{sat} \left(\frac{s_1 g_1(\mathbf{x})}{\Phi_1} \right), \quad k_1 > \frac{D_1}{|g_1(\mathbf{x})|} \tag{5.10}$$

¹SIMO - Sigle Input Multiple Outputs (por sus siglas en inglés)

con

$$\hat{u}_1 = \frac{-c_1 x_2 - f_1(\mathbf{x})}{g_1(\mathbf{x})} \quad (5.11)$$

o bien como

$$u_2 = \hat{u}_2 - k_2 \text{sat} \left(\frac{s_2 g_2(\mathbf{x})}{\Phi_2} \right), \quad k_2 > \frac{D_2}{|g_2(\mathbf{x})|} \quad (5.12)$$

con

$$\hat{u}_2 = \frac{-c_1 x_4 - f_2(\mathbf{x})}{g_2(\mathbf{x})} \quad (5.13)$$

donde $\Phi_1, \Phi_2 > 0$ representan el espesor de la capa límite para suavizar la entrada de control correspondiente [Slotine and Li, 1991]. Si se selecciona $u = u_1$, los estados x_1, x_2 convergen asintóticamente a cero a lo largo de la superficie deslizante s_1 , y los estados x_3, x_4 convergen a lo largo de la superficie deslizante s_2 cuando se selecciona $u = u_2$ como entrada de control. Sin embargo, dado que se desea controlar todos los estados únicamente mediante la entrada de control u_1 , la superficie deslizante s_1 es modificada de la siguiente forma:

$$s_1 = c_1(x_1 - z) + x_2 \quad (5.14)$$

con

$$z = \text{sat} \left(\frac{s_2}{\Phi_z} \right) z_v \quad (5.15)$$

donde z es proporcional a s_2 , transferida desde s_2 y $\Phi_z > 0$, es la capa límite para s_2 , empleada para suavizar z . El valor de Φ_z , coloca la superficie s_2 en un rango adecuado para x_1 . La ecuación (5.14) implica que la entrada de control u_1 es modificada de tal forma que x_1 y x_2 convergen a: $x_1 = z$ y $x_2 = 0$, [Lo and Kuo, 1998].

Puesto que debe garantizarse que la entrada de control empleada para controlar todo el sistema sea acotada, se supondrá entonces que:

$$|z| < z_v, \quad 0 < z_v < 1 \quad (5.16)$$

y dado que z_v es menor a la unidad, entonces z es una señal oscilatoria decreciente.

Para generar la entrada de control, el término de control aproximado (5.10) se considera de forma general como:

$$\hat{u}_1 = u = \frac{1}{\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)} \left[-c_1 x_2 - \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f) - k_p \text{sat} \left(\frac{s_1 \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)}{\Phi_1} \right) \right] \quad (5.17)$$

siendo $\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)$ y $\hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f)$ los valores estimados de las funciones $g_1(\mathbf{x})$ y $f_1(\mathbf{x})$ del sistema (6.8), respectivamente, y con $\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g) > 0$. Puesto que estas funciones en general son no lineales y contienen parámetros y dinámicas que suelen no considerarse en el modelado de la planta a controlar, se propone que sean aproximadas mediante una técnica difuso adaptable.

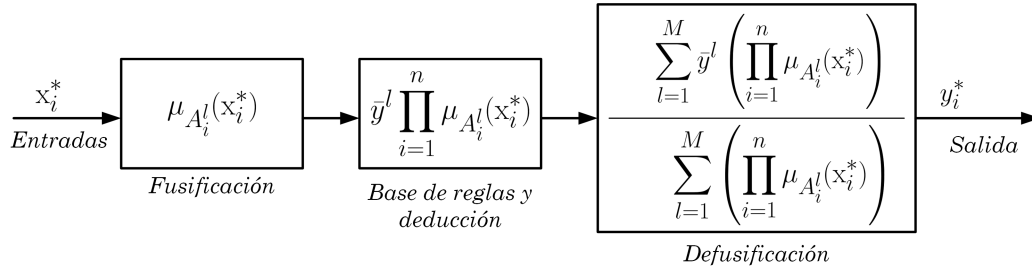


Figura 5.5: Diagrama de bloques del controlador difuso

5.5. Diseño del sistema difuso

Como ya se vio, los sistemas difusos son aproximadores universales y pueden aproximar funciones no lineales con cierta precisión arbitraria. Por tanto, se utilizarán sistemas difusos para aproximar las funciones $\hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f)$ y $\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)$ necesarias en la ley de control (5.17), donde:

$$\hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f) = \frac{\sum_{l_1=1}^{p_1} \cdots \sum_{l_n=1}^{p_n} \bar{y}_f^{l_1 l_2 \cdots l_n} \left(\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right)}{\sum_{l_1=1}^{p_1} \cdots \sum_{l_n=1}^{p_n} \left(\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right)} \quad (5.18)$$

$$\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g) = \frac{\sum_{l_1=1}^{q_1} \sum_{l_2=1}^{q_2} \cdots \sum_{l_n=1}^{q_n} \bar{y}_g^{l_1 l_2 \cdots l_n} \left(\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i) \right)}{\sum_{l_1=1}^{q_1} \cdots \sum_{l_n=1}^{q_n} \left(\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i) \right)} \quad (5.19)$$

donde x_i representa las variables de entrada a los sistemas difusos las cuales pueden ser los estados del sistema o estar en función de ellos. Los sistemas difusos propuestos (5.18) y (5.19) poseen fusificador impulsivo, inferencia max-prod y defusificador centro promedio; además son construidos mediante una base de reglas de tipo

Regla $(l_1 \dots l_n)$: SI x_1 es $A_1^{l_1}$ Y ... Y x_n es $A_n^{l_n}$ ENTONCES $\hat{f}_1(\mathbf{x})$ es $C^{l_1 \cdots l_n}$

para describir $f_1(\mathbf{x})$ con $l_i = 1, 2, \dots, p_i$ y

Regla $(l_1 \dots l_n)$: SI x_1 es $B_1^{l_1}$ Y ... Y x_n es $B_n^{l_n}$ ENTONCES $\hat{g}_1(\mathbf{x})$ es $D^{l_1 \cdots l_n}$

para describir $g_1(\mathbf{x})$ con $l_i = 1, 2, \dots, q_i$, $i = 1, 2, \dots, n$; donde A, B, C, D son funciones de pertenencia, y p_i y q_i son el número de funciones de pertenencia de la entrada x_i para aproximar $f_1(\mathbf{x})$ y $g_1(\mathbf{x})$, respectivamente. La figura 5.5 ilustra la estructura que lleva un controlador difuso de estas características. Para mejorar la aproximación de las funciones $\hat{f}_1(\mathbf{x})$ y $\hat{g}_1(\mathbf{x})$ se dejan algunos parámetros libres de cambiar durante la operación en línea y por lo tanto la precisión de la aproximación mejora con el paso del tiempo; estos parámetros libres sustituyen a las funciones de pertenencia de salida del sistema difuso, que normalmente son fijas. Sustituyendo $\bar{y}_f^{l_1 \cdots l_n}$ por los parámetros libres $\boldsymbol{\theta}_f \in \mathbb{R}^{\prod_{i=1}^{p_i} \times 1}$ en $\hat{f}_1(\mathbf{x})$, y $\bar{y}_g^{l_1 \cdots l_n}$ por

los parámetros libres $\boldsymbol{\theta}_g \in \mathbb{R}^{\prod_{i=1}^n q_i \times 1}$ en $\hat{g}_1(\mathbf{x})$, entonces (5.18) y (5.19) pueden reescribirse como:

$$\hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f) = \boldsymbol{\theta}_f^T \boldsymbol{\xi}(\mathbf{x}) \quad (5.20)$$

$$\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g) = \boldsymbol{\theta}_g^T \boldsymbol{\eta}(\mathbf{x}) \quad (5.21)$$

con el vector $\boldsymbol{\xi}(\mathbf{x})$ de dimensiones $\prod_{i=1}^n p_i \times 1$, cuyos elementos son:

$$\xi_{l_1 \dots l_n}(\mathbf{x}) = \frac{\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i)}{\sum_{l_1=1}^{p_1} \dots \sum_{l_n=1}^{p_n} \left(\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right)} \quad (5.22)$$

y con el vector $\boldsymbol{\eta}(\mathbf{x})$ de dimensiones $\prod_{i=1}^n q_i \times 1$, cuyos elementos son:

$$\eta_{l_1 \dots l_n}(\mathbf{x}) = \frac{\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i)}{\sum_{l_1=1}^{q_1} \dots \sum_{l_n=1}^{q_n} \left(\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i) \right)} \quad (5.23)$$

5.6. Diseño de la ley de adaptación

Considérese el error mínimo de aproximación como:

$$\omega = \left[f_1(\mathbf{x}|t) - \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f^*) \right] + \left[g_1(\mathbf{x}|t) - \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g^*) \right] u \quad (5.24)$$

siendo $\hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f^*)$ y $\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g^*)$, las mejores aproximaciones de $f_1(\mathbf{x}, t)$ y $g_1(\mathbf{x}, t)$, respectivamente, con $\boldsymbol{\theta}_f^*$ y $\boldsymbol{\theta}_g^*$ como los parámetros de adaptación óptimos. La tarea es encontrar un mecanismo de adaptación (ley adaptable) tal que $\boldsymbol{\theta}_f \rightarrow \boldsymbol{\theta}_f^*$ y $\boldsymbol{\theta}_g \rightarrow \boldsymbol{\theta}_g^*$, de forma que el error de aproximación sea mínimo; esta ley de adaptación se obtiene a partir del análisis de estabilidad de Lyapunov.

5.6.1. Función candidata de Lyapunov

Se propone la siguiente función candidata de Lyapunov ([Azimi et al., 2014]):

$$V(s_1, \tilde{\boldsymbol{\theta}}_f, \tilde{\boldsymbol{\theta}}_g) = \frac{1}{2} s_1^2 + \frac{1}{2\gamma_1} \tilde{\boldsymbol{\theta}}_f^T \tilde{\boldsymbol{\theta}}_f + \frac{1}{2\gamma_2} \tilde{\boldsymbol{\theta}}_g^T \tilde{\boldsymbol{\theta}}_g \quad (5.25)$$

la cual es una función definida positiva; γ_1, γ_2 son constantes positivas y $\tilde{\boldsymbol{\theta}}_f = \boldsymbol{\theta}_f^* - \boldsymbol{\theta}_f$ y $\tilde{\boldsymbol{\theta}}_g = \boldsymbol{\theta}_g^* - \boldsymbol{\theta}_g$ denotan los errores de estimación de los parámetros de adaptación.

5.6.2. Derivada de la función candidata de Lyapunov

La derivada de la función candidata de Lyapunov se calcula como:

$$\dot{V}(s_1, \tilde{\boldsymbol{\theta}}_f, \tilde{\boldsymbol{\theta}}_g) = s_1 \dot{s}_1 + \frac{1}{\gamma_1} \tilde{\boldsymbol{\theta}}_f^\top \dot{\tilde{\boldsymbol{\theta}}}_f + \frac{1}{\gamma_2} \tilde{\boldsymbol{\theta}}_g^\top \dot{\tilde{\boldsymbol{\theta}}}_g; \quad (5.26)$$

ahora es necesario evaluar la derivada de la función candidata de Lyapunov a lo largo de las trayectorias del sistema; para esto, primero se evaluará la derivada temporal de (5.14), resultando en:

$$\dot{s}_1 = c_1 \dot{x}_1 - c_1 \dot{z} + \dot{x}_2 \quad (5.27)$$

Ahora bien, haciendo uso de la definición (5.24) es posible expresar \dot{x}_2 de (5.6) como:

$$\begin{aligned} \dot{x}_2 &= \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f^*) + \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g^*)u + \omega + d_1(t) \\ &= \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f^*) + \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g^*)u + \underbrace{\left[f_1(\mathbf{x}|t) - \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f^*) \right] + \left[g_1(\mathbf{x}|t) - \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g^*) \right]}_{\omega} u + d_1(t) \\ &= f_1(\mathbf{x}|t) + g_1(\mathbf{x}|t)u + d_1(t) \end{aligned} \quad (5.28)$$

y tomando en cuenta que $\dot{x}_1 = x_2$, la ec. (5.28) al sustituirla en (5.27), se tiene:

$$\begin{aligned} \dot{s}_1 &= c_1 x_2 - c_1 \dot{z} + \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f^*) + \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g^*)u + \omega + d_1(t) \\ &= c_1 x_2 - c_1 \dot{z} + \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f^*) + \underbrace{\hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f) - \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f^*)}_{\text{cero}} + \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g^*)u + \underbrace{[\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g) - \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g^*)]}_{\text{cero}} u \\ &\quad + \omega + d_1(t) \\ &= c_1 x_2 - c_1 \dot{z} + \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f^*) - \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f) + [\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g^*) - \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)]u + \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f) \\ &\quad + \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)u + \omega + d_1(t) \end{aligned} \quad (5.29)$$

Tomando en cuenta las aproximaciones (5.18)-(5.19), entonces $\hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f^*) = \boldsymbol{\theta}_f^{*\top} \boldsymbol{\xi}(\mathbf{x})$, $\hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f) = \boldsymbol{\theta}_f^\top \boldsymbol{\xi}(\mathbf{x})$, $\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g^*) = \boldsymbol{\theta}_g^{*\top} \boldsymbol{\eta}(\mathbf{x})$, y $\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g) = \boldsymbol{\theta}_g^\top \boldsymbol{\eta}(\mathbf{x})$. Así pues (5.29) se expresa como:

$$\begin{aligned} \dot{s}_1 &= c_1 x_2 - c_1 \dot{z} + \underbrace{[\boldsymbol{\theta}_f^{*\top} - \boldsymbol{\theta}_f^\top]}_{\tilde{\boldsymbol{\theta}}_f^\top} \boldsymbol{\xi}(\mathbf{x}) + \underbrace{[\boldsymbol{\theta}_g^{*\top} - \boldsymbol{\theta}_g^\top]}_{\tilde{\boldsymbol{\theta}}_g^\top} \boldsymbol{\eta}(\mathbf{x})u + \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f) \\ &\quad + \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g) \left[\frac{1}{\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)} \left(-c_1 x_2 - \hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f) - k_p \text{sat} \left(\frac{s_1 \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)}{\Phi_1} \right) \right) \right] + \omega + d_1(t) \\ &= -c_1 \dot{z} + \tilde{\boldsymbol{\theta}}_f^\top \boldsymbol{\xi}(\mathbf{x}) + \tilde{\boldsymbol{\theta}}_g^\top \boldsymbol{\eta}(\mathbf{x})u - p(\mathbf{x}) + \omega + d_1(t) \end{aligned} \quad (5.30)$$

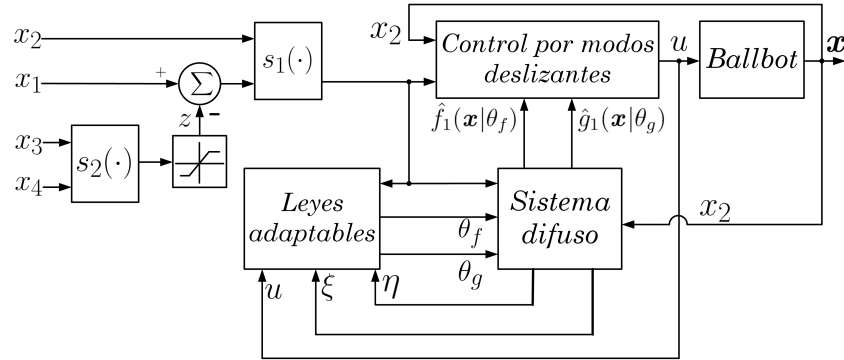


Figura 5.6: Diagrama de control por modos deslizantes con aproximación difuso adaptable

con $p(\mathbf{x}) = k_p \text{sat} \left(\frac{s_1 \hat{g}(\mathbf{x}|\theta_g)}{\Phi_1} \right)$. Sustituyendo (5.29) en la derivada (5.26) y tomando en consideración que $\dot{\tilde{\theta}}_g = -\dot{\theta}_g$ y $\dot{\tilde{\theta}}_f = -\dot{\theta}_f$, se tiene que:

$$\begin{aligned} \dot{V}(\cdot) &= -c_1 s_1 \dot{z} + s_1 \tilde{\theta}_f^\top \boldsymbol{\xi}(\mathbf{x}) + s_1 \tilde{\theta}_g^\top \boldsymbol{\eta}(\mathbf{x}) u - s_1 p(\mathbf{x}) + s_1 (\omega + d_1(t)) + \frac{1}{\gamma_1} \tilde{\theta}_f^\top \dot{\tilde{\theta}}_f + \frac{1}{\gamma_2} \tilde{\theta}_g^\top \dot{\tilde{\theta}}_g \\ &= -c_1 s_1 \dot{z} - s_1 p(\mathbf{x}) + s_1 (\omega + d_1(t)) + \left[s_1 \boldsymbol{\xi}(\mathbf{x}) + \frac{1}{\gamma_1} \dot{\tilde{\theta}}_f \right] \tilde{\theta}_f^\top + \left[s_1 \boldsymbol{\eta}(\mathbf{x}) u + \frac{1}{\gamma_2} \dot{\tilde{\theta}}_g \right] \tilde{\theta}_g^\top \\ &= -c_1 s_1 \dot{z} - s_1 p(\mathbf{x}) + s_1 (\omega + d_1(t)) + \left[s_1 \boldsymbol{\xi}(\mathbf{x}) - \frac{1}{\gamma_1} \dot{\theta}_f \right] \tilde{\theta}_f^\top + \left[s_1 \boldsymbol{\eta}(\mathbf{x}) u - \frac{1}{\gamma_2} \dot{\theta}_g \right] \tilde{\theta}_g^\top \end{aligned}$$

Para minimizar los errores de los parámetros de adaptación $(\theta_f - \theta_f^*)$ y $(\theta_g - \theta_g^*)$, y a la vez, minimizar V , se selecciona la siguiente ley de adaptación

$$\dot{\theta}_f = \gamma_1 s_1 \boldsymbol{\xi}(\mathbf{x}) \quad (5.31)$$

$$\dot{\theta}_g = \gamma_2 s_1 \boldsymbol{\eta}(\mathbf{x}) u \quad (5.32)$$

La figura 5.6 ilustra el diagrama de bloques del sistema de control en lazo cerrado, con superficies deslizantes (5.8) y (5.14)-(5.15); ley de control (5.17); sistema difuso (5.20)-(5.23) y ley de adaptación (5.31)-(5.32).

5.7. Análisis de estabilidad

Del resultado anterior, la derivada temporal de V se reduce a

$$\begin{aligned} \dot{V}(s_1, \theta_f, \theta_g) &\leq -c_1 s_1 \dot{z} - s_1 p(\mathbf{x}) + s_1 (\omega + d_1(t)) \\ &\leq -c_1 s_1 \dot{z} - s_1 k_p \text{sat} \left(\frac{s_1 \hat{g}_1(\mathbf{x}|\theta_g)}{\Phi} \right) + s_1 \omega + s_1 d_1(t) \end{aligned}$$

Considerando el hecho de que $d_1(t) \leq D_1$, $\text{sat}(s_1 \hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)/\Phi_1) \leq \text{sgn}(s_1)$ y $s_1 \leq |s_1|$ entonces $\dot{V}(\cdot)$ se reduce a

$$\begin{aligned} \dot{V}(s_1, \boldsymbol{\theta}_f, \boldsymbol{\theta}_g) &\leq -s_1 c_1 \dot{z} - s_1 k_p \text{sgn}(s_1) + s_1 \omega + D_1 |s_1| \\ &\leq -s_1 c_1 \dot{z} - k_p |s_1| + s_1 \omega + D_1 |s_1| \\ &\leq -s_1 c_1 \dot{z} - (k_p - D_1) |s_1| + s_1 \omega \\ &\leq -s_1 h - \eta_1 |s_1| + s_1 \omega \end{aligned} \quad (5.33)$$

con $h = c_1 \dot{z}$ y $\eta_1 = k_p - D_1$, $\eta_1 > 0$. De acuerdo a la definición de z en (5.16) y eligiendo un valor lo suficientemente grande de Φ_z , es posible concluir que \dot{z} está acotada y es *uniformemente continua*, entonces $\dot{z} \rightarrow 0$ conforme $t \rightarrow \infty$, y por (5.14), $s_1 \rightarrow 0$. Si $x_1(0)$ es acotado, entonces $x_1(t)$ permanece acotado para todo t , y todos los estados del sistema \mathbf{x} , permanecerán acotados.

A continuación sigue demostrar que $\dot{s}_1 \rightarrow 0$ conforme $t \rightarrow \infty$. Para ello, de (5.33), es posible establecer una cota superior tal que $|s_1| \leq \eta_s$, entonces la derivada de V se expresa como

$$\dot{V}(s_1, \boldsymbol{\theta}_f, \boldsymbol{\theta}_g) \leq s_1 |\omega| - \eta_1 |s_1| \leq \underbrace{\eta_s |\omega| - \eta_1 |s_1|}_{\text{cota superior}} \quad (5.34)$$

Integrando ambos lados de (5.34) se tiene

$$\frac{dV}{dt} \leq \eta_s |\omega| - \eta_1 |s_1| \quad (5.35)$$

$$\begin{aligned} \int_0^t dV &\leq \int_0^t (\eta_s |\omega| - \eta_1 |s_1|) dt \\ V(t) - V(0) &\leq \int_0^t (\eta_s |\omega| - \eta_1 |s_1|) dt \end{aligned} \quad (5.36)$$

Entonces

$$V(0) \geq V(t) + \int_0^t (\eta_1 |s_1| - \eta_s |\omega|) dt \quad (5.37)$$

Por tanto se tiene que

$$\lim_{t \rightarrow \infty} \int_0^t (\eta_1 |s_1| - \eta_s |\omega|) dt < \infty \quad (5.38)$$

Así pues de (5.38) existen

$$\lim_{t \rightarrow \infty} \int_0^t (\eta_1 |s_1|) dt < \infty \quad \text{y} \quad \lim_{t \rightarrow \infty} \int_0^t (\eta_s |\omega|) dt < \infty \quad (5.39)$$

Por definición de (5.24), se sabe que ω permanece acotada. Por lo tanto se concluye que

$$s_1 \in L_2$$

De (5.36) y (6.30) se tiene que

$$\frac{1}{2}s_1^2 \leq V(t) \leq V(0) + \int_0^t (\eta_s|\omega| - \eta_1|s_1|) dt \leq V(0) < \infty \quad (5.40)$$

por tanto

$$s_1 \in L_\infty \quad (5.41)$$

De (5.35) y (5.26) es posible escribir

$$\frac{dV}{dt} = s \frac{ds_1}{dt} \leq \eta_s|\omega| - \eta_1|s_1| < \infty \quad (5.42)$$

Por lo tanto

$$\dot{s}_1 \in L_\infty \quad (5.43)$$

Así pues, en términos del Lema de Barbalat se puede concluir que $s_1(t) \rightarrow 0$ conforme $t \rightarrow \infty$ y por tanto $z \rightarrow 0$ y $s_2 \rightarrow 0$, de tal forma que el sistema en lazo cerrado es estable y todas las señales permanecen acotadas y el error de aproximación convergerá asintóticamente a cero.

En resumen, la operación del sistema de aproximación difuso adaptable expresa lo siguiente:

- Los valores deseados son comparados con las mediciones de los estados (que se suponen disponibles) y esto genera los errores.
- Por otra parte, el sistema difuso recibe como entradas, para sus módulos de fusificación, a los estados que intervienen en la aproximación de las funciones $\hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f)$ y $\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)$, y los parámetros adaptables $\boldsymbol{\theta}_f$ y $\boldsymbol{\theta}_g$ como centros de las funciones de pertenencia de salida; las funciones aproximadas se incorporan a la ley de control por modos deslizantes desacoplados.
- Los parámetros adaptables $\boldsymbol{\theta}_f$ y $\boldsymbol{\theta}_g$ se calculan de acuerdo a la ley de adaptación que se obtuvo con el análisis de Lyapunov, utilizando los errores y los vectores $\boldsymbol{\xi}(\mathbf{x})$ y $\boldsymbol{\eta}(\mathbf{x})$ de los sistemas difusos; adicionalmente se incluyen condiciones iniciales para los parámetros adaptables con el objetivo de reducir el tiempo de adaptación y también para evitar que $\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)$ sea cero en un inicio. Estas condiciones iniciales pueden obtenerse a partir del conocimiento previo de la planta o del mismo modelo matemático, si se tiene disponible.

5.8. Resultados de simulación

En esta sección se presentan los resultados de simulación para la estabilización del *ballbot*. Las simulaciones fueron desarrolladas en MATLAB Simulink 2016a empleando el integrador de Euler (ode1) de tipo paso fijo a un periodo de muestreo de 0.001 [s].

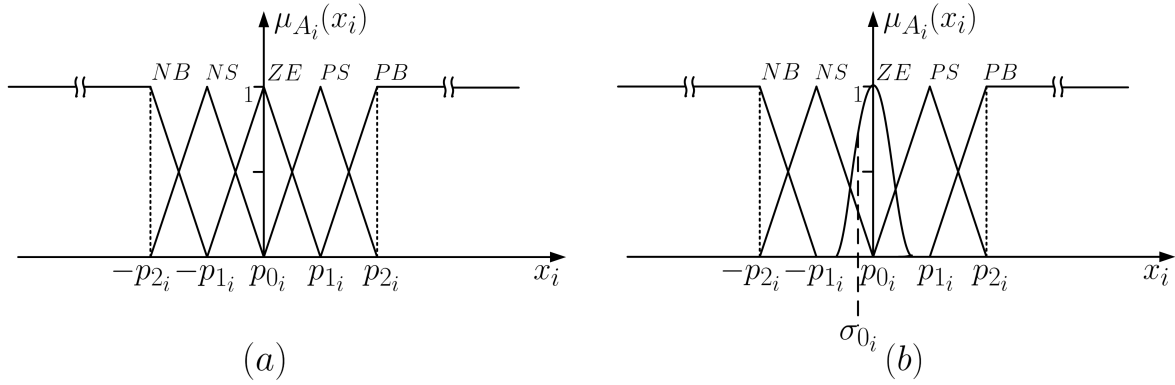


Figura 5.7: Funciones de pertenencia de entrada: (a) Caso I y (b) Caso II

5.8.1. Parámetros de diseño

El controlador por modos deslizantes basado en aproximación difuso adaptable posee diferentes parámetros propios del diseño que se requieren para estabilizar el desplazamiento angular del cuerpo $x_1 = q_1$.

Las ganancias utilizadas para el controlador fueron seleccionadas en base a la experiencia adquirida durante un proceso de sintonización de prueba y error, resultando los siguientes valores:

$$\begin{array}{lll}
 c_1 = 7 \text{ [1/s]} & k_p = 45 & z_v = 0.2 \\
 c_2 = 0.8 \text{ [1/s]} & \Phi_1 = 1 & \Phi_z = 14
 \end{array}$$

Constantes de adaptación

Las constantes de adaptación, γ_1 y γ_2 , también se seleccionaron mediante un proceso de sintonización basado en la experiencia, resultando los siguientes valores:

$$\begin{array}{l}
 \gamma_1 = 100 \\
 \gamma_2 = 3
 \end{array}$$

Sistema difuso

Como ya se vio, el sistema difuso funciona como un aproximador de funciones; como se están tratando de aproximar las funciones $f_1(\mathbf{x})$ y $g_1(\mathbf{x})$, las variables de entrada son los

estados del sistema a través de la superficie deslizante s_1 , y se requieren dos sistemas difusos para conseguir este objetivo. Para cada sistema de aproximación difuso se emplearon cinco funciones de pertenencia o conjuntos difusos de entrada para cada una de las dos entradas; considerando dos posibles casos:

- **Caso I:** Tres funciones de pertenencia de entrada tipo triangular, una de tipo trapezoidal izquierda y una de tipo trapezoidal derecha (figura. 5.7a).
- **Caso II:** Dos funciones de pertenencia de entrada tipo triangular, una de tipo gaussiana en el centro, una de tipo trapezoidal izquierda y una de tipo trapezoidal derecha, similar a [Hamed et al., 2017] (figura. 5.7b). La intención de cambiar por una función gaussiana en el centro es debido al hecho de que alrededor de velocidad angular cero, las variaciones de velocidad son grandes y el sistema es sensible en esa región.

Las mismas funciones de pertenencia de entrada fueron utilizadas en los dos sistemas difusos, para el cálculo tanto de $\hat{f}_1(\mathbf{x}|\boldsymbol{\theta}_f)$ como de $\hat{g}_1(\mathbf{x}|\boldsymbol{\theta}_g)$. Cada uno de los dos sistemas difusos fue particionado en cinco conjuntos difusos: $A_i^{-2} = NB$ (Negativo Grande), $A_i^{-1} = NS$ (Negativo Pequeño), $A_i^0 = ZE$ (Cero), $A_i^1 = PS$ (Positivo Pequeño) y $A_i^2 = PB$ (Positivo Grande), con $i = 1, 2$ como el número de entrada.

Las particiones de los universos de discurso para ambos casos, utilizando la notación $p_{A_i} = \{-p_{2_i}, -p_{1_i}, p_{0_i}, p_{1_i}, p_{2_i}\}$, fueron seleccionadas como:

$$p_{s_1} = \{-10, -5, 0, 5, 10\} \left[\frac{\text{grados}}{\text{s}} \right]$$

$$p_{x_2} = \{-330, -180, 0, 180, 330\} \left[\frac{\text{grados}}{\text{s}} \right]$$

recordando que para el Caso II, la función de membresía central es reemplazada por una función de tipo gaussiana con desviaciones estándar:

$$\sigma_{0_{s_1}} = \{1\} \left[\frac{\text{grados}}{\text{s}} \right]$$

$$\sigma_{0_{x_2}} = \{15\} \left[\frac{\text{grados}}{\text{s}} \right]$$

Como ya se mencionó en las consideraciones teóricas del sistema difuso adaptable, las funciones de pertenencia de salida son impulsivas pero no son fijas, sino que son parámetros libres que provienen de la ley de adaptación. El uso de *singletons* simplifica el proceso de defusificación, ya que permite calcular la salida directamente como un promedio ponderado. En la figura 5.8 se ilustra una representación de esto. Puesto que se tienen cinco funciones de pertenencia de entrada por cada entrada, entonces $N_1 = 5$, $N_2 = 5$ y $M = 25$; es decir, se tienen 25 reglas e igual número de parámetros de adaptación $\boldsymbol{\theta}_f$ y $\boldsymbol{\theta}_g$. La tabla 5.1 muestra la disposición de las reglas de cada sistema difuso, donde $r = f, g$ denota el sistema difuso.

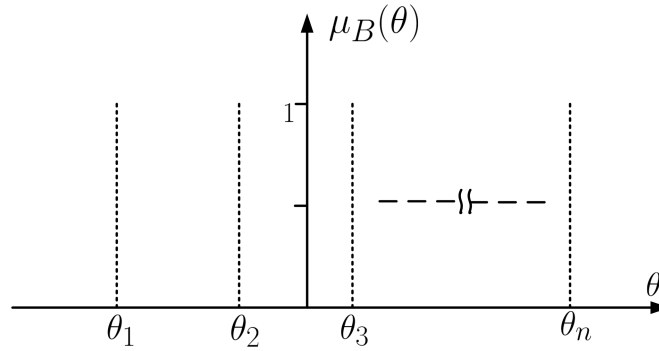


Figura 5.8: Funciones de pertenencia de salida

Tabla 5.1: Tabla de búsqueda del sistema difuso adaptable

| $x_2 \backslash s_1$ | $l_1 = -2$ NB | $l_1 = -1$ NS | $l_1 = 0$ Z | $l_1 = 1$ PS | $l_1 = 2$ PB |
|----------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| $l_2 = -2$ NB | θ_{r_1} | θ_{r_2} | θ_{r_3} | θ_{r_4} | θ_{r_5} |
| $l_2 = -1$ NS | θ_{r_6} | θ_{r_7} | θ_{r_8} | θ_{r_9} | $\theta_{r_{10}}$ |
| $l_2 = 0$ Z | $\theta_{r_{11}}$ | $\theta_{r_{12}}$ | $\theta_{r_{13}}$ | $\theta_{r_{14}}$ | $\theta_{r_{15}}$ |
| $l_2 = 1$ PS | $\theta_{r_{16}}$ | $\theta_{r_{17}}$ | $\theta_{r_{18}}$ | $\theta_{r_{19}}$ | $\theta_{r_{20}}$ |
| $l_2 = 2$ PB | $\theta_{r_{21}}$ | $\theta_{r_{22}}$ | $\theta_{r_{23}}$ | $\theta_{r_{24}}$ | $\theta_{r_{25}}$ |

5.8.2. Resultados

Con el propósito de comprobar la efectividad del esquema de control propuesto para la estabilización del sistema *ballbot* su desempeño es evaluado mediante simulaciones numéricas. Los parámetros del *ballbot* son tomados de la tabla 2.3. El vector inicial de estados es $\mathbf{x}(0) = [5 \ 0 \ 0 \ 0]^T$ y el vector de estados deseado es $\mathbf{x}_d(0) = [0 \ 0 \ 0 \ 0]^T$. Esto significa que el objetivo de control es estabilizar el cuerpo en la posición vertical. Las simulaciones son desarrolladas considerando el movimiento del *ballbot* solo en un plano. Para probar el esquema de control bajo incertidumbres paramétricas y perturbaciones externas; se considera que los parámetros de masa e inercia varían en un 20 % de su valor nominal, tomando los valores de: $m_B = 0.736 + 0.1472 \text{sen}(t)$ [kg], $m_b = 0.126 + 0.252 \text{sen}(t)$ [kg], $I_B = 0.0043 + 0.00086 \text{sen}(t)$ [kgm²], $I_b = [5.6784 + 1.1357 \text{sen}(t)] \times 10^{-6}$ [kgm²], $I_M = [1.0 + 0.4 \text{sen}(t)] \times 10^{-5}$ [kgm²]; una perturbación externa es considerada con valor $d_1(t) = 1.6 \cos(5t)$ más ruido blanco aplicado en la medición del ángulo del cuerpo. La figura 5.9 muestra los resultados de simulación de la estrategia de control estudiada en este capítulo, basada en aproximación difuso adaptable, la cual es contrastada con una estrategia de control lineal LQR ampliamente reportada en trabajos de control sobre el *ballbot* en sus versiones LEGO y a gran escala.

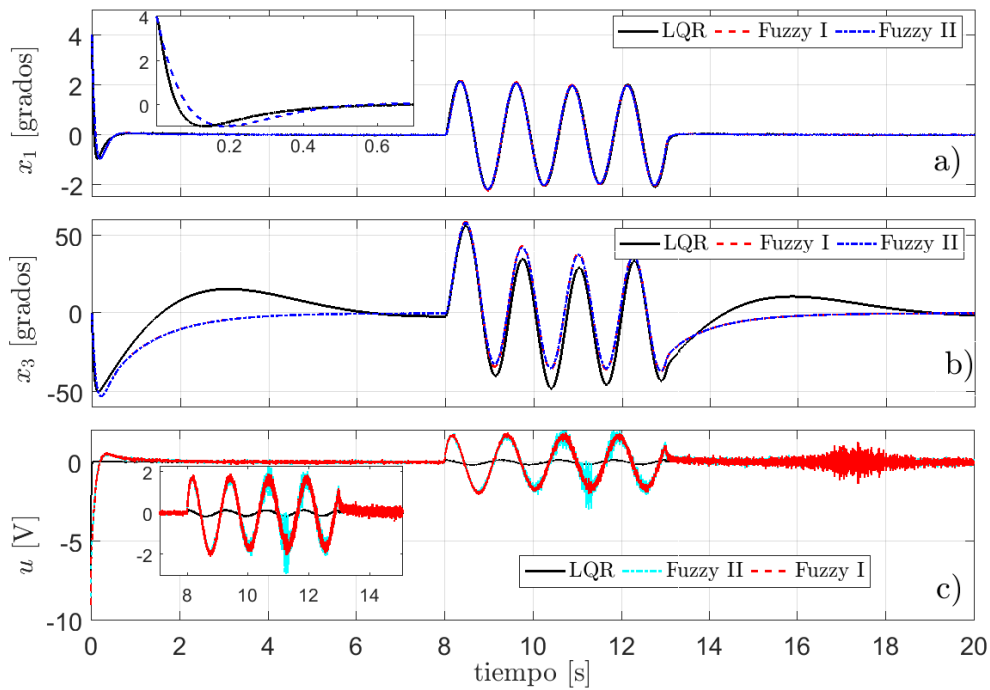


Figura 5.9: Simulación de control por modos deslizantes basado en aproximación difuso adaptable

En la figura 5.9a se observa que ambas estrategias de control –lineal y difuso adaptable– logran estabilizar el cuerpo del *ballbot* en su posición vertical hacia arriba en un tiempo relativamente corto. La figura 5.9b muestra la respuesta de la posición angular del motor de la llanta; en la que se observa que el control basado en aproximación difuso adaptable tiene una respuesta menos oscilatoria que el control lineal LQR (ver apéndice B) basado en el modelo. Esto significa el *ballbot* tendrá un menor desplazamiento en el plano $x - y$, ya que el par de la llanta influye sobre el desplazamiento de la bola (esfera) que soporta al *ballbot*. Finalmente en la figura 5.9c se muestra la señal de control demandada por el motor para lograr la estabilización del cuerpo.

5.9. Comentarios

- (i) En este capítulo se introdujo una estrategia de control no lineal por modos deslizantes basada en una aproximación difuso adaptable; la cual es una alternativa de solución al problema de control de sistemas subactuados que presenten incertidumbres paramétricas y dinámicas no modeladas, manteniendo su estabilidad y buen desempeño.
- (ii) Aunque existen cuatro funciones desconocidas en el sistema bajo consideración (5.6), solamente es necesario aproximar dos de ellas para el cálculo de la señal de control.

Esto facilita la implementación del algoritmo de control propuesto.

- (iii) El costo computacional requerido por el sistema difuso para aproximar las funciones no lineales es considerable, por lo tanto la capacidad de cómputo para la implementación real de este algoritmo es un factor importante a considerar.

Control por modos deslizantes basado en aproximación neuronal adaptable

En este capítulo se trata lo referente al control de sistemas subactuados mediante técnicas de control robusto y redes neuronales. Se aborda la descripción básica de las redes neuronales; así también, se mencionan las propiedades de las redes neuronales como aproximadores universales. Finalmente, aprovechando estas propiedades, se combinan dos tipos de redes neuronales con el control por modos deslizantes para formular una estrategia de control que permita estabilizar a un sistema no lineal subactuado conocido como *ballbot*.

6.1. Control Neuronal

6.1.1. ¿Por qué el control neuronal?

Los sistemas mecánicos subactuados (UMS¹) han ido en incremento en el campo de la robótica. Estos se caracterizan por tener menos actuadores que grados de libertad. Una de las razones por las que la subactuación surge puede ser con el propósito de reducir el peso del robot o puede ser también causada por la falla de algún actuador.

La dificultad para controlar los mecanismos subactuados se debe al hecho de que las técnicas desarrolladas para sistemas completamente actuados no se pueden usar directamente. Debido a esto, los sistemas mecánicos subactuados presentan retos desafiantes de control. Además, se demuestra que esta clase de sistemas no pueden ser estabilizados mediante algoritmos de control lineal –excepto que se haga uso de alguna técnica de linealización y se tenga pleno conocimiento del modelo de la planta y sus parámetros–. El enfoque de control por modos

¹UMS - Underactuated Mechanical Systems –por sus siglas en inglés–

deslizantes (SMC) es una de las técnicas más comunes que se utiliza para controlar sistemas mecánicos subactuados. La efectividad del control por modos deslizantes para el control de sistemas no lineales en general ha sido probada por la comunidad de ingeniería de control en virtud de su robustez contra perturbaciones externas y variaciones en los parámetros del modelo. Sin embargo, como ya se ha mencionado, el uso de este enfoque provoca en la práctica el fenómeno indeseado del chattering, el cual puede provocar la excitación de dinámicas de alta frecuencia. Para evitar este efecto, la capa límite de la superficie de conmutación puede ser reducida. Sin embargo esta aproximación no asegura la convergencia de las trayectorias de los estados del sistema sobre la superficie deslizante. Por otra parte, el diseño de controladores por modos deslizantes requiere del conocimiento del modelo matemático de la planta, lo cual demerita el desempeño en algunas aplicaciones donde el modelo matemático del sistema no está bien definido o cuando el conocimiento de los parámetros es incompleto. Para sobrellevar este problema, algunos investigadores han empleado redes neuronales artificiales (ANN –Artificial Neural Networks–) ([Qian et al., 2011], [Mahjoub et al., 2014], [Chaudhari and Kar, 2017] y [Su et al., 2017]).

Desde la década de 1940 se introdujo la idea de las habilidades computacionales de las redes compuestas de modelos simples de neuronas [McCulloch and Pitts, 1943]. Las técnicas de redes neuronales han experimentado grandes desarrollos y se han aplicado con éxito en muchos campos, como el aprendizaje, el reconocimiento de patrones, el procesamiento de señales, el modelado y el control de sistemas. Sus principales ventajas son la capacidad de aprendizaje, la aproximación de funciones no lineales, la tolerancia a fallas y la implementación eficiente para aplicaciones en tiempo real, lo que motiva enormemente el uso de redes neuronales en la identificación y el control de sistemas no lineales.

Como ya se mencionó, en muchas aplicaciones del mundo real hay muchas no linealidades, dinámicas sin modelar, ruido no medible, lazos múltiples, etc., que plantean problemas para que los ingenieros implementen estrategias de control.

Existen varias razones que han motivado el interés de investigación en la aplicación de redes neuronales con fines de control como alternativas a los métodos de control tradicionales, entre las cuales se encuentran:

- Las redes neuronales pueden ser entrenadas para aprender cualquier función. Por lo tanto, esta capacidad de autoaprendizaje de las redes neuronales elimina el uso de análisis matemáticos complejos y difíciles que son dominantes en muchos métodos tradicionales de control adaptable y óptimo.
- La inclusión de las funciones de activación en las neuronas ocultas de las redes neuronales de múltiples capas ofrecen una capacidad de mapeo no lineal para resolver problemas de control altamente no lineales donde, para este fin, los enfoques de control tradicionales aún no tienen una solución práctica.
- El requisito de una vasta información a priori con respecto a la planta que se va a controlar, como el modelado matemático, es una necesidad previa en las técnicas tradicionales

de control adaptable y óptimo antes de que puedan implementarse. Debido a la capacidad de autoaprendizaje de las redes neuronales, tal información no es necesaria para los controladores neuronales. Por lo tanto, los controladores neuronales parecen poder aplicarse bajo un rango más amplio de incertidumbre.

- El paralelismo masivo de las redes neuronales ofrece una técnica de multiprocesamiento muy rápida cuando se implementa utilizando chips neuronales o hardware paralelo.
- El daño a algunas partes del hardware de la red neuronal puede no afectar el rendimiento debido a su arquitectura masiva de procesamiento paralelo.

6.1.2. Revisión del control por redes neuronales

Los métodos convencionales de diseño de controladores para una planta MIMO (Multiple Input Multiple Output) como un robot de múltiples eslabones generalmente requieren, como mínimo, el conocimiento de la estructura y el modelo matemático preciso de la planta. En muchos casos, los valores de los parámetros del modelo también deben ser conocidos con precisión.

Las redes neuronales, que pueden aprender el comportamiento dinámico de plantas complejas en línea, ofrecen métodos alternativos para realizar controladores MIMO capaces de adaptarse a los cambios del entorno de la planta. En teoría, el diseño de un sistema de control basado en redes neuronales es relativamente sencillo, ya que no requiere ningún conocimiento previo sobre la planta. Las habilidades de aproximación de las redes neuronales han sido probadas en varios trabajos de investigación ([Barron, 1991], [Barron, 1993] y [Chen and Chen, 1995]), y algunos controladores adaptables de redes neuronales basados en las habilidades de aproximación se presentan en algunos libros ([Lewis et al., 1998], [Ge et al., 2002]); varios grupos de investigación han participado en el desarrollo de técnicas de control empleando redes neuronales adaptables.

Se han publicado muchos artículos sobre control con redes neuronales. Por ejemplo, en [Polycarpou, 1996] se propuso un marco unificado para la identificación y control de sistemas dinámicos no lineales, en el que se puede aplicar el método paramétrico tanto del control no lineal adaptable como de la teoría del control lineal adaptable para realizar el análisis de estabilidad. Los típicos esquemas de control por aproximación de redes neuronales estables basados en el diseño de entrenamiento de Lyapunov se dan en [Zhang and Ge, 2009], [Ssu-Hsin and Anuradha, 1997] y [Ssu-Hsin and Anuradha, 1998]. La popularización de la red neuronal de propagación hacia atrás (BP –Back Propagation–) y la red neuronal de función de base radial (RBF –Radial Basis Function–) han impulsado enormemente el desarrollo del control neuronal.

Recientemente se ha reportado un trabajo sobre control de múltiples *ballbots* empleando una combinación de control deslizante, sistemas difusos y redes neuronales ([Tsai and Tai, 2016]).

La combinación de algoritmos de control inteligente y el control por modos deslizantes ha atraído la atención de investigadores. La idea de combinar el control robusto y los métodos basados en redes neuronales es utilizada como una manera de mejorar el desempeño y robustez de los sistemas de control ante las imprecisiones del modelado.

6.2. Redes neuronales

Es bien sabido que los sistemas biológicos pueden realizar tareas complejas sin recurrir a operaciones cuantitativas explícitas. En particular, los organismos biológicos son capaces de aprender gradualmente con el paso del tiempo. Esta capacidad de aprendizaje refleja la capacidad de las neuronas biológicas para aprender a través de la exposición a estímulos externos. Tales propiedades de las neuronas biológicas las hace atractivas como modelos computacionales que pueden ser diseñados para procesar datos complejos.

Las redes neuronales, o más específicamente, las redes neuronales artificiales son modelos matemáticos inspirados en nuestro entendimiento del sistema nervioso biológico. Son atractivas como sistemas de cómputo que pueden aceptar un gran número de entradas y aprender únicamente de muestras de entrenamiento. Como modelos matemáticos para los sistemas nerviosos biológicos, las redes neuronales artificiales son útiles para establecer relaciones entre las entradas y salidas de cualquier tipo de sistema.

Rigurosamente hablando, una **red neuronal** es una colección de neuronas artificiales. Una **neurona artificial** es un modelo matemático de una neurona biológica en su forma más simple ([T. Nguyen et al., 2002]). Desde nuestro entendimiento, las neuronas biológicas son vistas como unidades elementales para el procesamiento de información en cualquier sistema nervioso. Sin pretender su validez neurobiológica, el modelo matemático de una neurona artificial se basa en la siguiente tesis:

1. Las neuronas son unidades elementales en un sistema nervioso en el cual ocurre el procesamiento de información.
2. La información entrante se presenta en forma de señales que se transmiten entre las neuronas a través de enlaces de conexión (*connection links*).
3. Cada enlace de conexión tiene un *peso* adecuado que multiplica a la señal transmitida.
4. Cada neurona tiene una acción interna, dependiente de un *umbral de disparo*, lo que resulta en una función de activación que se aplica a la suma ponderada de las señales de entrada para producir una señal de salida.

Así, cuando las señales de entrada x_1, x_2, \dots, x_n llegan a la neurona a través de los enlaces de conexión con pesos asociados w_1, w_2, \dots, w_n , respectivamente, la entrada resultante a la

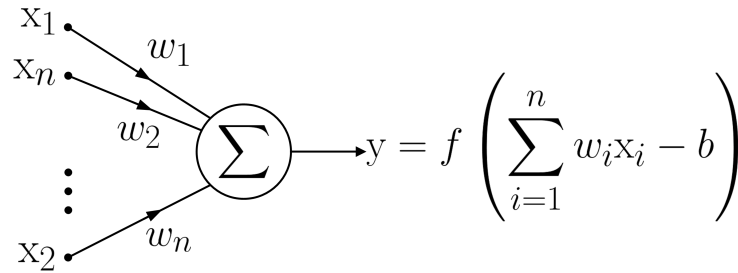


Figura 6.1: Primer modelo para una neurona artificial

neurona, llamada **entrada neta**, es la suma ponderada $\sum_{i=1}^n w_i x_i$. La figura 6.1 muestra el modelo para una neurona artificial, el cual es descrito por:

$$y = f \left(\sum_{i=1}^n w_i x_i - b \right)$$

donde:

x_i : señales de entrada, $i = 1, 2, \dots, n$

w_i : pesos sinápticos

b : umbral de disparo o polarizador

$f(\cdot)$: función de activación.

y : señal de salida de la neurona

En el primer modelo computacional ([McCulloch and Pitts, 1943]) para neuronas artificiales, las salidas son binarias, y la función f es la función escalón definida por:

$$f(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$$

para que la activación de la neurona sea

$$f \left(\sum_{i=1}^n w_i x_i - b \right) = \begin{cases} 1 & \text{si } \sum_{i=1}^n w_i x_i \geq b \\ 0 & \text{si } \sum_{i=1}^n w_i x_i < b \end{cases}$$

Una neurona artificial es caracterizada por los parámetros

$$\theta = (w_1, w_2, \dots, w_n, b, f)$$

El polarizador b puede ser tratado como otro “peso” añadiendo un nodo de entrada x_0 que siempre tome el valor de entrada $x_0 = +1$ y estableciendo $w_0 = -b$. Con esta representación,

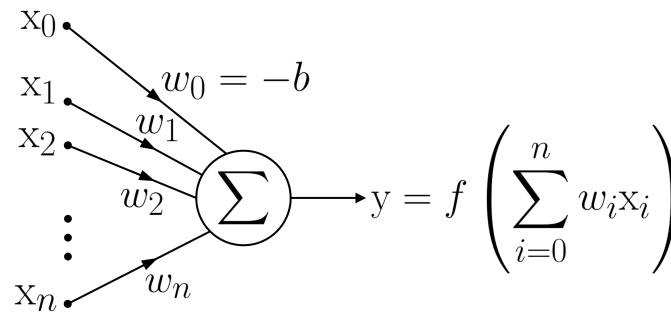


Figura 6.2: Neurona artificial con polarizador como peso

ajustar el polarizador se logra ajustando los pesos (figura 6.2). La configuración de red más sencilla son las redes neuronales con propagación hacia adelante (*feedforward*), es decir, la información solo se propaga hacia adelante como lo indica la dirección de las flechas.

Al ver las neuronas artificiales como unidades elementales para el procesamiento de la información, se llega a redes neuronales simples considerando varias neuronas a la vez. La red neuronal de la figura 6.3 consiste en una **capa de entrada** (– o capa de nodos de entrada–) y una **capa de salida** consistente de neuronas. Esta es conocida como una **red neuronal de una sola capa** debido a que la capa de entrada no es una capa de neuronas, esto es, no hay cálculos en los nodos de entrada. Esta red neuronal de una sola capa es llamada **perceptrón**.

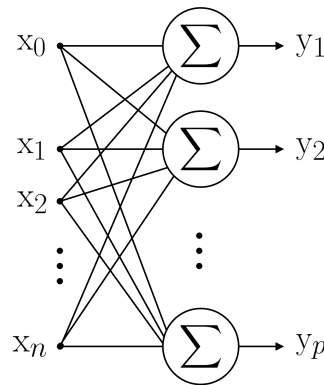


Figura 6.3: Perceptrón

Una **red neuronal multicapa** es una red con más de una capa de neuronas. Nótese que las funciones de activación de las diferentes neuronas puede ser diferente. Las neuronas de una capa tienen conexiones ponderadas con neuronas en la siguiente capa, pero no conexiones entre neuronas de la misma capa.

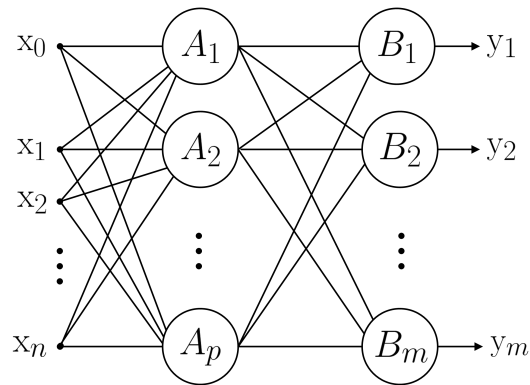


Figura 6.4: Red neuronal de dos capas

Una red neuronal de dos capas se ilustra en la figura 6.4. Las funciones de activación de diferentes neuronas pueden ser diferentes. La capa de entrada (o capa 0) tiene $n + 1$ nodos, la capa del medio, llamada **capa oculta**, tiene p nodos, y la capa de salida tiene m nodos. Esta es llamada red neuronal $n-p-m$.

Las neuronas (nodos) en cada capa son similares. Las neuronas en la capa oculta están ocultas en el sentido de que no se puede observar directamente su salida. A partir de los patrones de entrada sólo se pueden observar los patrones de salida desde la capa de salida. Por supuesto, una red neuronal multi-capas puede tener más de una capa oculta.

La red neuronal de dos capas de la figura 6.4 es un típico **perceptrón multi-capas** (MLP – multi-layer perceptron–). Una red neuronal multicapa cuyas neuronas realizan la misma función en las entradas, generalmente un compuesto de la suma ponderada y una función de activación no lineal diferenciable, o función de transferencia, tales como una función tangente hiperbólica. Los perceptrones multi-capas son las estructuras de redes neuronales más comúnmente utilizadas para un gran número de aplicaciones.

6.2.1. Funciones de base radial (RBF)

En los últimos años, el estudio de control adaptable de sistemas no lineales que utilizan la aproximación de función universal de base radial (RBF) ha atraído la atención de la comunidad científica de control; algunos de estos métodos se dan en [Sundararajan et al., 2002], [Sanner and Slotine, 1992], [Huang et al., 2002], [Li et al., 2004] y [Zhu et al., 2008].

La adaptación de la red con RBF puede mejorar efectivamente el rendimiento de control ante una gran incertidumbre en un sistema. La ley de adaptación es obtenida utilizando el método de Lyapunov para garantizar la estabilidad de todo el sistema y la convergencia de los pesos de adaptación.

Red neuronal con Función de Base Radial

En 1990, se propusieron por primera vez redes neuronales artificiales para el control adaptable de sistemas dinámicos no lineales [Narendra and Parthasarathy, 1990]. Desde entonces, las redes neuronales multicapa (MNN -Multilayer Neural Networks-) y la función de base radial (RBF) se han utilizado en numerosas aplicaciones para la identificación y el control de sistemas dinámicos.

Las redes neuronales con funciones de base radial (RBFNN, radial basis function neural networks) se abordaron por vez primera en 1988, [Broomhead and Lowe, 1988]; y recientemente han llamado mucho la atención debido a su buena capacidad de generalización y una estructura de red simple que evita cálculos innecesarios y largos en comparación con la red de múltiples capas hacia adelante (MFN -Multilayer feed-forward network). Investigaciones anteriores de teoremas de aproximación universal en RBF han demostrado que cualquier función no lineal sobre un conjunto compacto con precisión arbitraria puede ser aproximada por la red neuronal con RBF ([Hartman et al., 1990], [Park and Sandberg, 1991]).

En su forma general, la red neuronal con función de base radial RBFNN consta de tres capas: la capa de entrada, la capa oculta y la capa de salida. Las neuronas en la capa oculta son activadas por una función de base radial. La capa oculta consiste en una matriz de unidades informáticas llamadas nodos ocultos. Cada nodo oculto contiene un vector \mathbf{c} central que es un vector de parámetros de la misma dimensión que el vector de entrada \mathbf{x} ; la distancia euclidiana entre el centro y el vector de entrada a la red \mathbf{x} se define por $\|\mathbf{x}(t) - \mathbf{c}_j(t)\|$.

La salida de la capa oculta puede producirse a través de una función de activación no lineal $\mathbf{h}_j(t)$ como sigue:

$$h_j(t) = \exp\left(-\frac{\|\mathbf{x}(t) - \mathbf{c}_j(t)\|^2}{2\sigma_j^2}\right), \quad j = 1, \dots, m \quad (6.1)$$

donde, en este contexto, σ_j es un escalar positivo llamado “ancho” (desviación estándar) y m describe el número de nodos ocultos. La capa de salida es una combinación lineal de pesos

$$y_i = \sum_{j=1}^m w_j h_j(t), \quad i = 1, \dots, n \quad (6.2)$$

donde w , es la capa de pesos de salida, n es el número de salidas mientras que la variable “y” denota la salida de la red.

Propiedades de aproximación de las redes neuronales

Se ha demostrado que la red neuronal hacia adelante es capaz de aproximar clases genéricas de funciones en conjuntos compactos dentro de cualquier tolerancia pre-especificada. Esta

propiedad de la red neuronal hacia adelante a menudo es referida como aproximación universal, mientras que las redes neuronales por si mismas a menudo son llamadas aproximadores universales.

Teorema 6.1 Teorema de Micchelli, ([*Micchelli, 1986*]). Sea $\varphi = \varphi(r)$ una función gaussiana. Sea $\{\mathbf{x}_i\}_{i=1}^N$, un conjunto de puntos distintos en \mathbb{R}^n . Entonces, la matriz de interpolación Φ de $N \times N$, cuyo elemento (i, j) es $\varphi_{ij} = \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|)$ es no singular. ■

Existe una gran clase de funciones de base radial (RBF) que son abarcadas por el teorema de Micchelli. De hecho, este teorema proporciona una base teórica para la aproximación de funciones basadas en RBF y técnicas de regresión. Específicamente, usando una RBF $\varphi = \varphi(r)$ y un conjunto finito de N puntos $\{\mathbf{x}_i\}_{i=1}^N$ en \mathbb{R}^n , el teorema mencionado asegura que siempre es posible aproximar funciones $f(x)$ en una región de puntos, utilizando una combinación lineal de RBF en la forma $\hat{f}(x) = \sum_j N\theta_j\varphi(x - x_j)$, tal que $f(x_i) = \hat{f}(x_i)$ para toda $\{\mathbf{x}_i\}_{i=1}^N$.

Teorema 6.2 Teorema de aproximación universal para RBFNN, ([*Park and Sandberg, 1991*]). Sea $\varphi(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ una función continua acotada e integrable, y supóngase que

$$\int_{\mathbb{R}^n} \varphi(x) dx \neq 0$$

entonces para cualquier función continua $f(x)$ y cualquier $\varepsilon > 0$, existe una red neuronal de RBF $NN(x)$ con N neuronas, un conjunto de centros $\{C\}_{i=1}^N$, y un ancho común $\sigma > 0$,

$$\hat{f}(x) = \sum_{i=1}^N \theta_i \underbrace{\varphi\left(\frac{x - C_i}{\sigma}\right)}_{\varphi_i(x)} = \Theta^\top \Phi(x)$$

tal que

$$\|f(x) - NN(x)\|_{L_2}^2 = \int_{\|\mathbf{x}\| \leq r} (f(x) - NN(x))^2 dx \leq \varepsilon = O\left(N^{-\frac{1}{n}}\right)$$

■

En conclusión:

- Las RBF son aproximadores universales.
- Una RBFNN depende de las distancias euclidianas entre el vector de entrada x y los centros C_i .
- La velocidad de aproximación para las RBFNN es del orden $O\left(N^{-\frac{1}{2n}}\right)$, consecuentemente, decrece exponencialmente conforme la dimensión del vector de entrada x incrementa. Este fenómeno es conocido como “curso de dimensionalidad” (debido a R.E. Bellman).

- Una red neuronal de RBF tiene la capacidad de aprendizaje y adaptación.

6.2.2. Redes neuronales *wavelet* (ondaleta)

Las redes neuronales *wavelet* (*wavelet neural networks*, *WNN*) son una estructura que combina el concepto de funciones *wavelet* –en español llamada ondaleta– con el recurso de las ANN con el objetivo de mejorar el desempeño de la identificación [Vázquez and Jurado, 2014]-[Wai and Chang, 2004]. A diferencia de las funciones de activación sigmoidales utilizadas en las ANN convencionales, en las WNN se utilizan funciones de activación *wavelet*. Las WNN poseen al mismo tiempo las ventajas de las ANN y su capacidad de aprendizaje sumada a la habilidad de una función *wavelet* para descomponer una señal. Los sistemas de control basados en WNN, han sido adoptados para el control de sistemas dinámicos complejos debido a sus propiedades de aprendizaje rápido y su capacidad de convergencia [Ho et al., 2001] y [Wai and Chang, 2004].

Funciones ondaleta

Las ondaletas son un descubrimiento relativamente nuevo en las matemáticas aplicadas. Deben su nombre a los trabajos de Morlet, Arens, Fourgeau y Giard (1982), Morlet (1983), Grossman y Morlet (1984), (Daubechies, 2004). El interés por las ondaletas ha crecido en las últimas dos décadas por diferentes razones. Por un lado, el concepto de ondaleta puede verse como una síntesis de ideas que se originaron en los últimos treinta años en el área de la ingeniería, de la física y de las matemáticas puras, por otro lado, las ondaletas proveen una herramienta matemática muy sencilla con una gran variedad de aplicaciones. Por ejemplo, en el análisis de señales de sonido e imagen se pueden encontrar en (Daubechies, 2004) a Kronland–Martinet, Morlet y Grossmann (1987), Mallat (1989); y en análisis numérico a Beylkin, Coifman y Rokhlin (1991). La transformada de ondaleta aparece, entonces, como una necesidad de analizar funciones no estacionarias en su escala de frecuencia–tiempo. Con base en lo anterior, se adoptará el término ondaleta para hacer referencia a la palabra *wavelet*.

El análisis ondaleta consiste en la descomposición de una señal en un conjunto jerárquico de aproximaciones y características (rasgos) que permiten representar casi fielmente a una determinada señal original. En cada nivel de jerarquía se obtiene una señal que aproxima y otra que representa los rasgos de dicha función para lo cual se toman en cuenta las frecuencias bajas que permiten aproximar la señal original mientras que la señal que muestra los rasgos corresponde a las componentes de alta frecuencia. Las ondaletas son funciones que permiten descomponer una señal en distintas componentes de frecuencia y después analizar cada una en una resolución acorde a su escala. La teoría de las funciones ondaleta fue desarrollada en distintos campos del conocimiento como matemáticas, física e ingeniería eléctrica.

Para que una función $\psi(\cdot)$ definida sobre el eje real $(-\infty, \infty)$ sea considerada una función ondaleta deberá satisfacer las siguientes propiedades [Daubechies, 1992]-[Addison, 2017]:

1. Una ondaleta debe tener energía finita:

$$E = \int_{-\infty}^{\infty} |\psi(\cdot)|^2 dt < \infty \quad (6.3)$$

donde E es la energía de una función igual a la integral de su magnitud cuadrada, las barras verticales representan el *valor absoluto*, el cual da la magnitud de $\psi(\cdot)$. Si $\psi(\cdot)$ es una función compleja, la magnitud se debe encontrar usando su parte real y compleja.

2. Si $\hat{\psi}(\cdot)$ es la *transformada de Fourier* de $\psi(\cdot)$, esto es

$$\hat{\psi}(f) = \int_{-\infty}^{\infty} \psi(t)e^{-(2\pi f)t} dt \quad (6.4)$$

la siguiente condición deberá mantenerse, *i.e.*,

$$C_\psi \equiv \int_0^\infty \frac{|\hat{\psi}(f)|^2}{f} \text{ satisface } 0 < C_\psi < \infty. \quad (6.5)$$

Esto implica que la ondaleta no tiene componente de frecuencia cero, $\hat{\psi}(0) = 0$, ó dicho de otra manera, la ondaleta ψ debe tener media cero. La ecuación (6.5) es conocida como *condición de admisibilidad* y C_ψ es llamada *constante de admisibilidad*.

3. Un criterio adicional que debe aplicarse a las *wavelets* complejas es que la transformada de Fourier debe ser real y, al mismo tiempo, también debe desvanecerse para las frecuencias negativas.

Transformada ondaleta continua

La transformada ondaleta continua (*continuous wavelet transform, CWT*, [Daubechies, 1992]), es empleada para hacer la transformación de una señal $x(\cdot)$ definida en tiempo continuo; por lo tanto, los parámetros λ y t en (6.6) se utilizan para crear una familia ondaleta que varíe en el dominio del tiempo continuo. La intención de la transformada es que, dada una dilatación λ y una traslación t sobre la función ondaleta ψ , se calcule el coeficiente de amplitud que hace de $\psi_{\lambda,t}$ la mejor aproximación sobre la señal $x(\cdot)$, *i.e.*, integrar el producto de la señal con la función ondaleta:

$$\langle x, \psi_{\lambda,t} \rangle = \int_{-\infty}^{\infty} \psi_{\lambda,t}(u)x(u)du \quad (6.6)$$

Variando el coeficiente λ la función ondaleta aproxima de mejor manera a la señal en análisis, *i.e.*, de un valor de la dilatación a otro la función puede ser reconstruida. Realizando la variación de t la naturaleza de la señal se puede desplazar en el eje del tiempo. La colección

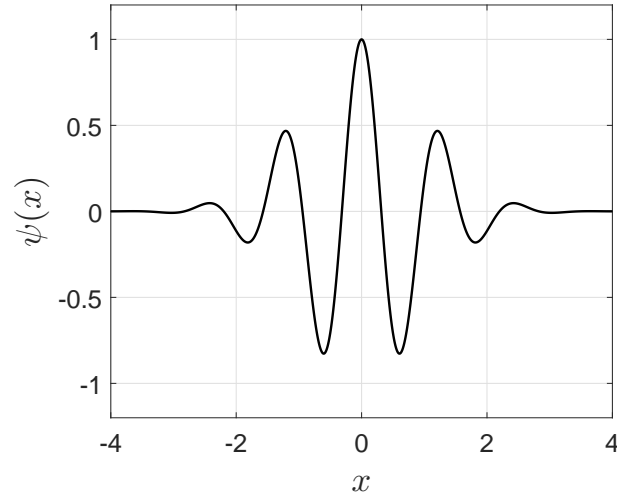


Figura 6.5: Ondaleta Morlet

de coeficientes $\{\langle x, \psi_{\lambda,t} \rangle, |\lambda > 0, -\infty < t < \infty\}$ es llamada la CWT de $x(\cdot)$. Un hecho fundamental sobre la CWT es que conserva toda la información de la señal original $x(\cdot)$. Si la función ondaleta $\psi(\cdot)$ satisface (6.6) y además si

$$\int_{-\infty}^{\infty} x^2(t)dt < \infty$$

se satisface, entonces la señal $x(\cdot)$ puede ser reconstruida a partir de su CWT aplicando la transformada inversa

$$x(t) = \frac{1}{C_{\psi}} \int_0^{\infty} \left[\int_{-\infty}^{\infty} \langle x, \psi_{\lambda,u} \rangle \psi_{\lambda,u}(u) du \right] \frac{\lambda}{\lambda^2}$$

por lo tanto $x(\cdot)$ y su CWT son dos representaciones de la misma cantidad. La CWT presenta a $x(\cdot)$ en una nueva forma que permite obtener más información que de otra manera no se podría conocer con facilidad.

Ondaleta Morlet

La función ondaleta más comúnmente utilizada es la ondaleta Morlet compleja, que se presentó por primera vez en [Kronland-Martinet et al., 1987]. Por otra parte, el valor real de la ondaleta Morlet, es la parte real de la versión compleja de la ondaleta Morlet [Addison et al., 2002], el cual tiene la forma

$$\psi = e^{-x^2/\rho} \cos(\alpha x) \tag{6.7}$$

con parámetros ρ y α representando la expansión y dilatación, respectivamente. La figura 6.5 muestra la representación gráfica de una función ondaleta Morlet.

6.3. Control por modos deslizantes basado en aproximación neuronal adaptable

En esta sección, de forma similar a como se hizo en el capítulo 5, se describe al sistema *ballbot* como un sistema no lineal subactuado de cuarto orden y se sigue la misma estrategia para desacoplar el sistema en dos subsistemas de segundo orden conectados mediante una variable intermedia.

6.3.1. Descripción del sistema subactuado

Sean las variables de estado $x_1 = q_1, x_2 = \dot{q}_1, x_3 = q_2$ y $x_4 = \dot{q}_2$; donde q_1 , describe el desplazamiento angular de la vertical de cuerpo; \dot{q}_1 , la correspondiente velocidad angular; q_2 , el desplazamiento angular del motor de la llanta que acciona a la bola (esfera) y \dot{q}_2 , su correspondiente velocidad angular (figura 2.1). Entonces es posible considerar el *ballbot* como un sistema no lineal subactuado, con incertidumbres paramétricas y perturbaciones externas descrito por:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_1(\mathbf{x}) + g_1(\mathbf{x})u + d_1(t) && \text{sistema A} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= f_2(\mathbf{x}) + g_2(\mathbf{x})u + d_2(t) && \text{sistema B} \end{aligned} \tag{6.8}$$

donde $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^\top$ es el vector de estados, $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, $g_1(\mathbf{x})$ y $g_2(\mathbf{x})$ son funciones no lineales desconocidas y se supone que se conoce la cota máxima de $|g_1(x)| \leq g_1$; u es la entrada de control y $d_1(t)$ y $d_2(t)$ son perturbaciones externas, *i.e.* $|d_1(t)| \leq D_1(t)$, $|d_2(t)| \leq D_2(t)$ y con variaciones de los parámetros del sistema por separado.

Problema de control

El problema de control consiste en desacoplar el sistema (6.8) en dos subsistemas de segundo orden, para entonces diseñar una superficie deslizante para cada uno, incluyendo una variable intermedia, con el propósito de interconectar ambos subsistemas de tal manera que sea posible diseñar un control por modos deslizantes basado en aproximación neuronal adaptable, tal que se garantice la estabilidad y el acotamiento de todas las señales en lazo cerrado.

6.3.2. Diseño del controlador

La idea de desacoplar el sistema subactuado (6.8) ha sido originalmente presentada en [Lo and Kuo, 1998]. Con esta estrategia el sistema (6.8) es desacoplado en dos subsistemas

de segundo orden y sus correspondientes superficies deslizantes son:

$$s_1 = c_1(x_1 - z) + x_2, \quad (6.9)$$

$$s_2 = c_2x_3 + x_4 \quad (6.10)$$

con las constantes $c_1, c_2 > 0$ y una función intermedia z que conecta a los dos subsistemas, la cual se define como:

$$z = \text{sat} \left(\frac{s_2}{\Phi_z} \right) z_u \quad (6.11)$$

donde z es proporcional a s_2 , transferida desde s_2 y $\Phi_z > 0$, es la capa límite para s_2 empleada para suavizar z . El valor de Φ_z , coloca la superficie s_2 en un rango adecuado para x_1 . La función saturación es definida como en (5.9). Debido a que la señal de control u es empleada para controlar a todo el sistema, el acotamiento de x_1 se puede garantizar considerando

$$|z| < z_u, \quad 0 < z_u < 1 \quad (6.12)$$

Puesto que z_u es un factor menor a la unidad, entonces z es una señal oscilatoria decreciente.

La idea general de esta estrategia de control es desacoplar el sistema completo (6.8) en dos subsistemas A y B . El sistema A contiene los estados x_1, x_2 y su correspondiente superficie deslizante sería $s_1 = c_1x_1 + x_2$. El sistema B contiene x_3, x_4 y su correspondiente superficie deslizante sería $s_2 = c_2x_3 + x_4$. Al hacerlo, el principal objetivo de control es lograr que los estados del sistema A se muevan hacia la superficie $s_1 = 0$ y así los estados x_1, x_2 convergen a $x_1 = x_2 = 0$ de manera asintótica, mientras que un sub-objetivo es mover los estados del sistema B hacia la superficie $s_2 = 0$ y que x_3, x_4 , se deslicen de manera asintótica a $x_3 = x_4 = 0$.

Puesto que el principal objetivo es estabilizar el sistema A , es razonable considerar la información del sistema B como secundaria y esta información secundaria debe reflejarse en el primer sistema de alguna manera. La variable intermedia z , que representa la información secundaria es incorporada dentro de s_1 y por lo tanto la superficie s_1 es modificada para tomar la forma (6.10). Esta modificación de s_1 refleja el hecho de que el principal objetivo es cambiado de $x_1 = 0, x_2 = 0$ a $x_1 = z, x_2 = 0$ donde z es una función de s_2 . Notando que el sub-objetivo del sistema B , que $s_2 = 0$ está contenido en el principal objetivo a través de la variable z .

Red neuronal con RBF

Considérese una red neuronal de función de base radial (RBFNN) con una capa de entrada y una capa de salida.

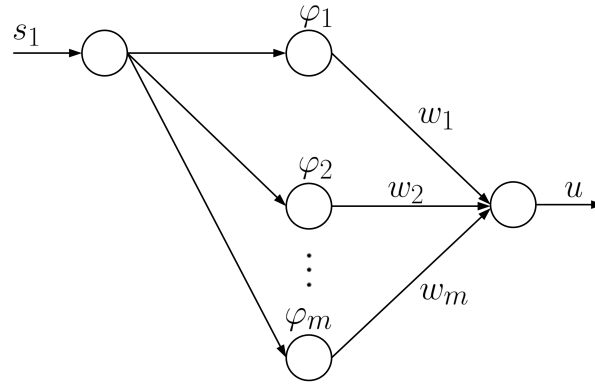


Figura 6.6: Estructura de la red neuronal de RBF

La salida de la capa oculta es generada a través de una función de activación no lineal φ_i de la forma

$$\varphi_i(t) = \exp\left(-\frac{\|\mathbf{x}(t) - \mathbf{c}_i(t)\|^2}{\sigma_i^2}\right) \quad (6.13)$$

con $i = 1, \dots, m$, siendo m el número de nodos ocultos; c_i es el punto central por nodo i ; la desviación estándar (ancho) σ es un parámetro de elección libre y determina el ancho de la función base alrededor de los centros c_i . La capa de salida es una combinación lineal de los pesos, *i.e.*,

$$\mathbf{y}_j(t) = \sum_{i=1}^m w_{ij} \varphi_i \quad (6.14)$$

con $j = 1, \dots, n$, siendo n el número de salidas; w son los pesos de la capa de salida. La red neuronal con función de base radial (RBFNN) realiza la función de mapeo hacia adelante. Debido a su mecanismo de ajuste en línea, los valores iniciales de los pesos pueden ser iniciados en cero.

La estructura empleada para la red de función de base radial (RBFNN) utilizada se muestra en la figura 6.6, con señal de entrada s_1 y señal de salida dada por:

$$u = \sum_{i=1}^m w_i \varphi_i = \mathbf{w}^\top \boldsymbol{\varphi}(s_1) \quad (6.15)$$

donde $\mathbf{w}^\top = [w_1 \ w_2 \ \dots \ w_m]$ es el vector de pesos; $\boldsymbol{\varphi}(s_1) = [\varphi_1(s_1) \ \varphi_2(s_1) \ \dots \ \varphi_m(s_1)]^\top$, es el vector de funciones base. Por lo tanto, las funciones base empleadas en la capa oculta son funciones gaussianas definidas como:

$$\varphi_i(s_1) = \exp\left(-\frac{\|s_1 - c_i\|^2}{\sigma_i^2}\right).$$

Ley adaptable para actualización de los pesos

Para obtener la ley de adaptación de los pesos, de acuerdo a la teoría de Lyapunov, es necesario considerar la condición de deslizamiento $s_1 \dot{s}_1 < 0$, estudiada en el capítulo 4; si la entrada de control satisface esta condición, entonces las trayectorias del sistema convergen al origen del plano de fase y el control por modo deslizante se puede realizar, [Slotine and Li, 1991]. La RBFNN es utilizada para mapear la superficie deslizante y la entrada del sistema, los pesos de RBFNN se adaptan ajustando la condición de alcance. La elección de la función de ajuste de pesos esta dada por ([Su et al., 2017]):

$$\dot{s}_1 + \beta s_1 = 0 \quad (6.16)$$

con $\beta > 0$. Si β se selecciona demasiado grande, la velocidad de alcance a la superficie deslizante es más rápida; sin embargo, valores grandes podrían provocar grandes sobreimpulsos; por lo tanto, debe ser seleccionada adecuadamente de acuerdo a los requerimientos del sistema. Con el objetivo de diseñar una ley adaptable para el ajuste de los pesos se considera la siguiente función costo:

$$E = \frac{1}{2}(\dot{s}_1 + \beta s_1)^2 \quad (6.17)$$

la cual es minimizada por el método de *gradiente descendiente*

$$\dot{w}_i = -\eta \frac{\partial E}{\partial w_i} \quad (6.18)$$

donde $\eta > 0$ es una constante conocida como *velocidad de aprendizaje*. Utilizando la regla de la cadena, la derivada parcial $\partial E / \partial w_i$ puede escribirse como:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial u} \frac{\partial u}{\partial w_i} \quad (6.19)$$

donde

$$\begin{aligned} \frac{\partial E}{\partial u} &= \frac{d}{du} \left[\frac{1}{2}(\dot{s}_1 + \beta s_1) \right] \\ &= (\dot{s}_1 + \beta s_1) \frac{d}{du} (\dot{s}_1 + \beta s_1) \\ &= (\dot{s}_1 + \beta s_1) \frac{d}{du} \underbrace{(c_1 \dot{x}_1 - c_1 \dot{z} + \dot{x}_2)}_{\dot{s}_1} + \beta s_1 \\ &= (\dot{s}_1 + \beta s_1) \frac{d}{du} (c_1 \dot{x}_1 - c_1 \dot{z} + \underbrace{f_1(x) + g_1 u + d_1(t)}_{\dot{x}_2}) + \beta s_1 \\ &= g_1 (\dot{s}_1 + \beta s_1) \end{aligned} \quad (6.20)$$

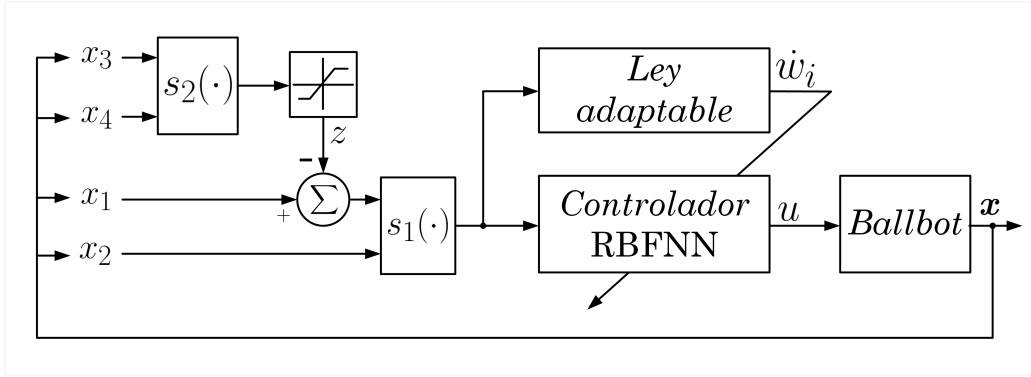


Figura 6.7: Diagrama de bloques del control por modo deslizante basado en aproximación neuronal adaptable

donde g_1 representa la cota máxima de $g_1(x)$ en (6.8). Por otra parte

$$\frac{\partial E}{\partial w_i} = \frac{\partial}{\partial w_i} \left(\sum_{i=1}^m w_i \varphi_i \right) = \varphi_i. \quad (6.21)$$

Por lo tanto, la ley adaptable puede obtenerse mediante (6.18) y (6.19), obteniendo:

$$\dot{w}_i = -\gamma(\dot{s}_1 + \beta s_1)\varphi_i \quad (6.22)$$

con $\gamma = \eta g_1$. El sistema en lazo cerrado se obtiene combinando la dinámica del *ballbot* dada en espacio de estado (6.8), con las superficies deslizantes (6.9)-(6.10), la ley de control (6.15) y la ley adaptable (6.22). La figura 6.7 muestra el diagrama de bloques del controlador por modos deslizantes basado en aproximación neuronal adaptable. El mecanismo de ajuste de los pesos para la RBFNN (6.22) puede lograr que s_1 se aproxime al origen del espacio de estado mediante (6.17) y (6.11); cuando $s_1 \rightarrow 0$, entonces $x_1 \rightarrow z$ y $x_2 \rightarrow 0$, cuando $s_2 \rightarrow 0$ entonces $z \rightarrow 0$, y por lo tanto $x_3 \rightarrow 0$ y $x_4 \rightarrow 0$. Cada subsistema logra el objetivo de control simultáneamente mediante el control por modos deslizantes desacoplado.

Análisis de estabilidad

Para el análisis de estabilidad, considérese la entrada de control ideal para el sistema A de (6.8):

$$u_{\text{eq}} = \frac{1}{g_1} [\dot{x}_2 - f_1(x) - d_1(t) + \dot{s}_1 + \lambda s_1] \quad (6.23)$$

al hacer $u = u_{\text{eq}}$ y sustituyéndola en el sistema A de (6.8) se obtiene:

$$\dot{s}_1 + \lambda s_1 = 0. \quad (6.24)$$

La salida real de la RBFNN puede expresarse en función de la diferencia entre la entrada de control equivalente y la entrada de equivalente, esto es:

$$\begin{aligned} u_{\text{eq}} &= \frac{1}{g_1} \left[\underbrace{f_1(x) + g_1 u + d_1(t)}_{\dot{x}_2} - f_1(x) - d_1(t) + \dot{s}_1 + \lambda s_1 \right] \\ g_1 u_{\text{eq}} &= g_1 u + \dot{s}_1 + \lambda s_1 \\ \dot{s}_1 &= -\lambda s_1 + g_1 [u_{\text{eq}} - u]. \end{aligned} \quad (6.25)$$

La RBFNN puede aproximar cualquier función no lineal con cierta precisión razonable. Suponiendo que los pesos óptimos existen, entonces la salida u de la RBFNN tiene un error menor que cierto ε , y alcanza a u_{eq} . Así, el error máximo de aproximación está dado por:

$$\max |u^*(\mathbf{x}, \mathbf{w}^*) - u_{\text{eq}}(\mathbf{x})| < \varepsilon \quad (6.26)$$

donde u^* es la salida óptima de la red neuronal y \mathbf{w}^* es el vector de pesos óptimos. Por lo tanto, la salida óptima es:

$$u^*(\mathbf{x}, \mathbf{w}) = \sum_{k=1}^n w_k^* \varphi_k = (\mathbf{w}^*)^\top \boldsymbol{\varphi}. \quad (6.27)$$

El error de estimación para los pesos de la red neuronal se obtiene mediante la diferencia entre los pesos óptimos y los pesos estimados, *i.e.*, $\tilde{\mathbf{w}} = \mathbf{w}^* - \mathbf{w}$. Entonces

$$u_{\text{eq}}(\mathbf{x}) = (\mathbf{w}^*)^\top \boldsymbol{\varphi} + \varepsilon. \quad (6.28)$$

Al sustituir (6.28) y (6.15) en (6.25), se obtiene:

$$\begin{aligned} \dot{s}_1 &= -\lambda s_1 + g_1 \left[\underbrace{(\mathbf{w}^*)^\top \boldsymbol{\varphi} + \varepsilon}_{u_{\text{eq}}} - \underbrace{\mathbf{w}^\top \boldsymbol{\varphi}}_u \right] \\ &= -\lambda s_1 + g_1 [\tilde{\mathbf{w}}^\top \boldsymbol{\varphi} + \varepsilon]. \end{aligned} \quad (6.29)$$

Considérese la función candidata de Lyapunov ([Su et al., 2017]):

$$V(s_1, \tilde{\mathbf{w}}) = \frac{1}{2} s_1^2 + \frac{g_1}{2\gamma} \tilde{\mathbf{w}}^\top \tilde{\mathbf{w}}. \quad (6.30)$$

Su derivada temporal es:

$$\dot{V}(s_1, \tilde{\mathbf{w}}) = s_1 \dot{s}_1 + \frac{g_1}{\gamma} \tilde{\mathbf{w}}^\top \dot{\tilde{\mathbf{w}}}. \quad (6.31)$$

Sustituyendo (6.29) y (6.22) en (6.31) y considerando el hecho de que $\dot{\tilde{\mathbf{w}}} = -\dot{\mathbf{w}}$, se tiene:

$$\begin{aligned}
 \dot{V}(s_1, \tilde{\mathbf{w}}) &= s_1 \left[\underbrace{-\lambda s_1 + g_1 (\tilde{\mathbf{w}} \boldsymbol{\varphi} + \varepsilon)}_{\dot{s}_1} \right] + \frac{g_1}{\gamma} \tilde{\mathbf{w}}^\top \left[\underbrace{\gamma (\dot{s}_1 + \beta s_1 - 1)}_{\dot{\tilde{\mathbf{w}}} = -\dot{\mathbf{w}}} \right] \\
 &= -\lambda s_1^2 + s_1 g_1 \tilde{\mathbf{w}}^\top \boldsymbol{\varphi} + s_1 g_1 \varepsilon + g_1 \left[\underbrace{\dot{s}_1 + \beta s_1}_{\text{cero}} \right] \tilde{\mathbf{w}}^\top \boldsymbol{\varphi} \\
 &= -\lambda s_1^2 + s_1 g_1 \tilde{\mathbf{w}}^\top \boldsymbol{\varphi} + s_1 g_1 \varepsilon
 \end{aligned} \tag{6.32}$$

Si los pesos son ajustados de tal manera que el error mínimo de aproximación $\tilde{\mathbf{w}}$ sea pequeño, tanto que pueda decirse que $\dot{V}(\cdot)$ ha sido minimizada, entonces el término $s_1 g_1 \tilde{\mathbf{w}}^\top \boldsymbol{\varphi} \approx 0$. Por lo tanto la derivada temporal de V se reduce a:

$$\dot{V}(s_1, \tilde{\mathbf{w}}) \leq |s_1| (-\lambda |s_1| + g_1 \varepsilon) \tag{6.33}$$

De lo anterior, cuando $|s_1| > g_1 \varepsilon / \lambda$, resulta que $\dot{V}(\cdot) < 0$, o por lo menos puede decirse que $\dot{V}(\cdot)$ ha sido minimizada. Esto significa que la función de Lyapunov es gradualmente decreciente y por lo tanto s_1 converge a la pequeña capa límite en la superficie $s_1 = 0$ del plano de fase y el error de salida del controlador disminuye cuando la precisión del mapeo no lineal de la red neuronal aumenta.

6.4. Resultados de simulación

En esta sección se presentan los resultados de simulación para la estabilización del *ballbot* empleando el controlador por modos deslizantes basado en aproximación neuronal adaptable. Las simulaciones fueron desarrolladas en MATLAB Simulink 2016a usando el integrador de Euler hacia adelante (ode1) con paso fijo a un periodo de muestreo de 0.001 [s].

6.4.1. Parámetros de diseño

El controlador por modos deslizantes basado en aproximación neuronal adaptable posee diferentes parámetros propios del diseño que se requieren para estabilizar la posición angular del cuerpo $x_1 = q_1$. El primer grupo de parámetros es el que contiene los relacionados al diseño de la red neuronal y el segundo grupo de parámetros es el que contiene los relacionados a las superficies deslizantes.

Tabla 6.1: Parámetros de simulación para los controladores neuronales basados en RBF y ondaleta

| Parámetro | Controlador neuronal | | |
|-----------|----------------------|------------|----------|
| | RBF | Ondaleta | Unidades |
| c_1 | 4.7 | 5.1 | [1/s] |
| c_2 | 1.5 | 1.5 | [1/s] |
| Φ_z | 18 | 16 | – |
| z_u | 0.3 | 0.3 | – |
| η | 100 | 20 | – |
| g_1 | 0.28058 | 0.28058 | – |
| γ | ηg_1 | ηg_1 | – |
| β | 15 | 18 | [1/s] |

Diseño de la red neuronal con RBF

Para aproximar la señal de control (6.15), se propone la arquitectura de red neuronal mostrada en la figura 6.6. La capa oculta contiene siete neuronas y son activadas con RBF, siendo el vector regresor:

$$\varphi = [\varphi_1(s_1) \quad \varphi_2(s_1) \quad \cdots \quad \varphi_7(s_1)]^T \quad (6.34)$$

Las RBF son distribuidas uniformemente con valores de centros:

$$c_1 = -3, \quad c_2 = -2, \quad c_3 = -1, \quad c_4 = 0, \quad c_5 = 1, \quad c_6 = 2, \quad c_7 = 3.$$

y ancho $\sigma_i = 0.9$ para $i = 1, 2, \dots, 7$. Las siete RBF tienen la misma señal de entrada:

$$\varphi_i = \exp\left(-\frac{(s_1 - c_i)^2}{\sigma_i^2}\right) \quad i = 1, 2, \dots, 7$$

Diseño de la red neuronal con ondaleta

Como segunda alternativa, para aproximar la señal de control (6.15) se utiliza nuevamente la arquitectura de red neuronal mostrada en la figura 6.6, pero ahora la capa oculta contiene solamente dos neuronas que son activadas con funciones ondaleta, tal como se muestra a continuación:

$$\begin{aligned} \varphi_1 &= e^{-(s_1)^2/\beta_1} \cos(\lambda_1 s_1) \\ \varphi_2 &= e^{-(s_1)^2/\beta_2} \cos(\lambda_2 s_1) \end{aligned}$$

con $\beta_1 = \beta_2 = 5$, $\lambda_1 = 0.1$ y $\lambda_2 = 0.01$. La tabla 6.1 muestra los valores empleados para la simulación de ambos controladores.

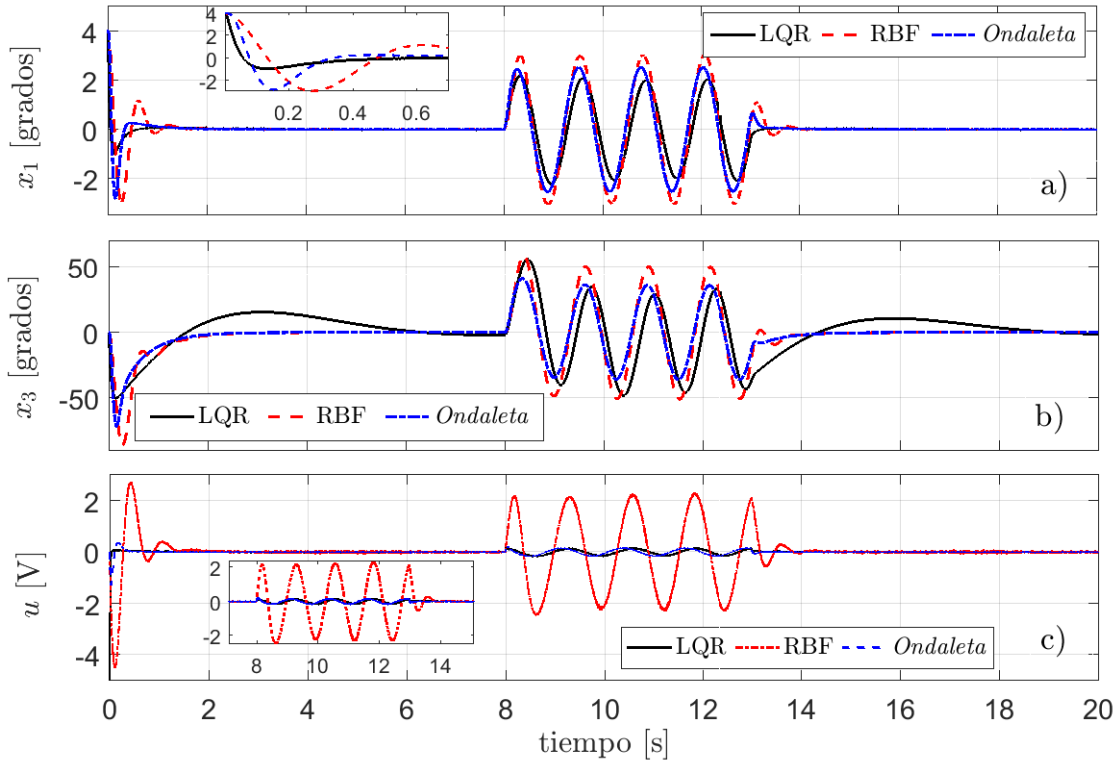


Figura 6.8: Simulación control por modos deslizantes basado en aproximación neuronal adaptable

6.4.2. Resultados

Al igual que en capítulo anterior, el desempeño de esta estrategia de control es evaluado mediante simulaciones numéricas. Los parámetros del *ballbot* son tomados de la tabla 2.3. El vector inicial de estados es $\mathbf{x}(0) = [5 \ 0 \ 0 \ 0]^\top$ y el vector de estados deseado es $\mathbf{x}_d(0) = [0 \ 0 \ 0 \ 0]^\top$. Una vez más, el objetivo de control es estabilizar el cuerpo en la vertical. Las simulaciones son desarrolladas considerando el movimiento del *ballbot* solo en un plano. Se considera también el sistema bajo incertidumbres paramétricas y perturbaciones externas, es decir, se supone que los parámetros de masa e inercia varían en un 20% de su valor nominal, tomando los valores de: $m_B = 0.736 + 0.1472 \sin(t)$ [kg], $m_b = 0.126 + 0.252 \sin(t)$ [kg], $I_B = 0.0043 + 0.00086 \sin(t)$ [kgm²], $I_b = [5.6784 + 1.1357 \sin(t)] \times 10^{-6}$ [kgm²], $I_M = [2.0 + 0.4 \sin(t)] \times 10^{-5}$ [kgm²]; una perturbación externa es considerada con valor $d_1(t) = 1.6 \cos(5t)$ más ruido blanco aplicado en la medición del ángulo del cuerpo. La figura 6.8 muestra los resultados de simulación de la estrategia de control estudiada en este capítulo basada en aproximación neuronal adaptable, la cual, es contrastada con una estrategia de control lineal LQR.

En la figura 6.8a se observa que ambas estrategias de control (–lineal y neuronal adaptable–) logran estabilizar el cuerpo del *ballbot* en su vertical hacia arriba en un tiempo relativamente corto. La figura 6.8b muestra la respuesta del desplazamiento angular del motor de la llanta; en la que se observa que el control basado en aproximación neuronal adaptable con funciones ondaleta, presenta mejor respuesta en comparación con el control lineal LQR basado en el modelo. Esto significa el *ballbot* tendrá un menor desplazamiento en el plano $x - y$, ya que el torque de la llanta influye sobre el desplazamiento de la bola (esfera) que soporta al *ballbot*. Finalmente en la figura 6.8c se muestra la señal de control demandada por el motor para lograr la estabilización del cuerpo. Es de resaltar que, a pesar de que el sistema está sometido a cierto nivel de ruido, la respuesta de las señales de control de los dos esquemas basados en aproximación neuronal adaptable es robusta ante la presencia de ruido.

6.5. Comentarios

- (i) En este capítulo se presentó una estrategia de control no lineal por modos deslizantes basada en aproximación neuronal adaptable, la cual es una alternativa de solución al problema de control de sistemas subactuados que presenten incertidumbres paramétricas y dinámicas no modeladas, manteniendo su estabilidad y buen desempeño.
- (ii) Aunque existen cuatro funciones desconocidas en el sistema bajo consideración (ecuación 6.8), únicamente es necesario conocer la cota de una de ellas para la aproximación de la señal de control; esto facilita bastante la implementación del algoritmo de control propuesto.
- (iii) El uso de funciones ondaleta como funciones de activación de la red neuronal que aproxima a la señal de control necesitó solamente dos neuronas en comparación con las siete neuronas utilizadas en la red neuronal con RBF, mostrando un mejor desempeño.
- (iv) Finalmente, el controlador diseñado en este capítulo muestra las ventajas de su estructura simple, rápida respuesta y robustez.

Resultados experimentales

En este capítulo se presentan los resultados experimentales obtenidos para la estabilización del prototipo *ballbot* construido con la tecnología LEGO MINDSTORMS EV3. La configuración del *ballbot* (figura 2.2) está inspirada en un trabajo previo basado en la tecnología LEGO MINDSTORMS NXT ([Fong et al., 2009]). Los actuadores del robot son motores de corriente directa que reciben un voltaje como señal de control. Las posiciones angulares de los motores y del gyro-sensor que mide la posición vertical del cuerpo se envían a la computadora interna del LEGO MINDSTORMS EV3 incorporada en la misma estructura del *ballbot*, el cual ejecuta el algoritmo de control previamente cargado mediante Simulink 2018b® en una computadora huésped con un período de muestreo de 4 ms.

Tabla 7.1: Parámetros experimentales para control neuronal con RBF.

| Parámetro | Control neuronal con RBF | | |
|-----------|--------------------------|------------|----------|
| | plano x | plano y | Unidades |
| c_1 | 5.8 | 5.8 | [1/s] |
| c_2 | 2.5 | 3 | [1/s] |
| Φ_z | 87 | 87 | – |
| z_u | 0.4 | 0.3 | – |
| η | 16 | 16 | – |
| g_1 | 0.28058 | 0.28058 | – |
| γ | ηg_1 | ηg_1 | – |
| β | 25 | 25 | [1/s] |

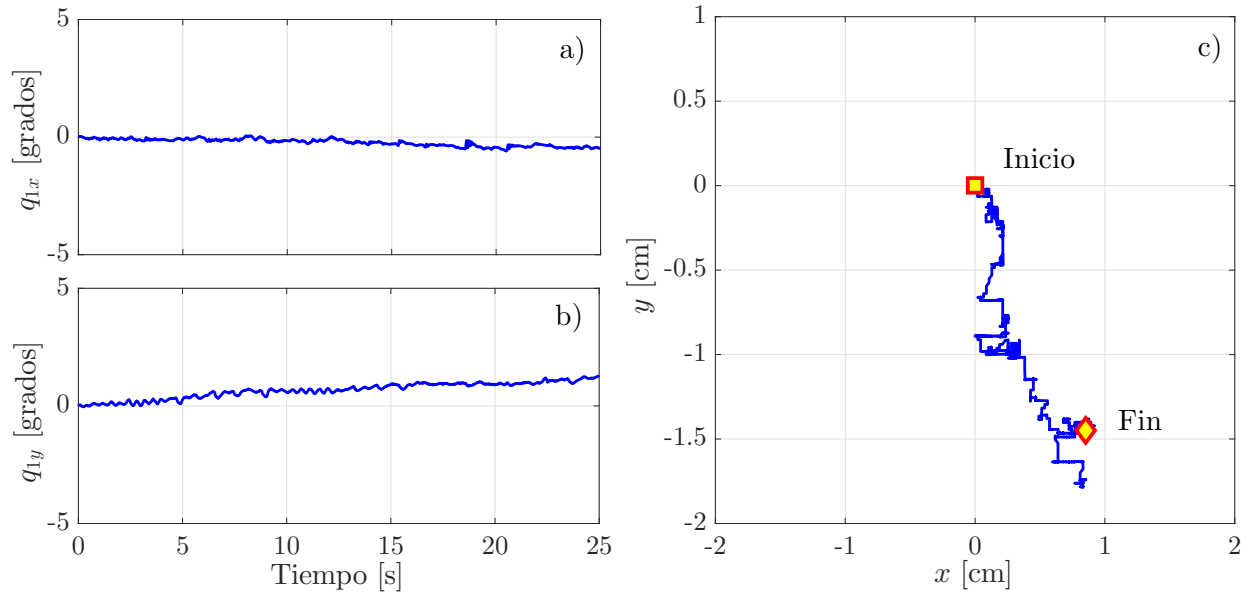


Figura 7.1: Resultados experimentales del controlador basado en RBFNN: a) desplazamiento angular q_{1x} del cuerpo; b) desplazamiento angular q_{1y} del cuerpo; c) posición en el plano x - y

7.1. Control basado en aproximación neuronal con RBF's

En la tabla 7.1, se describen los parámetros seleccionados para la ejecución del experimento del controlador basado en aproximación neuronal adaptable con RBF. Para este experimento, en la capa oculta de la red neuronal se proponen nueve neuronas activadas con RBF. Como ya se mencionó, el objetivo de control es mantener el cuerpo del *ballbot* estable en su posición vertical hacia arriba. La figura 7.1a y 7.1b muestra el desplazamiento angular de la vertical del cuerpo en los planos x - y , respectivamente, siendo cero grados el ángulo de la posición vertical hacia arriba del cuerpo. Por otra parte la figura 7.1c, muestra la posición de la bola (esfera) en el plano x - y . En las figuras 7.2a y 7.2b se muestra la señal de control aplicada a cada motor en su plano correspondiente. Es interesante destacar que el nivel de voltaje está muy por debajo del voltaje máximo permitido $|u_{\text{máx}}| \leq 9$ [V].

Respecto a los parámetros asociados a la red neuronal, en las figuras 7.3a y 7.3b se muestran las evoluciones en el tiempo de los pesos adaptables de cada red neuronal por plano, donde se observa que estos permanecen acotados. Por último en las figuras 7.3c y 7.3d se muestra la evolución de la superficie deslizante s_1 por cada plano, comprobando que se mantiene alrededor del origen del espacio de estado, tal como era de esperarse.

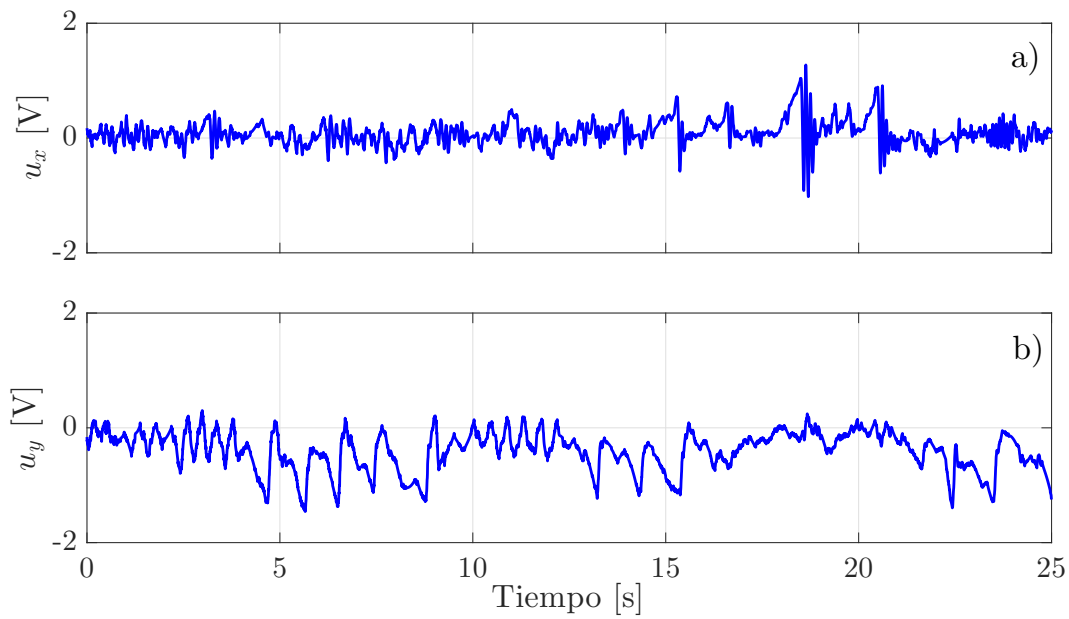


Figura 7.2: Señales de control experimentales por cada plano para control basado en RBFNN

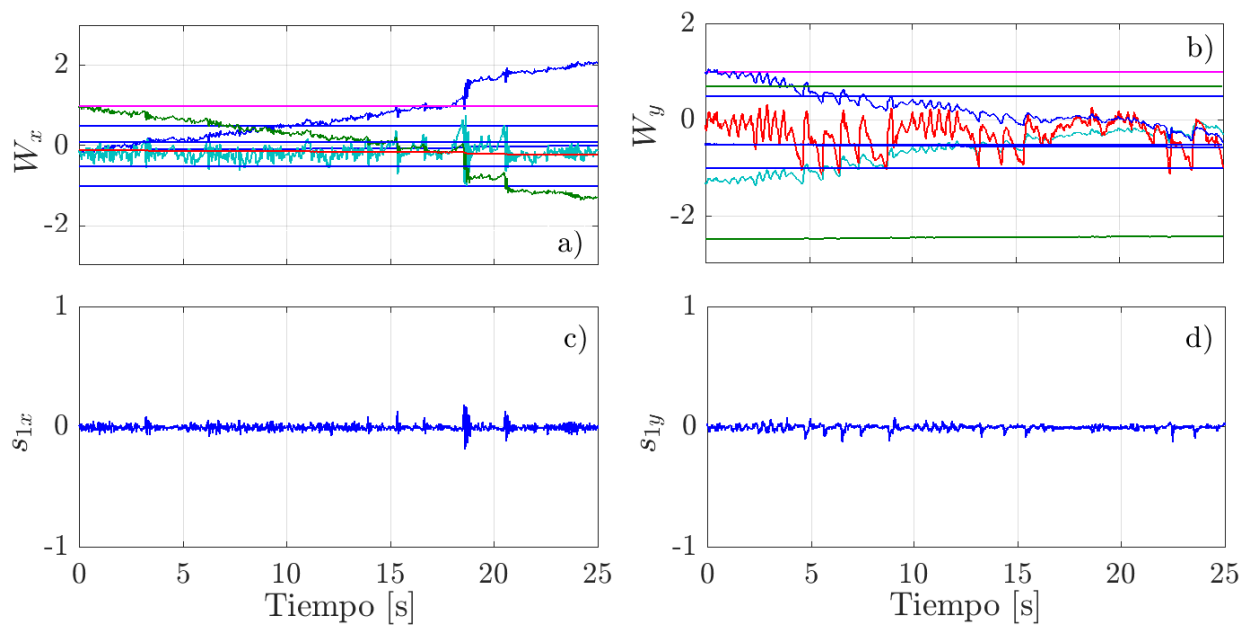


Figura 7.3: Evolución de los pesos adaptables y superficie deslizante s_1 por cada plano para control basado en RBFNN

7.2. Control basado en aproximación neuronal con funciones ondaleta

En la tabla 7.2, se describen los parámetros seleccionados para la ejecución del experimento. Para este experimento, en la capa oculta de la red neuronal se proponen dos neuronas activadas con funciones ondaleta. Como ya se mencionó, el objetivo de control es mantener el cuerpo del *ballbot* estable en su posición vertical hacia arriba. La funciones ondaleta utilizadas son:

$$\varphi_1(s_1) = e^{-s_1^2/\rho_1} \cos(\alpha_1 s_1) \quad (7.1)$$

$$\varphi_1(s_2) = e^{-s_1^2/\rho_2} \cos(\alpha_2 s_1) \quad (7.2)$$

con parámetros $\rho_1 = 35$, $\rho_2 = 5$ y $\alpha_1 = 0.1$ y $\alpha_2 = 0.01$.

Tabla 7.2: Parámetros experimentales para control neuronal con funciones ondaleta

| Parámetro | Control neuronal con funciones ondaleta | | |
|-----------|---|------------|----------|
| | plano x | plano y | Unidades |
| c_1 | 3.5 | 3.5 | [1/s] |
| c_2 | 2 | 2.3 | [1/s] |
| Φ_z | 25 | 25 | – |
| z_u | 0.4 | 0.4 | – |
| η | 10 | 10 | – |
| g_1 | 0.28058 | 0.28058 | – |
| γ | ηg_1 | ηg_1 | – |
| β | 30 | 25 | [1/s] |

Las figuras 7.4a y 7.4b muestran el desplazamiento angular de la vertical del cuerpo en los planos x - y , respectivamente, siendo cero grados el ángulo de la vertical hacia arriba del cuerpo. Por otra parte la figura 7.4c, muestra la posición de la bola (esfera) en el plano x - y . La evolución en el tiempo de las señales de control se muestra en las figuras 7.5a y 7.5b, donde se muestra la señal de control aplicada a cada motor en su plano correspondiente. Es interesante destacar nuevamente que el nivel de voltaje se encuentra por debajo del voltaje máximo permitido.

Respecto al comportamiento de los parámetros asociados a la red neuronal basada en funciones ondaleta, en las figuras 7.6a y 7.6b se muestra la evolución en el tiempo de los pesos adaptables de cada red neuronal por plano, comprobándose que estos permanecen acotados. Por último en las figuras 7.6c y 7.6d se muestra la evolución de la superficie deslizante s_1 por cada plano, comprobando que se mantiene alrededor del origen del espacio de estado.

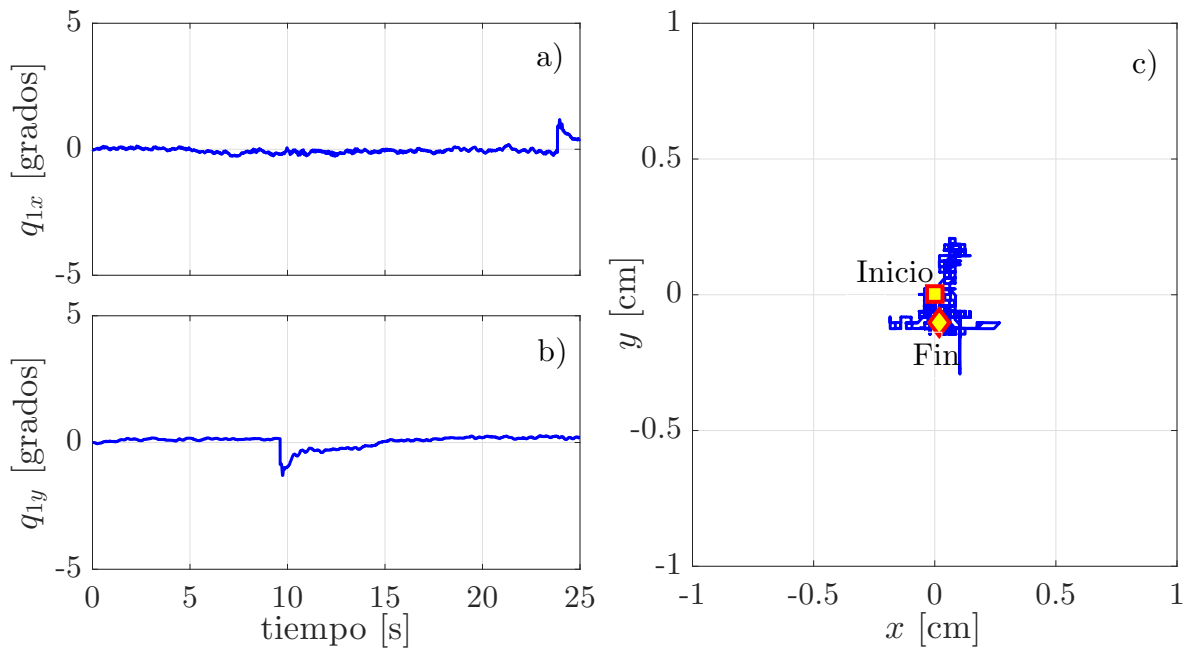


Figura 7.4: Resultados experimentales del controlador basado en funciones ondaleta: a) desplazamiento angular q_{1x} del cuerpo; b) desplazamiento angular q_{1y} del cuerpo; c) posición en el plano x - y

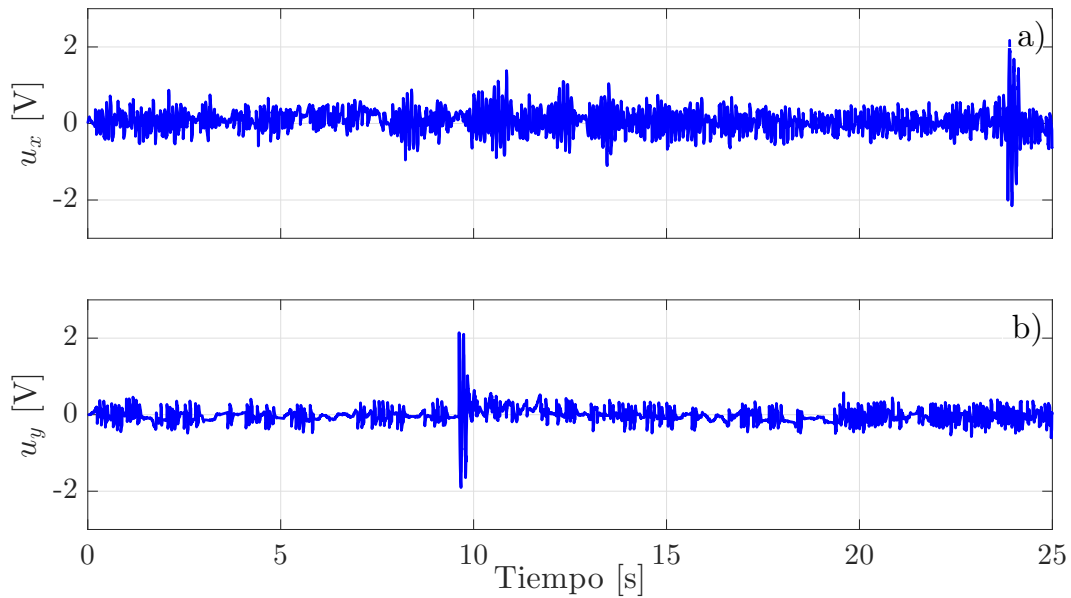


Figura 7.5: Señales de control experimentales por cada plano para control basado en red neuronal con funciones ondaleta

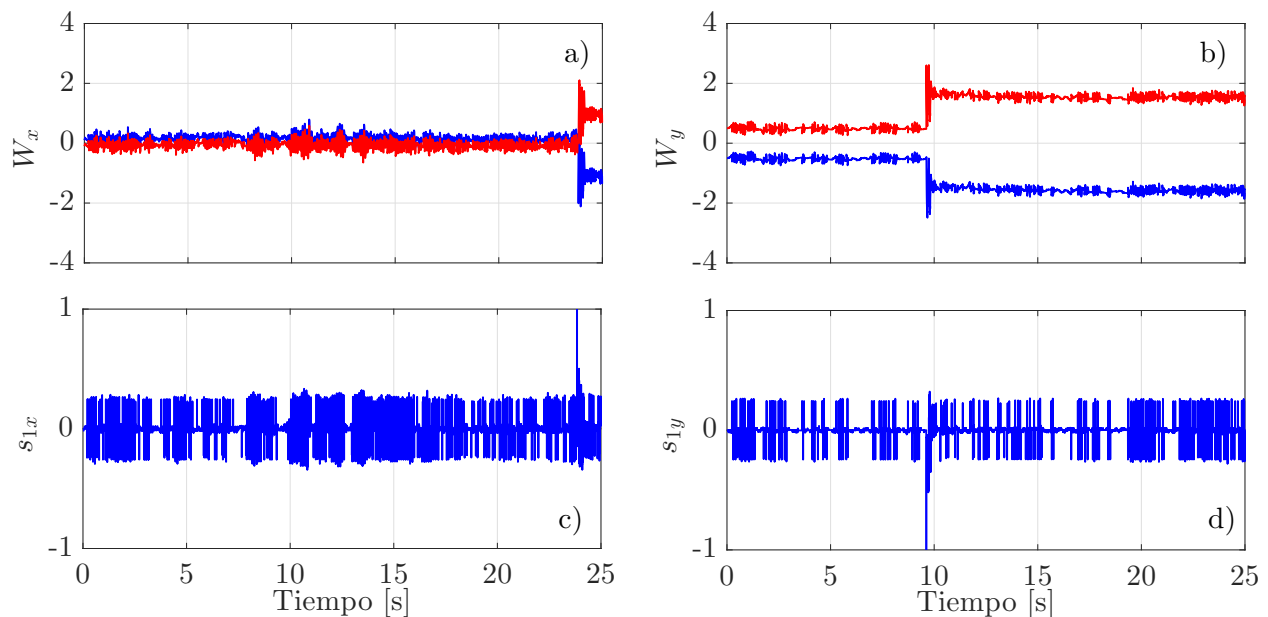


Figura 7.6: Evolución de los pesos adaptables y superficie deslizante s_1 por cada plano para control basado en red neuronal con funciones ondaleta

7.3. Control lineal basado en el modelo

Con el propósito de comparar la estrategia de control no lineal propuesta en este trabajo contra un control lineal óptimo (LQR), el cual ha sido la estrategia de control más empleada en la literatura para el control de *ballbot*, en esta sección se muestran los resultados experimentales obtenidos al aplicar tal control lineal al prototipo *ballbot* construido. El diseño de este controlador se describe en el apéndice B.

Las figuras 7.7a y 7.7b muestran el desplazamiento angular de la vertical del cuerpo en los planos x - y , respectivamente, siendo cero grados el ángulo de la posición vertical hacia arriba del cuerpo. Por otra parte, la figura 7.7c, muestra la posición de la bola (esfera) en el plano x - y .

El comportamiento de las señales de control, se muestra en las figuras 7.8a y 7.8b, donde se observa la señal de control aplicada a cada motor en su plano correspondiente. Es interesante destacar que para este caso, el nivel de voltaje se encuentra con valores cercanos al nivel de voltaje máximo permitido.

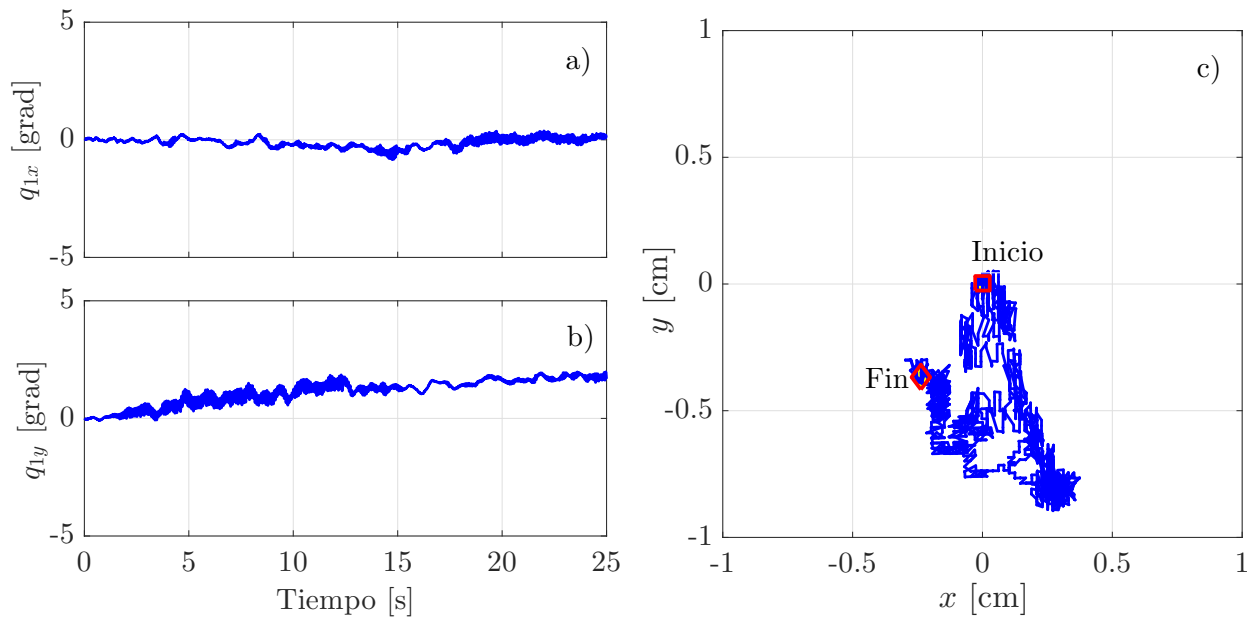


Figura 7.7: Resultados experimentales del controlador lineal LQR: a) desplazamiento angular q_{1x} del cuerpo; b) desplazamiento angular q_{1y} del cuerpo; c) posición en el plano $x-y$

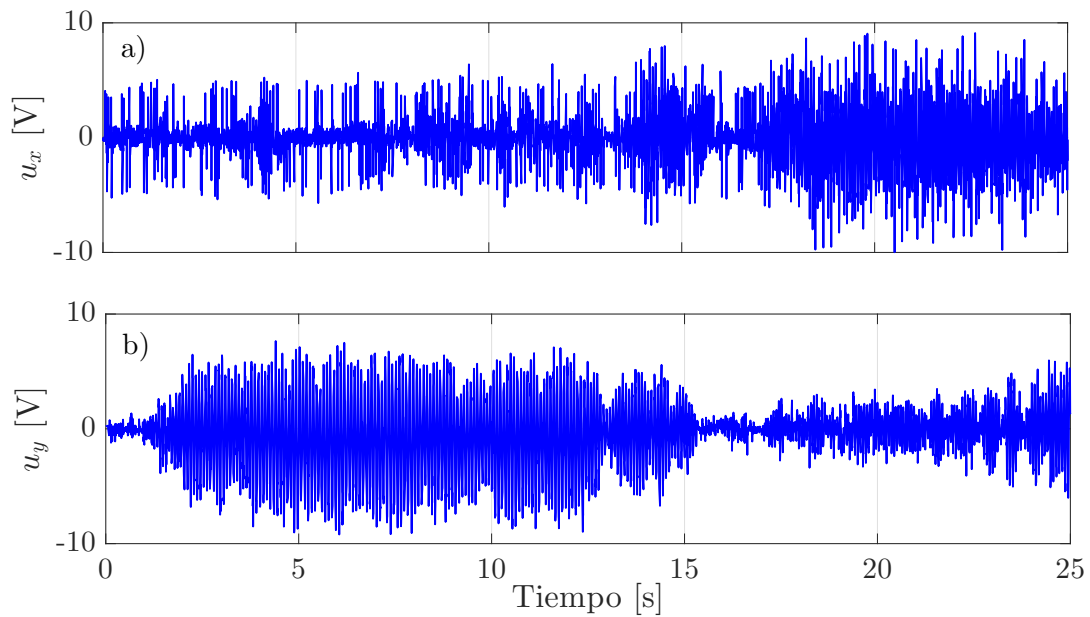


Figura 7.8: Señales de control experimentales por cada plano para el control LQR

7.4. Desempeño de los controladores

Para evaluar el desempeño de los resultados experimentales obtenidos, se empleará la norma \mathcal{L}_2 del desplazamiento angular del cuerpo, ya que el ángulo deseado es cero en la posición vertical del cuerpo ; así también, se obtendrá la norma \mathcal{L}_2 de las señales de control por cada plano. La norma \mathcal{L}_2 está dada por:

$$\mathcal{L}_2 = \sqrt{\frac{\int_0^T \nu(t)^2 dt}{T}}$$

donde $\nu(t)$ es la señal que se desea analizar y T es el intervalo de tiempo que dura el experimento.

Tabla 7.3: Norma \mathcal{L}_2 para resultados experimentales

| Controlador | \tilde{q}_{1x} | \tilde{q}_{1y} | u_x | u_y |
|--------------------|------------------|------------------|--------|--------|
| LQR | 0.7989 | 0.9596 | 1.9508 | 2.5312 |
| RBFNN | 0.2741 | 0.7425 | 0.2254 | 0.5226 |
| <i>Ondeleta</i> NN | 0.1725 | 0.2340 | 0.3227 | 0.2069 |

La tabla 7.3 muestra los valores RMS de las señales experimentales de la inclinación del cuerpo y del voltaje de control del motor en ambos planos. Los valores RMS avalan las observaciones realizadas sobre las gráficas de los desplazamientos angulares del cuerpo. Los controladores basados en aproximación neuronal adaptable muestran mejor desempeño que el controlador lineal LQR basado en el modelo. Por otra parte, el controlador basado en aproximación neuronal con funciones ondaleta muestra un mejor desempeño en comparación con el controlador basado en aproximación neuronal con RBF.

7.5. Comentarios

Se han presentado en este capítulo los resultados experimentales que permiten evaluar el desempeño del control presentado en el capítulo 6. Estos resultados son contrastados con una estrategia de control lineal ampliamente conocida y difundida en la literatura. Los experimentos se han realizado empleando MATLAB Simulink 2018b® y el prototipo *ballbot* LEGO MINDSTORMS EV3.

Algunos aspectos a destacar son los siguientes:

- (i) Para la construcción del robot *ballbot*, en un primer intento se replicó una configuración inspirada en la propuesta de [Yamamoto, 2009] (fig. 1.5a) sin embargo no se tuvo éxito en estabilizar al robot en las pruebas experimentales.

- (ii) Como segundo intento, se replicó el prototipo experimental del robot *ballbot* inspirado en la propuesta de [Fong et al., 2009], logrando obtener los resultados presentados de este trabajo de tesis.
- (iii) Se desarrolló una plataforma de comunicación entre MATLAB Simulink 2018b® y LEGO MINDSTORMS EV3 que permitió comunicar al robot *ballbot* con una computadora huésped y poder implementar los controladores propuestos.
- (iv) Se demuestra que los controladores no lineales muestran un mejor desempeño que un control lineal basado en el modelo, aún y considerando los inconvenientes de la aproximación de las señales en tiempo discreto para la realización de los experimentos.
- (v) Se demuestra que el controlador basado en aproximación neuronal con funciones ondaleta, presenta un mejor desempeño, además de requerir el menor número de neuronas en la capa oculta y por lo tanto requiere un menor costo computacional.

Conclusiones

Desde su planteamiento original, el presente trabajo de tesis se formuló con dos vertientes bien definidas. La principal de ellas está enfocada al estudio teórico y propuesta de un controlador para resolver el problema de control para la estabilización del *ballbot* mediante técnicas de control no lineal basadas en aproximación adaptable. Como resultado de ello, se logró proponer dos estrategias de control fundamentadas en el control por modos deslizantes que dan solución a este problema. La primera solución se apoya de una aproximación difuso-adaptable que aproxima dos funciones no lineales desconocidas de la ley de control, y la otra se apoya en una aproximación neuronal-adaptable que aproxima la ley de control. Por otro lado, como segunda vertiente, buena parte del tiempo de este trabajo se ha dedicado al desarrollo del prototipo del *ballbot*, así como al desarrollo de lo necesario para lograr los resultados experimentales que se presentan. Esto se refiere principalmente a la réplica y puesta a punto del prototipo experimental, así como al desarrollo del ambiente de programación en MATLAB Simulink 2018b[®] para lograr la comunicación con la tecnología LEGO MINDSTORMS EV3.

Teniendo en cuenta lo anterior, se consideran las siguientes conclusiones generales:

Respecto al control presentado en el capítulo 5, se propone una estrategia de control para estabilizar el *ballbot* inspirada en [Azimi et al., 2014]. Es importante decir que en dicho trabajo no se detalla sobre la técnica difuso adaptable utilizada, ni tampoco se describe sobre la obtención de las leyes adaptables por el método de Lyapunov. Por lo que, en este trabajo de tesis se detalla el análisis de estabilidad para la obtención de las leyes adaptables, así como la descripción de la técnica de aproximación difuso adaptable inspirada en [Llama et al., 2009]. Se demuestra mediante simulación que este controlador logra el cumplimiento del objetivo de control y es comparado con respecto a un controlador LQR. Finalmente, la implementación experimental de este controlador presentó dificultades para sintonizar los valores adecuados de las ganancias del controlador o quizás el ajuste de los parámetros relacionados al sistema

difuso adaptable, ya que se observó que este algoritmo de control presentaba mayor sensibilidad a las variaciones en el período de muestreo. Queda como tarea pendiente el sintonizar los valores adecuados para el controlador y sistema difuso de este algoritmo.

Por otra parte, se llevaron a cabo experimentos con dos algoritmos de control; uno de ellos es el neuronal adaptable, presentado en el capítulo 6, y el otro, un controlador LQR, empleado con fines comparativos y el cual se describe en el apéndice B. En la tabla 7.3 se puede observar que el controlador basado en aproximación neuronal con funciones ondaleta presenta un mejor índice de desempeño en la posición del cuerpo de 0.1725 en el plano x y de 0.2340 en el plano y ; mejorando en un 62.29% y en un 31.51% con respecto al controlador basado en aproximación neuronal con RBF, respectivamente. Los resultados obtenidos revisten de relevancia, ya que la mayoría de los controladores reportados en la literatura para estabilizar el *ballbot* son controladores lineales, mientras que los controladores no lineales propuestos en este trabajo mejoran su desempeño con respecto a los ya existentes además de contribuir como una alternativa no lineal para resolver el problema de estabilización del *ballbot*.

8.1. Problemas abiertos y trabajo a futuro

Al inicio de este trabajo de tesis los puntos mencionados en sus objetivos eran vistos como problemas abiertos. Los resultados obtenidos muestran el cumplimiento de estos objetivos y dan pauta, en base al conocimiento y experiencia adquirida, al planteamiento de nuevos problemas y mejoras al mismo, que pueden ser temas potenciales de investigación en el futuro, como lo son:

- Explorar nuevos modelos matemáticos que mejoren la descripción del comportamiento del *ballbot*.
- Mejorar la configuración del *ballbot* para darle mayor estabilidad.
- Elegir los sistemas de filtrado de señales discretas apropiados, que permitan mejorar la estimación de las señales a diferentes periodos de muestreo, siendo este un factor importante en el desempeño de los controladores.
- Buscar una mejor estrategia para sintonizar los parámetros y ganancias del controlador basado en aproximación difuso adaptable, ya que resultó difícil el encontrar sus valores y período de muestreo adecuados.
- Abordar nuevos controladores no lineales (adaptables, neuronales, modos deslizantes, etc), que permitan dar solución al problema de seguimiento de trayectorias.
- Estudiar esquemas de control basados en identificación neuronal, suponiendo que se desconoce el modelo de la planta.

8.2. Producción académica

Durante el desarrollo de esta tesis se presentaron los siguientes trabajos en congresos:

- Rubén Hernández and Francisco Jurado. “ADAPTIVE NEURAL SLIDING MODE CONTROL OF AN INVERTED PENDULUM MOUNTED ON A BALL SYSTEM”. *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. Mexico City. September 5-7, 2018.
- Leonardo Herrera, Rubén Hernández and Francisco Jurado. “CONTROL AND EXTENDED KALMAN FILTER BASED ESTIMATION FOR A BALLBOT ROBOTIC SYSTEM”. *XX Congreso Mexicano de Robótica COMRob2018*, pp: 32-37. Ensenada, Mexico. September 12-14, 2018.
- Francisco Jurado and Rubén Hernández. “DECENTRALIZED MRAC WITH INTEGRAL ACTION FOR ATTITUDE CONTROL OF A QUADROTOR UAV”. *2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC 2018)*. Ixtapa, Mexico. November, 2018.
- Rubén Hernández and Francisco Jurado. “ADAPTIVE FUZZY SLIDING MODE CONTROL OF AN INVERTED PENDULUM MOUNTED ON A BALL SYSTEM”. *2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*. Guadalajara, Mexico. November 7-9, 2018.

Matlab-Simulink-LEGO EV3

A.1. Plataforma experimental

Los ambientes Matlab 2018b® y Simulink 2018b®, contienen herramientas gratuitas para programar y comunicarse con la tecnología LEGO MINDSTORMS EV3. Las siglas “EV” hacen referencia a la palabra en inglés “evolution”, y el número 3, representa la generación del set.

El cerebro del kit LEGO MINDSTORMS EV3 es el llamado “ladrillo” inteligente, que contiene un microcontrolador ARM9 de 300 MHz, el cual reúne la información proveniente de los sensores, la procesa y envía comandos a los actuadores. El “ladrillo” puede comunicarse con computadoras, teléfonos inteligentes y otros robots gracias a su conectividad USB, bluetooth y wifi. Este cerebro tiene 16MB de memoria flash y 64MB de memoria RAM y un sistema operativo Linux de código abierto.

Para el control del prototipo *ballbot*, se utilizan los bloques de Simulink llamados “motor” y “encoder” (sensor de rotación). Estos bloques se encuentran disponibles en el soporte que ofrece Simulink para LEGO MINDSTORMS EV3. El valor de entrada al bloque de Simulink llamado “motor” está entre -100 y +100, es decir, que el voltaje de entrada que acepta cada motor (± 9 [V]) es traducido a un ciclo de trabajo del modulador de ancho de pulso (PWM) con valores entre $\pm 100\%$; el puerto de conexión se especifica en la configuración del bloque. Por otra parte, el bloque de Simulink llamado “encoder”, permite leer el desplazamiento angular del motor en [grados]. El desplazamiento angular de la vertical del cuerpo, se obtiene a través de un gyro-sensor ubicado en cada plano, el cual proporciona la velocidad angular del cuerpo en [grados/s] con un error de ± 3 [grados/s]; el desplazamiento angular de la vertical en [grados], es calculado mediante un bloque de Simulink de integración que emplea el método de Euler hacia adelante.



Figura A.1: “Ladrillo” inteligente, sensores y actuadores del kit LEGO MINDSTORMS EV3

La figura A.1 ilustra el “ladrillo”, los sensores y actuadores del kit LEGO MINDSTORMS EV3.

El soporte que ofrece Simulink para LEGO MINDSTORMS EV3 es sencillo de utilizar, ya que funciona como bloques comunes de Simulink. La figura A.2 muestra los componentes que ofrece el soporte de Simulink para el kit LEGO MINDSTORMS EV3. Cuando el código es concluído, Simulink lo traduce a código C++ y los transfiere al bloque LEGO MINDSTORMS EV3 mediante conexión: bluetooth, Wifi, USB o ethernet.

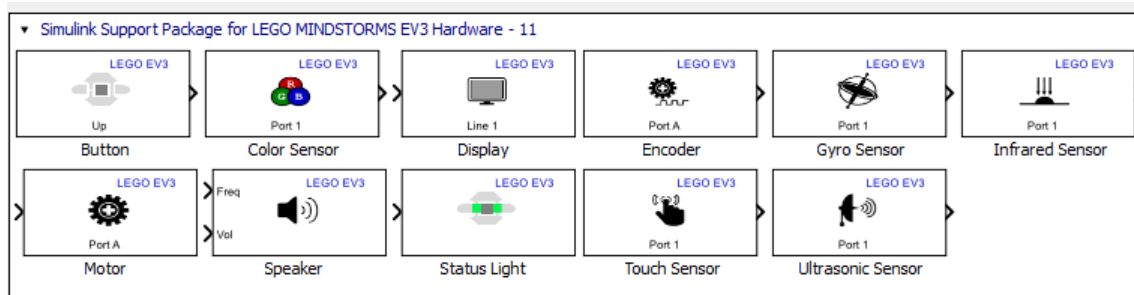


Figura A.2: Bloques de Simulink para programar el bloque LEGO MINDSTORMS EV3

Una de las principales dificultades que se presentan para lograr conexión entre el “ladrillo” LEGO MINDSTORMS EV3 y Simulink, es encontrar la versión de firmware¹ apropiada para el bloque LEGO MINDSTORMS EV3 y que sea compatible con la versión de MATLAB que se está utilizando. En este trabajo, la conexión entre el “ladrillo” LEGO MINDSTORM

¹En este trabajo se utiliza la versión EV3 20V1.08H

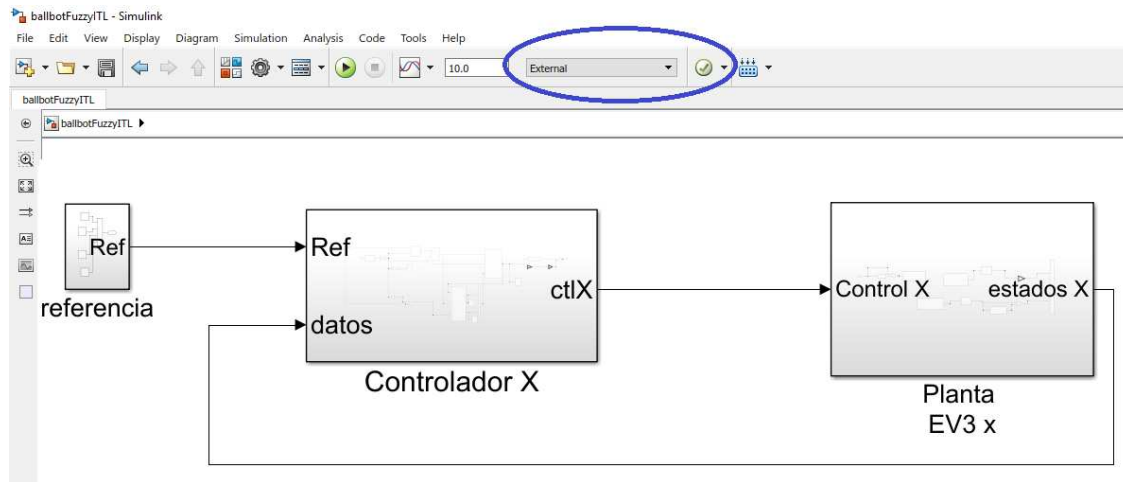


Figura A.3: Diagrama general en Simulink de controlador y planta LEGO EV3

EV3 y la computadora huésped que ejecuta Simulink, se realizó mediante una red local vía ethernet ya que fue la conexión que dio mejores resultados respecto a la conexión vía wifi. También es importante mencionar que con esta versión de Simulink y firmware de LEGO MINDSTORM EV3, no fue posible establecer una conexión vía bluetooth.

A.1.1. Ejecución en modo externo

Con respecto a la ejecución del algoritmo de control sobre el robot prototipo con LEGO MINDSTORMS EV3, algunas opciones están disponibles para el programador. Una de ellas, utilizada en el presente trabajo, es el *modo externo de Simulink*. En el modo externo en tiempo-real, Simulink Coder™, vincula dinámicamente el código del algoritmo generado con el código del controlador generado a partir de los bloques de E/S. El archivo ejecutable resultante se ejecuta en el modo de kernel del sistema operativo en la computadora huésped e intercambia datos de parámetros con Simulink a través de una interfaz de memoria compartida. En la figura A.3 se muestra la forma en que se implementa el diagrama general del controlador y la planta en Simulink; también en la parte superior de la figura A.3 se muestra encerrada en un óvalo, la opción de ejecución en modo externo.

Una vez que se tiene el diagrama de bloques del controlador y de la planta en Simulink, lo que sigue indicarle a Simulink sobre que hardware se ejecutará el programa. Esto se realiza en el panel de *configuración de parámetros* en la opción de *implementación de hardware*. En esta ventana de configuración también se especifica el tipo de conexión que se utilizará para comunicarse con el dispositivo LEGO MINDSTORMS EV3, el identificador (ID) del dispositivo LEGO MINDSTORMS EV3 y la dirección IP en el caso de conexión wifi o ethernet. En la figura A.4 se muestra una imagen de lo anteriormente descrito.

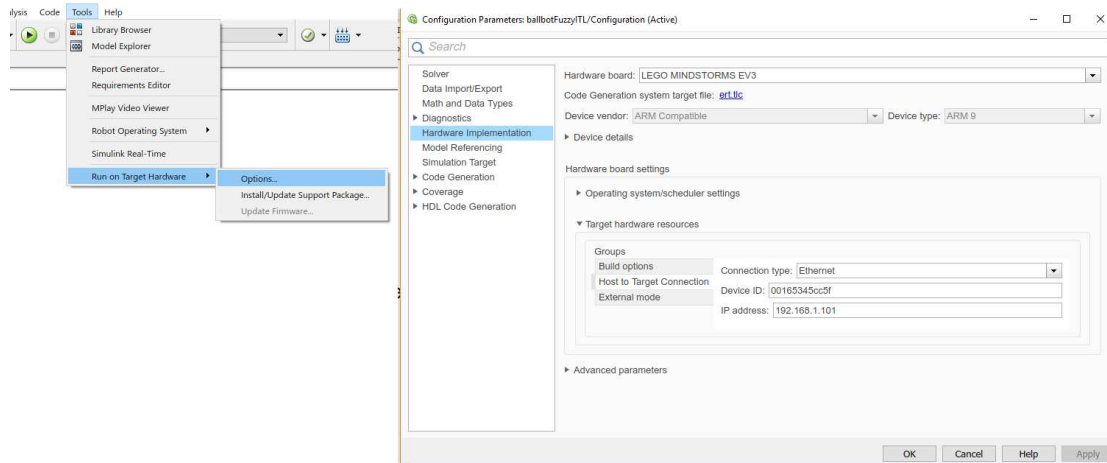


Figura A.4: Configuración de parámetros de Simulink

A.1.2. Descripción de la planta

En la figura A.5 se muestra una imagen del interior del bloque de Simulink “Planta EV3 x” (que corresponde a la planta real del *ballbot* en el plano x). Como se observa, en la entrada de este bloque se encuentra el “bloque” de Simulink correspondiente al motor conectado a un puerto del “ladrillo” LEGO MINDSTROM EV3 el cual es definido por el usuario, el motor, recibe la señal de control que proviene del bloque “Controlador X” (ver A.3). Por otra parte las señales de salida de este bloque son el desplazamiento angular del motor que proviene del bloque de Simulink llamado “encoder” y la velocidad angular proveniente del bloque gyro-sensor de Simulink.

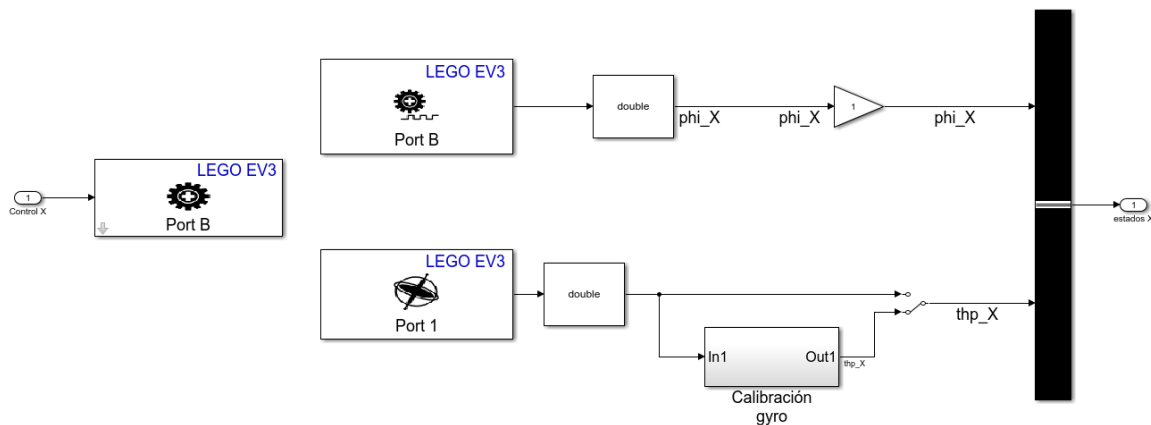


Figura A.5: Diagrama de bloque de Simulink de la planta LEGO EV3

Controlador LQR

El controlador utilizado como referencia de comparación respecto de los controladores no lineales propuestos, es un controlador lineal óptimo en espacio de estado conocido como controlador LQR (Linear Quadratic Regulator). Este controlador requiere el conocimiento de las ecuaciones de movimiento del sistema a controlar para poder representarlo en espacio de estado.

Como ya se vio, el *ballbot* es tratado como dos sistemas idénticos e independientes. Por tal motivo, únicamente el diseño en un plano es abordado en este apartado, suponiendo que el segundo plano es idéntico.

B.1. Control en espacio de estado

El control en espacio de estado es una herramienta útil para controlar sistemas de múltiples entradas y múltiples salidas (MIMO - por sus siglas en inglés-). Para el caso del *ballbot*, esto es particularmente relevante ya que se requieren varios sensores por cada plano para tener control sobre el ángulo del cuerpo y la posición del *ballbot*.

La mayor desventaja de utilizar un controlador en espacio de estado es que es un controlador lineal. Esto es, que utiliza un modelo lineal del sistema, aún si el sistema por si mismo es no lineal. Su éxito para controlar un robot del tipo *ballbot* ha sido probado en [Lauwers et al., 2005] y en [Kumagai and Ochiai, 2008].

La forma general de un controlador en espacio de estado se muestra en la figura B.1. El

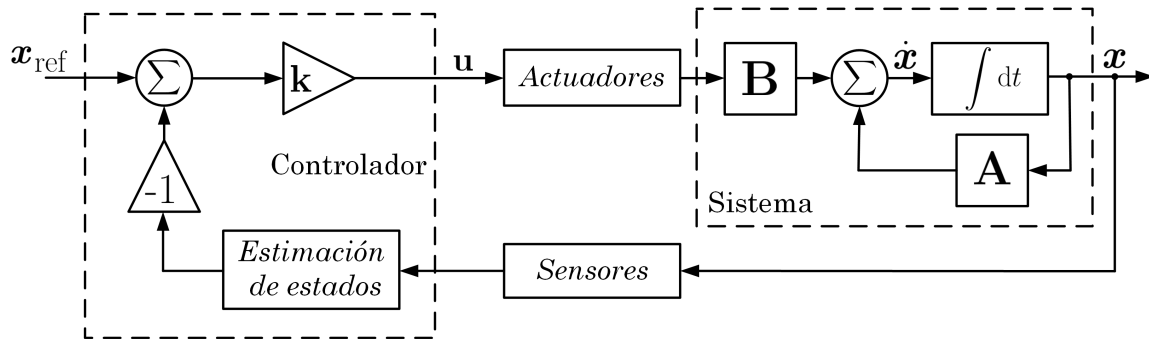


Figura B.1: Diagrama de bloques de un controlador en espacio de estado

control en espacio estado, modela el sistema como un sistema de ecuaciones diferenciales lineales, de la forma:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (\text{B.1})$$

donde \mathbf{x} representa los estados del sistema, \mathbf{u} es la señal de control, las matrices \mathbf{A} y \mathbf{B} contienen la dinámica lineal del sistema. En un controlador en espacio de estado, la señal de control \mathbf{u} se basa en la diferencia entre los estados deseados \mathbf{x}_{ref} y los estados del sistema \mathbf{x} , esto es:

$$\begin{aligned} \mathbf{u} &= \mathbf{k}(\mathbf{x}_{\text{ref}} - \mathbf{x}) \\ &= \mathbf{k}\mathbf{e}_x \end{aligned} \quad (\text{B.2})$$

donde \mathbf{k} es la matriz de ganancias. Por lo tanto, para aplicar este controlador al sistema *ballbot*, se deben de calcular las ganancias adecuadas. Además, dependiendo de los estados medidos disponibles, se puede requerir un estimador de estados para estimar cualquier estado no medible.

B.1.1. Regulador cuadrático lineal

El regulador cuadrático lineal (LQR –por sus siglas en inglés–) intenta elegir ganancias que optimicen las señales de control, basadas en un proceso heurístico. La desventaja significativa de este enfoque es que a menudo es difícil seleccionar un método heurístico cuantificable apropiado y esta elección es a menudo superficial.

La heurística desarrollada para el controlador LQR requiere una matriz \mathbf{Q} y una matriz \mathbf{R} . La matriz \mathbf{Q} es una matriz de ponderación relacionada con el desempeño. La matriz \mathbf{R} es una matriz que penaliza el uso de entradas de control más grandes.

Estas matrices se usan para encontrar la matriz de ganancia \mathbf{K} mediante la siguiente ecuación:

$$\int_t^T [\mathbf{x}^\top(\tau) (\mathbf{Q} + \mathbf{Q}^\top \mathbf{R} \mathbf{K}) \mathbf{x}(\tau)] d\tau \quad (\text{B.3})$$

El comando de MATLAB `lqr` puede usarse para resolver esta ecuación, proporcionando la dinámica lineal en espacio de estado y la selección apropiada de \mathbf{Q} y \mathbf{R} .

B.1.2. Modelo lineal en espacio de estado

Para el caso del *ballbot*, el vector de estados del sistema se define como: $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^\top$, donde:

- $x_1 = q_1$ - Es el ángulo del cuerpo del *ballbot* relativo a su eje vertical por cada plano.
- $x_2 = q_2$ - Es el ángulo del eje del motor, relativo al cuerpo por cada plano.
- $x_3 = \dot{q}_1$ - Es la velocidad angular del cuerpo del *ballbot*, por cada plano.
- $x_4 = \dot{q}_2$ - Es la velocidad angular del eje del motor, por cada plano.

y la señal de control \mathbf{u} es el voltaje del motor v , por cada plano.

Las ecuaciones de espacio de estado no lineales se linealizan utilizando una aproximación de primer orden de la siguiente manera:

$$\hat{\dot{\mathbf{x}}} = f(\bar{\mathbf{x}}, v) + J(\bar{\mathbf{x}}) \hat{\mathbf{x}} \quad (\text{B.4})$$

donde $\hat{\mathbf{x}}$ es el estado linealizado alrededor del punto de operación $\bar{\mathbf{x}} = [0 \ x_2 \ 0 \ 0]^\top$, y $J(\bar{\mathbf{x}})$ es el Jacobiano en ese punto. Tomando el estado linealizado $\hat{\mathbf{x}}$ para que sea el vector de estado resultante en el modelo de espacio de estado lineal

$$\dot{\hat{\mathbf{x}}} = \mathbf{A} \hat{\mathbf{x}} + \mathbf{B} v \quad (\text{B.5})$$

donde se tienen las siguientes matrices:

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ A_{(3,1)} & 0 & A_{(3,3)} & A_{(3,4)} \\ A_{(4,1)} & 0 & A_{(4,3)} & A_{(4,4)} \end{pmatrix} \quad (\text{B.6})$$

$$\mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ B_{(3,1)} \\ B_{(4,1)} \end{pmatrix} \quad (\text{B.7})$$

donde

$$\begin{aligned}
 A_{(3,1)} &= \frac{\mathbf{g}Lm_B(I_M + I_b\delta^2 + m_B\delta^2R_b^2 + m_B\delta^2R_b^2)}{D} \\
 A_{(4,1)} &= -\frac{\mathbf{g}Lm_B(I_M + I_b\delta + m_B\delta R_b^2 + m_B\delta R_b^2 + Lm_B\delta R_b)}{D} \\
 A_{(3,3)} &= -\frac{f_{Bg}(I_M + I_b\delta^2 + m_B\delta^2R_b^2 + m_B\delta^2R_b^2)}{D} \\
 A_{(3,4)} &= \frac{\left(f_{Bb} + \frac{k_b k_t}{R_m}\right)(I_M + I_b\delta + m_B\delta R_b^2 + m_b\delta R_b^2 + Lm_B\delta R_b)}{D} \\
 A_{(4,4)} &= -\frac{\left(f_{Bb} + \frac{k_b k_t}{R_m}\right)(I_B + I_M + I_b + L^2m_B + m_B R_b^2 + m_b R_b^2 + 2Lm_B R_b)}{D} \\
 B_{(3,1)} &= -\frac{k_t(I_M + I_b\delta + m_B\delta R_b^2 + m_B\delta R_b^2 + Lm_B\delta R_b)}{DR_m} \\
 B_{(4,1)} &= \frac{k_t(I_B + I_M + I_b + L^2m_B + m_B R_b^2 + m_b R_b^2 + 2Lm_B R_b)}{DR_m} \\
 D &= I_B I_M + I_M I_b - 2I_M I_b \delta + I_B I_b \delta^2 + I_M I_b \delta^2 + I_M L^2 m_B + I_M m_B R_b^2 + I_M m_b R_b^2 \\
 &\quad + I_b L^2 m_B \delta^2 + I_B m_B \delta^2 R_b^2 + I_M m_B \delta^2 R_b^2 + I_B m_b \delta^2 R_b^2 + I_M m_b \delta^2 R_b^2 + 2I_M L m_B R_b \\
 &\quad - 2I_M m_B \delta R_b^2 - 2I_M m_b \delta R_b^2 + L^2 m_B m_b \delta^2 R_b^2 - 2I_M L m_B \delta R_b
 \end{aligned}$$

Los valores de estos parámetros se reportan en la tabla 2.3 del capítulo 2. Los estados en este modelo reultan, en términos de la teoría de control tradicional, en un controlador proporcional-derivativo (PD). En adición, el modelo se amplía para incluir un estado extra:

- $\int q_2 = \int x_2$, la integral de la posición angular del eje del motor relativa al cuerpo

El añadir este estado extra elimina el error en estado estacionario, el cual puede ser inducido debido a perturbaciones, ruido u otras propiedades del sistema.

El corazón de este controlador en espacio de estado son las ganancias de control \mathbf{k} . Para ello se requiere la dinámica del sistema \mathbf{A} y \mathbf{B} , así como la selección de las matrices \mathbf{Q} y \mathbf{R} . La dinámica fue encontrada aplicando los parámetros definidos en la tabla 2.3. Las matrices \mathbf{Q} y \mathbf{R} se seleccionaron como:

$$\mathbf{Q} = \begin{pmatrix} 60000 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 400 \end{pmatrix}, \quad \mathbf{R} = [1000]$$

El comando `lqr` de MATLAB fue utilizado para generar las ganancias de control óptimo, \mathbf{k} , basado en la dinámica y las matrices \mathbf{Q} y \mathbf{R} anteriormente mencionadas.

$$\mathbf{k} = [99.8510 \quad -1.0181 \quad 14.9155 \quad -1.3330 \quad -0.6325] \tag{B.8}$$

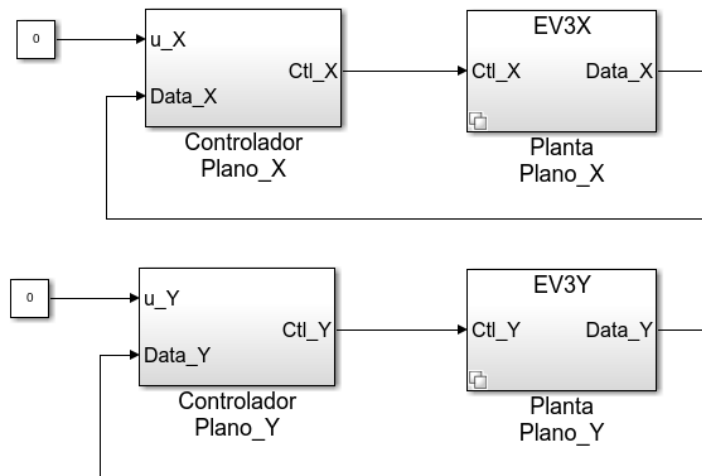


Figura B.2: Bloques en Simulink de controlador LQR en los planos $x - y$

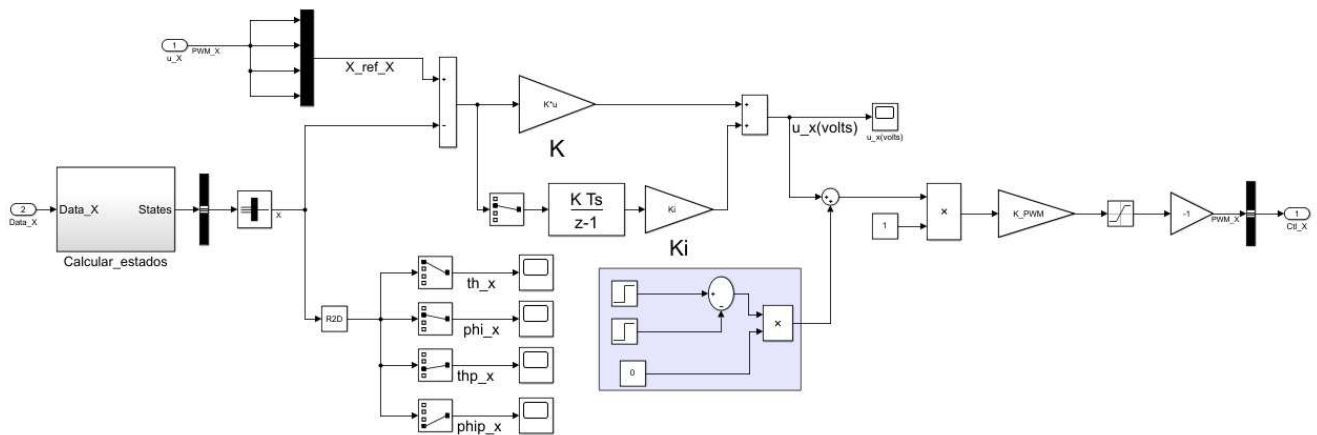


Figura B.3: Controlador LQR en Simulink en el plano x

En la figura B.2 se muestra el diagrama de bloques principal en Simulink de los controladores independientes aplicados a cada plano del *ballbot*.

Durante el proceso de prueba del controlador se ajustaron los valores de las ganancias. Este hecho se debió a que se observó un mejor desempeño al ir modificando estos valores. Con base en esto, se determinó que el siguiente conjunto de valores de ganancias dio como resultado en un mejor desempeño del sistema:

$$\mathbf{k} = [100 \quad -1.2 \quad 15 \quad -1 \quad -0.3] \quad (\text{B.9})$$

Estas ganancias son muy similares a las calculadas por el método LQR. Esto puede atribuirse a tener una buena descripción del modelo o una buena elección de las matrices \mathbf{Q} y \mathbf{R} , sin embargo, puede solo ser una coincidencia.

Bibliografía

- [Acosta and López-Martínez, 2005] Acosta, J. A. and López-Martínez, M. (2005). Constructive feedback linearization of underactuated mechanical systems with 2-dof. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4909–4914, Seville, Spain.
- [Addison, 2017] Addison, P. (2017). *The illustrated wavelet transform handbook : introductory theory and applications in science, engineering, medicine and finance*. CRC Press, Taylor & Francis, 2nd edition.
- [Addison et al., 2002] Addison, P., Watson, J., and Feng, T. (2002). Low-oscillation complex wavelets. *Journal of Sound and Vibration*, 254(4):733–762.
- [Akmal et al., 2017] Akmal, M., Jamin, N., and Ghani, N. A. (2017). Fuzzy logic controller for two wheeled ev3 lego robot. In *2017 IEEE Conference on Systems, Process and Control (ICSPC)*, pages 134–139, Malacca, Malaysia, 15-17 December.
- [Azar and Serrano, 2015] Azar, A. T. and Serrano, F. E. (2015). Adaptive sliding mode control of the furuta pendulum. In *Advances and applications in Sliding Mode Control Systems*, chapter 1, pages 1–42. Springer.
- [Azeloglu et al., 2016] Azeloglu, C. O., Sagirli, A., and Edincliler, A. (2016). Vibration mitigation of nonlinear crane system against earthquake excitations with the self-tuning fuzzy logic pid controller. *Nonlinear Dynamics*, 84(4):1915–1928.
- [Azimi et al., 2014] Azimi, M. M., Koofigar, H. R., and Edrisi, M. (2014). Stabilization of underactuated mechanical systems with time-varying uncertainty using adaptive fuzzy sliding mode. *Mediterranean Journal of Measurement and Control*, 10:252–259.
- [Ba et al., 2016] Ba, P. D., Lee, S., Back, S., Kim, J., and Lee, M. K. (2016). Balancing and translation control of a ball Segway that a human can ride. In *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, Gyeongju, Korea. October 16-19.
- [Barron, 1991] Barron, A. (1991). Approximation and estimation bounds for artificial neural

- networks. In *Proceedings of the 4th Annual Workshop on Computational Learning Theory*, pages 243–249, Santa Cruz, California, USA — August 05-07.
- [Barron, 1993] Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945.
- [Bobtsov et al., 2009] Bobtsov, A. A., Pyrkin, A. A., and Kolyubin, S. A. (2009). Adaptive stabilization of a reaction wheel pendulum on moving LEGO platform. In *2009 IEEE Control Applications, (CCA) Intelligent Control, (ISIC)*, St. Petersburg, Russia. 8-10 july.
- [Bobtsov et al., 2011] Bobtsov, A. A., Pyrkin, A. A., Kolyubin, S. A., Shavetov, S. V., Chepinskiy, S. A., Kapitanyuk, Y. A., Kapitonov, A. A., Bardov, V. M., Titov, A. V., and Surov, M. O. (2011). Using of LEGO Mindstorms NXT Technology for teaching of Basics of Adaptive Control Theory*. In *18th IFAC World Congress*, Milano, Italy. August 28 - September 2.
- [Bonci, 2016] Bonci, A. (2016). New dynamic model for a ballbot system. In *2016 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6.
- [Broomhead and Lowe, 1988] Broomhead, D. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2(3):321–355.
- [Chang et al., 2016] Chang, W.-J., Chen, P.-H., and Ku, C.-C. (2016). Variance and passivity constrained sliding mode fuzzy control for continuous stochastic non-linear systems. *Neurocomputing*, 201:29 – 39.
- [Chaudhari and Kar, 2017] Chaudhari, A. and Kar, I. (2017). Adaptive control of underactuated systems using neural networks. In *2017 Indian Control Conference (ICC)*, Guwahati, India. January 4-7.
- [Chen and Chen, 1995] Chen, T. and Chen, H. (1995). Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks. *IEEE Transactions on Neural Networks*, 6(4):904–910.
- [Chen and Huang, 2012] Chen, Y.-F. and Huang, A.-C. (2012). Controller design for a class of underactuated mechanical systems. *IET control theory & applications*, 6(1):103–110.
- [Chen and Huang, 2014] Chen, Y.-F. and Huang, A.-C. (2014). Adaptive control of rotary inverted pendulum system with time-varying uncertainties. *Nonlinear Dynamics*, 76(1):95–102.
- [Chwa, 2015] Chwa, D. (2015). Fuzzy adaptive output feedback tracking control of VTOL aircraft with uncertain input coupling and input-dependent disturbances. *IEEE Transactions on Fuzzy Systems*, 23(5):1505–1518.
- [Daubechies, 1992] Daubechies, I. (1992). *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics.

- [Donaire et al., 2017] Donaire, A., Romero, J. G., Ortega, R., Siciliano, B., and Crespo, M. (2017). Robust IDA-PBC for underactuated mechanical systems subject to matched disturbances. *International Journal of Robust and Nonlinear Control*, 27(6):1000–1016.
- [El-Hawwary et al., 2006] El-Hawwary, M. I., Elshafei, A.-L., Emara, H. M., and Fattah, H. A. A. (2006). Adaptive fuzzy control of the inverted pendulum problem. *IEEE Transactions on Control Systems Technology*, 14(6):1135–1144.
- [Flaßkamp et al., 2017] Flaßkamp, K., Ansari, A. R., and Murphey, T. D. (2017). Hybrid control for tracking of invariant manifolds. *Nonlinear Analysis: Hybrid Systems*, 25:298 – 311.
- [Fong et al., 2009] Fong, J., Uppill, S., and Cazzolato, B. (2009). *899: BALLBOT, internal report*. Bachelor’s thesis, University of Adelaide, Australia. http://data.mecheng.adelaide.edu.au/robotics/projects/2009/Ballbot/Ballbot_Final_Report.pdf.
- [García-García and Arias-Montiel, 2016] García-García, R. A. and Arias-Montiel, M. (2016). Linear Controllers for the NXT Ballbot with Parameter Variations Using Linear Matrix Inequalities. *IEEE Control Systems Magazine*, 36(3):121–136.
- [Ge et al., 2002] Ge, S., Hang, C., Lee, T., and Zhang, T. (2002). *Stable Adaptive Neural Network Control*. Springer.
- [Hamed et al., 2017] Hamed, N., Soroush, S., Sepehr, R., and Jacky, B. (2017). Position Control of the Single Spherical Wheel Mobile Robot by Using the Fuzzy Sliding Mode Controller. *Advances in Fuzzy Systems*, 2017:10 pages.
- [Hartman et al., 1990] Hartman, E. J., Keeler, J. D., and Kowalski, J. M. (1990). Layered Neural Networks with Gaussian Hidden Units as Universal Approximations. *Neural Computation*, 2(2):210–215.
- [Havasi, 2005] Havasi, L. (2005). Errosphere: an equilibrator robot. In *2005 International Conference on Control and Automation*, Budapest, Hungary. June 26-29.
- [Haykin, 1998] Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2nd edition.
- [Ho et al., 2001] Ho, D. W. C., Ping-Au, Z., and Jinhua, X. (2001). Fuzzy wavelet networks for function learning. *IEEE Transactions on Fuzzy Systems*, 9(1):200–211.
- [Huang et al., 2015] Huang, A.-C., Chen, Y.-F., and Kai, C.-Y. (2015). *Adaptive Control of Underactuated Mechanical Systems*. World Scientific Publishing Co., Inc.
- [Huang et al., 2002] Huang, S., Tan, K., and Lee, T. (2002). Adaptive motion control using neural network approximations. *Automatica*, 38(2):227–233.
- [Huda and Yu, 2015] Huda, M. and Yu, H. (2015). Trajectory tracking control of an underactuated capsbot. *Autonomous Robots*, 39(2):183–198.

- [Isidori, 1985] Isidori, A. (1985). *Nonlinear Control Systems: An Introduction*. Springer.
- [Jantzen, 2007] Jantzen, J. (2007). *Foundations of Fuzzy Control*. John Wiley & Sons.
- [Jurado and Ontiveros-Guerrero, 2017] Jurado, F. and Ontiveros-Guerrero, R. (2017). Decentralized Direct MRAC of a Self-Balancing Robot Mounted on a Ball. In *2017 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, Cuernavaca, Morelos, México. Noviembre 21-24.
- [Kelly and Santibañez, 2003] Kelly, R. and Santibañez, V. (2003). *Control de Movimiento de Robots Manipuladores*. Pearson Education, first edition.
- [Khalil, 2002] Khalil, H. (2002). *Nonlinear Systems*. Upper Saddle River: Prentice Hall, third edition.
- [Kim et al., 2016] Kim, J., Joe, H., Yu, S., Lee, J. S., and Kim, M. (2016). Time-delay controller design for position control of autonomous underwater vehicle under disturbances. *IEEE Transactions on Industrial Electronics*, 63(2):1052–1061.
- [Krener, 1973] Krener, A. (1973). On the equivalence of control systems and the linearization of nonlinear systems. *SIAM Journal on Control*, 11(4):670–676.
- [Kronland-Martinet et al., 1987] Kronland-Martinet, R., Morlet, J., and Grossmann, A. (1987). Analysis of sound patterns through wavelet transforms. *International Journal of Pattern Recognition and Artificial Intelligence*, 01(02):273–302.
- [Kumagai and Ochiai, 2008] Kumagai, M. and Ochiai, T. (2008). Development of a robot balancing on a ball. In *2008 International Conference on Control, Automation and Systems*, Seoul, South Korea, October, 14-17.
- [Lai et al., 2016] Lai, X., Wang, Y., Wu, M., and Cao, W. (2016). Stable control strategy for planar three-link underactuated mechanical system. *IEEE/ASME Transactions on Mechatronics*, 21(3):1345–1356.
- [Lauwers et al., 2005] Lauwers, T., Kantor, G., and Hollis, R. (2005). One is enough! In *Proceedings of the International Symposium for Robotics Research*, San Francisco, CA., October, 12-15.
- [Lewis et al., 1998] Lewis, F. L., Yesildirak, A., and Jagannathan, S. (1998). *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor & Francis.
- [Li et al., 2004] Li, Y., Qiang, S., Zhuang, X., and Kaynak, O. (2004). Robust and adaptive backstepping control for nonlinear systems using RBF neural networks. *IEEE Transactions on Neural Networks*, 15(3):693–701.
- [Liao et al., 2008] Liao, C.-W., Tsai, C.-C., Li, Y. Y., and Chan, C.-K. (2008). Dynamic modeling and sliding-mode control of a Ball robot with inverse mouse-ball drive. In *2008 SICE Annual Conference*, Tokio, Japan., 20-22 August. IEEE.

- [Liu and Yu, 2013] Liu, Y. and Yu, H. (2013). A survey of underactuated mechanical systems. *IET Control Theory & Applications*, 7(7):921–935.
- [Llama et al., 2009] Llama, M., Kelly, R., Santibañez, V., and Centeno, H. (2009). An adaptive fuzzy controller for robot manipulators: Theory and Experimentation. *Recent Advances in Control Systems, Robotics and Automation*, 2:109–118.
- [Lo and Kuo, 1998] Lo, J. and Kuo, Y. H. (1998). Decoupled fuzzy sliding-mode control. *IEEE Transactions on Fuzzy Systems*, 6(3):426–435.
- [Mahjoub et al., 2014] Mahjoub, S., Mnif, F., Derbel, N., and Hamerlain, M. (2014). Radial-basis-functions neural network sliding mode control for underactuated mechanical systems. *International Journal of Dynamics and Control*, 2(4):533–541.
- [McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.
- [Mehdi et al., 2014] Mehdi, N., Rehan, M., Malik, F. M., Bhatti, A. I., and Tufail, M. (2014). A novel anti-windup framework for cascade control systems: An application to underactuated mechanical systems. *ISA Transactions*, 53(3):802–815.
- [Michelli, 1986] Michelli, C. A. (1986). Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22.
- [Moreno-Valenzuela and Aguilar-Avelar, 2018] Moreno-Valenzuela, J. and Aguilar-Avelar, C. (2018). *Motion Control of Underactuated Mechanical Systems*. Springer International Publishing.
- [Narendra and Parthasarathy, 1990] Narendra, K. S. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27.
- [Nguyen and Dankowicz, 2015] Nguyen, K.-D. and Dankowicz, H. (2015). Adaptive control of underactuated robots with unmodeled dynamics. *Robotics and Autonomous Systems*, 64:84 – 99.
- [Olfati-Saber, 2001] Olfati-Saber, R. (2001). *Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles*. PhD thesis, Massachusetts Institute of Technology, Boston, MA., USA.
- [Park and Sandberg, 1991] Park, J. and Sandberg, I. W. (1991). Universal Approximation Using Radial-basis-function Networks. *Neural Computing*, 3(2):246–257.
- [Petersen et al., 2017] Petersen, C. D., Leve, F., and Kolmanovsky, I. (2017). Model predictive control of an underactuated spacecraft with two reaction wheels. *Journal of Guidance, Control, and Dynamics*, 40(2):320–332.

- [Polycarpou, 1996] Polycarpou, M. M. (1996). Stable adaptive neural control scheme for nonlinear systems. *IEEE Transactions on Automatic Control*, 41(3):447–451.
- [Pérez-Alcocer et al., 2016] Pérez-Alcocer, R., Moreno-Valenzuela, J., and Miranda-Colorado, R. (2016). A robust approach for trajectory tracking control of a quadrotor with experimental validation. *ISA Transactions*, 65:262 – 274.
- [Prieto et al., 2012] Prieto, S. S., Navarro, T. A., Plaza, M. G., and Polo, O. R. (2012). A Monoball Robot Based on LEGO Mindstorms [focus on education]. *IEEE Control Systems Magazine*, 32(2):71–83.
- [Qian et al., 2011] Qian, D., Liu, X., Yi, J., and Li, C. (2011). Neuro-hierarchical sliding mode control for a class of under-actuated systems. *International Journal of Modelling, Identification and Control*, 13(4):243–250.
- [Ramírez-Neria et al., 2016] Ramírez-Neria, M., Sira-Ramírez, H., Moctezuma, R. G., and Luviano-Juárez, A. (2016). On the linear control of underactuated nonlinear systems via tangent flatness and active disturbance rejection control: The case of the ball and beam system. *Journal of Dynamic Systems, Measurement, and Control*, 138(10).
- [Rudra et al., 2014] Rudra, S., Barai, R. K., and Maitra, M. (2014). Nonlinear state feedback controller design for underactuated mechanical system: A modified block backstepping approach. *ISA transactions*, 53(2):317–326.
- [Sanner and Slotine, 1992] Sanner, R. M. and Slotine, J. . E. (1992). Gaussian networks for direct adaptive control. *IEEE Transactions on Neural Networks*, 3(6):837–863.
- [Sastry, 1999] Sastry, S. (1999). *Nonlinear Systems*. Springer-Verlag, 1 edition.
- [Scheerer, 2006] Scheerer, E. M. (2006). Modeling dynamics and exploring control of a single-wheeled dynamically stable mobile robot with arms. Master’s thesis, Carnegie-Mellon University Pittsburgh, PA., USA.
- [She et al., 2012] She, J., Zhang, A., Lai, X., and Wu, M. (2012). Global stabilization of 2-DOF underactuated mechanical systems—an equivalent-input-disturbance approach. *Nonlinear Dynamics*, 69(1-2):495–509.
- [Slotine and Li, 1991] Slotine, J. J. and Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall.
- [Smoczek and Szpytko, 2017] Smoczek, J. and Szpytko, J. (2017). Particle swarm optimization-based multivariable generalized predictive control for an overhead crane. *IEEE/ASME Transactions on Mechatronics*, 22(1):258–268.
- [Spong, 1994] Spong, M. W. (1994). Partial feedback linearization of underactuated mechanical systems. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’94)*, volume 1, pages 314–321, Munich, Germany., 12-16 September.

- [Spong et al., 2001] Spong, M. W., Corke, P., and Lozano, R. (2001). Nonlinear control of the reaction wheel pendulum. *Automatica*, 37(11):1845–1851.
- [Ssu-Hsin and Anuradha, 1997] Ssu-Hsin, Y. and Anuradha, M. A. (1997). Adaptive control of nonlinear dynamic systems using θ -adaptive neural networks. *Automatica*, 33(11):1975 – 1995.
- [Ssu-Hsin and Anuradha, 1998] Ssu-Hsin, Y. and Anuradha, M. A. (1998). Stable neural controllers for nonlinear dynamic systems. *Automatica*, 34(5):641 – 650.
- [Su et al., 2017] Su, G. J., Gong, Q. L., and Li, H. Y. (2017). Adaptive sliding-mode control based on decoupled method for a class of underactuated system. In *36th Chinese Control Conference (CCC)*, pages 639–643, Da lian, China, on July 26 to 28.
- [Sundararajan et al., 2002] Sundararajan, N., Saratchandran, P., and Li, Y. (2002). *Fully Tuned Radial Basis Function Neural Networks for Flight Control*. Springer.
- [T. Nguyen et al., 2002] T. Nguyen, H., R. Prasad, N., and L. Walker, C. (2002). *A First Course in Fuzzy and Neural Control*. Taylor & Francis.
- [Tao et al., 2010] Tao, C. W., Taur, J., Chang, J. H., and Su, S. (2010). Adaptive fuzzy switched swing-up and sliding control for the double-pendulum-and-cart system. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(1):241–252.
- [Tsai and Tai, 2016] Tsai, C. and Tai, F. (2016). Distributed sliding-mode formation control using recurrent interval type 2 fuzzy neural networks for uncertain multi-ballbots. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Vancouver, Canada., July 24-29.
- [Van Kien et al., 2016] Van Kien, C., Son, N. N., and Anh, H. P. H. (2016). A stable lyapunov approach of advanced sliding mode control for swing up and robust balancing implementation for the pendubot system. In *AETA 2015: Recent Advances in Electrical Engineering and Related Sciences*, volume 371, pages 411–425. Springer.
- [Vázquez and Jurado, 2014] Vázquez, L. A. and Jurado, F. (2014). Continuous-time decentralized wavelet neural control for a 2 dof robot manipulator. In *2014 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pages 1–6, Ciudad de México, Septiembre.
- [Wai and Chang, 2004] Wai, R.-J. and Chang, H. H. (2004). Backstepping wavelet neural network control for indirect field-oriented induction motor drive. *IEEE Transactions on Neural Networks*, 15(2):367–382.
- [Wang, 1997] Wang, L.-X. (1997). *A Course in Fuzzy Systems and Control*. Prentice-Hall.
- [Xu and Hu, 2013] Xu, L. and Hu, Q. (2013). Output-feedback stabilisation control for a class of under-actuated mechanical systems. *IET Control Theory & Applications*, 7(7):985–996.

-
- [Yamamoto, 2009] Yamamoto, Y. (2009). *NXT ballbot Model-Based Design-Control of a Self-Balancing robot on a ball, built with LEGO Mindstorms NXT*. Internal report. url:z2.ifrm.com/10716/53/0/p1025571/NXT_Ballbot_Model_Based_Design.pdf.
- [Yue et al., 2014] Yue, M., Wei, X., and Li, Z. (2014). Adaptive sliding-mode control for two-wheeled inverted pendulum vehicle based on zero-dynamics theory. *Nonlinear Dynamics*, 76(1):459–471.
- [Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338 – 353.
- [Zhang and Ge, 2009] Zhang, T. and Ge, S. S. (2009). Adaptive Neural Network Tracking Control of MIMO Nonlinear Systems With Unknown Dead Zones and Control Directions. *IEEE Transactions on Neural Networks*, 20(3):483–497.
- [Zhang et al., 2016] Zhang, X., Huang, X., Lu, H., and Zhang, H. (2016). Forwarding-based immersion and invariance control for n-dimensional strict-feedback nonlinear systems. *Nonlinear Dynamics*, 83(1):483–496.
- [Zhu et al., 2008] Zhu, Q., Fei, S., Zhang, T., and Li, T. (2008). Letters: Adaptive RBF Neural-networks Control for a Class of Time-delay Nonlinear Systems. *Neurocomputing*, 71(16-18):3617–3624.