



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

IMPLEMENTACIÓN DE CONTROLADORES POR
MODOS DESLIZANTES DE ALTO ORDEN EN UN
ROBOT PARALELO

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

Ingeniero Eléctrico - Electrónico

PRESENTA:

Hernández Mendoza Pablo Antonio

DIRECTOR DE TESIS:

Dr. Leonid Fridman



2014

Agradecimientos

L. Fridman y R. Iriarte agradecen el apoyo de la Universidad Nacional Autónoma de México otorgado para la culminación de este trabajo a través del proyecto
PAPIIT IN113613

Agradecimientos Personales

A mis padres Guillermo y Sara por todo el apoyo que me han brindado, y todas las oportunidades que me han dado.

A mis hermanos Sandra y Guillermo por estar siempre cuando los necesitaba.

A Celia, Armando, Alfredo, Luis y Elias por su camaradería a lo largo de estos años en la Facultad.

Al PROTECO por permitirme ser parte de un grupo de trabajo excepcional.

A todo el equipo de modos deslizantes, pero especialmente a Marco por asesorarme en esta última etapa de la carrera y brindarme su amistad.

Al Dr. Fridman y al Dr. Rafael Iriarte por su consejo siempre oportuno y por compartir su conocimiento conmigo. Sin ellos no hubiera podido conocer las virtudes de este gran campo llamado control.

A la UNAM por la formación que me ha entregado y por darme la posibilidad de mejorar como persona.

Contenido

1. Introducción	1
1.1. Presentación	1
1.2. Objetivo	2
1.3. Motivación	2
1.3.1. Estado del arte	4
1.4. Planteamiento del problema	6
1.5. Metodología	7
1.6. Contribuciones	8
1.7. Estructura de la tesis	9
2. Base Teórica	11
2.1. Robots paralelos	11
2.1.1. Conceptos básicos de robótica	11
2.1.2. Precisión de un robot	13
2.1.3. Robots sobreactuados	13
2.1.4. Dinámica del robot	14
2.2. Ecuaciones de Euler-Lagrange	15
2.2.1. Formulación básica	15
2.3. Grado relativo de un sistema	18
2.4. Introducción a los Modos Deslizantes	19
2.4.1. Orden de un Modo Deslizante	21
2.5. Modos Deslizantes de Segundo Orden	22
2.5.1. Algoritmo Twisting	22
2.5.2. Algoritmo Super-Twisting	23
2.6. Modos Deslizantes de Alto Orden (HOSM)	24

3. Modelo y descripción del robot	25
3.1. Descripción de la planta	25
3.1.1. Espacio de trabajo del robot paralelo	27
3.2. Modelo matemático del robot	28
3.2.1. Cinemática del robot paralelo	29
3.2.2. Modelo dinámico: Cadena cinemática abierta	31
3.2.3. Modelo dinámico sin fuerzas de restricción	33
3.2.4. Modelo dinámico con fuerzas de restricción	34
3.2.5. Modelo en espacio de estados	36
3.3. Definición del error de seguimiento	37
3.4. Estabilidad en lazo cerrado con control PD	37
4. Controladores a utilizar	39
4.1. Objetivo de control	39
4.2. Estrategia de control	39
4.3. Control nominal	40
4.4. Control robusto ante perturbaciones	41
4.4.1. Estructura del control no lineal	41
4.4.2. Super-Twisting	42
4.4.3. Controladores por modos deslizantes de orden arbitrario	43
4.4.3.1. Controladores deslizantes Anidados	43
4.4.3.2. Controladores deslizantes Cuasi-Continuos	44
4.4.3.3. Diferenciadores exactos y robustos de orden arbitrario	45
4.4.4. Sintonización de los controladores	49
4.4.4.1. Sintonización paramétrica	49
4.4.5. Presentación de los algoritmos a programar	50
5. Simulaciones	51
5.1. Parámetro dinámicos usados	51
5.2. Resultados de simulaciones	53
5.2.1. Controlador Super-Twisting	53
5.2.2. Controlador por Modos Deslizantes de tercer orden	54
5.2.2.1. Controlador Cuasi-Continuo	54
5.2.2.2. Controlador Anidado	55

5.2.3.	Controlador por Modos Deslizantes de Cuarto Orden	57
5.2.3.1.	Controlador Cuasi-Continuo	57
5.2.3.2.	Controlador Anidado	58
6.	Experimentos en el robot	63
6.1.	Programación de controladores	63
6.2.	Resultados experimentales con trayectoria recta	64
6.2.1.	Controlador Super-Twisting	64
6.2.2.	Controlador por Modos Deslizantes de tercer orden	65
6.2.2.1.	Controlador Cuasi-Continuo	65
6.2.2.2.	Controlador Anidado	66
6.2.3.	Controlador por Modos Deslizantes de cuarto orden	68
6.2.3.1.	Controlador Cuasi-Continuo	69
6.2.3.2.	Controlador Anidado	69
6.3.	Resultados experimentales con trayectoria circular	71
6.3.1.	Controlador Super-Twisting	73
6.3.2.	Controlador por Modos Deslizantes de tercer orden	74
6.3.2.1.	Controlador Cuasi-Continuo	74
6.3.2.2.	Controlador Anidado	76
6.3.3.	Controlador por Modos Deslizantes de cuarto orden	76
6.3.3.1.	Controlador Cuasi-Continuo	77
6.3.3.2.	Controlador Anidado	77
6.4.	Cálculo de la precisión	78
6.4.1.	Error máximo	79
6.4.2.	Error de estado estable	81
6.5.	Tiempos de ejecución	82
7.	Conclusiones	85
7.1.	Comparación entre controladores	85
7.2.	Conclusiones Generales	86

CONTENIDO

A.	89
A.1. Programas	89
A.1.1. Diferenciador de Levant	89
A.1.2. Controladores HOSM	93
A.2. Hardware	96
A.2.1. Tarjeta GT-400-SV	96
A.2.2. Sistema de Servo Amplificadores PY2	97
A.2.3. Calibración de los encoders absolutos	99
Bibliografía	103

Introducción

1.1. Presentación

El control automático ha tenido un papel importante en el avance de la ingeniería y de la ciencia. La teoría de control se ocupa del entendimiento de segmentos del medio ambiente, llamados sistemas, y el control de estos en beneficio de la sociedad. Es por esto que el desarrollo de este campo ha tenido un gran impacto en la civilización moderna y en la tecnología.

Los sistemas de control se encuentran en muchos de los sectores de la industria como, por ejemplo, líneas de ensamble automático, control de máquinas eléctricas, tecnología espacial, sistemas de transporte, sistemas de potencia, sistemas robóticos, entre otros. Incluso podría decirse que cada aspecto de las actividades de nuestra vida diaria está afectado por algún tipo de sistema de control. El reto actual para los ingenieros es el modelado y control de este tipo de sistemas modernos y complejos.

Dentro de los problemas a los que se enfrentan los profesionistas especializados en este ámbito se encuentra principalmente el control de los sistemas con incertidumbres y sujetos a perturbaciones, ya que en la formulación de cualquier problema de control se presentan discrepancias entre la planta real y su modelo matemático usado para el diseño del controlador deseado. Una de las estrategias más efectivas para trabajar con este tipo de sistemas es el control por modos deslizantes.

Los modos deslizantes son robustos ante incertidumbres del modelo matemático, ya que no dependen de él, e insensibles a las perturbaciones acopladas a la señal de

control; aunque existen algunos inconvenientes en su implementación, entre ellos, insensibilidad únicamente con perturbaciones acopladas, convergencia de los estados sólo asintóticamente, diseño de la superficie de deslizamiento de grado relativo uno con respecto al control. Quizás el mayor de estos inconvenientes son las oscilaciones de alta frecuencia comúnmente conocidas como *chattering*, ya que debido a la imposibilidad de los actuadores para conmutar a una frecuencia infinita este efecto puede llegar a ser muy dañino para el sistema.

1.2. Objetivo

El objetivo de este trabajo de tesis es implementar diferentes controladores continuos utilizando la técnica de modos deslizantes, basándose en algoritmos de alto orden, para un robot paralelo sobreactuado de dos grados de libertad. Se comparará la precisión de cada uno de estos algoritmos a través de simulaciones y experimentos, tomando en consideración que el sistema está sujeto a dinámicas no modeladas, en este caso, la fricción seca de los actuadores.

1.3. Motivación

En muchos sistemas de control prácticos es importante evitar el efecto *chattering*, produciendo una señal de control continua. En el caso de la robótica, los actuadores de las uniones activas dentro de la configuración de la cadena cinemática no pueden moverse hacia delante y hacia atrás a alta frecuencia; pero al mismo tiempo es deseable mantener la robustez e insensibilidad de los sistemas de control ante las incertidumbres acotadas del modelo y perturbaciones externas.

Cuando los modos deslizantes de segundo orden aparecieron, se pensó que el problema del *chattering* había sido resuelto. Esto no es completamente cierto ya que tan sólo se logra una reducción de este efecto, pero el fenómeno de alta frecuencia sigue apareciendo. De hecho, la propiedad libre de *chattering* asociada a los algoritmos de segundo orden no ha sido alcanzada, inclusive en los casos en donde se tiene una ley de control continua [1].

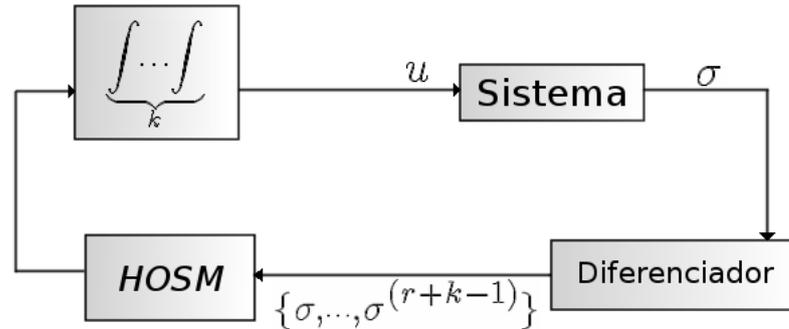


Figura 1.1: Diagrama de control de una 'caja negra'

En el 2001, se introdujo el primer controlador por modos deslizantes de orden arbitrario en [2]. Tales controladores permitían cumplir con las propiedades de un modo deslizante de orden r y compensaban exactamente las perturbaciones o incertidumbres acopladas. De igual forma en [3] se expone que, dado el grado relativo r de la salida, los controladores de alto orden o HOSM (High Order Sliding Mode) se construyen utilizando un algoritmo de recursión. Estos algoritmos proveen estabilización en tiempo finito de la variable de deslizamiento σ , $\dot{\sigma} = 0$ y, por lo tanto, de sus derivadas hasta $\sigma^{(r-1)}$.

Dado que los controladores utilizan la salida y sus derivadas sucesivas, se introdujeron en [2] los diferenciadores de orden arbitrario basados en modos deslizantes de alto orden. El uso de estos diferenciadores, en conjunto con los controladores, permitieron el diseño e implementación de algoritmos de control por salida realimentada de orden arbitrario para sistemas de una entrada y una salida, asegurando así la estabilización de la salida en tiempo finito en presencia de perturbaciones.

Este enfoque permitió resolver el problema de estabilización de la salida de un sistema parecido a una 'caja negra' [1], ver Figura 1.1.

La única información necesaria es:

- La cota superior para el grado relativo r de la planta;
- El nivel de suavidad k de la señal de control, que es tolerable para los actuadores del sistema.

1. INTRODUCCIÓN

- Si σ es conocida para ser $(r + k - 1)$ veces diferenciable, el controlador puede ser k veces diferenciable si se incluyen k integradores en la entrada de control, atenuando así el efecto *chattering*.

En la Tabla 1.1 se puede observar una recapitulación de las estrategias utilizadas en los controladores por modos deslizantes. Aunque la aplicación de los HOSM tiene muchas ventajas, uno de sus puntos en contra es que sólo se asegura exactitud asintótica, además de que las constantes para estimar la exactitud necesitan ser calculadas con base en simulaciones [3].

Algoritmo	Convergencia	Señal de Control	Información
Modo Deslizante convencional	Asintótica	Discontinua	$\sigma, \dot{\sigma}$
Super-Twisting	Asintótica	Continua	$\sigma, \dot{\sigma}$
Twisting	Tiempo finito	Discontinua	$\sigma, \dot{\sigma}$
Modo Deslizante de Alto Orden	Asintótica	Continua	$\sigma, \dots, \sigma^{(r+k-1)}$

Tabla 1.1: Diferentes estrategias de control para el sistema perturbado de segundo orden con salida σ

1.3.1. Estado del arte

En los últimos años los robots paralelos han llamado la atención dadas las ventajas que ofrecen con respecto a los manipuladores serie, ver [4]. Su utilización en diversos campos, como la medicina y las máquinas-herramienta, los ha convertido en un área de investigación y desarrollo importante; aunque con muchos problemas abiertos aún por tratar.

De igual forma la creciente demanda de los manipuladores paralelos en el sector industrial, ha permitido que entes académicos se interesen en el estudio de métodos para solucionar problemas vigentes en este tipo de sistemas; los cuales tienen que ver directamente con la complejidad en su topología. Esto es posible a través de la construcción de prototipos o equipos de ensayos que hacen más fácil el conocer su funcionamiento, obteniendo los valores de los parámetros que los caracterizan; teniendo así una mejor

operación del manipulador. Dado lo anterior, varias empresas han empezado a producir diversos equipos académicos con el fin de estudiar este tipo de robots, una de ellas es la empresa china *Googol Technology*, cuyo robot paralelo, que se encuentra en el laboratorio de modos deslizantes de la UNAM, fue utilizado para las pruebas experimentales de este trabajo.

En lo relativo a la construcción de estos a nivel comercial se pueden enlistar los siguientes ejemplos tomados de [5], donde se aprecia su extensa aplicación en diversas áreas.

- El robot Delta desarrollado por R. Clavel del cual se calcula que existen más de 4000 unidades en funcionamiento a nivel mundial, y es utilizado especialmente en industrias de manipulación de alimentos.
- El robot Tricept patentado por K.N. Neumann del cual se estima que existen cerca de 300 unidades, y es utilizado en máquinas-herramienta de precisión y también ha incursionado en el campo de la medicina.
- La plataforma de Gough, la cual es el manipulador paralelo más construido con cerca de 20000 unidades, y es utilizada para orientación de antenas, telescopios, paneles solares, aislamiento y producción de vibraciones, posicionamiento de microscopios y pacientes, simuladores de vuelo de avión, vehículos elevadores, ensamble de componentes, posicionamiento de piezas y como máquina-herramienta de precisión media.

De igual forma en [5] se mencionan algunas de las **líneas actuales de investigación**, que surgen por el afán de solucionar problemas aún presentes en los manipuladores paralelos, las cuales se listan a continuación:

- *Diseño mecánico*
- *Cinemática*
- *Análisis de singularidades*
- *Espacio de trabajo*

- *Dinámica*
- *Control*
- *Análisis de precisión*

Este trabajo se centra en el aspecto del control y de su precisión asociada. El cual es todavía un campo abierto para la investigación y experimentación. De este problema en específico se desprenden diversos trabajos directamente relacionados: [6], [7], [8] y [9]. En donde se aborda a detalle el estudio de las restricciones geométricas y modelado de la planta, así como la implementación de controladores adaptables no lineales para seguimiento de trayectorias.

En lo que respecta al uso de modos deslizantes en robots paralelos sobresalen los trabajos: [10], y [11], donde igualmente se hace control de seguimiento de trayectorias en tiempo real, pero utilizando controladores por modos deslizantes convencionales para estimar perturbaciones. También en la tesis de licenciatura [12], se hace una comparación entre un controlador discontinuo por modos deslizantes con un controlador clásico PD con el mismo robot que se estudia en este trabajo.

1.4. Planteamiento del problema

Se desea implementar una ley de control continua para el seguimiento de trayectorias en un sistema sobreactuado. Para ello se propone el siguiente controlador:

$$u(t) = u_0(t) + u_1(t) \tag{1.1}$$

donde el $u_0(t)$ es el control nominal PD y el $u_1(t)$ es un controlador por modos deslizantes. Se utilizará para el caso del controlador robusto un Super-Twisting y también dos controladores basados en algoritmos de alto orden, logrando con esto la atenuación del *chattering*. En la Figura 1.2 se muestra una representación del sistema en estudio que ejemplifica lo mencionado anteriormente.

En el esquema propuesto, $\delta(t)$ representa las perturbaciones debidas a las dinámicas no modeladas de la planta; para nuestro caso en particular se tomará la fricción seca y fricción viscosa en las uniones activas del robot. La adquisición de los datos de la

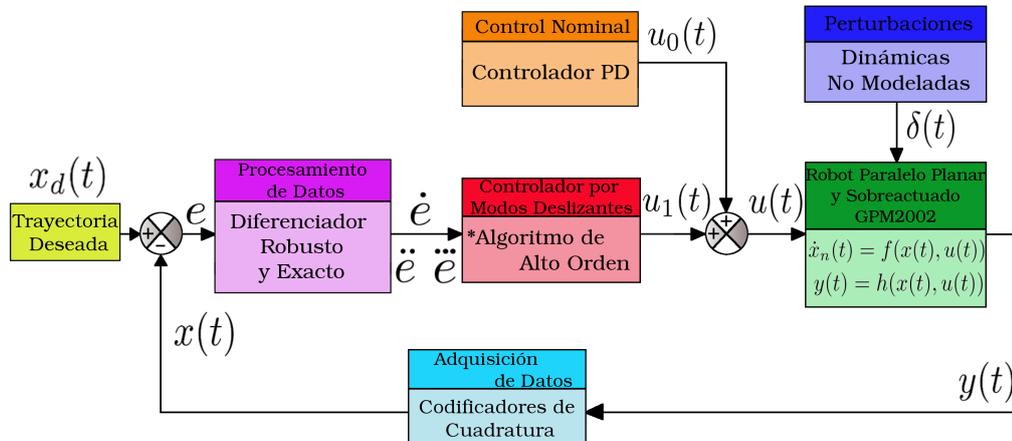


Figura 1.2: Esquema de control a utilizar para la planta en estudio

salida $y(t)$ se hace a través de codificadores ópticos, y el error, representado por la variable e , es la diferencia entre la trayectoria deseada $x_d(t)$ y la trayectoria real $x(t)$ que tiene que seguir la dinámica del manipulador paralelo. De igual forma se utilizan diferenciadores exactos y robustos para obtener las derivadas del error, necesarias para poder implementar los controladores de alto orden.

1.5. Metodología

1. Análisis y modelado de la planta.

En esta etapa se analizan las restricciones físicas que caracterizan al manipulador paralelo utilizando las bases teóricas correspondientes para llegar a un modelo matemático que describe la dinámica del propio robot.

2. Selección del algoritmo de control.

Una vez evaluada la planta de acuerdo a sus características se debe seleccionar el algoritmo de control que mejor cumpla con los objetivos propuestos, en este caso se desea implementar algoritmos por modos deslizantes de alto orden con base en diferenciadores exactos y robustos.

3. Diseño del esquema de control.

En esta etapa se sintonizan las ganancias de los controladores con base en la

teoría matemática de los HOSM, para después hacer simulaciones numéricas que aseguren la convergencia de los errores de seguimiento del robot a cero.

4. Implementación física.

Finalmente se probarán los diferentes tipos de controladores en la planta real, donde podrá ser necesario un ajuste en las ganancias debido a que existen muchos factores externos que influyen en el sistema y que no se consideran en la simulación. El propósito de control, una vez programado, es que el efector final del robot siga trayectorias punto a punto previamente calculadas, este procedimiento es llamado 'tracking'.

En el último paso de esta metodología, los controladores y las trayectorias programadas utilizan las funciones de la tarjeta controladora de movimiento **GT-400-SV**, que permiten una interfaz PC-Robot, las cuales están escritas en lenguaje C++; es decir que para la programación de los algoritmos se utiliza el paradigma orientado a objetos. También cabe mencionar que, aunque la adquisición de datos de los experimentos es en tiempo real, la información es rescatada en archivos de texto que se procesan posteriormente en un lenguaje de programación de más alto nivel, en este caso *Matlab*, para poder presentar gráficas y resultados numéricos. Para más detalle e información sobre la tarjeta consultar el apéndice correspondiente.

1.6. Contribuciones

En este trabajo se usan controladores por modos deslizantes de alto orden, también llamados de orden superior, para compensar perturbaciones debidas a la fricción y mejorar la precisión de seguimiento de trayectorias. Se proponen dos estrategias:

- **Control continuo de tercer orden** con compensación de perturbaciones desconocidas utilizando el estimado de las derivadas con base en un diferenciador de cuarto orden robusto y exacto.
- **Control continuo de cuarto orden** con compensación de perturbaciones desconocidas utilizando el estimado de las derivadas con base en un diferenciador de quinto orden robusto y exacto.

Se realiza un análisis de los controladores antes mencionados y se compara con un Super-Twisting, el cual es un controlador de orden dos bastante estudiado. La información que arroja el análisis comparativo entre las estrategias de control antes mencionadas ayuda a dar un panorama más amplio de las acciones a tomar si se requiere controlar un robot paralelo y sobreactuado.

1.7. Estructura de la tesis

Este trabajo está dividido en 7 capítulos. Se introducen las bases teóricas en el Capítulo 2. Después, en el Capítulo 3, se lleva a cabo el análisis del robot paralelo de dos grados de libertad para obtener su modelo matemático. En el Capítulo 4 se presentan las leyes de control que se utilizan para implementarse en el sistema sobreactuado para, posteriormente, mostrar las simulaciones realizadas en el Capítulo 5, determinando así el desempeño teórico del esquema diseñado. En el Capítulo 6 se exponen los resultados obtenidos en la etapa de experimentación y su análisis correspondiente. Finalmente, en el Capítulo 7 se muestran las conclusiones generales de este trabajo.

En este capítulo se exponen de manera general las matemáticas aplicadas a lo largo de la elaboración de este trabajo. No se profundiza en la estructura de los controladores a utilizar ya que estos se ven a detalle en el Capítulo 4. Sin embargo sí se presenta una introducción al control por modos deslizantes justificando su utilización, así como también una introducción a la dinámica de los robots manipuladores y un compilado de conceptos fundamentales de robótica, los cuales son indispensables para el entendimiento de la planta.

2.1. Robots paralelos

2.1.1. Conceptos básicos de robótica

Dentro del ámbito de los sistemas mecánicos existe un tipo especial que permite que un cuerpo rígido, denominado *efector final*, se mueva con respecto a una base fija. Este cuerpo rígido en el espacio puede moverse de diferentes maneras: en un movimiento de traslación o en un movimiento de rotación, llamados *grados de libertad*. De igual forma la posición y orientación del efector final, llamada *pose*, puede ser descrita por coordenadas generalizadas, las cuales son el conjunto de variables que determinan el estado del sistema y son usualmente las coordenadas de un punto específico del efector final y los ángulos que definen su orientación. En [13] se encuentra una definición más formal de estos conceptos.

Si es posible controlar los grados de libertad del efector final via un sistema mecánico, el sistema puede ser llamado robot [4]. Cabe aclarar que el efector final del robot únicamente puede moverse dentro de un espacio denominado *espacio de trabajo*, el cual

2. BASE TEÓRICA

es un subconjunto del espacio cartesiano, y está delimitado por la configuración propia del robot.

Una definición más precisa del término robot viene del Instituto de Robots de America (RIA por sus siglas en Inglés): *Un robot es un manipulador multifuncional reprogramable diseñado para mover material, partes, herramientas, o dispositivos especializados a través de movimientos variables programados para la realización de diversas tareas* [13].

La mayoría de los manipuladores existentes se constituyen de una sucesión de cuerpos rígidos denominados eslabones, cada uno de los cuales está ligado a su predecesor y a su sucesor por una articulación de un grado de libertad. A esta arquitectura se le denomina robot en serie, haciendo una analogía con los circuitos eléctricos. También existen otro tipo de robots llamados **manipuladores paralelos** los cuales se pueden definir de la siguiente manera: *mecanismo de cadenas cinemáticas cerradas en la cual el efector final está ligado a la base por varias cadenas cinemáticas independientes* [4].

En la Figura 2.1 se muestra un ejemplo de un manipulador serie y uno paralelo tomado de [14].

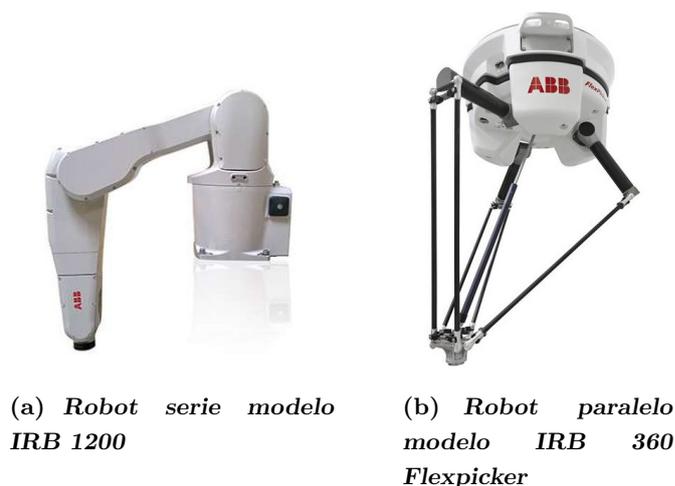


Figura 2.1: Robots industriales de la empresa ABB

2.1.2. Precisión de un robot

Una de las características importantes de los robots es la concerniente a la precisión y exactitud de posicionamiento. Mertel en [4] explica dos conceptos relacionados con el desempeño de un robot: la *exactitud absoluta*, definida como la distancia entre la posición deseada y la posición real del efector final, y la *repetibilidad*, la cual es la distancia máxima entre dos posiciones del efector final alcanzadas para la misma pose deseada desde diferentes posiciones iniciales.

En ese mismo escrito se plantea que la exactitud absoluta de un robot en serie es pobre; lo que los hace inapropiados para tareas que requieren la manipulación de cargas pesadas y exactitud de posicionamiento. Sin embargo, para propósitos de esta tesis, el aspecto de interés es el de repetibilidad de un robot, llamada a partir de ahora *precisión*, ya que esta se encuentra condicionada, de entre otros factores, por la ley de control que provoca el movimiento del manipulador.

Por otra parte los manipuladores paralelos tienen la ventaja de baja inercia, alta rigidez y alta capacidad de sobrecarga impulsada [7]. Esto ha abierto grandes oportunidades a los robots paralelos para ser aplicados en diferentes ámbitos industriales y en líneas actuales de investigación, lo cual se detalla en [5]. Sin embargo una de las dificultades en las aplicaciones reales recae en el diseño de un controlador adecuado, el cual se ve limitado por las restricciones inherentes en lazo cerrado, ya que esto provoca que las articulaciones de los manipuladores paralelos estén estrechamente acopladas y sus características dinámicas sean altamente **no lineales**.

2.1.3. Robots sobreactuados

La configuración de ciertos tipos de manipuladores paralelos muestra muchas singularidades y pobre espacio de trabajo si las limitaciones físicas de las articulaciones entran en juego. La *redundancia* ofrece una buena oportunidad de lidiar con estos problemas importantes y complejos, mejorando las habilidades y desempeño de estos robots en ciertas maneras.

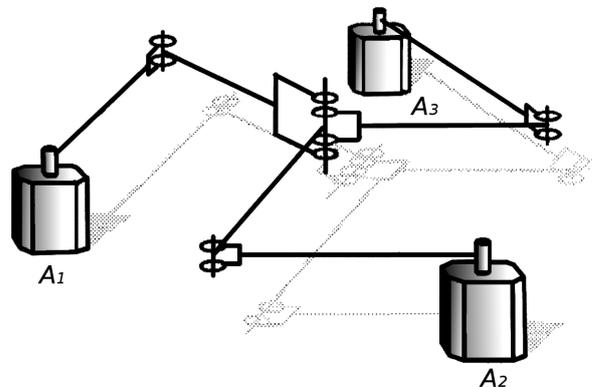


Figura 2.2: Principio de funcionamiento de un manipulador x-y

En [4] se citan diferentes tipos de redundancia, de las cuales, la que define el tipo de sistema en estudio de esta tesis es la redundancia de actuadores o *sobreactuación*. Esto significa que, en general, asumiendo articulaciones y eslabones rígidos, cargas internas pueden ser generadas. Esto es útil cuando una o más cadenas estén en posición singular ya que los actuadores extra aseguran movilidad del manipulador.

En este trabajo se hacen experimentos con un manipulador planar en x-y (el cual se describe en el Capítulo 3). Este es de clasificación tipo RRR [13] donde cada R representa un grado de libertad rotacional. Las tres cadenas cinemáticas, o ‘brazos’, que conforman al robot se unen en una articulación central, donde una herramienta podría ser montada. Por lo tanto se pierde un grado de libertad rotacional, pero se obtiene una unión de eslabones con un actuador redundante. La Figura 2.2 muestra el principio de funcionamiento del manipulador en cuestión.

2.1.4. Dinámica del robot

Para describir la dinámica del robot manipulador paralelo mostrado en la sección anterior, se utiliza un conjunto de ecuaciones diferenciales ordinarias de segundo orden no lineales, las cuales dependen de las propiedades cinemáticas e inerciales del robot. Las ecuaciones dinámicas son importantes ya que describen la relación entre la fuerza y el movimiento, y para este trabajo se hace uso de la deducción ‘Lagrangiana’ de la dinámica de la planta. Esta técnica tiene la ventaja de requerir únicamente de las

energías cinética y potencial del sistema a ser calculado, y por lo tanto tiende a ser menos propenso a errores [15], a diferencia de sumar juntas la fuerza inercial, de Coriolis, centrífuga, del actuador, y otras fuerzas que actúan sobre los eslabones del robot. De igual forma permite a las propiedades estructurales ser determinadas y explotadas.

Una vez que las ecuaciones de movimiento para el manipulador son conocidas se puede tratar ahora el problema del *control* del robot manipulador, que se aborda en el Capítulo 4, lo que conlleva a encontrar las fuerzas en los actuadores que causan que el manipulador se mueva a través de una trayectoria dada. Sin embargo hay que tener en consideración que todo modelo matemático está sujeto a discrepancias con la planta real que evitan que el sistema converga a la trayectoria deseada en su totalidad, esto debido a la presencia de errores en las condiciones iniciales, ruidos del sensor o incluso errores del modelado.

2.2. Ecuaciones de Euler-Lagrange

En esta sección se introducen las llamadas ecuaciones de **Euler-Lagrange** tomadas de [15], las cuales se derivan de la segunda ley de Newton y describen la evolución de los sistemas mecánicos en el tiempo sujetos a restricciones holonómicas (término que se definirá más adelante). Esto para dar pie a la realización del análisis matemático en el siguiente capítulo.

Aunque existen muchos métodos para generar las ecuaciones dinámicas de un sistema mecánico, todos los métodos generan un conjunto equivalente de ecuaciones; sin embargo diferentes formas de las ecuaciones pueden ser más adecuadas para los cálculos o el análisis. En este trabajo se decidió utilizar este enfoque ya que las ecuaciones resultantes pueden ser calculadas en lazo cerrado, permitiendo el análisis detallado de las propiedades del sistema.

2.2.1. Formulación básica

Se considera un sistema de n partículas que obedece la segunda ley de Newton; la velocidad de variación de impulso de una partícula es proporcional a la fuerza aplicada a

2. BASE TEÓRICA

ésta. Sea F_i la fuerza aplicada sobre la i -ésima partícula, m_i la masa de dicha partícula, y r_i sea su posición, entonces la ley de Newton se convierte en:

$$F_i = m_i \ddot{r}_i \quad r_i \in \mathbb{R}^3, i = 1, \dots, n. \quad (2.1)$$

El interés de este desarrollo no está en un conjunto de partículas independientes, sino más bien en partículas que están unidas y tienen grados limitados de libertad. Para describir esta interconexión, se introducen restricciones entre las posiciones de nuestras partículas. Cada restricción está representada por una función $g_j : \mathbb{R}^{3n} \rightarrow \mathbb{R}$ tal que:

$$g_j(r_1, \dots, r_n) = 0 \quad j = 1, \dots, k. \quad (2.2)$$

Una restricción que se puede escribir de esta forma, como una relación algebraica entre las posiciones de las partículas, se le denomina una restricción holonómica. Restricciones más generales de los cuerpos rígidos, lo que implica r_i , también se pueden presentar en múltiples cadenas o eslabones.

Una restricción actúa sobre un sistema de partículas mediante la aplicación de fuerzas de restricción. Estas se determinan de tal manera que la restricción en la ecuación (2.2) se cumple siempre. Si consideramos la restricción como una superficie lisa en \mathbb{R}^n , las fuerzas de restricción son normales a la superficie y se debe restringir la velocidad del sistema para que sea tangente a la superficie en todo momento. Por lo tanto, se puede reescribir (2.1) como:

$$F = \begin{bmatrix} m_1 I & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & m_n I \end{bmatrix} \begin{bmatrix} \ddot{r}_1 \\ \vdots \\ \ddot{r}_1 \end{bmatrix} + \sum_{j=1}^k \Gamma_j \lambda_j, \quad (2.3)$$

donde los vectores $\Gamma_1, \dots, \Gamma_k \in \mathbb{R}^{3n}$ son una base para las fuerzas de ligadura y λ_j es el factor de escala para el elemento de base j -ésimo. No se requiere que $\Gamma_1, \dots, \Gamma_k$ sea ortogonal. Por restricciones de la forma en (2.2), Γ_j puede ser tomado como el gradiente de g_j , que es perpendicular al nivel establecido $g_j(r) = 0$.

Los escalares $\lambda_1, \dots, \lambda_k$ se denominan multiplicadores de Lagrange. Se deben determinar los multiplicadores de Lagrange para resolver las $3n + k$ ecuaciones dadas por

(2.2) y (2.3) para las $3n + k$ variables $r \in \mathbb{R}^{3n}$ y $\lambda \in \mathbb{R}^k$. Los valores λ_i sólo dan las magnitudes relativas de las fuerzas de restricción ya que los vectores Γ_j no son necesariamente ortonormales.

Esta forma de lidiar con las restricciones holonómicas es intuitivamente simple pero computacionalmente complejo, ya que se debe realizar un seguimiento del estado de todas las partículas en el sistema, aunque no sean capaces de un movimiento independiente. Un enfoque más atractivo es describir el movimiento del sistema en términos de un conjunto más pequeño de variables que describe completamente la configuración del sistema. Para un sistema de n partículas con k restricciones, buscamos un conjunto de $m = 3n - k$ variables q_1, \dots, q_m y funciones suaves f_1, \dots, f_n tales que:

$$r_i = f_i(q_1, \dots, q_m) \iff g_j(r_1, \dots, r_n) = 0 \quad (2.4)$$

$$i = 1, \dots, n \qquad j = 1, \dots, k.$$

A las q_i se les llama un conjunto de coordenadas generalizadas para el sistema. Para una robot manipulador que consiste en uniones rígidas, estas coordenadas generalizadas casi siempre son elegidas para ser los ángulos de las articulaciones. La especificación de estos ángulos determina la posición de todas las partículas que constituyen el robot.

Dado que los valores de las coordenadas generalizadas son suficientes para especificar la posición de las partículas, se pueden volver a escribir las ecuaciones de movimiento para el sistema en términos de estas. Para ello, también se expresan las fuerzas externas aplicadas al sistema en términos de componentes a lo largo de estas mismas coordenadas generalizadas. Se les llama a estas fuerzas las *fuerzas generalizadas* para distinguirlas de las fuerzas físicas, que están siempre representadas como vectores en \mathbb{R}^3 . Para un robot manipulador con los ángulos de las articulaciones que actúan como coordenadas generalizadas, las fuerzas generalizadas son los pares aplicados sobre los ejes de las articulaciones.

Para escribir las ecuaciones de movimiento, se define la función de Lagrange, L , como la diferencia entre la energía cinética y la energía potencial del sistema, es decir:

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q), \quad (2.5)$$

2. BASE TEÓRICA

donde T es la energía cinética y V es la energía potencial del sistema, ambos escritos en coordenadas generalizadas.

Finalmente se establece el siguiente Teorema:

Teorema 2.1 [15] *Las ecuaciones de movimiento para un sistema mecánico con coordenadas generalizadas $q \in \mathbb{R}^m$ y Lagrangiano L vienen dadas por:*

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i \quad (2.6)$$

donde τ_i es la fuerza externa actuando en la i -ésima coordenada generalizada. La expresión anterior se conoce como ecuación de Lagrange.

2.3. Grado relativo de un sistema

El conocimiento del grado relativo de un sistema es fundamental para poder trabajar con los modos deslizantes. El valor de este grado es uno de los pocos datos necesarios que da pie a las leyes de control con las que se trabaja en esta tesis. Formalmente se define de la siguiente manera

Definición. [16] *Considerando un sistema dinámico SISO (una entrada y una salida) con la salida $y \in \mathbb{R}$, un vector de estados $x \in \Theta \subset \mathbb{R}^n$, y una entrada de control $u \in \mathbb{R}$. Si $y^{(i)}$ es independiente de u para todo $i = 1, 2, \dots, k-1$, pero $y^{(k)}$ es proporcional a u con un coeficiente de proporcionalidad diferente de cero en un dominio razonable $\Omega \subset \Theta \subset \mathbb{R}^n$, entonces k es llamado grado relativo bien definido.*

Es decir que el *grado relativo* es el número entero positivo igual al número de veces que la variable de salida es sucesivamente diferenciada con respecto al tiempo para que el control de entrada u aparezca por primera vez.

Por ejemplo, para el siguiente sistema mecánico

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u + f(x_1, x_2, t) \\ y = x_1, \end{cases} \quad (2.7)$$

con $x_1(0) = x_{10}$ y $x_2(0) = x_{20}$. Donde x_1, x_2 son las posiciones y velocidad de la unidad de masa, y es una salida controlada, u es la fuerza de control, y el término $f(x_1, x_2, t)$ es la perturbación, que comprende fuerza de fricción viscosa y seca así como otras fuerzas de resistencia desconocidas.

Para el sistema (2.7) la dinámica de entrada-salida tiene grado relativo $k = 2$ ya que

$$y^{(2)} = u + f(y, \dot{y}, t). \quad (2.8)$$

Se debe remarcar también que para este caso específico el grado del sistema es igual al orden del sistema, lo que significa que el sistema (2.7) no tiene ninguna dinámica interna.

2.4. Introducción a los Modos Deslizantes

El control en presencia de incertidumbres es uno de los tópicos principales de la teoría moderna de control, ya que en la formulación de cualquier problema siempre hay discrepancias entre la dinámica de la planta real y su modelo matemático usado para el diseño del controlador. Estas discrepancias o desajustes mayoritariamente vienen de perturbaciones externas, parámetros desconocidos de la planta, y dinámicas parásitas. Es por esto que el diseño e implementación de leyes de control que proveen un comportamiento deseado de un sistema en lazo cerrado en presencia de estas perturbaciones o incertidumbres es una tarea importante para un ingeniero en control.

La técnica de *controladores por modos deslizantes* (SMC por sus siglas en inglés) es una de las más exitosas dentro de los diversos métodos de control robusto que se han desarrollado para resolver este problema, y su uso resulta bastante adecuado cuando se trata de manejar perturbaciones o incertidumbres acotadas y dinámicas parásitas [16].

Los modos deslizantes fueron propuestos como un modo especial en los sistemas de estructura variable (VSS por sus siglas en inglés). Estos sistemas comprenden una variedad de estructuras, con reglas para conmutar entre ellas en tiempo real para alcanzar un comportamiento adecuado para la planta. El resultado del estudio de los VSS desde

2. BASE TEÓRICA

1950 hasta años recientes se encuentra en [16] y la división por generaciones del desarrollo de los modos deslizantes viene reportado en [1]. En estos trabajos se especifica el procedimiento para el diseño de un controlador por modos deslizantes, el cual está basado en dos etapas:

- Diseño de una superficie de deslizamiento.
- Controladores discontinuos que garanticen los modos deslizantes.

Las principales ventajas reportadas de este tipo de controladores son:

- Insensibilidad ante perturbaciones acotadas y acopladas a la señal de control.
- Reducción del orden de las ecuaciones del sistema.
- Convergencia en tiempo finito a la superficie de deslizamiento.

Sin embargo las principales desventajas de los SMC son evidentes:

- El controlador usualmente produce una acción de conmutación a alta frecuencia que provoca el llamado efecto *chattering*, el cual es difícil de atenuar.
- Insensibilidad sólo con perturbaciones acopladas.
- Las variables de deslizamiento convergen en tiempo finito, pero las variables de estado sólo convergen asintóticamente.
- El desempeño en lazo cerrado no es ideal en presencia de discretización, ruido y dinámicas parásitas.
- El diseño de la superficie de deslizamiento se limita a tener un grado relativo igual a uno con respecto al control, lo que significa que el control necesita aparecer explícitamente en la primera derivada temporal de la variable de deslizamiento. Esto puede restringir la elección de la variable de deslizamiento.

2.4.1. Orden de un Modo Deslizante

La idea principal del SMC, tomada de [16], es mantener en cero una restricción elegida apropiadamente en el modo deslizante. Se pueden clasificar a los modos deslizantes por el grado de suavidad de la función de restricción calculada a lo largo de las trayectorias del sistema. Para ser más precisos, se considera que la restricción sea de la forma $\sigma = 0$, donde σ es alguna función suave del estado y el tiempo, y se mantiene idénticamente sobre trayectorias *Filippov* de un sistema dinámico discontinuo. No se considera la función σ en las trayectorias del modo deslizante, ya que en esa situación es idéntica a cero. Por otro lado el modo deslizante $\sigma \equiv 0$ puede ser clasificado por la primera derivada total $\sigma^{(r)}$ la cual contiene una discontinuidad en una pequeña vecindad de las trayectorias del modo deslizante. El número r es llamado el **orden de deslizamiento**. La definición formal es la siguiente.

Definición.[16] *Considerando una ecuación diferencial discontinua $\dot{x} = f(x)$ (Inclusión diferencial de Filippov $\dot{x} \in F(x)$) con una función de salida suave $\sigma = \sigma(x)$, y entendiéndose en el sentido de Filippov. Entonces, si*

1. Las derivadas temporales $\sigma, \dot{\sigma}, \dots, \sigma^{(r-1)}$ son funciones continuas de x .
2. El conjunto

$$\sigma = \dot{\sigma} = \dots = \sigma^{(r-1)} = 0 \tag{2.9}$$

es un conjunto de integrales no vacías (i.e., consiste de trayectorias de Filippov).

3. El conjunto Filippov de velocidades admisibles en los puntos del modo deslizante de orden r en la ecuación (2.9) contiene más que un vector.

Se dice que en el movimiento sobre el conjunto de la ecuación (2.9) existe en un modo deslizante de orden r .

2.5. Modos Deslizantes de Segundo Orden

Las desventajas intrínsecas del uso de los modos deslizantes clásicos o de primer orden fue la motivación principal para que surgieran los llamados *modos deslizantes de segundo orden* (SOSM por sus siglas en inglés). De esta manera pueden mitigarse las dificultades que conlleva el uso de los SMC. Un SOSM puede aplicarse a sistemas de segundo orden con grado relativo dos; pero también existen algoritmos que proveen una señal de control continua, reduciendo significativamente el efecto *chattering*.

A continuación se presentan dos importantes algoritmos de los modos deslizantes de segundo orden tomados de [16]. Considerando un sistema dinámico en variables de estado de la forma:

$$\dot{x} = a(t, x) + b(t, x)u, \quad \sigma = \sigma(t, x). \quad (2.10)$$

Donde $x \in \mathbb{R}^n$, $u \in \mathbb{R}$ es el control, σ es la única salida medible, y las funciones a , b , σ y la dimensión n son desconocidas. La tarea es hacer que la salida σ se desvanezca en tiempo finito y mantener $\sigma \equiv 0$ por medio de un control realimentado acotado globalmente y discontinuo.

2.5.1. Algoritmo Twisting

El primer algoritmo y el más simple de los SOSM es el llamado algoritmo “Twisting”. Para un sistema de segundo orden con grado relativo dos toma la siguiente estructura:

$$u = -r_1 \operatorname{sign}(\sigma) - r_2 \operatorname{sign}(\dot{\sigma}), \quad r_1 > r_2 > 0. \quad (2.11)$$

Con el uso del algoritmo Twisting, *ya no se necesita diseñar superficie de deslizamiento* siendo posible implementarla directamente a los estados del sistema; en ese caso son los estados los que presentan convergencia en tiempo finito a cero (tanto para σ como para $\dot{\sigma}$), con los parámetros r_1 y r_2 seleccionados apropiadamente, i.e. existe $T > 0$ tal que para todo $t > T$, $\sigma = \dot{\sigma} = 0$. Sin embargo tiene la desventaja de que sigue presentando una señal de control discontinua.

2.5.2. Algoritmo Super-Twisting

El algoritmo Twisting es útil cuando se tiene un sistema de grado relativo dos, sin embargo, este enfoque requiere la derivada $\dot{\sigma}$ y no atenúa la discontinuidad del controlador presente en los SMC convencionales. Si se considera un sistema como el de la ecuación (2.10) de grado relativo 1 y se supone que

$$\dot{\sigma} = h(t, x) + g(t, x)u, \quad (2.12)$$

además se asume que para algunas constantes positivas C, K_m, K_M, U_M, q

$$|\dot{h}| + U_M|\dot{g}| \leq C, \quad 0 \leq K_m \leq g(t, x) \leq K_M, \quad |h/g| < qU_M, \quad 0 < q < 1, \quad (2.13)$$

el algoritmo Super-Twisting se puede definir de la siguiente manera:

$$u = -\lambda |\sigma|^{1/2} \text{sign}(\sigma) + \nu, \quad (2.14)$$

$$\dot{\nu} = \begin{cases} -u, & |u| > U_M \\ -\alpha \text{sign}(\sigma), & |u| \leq U_M. \end{cases}$$

Obteniendo el resultado siguiente:

Teorema 2.2 [16] *Con $K_m \alpha > C$ y λ lo suficientemente grande, el controlador (2.14) garantiza un modo deslizante de segundo orden $\sigma = \dot{\sigma} = 0$ en el sistema (2.12) el cual atrae a la trayectoria en tiempo finito. El controlador u entra en tiempo finito al segmento $[-U_M, U_M]$ y se queda ahí. Nunca deja el segmento, si la condición inicial se encuentra adentro desde el principio.*

Es decir que con incertidumbres o perturbaciones acotadas, y para ciertas constantes λ y α , se asegura la convergencia en tiempo finito del modo deslizante de segundo orden, i.e. $\sigma = \dot{\sigma} = 0$. Cabe recalcar que la atenuación del efecto *chattering*, para sistemas con grado relativo 1, se logra ya que el controlador está formado por dos términos, el primero de ellos es continuo y el segundo es la integral de un término discontinuo; conservando así la robustez del controlador pero utilizando una señal de control continua. Si se desea aplicar este controlador a sistemas de grado relativo dos, es necesario diseñar una variable de deslizamiento con lo que se tiene convergencia asintótica de los estados a cero.

2.6. Modos Deslizantes de Alto Orden (HOSM)

La utilización de los modos deslizantes de segundo orden se resume a que hacen que las variables de deslizamiento se ‘desvanezcan’ en tiempo finito, cuando el grado relativo de la variable es igual a dos, y son capaces de resolver el mismo problema por medio de un control continuo, si el grado relativo es uno. Esto ayuda a remover las vibraciones peligrosas causadas por el *chattering*. Los llamados modos deslizantes de alto orden (HOSM por sus siglas en inglés) resuelven estos problemas para grados relativos arbitrarios. El inconveniente recae en que la realización del esquema requiere más información; usualmente se necesitan calcular o medir las derivadas de las variables de deslizamiento un número de veces sucesivas. Sin embargo, el mismo estudio de los HOSM ha dado como resultado el desarrollo de diferenciadores robustos de orden arbitrario, teniendo su propio significado en términos de la teoría general de observación. En particular, los problemas de seguimiento se resuelven en tiempo finito y con exactitud ideal, por medio de un control continuo, si el grado relativo de la salida es conocido. La exactitud se mantiene alta en presencia de ruidos pequeños, conmutando inexactitudes y retardos [3].

Aunque la teoría de este tipo de controladores puede ser considerada complicada, su aplicación es muy simple y directa. Suponiendo que el problema estudiado es hacer una salida suave y escalar σ desvanecerse y mantenerse en cero, asumiendo que el sistema es suave y afín en el control, y de grado relativo r . Las derivadas $\dot{\sigma}, \dots, \sigma^{(r-1)}$ que son requeridas para implementar el controlador pueden ser calculadas en tiempo real por medio de algún diferenciador de modos deslizantes de alto orden. El diferenciador es robusto con respecto a ruidos de entrada y exacto en su ausencia.

Hasta este punto se puede empezar a usar únicamente los resultados sin tomar mucha consideración a los desarrollos teóricos. La teoría requerida para el entendimiento profundo de este método propuesto, su exactitud y sus limitaciones, vienen descritas a detalle en [3] y [17] donde se usa la teoría de homogeneidad para simplificar la prueba y el diseño de control, y para calcular la exactitud de los controladores y diferenciadores.

Modelo y descripción del robot

En este capítulo se presenta una descripción de los elementos que componen a la planta, y de su funcionamiento. Se obtiene el modelo matemático mediante el método ‘Lagrangiano’, presentado en el Capítulo 2, para después ser transformado a un modelo en variables de estado. Este modelo será de utilidad para posteriormente realizar las simulaciones, las cuales facilitan el análisis del sistema en lazo cerrado.

3.1. Descripción de la planta

El sistema en estudio, utilizado para las pruebas experimentales, consiste en un robot paralelo sobreactuado de dos grados de libertad de la compañía *Googol Technologies*, ver Figura 3.1, el cual tiene un movimiento planar para el efector final. El equipo es el modelo **GPM2002**, que cuenta con tres servomotores de corriente alterna en las articulaciones activas, los cuales funcionan como actuadores del sistema. De igual forma incluyen un codificador incremental óptico para cada servomotor, que sirven para medir la posición de cada uno de los ‘brazos’ del manipulador. Las características más importantes de los actuadores se muestran en la Tabla 3.1, tomada de [18].

El equipo también cuenta con un módulo de potencia para los servomotores, compuesto de 3 sistemas de mando llamados servo-amplificadores, provistos por la empresa *Sanyo Denki*. La comunicación con la computadora se logra por medio de una tarjeta llamada GT-400-SV, con lo que se logra crear la interfaz entre el robot y los comandos de control programados.

3. MODELO Y DESCRIPCIÓN DEL ROBOT

Característica	Valor	Unidad
Velocidad Nominal de Revolución	3000	rpm
Máxima Velocidad de Revolución	4500	rpm
Par Nominal	0.637	Nm
Par de Parada Continua	0.686	Nm
Corriente de Armadura Nominal	1.6	A_{rms}
Corriente de Armadura de Parada Continua	1.7	A_{rms}
Constante Par-Corriente	0.436	Nm/A_{rms}
Resistencia de Armadura por Fase	3.4	Ω
Resolución del Codificador	8192	P/R

Tabla 3.1: Especificaciones del servomotor de corriente alterna marca PY modelo P50B05020D



Figura 3.1: Vista superior del robot paralelo

En la Figura 3.2 se muestra un esquema con las partes que conforman el equipo de experimentación, necesarias para el control del manipulador. De forma abreviada, el funcionamiento del sistema consiste en calcular previamente la trayectoria a seguir en coordenadas cartesianas, para después convertirlas a posiciones de los actuadores, es decir a coordenadas generalizadas, utilizando un algoritmo de cinemática inversa. Los ángulos de los actuadores son guardados en un arreglo de igual dimensión al número de puntos que conforman la trayectoria. Una vez que inicia el movimiento del robot la acción de control programada mueve los 3 actuadores del robot a base de señales de voltaje que se traducen en par de fuerzas para cada servomotor; tratando que el error

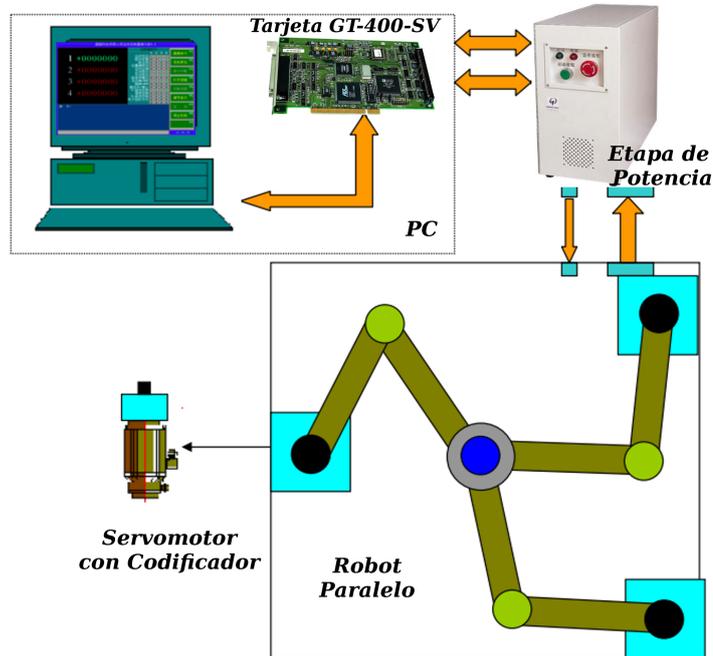


Figura 3.2: Interfaz entre la computadora y el robot paralelo

existente entre los ángulos calculados y los leídos por los codificadores en tiempo real sea igual a cero. De igual forma la posición del efector final en un punto del plano se puede monitorizar por medio del uso de la cinemática directa. Si se desea conocer más datos técnicos del equipo se pueden consultar los manuales [18] y [19].

3.1.1. Espacio de trabajo del robot paralelo

La determinación del *espacio de trabajo* es crítico para la cinemática del mecanismo paralelo, ver Capítulo 2. Para poder garantizar que una trayectoria planeada esté contenida dentro de él, la primera tarea es determinar dónde está ubicado dicho espacio.

Dada la cinemática del mecanismo paralelo (ver Figura 3.3), se pueden deducir las siguientes relaciones:

$$\begin{aligned}
 g1 &: (x - x_1)^2 + (y - y_1)^2 = R^2 \\
 g2 &: (x - x_2)^2 + (y - y_2)^2 = R^2 \\
 g3 &: (x - x_3)^2 + (y - y_3)^2 = R^2.
 \end{aligned}$$

Donde R es una variable que denota la distancia entre el punto central O y $A1$, $A2$ y $A3$; x_i y y_i son las coordenadas del extremo final de cada una de las cadenas cinemáticas independientes que conforman al robot. Aquellas (x, y) que satisfagan las ecuaciones anteriores se encuentran en el espacio de trabajo. De igual forma existen dos valores, R_{max} y R_{min} , las cuales son el valor más grande y más chico de R respectivamente, tal que las ecuaciones tienen soluciones reales. Es evidente que las ecuaciones representan círculos geométricos, por lo tanto, el espacio de trabajo es la intersección de las tres familias de círculos de diferentes centros y con radio de R_{min} a R_{max} . La Figura 3.3 muestra esquemáticamente el espacio de trabajo en color rojo. En la figura, G_1 , G_2 y G_3 son los puntos de intersección de g_2 y g_3 , g_1 y g_3 , y g_1 y g_2 . Es claro que:

$$\begin{aligned} \widehat{G_1G_2} &\text{ está sobre } g_3; \\ \widehat{G_2G_3} &\text{ está sobre } g_1; \\ \widehat{G_3G_1} &\text{ está sobre } g_2. \end{aligned}$$

El espacio de trabajo es por lo tanto la zona de intersección de:

$$\widehat{G_1G_2}, \quad \widehat{G_2G_3} \quad \text{y} \quad \widehat{G_3G_1}.$$

A medida que el efector final del manipulador se aleja del espacio en rojo las ecuaciones del modelo entran en indeterminación. Lo que se traduce en pérdida de exactitud e incluso de controlabilidad [4]. De la Figura 3.3 también se puede observar que el centro del espacio de trabajo se encuentra en $(216, 250)$ [mm]. Coordenada que se tomará como punto inicial para las trayectorias experimentales en el Capítulo 6.

3.2. Modelo matemático del robot

En esta sección se comienza la deducción matemática para poder llegar a las 6 ecuaciones diferenciales que describen al robot. Se trabaja en principio con la cinemática del robot paralelo en base a los trabajos [20] y [8], del 2007 y 2009 respectivamente. Primero la cinemática directa y después la cinemática inversa. El modelo dinámico se obtiene en función de los ángulos de los actuadores. Teniendo este último modelo en función de estos ángulos se lleva al espacio de tareas, con las restricciones algebraicas del sistema. Más adelante esas ecuaciones diferenciales serán reducidas para obtener finalmente un modelo en espacio de estados.

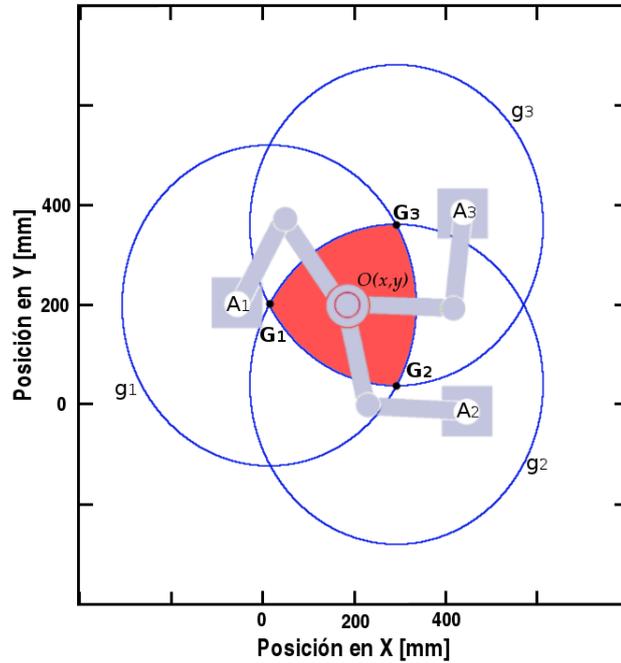


Figura 3.3: Efector final del robot dentro del espacio de trabajo

3.2.1. Cinemática del robot paralelo

La *cinemática directa* consiste en obtener la posición del efector final $p = (x, y)$ a partir de los valores de los ángulos $z = (\theta_{a1}, \theta_{a2}, \theta_{a3}, \theta_{b1}, \theta_{b2}, \theta_{b3})$; por simplificación escribiremos $z = (a_1, a_2, a_3, b_1, b_2, b_3)$.

En lo que respecta al análisis de los robots paralelos, no es posible aplicar algoritmos como el de Denavit-Hartenberg [21] para obtener de manera sistemática la *cinemática directa*. Sin embargo, existen otras técnicas más adecuadas si se desea obtener la caracterización de dicha cinemática.

Con este fin, en la Figura 3.4 se muestran los parámetros geométricos correspondientes a nuestra planta; la longitud de los 6 eslabones es de 244 mm, y es la misma para cada eslabón. Denotaremos esta longitud con el símbolo l , y las coordenadas de las posiciones de las articulaciones activas A_1 , A_2 y A_3 por los símbolos (x_{ai}, y_{ai}) , $i = 1, 2, 3$ que equivalen a $(0, 250)$, $(430, 0)$, y $(433, 500)$, respectivamente.

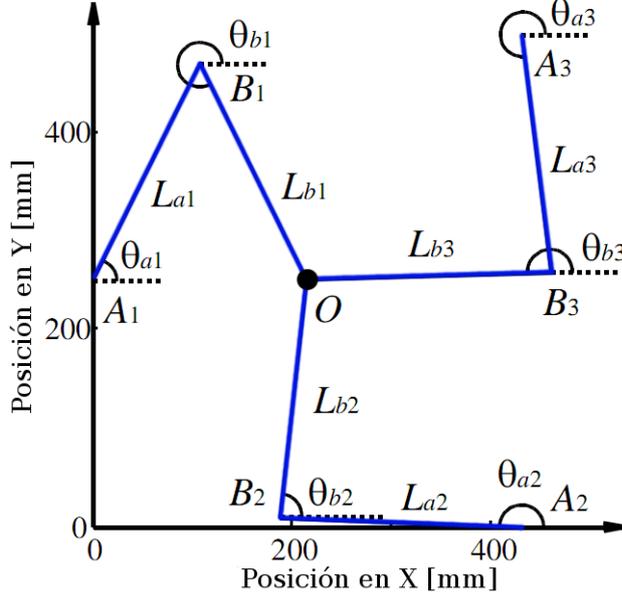


Figura 3.4: Estructura del robot paralelo GPM2002

La cinemática inversa [22] consiste en obtener los valores de los ángulos $z = (a_1, a_2, a_3, b_1, b_2, b_3)$ a partir de la posición del efector final $p = (x, y)$. En [20] se obtienen los ángulos de las uniones activas a_i , $i = 1, 2, 3$ mediante la siguiente expresión:

$$a_i = \arctan 2(x - x_{ai}, y - y_{ai}) + \arccos \frac{d_i}{2l_i}, \quad i = 1, 2, 3, \quad (3.1)$$

donde,

$$d_i = \sqrt{(x - x_{ai})^2 + (y - y_{ai})^2}, \quad i = 1, 2, 3.$$

De igual forma, los ángulos en las uniones pasivas b_i , $i = 1, 2, 3$ pueden ser obtenidas con las siguientes ecuaciones:

$$b_i = \arctan 2(x - x_{bi}, y - y_{bi}) \in (-\pi, \pi]; \quad i = 1, 2, 3. \quad (3.2)$$

En la ecuación (3.2), los símbolos x_{bi} , y_{bi} , $i = 1, 2, 3$ son las coordenadas de las uniones pasivas del robot paralelo de la Figura 3.4, las cuales pueden ser calculadas con las siguientes ecuaciones:

$$\begin{cases} x_{bi} = x_{ai} + l \cos a_i \\ y_{bi} = y_{ai} + l \sin a_i, \quad i = 1, 2, 3. \end{cases}$$

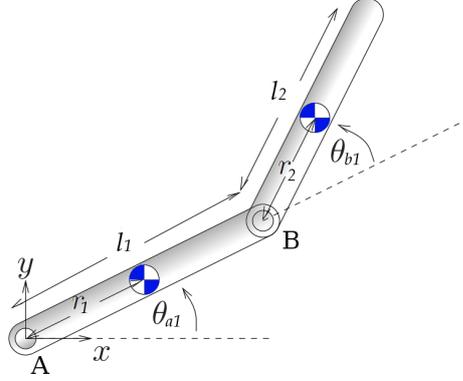


Figura 3.5: Cadena cinemática abierta

Con las ecuaciones (3.1) y (3.2), los ángulos de las articulaciones activas y pasivas correspondientes a la coordenada (x, y) del efector final pueden ser calculadas, y de esta forma el error de seguimiento del manipulador paralelo puede ser expresado como errores de seguimiento de las articulaciones.

3.2.2. Modelo dinámico: Cadena cinemática abierta

Antes de poder obtener el modelo completo del sistema, primero se necesita estudiar una cadena individual de las tres con las que cuenta nuestro robot paralelo. En el enfoque de Euler-Lagrange se necesita calcular primeramente el Lagrangiano.

Basándonos en la Figura 3.5, tenemos:

$$\begin{aligned}
 \bar{x}_1 &= r_1 c_1 & \dot{\bar{x}}_1 &= -r_1 s_1 \dot{a}_1 \\
 \bar{y}_1 &= r_1 s_1 & \dot{\bar{y}}_1 &= -r_1 c_1 \dot{a}_1 \\
 \bar{x}_2 &= l_1 c_1 + r_2 c_{12} & \dot{\bar{x}}_2 &= -(l_1 s_1 + r_2 s_{12})\dot{a}_1 - r_2 s_{12} \dot{b}_2 \\
 \bar{y}_2 &= l_1 s_1 + r_2 s_{12} & \dot{\bar{y}}_2 &= (l_1 c_1 + r_2 s_{12})\dot{a}_1 + r_2 c_{12} \dot{b}_2.
 \end{aligned} \tag{3.3}$$

Donde: $s_i = \sin a_i$, $s_{ij} = \sin(a_i - a_j)$ y de manera similar para c_i y c_{ij} .

La energía cinética queda de la siguiente forma:

$$T(\theta, \dot{\theta}) = \frac{1}{2} m_{a1} (\dot{\bar{x}}_1^2 + \dot{\bar{y}}_1^2) + \frac{1}{2} J_{a1} \dot{a}_1^2 + \frac{1}{2} m_{b1} (\dot{\bar{x}}_2^2 + \dot{\bar{y}}_2^2) + \frac{1}{2} J_{b1} (\dot{a}_1^2 + \dot{b}_1^2) \tag{3.4}$$

3. MODELO Y DESCRIPCIÓN DEL ROBOT

Luego de agrupar algunos términos y hacer $T = L$, se resuelve la ecuación de Euler-Lagrange. Después, utilizando algunas sustituciones algebraicas, el sistema se lleva a una forma matricial; resultando la siguiente expresión:

$$M_i(z)\ddot{z} + C_i(z, \dot{z})\dot{z} + f_i(z, \dot{z}) = \tau_i; \quad i = 1, 2, 3 \quad (3.5)$$

donde:

- M_i es la matriz de inercia.
- C_i es la matriz de fuerzas centrífugas y de Coriolis (en nuestro caso sólo contamos con fuerzas centrífugas).
- f_i representa las fuerzas gravitatoria y de fricción (debido a que el robot se encuentra en posición horizontal se considera despreciable la fuerza gravitatoria)
- τ_i es el par del actuador A_i

Dichas matrices tienen la forma siguiente:

$$M_i = \begin{bmatrix} \alpha_i & \gamma_i \cos(a_i - b_i) \\ \gamma_i \cos(a_i - b_i) & \beta_i \end{bmatrix} \quad (3.6)$$

$$C_i = \begin{bmatrix} 0 & (\gamma_i \cos(a_i - b_i)) \dot{b}_i \\ (\gamma_i \cos(a_i - b_i)) \dot{b}_i & 0 \end{bmatrix} \quad (3.7)$$

donde:

- $\alpha_i = J_{a1} + m_{ai} r_{ai}^2 + m_{bi} l_2$.
- $\beta_i = J_{bi} + m_{bi} r_{ai}^2$.
- $\gamma_i = m_{bi} r_{bi} l; \quad i = 1, 2, 3$.

Propiedades:

- M_i es una matriz simétrica definida positiva.
- $\frac{dM_i}{dt} - 2C_i$ es una matriz antisimétrica.

3.2.3. Modelo dinámico sin fuerzas de restricción

Cuando se combinan las dinámicas de las tres cadenas cinemáticas que componen al robot se tiene el siguiente modelo:

$$M(z)\ddot{z} + C(z, \dot{z})\dot{z} + f(\dot{z}) = \tau_i \quad (3.8)$$

Donde $f \in \mathbb{R}^6$ es la fuerza de fricción en las articulaciones y $\tau = (\tau_{a1}, \tau_{a2}, \tau_{a3}, 0, 0, 0)^T$ es el par en las uniones.

Las matrices M y C tienen la siguiente forma:

$$M = \begin{bmatrix} \alpha_1 & 0 & 0 & \gamma_1 \cos ab_1 & 0 & 0 \\ 0 & \alpha_2 & 0 & 0 & \gamma_2 \cos ab_2 & 0 \\ 0 & 0 & \alpha_3 & 0 & 0 & \gamma_i \cos ab_3 \\ \gamma_1 \cos ab_1 & 0 & 0 & \beta_1 & 0 & 0 \\ 0 & \gamma_2 \cos ab_2 & 0 & 0 & \beta_1 & 0 \\ 0 & 0 & \gamma_3 \cos ab_3 & 0 & 0 & \beta_1 \end{bmatrix} \quad (3.9)$$

$$C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ (\gamma_1 \cos ab_1)\dot{b}_1 & 0 & 0 \\ 0 & (\gamma_2 \cos ab_2)\dot{b}_2 & 0 \\ 0 & 0 & (\gamma_3 \cos ab_3)\dot{b}_3 \\ & (\gamma_1 \cos ab_1)\dot{b}_1 & 0 & 0 \\ & 0 & (\gamma_2 \cos ab_2)\dot{b}_2 & 0 \\ & 0 & 0 & (\gamma_3 \cos ab_3)\dot{b}_3 \\ & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ & 0 & 0 & 0 \end{bmatrix} \quad (3.10)$$

Por simplificación, el término $ab_i = a_i - b_i$, $i = 1, 2, 3$.

Al igual que en (3.6) y (3.7) las matrices cumplen con las siguientes propiedades:

- M_i es una matriz simétrica definida positiva.
- $\frac{dM_i}{dt} - 2C_i$ es una matriz antisimétrica.

3.2.4. Modelo dinámico con fuerzas de restricción

Tomando en cuenta el modelo de cinemática abierta y las fuerzas de restricción debidas al sistema en cadena cerrada, el modelo matemático del robot en el espacio de uniones queda de la siguiente manera:

$$M(z)\ddot{z} + C_i(z, \dot{z})\dot{z} + f(\dot{z}) = \tau + R^T\lambda \quad (3.11)$$

Donde $R^T\lambda$ es el vector de las fuerzas de restricción. La matriz R es el diferencial de las restricciones de lazo cerrado y λ es el multiplicador que representa la magnitud de las fuerzas de restricción. Las restricciones de movimiento en cadena cerrada del robot vienen dadas por la matriz L :

$$L(q) = \begin{bmatrix} x_{a_1} + l \cos(a_1) + l \cos(b_1) - x_{a_2} - l \cos(a_2) - l \cos(b_2) \\ y_{a_1} + l \sin(a_1) + l \sin(b_1) - y_{a_2} - l \sin(a_2) - l \sin(b_2) \\ x_{a_1} + l \cos(a_1) + l \cos(b_1) - x_{a_3} - l \cos(a_3) - l \cos(b_3) \\ y_{a_1} + l \sin(a_1) + l \sin(b_1) - y_{a_3} - l \sin(a_3) - l \sin(b_3) \end{bmatrix} = 0 \quad (3.12)$$

Diferenciando, tenemos $\frac{dL(q)}{dt} = \frac{\partial L(q)}{\partial q} \dot{q} = R(q)\dot{q} = 0$ y la matriz R puede escribirse así:

$$R = \begin{bmatrix} -l \sin(a_1) & l \sin(a_2) & 0 & -l \sin(b_1) & l \sin(b_2) & 0 \\ l \cos(a_1) & -l \cos(a_2) & 0 & l \cos(b_1) & -l \cos(b_2) & 0 \\ -l \sin(a_1) & 0 & -l \sin(a_3) & -l \sin(b_1) & 0 & l \sin(b_3) \\ l \cos(a_1) & 0 & -l \cos(a_3) & l \cos(b_1) & 0 & -l \cos(b_3) \end{bmatrix} \quad (3.13)$$

Las fuerzas de restricción $R^T\lambda$ son desconocidas y es difícil medirlas directamente, afortunadamente pueden ser eliminadas por la expresión del espacio nulo de la matriz R . Con la matriz Jacobiana W , se tiene:

$$\dot{z} = W\dot{p} \quad (3.14)$$

donde:

- $\dot{z} = [\dot{a}_1, \dot{a}_2, \dot{a}_3, \dot{b}_1, \dot{b}_2, \dot{b}_3]^T$ representa el vector velocidad de todas las uniones.
- $\dot{p} = [\dot{x}, \dot{y}]^T$ representa el vector velocidad del efector final.

Y la matriz Jacobiana puede definirse como:

$$W = \begin{bmatrix} r_1 \cos(b_1) & r_1 \sin(b_1) \\ r_2 \cos(b_2) & r_2 \sin(b_2) \\ r_3 \cos(b_3) & r_3 \sin(b_3) \\ -r_1 \cos(a_1) & -r_1 \sin(a_1) \\ -r_2 \cos(a_2) & -r_2 \sin(a_2) \\ -r_3 \cos(a_3) & -r_3 \sin(a_3) \end{bmatrix} \quad \text{donde : } r_i = \frac{1}{l \sin(b_i - a_i)} \quad (3.15)$$

Considerando la ecuación de restricción $R\dot{z} = 0$ (Principio de D'Alembert [23]) podemos tener $RW\dot{p}_e = 0$ por la relación Jacobiana. El vector velocidad \dot{p} del efector final contiene coordenadas generalizadas independientes.

Por tanto: $RW = 0$, o su equivalente: $W^T R^T = 0$

De esta forma se puede eliminar el término $R^T \lambda$ y el modelo dinámico en el espacio de tareas puede ser expresado de la siguiente forma:

$$W^T M(z)\ddot{z} + W^T C(z, \dot{z})\dot{z} = W^T \tau - W^T f(z, \dot{z}) + W^T R^T \lambda = W^T \tau - W^T f(z, \dot{z}) \quad (3.16)$$

Derivando \dot{z} de la relación Jacobiana (3.15), tenemos:

$$\ddot{z} = \dot{W}\dot{p} + W\ddot{p}. \quad (3.17)$$

Considerando que la fricción en las uniones pasivas es despreciable, el modelo puede simplificarse. Con τ_a definido como el par en las uniones activas y f_a las fricciones de las mismas, entonces $W^T \tau = S^T \tau_a$ y $W^T f = S^T f_a$. S es la matriz Jacobiana entre las velocidades del efector final y la de las tres uniones activas, definida como:

$$S = \begin{bmatrix} r_1 \cos b_1 & r_1 \sin b_1 \\ r_2 \cos b_2 & r_2 \sin b_2 \\ r_3 \cos b_3 & r_3 \sin b_3 \end{bmatrix},$$

y sustituyendo (3.15) y (3.16) en (3.17) se llega a:

$$\widetilde{M}(z)\ddot{p} + \widetilde{C}(z, \dot{p})\dot{p} = S^T \tau - S^T f(\dot{z}) \quad (3.18)$$

donde:

- $\widetilde{M} = W^T M W$ representa la matriz de inercia del espacio de tareas.

- $\tilde{C} = W^T(M\dot{W} + CW)$ representa la matriz de fuerzas centrífugas en el espacio de tareas.

De la misma manera que en el sistema de cadena cinemática abierta, las matrices \tilde{M} y \tilde{C} cumplen las propiedades:

- \tilde{M} es una matriz simétrica definida positiva.
- $\frac{d\tilde{M}}{dt} - 2\tilde{C}$ es una matriz antisimétrica

Es importante resaltar que el sistema es de grado relativo dos, dado que la matriz de par de fuerzas τ , que en nuestro esquema de control representa las señales de los actuadores, aparece explícitamente con la segunda derivada de la matriz de ángulos, que para propósitos de este trabajo serán las salidas.

3.2.5. Modelo en espacio de estados

Con el objetivo de aplicar las técnicas y análisis de control al sistema, se transforma la ecuación (3.18) del modelo a la forma de Cauchy. Se considera el cambio de variables:

$$\begin{aligned}x_1 &= p_x \\x_2 &= p_y \\x_3 &= \dot{p}_x \\x_4 &= \dot{p}_y.\end{aligned}$$

Y por lo tanto, el modelo queda expresado de la siguiente manera:

$$\dot{X} = A(z, \dot{z})x + B(z)u \quad (3.19)$$

$$Y = (a_1, a_2, a_3)^T \quad (3.20)$$

Donde:

- $X = (x_1, x_2, x_3, x_4)^T$ Es el vector de estados (posiciones y velocidades del efector final).
- $Y = (a_1, a_2, a_3)^T$ Es el vector de mediciones (posición angular de las uniones activas).

- $u = (u_1, u_2, u_3)$ Es el control expresado como el par de fuerzas de los actuadores.

Y las matrices A y B vienen dadas por las expresiones siguientes:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & A_1[1, 1] & A_1[1, 2] \\ 0 & 0 & A_1[2, 1] & A_1[2, 2] \end{bmatrix}, A_1 = -(\tilde{M})^{-1}\tilde{C}, \quad (3.21)$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ B_1[1, 1] & B_1[1, 2] & B_1[1, 3] \\ B_1[2, 1] & B_1[2, 2] & B_1[1, 3] \end{bmatrix}, B_1 = -(\tilde{M})^{-1}S^T. \quad (3.22)$$

3.3. Definición del error de seguimiento

Considerando $q_e^d(t)$ como la trayectoria deseada del efector final en términos de la posición angular de las articulaciones activas del robot, y $q_e(t)$ como el ángulo real que detecta el codificador de los servomotores. Entonces se puede definir al error de seguimiento como:

$$e = q_e^d - q_e. \quad (3.23)$$

En capítulos posteriores, cuando se hable del error se estará haciendo referencia al error de seguimiento definido en la ecuación (3.23).

3.4. Estabilidad en lazo cerrado con control PD

Para mostrar las propiedades de la ley de control nominal (PD) del sistema en operación, consideremos la siguiente función candidata de Lyapunov, basada en [13]:

$$V = \frac{1}{2}\dot{z}^T M(z)\dot{z} + \frac{1}{2}\tilde{z}^T K_p \tilde{z}, \quad (3.24)$$

de la cual

- $\frac{1}{2}\dot{z}^T M(z)\dot{z}$ es la energía cinética del robot.

3. MODELO Y DESCRIPCIÓN DEL ROBOT

- $\frac{1}{2}\dot{z}^T K_p \tilde{z}$ es la realimentación proporcional.

Analizando la derivada temporal de V , resulta:

$$\dot{V} = \dot{z}^T M(z)\ddot{z} + \frac{1}{2}\dot{z}^T \dot{M}(z)\dot{z} - \dot{z}^T K_p \tilde{z}.$$

Resolviendo para $M(z)\ddot{z}$;

$$\begin{aligned}\dot{V} &= \dot{z}^T (u - C(z, \dot{z})\dot{z}) + \frac{1}{2}\dot{z}^T \dot{M}(z)\dot{z} - \dot{z}^T K_p \tilde{z} \\ &= \dot{z}^T (u - K_p \tilde{z}) + \frac{1}{2}\dot{z}^T \underbrace{(\dot{M}(z) - 2C(z, \dot{z}))\dot{z}}_0 \\ &= \dot{z}^T (u - K_p \tilde{z}),\end{aligned}$$

donde en la última desigualdad se ha utilizado el hecho de que $\dot{M}(z) - 2C(z, \dot{z})$ es antisimétrica. Sustituyendo la ley de control $u = K_p \tilde{z} + K_d \dot{\tilde{z}}$, tenemos con $K_d > 0$:

$$\dot{V} = -\dot{z}^T K_d \dot{\tilde{z}} \leq 0,$$

usualmente se toma en cuenta un término gravitacional $g(z)$ o, para nuestro caso, podemos añadir otro tipo de perturbaciones, usando la notación $\delta(z, \dot{z})$ (fricción seca-viscosa) por lo que se tiene

$$\dot{V} = \dot{z}^T (u - \delta(t, z, \dot{z}) - K_p \tilde{z}).$$

Controladores a utilizar

Los modos deslizantes han mostrado ser una estrategia de control efectiva cuando se desea controlar un sistema con incertidumbres, no obstante, debido a su naturaleza discontinua, su implementación se ve afectada por oscilaciones de alta frecuencia conocidas como *chattering*. El objetivo principal es proponer un control robusto a las perturbaciones pero con atenuación de *chattering*.

4.1. Objetivo de control

Lo que se busca al implementar la técnica de modos deslizantes en el sistema en estudio es que el robot mejore significativamente la precisión de seguimiento de una trayectoria. Esta se calcula con anterioridad mediante un ciclo ‘for’, para formar una curva paramétrica a la cual hará un seguimiento punto a punto, este método es conocido como ‘tracking’. Dicho seguimiento de trayectorias se efectúa siempre dentro del espacio de trabajo, en donde la posición del robot no presentará indeterminaciones.

4.2. Estrategia de control

Se tiene el siguiente sistema mecánico de segundo orden y grado relativo dos, que se encuentra bajo incertidumbres o perturbaciones debidas a dinámicas no modeladas. A partir de las ecuaciones de Lagrange se tiene.

$$M(z)\ddot{z} + C(z, \dot{z})\dot{z} + f(\dot{z}) = \tau + \delta(t, z, \dot{z}) \quad (4.1)$$

donde:

- M es la matriz de inercia
- C es la matriz de fuerzas centrífugas y de Coriolis (en nuestro caso sólo se cuenta con fuerzas centrífugas).
- f representa la fuerza de fricción.
- τ_i es el par de fuerzas del actuador A_i

y se desea diseñar una ley de control para seguimiento de trayectoria. Por lo que se propone el siguiente controlador:

$$u(t) = u_0(t) + u_1(t) \quad (4.2)$$

donde el control $u_0 \in \mathbb{R}^3$ es el control nominal de tipo Proporcional-Derivativo (PD), el cual es propio del sistema; este tiene el objetivo de realizar el seguimiento de una trayectoria deseada (conocida), además de compensar las dinámicas modeladas en el capítulo anterior. Y el control $u_1 \in \mathbb{R}^3$ es la parte que garantiza la compensación de incertidumbres (fuerzas de fricción) desconocidas del robot en las uniones activas. Cabe mencionar que u_1 es alguno de los controladores continuos por modos deslizantes que se proponen en este capítulo; dichas señales continuas de control están basadas en controladores modernos de alto orden (ver Capítulo 2), cuya estructura se explicará a mayor detalle más adelante.

4.3. Control nominal

En este apartado se expone el control nominal que posee el sistema en estudio, es necesario tomar en cuenta el trabajo que desempeña y su estructura para poder implementar el segundo controlador por modos deslizantes en la ley de control. Dado que el diseño del PD no es prioridad en este trabajo, únicamente se mencionarán los conceptos básicos del mismo para posteriormente abordar la ley de control robusto por modos deslizantes.

La ley de control que rige el comportamiento de cada uno de los actuadores viene dada por la siguiente expresión de un controlador proporcional y derivativo (PD):

$$u_0 = K_p(z_d - z) + K_d(\dot{z}_d - \dot{z}) \quad (4.3)$$

Donde:

- z_d es el vector de los ángulos deseados para la trayectoria a seguir.
- z son los ángulos en tiempo real de cada servomotor.
- \dot{z}_d y \dot{z} son las respectivas derivadas de z_d y z .

4.4. Control robusto ante perturbaciones

Las ventajas de los controladores no lineales se mencionaron en el Capítulo 1 y Capítulo 2 de este trabajo de manera general. En las siguientes secciones se explicará de manera particular cada una de las leyes de control por modos deslizantes implementadas experimentalmente en el Capítulo 6. Debido a que el controlador PD, que en primera instancia está dentro del sistema, no puede garantizar convergencia del error a cero en presencia de la fricción e incertidumbres del modelo matemático (ver [12]), se necesitan utilizar las siguientes técnicas de control robusto.

4.4.1. Estructura del control no lineal

Los modos deslizantes de suavidad arbitraria pueden lograrse incrementando artificialmente el grado relativo del sistema, atenuando significativamente el efecto del *chattering*. Por ejemplo, la función continua de control puede obtenerse si un ‘control virtual’ en términos de la derivada del control se diseña mediante un SMC. En este caso, la señal de control será continua, dado que es igual a la integral de la función de conmutación a alta frecuencia. En el caso de dinámicas no modeladas la función del modo deslizante conmutará a frecuencia más baja [16]. El diseño del SMC en términos de la derivada de la función de control dará paso a la atenuación del *chattering*.

En la ecuación (4.4) se ejemplifica un sistema mecánico de segundo orden y grado relativo dos, si se toma como la salida a x_1 .

$$\begin{cases} \dot{x}_1 = x_2 & x_1(0) = x_{10} \\ \dot{x}_2 = u + f(x_1, x_2, t) & x_2(0) = x_{20} \\ \dot{u} = \nu & u(0) = 0. \end{cases} \quad (4.4)$$

Donde x_1 y x_2 son posición y velocidad de una masa, u es la fuerza de control, y el término de la perturbación $f(x_1, x_2, t)$, el cual puede constar de las fricciones seca y viscosa así como de alguna otra fuerza desconocida, se asume acotada, i.e., $|f(x_1, x_2, t)| \leq L > 0$ y en adición que es suave con derivada acotada $|\dot{f}(x_1, x_2, t)| \leq \hat{L}$.

Para este ejemplo el controlador por modos deslizantes se encuentra en ν , aunque realmente se esté aplicando la integral del mismo en el sistema original.

El ‘control virtual’ aplicado deberá ser de un orden mayor al grado relativo del sistema. Para el ejemplo en (4.4), ν tiene que ser un HOSM de orden tres para poder asegurar convergencia asintótica de los estados. Este procedimiento se puede repetir si se desea utilizar un controlador por modos deslizantes de cuarto orden, definiendo en ese caso el HOSM como la segunda derivada del control real que se inyecta en el sistema.

4.4.2. Super-Twisting

Uno de los algoritmos por modos deslizantes de segundo orden más estudiados es el llamado Super-Twisting [1]. Este controlador se implementará para poder realizar una comparación entre el uso de un SOSM y alguno de los HOSM presentados más adelante. Cabe señalar que, para el caso del Super-Twisting, se requiere diseñar una superficie de deslizamiento que asegure su convergencia en tiempo finito.

La estructura del controlador se tomó de [16], y tiene la forma:

$$\begin{aligned} u &= -k_1 |\sigma|^{1/2} \text{sign}(\sigma) + z \\ \dot{z} &= -k_2 \text{sign}(\sigma), \end{aligned} \tag{4.5}$$

en donde $k_1 = 1.5\sqrt{C}$ y $k_2 = 1.1C$, C es la cota superior de la derivada de la perturbación, i.e. $|\dot{f}(x_1, x_2, t)| \leq C$.

4.4.3. Controladores por modos deslizantes de orden arbitrario

A continuación se definen las dos más conocidas familias de controladores por modos deslizantes de orden r , tomadas de [3] y [24]. La forma de los controladores

$$u = -\alpha \Psi_{r-1,r}(\sigma, \dot{\sigma}, \dots, \sigma^{(r-1)}) \quad (4.6)$$

está definida por procedimientos recursivos, tienen magnitud $\alpha > 0$, y resuelven el problema general de regulación de salida para un sistema del tipo

$$\sigma^r = h(t, x) + g(t, x) u \quad (4.7)$$

Los parámetros de los controladores pueden escogerse de manera previa para cada grado relativo r . Únicamente la magnitud α tiene que ser ajustada, muy convenientemente por simulación en computadora, para así evitar estimaciones largas, redundantes y complicadas. Se debe considerar también que α tiene que ser negativa con $\frac{\partial}{\partial u}(\sigma^{(r)}) < 0$. De igual manera las constantes $\beta_1, \dots, \beta_r > 0$ presentadas en las secciones siguientes son parámetros del controlador, los cuales definen la velocidad de convergencia.

4.4.3.1. Controladores deslizantes Anidados

El siguiente procedimiento define los controladores deslizantes de orden r ‘Anidados’, basados en una estructura pseudo-anidada de modos deslizantes de orden 1. Con $q \in \mathbb{N}$ y $q > 1$. Los controladores están contruidos por el siguiente procedimiento recursivo:

$$N_{i,r} = (|\sigma|^{q/r} + |\dot{\sigma}|^{q/(r-1)} + \dots + |\sigma^{(i-1)}|^{q/(r-i+1)})^{1/q} \quad (4.8)$$

$$\Psi_{0,r} = \text{sign}(\sigma), \quad \Psi_{i,r} = \text{sign}(\sigma^{(i)} + \beta_i N_{i,r} \Psi_{i-1,r}). \quad (4.9)$$

Los que se muestran a continuación son los controladores anidados por modos deslizantes para $r \leq 4$ con q siendo el mínimo múltiplo de $1, \dots, r$:

1. $u = -\alpha \text{sign}(\sigma)$
2. $u = -\alpha \text{sign}(\dot{\sigma} + |\sigma|^{1/2} \text{sign}(\sigma))$

3. $u = -\alpha \operatorname{sign} \left(\ddot{\sigma} + 2 (|\dot{\sigma}|^3 + |\sigma|^2)^{1/6} \times \operatorname{sign} (\dot{\sigma} + |\sigma|^{2/3} \operatorname{sign}(\sigma)) \right)$
4. $u = -\alpha \operatorname{sign} \left(\ddot{\sigma} + 3 (\ddot{\sigma}^6 + \dot{\sigma}^4 + |\sigma|^3)^{1/12} \times \operatorname{sign} \left(\ddot{\sigma} + (\dot{\sigma}^4 + |\sigma|^3)^{1/6} \operatorname{sign} (\dot{\sigma} + 0.5|\sigma|^{3/4} \operatorname{sign}(\sigma)) \right) \right)$

A estos controladores se les puede dar una explicación intuitiva e inexacta basada en modos deslizantes convencionales anidados recursivamente. Esta se encuentra detallada en [3], donde se explica el establecimiento de un modo deslizante para un controlador de tercer orden, i.e. $\sigma = \dot{\sigma} = \ddot{\sigma} = 0$.

4.4.3.2. Controladores deslizantes Cuasi-Continuos

La desventaja obvia del controlador por modo deslizante Anidado es el salto en alta frecuencia de las trayectorias transitorias cuando se acerca al HOSM. Esto puede causar problemas en la práctica.

Los controladores cuasi-continuos se propusieron en 2005, tomados de [24], en donde Levant explica que esta ley de control debe ser discontinua al menos en el HOSM.

$$\sigma = \dot{\sigma} = \dots = \sigma^{(r-1)} = 0 \quad (4.10)$$

Un controlador por modos deslizantes de orden r de la forma (4.6) es llamado *Cuasi-Continuo*, si es continuo en todos lados excepto en la ecuación (4.10). En la práctica con $r > 1$, tal ley de control permanece continua, esto dado el ruido, retardos, y otras imperfecciones, y las igualdades de r en la ecuación (4.10) nunca se mantienen simultáneas.

El siguiente procedimiento define una familia de controladores cuasi-continuos:

$$\begin{aligned} \varphi_{0,r} &= \sigma, \quad N_{0,r} = |\sigma|, \quad \Psi_{0,r} = \frac{\varphi_{0,r}}{N_{0,r}} = \operatorname{sign}(\sigma) \\ \varphi_{i,r} &= \sigma^{(i)} + \beta_i N_{i-1,r}^{(r-1)/(r-i+1)} \Psi_{i-1,r} \\ N_{i,r} &= |\sigma^{(i)}| + \beta_i N_{i-1,r}^{(r-i)/(r-i+1)}, \quad \Psi_{i,r} = \frac{\varphi_{i,r}}{N_{i,r}}. \end{aligned}$$

Los siguientes son controladores cuasi-continuos con $r < 4$, los cuales tienen una β_i probada en simulaciones, tomada de [24]:

1. $u = -\alpha \operatorname{sign}(\sigma)$
2. $u = -\alpha (\dot{\sigma} + |\sigma|^{1/2} \operatorname{sign}(\sigma)) / (|\dot{\sigma}| + |\sigma|^{1/2})$
3. $u = -\alpha \frac{\ddot{\sigma} + 2(|\dot{\sigma}| + |\sigma|^{2/3})^{-1/2} (\dot{\sigma} + |\sigma|^{2/3} \operatorname{sign}(\sigma))}{|\ddot{\sigma}| + 2(|\dot{\sigma}| + |\sigma|^{2/3})^{1/2}}$
4. $u = -\alpha \varphi_{3,4} / N_{3,4}$ donde

$$\varphi_{3,4} = \ddot{\sigma} + 3 \left[\ddot{\sigma} + (|\dot{\sigma}| + 0.5|\sigma|^{3/4})^{-1/3} (\dot{\sigma} + 0.5|\sigma|^{3/4} \operatorname{sign}(\sigma)) \right] \times \\ \left[|\ddot{\sigma}| + (|\dot{\sigma}| + 0.5|\sigma|^{3/4})^{2/3} \right]^{1/2}$$

y

$$N_{3,4} = |\ddot{\sigma}| + 3 \left[|\ddot{\sigma}| + (|\dot{\sigma}| + 0.5|\sigma|^{3/4})^{2/3} \right]^{1/2}.$$

Es fácil ver que el conjunto de parámetros β_i son los mismos tanto para el controlador cuasi-continuo como para el anidado con $r < 4$. Se debe notar que mientras se agranda α incrementa la clase de sistemas de la ecuación (4.7), para el cual el controlador es aplicable, los parámetros β_i se sintonizan para proveer la velocidad de convergencia requerida. En adición a la reducción del *chattering*, otra ventaja de estos controladores es la simplicidad en términos de ajuste de sus coeficientes.

La idea de los controladores cuasi-continuos es una generalización de la idea de los controladores por modos deslizantes de orden 2, ver Capítulo 2.

4.4.3.3. Diferenciadores exactos y robustos de orden arbitrario

Cualquier controlador homogéneo deslizante de orden r puede ser complementado por un diferenciador de orden $(r - 1)$ produciendo un controlador con salida realimentada. Para preservar la exactitud demostrada, estabilidad en tiempo finito, y las propiedades asintóticas correspondientes, la manera natural de calcular $\dot{\sigma}, \dots, \sigma^{(r-1)}$ en tiempo real es por medio de un diferenciador *homogéneo* exacto y robusto, convergente en tiempo finito.

Si se deja a la señal de entrada $f(t)$ sea una función definida en $[0, \infty)$ que consiste de un ruido acotado de tipo medible *Lebesgue* con características desconocidas y de una señal base desconocida $f_0(t)$, para la cual la derivada de orden k tiene una constante Lipschitz conocida $L > 0$. El problema de encontrar una estimación robusta en tiempo real de $\dot{f}_0(t), \ddot{f}_0(t), \dots, f_0^{(k)}(t)$ las cuales son exactas en la ausencia de ruidos mensurables es resuelto por el siguiente diferenciador propuesto por Levant

$$\begin{aligned}
 \dot{z}_0 &= v_0, v_0 = -\lambda_k L^{1/(k+1)} |z_0 - f(t)|^{k/(k+1)} \text{sign}(z_0 - f(t)) + z_1 \\
 \dot{z}_1 &= v_1, v_1 = -\lambda_{k-1} L^{1/k} |z_1 - v_0|^{(k-1)/k} \text{sign}(z_1 - v_0) + z_2 \\
 &\vdots \\
 \dot{z}_{k-1} &= v_{k-1}, v_{k-1} = -\lambda_1 L^{1/2} |z_{k-1} - v_{k-2}|^{1/2} \text{sign}(z_{k-1} - v_{k-2}) + z_k \\
 \dot{z}_k &= -\lambda_0 L \text{sign}(z_k - v_{k-1})
 \end{aligned} \tag{4.11}$$

Si los parámetros $\lambda_0, \lambda_1, \dots, \lambda_k > 0$ se escogen adecuadamente, las siguientes igualdades son verdaderas en la ausencia de ruido de entrada, después de un proceso transitorio en tiempo finito:

$$z_0 = f_0(t), \dots, z_i = v_{i-1} = f_0^{(i-1)}(t), \quad i = 1, 2, \dots, k. \tag{4.12}$$

Se debe hacer notar que el diferenciador tiene estructura recursiva. Una vez que $\lambda_0, \lambda_1, \dots, \lambda_{k-1}$ se escogen adecuadamente para el diferenciador de orden $(k-1)$ con constante Lipschitz L , sólo un parámetro λ_k necesita ser sintonizado para el diferenciador de orden k con la misma constante Lipschitz y este parámetro sólo tiene que ser lo suficientemente grande. Cualquier $\lambda_0 > 1$ puede ser usada para iniciar con este proceso.

Teorema Para cualquier $\lambda_0 > 1$ existe una secuencia positiva $\{\lambda_n\}$, tal que para cada k natural los parámetros $\lambda_0, \lambda_1, \dots, \lambda_k$ provee convergencia en tiempo finito del diferenciador de orden k de la ecuación (4.11).

Una opción posible de los parámetros del diferenciador para $k \leq 5$ es $\lambda_0 = 1.1, \lambda_1 = 1.5, \lambda_2 = 3, \lambda_3 = 5, \lambda_4 = 8, \lambda_5 = 12$. Otra posible elección es $\lambda_0 = 1.1, \lambda_1 = 1.5, \lambda_2 = 2, \lambda_3 = 3, \lambda_4 = 5, \lambda_5 = 8$.

Ejemplo 1

El siguiente es un diferenciador de tercer orden:

$$\begin{aligned}
\dot{z}_0 &= v_0, v_0 = -3L^{1/4} |z_0 - f(t)|^{3/4} \text{sign}(z_0 - f(t)) + z_1 \\
\dot{z}_1 &= v_1, v_1 = -2L^{1/3} |z_1 - v_0|^{2/3} \text{sign}(z_1 - v_0) + z_2 \\
\dot{z}_2 &= v_2, v_2 = -1.5L^{1/2} |z_2 - v_1|^{1/2} \text{sign}(z_2 - v_1) + z_3 \\
\dot{z}_3 &= -1.1L \text{sign}(z_3 - v_2), \quad |f^{(4)}(t)| \leq L.
\end{aligned} \tag{4.13}$$

Se aplicó con $L = 1$ para diferenciar la función

$$f(t) = \sin(0.5t) + \cos(0.5t) \quad |f^{(6)}(t)| \leq 1, \tag{4.14}$$

el cual fue programado en lenguaje C++. El código se encuentra en el apéndice correspondiente, que es el mismo que se implementó para las pruebas experimentales.

Los valores iniciales del diferenciador se toman igual a cero. En la práctica es razonable tomar los valores iniciales de z_0 igual al valor de la muestra actual de $f(t)$, haciendo significativamente más corto el transitorio. La convergencia del diferenciador se muestra en la Figura 4.1. La tercera derivada no es exacta debido a las restricciones del software.

Ejemplo 2

Utilizando un diferenciador de Levant de cuarto orden con $L = 1$ se tiene:

$$\begin{aligned}
\dot{z}_0 &= v_0, v_0 = -5L^{1/5} |z_0 - f(t)|^{4/5} \text{sign}(z_0 - f(t)) + z_1 \\
\dot{z}_1 &= v_1, v_1 = -3L^{1/4} |z_1 - v_0|^{3/4} \text{sign}(z_1 - v_0) + z_2 \\
\dot{z}_2 &= v_2, v_2 = -2L^{1/3} |z_2 - v_1|^{2/3} \text{sign}(z_2 - v_1) + z_3 \\
\dot{z}_3 &= v_3, v_3 = -1.5L^{1/2} |z_3 - v_2|^{1/2} \text{sign}(z_3 - v_2) + z_4 \\
\dot{z}_4 &= -1.1L \text{sign}(z_4 - v_3), \quad |f^{(5)}(t)| \leq L.
\end{aligned} \tag{4.15}$$

Haciendo pruebas sobre la misma función del Ejemplo 1, y graficando los datos se obtienen las gráficas de la Figura 4.2.

Se puede observar de los ejemplos anteriores que las primeras derivadas resultantes convergen al valor real, de mejor manera, entre más alto sea el orden del diferenciador,

4. CONTROLADORES A UTILIZAR

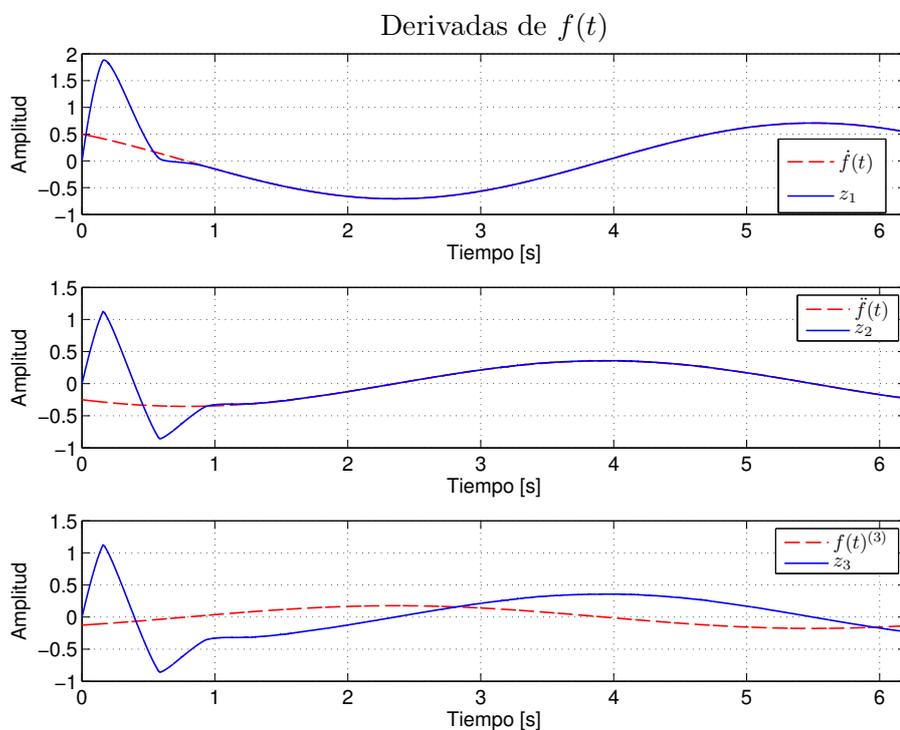


Figura 4.1: Diferenciador exacto y robusto de tercer orden

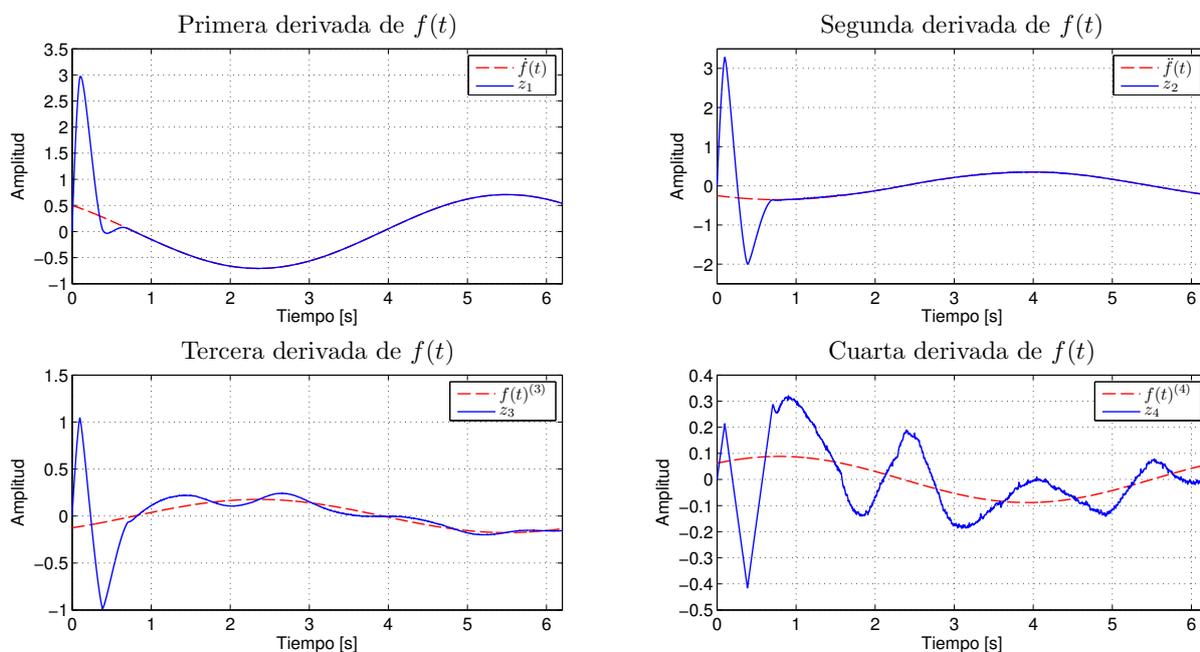


Figura 4.2: Diferenciador exacto y robusto de cuarto orden

aunque tiene la desventaja de que esto puede llegar a consumir más recursos computacionales, dado que las operaciones matemáticas aumentan. Para motivos experimentales de este trabajo se ocuparon los diferenciadores de cuarto orden en combinación con controladores de tercer orden, y para los controladores de cuarto orden se utilizó un diferenciador de quinto orden. Esto debido a que se obtienen las primeras derivadas requeridas de forma más suave, sin comprometer el tiempo de operación del esquema de control programado.

4.4.4. Sintonización de los controladores

La sintonización de los parámetros de los controladores es necesaria para regular la velocidad de convergencia. En [25] Levant plantea una técnica de sintonización para los HOSM, la cual se presenta a continuación.

4.4.4.1. Sintonización paramétrica

Los parámetros del controlador presentados en la sección del HOSM Cuasi-Continuo, proveen una solución formal al problema planteado. Sin embargo, en la práctica comúnmente se necesita ajustar la velocidad de convergencia, ya sea haciéndola más lenta o más rápida (relajando la carga en los actuadores) o para acelerarla a fin de que cumpla algunos requerimientos del sistema. Hay que hacer notar que, en este contexto, agrandar excesivamente la magnitud del parámetro α no acelera la convergencia, únicamente incrementa el *chattering*, mientras que su reducción puede llevar a la pérdida de convergencia.

Una mejor aproximación es tomar el controlador

$$u = \lambda^r \alpha \Psi_{r-1,r} \left(\sigma, \dot{\sigma}/\lambda, \dots, \sigma^{(r-1)}/\lambda^{r-1} \right), \lambda > 0 \quad (4.16)$$

en vez de

$$u = -\lambda \Psi_{r-1,r} \left(\sigma, \dot{\sigma}, \dots, \sigma^{(r-1)} \right), \lambda > 0 \quad (4.17)$$

lo cual provee aproximadamente una reducción de λ veces en el tiempo de convergencia.

En el caso de los controladores cuasi-continuos la forma del controlador se conserva. Los nuevos parámetros $\tilde{\beta}_1, \dots, \tilde{\beta}_{r-1}, \tilde{\alpha}$ son calculados de acuerdo a las fórmulas $\tilde{\beta}_1 = \lambda\beta_1, \tilde{\beta}_2 = \lambda^{r/(r-1)}\beta_2, \dots, \tilde{\beta}_{r-1} = \lambda^{r/2}\beta_{r-1}, \tilde{\alpha} = \lambda_r\alpha$.

4.4.5. Presentación de los algoritmos a programar

En esta sección se formalizan los algoritmos HOSM que se implementan para las pruebas experimentales del Capítulo 6. Utilizando las estrategias y estructuras presentadas a lo largo de este capítulo; considerando también la sintonización paramétrica.

Controladores HOSM de tercer orden:

- **Anidado**

$$\dot{u} = -\alpha \operatorname{sign} \left(\ddot{\sigma} + 2\lambda^{3/2} (|\dot{\sigma}|^3 + |\sigma|^2)^{1/6} \times \operatorname{sign} (\dot{\sigma} + \lambda|\sigma|^{2/3}\operatorname{sign}(\sigma)) \right)$$

- **Cuasi-Continuo**

$$\dot{u} = -\alpha \frac{\ddot{\sigma} + 2\lambda^{3/2} (|\dot{\sigma}| + \lambda|\sigma|^{2/3})^{-1/2} (\dot{\sigma} + \lambda|\sigma|^{2/3}\operatorname{sign}(\sigma))}{|\ddot{\sigma}| + 2\lambda^{3/2} (|\dot{\sigma}| + \lambda|\sigma|^{2/3})^{1/2}}$$

Controladores HOSM de cuarto orden:

- **Anidado**

$$\ddot{u} = -\alpha \operatorname{sign} \left(\ddot{\sigma} + 3\lambda^2 (\ddot{\sigma}^6 + \dot{\sigma}^4 + |\sigma|^3)^{1/12} \times \operatorname{sign} \left(\ddot{\sigma} + \lambda^{4/3} (\dot{\sigma}^4 + |\sigma|^3)^{1/6} \operatorname{sign} (\dot{\sigma} + 0.5\lambda|\sigma|^{3/4}\operatorname{sign}(\sigma)) \right) \right)$$

- **Cuasi-Continuo**

$\ddot{u} = -\alpha \varphi_{3,4}/N_{3,4}$, donde

$$\varphi_{3,4} = \ddot{\sigma} + 3\lambda^2 \left[\ddot{\sigma} + \lambda^{4/3} (|\dot{\sigma}| + 0.5\lambda|\sigma|^{3/4})^{-1/3} (\dot{\sigma} + 0.5\lambda|\sigma|^{3/4}\operatorname{sign}(\sigma)) \right] \times \left[|\ddot{\sigma}| + \lambda^{4/3} (|\dot{\sigma}| + 0.5\lambda|\sigma|^{3/4})^{2/3} \right]^{1/2}$$

y

$$N_{3,4} = |\ddot{\sigma}| + 3\lambda \left[|\ddot{\sigma}| + \lambda^{4/3} (|\dot{\sigma}| + 0.5\lambda|\sigma|^{3/4})^{2/3} \right]^{1/2}$$

Simulaciones

En este capítulo se muestran los resultados de trabajar con las ecuaciones del modelo matemático en lazo cerrado para el seguimiento de trayectorias del robot paralelo, teniendo como objetivo que el error e sea igual a cero en un tiempo $t > 0$. Utilizando para esto la ayuda de la herramienta *Simulink* incluida en el software *Matlab*. Los valores de los parámetros obtenidos para los controladores servirán más adelante como un punto de partida para la experimentación real.

5.1. Parámetro dinámicos usados

Para la simulación se necesitan medir los parámetros físicos que describen al sistema real. Los parámetros dinámicos de los eslabones (L_{ai} y L_{bi}) como las masas, momentos de inercia relativos a los centros de masa, distancias entre los centros de masas, articulaciones y longitudes pueden ser medidos directamente, como se muestra en la Tabla 5.1. La notación utilizada es la misma que se utilizó en la Figura 3.4 en el capítulo del modelo matemático.

Considerando los valores anteriores, los parámetros α_i , β_i y γ_i de las matrices M y C del modelo matemático (ver Capítulo 3) se muestran en la Tabla 5.2

Por otro lado, para identificar y simular los parámetros de fricción y las dinámicas desconocidas del robot GPM2002 se ocuparon los datos presentados en el trabajo de Zhang y Cong del 2010 en [8], donde se define un vector de par de fuerzas para la fricción como $f_i = [f_{ai} \quad f_{bi}]^T$ con $i = 1, 2, 3$, donde se ignora el par de fricción en las

	Masa (Kg)	Longitud (m)	Distancia entre el centro de masa y la articulación (m)	Momento de inercia relativo al centro de masa (Kg m ²)
L_{a1}	1.2525	0.2440	0.1156	0.0124
L_{a2}	1.3663	0.2440	0.0657	0.0122
L_{a3}	1.3663	0.2440	0.0657	0.0122
L_{b1}	1.0771	0.2440	0.1621	0.0098
L_{b2}	0.4132	0.2440	0.1096	0.0036
L_{b3}	0.4132	0.2440	0.1096	0.0036

Tabla 5.1: Parámetros dinámicos reales de los eslabones

Parámetro (Kg m ²)	Valor	Parámetro (Nm) (Kg m ²)	Valor	Parámetro (Kg m ²)	Valor
α_1	0.0932	β_1	0.0381	γ_1	0.0426
α_2	0.0427	β_2	0.0085	γ_1	0.0111
α_3	0.0427	β_3	0.0085	γ_1	0.0111

Tabla 5.2: Valores iniciales de los parámetros dinámicos

uniones pasivas f_{bi} y únicamente el par de fricción en las uniones de los actuadores es considerado. El par de fuerzas de fricción en las articulaciones actuadas puede entonces ser formulado como:

$$f_{ai} = \text{sign}(\dot{q}_{ai}) f_{ci} + f_{vi} \dot{q}_{ai},$$

donde \dot{q} es la velocidad angular en las articulaciones, f_{ci} representa la fricción de Coulomb, y f_{vi} representa el coeficiente de la fricción viscosa. En la Tabla 5.3 se observan las constantes de los valores de fricción utilizadas en la simulación.

Basados en los valores de los parámetros de las tablas 5.1, 5.2 y 5.3 se realizaron las simulaciones mostradas a continuación. Cabe resaltar que únicamente se simuló el seguimiento de trayectoria de una línea recta con un punto inicial en (0.216 , 0.25) y

Parámetro (Ns)	Valor	Parámetro (Nm)	Valor
f_{v1}	2.9936	f_{c1}	0.4976
f_{v2}	2.7617	f_{c2}	0.4570
f_{v3}	2.8771	f_{c3}	0.3006

Tabla 5.3: Valores iniciales de los parámetros de fricción en las articulaciones actuadas

un punto final en (0.316 , 0.15), por lo tanto la distancia de movimiento es de 0.14142 m. Esto es suficiente para poder ver la respuesta del sistema y obtener la sintonización adecuada de las leyes de control.

5.2. Resultados de simulaciones

Los parámetros de simulación para el seguimiento de la trayectoria con los algoritmos HOSM de tercer orden y el Super-Twisting, con una velocidad de 0.025 m/s , son:

- **Tiempo de simulación:** 0 - 5.65 seg.
- **Opciones de solución:**
 - **Tipo:** Fixed-step (Paso fijo)
 - **Método de solución:** ode1 (Euler)
- **Paso de muestreo:** 0.0002

5.2.1. Controlador Super-Twisting

En la Figura 5.1 se muestran las señales de error definidas con anterioridad para cada motor, con un acercamiento en cada una de las gráficas. Se observa que convergen a un valor de cero asintóticamente como era de esperarse. Para cada una de las señales de control de los motores se ve un comportamiento continuo en el tiempo, sin embargo, en los acercamientos se observa que las señales aún tienen algunas oscilaciones a alta frecuencia, aunque de amplitud muy pequeña.

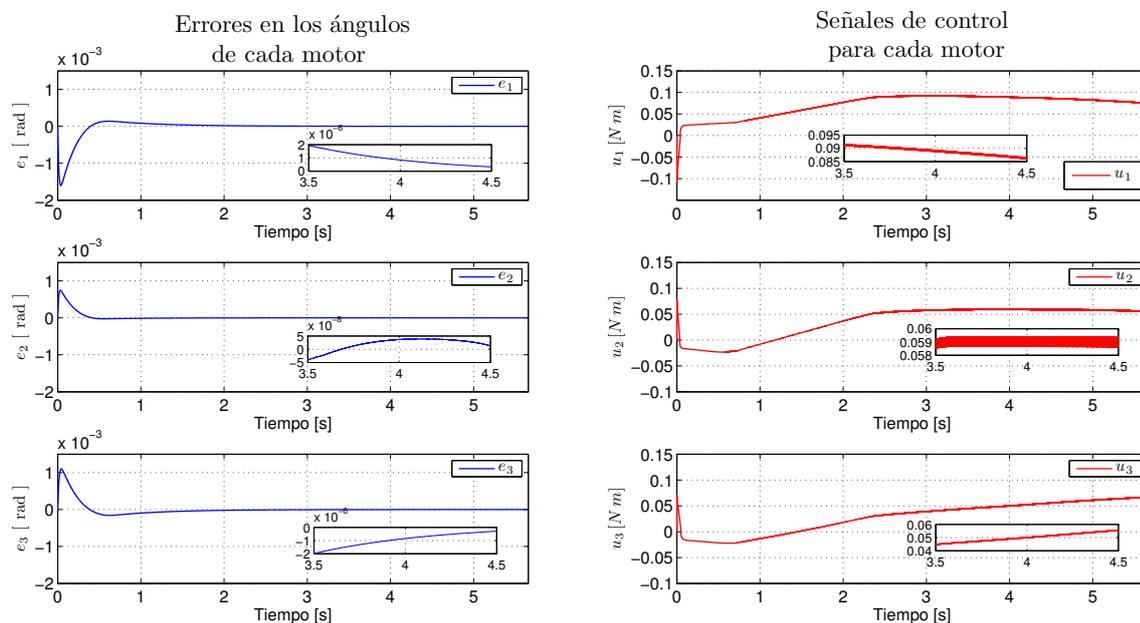


Figura 5.1: Errores en los ángulos de los motores y señales de control para cada motor, Super-Twisting con diferenciador de cuarto orden

Con respecto a la superficie de deslizamiento, en este caso definida como $\sigma = e + \dot{e}$, se muestra que convergen a cero en tiempo finito, con lo que se asegura que ha entrado al modo deslizante, como se observa en la Figura 5.2.

5.2.2. Controlador por Modos Deslizantes de tercer orden

En esta sección se prueban los controladores de tercer orden programados con un diferenciador de Levant de cuarto orden, para obtener la salida de error e y sus dos primeras derivadas \dot{e} y \ddot{e} . Este procedimiento se realizó para las dos familias de HOSM, como se muestra a continuación.

5.2.2.1. Controlador Cuasi-Continuo

En la Figura 5.3 se observa que, al igual que con el Super-Twisting, los errores convergen a cero de manera asintótica, y la señal de control es continua. Sin embargo, como se aprecia en los acercamientos de las gráficas de las señales de control, aún está

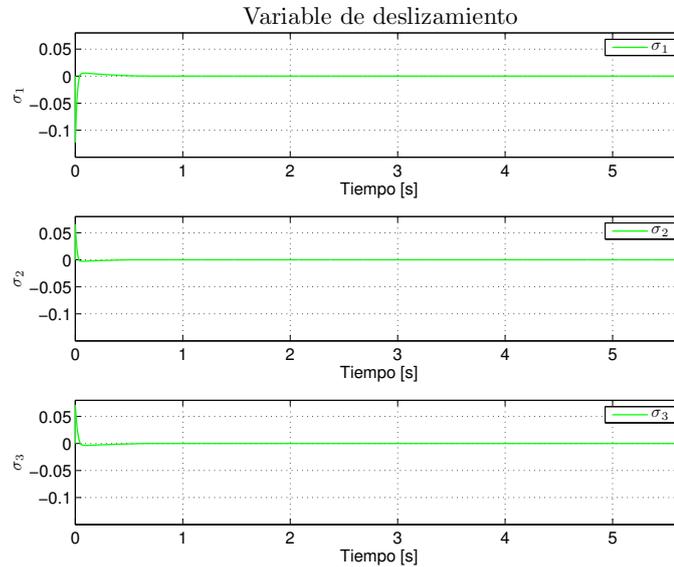


Figura 5.2: Variable de deslizamiento y su primera derivada para cada motor, Super-Twisting con diferenciador de cuarto orden

presente una conmutación en esta señal cuando se llega al modo deslizante, aproximadamente en el segundo 2.8. Este comportamiento oscilatorio, aunque es a una frecuencia menor que con el Super-Twisting, puede afectar en cierta manera la respuesta de los actuadores del robot.

Con respecto a la primera y segunda derivada del error, \dot{e} y \ddot{e} , se observa que convergen a cero en tiempo finito, asegurando así que los diferenciadores están haciendo su trabajo, y que el controlador funciona adecuadamente, ver Figura 5.4.

5.2.2.2. Controlador Anidado

Para el controlador Anidado se tiene un comportamiento similar en los errores convergentes a cero. Sin embargo la señal de control presenta más ‘picos’ en el transitorio comparado con el cuasi-continuo, con lo que se podría decir que es una señal menos suave. La entrada al modo deslizante de la variable de error e se logra aproximadamente en el segundo 1.7, ver Figura 5.5. Esto es, logra la convergencia a cero en un tiempo menor que el cuasi-continuo.

5. SIMULACIONES

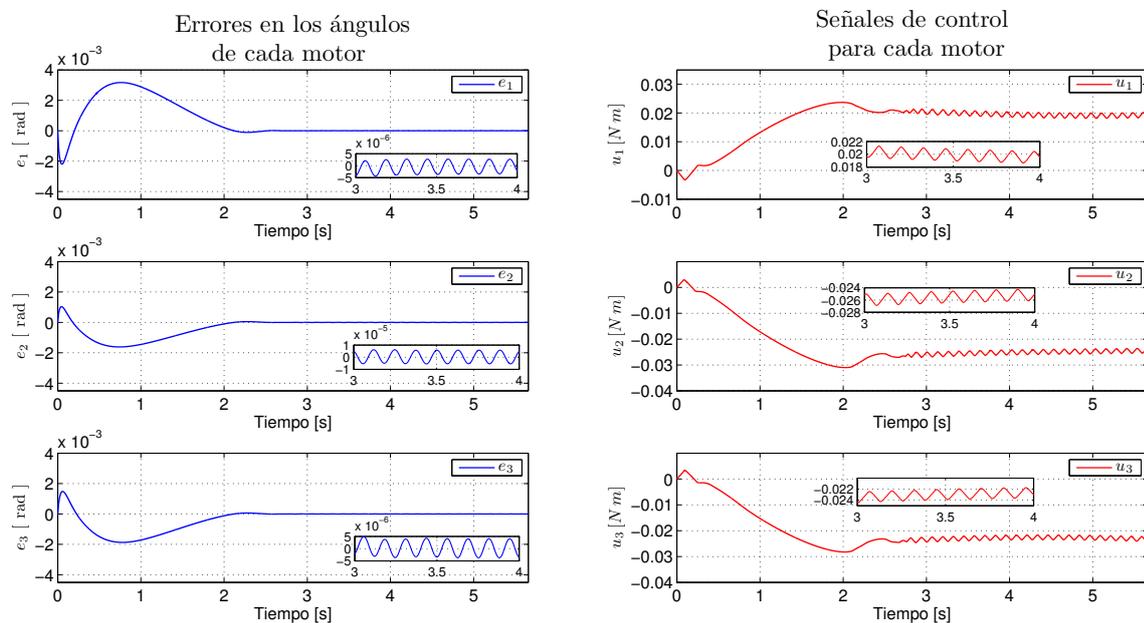


Figura 5.3: Errores en los ángulos de los motores y señales de control para cada motor, Algoritmo Cuasi-Continuo con diferenciador de Levant de cuarto orden

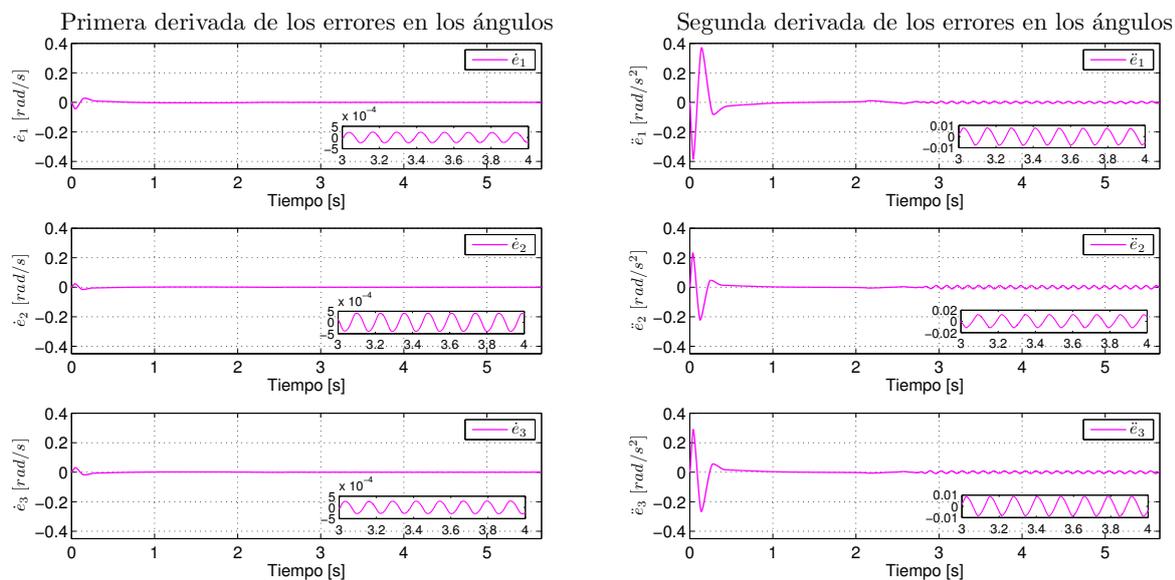


Figura 5.4: Primera y segunda derivada de los errores en los ángulos de los motores, Algoritmo Cuasi-Continuo con diferenciador de Levant de cuarto orden

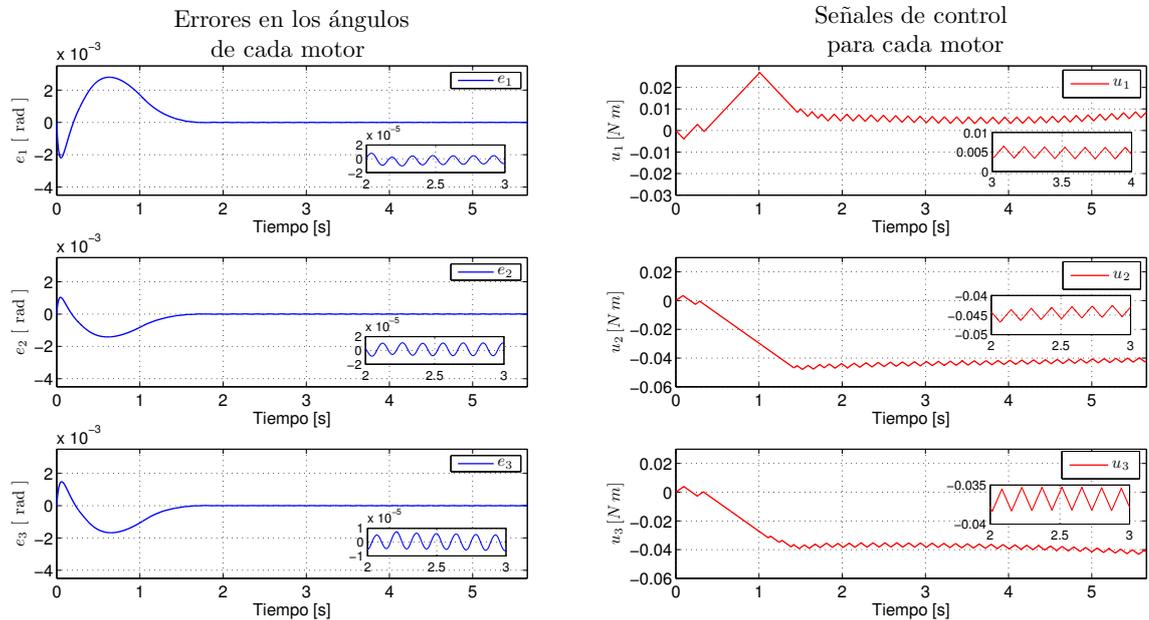


Figura 5.5: Errores en los ángulos de los motores y señales de control para cada motor, Algoritmo Anidado con diferenciador de Levant de cuarto orden

En la Figura 5.6 se observan a las derivadas de los errores converger en tiempo finito a cero, esto gracias a los diferenciadores de cuarto orden.

5.2.3. Controlador por Modos Deslizantes de Cuarto Orden

En las siguientes simulaciones se observa un cambio en la señal de control con respecto a los de tercer orden. El inconveniente de este tipo de leyes de control está en que el tiempo de convergencia de los errores en los motores aumenta drásticamente. Fue por esto que para los HOSM de cuarto orden se utilizó una velocidad de 0.014 m/s , para así poder hacer más largo el experimento, esta vez de 0 a 10.1 seg.

5.2.3.1. Controlador Cuasi-Continuo

El algoritmo cuasi-continuo de cuarto orden presenta una convergencia de los errores a cero en aproximadamente 7 segundos, aunque la señal de control es claramente más suave, ver Figura 5.7. Se observa en los acercamientos de las gráficas que la señal de control no presenta oscilaciones a alta frecuencia, esto debido a la doble integración

5. SIMULACIONES

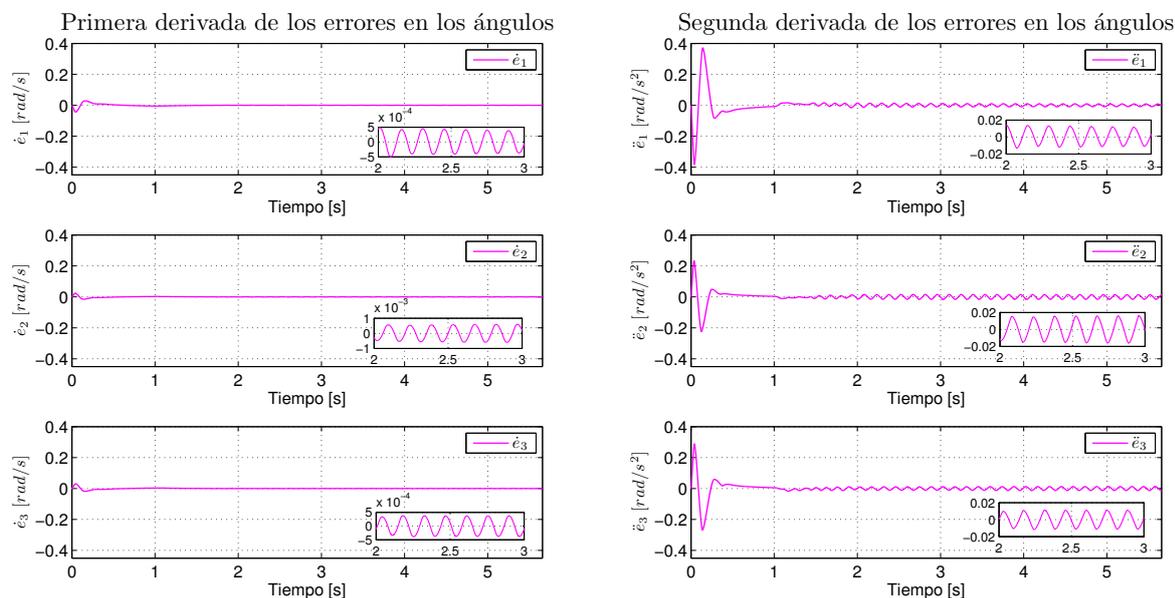


Figura 5.6: Primera y segunda derivada de los errores en los ángulos de los motores, Algoritmo Anidado con diferenciador de Levant de cuarto orden

de la señal real de control, lo que provoca la atenuación del *chattering*.

Las derivadas de los errores para cada motor definidos anteriormente que se observan en la Figura 5.8 y 5.9 (en color magenta), ponen en evidencia que los diferenciadores de Levant de quinto orden funcionan de manera adecuada, asegurando un modo deslizante en $\dot{e} = \ddot{e} = \dddot{e} = 0$, ya que se trata de un HOSM de cuarto orden. A pesar de que existen aún oscilaciones, estas tienen un valor de amplitud muy pequeño.

5.2.3.2. Controlador Anidado

El último controlador para el seguimiento de la trayectoria recta en simulación es el Anidado de cuarto orden. Se observa en 5.10 que, aunque la señal de control cambia con respecto al algoritmo anterior, esta sigue siendo más suave que las señales de los de tercer orden, llevando a los errores de posición a un valor igual a cero en aproximadamente 4 segundos. Es decir que el tiempo de convergencia de la restricción a cero para el Anidado es mejor que con un Cuasi-Continuo.

De igual forma cabe aclarar que aunque la señales de control en rojo parecen ser

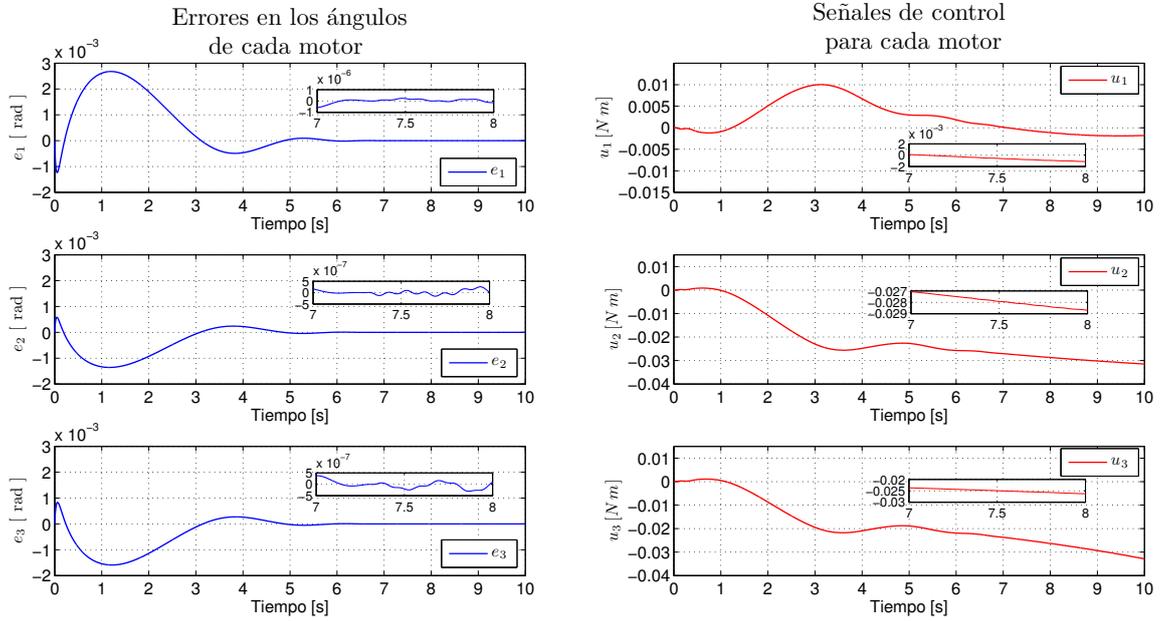


Figura 5.7: Errores en los ángulos de los motores y señales de control para cada motor, Algoritmo Cuasi-Continuo con diferenciador de Levant de quinto orden

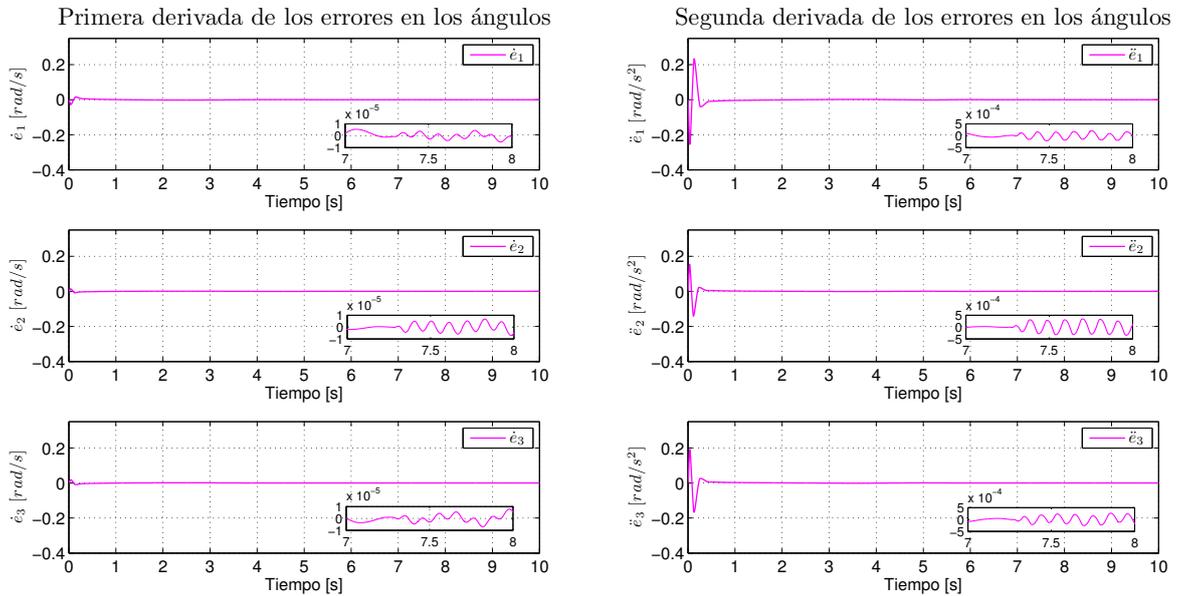


Figura 5.8: Primera y segunda derivada de los errores en los ángulos de los motores, Algoritmo Cuasi-Continuo con diferenciador de Levant de quinto orden

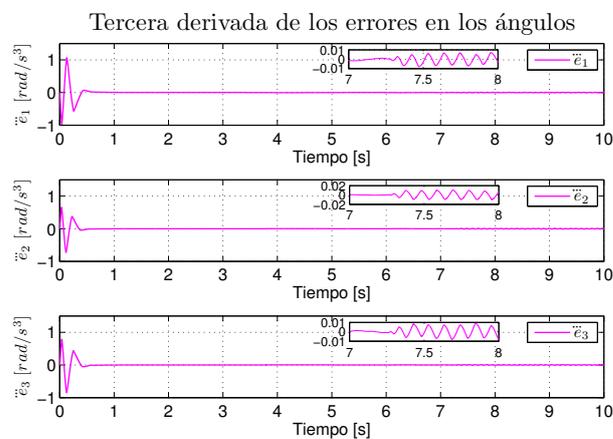


Figura 5.9: Tercera derivada de los errores en los ángulos de los motores, Algoritmo Cuasi-Continuo con diferenciador de Levant de quinto orden

crecientes, estas no pueden rebasar los valores de los parámetros asignados al controlador; en este caso se utilizó un parámetro $\alpha = 0.01$, ver la definición del controlador en el Capítulo 4. Asegurando así una saturación en la señal de control que evita el daño del equipo.

Las Figuras 5.11 y 5.12 muestran la convergencia de las derivadas de los errores para cada motor a cero en tiempo finito, utilizando diferenciadores exactos y robustos de quinto orden, y se mantienen ahí para todo el tiempo siguiente.

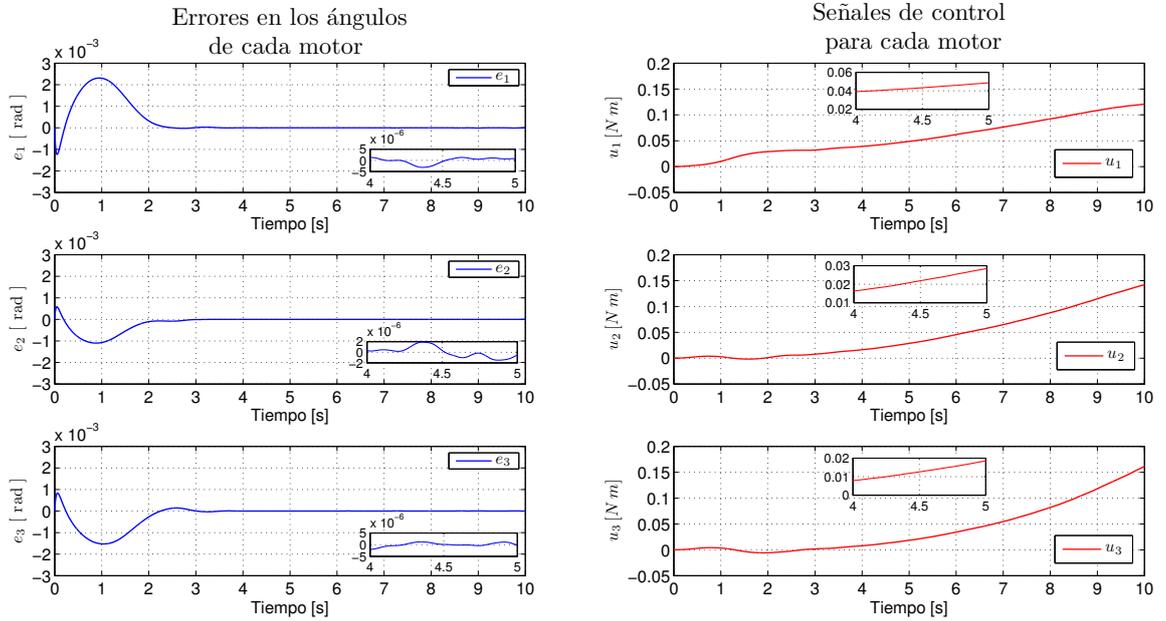


Figura 5.10: Errores en los ángulos de los motores y señales de control para cada motor, Algoritmo Anidado con diferenciador de Levant de quinto orden

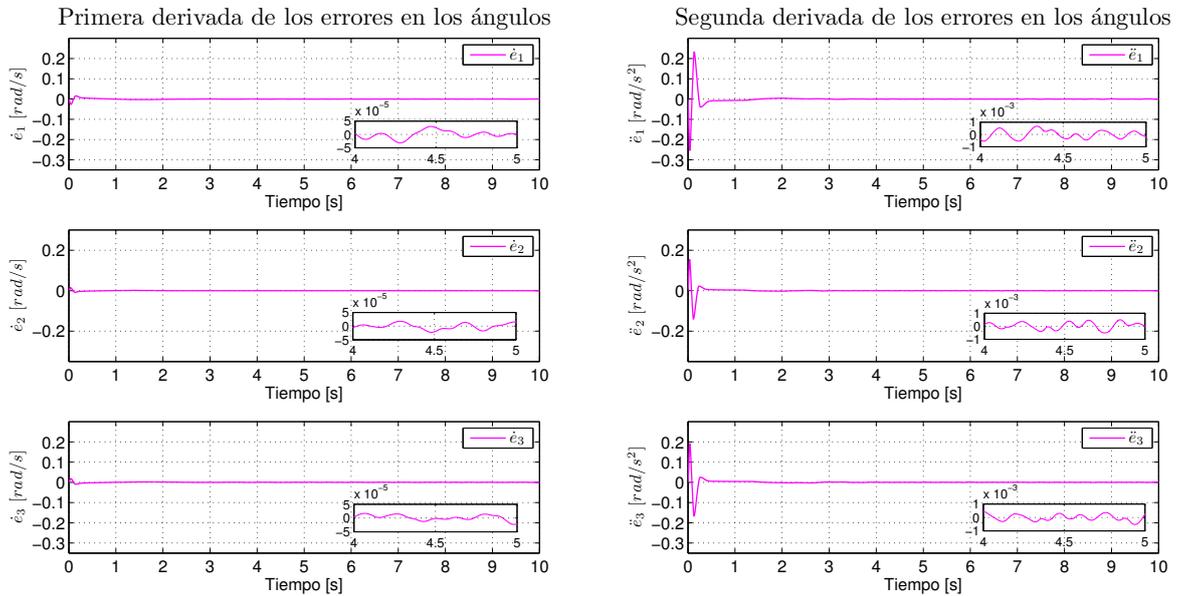


Figura 5.11: Primera y segunda derivada de los errores en los ángulos de los motores, Algoritmo Anidado con diferenciador de Levant quinto orden

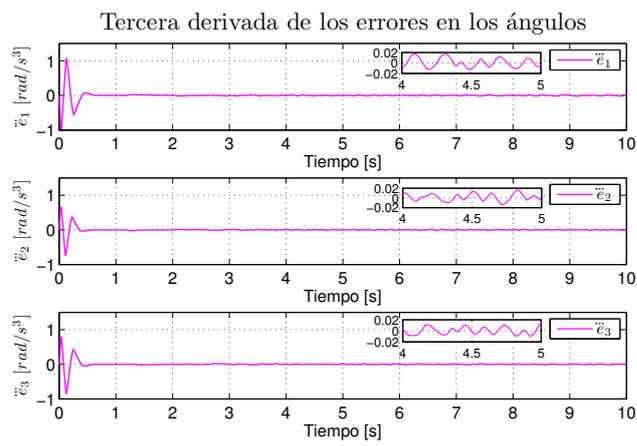


Figura 5.12: Tercera derivada de los errores en los ángulos de los motores, Algoritmo Anidado con diferenciador de Levant quinto orden

Experimentos en el robot

Este capítulo presenta la aplicación de los modos deslizantes de alto orden (HOSM) con base en la metodología de control explicada en capítulos anteriores. El esquema de control propuesto es verificado experimentalmente en un robot manipulador GPM2002 descrito en el Capítulo 3.

6.1. Programación de controladores

El manipulador paralelo está equipado con tres servomotores síncronos. Los ángulos en las articulaciones activas son medidas con codificadores ópticos absolutos, y las articulaciones son controladas por la tarjeta de control de movimiento GT-400-PCI-SV de Googol Tech Ltd [18]. Se implementaron los controladores HOSM y el controlador Super-Twisting con Visual C++. En el experimento, los algoritmos de control se ejecutaron sobre un CPU Pentium(R) Dual-Core a 2.50 GHZ, y 3GB de RAM, con un periodo de muestreo de 3 (ms). Las trayectorias del efector final son una recta y una circunferencia, cuyas características se describen al inicio de cada sección correspondiente. El perfil de la velocidad deseada es una curva trapezoidal [19].

En la Figura 6.1 se muestra el principio de funcionamiento descrito anteriormente. Primero la computadora envía comandos de movimiento a través de un bus PCI, después obtiene el estatus de la tarjeta de movimiento y los parámetros a través de los codificadores ópticos. La tarjeta de movimiento dibuja la trayectoria y realiza operaciones de entrada y salida continuamente, aplicando la cinemática directa e inversa previamente programada. Esto es lo que permite controlar el robot desde la computadora.

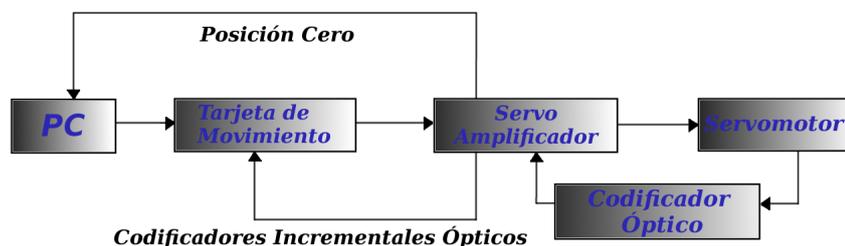


Figura 6.1: Principio de operación del equipo experimental

Los codificadores ópticos absolutos son usados para detectar la posición en línea y posición en cero del servo-amplificador, el cual es transmitido a la PC por comunicación RS-232. Se cuenta con valores de salida máximos del servo digital de $\pm 2^{15}$, con su salida correspondiente analógica de $\pm 10V$. Los datos son obtenidos y guardados en archivos de texto para después interpretarlos en *Matlab* y graficarlos.

6.2. Resultados experimentales con trayectoria recta

A continuación se presentan las gráficas de los experimentos de una trayectoria recta para el error de posición descrito en la ecuación (3.23). El punto de inicio, en coordenadas cartesianas, situado en el espacio de trabajo del robot es (0.2165,0.25), y el punto final es (0.316,0.15), por lo tanto, la distancia de movimiento es 0.1414 m. El objetivo de los controladores, al igual que en las simulaciones, es llevar el error de los ángulos en los motores a cero. Cabe aclarar que los experimentos se realizaron para un tiempo de 11.4 segundos, aunque en los primeros segundos (aproximadamente 1.2) no existe movimiento. Esto debido a que ese es el tiempo que se tarda el robot en calcular la trayectoria a realizar, una vez que inicia el experimento.

6.2.1. Controlador Super-Twisting

Los resultados obtenidos con el algoritmo Super-Twisting propuesto, aplicado al robot paralelo, se muestran en la Figura 6.2. Se puede observar que el desempeño de seguimiento de trayectorias es bueno ya que los errores convergen a cero asintóticamente aproximadamente en el segundo 5, sin embargo la señal de control no es muy

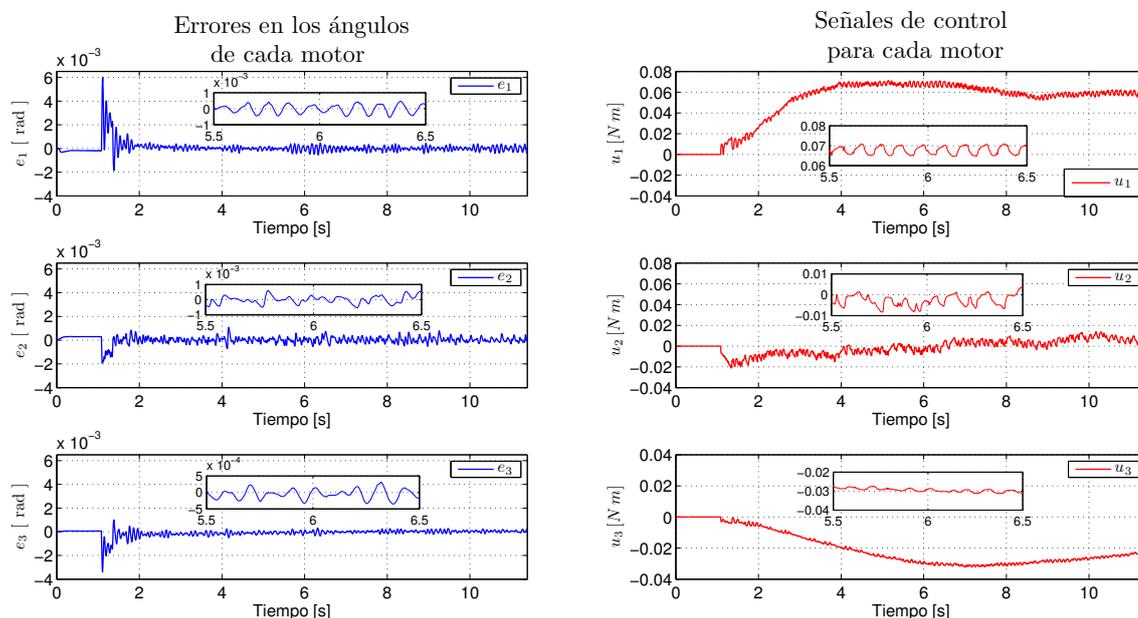


Figura 6.2: Errores en los ángulos de los motores y señales de control Super-Twisting para cada motor de una trayectoria recta

suave y presenta algunas oscilaciones minúsculas. De esta forma los errores tienen una oscilación de magnitud de 10^{-3} radianes.

En relación a la superficie de deslizamiento σ , esta converge en tiempo finito a cero, como se observa en la Figura 6.3. Lo que garantiza el modo deslizante en $\sigma = \dot{\sigma} = 0$.

6.2.2. Controlador por Modos Deslizantes de tercer orden

A continuación se presentan los experimentos para las leyes de control basadas en algoritmos de alto orden. Estos controladores fueron sintonizados tomando de referencia los valores de los parámetros en simulaciones. Sin embargo en la práctica variaron un poco.

6.2.2.1. Controlador Cuasi-Continuo

El controlador Cuasi-Continuo de tercer orden muestra unas pequeñas oscilaciones a alta frecuencia, como se observa en la Figura 6.4, aunque son de amplitud pequeña

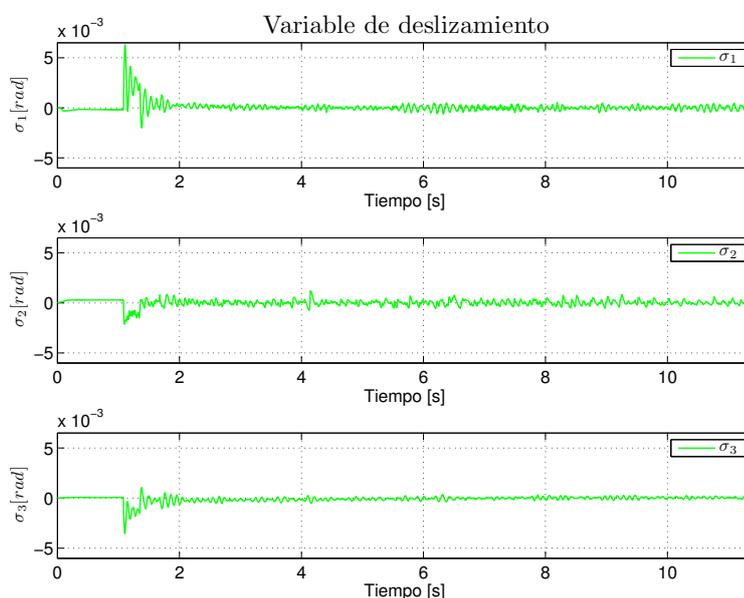


Figura 6.3: Variable de deslizamiento del Super-Twisting para cada motor

están presentes cuando los errores convergen a cero, aproximadamente en el segundo 4.

Los diferenciadores de Levant de cuarto orden programados para este control aseguran la convergencia del valor real de las derivadas de los errores en tiempo finito, i.e. $\dot{e} = \ddot{e} = 0$ como se observa en la Figura 6.5. Sin embargo el valor nunca llega a cero completamente, sino que se mantiene oscilando a lo largo del tiempo. Este problema es debido a la limitación del manejo numérico que tienen las computadoras. Es decir que en el manejo de números reales siempre se va a tener algún tipo de error numérico por truncamiento.

6.2.2.2. Controlador Anidado

El controlador Anidado de tercer orden asegura la convergencia asintótica de los errores a cero, como se observa en la Figura 6.6. Sin embargo, al igual que la simulación, la señal de control muestra ‘picos’, con oscilación a alta frecuencia. Lo que, en cierta medida, puede ser perjudicial para los actuadores controlados. La llegada de los errores a cero se logra aproximadamente en el segundo 4.5 del experimento.

6.2 Resultados experimentales con trayectoria recta

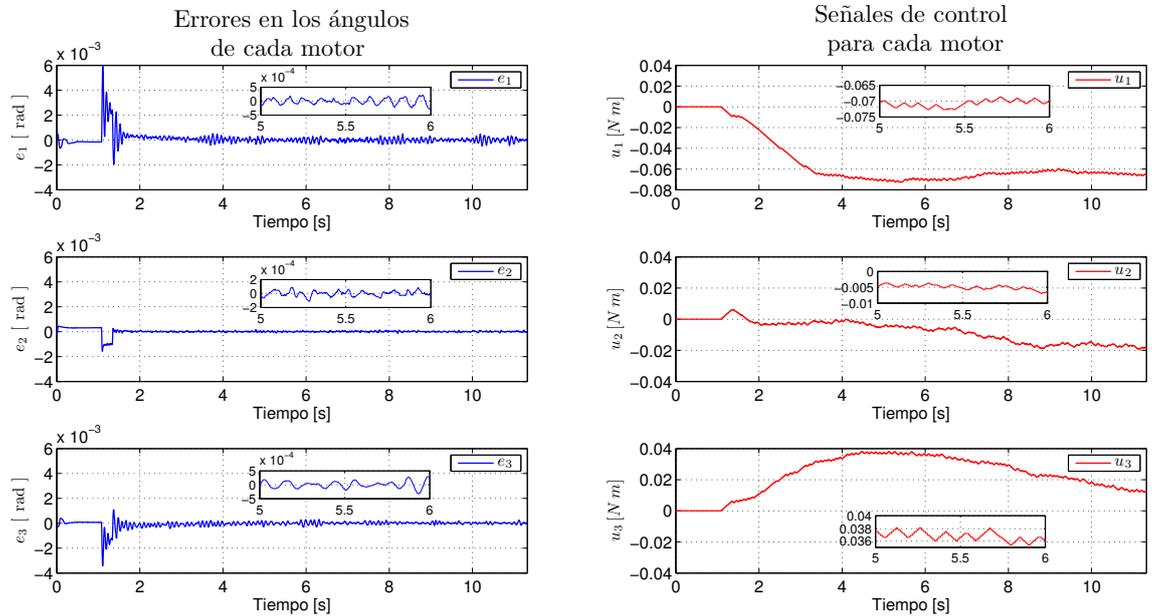


Figura 6.4: Errores en los ángulos de los motores y señales de control para cada motor, algoritmo Cuasi-Continuo con diferenciador de Levant de cuarto orden para una recta

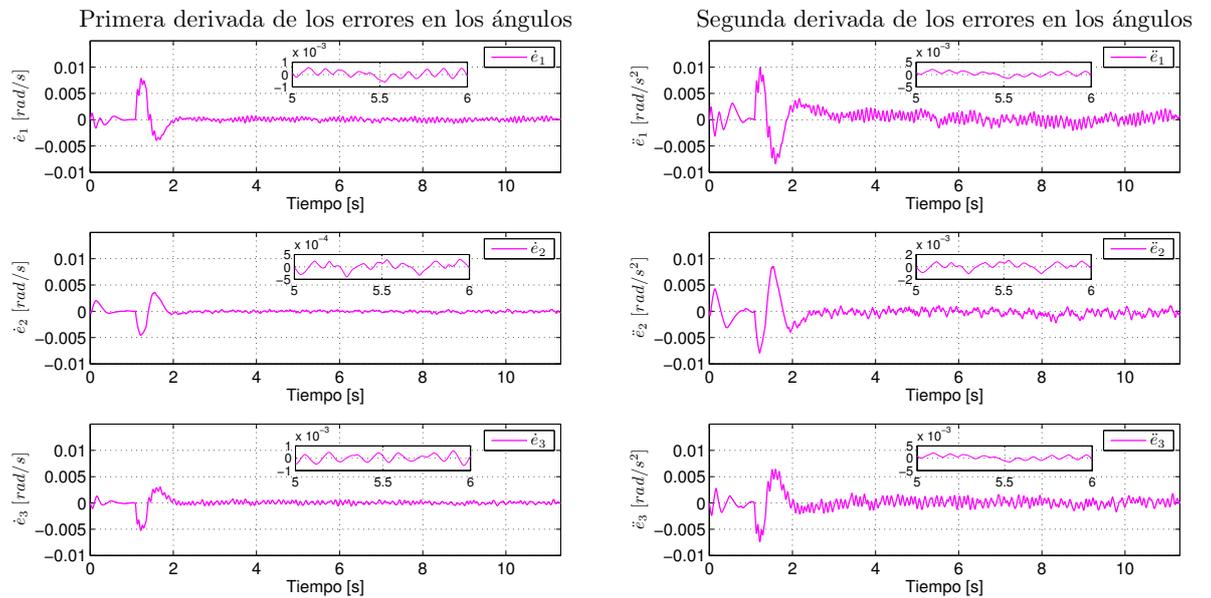


Figura 6.5: Primera y segunda derivada de los errores en los ángulos de los motores, algoritmo Cuasi-Continuo con diferenciador de Levant de cuarto orden para una recta

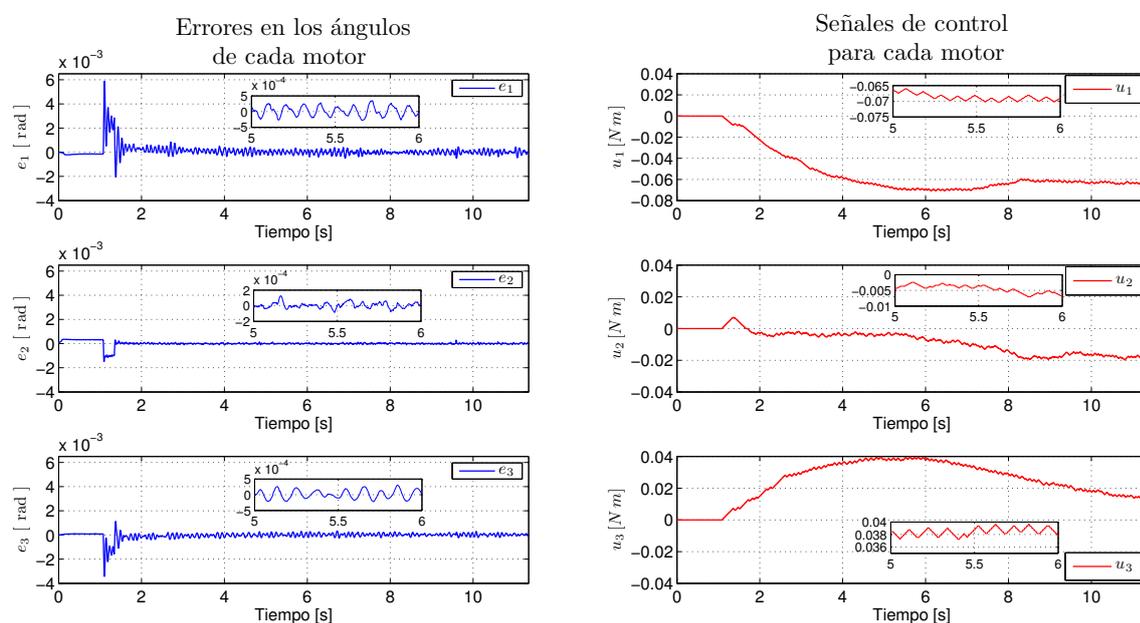


Figura 6.6: Errores en los ángulos de los motores y señales de control para cada motor, algoritmo Anidado con diferenciador de Levant de cuarto orden para una recta

Como se observa en la Figura 6.7, las primeras dos derivadas de los errores \dot{e} y \ddot{e} convergen a cero en tiempo finito, aunque no de manera perfecta, se mantienen oscilando en este valor dadas las limitaciones numéricas computacionales. De todas formas se puede apreciar que los diferenciadores están realizando bien su labor al obtener estos valores numéricos cercanos a su valor verdadero en tiempo real.

6.2.3. Controlador por Modos Deslizantes de cuarto orden

Uno de los principales problemas con los HOSM de cuarto orden es que en su estructura, para un sistema de grado relativo dos, se necesitan las primeras tres derivadas temporales de la restricción con la que se trabaja. Esto puede acarrear errores numéricos que se traducen en imperfecciones en la convergencia de los estados o, en nuestro caso, en los errores de los ángulos del motor. Para compensar este problema se utilizaron diferenciadores robustos y exactos de quinto orden para obtener \dot{e} , \ddot{e} y $\ddot{\ddot{e}}$, como se muestra a continuación.

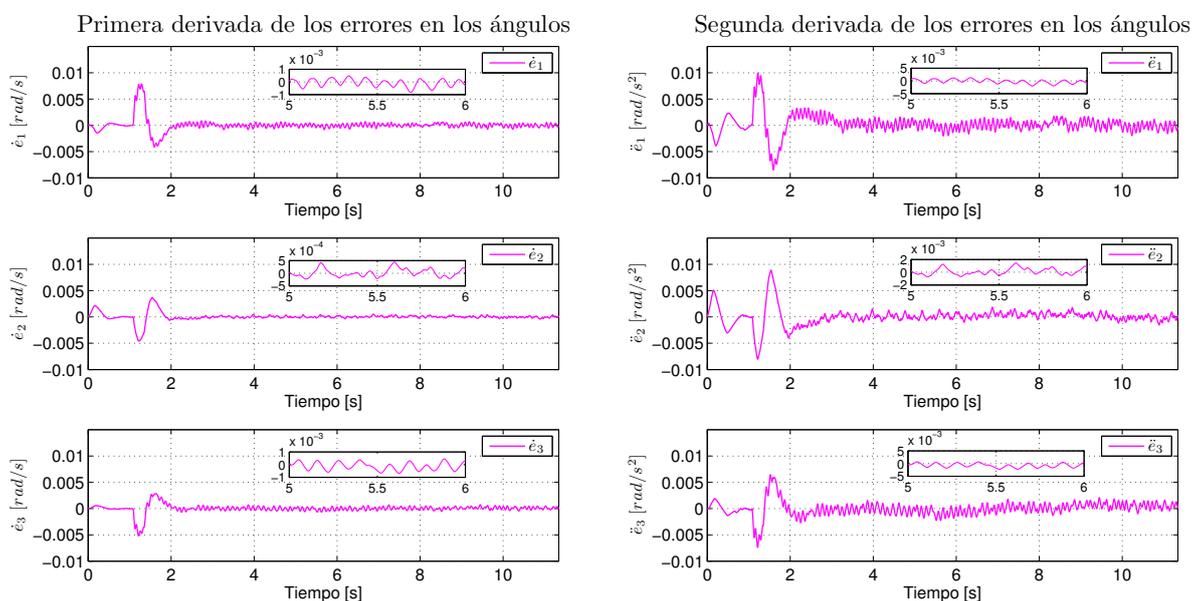


Figura 6.7: Primera y segunda derivada de los errores en los ángulos de los motores, algoritmo Anidado con diferenciador de Levant de cuarto orden para una recta

6.2.3.1. Controlador Cuasi-Continuo

Lo primero a resaltar de los resultados con el controlador Cuasi-Continuo de cuarto orden es que, de acuerdo con las simulaciones, se necesitó de más tiempo para que los errores convergieran a cero, aproximadamente 6 segundos, lo cual es considerablemente malo si es comparado con los de tercer orden. En realidad no se puede asegurar una convergencia exacta a cero ya que los errores oscilan en un valor muy cercano a este. Por otro lado, la ventaja de este esquema es que las señales de control son bastante suaves, como se aprecia en la Figura 6.8.

Para las derivas temporales de los errores, se observa en las figuras 6.9 y 6.10 que convergen a cero en un tiempo muy corto. Aunque para \ddot{e} la señal resulta bastante imprecisa y oscilatoria.

6.2.3.2. Controlador Anidado

El último experimento para la trayectoria recta se realizó con un controlador Anidado de cuarto orden, en donde se observa una señal suave para las señales de control, ver

6. EXPERIMENTOS EN EL ROBOT

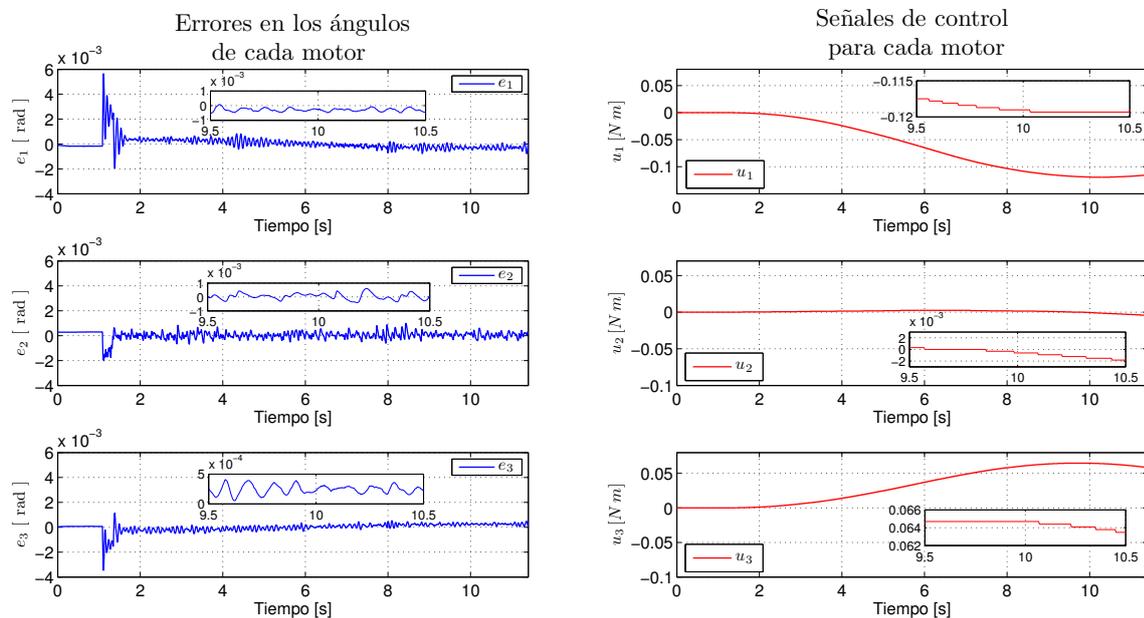


Figura 6.8: Errores en los ángulos de los motores y señales de control para cada motor, algoritmo Cuasi-Continuo con diferenciador de Levant de quinto orden para una recta

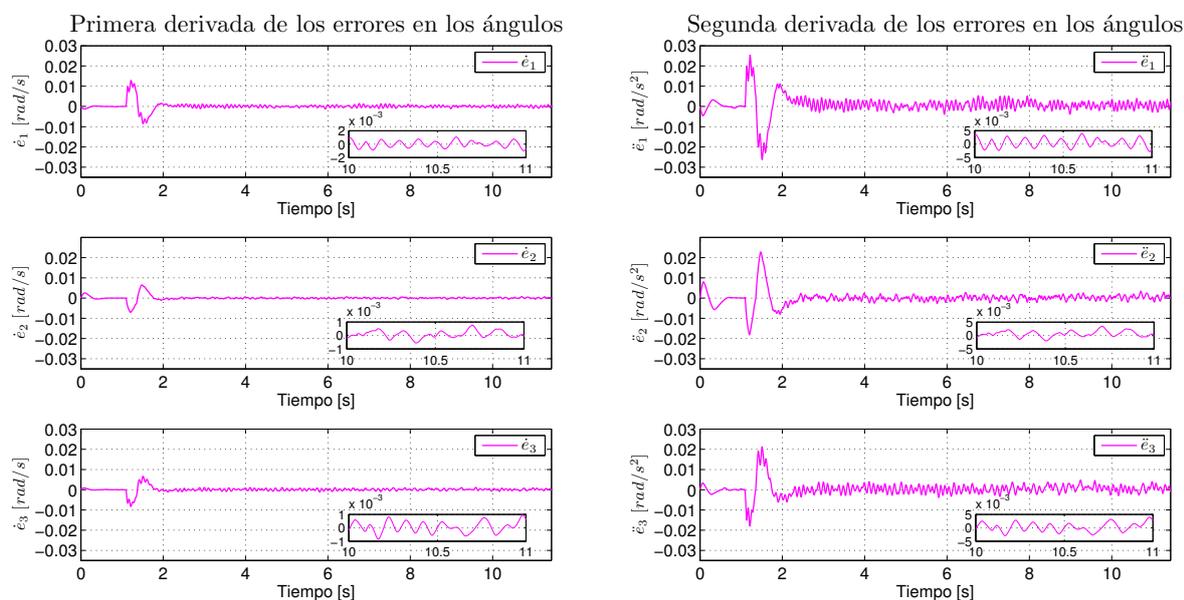


Figura 6.9: Primera y segunda derivada de los errores en los ángulos de los motores, algoritmo Cuasi-Continuo con diferenciador de Levant de quinto orden para una recta

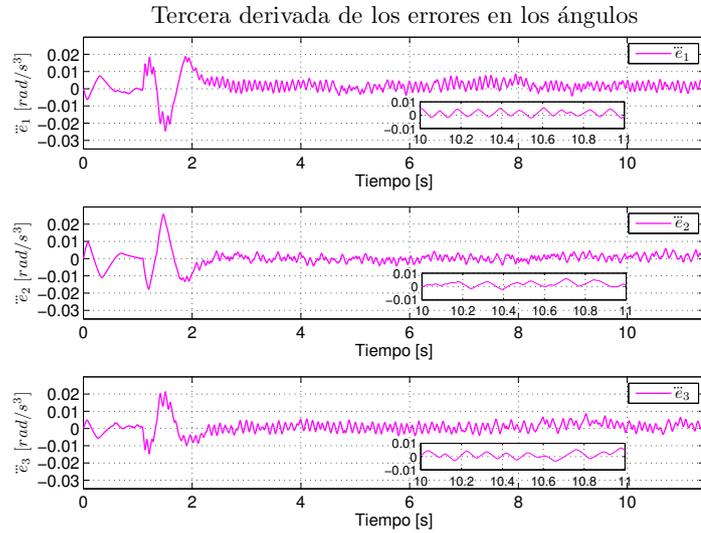


Figura 6.10: Tercera derivada de los errores en los ángulos de los motores, algoritmo Cuasi-Continuo con diferenciador de Levant de quinto orden para una recta

Figura 6.11, sin oscilaciones a alta frecuencia. Los errores presentan una convergencia a cero en aproximadamente 5.5 segundos, mejor que para el Cuasi-Continuo de cuarto orden, pero no tan bueno con respecto a cualquiera de tercer orden.

Las gráficas para las derivadas de los errores presentadas en 6.12 y 6.13, no muestran gran diferencia con respecto al Cuasi-Continuo. Ya que, como se puede observar en los acercamientos respectivos, el valor oscila en cero y nunca se mantiene completamente en este valor a largo del experimento; siendo la tercera derivada la que visualmente oscila más.

6.3. Resultados experimentales con trayectoria circular

El segundo experimento realizado se trató de la combinación de una trayectoria recta y otra circular. Como se muestra en la Figura 6.14. La recta calculada parte del punto central **A** del espacio de trabajo del robot, cuya coordenada en metros es (0.2165 , 0.25), al punto **B**. Una vez que llega a esta segunda posición, el efecto final empieza a seguir una trayectoria circular hasta llegar otra vez al punto **B**, es decir, a la coordenada (0.2165 , 0.2). Formando así una circunferencia con un radio de 50 mm.

6. EXPERIMENTOS EN EL ROBOT

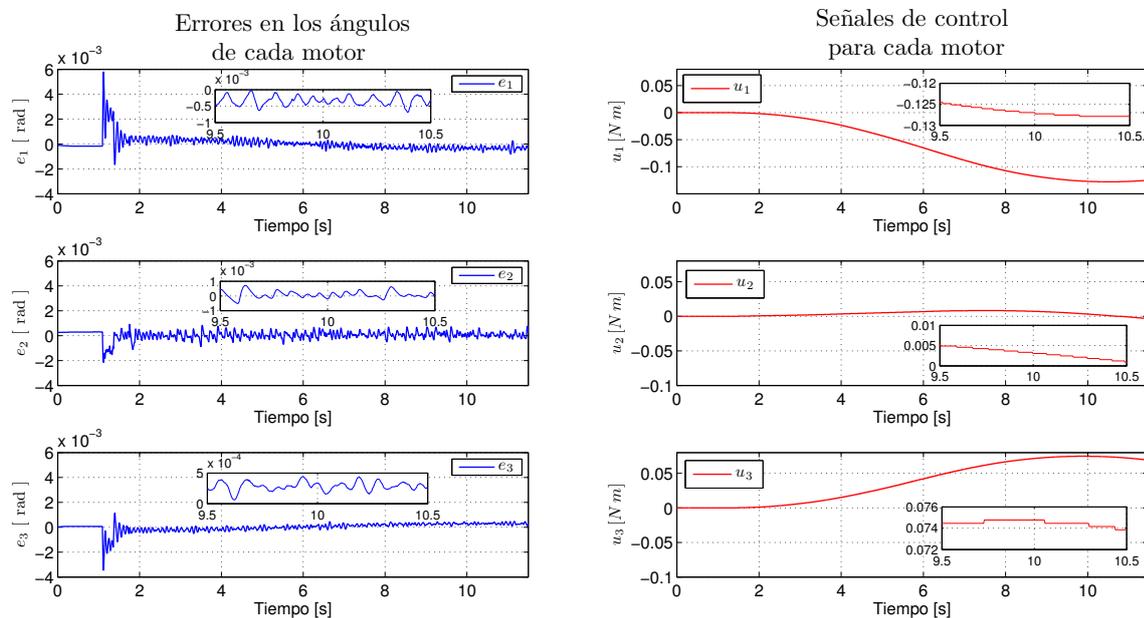


Figura 6.11: Errores en los ángulos de los motores y señales de control para cada motor, algoritmo Anidado con diferenciador de Levant de quinto orden para una recta

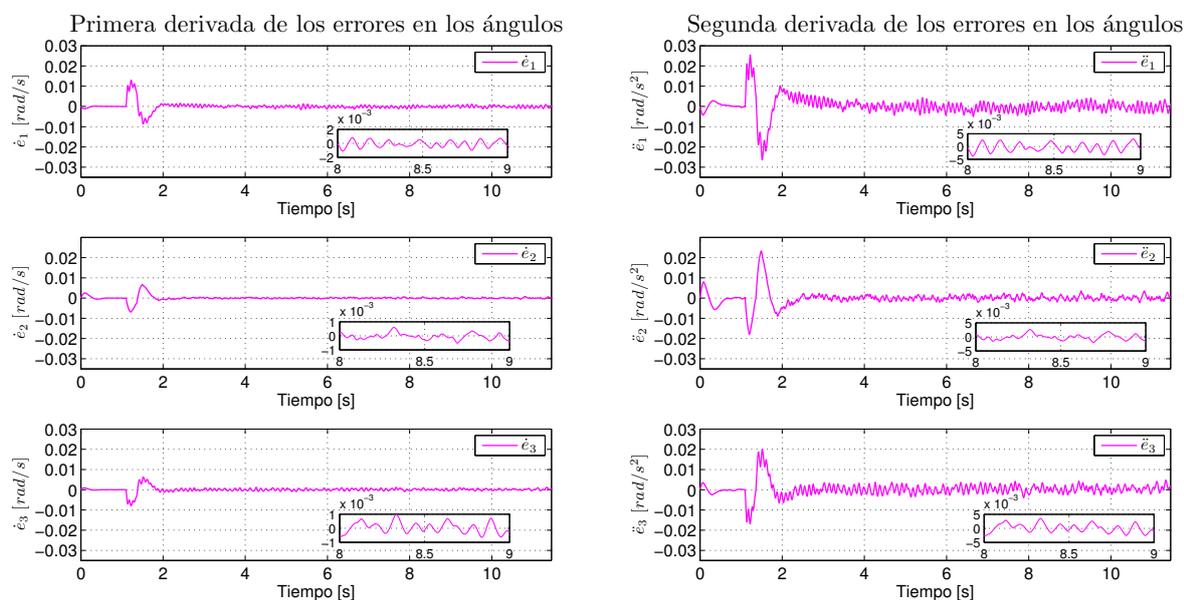


Figura 6.12: Primera y segunda derivada de los errores en los ángulos de los motores, algoritmo Anidado con diferenciador de Levant quinto orden para una recta

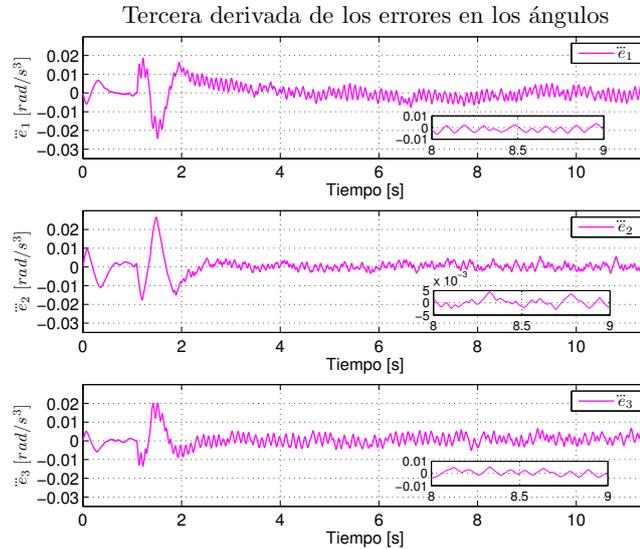


Figura 6.13: Tercera derivada de los errores en los ángulos de los motores, algoritmo Anidado con diferenciador de Levant quinto orden para una recta

Las líneas azules en la imagen señalan la dirección del movimiento del efector.

En las gráficas de las señales de error y señales de control que se muestran a continuación existen dos movimientos visibles. El primero, el de más corta duración, corresponde a la trayectoria recta que hace el manipulador; el segundo corresponde a la circunferencia trazada por el robot. Antes de cada uno de estos dos movimientos hay un lapso de tiempo en donde las señales en las gráficas parecen constantes. Esto debido a que el equipo necesita ese tiempo para realizar los cálculos de los puntos que debe seguir el robot, antes de empezar el movimiento del mismo. Estos dos lapsos no son considerados cuando se hace el análisis de precisión en el siguiente capítulo. También es necesario aclarar que para este experimento en particular únicamente se tomarán en cuenta los valores del error de seguimiento de la circunferencia, a partir del segundo 7 aproximadamente, ya que la trayectoria recta sólo se usa para poder situar el efector final en el punto **B** y así poder iniciar el seguimiento de la trayectoria circular.

6.3.1. Controlador Super-Twisting

Se observa en la Figura 6.15 que a partir del segundo 10 los controladores tratan de corregir las señales de error llevándolas a cero, cuando el efector ya está dibujando

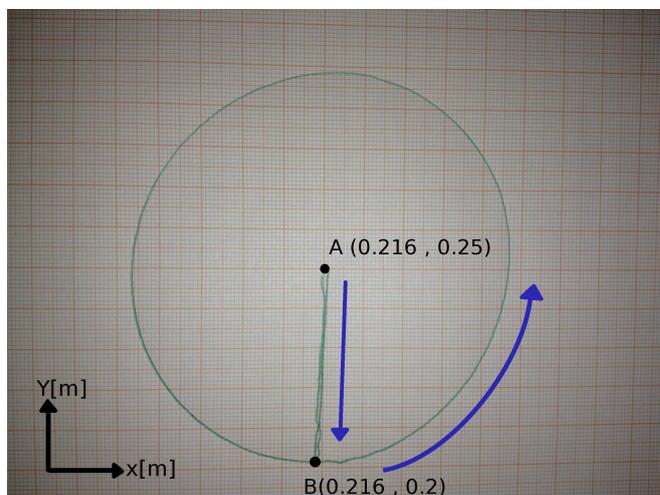


Figura 6.14: Trayectoria conformada por una recta y una circunferencia dibujada por el manipulador

do la circunferencia. Los errores convergen a cero de manera asintótica, y las señales de control aún presentan oscilaciones muy pequeñas como se ve en los acercamientos respectivos.

6.3.2. Controlador por Modos Deslizantes de tercer orden

Los algoritmos de tercer orden que se usaron para la trayectoria circular fueron los mismos que se programaron para la trayectoria recta. Esto quiere decir que se implementaron los diferenciadores de Levant de cuarto orden para obtener las dos primeras derivadas de los errores \dot{e} y \ddot{e} . Estas señales de las derivadas tienen el mismo comportamiento que con la línea recta, convergen a cero asintóticamente y oscilan en este valor a lo largo del seguimiento de la trayectoria. A continuación únicamente se presentan las gráficas de los errores y las leyes de control, las cuales servirán para realizar un análisis posterior.

6.3.2.1. Controlador Cuasi-Continuo

El controlador Cuasi-Continuo de tercer orden muestra unas pequeñas oscilaciones a alta frecuencia al igual que con la recta, como se observa en la Figura 6.16, aunque son de amplitud pequeña están presentes cuando los errores convergen a cero, aproximadamente en el segundo 11.5.

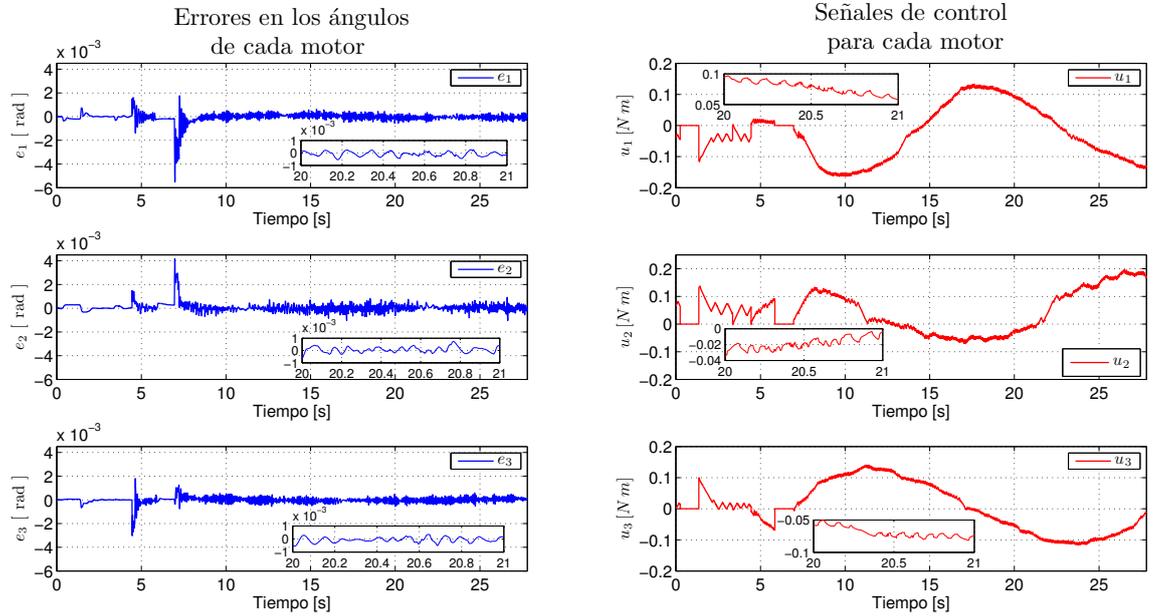


Figura 6.15: Errores en los ángulos de los motores y señales de control Super-Twisting para cada motor de una trayectoria circular

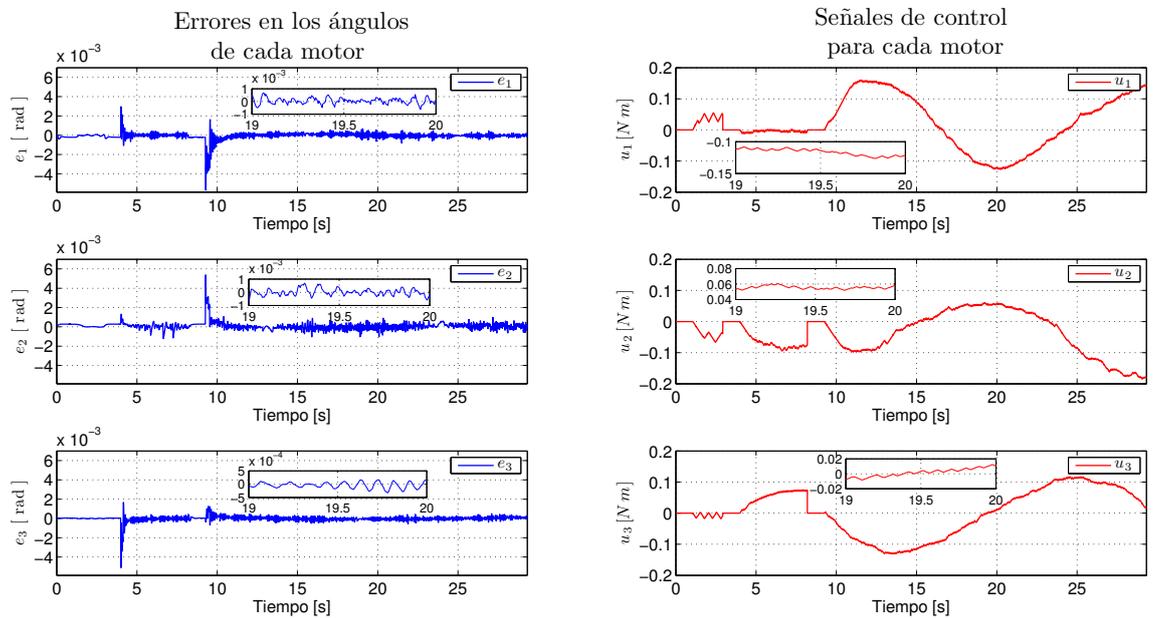


Figura 6.16: Errores en los ángulos de los motores y señales de control para cada motor, algoritmo Cuasi-Continuo con diferenciador de Levant de cuarto orden para una circunferencia

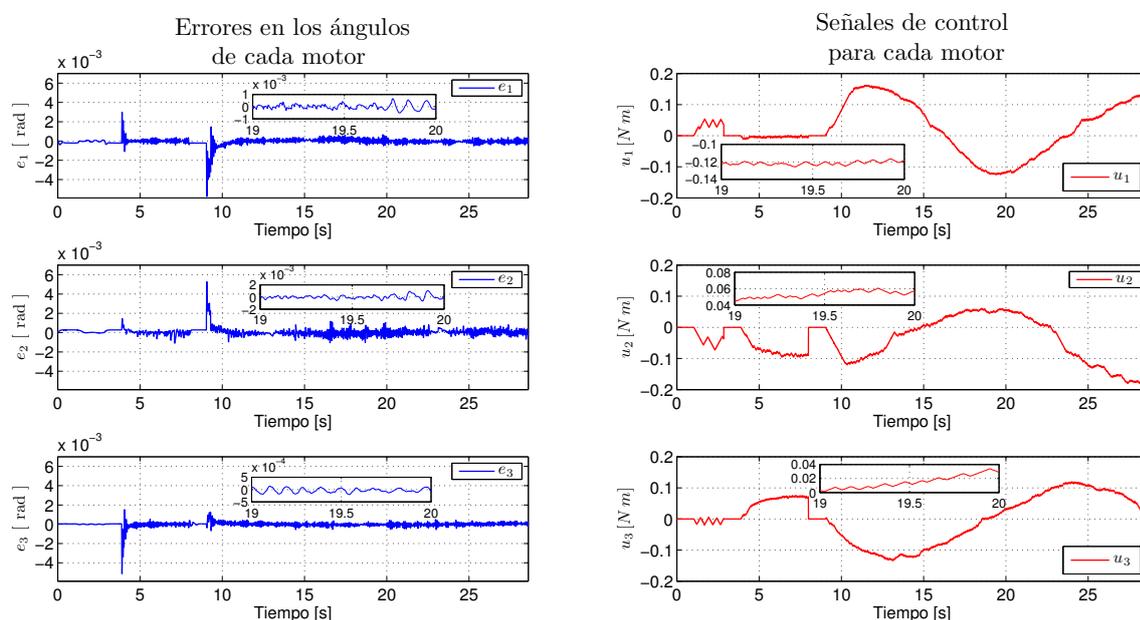


Figura 6.17: Errores en los ángulos de los motores y señales de control para cada motor, algoritmo Anidado con diferenciador de Levant de cuarto orden para una circunferencia

6.3.2.2. Controlador Anidado

En la Figura 6.17 se observa que el controlador Anidado de tercer orden, al igual que el algoritmo Cuasi Continuo, asegura la convergencia asintótica de los errores a cero. Las señales de control presentan las mismas características de las oscilaciones a alta frecuencia con pequeña amplitud que en el experimento de la recta, aunque la forma de la señal cambia más, ya que tiene que compensar un error más grande de movimiento en el efector.

6.3.3. Controlador por Modos Deslizantes de cuarto orden

Para la realización del experimento con controladores de cuarto orden se utilizó la misma estructura que en el de la trayectoria recta. Es decir que se programaron los controladores con diferenciadores robustos y exactos de quinto orden para obtener las derivadas temporales del error \dot{e} , \ddot{e} y $\ddot{\ddot{e}}$. A continuación se muestran únicamente las señales de error y señales de control de cada uno de los dos algoritmos HOSM, esto para poder realizar un análisis de precisión para cada una de las familias de controladores

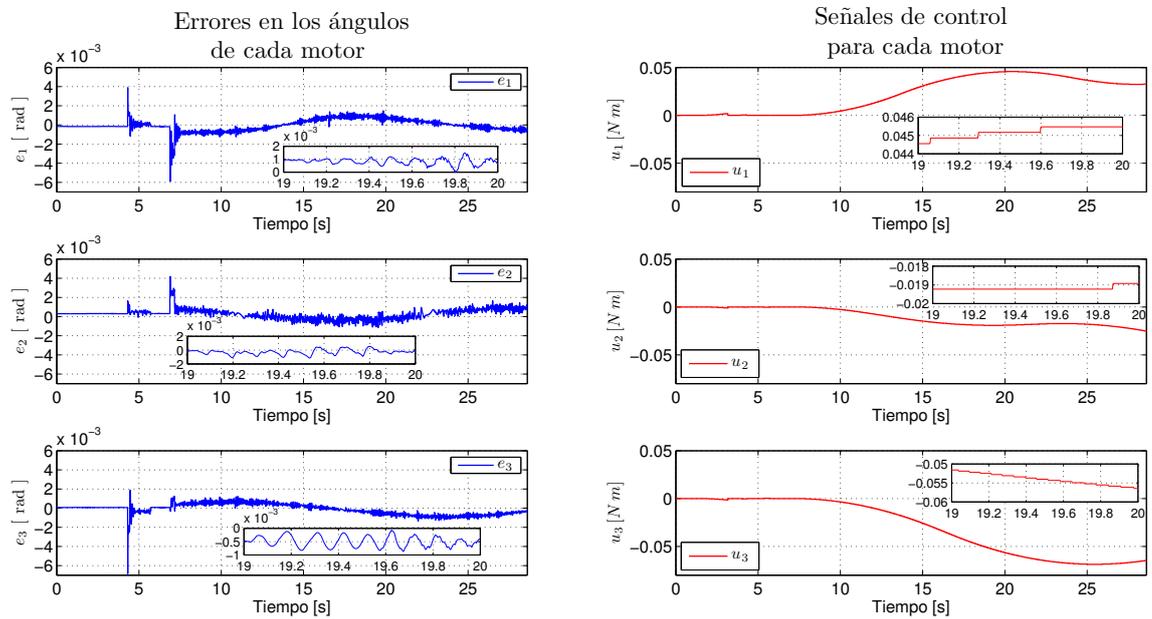


Figura 6.18: Errores en los ángulos de los motores y señales de control para cada motor, algoritmo Cuasi-Continuo con diferenciador de Levant de quinto orden para una circunferencia

de alto orden en la siguiente sección.

6.3.3.1. Controlador Cuasi-Continuo

De la Figura 6.18 se observa que las señales de control para cada motor son muy suaves ya que no presentan las pequeñas oscilaciones como en el caso del Super-Twisting o de los algoritmos de tercer orden, una vez que inicia la trayectoria circular a partir del segundo 7. Sin embargo en los acercamientos se aprecia que la resolución de las señales es muy pobre, ya que pueden verse pequeños saltos en el valor de la señal de color rojo. La convergencia de los errores a cero no se garantiza del todo ya que, como se aprecia en las gráficas, los valores tienen un comportamiento oscilatorio y nunca llegan al valor deseado completamente.

6.3.3.2. Controlador Anidado

La principal ventaja de los algoritmos de cuarto orden Anidados es que la señal de control es muy suave, como se observa en la Figura 6.19, aunque la resolución de

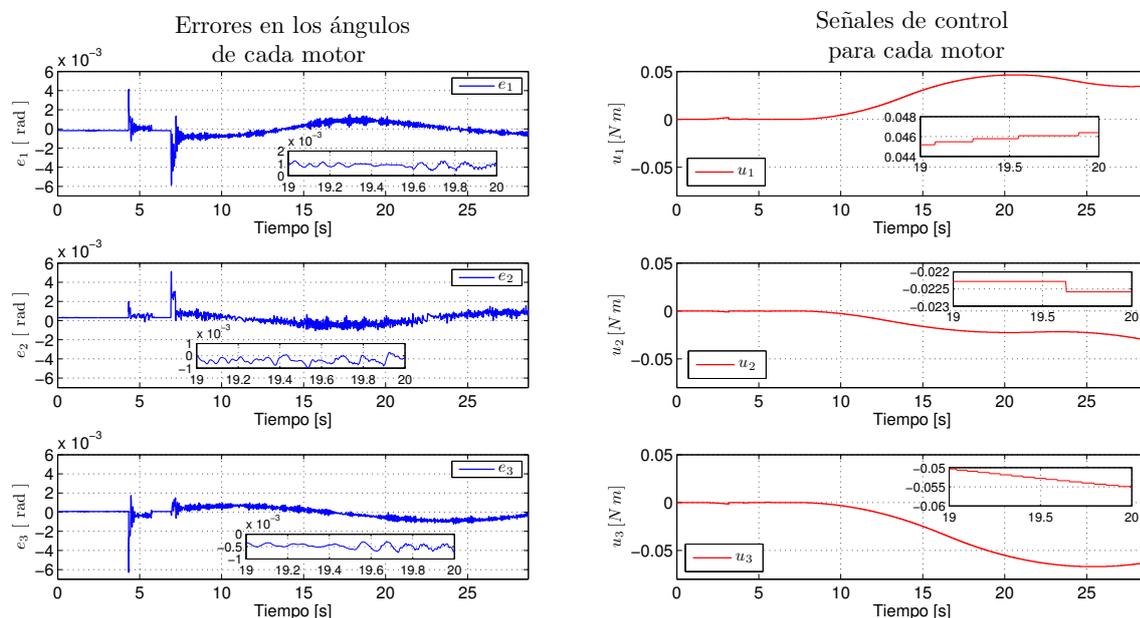


Figura 6.19: Errores en los ángulos de los motores y señales de control para cada motor, algoritmo Anidado con diferenciador de Levant de quinto orden para una circunferencia

la señal no llega a ser muy buena cuando se presta atención a los acercamientos. La desventaja de estas leyes de control se aprecia en las gráficas mostradas, en donde los errores no logran llegar completamente a un valor igual a cero mientras la trayectoria de la circunferencia está trazándose, únicamente oscilan en un valor cercano a este.

6.4. Cálculo de la precisión

La evaluación de las propiedades de cada una de las estrategias de control implementadas anteriormente de manera experimental puede ser hecha apoyándose en los conceptos de error máximo (e_{max}) y error en estado estable (e_{ss}). En la Figura 6.20 se muestra la señal de error del motor 1 para el algoritmo Super-Twisting, en esta se puede apreciar la magnitud considerada de los errores que se tomarán en cuenta para concluir acerca de la precisión de los controladores. Esta consideración será la misma para cada una de las señales de error de los algoritmos de modos deslizantes mostrados en las secciones anteriores de este capítulo. Como ya se mencionó, en todos los experimentos existe un tiempo de cálculo de trayectorias necesario, mostrado en la imagen con una flecha verde, en donde no existe movimiento por parte del robot. Los errores iniciales

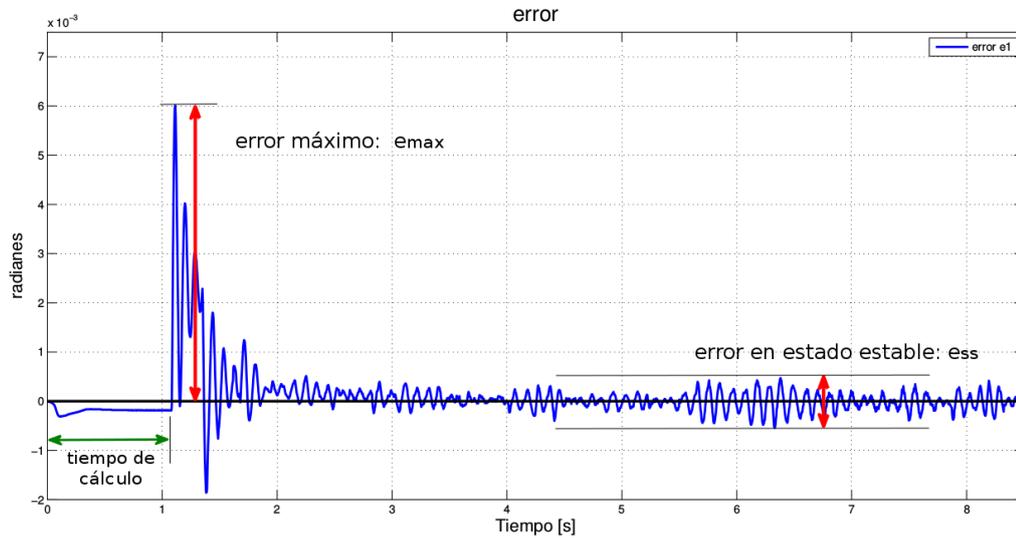


Figura 6.20: Error máximo y error en estado estable de la posición del motor 1 con controlador Super Twisting

presentes en este lapso de tiempo no se tomarán en cuenta para las conclusiones posteriores, ya que su variación en magnitud no depende del controlador programado.

6.4.1. Error máximo

El error máximo se puede definir como el mayor valor absoluto al que puede llegar la señal de error en el experimento. En la Tabla 6.1 se muestran los errores máximos de cada uno de los motores para todos los algoritmos de control implementados en el experimento de la trayectoria recta. De igual forma en la Tabla 6.2 se muestran los valores máximos de error, esta vez para la trayectoria circular.

En general se puede observar en las tablas que el error máximo en el experimento de la línea recta es muy parecido entre cada uno de los controladores, siendo el Super-Twisting el que presenta el mejor desempeño por muy poco. Para la trayectoria circular se tienen variaciones más grandes de estos valores, en especial para el motor 3, siendo de igual forma el Super-Twisting el que tiene el mejor conjunto de valores.

	$e_{max}[rad]$ motor 1	$e_{max}[rad]$ motor 2	$e_{max}[rad]$ motor 3
Super-Twisting	0.0060	0.0019	0.0034
Cuasi-Continuo tercer orden	0.0060	0.0021	0.0035
Anidado tercer orden	0.0061	0.0018	0.0035
Cuasi-Continuo cuarto orden	0.0057	0.0020	0.0035
Anidado cuarto orden	0.0058	0.0022	0.0035

Tabla 6.1: Tabla comparativa de experimentos de una trayectoria recta - máximo de error total para cada uno de los motores del robot, experimento con trayectoria recta

	$e_{max}[rad]$ motor 1	$e_{max}[rad]$ motor 2	$e_{max}[rad]$ motor 3
Super-Twisting	0.0055	0.0042	0.0030
Cuasi-Continuo tercer orden	0.0057	0.0054	0.0051
Anidado tercer orden	0.0057	0.0053	0.0051
Cuasi-Continuo cuarto orden	0.0059	0.0042	0.0068
Anidado cuarto orden	0.0058	0.0051	0.0063

Tabla 6.2: Tabla comparativa de experimentos de una trayectoria circular - máximo de error total para cada uno de los motores del robot, experimento con trayectoria circular

6.4.2. Error de estado estable

El error de estado estable se define para un conjunto de valores i como $e_{ss} = e_{imax} - e_{imin}$. Es decir que para nuestras señales el error de estado estable es igual al valor máximo de error menos el valor mínimo, una vez que se entró al modo deslizante en los últimos segundos del experimento. Debido a esto se obtiene un banda de error para cada uno de los algoritmo en donde para un valor más pequeño de e_{ss} se tiene una mejor precisión. En la Tabla 6.3 se muestran estos errores para cada motor en el experimento de la recta, y en la Tabla 6.4 se hace lo mismo para el experimento de la circunferencia.

	$e_{ss}[rad]$ motor 1	$e_{ss}[rad]$ motor 2	$e_{ss}[rad]$ motor 3
Super-Twisting	0.0010	0.0013	0.6676×10^{-3}
Cuasi-Continuo tercer orden	0.9896×10^{-3}	0.0017	0.8325×10^{-3}
Anidado tercer orden	0.0014	0.0014	0.63617×10^{-3}
Cuasi-Continuo cuarto orden	0.0011	0.0016	0.8482×10^{-3}
Anidado cuarto orden	0.0011	0.0014	0.7147×10^{-3}

Tabla 6.3: Tabla comparativa de experimentos de una trayectoria recta - máximo del error de estado estable para cada uno de los motores del robot, experimento con trayectoria recta

De manera general se observa que para el experimento de la trayectoria recta se tiene una mejor precisión en los algoritmos de tercer orden y Super-Twisting. En el caso de los HOSM de cuarto orden, tanto el Cuasi-Continuo como el Anidado, se tiene el peor desempeño en algunos de los motores. Esta característica se acentúa más observando el e_{ss} del experimento de la trayectoria circular, en donde es evidente que los controladores de cuarto orden tienen un valor de error mucho más grande que en el Super-Twisting y, por lo tanto, presentan la peor precisión.

	$e_{ss}[\text{rad}]$ motor 1	$e_{ss}[\text{rad}]$ motor 2	$e_{ss}[\text{rad}]$ motor 3
Super-Twisting	0.0012	0.0019	0.0011
Cuasi-Continuo tercer orden	0.0015	0.0020	0.0011
Anidado tercer orden	0.0012	0.0022	0.0010
Cuasi-Continuo cuarto orden	0.0024	0.0028	0.0019
Anidado cuarto orden	0.0023	0.0026	0.0019

Tabla 6.4: Tabla comparativa de experimentos de una trayectoria circular - máximo del error de estado estable para cada uno de los motores del robot, experimento con trayectoria circular

6.5. Tiempos de ejecución

El último aspecto que se tomará en cuenta para poder concluir acerca del desempeño de cada uno de los controladores es el tiempo de ejecución experimental, el cual se define como el tiempo que tarda el manipulador en realizar la trayectoria deseada, esto sin tomar en cuenta el tiempo de cálculo inicial. En las tablas 6.5 y 6.6 se presenta este valor para el experimento con la trayectoria recta y con la circunferencia respectivamente. En donde se aprecia que el Super-Twisting tarda un poco menos en ejecutar esta tarea que los algoritmos HOSM.

Algoritmo	Tiempo [s]
Super-Twisting	9.985
Cuasi-Continuo tercer orden	10
Anidado tercer orden	10.015
Cuasi-Continuo cuarto orden	10.016
Anidado cuarto orden	10.047

Tabla 6.5: Tiempos de ejecución. Tiempos que tardó cada experimento en hacer la tarea de control, experimento con línea recta

Algoritmo	Tiempo [s]
Super-Twisting	19.891
Cuasi-Continuo tercer orden	20.125
Anidado tercer orden	20.125
Cuasi-Continuo cuarto orden	20.219
Anidado cuarto orden	20.063

Tabla 6.6: Tiempos de ejecución. Tiempos que tardó cada experimento en hacer la tarea de control, experimento con circunferencia

Conclusiones

7.1. Comparación entre controladores

Debido al acoplamiento cinemático entre las cadenas ramificadas la dinámica y la fricción del robot paralelo son altamente no lineales. Con el fin de obtener un mejor funcionamiento del manipulador, en lo referente a la precisión de seguimiento de trayectorias, se hizo un análisis matemático de la dinámica del mismo para identificar las características de la planta. Una vez realizado esto, las dinámicas no modeladas y la fricción se compensan mediante el uso de controladores continuos por modos deslizantes, es decir, por medio de un control robusto ante perturbaciones acotadas y acopladas.

Los controladores HOSM propuestos se implementaron sobre una plataforma manipuladora paralela real, el robot sobreactuado GPM2002 de la marca *Googol Tech*, y con base en los resultados experimentales obtenidos para el seguimiento de una trayectoria recta y otra circular del efector final, se puede hacer una comparación entre los algoritmos de tercer orden, cuarto orden y Super-Twisting.

De los experimentos con los HOSM de cuarto orden se puede observar que la convergencia de los errores es errática para estos controladores, ya que, en los últimos segundos, el valor oscila cerca de cero conforme transcurre el tiempo experimental, además de que el cálculo de las derivadas sucesivas del error no fue completamente exacta. Esta característica es más evidente en el experimento de la circunferencia utilizando un HOSM Anidado y también con el Cuasi-Continuo.

7. CONCLUSIONES

Esto se debe primordialmente a que la resolución de la señal de control (de $\pm 2^{15}$ valores) no es suficiente para llevar a los errores a converger perfectamente. Dicho fenómeno es evidente al observar que la continuidad de las señales de control se ve menos definida conforme trata de compensar los errores. Esta es una desventaja considerable en estos algoritmos dado que no se puede reducir más los parámetros de convergencia del controlador porque este problema se amplificaría. Dicha peculiaridad no se observó en los otros controladores implementados.

Para los algoritmos de tercer orden y Super-Twisting no hubo mucha diferencia en cuanto a los valores de error máximo y error de estado estable, aunque el Super-Twisting presentaba siempre un valor más pequeño. Aspecto que fue más evidente para las trayectorias circulares. En lo referente a las señales de control, en el algoritmo de tercer orden se apreció una menor oscilación de alta frecuencia que en el de segundo orden. Dado que esta oscilación en los controladores siempre estuvo presente, se puede afirmar que no se eliminó por completo el efecto *chattering*, únicamente se atenuó.

Considerando el tiempo de ejecución de los algoritmos, el Super-Twisting presenta el mejor desempeño, ya que no necesita de muchos cálculos para su implementación. Por otro lado para los controladores basados en algoritmos de alto orden es necesario programar diferenciadores exactos y robustos, los cuales conllevan un cálculo matemático más complejo y por lo tanto necesitan un mayor procesamiento por parte de la computadora.

7.2. Conclusiones Generales

La posibilidad de aplicar la metodología de los modos deslizantes de alto orden en la robótica son tratadas en este trabajo. La idea es usar un esquema de control con base en los HOSM para trabajar con señales de control continuas. Esto con la finalidad de controlar el desempeño del movimiento de un robot teniendo como objetivo llevar los errores de posición de los actuadores a cero.

El esquema de control propuesto provee una robustez contra incertidumbres y perturbaciones acotadas y acopladas a la señal de control, por lo tanto permite compensar

posibles errores de imprecisión en el modelo matemático. Más aún, esto permite mitigar el problema del *chattering* por medio del diseño de una ley de control continua. Esto hace a los controladores por modos deslizantes propuestos realmente aplicables a un robot industrial del mismo tipo que la planta en estudio, dada la práctica abstinencia de las vibraciones inducidas.

Las pruebas experimentales aplicadas al robot GPM2002 de *Googol Technologies* demuestran que el error de seguimiento de la posición es forzado a cero, asegurando un buen desempeño del sistema controlado. Con base en los resultados obtenidos es posible señalar las características más importantes de los algoritmos programados; con lo que se puede tener un criterio más amplio si se desea extender esta metodología a otros manipuladores robóticos, o para una consideración en trabajos futuros.

Los controladores HOSM de tercer orden, tanto el Cuasi-Continuo como el Anidado, demuestran garantizar convergencia asintótica a cero para el error de seguimiento. Sin embargo, al igual que con el Super-Twisting, aún se tienen algunas pequeñas oscilaciones a alta frecuencia en las señales de control, presentando también mejores índices de error máximo y error de estado estable. Por otro lado, los controladores HOSM de cuarto orden tienen una menor precisión, aunque no definitiva, pero cuentan con la gran ventaja de producir señales de control totalmente continuas y muy suaves. De acuerdo al mejor conocimiento del autor, este es la primera vez que un controlador de cuarto orden por modos deslizantes es aplicado a un robot manipulador paralelo sobreactuado. Un análisis empírico implica que se puede mejorar la convergencia de los errores, utilizando un HOSM de alto orden, si también se mejora la resolución que el equipo experimental provee a las señales de los actuadores. Esto es debido a que las señales, en teoría continuas completamente, resultaron menos definidas conforme el experimento se llevaba a cabo. En última instancia las pruebas experimentales con un mejor equipo de adquisición de datos, un mejoramiento en la interfaz de programación (tratar de utilizar un ambiente de programación de más alto nivel), así como la estimación de las perturbaciones que afectan el movimiento del robot, quedarían como trabajo futuro.

A.1. Programas

A.1.1. Diferenciador de Levant

Las operaciones definidas para implementar los controladores HOSM, se pudieron realizar gracias al uso de los diferenciadores exactos y robustos. Para el caso del equipo experimental se tuvo que usar el lenguaje de programación C++ con el ambiente de Visual C++ para programar dichos diferenciadores.

Se utilizó el paradigma de programación orientado a objetos para tener cualquier algoritmo de orden n a disposición creando instancias de clases definidas, programando el método recursivo propuesto por Levant. A continuación se presenta la definición de las clases con sus métodos respectivos. Cabe mencionar que para la integración se utilizó el método numérico de la Integral Trapezoidal.

Definición de la operación signo: *signo.h*

```
1
2 #ifndef SIGNO_H_INCLUDED
3 #define SIGNO_H_INCLUDED
4
5 template <typename T> int sgn(T val) {
6     return (T(0) < val) - (val < T(0));
7 }
8
9 #endif // SIGNO_H_INCLUDED
```

Clase *Levant.h*

```
1 #ifndef LEVANT_H
2 #define LEVANT_H
3
4
5 class Levant
6 {
7     public:
8         Levant(int k , int _numOut);
9         virtual ~Levant();
10        double* diffL(double ft);
11        void setL(double _L);
12        void setH(double _h);
13        int const getNumOut() const { return this->numOut; }
14        double const getL() const { return this->L; }
15        double const getH() const { return this->h; }
16        int const getK() const {return this->k; }
17    private:
18        double lambda[7];
19        double ft;
20        int k;
21        int numOut;
22        double L;
23        double h; //Para la integral trapezoidal
24        double* z;
25        double* dz;
26        double* dz_ant;
27        double* v;
28        double* salidas;
29
30 };
31
32 #endif // LEVANT_H
```

Definición de métodos *Levant.cpp*

```
1 #include <iostream>
2 #include <cmath>
3 #include "Levant.h"
4 #include "signo.h"
5
6 using namespace std;
```

```
7
8
9
10 Levant::Levant(int _k , int _numOut):k(_k) , numOut(_numOut), L(1) , h
    (1)
11 {
12
13     if(_k < 2 )
14         this->k = 2;
15
16     if(_numOut < 1)
17         this->numOut = 1;
18
19     if(numOut > k)
20         numOut = k;
21
22     int i;
23     z = new double[ k +1 ];
24     dz = new double[ k + 1 ];
25     dz_ant = new double[ k +1 ];
26     salidas = new double[numOut];
27
28     double valLambda[7] = {1.1 , 1.5 , 3 , 5 , 8 , 12 , 15};
29     for(i = 0 ; i<7 ; i++)
30         lambda[i] = valLambda[i];
31
32     for (i = 0 ; i <=k ; i++){
33         z[i] = 0;
34         dz[i] = 0;
35         dz_ant[i] = 0;
36     }
37
38     for(i = 0 ; i < numOut ; i++)
39         salidas[i] = 0;
40
41 }
42
43 Levant::~~Levant()
44 {
45     delete [] z;
46     delete [] dz;
47     delete [] dz_ant;
48     delete [] salidas;
49 }
50
```

```

51 void Levant::setL(double _L){
52     if(_L < 0)
53         this->L = 1;
54     else
55         this->L = _L;
56 }
57
58 void Levant::setH(double _h){
59     if(_h < 0)
60         this->h = 1;
61     else
62         this->h = _h;
63 }
64
65 double* Levant::diffL(double ft){
66
67     int i;
68     double v, evalua, integ;
69
70     for(i = 0 ; i < k ; i++){
71         if(i == 0){
72             evalua = z[0] - ft;
73         }else{
74             evalua = z[i] - v;
75         }
76
77         v = -lambda[k - i] * pow( L, (double)1/(k+1-i) ) * pow(abs(evalua
78             ), (double)(k-i)/(k+1-i) ) * sgn(evalua) + z[i+1];
79         dz[i] = v;
80     }
81     evalua = z[k] - v;
82     v = -lambda[0] * L * sgn(evalua);
83     dz[k] = v;
84
85     for(i = 0 ; i < numOut ; i++)
86         salidas[i] = z[i+1];
87
88
89     //Integral Trapezoidal
90     for(i = 0 ; i <= k ; i++){
91         integ = (z[i] + h*(dz_ant[i] + dz[i])/2 );
92         z[i] = integ;
93     }
94

```

```

95     for(i = 0 ; i<=k ; i++)
96         dz_ant[i] = dz[i];
97
98     return salidas;
99 }

```

A.1.2. Controladores HOSM

Los siguientes son los controladores por modos deslizantes de alto orden programados únicamente para un sólo servo-motor del robot. Hay que resaltar que sólo se muestran algunos segmentos de código de todo el programa que hace funcionar al manipulador, el cual por su extensión no es posible documentarlo este trabajo.

Cuasi-Continuo de Tercer Orden

```

1 //DIFERENCIANDO CON LEVANT
2 derivs = dL_m1.diffL((double)actl_err1); //derivs es un arreglo con los
   datos que resultan de diferenciar
3
4 dot_err1 = derivs[0]; //Primera derivada
5 ddot_err1 = derivs[1]; //Segunda derivada
6
7 num = ddot_err1 + beta2m1*pow(abs(dot_err1)+ beta1m1*pow(abs(actl_err1),
   (double)2/3),-0.5) * (dot_err1 + beta1m1*pow(abs(actl_err1),
   (double)2/3)*sgn(actl_err1));
8 den = abs(ddot_err1) + beta2m1 * pow(abs(dot_err1) + beta1m1*pow(abs(
   actl_err1),(double)2/3),0.5);
9     if(den == 0){
10         uldot = 0;
11     }else{
12         uldot = (-alpha1 * num/den);
13     }
14
15
16 //Integramos para suavizar
17 u1 = (u1 + h*(uldot_ant+uldot)/2);
18 uldot_ant = uldot;
19
20 //Inyeccion a la salida del control
21 rtn = GT_SetMtrBias(-u1);
22 rtn = GT_Update();
23 rtn = GT_ClrSts();

```

Anidado de Tercer Orden

```

1 //DIFERENCIANDO CON LEVANT
2 derivs = dL_m1.diffL((double)actl_err1); //derivs es un arreglo con los
   datos que resultan de diferenciar
3
4 dot_err1 = derivs[0]; //Primera derivada
5 ddot_err1 = derivs[1]; //Segunda derivada
6
7 evaluar1 = dot_err1+ betalm1 * pow(abs(actl_err1),(double)2/3)*sgn(
   actl_err1);
8 evaluar2 = ddot_err1 + beta2m1 *pow(pow(abs(dot_err1),3.0)+pow(abs(
   actl_err1),2.0),(double)1/6) * sgn(evaluar1);
9
10 uldot = (-alpha1 * sgn(evaluar2));
11
12 //Integramos para suavizar
13 u1 = (u1 + h*(uldot_ant+uldot)/2);
14 uldot_ant = uldot;
15
16 //Inyeccion a la salida del control
17
18 rtn = GT_SetMtrBias(-u1);
19 rtn = GT_Update();
20 rtn = GT_ClrSts();

```

Cuasi-Continuo de Cuarto Orden

```

1 //DIFERENCIANDO CON LEVANT
2 derivs = dL_m1.diffL((double)actl_err1); //derivs es un arreglo con los
   datos que resultan de diferenciar
3
4 dot_err1 = derivs[0]; //Primera derivada
5 ddot_err1 = derivs[1]; //Segunda derivada
6 dddot_err1 = derivs[2]; // Tercera derivada
7
8 num = dddot_err1 + beta3m1*( ddot_err1 + beta2m1*pow( abs(dot_err1) +
   betalm1*pow(abs(actl_err1),(double)3/4) ), (double)-1/3) * (
   dot_err1 + betalm1*pow(abs(actl_err1),(double)3/4) * sgn(actl_err1)
   ) ) * pow(abs(ddot_err1) + beta2m1*pow( abs(dot_err1) + betalm1
   *pow( abs(actl_err1), (double)3/4) , (double)2/3) , (double)-1/2);
9 den = abs(dddot_err1) + beta3m1*pow( abs(ddot_err1) + beta2m1*pow(abs(
   dot_err1) + betalm1*pow( abs(actl_err1) , (double)3/4) , (double)
   2/3) , (double)1/2);
10     if(den == 0){
11         ulddot = 0;

```

```

12     }else{
13         ulddot = (-alpha1*(num/den));
14     }
15
16     //Integramos para suavizar
17     uldot = (uldot + h*(ulddot_ant+ulddot)/2);
18     u1 = (u1 + h*(uldot_ant+uldot)/2);
19     ulddot_ant = ulddot;
20     uldot_ant = uldot;
21
22     rtn = GT_SetMtrBias(-u1);
23     rtn = GT_Update();
24     rtn = GT_ClrSts();

```

Anidado de Cuarto Orden

```

1  //DIFERENCIANDO CON LEVANT
2  derivs = dL_m1.diffL((double)actl_err1); //derivs es un arreglo con los
      datos que resultan de diferenciar
3
4  dot_err1 = derivs[0]; //Primera derivada
5  ddot_err1 = derivs[1]; //Segunda derivada
6  dddot_err1 = derivs[2]; // Tercera derivada
7
8  evaluar1 = dot_err1+ beta1m1 * pow(abs(actl_err1),0.75)*sgn(actl_err1);
9  evaluar2 = ddot_err1 + beta2m1 *pow( pow( abs(dot_err1) ,4.0) + pow(
      abs(actl_err1),3.0), (double)1/6 ) * sgn(evaluar1);
10 evaluar3 = dddot_err1 + beta3m1*pow( pow(abs(ddot_err1) , 6.0) + pow(
      abs(dot_err1) , 4.0) + pow(abs(actl_err1) , 3.0) , (double)1/12 )
      * sgn(evaluar2 );
11
12  ulddot = (-alpha1 * sgn(evaluar3));
13
14  //Integramos para suavizar
15  uldot = (uldot + h*(ulddot_ant+ulddot)/2);
16  u1 = (u1 + h*(uldot_ant+uldot)/2);
17  ulddot_ant = ulddot;
18  uldot_ant = uldot;
19
20  rtn = GT_SetMtrBias(-u1);
21  rtn = GT_Update();
22  rtn = GT_ClrSts();

```



Figura A.1: Tarjeta GT-400-SV.

A.2. Hardware

A.2.1. Tarjeta GT-400-SV

Para la implementación de los algoritmos de control, desde el entorno de Windows, se utilizó la tarjeta GT400-SV la cual tiene un campo de aplicaciones amplio.

GT-400-SV de la serie de controladores de movimiento desarrollado por Googol Technology Ltd es un tipo de propósito general, de 4 ejes, controladores de movimiento basados en DSP. La serie tiene varios modelos de productos que se enumeran a continuación, ampliamente utilizado en aplicaciones que van desde el simple punto a punto de equipos de control de movimiento a los equipos de control de perfil muy complicado de movimiento, tales como máquinas de medición, máquinas de perforación PCB, SMT, máquina de grabado, tornos, centros de mecanizado, corte por chorro de agua, máquinas de corte por láser, robots y más [19].

Características principales:

- Puede controlar 4 servos / motores paso a paso.
- Periodo de muestreo programable. El periodo de interpolación mínimo de cuatro ejes es de 200us (GT-400-SG-S es de 400 us). El período mínimo de control de un solo eje de punto a punto el movimiento es 25US.
- Los modos de movimiento: movimiento de punto a punto, interpolación lineal, interpolación circular, control de velocidad, manual de entrada del generador de impulsos y ruedas de fricción electrónica.

- Curva programable trapecio, y S-curva de perfil de velocidad y los parámetros de actualización sobre la marcha.
- Todos los registros de los parámetros de cálculo y los parámetros de la trayectoria de planificación son de 32 bits.
- Se puede establecer el siguiente error límite, límite de la aceleración y el límite de la producción, para garantizar un control seguro y fiable.
- PID (Proporcional-Integral-Derivativo) un filtro digital con velocidad y la aceleración feedforward, y con el límite de la indemnización integral y la parcialidad y filtro de paso bajo.
- Movimiento coordinado de hasta 4 ejes, interpolación lineal, 2-4 ejes y 2 ejes de interpolación circular.
- Función de interpolación continua.
- Interrupción de eventos programables: Alarma de entrada externa, interrupción de evento y tiempo de interrupción.
- EEPROM para actualizar el firmware y los parámetros.
- Controladores y DLL para Windows98/2000/NT, C y C + + biblioteca de funciones.

En la figura A.2 se muestra el Filtro digital, fue modificado el parámetro "net difference compensation". Con el fin de poder aplicar el algoritmo Twisting de control:

A.2.2. Sistema de Servo Amplificadores PY2

Para la calibración de los encoders, para darle la potencia necesaria a los actuadores, se utilizó el sistema de Servo Amplificadores PY2. También gracias al sistema de alarma con que cuentan estos amplificadores, se detectó con mayor rapidez algunos problemas técnicos durante este trabajo.

Las unidades PY2 pueden operar desde una entrada de 230 VAC (configuración monofásica o trifásica está disponible) y tiene funciones integradas tales como el frenado dinámico y regenerativo de procesamiento de la energía (resistencia regenerativa

A.

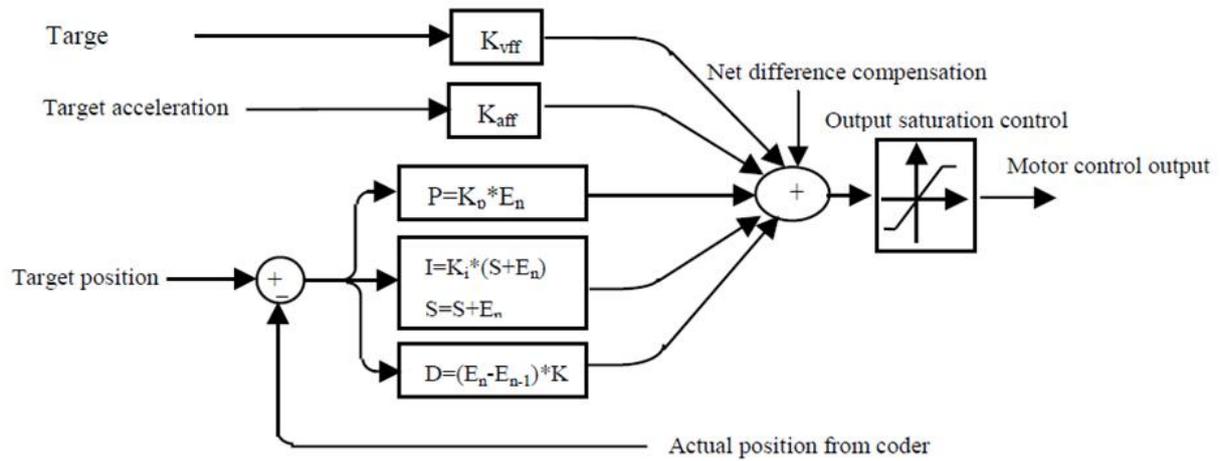


Figura A.2: Filtro digital para cada motor.

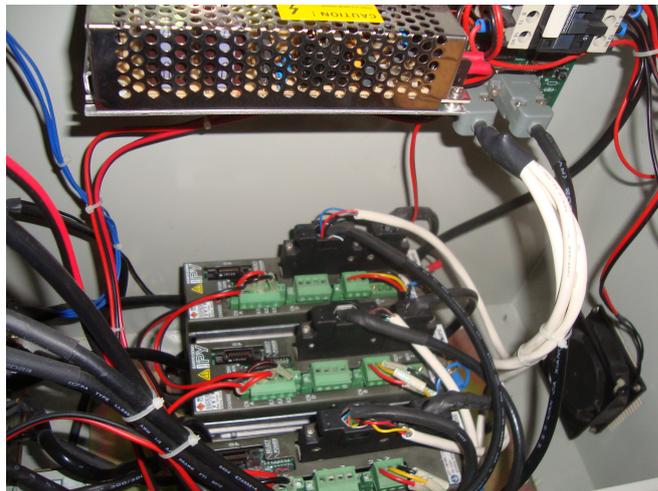


Figura A.3: Amplificadores y el cableado para comunicar con la tarjeta GT-400-SV, motores y encoders del robot.

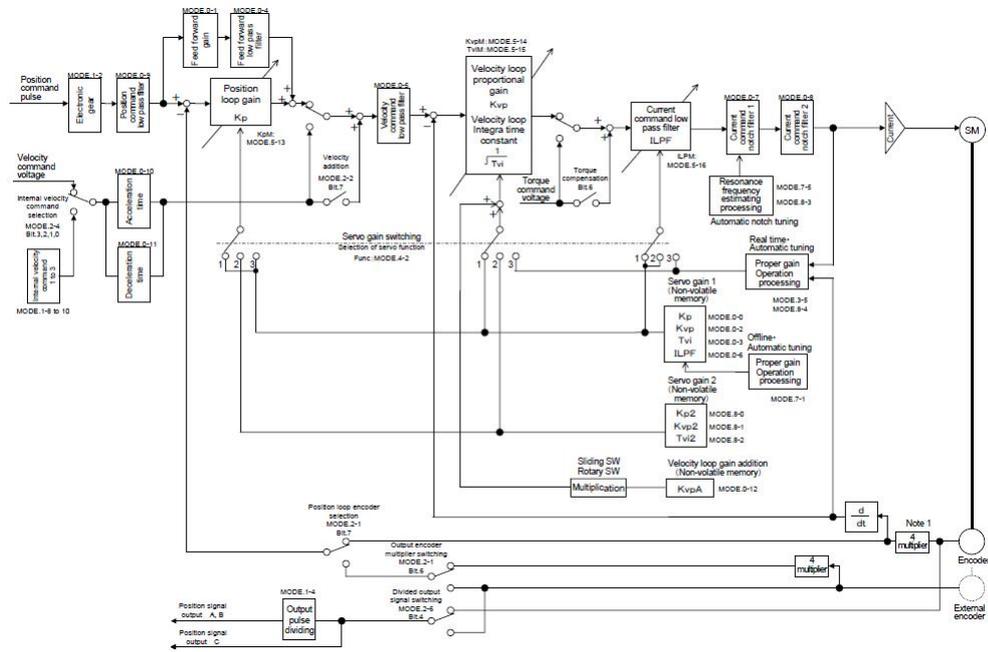


Figura A.4: Parámetros del amplificador PY2.

externa debe ser utilizado en algunas aplicaciones). Las unidades PY2 se fabrican en una instalación de la norma ISO-9001.

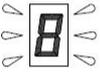
También se modificaron los parámetros del modo de operación 5-15 de los amplificadores para el correcto funcionamiento del motor número tres del robot. Al darle el valor máximo a la constante integral del modo de operación, el movimiento del motor ya no presentaba sobrepasos. Para más detalles técnicos se puede consultar en [[18]].

A.2.3. Calibración de los encoders absolutos

El sistema de servomotores de la unidad GPM2002 no puede operar a menos que los amplificadores PY2 no tengan ningún error en su funcionamiento. Una de las causas recurrentes en sus fallas es la necesaria recalibración de los encoders vía hardware. Para ello se utilizó comunicación RS232 además del software correspondiente. Cuando los amplificadores emitían una señal de error como se muestra en la figura A.7, se tenía que "limpiar" esa señal utilizando el software que se muestra en la figura A.8. Este problema era ocasionado por que los encoders requieren su propia fuente de energía independiente de 3.6 [V] cada uno y cuando se desenergizaban era necesario limpiar esa

A.

Alarm Status

Segment LED Display	Alarm Code ALM8,4,2,1	Abbreviation	Alarm Name	Alarm Clear	Contents
	1000	DE1	Encoder disconnection (sensor error)	Not possible	Disconnection of sensor signal (A, B, C or PS signal) line was detected
		DE2	Serial disconnection (sensor error)	Not possible	Disconnection of sensor signal (PS signal) line was detected
		DE3	Encoder initial error (sensor error)	Not possible	Initial data of motor sensor can not be read in.
		DE4	Serial receiving stop (sensor error)	Not possible	No feedback of absolute position data from absolute sensor
		EXDE	External encoder disconnection	Possible	Disconnection of full close sensor signal line was detected.

Alarm code 0,1 indicates: when Func2/ bit7,6 ="0,0", "0"= output open and "1"= output short.

Figura A.5: Señal de error en el amplificador PY

Esta falla es común en estos equipos.

señal de error.

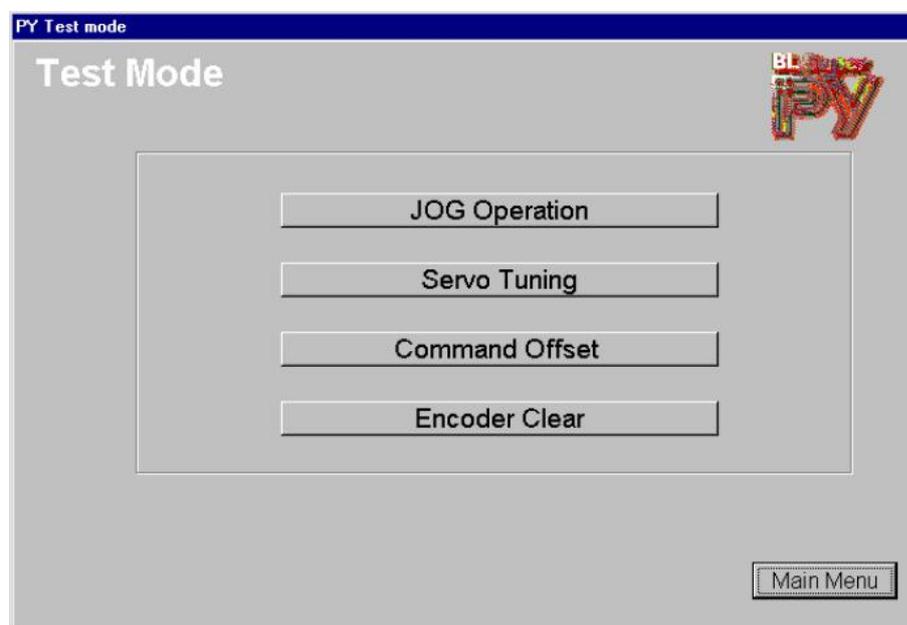


Figura A.6: Software PY para los amplificadores

El boton Encoder Clear sirve para corregir la alarma de error del amplificador PY en cada motor.

Bibliografía

- [1] LEONID FRIDMAN, JAIME MORENO, AND RAFAEL IRIARTE. *Sliding modes after the first decade of the 21st century: State of the art*, 412. Springer, 2011. 2, 3, 20, 42
- [2] ARIE LEVANT. **Universal single-input-single-output (SISO) sliding-mode controllers with finite-time convergence.** *Automatic Control, IEEE Transactions on*, 46(9):1447–1451, 2001. 3
- [3] ARIE LEVANT. **Higher-order sliding modes, differentiation and output-feedback control.** *International journal of Control*, 76(9-10):924–941, 2003. 3, 4, 24, 43, 44
- [4] J-P MERLET. *Parallel robots*, 128. Springer, 2006. 4, 11, 12, 13, 14, 28
- [5] P SARAVIA, DARLY BABETH, HERRERA LÓPEZ, MARLON JHAIR, AND HÉCTOR FABIO QUINTERO RIAZA. **Revisión del estado del arte de manipuladores paralelos.** *Scientia*, 2009. 5, 13
- [6] SÖNKE KOCK AND WALTER SCHUMACHER. **A parallel xy manipulator with actuation redundancy for high-speed and active-stiffness applications.** In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 3, pages 2295–2300. IEEE, 1998. 6
- [7] WEIWEI SHANG, SHUANG CONG, YAOXIN ZHANG, AND YANYANG LIANG. **Active joint synchronization control for a 2-DOF redundantly actuated parallel manipulator.** *Control Systems Technology, IEEE Transactions on*, 17(2):416–423, 2009. 6, 13
- [8] WEIWEI SHANG AND SHUANG CONG. **Nonlinear adaptive task space control for a 2-DOF redundantly actuated parallel manipulator.** *Nonlinear Dynamics*, 59(1-2):61–72, 2010. 6, 28, 51
- [9] WEIWEI SHANG AND SHUANG CONG. **Adaptive compensation of dynamics and friction for a planar parallel manipulator with redundant actuation.** In *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, pages 507–512. IEEE, 2010. 6

- [10] NAG-IN KIM, CHONG-WON LEE, AND PYUNG-HUN CHANG. **Sliding mode control with perturbation estimation: application to motion control of parallel manipulator.** *Control engineering practice*, 6(11):1321–1330, 1998. 6
- [11] ZHEN QI, JOHN E MCINROY, AND FARHAD JAFARI. **Trajectory tracking with parallel robots using low chattering, fuzzy sliding mode controller.** *Journal of Intelligent and Robotic Systems*, 48(3):333–356, 2007. 6
- [12] MARCO ANTONIO RUIZ GASCA. **Comparación de un controlador clásico y un controlador discontinuo en un sistema sobreactuado.** Tesis de licenciatura, 2012. Facultad de Ingeniería UNAM. 6, 41
- [13] MARK W SPONG AND MATHUKUMALLI VIDYASAGAR. *Robot dynamics and control.* John Wiley & Sons, 2008. 11, 12, 14, 37
- [14] ABB Group- Automation and Power Technologies catálogo de robots industriales. <http://new.abb.com/products/robotics/industrial-robots/>. Accessed: 2014-08-13. 12
- [15] S. SHANKAR SASTRY RICHARD M. MURRAY, ZEXIANG LI. **A Mathematical Introduction to Robotic Manipulation.** *CRC Press*, 1994. 15, 18
- [16] YURI SHTESSEL, CHRISTOPHER EDWARDS, LEONID FRIDMAN, AND ARIE LEVANT. *Sliding mode control and observation.* Springer, 2014. 18, 19, 20, 21, 22, 23, 41, 42
- [17] ARIE LEVANT. **Homogeneity approach to high-order sliding mode design.** *Automatica*, 41(5):823–830, 2005. 24
- [18] SANYO DENKI CO. **PY2 Servo Amplifier Instruction Manual.** 2003. 25, 27, 63, 99
- [19] GOOGOL TECHNOLOGY. **Programming Manual for GT Motion Controller.** 2003. 27, 63, 96
- [20] YAO-XIN ZHANG, SHUANG CONG, WEI-WEI SHANG, ZE-XIANG LI, AND SHI-LONG JIANG. **Modeling, identification and control of a redundant planar 2-DOF parallel manipulator.** *International Journal of Control, Automation, and Systems*, 5(5):559–569, 2007. 28, 30
- [21] L.F. PENIN. C. BLAGUER R. ARACIL BARRIENTOS, A. **Fundamentos de control y robótica.** *McGaw-Hill*, 1997. 29
- [22] WILLIAMS II R.L. **Inverse kinematics for planar parallel manipulators.** *Ohio University*, 1999. 30

- [23] L. MEIROVICHM. *Methods of analytical dynamics*. *McGaw-Hill*, 1970. 35
- [24] ARIE LEVANT. *Quasi-continuous high-order sliding-mode controllers*. *Automatic Control, IEEE Transactions on*, 50(11):1812–1816, 2005. 43, 44
- [25] ARIE LEVANT AND ALON MICHAEL. *Adjustment of high-order sliding-mode controllers*. *International Journal of Robust and Nonlinear Control*, 19(15):1657–1672, 2009. 49