



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
MAESTRÍA EN INGENIERÍA ELÉCTRICA – CONTROL

IMPLEMENTACIÓN DE CONTROLADORES CONTINUOS POR MODOS
DESLIZANTES A UN ROBOT PARALELO DE 2-GDL

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
RUIZ GASCA MARCO ANTONIO

TUTOR
DR. LEONID FRIDMAN,
FACULTAD DE INGENIERÍA

MÉXICO, D. F. OCTUBRE 2014

Jurado asignado:

Presidente: DR. ALVREZ ICAZA LONGORIA LUIS A.
Secretario: DR. MORENO PÉREZ JAIME ALBERTO
Vocal: DR. FRIDMAN LEONID
1er Suplente: DR. ARTEAGA PÉREZ MARCO ANTONIO
2do Suplente: DR. DÁVILA MONTOYA JORGE ÁNGEL

Lugar donde se realizó la tesis:

Universidad Nacional Autónoma de México, D.F., Facultad de Ingeniería

TUTOR DE TESIS:
DR. LEONID FRIDMAN

FIRMA

Dedicatoria y agradecimientos

Dedicatorias personales:

Este trabajo es producto del contacto que tuve con la gente que conocí durante y antes de esta etapa final de la maestría, dedico especial agradecimiento al Dr. Fridman por asesorarme y guiarme en éste último año, al Dr. Rafael y la Dra. Liset por su apoyo desde la licenciatura, y por supuesto a todo el equipo de modos deslizantes, en especial a Pablo, quien también se integro a éste mismo proyecto. Dedico personalmente este trabajo a mi familia, empezando por mis padres, Aurora y Reyes, también a mis hermanas; Casandra y Estefania. A mis abuelos a quienes les guardo profundo amor y respeto. A mis tíos, Adriana, Hector, José, quienes también me han apoyado en todo momento

A mis amigos, amigas y personas que me han acompañado los últimos años, Rin, Friederike, Silvia, Cuauhtémoc, Alan, Fernando, Jasmine, Arturo, Adriana, Mildred, Sebastian, Isaac, entre muchos otros.

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.

PAPIIT113613

También el autor agradece al Consejo Nacional de Ciencia y Tecnología por el apoyo recibido, sin el cuál no hubiera sido posible éste trabajo.

En la imagen de abajo un dibujo realizado por el autor de esta tesis, el cuál llevó más de 5 años de elaboración. El trabajo que se ama lleva su tiempo, sólo lo necesario.



In Xochitl in Cuicatl (La flor y el canto): “Poesía” - Forma poética del náhuatl. Contacto: marco_3_antonio@hotmail.com

Contenido

Lista de figuras	ix	
Lista de tablas	xi	
1	Introducción	1
1.1	Presentación	1
1.2	Estado del arte	2
1.3	Planteamiento del problema	6
1.4	Objetivo	6
1.5	Contribución	7
1.6	Metodología	7
1.7	Estructura de la tesis	8
2	Modelo matemático del robot paralelo	9
2.1	Cinemática del robot paralelo	9
2.2	Modelo dinámico: Cadena cinemática abierta	11
2.3	Modelo en 12 ecuaciones diferenciales y 4 algebraicas	13
2.3.1	Ecuaciones en espacio de estados $\dot{x} = A_0x + B_0u$	13
2.3.2	Restricciones algebraicas	15
2.3.3	Modelo dinámico sin fuerzas de restricción	15
2.4	Modelo dinámico del robot paralelo con fuerzas de restricción	16
2.4.1	Modelo en espacio de estados	20
2.5	Definición del error y superficie deslizante	21

2.6	Estabilidad en lazo cerrado con control PD	22
3	Algoritmos por modos deslizantes	25
3.1	Sistema mecánico de segundo orden con incertidumbre acoplada	25
3.2	Cambio de notación	26
3.3	Algoritmo convencional de primer orden	27
3.3.1	Condiciones de convergencia mediante función de Lyapunov	27
3.4	Modos deslizantes integrales (MDI)	28
3.4.1	Caso lineal	28
3.5	Super-Twisting	28
3.5.1	Condiciones necesarias para la convergencia del algoritmo Super-Twisting con función de Lyapunov	29
3.6	Twisting	30
3.6.1	Condiciones de convergencia con función de Lyapunov	30
3.7	Control Integral Discontinuo (CID)	31
3.7.1	Condiciones necesarias para la convergencia del algoritmo CID	31
3.7.2	Función $\psi(\alpha, \lambda)$	32
3.8	Super-Twisting Tercer Orden (3-STA)	33
3.8.1	Condiciones necesarias para la convergencia del algoritmo 3-STA	33
3.9	Diferenciador robusto exacto	34
4	Ajuste de ganancias para controladores	37
4.1	Seguimiento en espacio de uniones	37
4.2	Superficie deslizante	37
4.2.1	Superficie estandar	37
4.2.2	Superficie para MDI	38
4.3	Cota de la perturbación	39

4.4	Cota de la derivada de perturbación	39
4.5	Valores de las constantes obtenidas	44
4.6	Ajustes Super-Twisting	44
4.7	Ajuste Twisting	45
4.8	Ajuste para 3-STA y CID	45
4.9	Diferenciador con el experimento nominal	47
5	Resultados de simulación	49
5.1	Experimento a realizar	50
5.2	Parámetros usados	50
5.3	Modos deslizantes convencionales	51
5.4	Super-Twisting	52
5.5	Twisting	52
5.6	Control Integral Discontinuo (CID)	54
5.7	Super-Twisting Tercer Orden (3-STA)	54
6	Experimentos	57
6.1	Sistema de prueba GPM 2002	57
6.2	Experimento nominal PD	60
6.3	Algoritmo de primer orden	60
6.4	Super-Twisting	61
6.5	Resultados de 3-STA	64
6.6	Resultados de Control Integral Discontinuo (CID) . .	65
6.7	Experimentos con Diferenciador de Segundo Orden .	68
	6.7.1 3-STA y diferenciador	68
	6.7.2 Control integral discontinuo y diferenciador .	69
6.8	Múltiples líneas	69
7	Conclusiones	73
7.1	Análisis de resultados	73
	7.1.1 Errores de seguimiento	73
	7.1.2 Señales de control	76
	7.1.3 Tiempos de ejecución	78

CONTENIDO

7.2	Comentarios finales	78
A	Apendices	81
A.1	Tarjeta GT-400-SV	81
A.2	Filtro digital	81
A.3	Servoamplificadores	82
A.4	Relación voltaje-torque	83
A.5	Entorno de programación C++	84
	Bibliografía	85

Lista de figuras

1.1	Ejemplos de robots paralelos	5
2.1	Parámetros geométricos del robot	10
2.2	Cadena cinemática abierta	11
2.3	Configuraciones del robot paralelo:	20
4.1	Fricción en uniones activas	40
4.2	Derivada de la perturbación	43
4.3	Función Ψ 1	46
4.4	Función Ψ 2	46
4.5	Posiciones angulares deseadas y de los actuadores . .	47
4.6	Resultados de diferenciador	48
5.1	Diagrama Simulink	49
5.2	Trayectoria recta	50
5.3	Modos deslizantes convencionales Simulación	52
5.4	Super Twisting Simulación	53
5.5	Twisting Simulación	53
5.6	CID Simulación	54
5.7	3-STA Simulación	55
6.1	Sistema experimental Robot Paralelo	58
6.2	Sistema experimental diagama a bloques 1	58
6.3	Sistema experimental diagama a bloques 2	59
6.4	Salida medida, experimento nominal PD	61
6.5	Salida medida, experimento algoritmo de primer orden	62

LISTA DE FIGURAS

6.6	Señal de control experimento algoritmo primer orden	62
6.7	Salida medida, experimento algoritmo super-twisting	63
6.8	Señal de control experimento super-twisting	63
6.9	Salida medida experimento 3-STA	64
6.10	Señal de control 3-STA	65
6.11	Salida medida experimento CID	66
6.12	Señal de control CID	66
6.13	Experimentos en papel	67
6.14	3-STA y diferenciador de segundo orden	68
6.15	Control 3-STA con diferenciador segundo orden . . .	69
6.16	CID y diferenciador de segundo orden	70
6.17	Control CID con diferenciador segundo orden	70
6.18	Experimentos con trayectorias consecutivas	71
6.19	Experimento multiples líneas CID	72
6.20	Experimento multiples líneas 3-STA	72
7.1	Errores considerados	74
7.2	$e_{ss} = e_{imax} - e_{imin}$ pasado $t > t_1$	75
7.3	Señales de control continuas	77
7.4	Señales de control continuas (sólo un motor)	77
A.1	Tarjeta GT-400-SV	81
A.2	Filtro digital	82
A.3	Sistema de amplificadores	82
A.4	Parámetros del amplificador PY2	83
A.5	La relación voltaje torque es configurable en el sistema de servoamplificadores gracias al controlador interno	83
A.6	Entorno C++	84

Lista de tablas

1.1	Controladores Estandar	5
4.1	Cotas de perturbaciones	44
4.2	Tabla de constantes 1	44
5.1	Parametros	51
7.1	RSME	74
7.2	Errores en estado estable	75
7.3	Norma maxima	76
7.4	Norma maxima	77
7.5	Tiempos	78
7.6	Controladores	80

LISTA DE TABLAS

Introducción

1.1 Presentación

En este trabajo se presentan los resultados experimentales obtenidos, al aplicar algoritmos por modos deslizantes a un robot paralelo. Por esta razón se tienen antecedentes de ambas vertientes, por un lado lo que se ha trabajado con el mismo sistema experimental, y por otro lado la teoría relacionada a los modos deslizantes.

La información disponible para robots manipuladores es vasta, por ejemplo en [1], [2] se encuentran bases teóricas al igual que ejemplos de aplicaciones actuales. En referencias relacionadas a los robots manipuladores paralelos, la cual también es amplia, por mencionar algunas como [3] y [4] podemos encontrar análisis cinemático y dinámico. También hay variedad ejemplos de aplicaciones en robots paralelos (asistencia médica, montaje, ensamblado de maquinaria y circuitería, maniobras en construcciones, entre otras). En este trabajo se trata de un robot manipulador del tipo RRR, (R para uniones rotatorias, P para prismáticas). Es un robot paralelo sobre-actuado (número de entradas mayor al número de grados de libertad), considerando a las dos posiciones cartesianas del efector final como grados de libertad.

1.2 Estado del arte

Este tipo de manipulador paralelo ha atraído la atención de investigadores en las últimas dos décadas. Un ejemplo de trabajos donde implementar algoritmos de control robusto en este tipo de sistemas, se puede ver en [5], en el cual se combinan modos deslizantes con control difuso, sin embargo se utiliza un algoritmo de primer orden que requiere el diseño de una superficie.

En el trabajo [6] se presenta un análisis geométrico resultado de utilizar este tipo de configuración paralela (RRR), con un enfoque basado en el uso de Jacobianos (para la cinemática inversa) que reduce efectos producidos por fuerzas internas, aunque no se utiliza una herramienta formal de análisis (función de Lyapunov), que garantice y provea información acerca del tipo de estabilidad del punto de operación.

En el trabajo de [7] se propone un esquema basado en la cinemática inversa, probando diferentes algoritmos, (PD, PD aumentado, Torque Computado), algo de importancia que es reportado en este trabajo, es que conforme aumenta la complejidad del algoritmo, también es menos eficiente en ejecución de experimentos (tarda más tiempo cada prueba).

También se han hecho desarrollos totalmente basados en el problema de la dinámica inversa, como se puede encontrar en [8]. Aunque el esquema presentado es muy útil (sobre todo para resolver el problema de la sobreactuación), queda sin resolver el problema de la robustez.

En los trabajos de [9], [10], [11] se utilizan técnicas de control adaptable, con las cuales ofrecen pruebas de convergencia para esta aplicación, asegurando estabilidad exponencial, considerando un modelo no lineal. Los autores ofrecen las pruebas de convergencia del error de seguimiento. Cabe mencionar que estos autores utilizan el mismo equipo de pruebas que en este trabajo.

En el trabajo de [12], retoma el estudio de los efectos en la cinemática debidos a la redundancia. Se discute acerca de las implicaciones de incrementar los actuadores redundantes.

Finalmente uno de los trabajos más recientes para este tipo de sistemas es [13], se presenta un estudio de planeación eficiente de trayectorias en términos energéticos.

En todos los casos anteriores, lo que se desea es alta eficiencia en términos de minimizar el error de posición y velocidad durante el seguimiento de trayectorias deseadas, además de que el problema se vuelve más complejo al incorporar la necesidad de incluir un modelo y parámetros dentro de los algoritmos. Por tanto viene un compromiso entre complejidad del algoritmo y lo que éste pueda ofrecer en beneficio de la tarea de control, ya que también se debe tomar en cuenta las limitantes del procesamiento. Algunos ejemplos de manipuladores de este tipo utilizados en aplicación se encuentran en la Figura 1.1.

Para abordar el problema de control robusto planteado en el párrafo anterior, se pueden utilizar técnicas de control por modos deslizantes.

La teoría clásica principal de los modos deslizantes, en su versión en inglés, está reportada por el Prof. Utkin en [14].

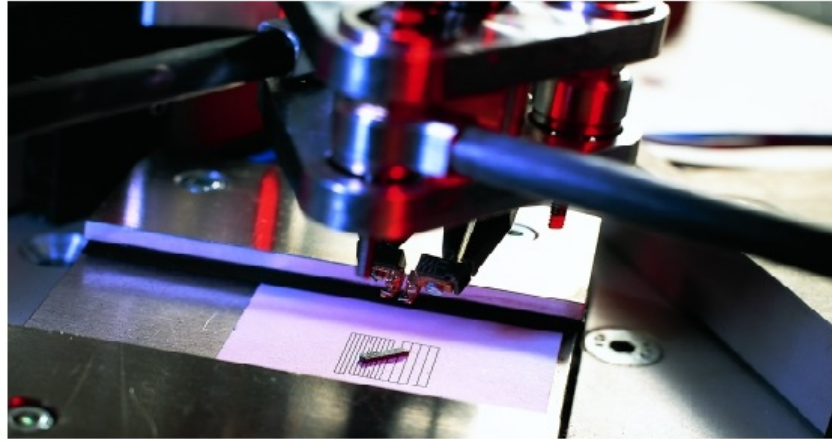
Para los algoritmos de segundo orden, en este trabajo se menciona el Twisting, en donde primero se presentan las condiciones de convergencia bajo la idea de curvas mayorantes en [15], dicho algoritmo, al ser de segundo orden asegura convergencia en tiempo finito de posición y velocidad (para un sistema de segundo orden), sin embargo el problema del *chattering* no es resuelto, lo cual implica un daño al sistema en su implementación directa sobre actuadores. En [16] se presenta el análisis para convergencia en tiempo finito para sistemas mecánicos de segundo orden utilizando una función de Lyapunov. Otro tipo de algoritmo por modos deslizantes es el Super-Twisting,

1. INTRODUCCIÓN

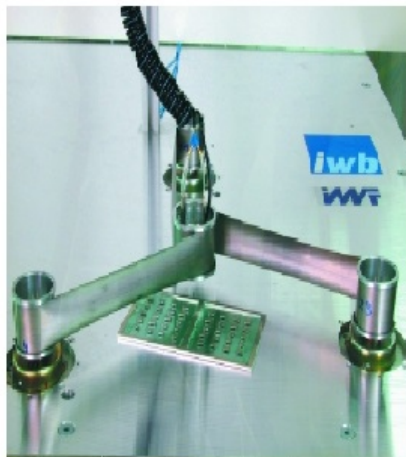
del cual lo atractivo para nuestro propósito es que nos ofrece una señal de control **continua**, según en [17] para alguna perturbación Lipshitz acotada, y eligiendo las apropiadas ganancias, este algoritmo estabiliza a la superficie deslizante en tiempo finito, aunque asintóticamente a los estados (para un sistema de segundo orden), después en [18] se presenta una función de Lyapunov para modos deslizantes de segundo orden, esta es una herramienta muy útil para estudiar condiciones y propiedades de convergencia. Además si se considera a la posición como salida medida, es posible utilizar el Super-Twisting como un diferenciador por modos deslizantes de segundo orden [19] (de esta manera obtenemos una señal de velocidad **continua**).

Otra de las técnicas posibles actualmente, es la utilización de “Modos deslizantes de alto orden” [20]. La idea principal detrás de la utilización de modos deslizantes de alto orden, es agregar un integrador para suavizar la señal de control, sin embargo esto implica la utilización de más información (sucesivas derivadas), y si se posee la información por ejemplo de \ddot{x} , entonces existen alternativas a los modos deslizantes para resolver el problema de robustez. En adición a estos esquemas mencionados, se probarán dos algoritmos nuevos, son versiones del Super-Twisting pero de tercer orden, se abundará con más detalle acerca de ellos en el Capítulo 3 de esta tesis.

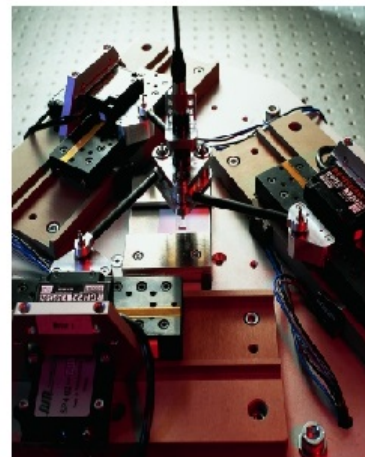
Parte de la literatura de modos deslizantes, para fines de revisión bibliográfica de trabajos actuales se encuentra en [21], [22]. Como resumen de los algoritmos descritos anteriormente, algunas de las estrategias ya existentes de control por modos deslizantes, aplicados a sistemas inciertos de segundo orden (como podría ser un sistema mecánico), se puede ver en la Tabla 1.1, donde se muestran los resultados de implementar diferentes algoritmos por modos deslizantes, aplicados a un doble integrador con dinámicas inciertas acopladas al canal de control.



a)



b)



c)

Figura 1.1: Ejemplos de robots paralelos - a) ejemplo de robot paralelo y un microchip en un proceso de ensamblado, b) Dispositivo de manipulación de alta precisión (IWF), c) Robot de micromontaje con estructura plana trípode.

Algoritmo	Convergencia	Chattering/Control	Información	Perturbación $\delta(t)$
M.D. Convencionales	Exponencial	✓/Discontinuo	x, \dot{x}	$ \delta(t) \leq D$
Súper Twisting	Exponencial	Continuo	x, \dot{x}	$ \frac{d}{dt}\delta(t) \leq \Delta$
Twisting	Tiempo Finito	✓/Discontinuo	x, \dot{x}	$ \delta(t) \leq D$
Tercer Orden	Tiempo Finito	Continuo	x, \dot{x}, \ddot{x}	$ \delta(t) \leq D$

Tabla 1.1: Propiedades teóricas de los controladores estandares por modos deslizantes. En esta Tabla se muestran los resultados de implementar diferentes algoritmos por modos deslizantes, aplicados a un doble integrador con dinámicas inciertas acopladas al canal de control.

1.3 Planteamiento del problema

Para nuestro caso, considerando tres entradas de los actuadores, y como salidas las dos posiciones cartesianas del robot, se tiene el siguiente sistema mecánico sobreactuado, que se encuentra bajo incertidumbres debido a variaciones paramétricas, además de perturbaciones por factores externos al sistema. A partir de las ecuaciones dinámicas,

$$M(z)\ddot{z} + C(z, \dot{z})\dot{z} + f(\dot{z}) = \tau + \delta(t, z, \dot{z}), \quad (1.1)$$

donde:

- $z \in \mathbb{R}^6$ son las coordenadas de posición en el espacio de uniones.
- $M_{6 \times 6}(z)$ es la matriz de inercia.
- $C_{6 \times 6}(z, \dot{z})$ es la matriz de Coriolis y fuerzas centrífugas.
- $f(\dot{z}) \in \mathbb{R}^6$ son las fuerzas de fricción.
- $\tau \in \mathbb{R}^6$ es el torque en cada unión (se considerarán en el modelo despreciables las uniones inactivas).
- $\delta(t, \dot{z}) \in \mathbb{R}^6$ incertidumbres.

Se desea diseñar una ley de control para seguimiento de trayectoria. Para ello se propone el siguiente controlador:

$$u = u_0 + u_1 \quad (1.2)$$

donde el u_0 es el PD y el u_1 algún controlador por modos deslizantes.

1.4 Objetivo

El objetivo es investigar qué sucedería si se aplican distintos controladores por modos deslizantes, aprovechando estas propiedades:

- Robustez ante perturbaciones e incertidumbres $\delta(t, z, \dot{z})$, acotadas $D > |\delta|$ y Lipschitz $\Delta > |\frac{d}{dt}\delta|$, dependiendo el caso.
- Convergencia en tiempo finito (máxima precisión posible).

Y tras un análisis, realizar la comparación de resultados para evaluar bajo algunos criterios cada los resultados.

1.5 Contribución

La contribución es el resultado del trabajo experimental realizado sobre este robot. Una labor de ajuste y sintonización, para estudiar las diferencias experimentales entre cada controlador implementado. Dicho resultado es la información obtenida y nos permite comprobar, de que manera se conservan o cambian las propiedades esperadas al aplicar algoritmos sobre el robot.

1.6 Metodología

Los pasos utilizados para este trabajo se enlistan en los siguientes puntos:

- Modelado matemático del sistema.
- Análisis del conjunto sistema-controlador mediante función de Lyapunov.
- Ajuste de constantes para controladores.
- Simulación numérica.
- Pruebas experimentales.
- Análisis de resultados.

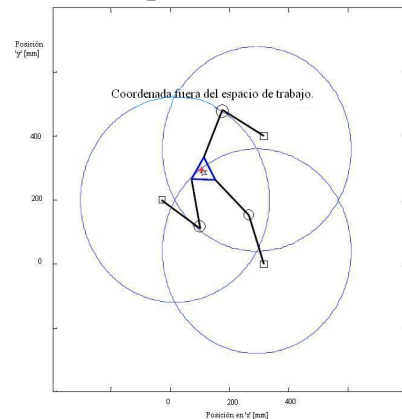
1.7 Estructura de la tesis

Este trabajo esta conformado, primero por una introducción que da una descripción general del problema, publicaciones relevantes más relacionadas y estado actual del tema. Seguido a esta sección se aborda el modelo del robot, las propiedades del sistema que nos pueden ser útiles, con ayuda de un análisis matemático. En el Capítulo 3 se presentan los diferentes controladores empleados, además de las condiciones bajo las cuales estos son convergentes. El Capítulo 4 contiene el procedimiento de los ajustes a las ganancias de los controladores para poder implementarlos. En el Capítulo 5, titulado “Resultados de simulación” se encuentran las graficas considerando las condiciones bajo las cuales las ganancias apropiadas hacen funcionar el algoritmo, para entonces poder probar los algoritmos en el esquema real. En el Capítulo 6 se presentan las gráficas obtenidas en experimentos, implementando los algoritmos propuestos. Finalmente en el Capítulo 6 algunas conclusiones y comentarios finales al respecto del trabajo realizado.

Modelo matemático del robot paralelo

En esta sección se presenta el estudio del modelo matemático del robot. El modelo es una herramienta que permite conocer las propiedades de nuestro sistema, como la estabilidad de su punto de operación. También esta información nos permitirá en el capítulo siguiente poder hacer simulaciones numéricas

previas a implementación.



2.1 Cinemática del robot paralelo

En esta sección abordaremos la cinemática del robot paralelo. En la Figura 2.1 se muestran los parámetros geométricos del robot. La cinemática directa (2.1) consiste en obtener la posición del efector final $p = (p_x, p_y)$ a partir de los valores de los ángulos $z = (\theta_{a1}, \theta_{a2}, \theta_{a3}, \theta_{b1}, \theta_{b2}, \theta_{b3})$, por simplificación, sólo cuando por motivos de espacio se requiera escribiremos $z = (a_1, a_2, a_3, b_1, b_2, b_3)$. Medi-

2. MODELO MATEMÁTICO DEL ROBOT PARALELO

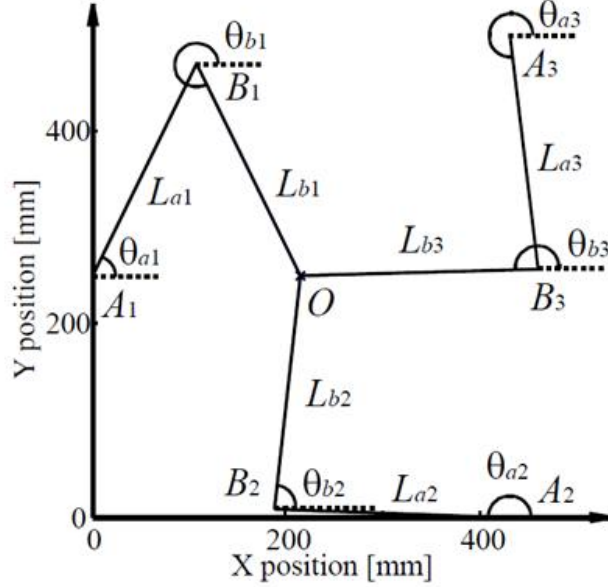


Figura 2.1: Parámetros geométricos del robot - En este diagrama se aprecian las variables del robot, 6 posiciones angulares de los eslabones y 2 cartesianas del efector final.

ante la siguiente expresión:

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{pmatrix} p_{x0} + l \cos a_i + l \cos b_i \\ p_{y0} + l \sin a_i + l \sin b_i \end{pmatrix}, i = 1, 2, 3. \quad (2.1)$$

La cinemática inversa (2.2) consiste en obtener los valores de los ángulos $z = (a_1, a_2, a_3, b_1, b_2, b_3)$, bajo las condiciones iniciales conocidas x_0, y_0 a partir de la posición del efector final $p = (p_x, p_y)$

$$a_i = \arctan \left(\frac{y_0 - p_x}{x_0 - p_y} \right) + \arccos \frac{d_i}{2l} \quad (2.2)$$

$$b_i = \arctan \left(\frac{y_0 - y_{ai} - l \sin a_i}{x_0 - x_{ai} - l \cos a_i} \right) + \arccos \frac{d_i}{2l}; i = 1, 2, 3 \quad (2.3)$$

$$\text{donde, } d_i = \sqrt{(x_0 - p_x)^2 + (y_0 - p_y)^2}$$

2.2 Modelo dinámico: Cadena cinemática abierta

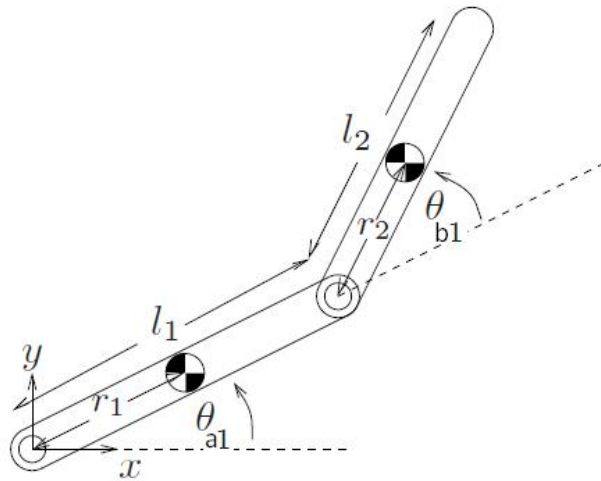


Figura 2.2: Cadena cinemática abierta - En este diagrama se muestran los dos ángulos de cada eslabon, distancias a centro de masas y longitudes.

En esta sección abordaremos una cadena individual de las tres con las que cuenta nuestro robot paralelo con el enfoque de Euler-Lagrange, más adelante combinaremos estas expresiones para obtener un modelo general. En el enfoque de Euler-Lagrange primero se calcula el Lagrangiano,

Basándonos en la Figura 2.2, tenemos las coordenadas cartesianas (\bar{x}_1, \bar{y}_1) para la primera union de la cadena cinemática, y para el efector final se define (\bar{x}_2, \bar{y}_2) , ambos en términos de los ángulos:

$$\begin{aligned}\bar{x}_1 &= r_1 c_1 \\ \bar{y}_1 &= r_1 s_1 \\ \bar{x}_2 &= l_1 c_1 + r_2 c_{12} \\ \bar{y}_2 &= l_1 s_1 + r_2 s_{12}\end{aligned}$$

y sus derivadas

2. MODELO MATEMÁTICO DEL ROBOT PARALELO

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

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 de un sólo eslabon queda de la siguiente forma:

$$T(\theta, \dot{\theta}) = \frac{1}{2} m_{a1} (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} J_{a1} \dot{\theta}_{a1}^2 + \frac{1}{2} m_{b1} (\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2} J_{b1} (\dot{\theta}_{a1}^2 + \dot{\theta}_{b1}^2) \quad (2.4)$$

Luego de agrupar algunos términos y hacer $T = L$, se resuelve la ecuación de Euler-Lagrange (2.6). Después de agrupar algunos términos y ciertas sustituciones, además de llevarlo a una forma matricial, tenemos la siguiente expresión:

$$M_i(z) \ddot{z} + C_i(z, \dot{z}) \dot{z} + f_i(\dot{z}) = \tau_i; i = 1, 2, 3 \quad (2.5)$$

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 torque de la unión activa.

Dichas matrices tienen la forma siguiente:

$$M_i = \begin{pmatrix} \alpha_i & \gamma_i \cos(a_i - b_i) \\ \gamma_i \cos(a_i - b_i) & \beta_i \end{pmatrix} \quad (2.6)$$

$$C_i = \begin{pmatrix} 0 & (\gamma_i \sin(a_i - b_i))\dot{b}_i \\ (\gamma_i \sin(a_i - b_i))\dot{b}_i & 0 \end{pmatrix} \quad (2.7)$$

donde:

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

Propiedades:

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

2.3 Modelo en 12 ecuaciones diferenciales y 4 algebraicas

2.3.1 Ecuaciones en espacio de estados $\dot{x} = A_0x + B_0u$

Retomando la ecuación (2.5), sin tomar en cuenta el término $f_i(\dot{z})$ correspondiente a la fricción, tenemos lo siguiente:

$$M_i(z)\ddot{z} + C_i(z, \dot{z})\dot{z} = \tau_i; \quad i = 1, 2, 3 \quad (2.8)$$

Ahora, con el objetivo de poder aplicar las técnicas de control por modos deslizantes, transformaremos el modelo (2.8) a la forma $\dot{x} = Ax + Bu$, debemos definir el vector de estados de la siguiente manera:

2. MODELO MATEMÁTICO DEL ROBOT PARALELO

$$\begin{aligned}x_1 &= \theta_{ai} \\x_2 &= \theta_{bi} \\x_3 &= \dot{\theta}_{ai} \\x_4 &= \dot{\theta}_{bi}\end{aligned}$$

Por tanto, el modelo queda expresado de la siguiente manera:

$$\dot{x} = A_0(z, \dot{z})x + B_0(z)u \quad (2.9)$$

$$y = (a_1, a_2, a_3)^T \quad (2.10)$$

Donde:

- $x = (x_1, x_2, x_3, x_4)^T$ Es el vector de estados (posiciones y velocidades angulares).
- $y = (\theta_{a1}, \theta_{a2}, \theta_{a3})^T$ Es el vector de mediciones (posición angular de las uniones activas).
- $u = (u_1, u_2, u_3)$ Es el control expresado como el torque de los actuadores.

Y las matrices A_0 y B_0 vienen dadas por las expresiones siguientes:

$$A_0 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & A_i[1, 1] & A_i[1, 2] \\ 0 & 0 & A_i[2, 1] & A_i[2, 2] \end{pmatrix}, A_i = -M_i^{-1}C_i, \quad i = 1, 2, 3. \quad (2.11)$$

$$B_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ B_1[1, 1] & B_1[1, 2] \\ B_i[2, 1] & B_i[2, 2] \end{pmatrix}, B_i = -M_i^{-1}, \quad i = 1, 2, 3. \quad (2.12)$$

Hay que aclarar que el subíndice de A_i , con $i = 1, 2, 3$ denota cada uno de los brazos, y cada matriz A_i es de dimensión 2×2 , por

tanto al decir $A[1, 1]$ se hace referencia al elemento $[1, 1]$ de dicha matriz, el cuál es de dimensión 1. Una manera más correcta de definir éstas matrices para esta sección sería:

$$A_0 = [0_{2 \times 2} I_{2 \times 2}; 0_{2 \times 2} A_i]$$

2.3.2 Restricciones algebraicas

Las restricciones de movimiento de lazo cerrado del robot vienen dadas por la matriz L :

$$L(z) = \begin{bmatrix} x_{a_1} + l \cos(\theta_{a_1}) + l \cos(\theta_{b_1}) - x_{a_2} - l \cos(\theta_{a_2}) - l \cos(\theta_{b_2}) \\ y_{a_1} + l \sin(\theta_{a_1}) + l \sin(\theta_{b_1}) - y_{a_2} - l \sin(\theta_{a_2}) - l \sin(\theta_{b_2}) \\ x_{a_1} + l \cos(\theta_{a_1}) + l \cos(\theta_{b_1}) - x_{a_3} - l \cos(\theta_{a_3}) - l \cos(\theta_{b_3}) \\ y_{a_1} + l \sin(\theta_{a_1}) + l \sin(\theta_{b_1}) - y_{a_3} - l \sin(\theta_{a_3}) - l \sin(\theta_{b_3}) \end{bmatrix} = 0 \quad (2.13)$$

Este conjunto de ecuaciones es linealmente independiente y representa las restricciones en lazo cerrado de las interconexiones entre los eslabones. Si bien se podría agregar las restricciones que tiene el eslabón 2 con el 3, quedaría sobrando ya que con éstas que hemos tomado es suficiente para representar las restricciones de los 3 eslabones en conjunto, es decir que la restricción de 2 con 3 quedaría linealmente dependiente. A partir de aquí se puede observar como un problema que originalmente tenía 6 grados de libertad, queda representado por solamente dos grados de libertad, en términos de las posiciones cartesianas.

2.3.3 Modelo dinámico sin fuerzas de restricción

Para poder plantear las ecuaciones dinámicas del sistema en el espacio de tareas, es necesario recordar el conjunto de ecuaciones de segundo grado sin las restricciones de movimiento. Cuando se combinan las dinámicas de las tres cadenas cinemáticas que componen

2. MODELO MATEMÁTICO DEL ROBOT PARALELO

al robot se tiene el siguiente modelo:

$$M(z)\ddot{z} + C(z, \dot{z})\dot{z} + f(\dot{z}) = \tau_i, \quad (2.14)$$

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

Las matrices M y C tienen la siguiente forma:

$$M = \begin{pmatrix} \alpha_1 & 0 & 0 & \gamma_1 c_{ab1} & 0 & 0 \\ 0 & \alpha_2 & 0 & 0 & \gamma_2 c_{ab2} & 0 \\ 0 & 0 & \alpha_3 & 0 & 0 & \gamma_i c_{ab3} \\ \gamma_1 c_{ab1} & 0 & 0 & \beta_1 & 0 & 0 \\ 0 & \gamma_2 c_{ab2} & 0 & 0 & \beta_1 & 0 \\ 0 & 0 & \gamma_3 c_{ab3} & 0 & 0 & \beta_1 \end{pmatrix} \quad (2.15)$$

$$C = \begin{pmatrix} 0 & 0 & 0 & \gamma_1 c_{ab1} \dot{\theta}_{b1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_2 c_{ab2} \dot{\theta}_{b2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \gamma_3 c_{ab3} \dot{\theta}_{b3} \\ \gamma_1 c_{ab1} \dot{\theta}_{b1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_2 c_{ab2} \dot{\theta}_{b2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_3 c_{ab3} \dot{\theta}_{b3} & 0 & 0 & 0 \end{pmatrix} \quad (2.16)$$

Por simplificación, el término $c_{abi} = \cos(\theta_{ai} - \theta_{bi})$, $i = 1, 2, 3$. Las matrices cumplen con las siguientes propiedades:

- M es una matriz simétrica definida positiva.
- $\frac{dM}{dt} - 2Ci$ es una matriz antisimétrica.

2.4 Modelo dinámico del robot paralelo con fuerzas de restricción

Tomando en cuenta el sistema de cinemática abierta y las fuerzas de restricción debido al sistema de lazo cerrado, el modelo matemático

del robot en el espacio de uniones queda de la siguiente manera:

$$M(z)\ddot{z} + C_i(z, \dot{z})\dot{z} + N(\dot{z}) = \tau + R^T \lambda \quad (2.17)$$

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 de Lagrange que representa la magnitud de las fuerzas de restricción. Las restricciones de movimiento de lazo cerrado del robot vienen dadas por la matriz L :

$$L(q) = \begin{bmatrix} x_{a_1} + l \cos(a_1) + l \cos(b_1) - x_{a_2} - l \cos(a_2) - l \cos(b_2) \\ y_{a_1} + l \sin(a_1) + l \sin(b_1) - y_{a_2} - l \sin(a_2) - l \sin(b_2) \\ x_{a_1} + l \cos(a_1) + l \cos(b_1) - x_{a_3} - l \cos(a_3) - l \cos(b_3) \\ y_{a_1} + l \sin(a_1) + l \sin(b_1) - y_{a_3} - l \sin(a_3) - l \sin(b_3) \end{bmatrix} = 0 \quad (2.18)$$

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:

$$R = \begin{bmatrix} -l \sin(a_1) & l \sin(a_2) & 0 & -l \sin(b_1) & l \sin(b_2) & 0 \\ l \cos(a_1) & -l \cos(a_2) & 0 & l \cos(b_1) & -l \cos(b_2) & 0 \\ -l \sin(a_1) & 0 & -l \sin(a_3) & -l \sin(b_1) & 0 & l \sin(b_3) \\ l \cos(a_1) & 0 & -l \cos(a_3) & l \cos(b_1) & 0 & -l \cos(b_3) \end{bmatrix} \quad (2.19)$$

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 , tenemos:

$$\dot{z} = W \dot{p} \quad (2.20)$$

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.

2. MODELO MATEMÁTICO DEL ROBOT PARALELO

- $\dot{p} = [\dot{x}, \dot{y}]^T$ representa el vector velocidad del efector final.

Y la matriz Jacobiana puede definirse como:

$$W = \begin{bmatrix} r_1 \cos(\theta_{b_1}) & r_1 \sin(\theta_{b_1}) \\ r_2 \cos(\theta_{b_2}) & r_2 \sin(\theta_{b_2}) \\ r_3 \cos(\theta_{b_3}) & r_3 \sin(\theta_{b_3}) \\ -r_1 \cos(\theta_{a_1}) & -r_1 \sin(\theta_{a_1}) \\ -r_2 \cos(\theta_{a_2}) & -r_2 \sin(\theta_{a_2}) \\ r_3 \cos(\theta_{a_3}) & -r_3 \sin(\theta_{a_3}) \end{bmatrix} \text{ donde: } r_i = \frac{1}{l \sin(\theta_{b_i} - \theta_{a_i})} \quad (2.21)$$

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:

$$W^T M(z)\ddot{z} + W^T C(z, \dot{z})\dot{z} = W^T \tau - W^T f(z, \dot{z}) + W^T R^T \lambda = W^T \tau - W^T f(z, \dot{z}) \quad (2.22)$$

Derivando \dot{z} de la relación Jacobiana (2.21), tenemos:

$$\ddot{z} = \dot{W}\dot{p} + W\ddot{p} \quad (2.23)$$

donde:

$$\dot{W} = \begin{bmatrix} r_{1p}(-\dot{\theta}_{b_1} \cos \theta_{a_1} + \dot{\theta}_{a_1} c_{ba1} \cos \theta_{b_1}) & r_{1p}(-\dot{\theta}_{b_1} \sin \theta_{a_1} + \dot{\theta}_{a_1} c_{ba1} \sin \theta_{b_1}) \\ r_{2p}(-\dot{\theta}_{b_2} \cos \theta_{a_2} + \dot{\theta}_{a_2} c_{ba2} \cos \theta_{b_2}) & r_{2p}(-\dot{\theta}_{b_2} \sin \theta_{a_2} + \dot{\theta}_{a_2} c_{ba2} \sin \theta_{b_2}) \\ r_{3p}(-\dot{\theta}_{b_3} \cos \theta_{a_3} + \dot{\theta}_{a_3} c_{ba3} \cos \theta_{b_3}) & r_{3p}(-\dot{\theta}_{b_3} \sin \theta_{a_3} + \dot{\theta}_{a_3} c_{ba3} \sin \theta_{b_3}) \\ r_{1p}(-\dot{\theta}_{a_1} \cos \theta_{b_1} + \dot{\theta}_{b_1} c_{ba1} \cos \theta_{a_1}) & r_{1p}(-\dot{\theta}_{b_1} \sin \theta_{b_1} + \dot{\theta}_{b_1} c_{ba1} \sin \theta_{a_1}) \\ r_{2p}(-\dot{\theta}_{a_2} \cos \theta_{b_2} + \dot{\theta}_{b_2} c_{ba2} \cos \theta_{a_2}) & r_{2p}(-\dot{\theta}_{b_2} \sin \theta_{b_2} + \dot{\theta}_{b_2} c_{ba2} \sin \theta_{a_2}) \\ r_{3p}(-\dot{\theta}_{a_3} \cos \theta_{b_3} + \dot{\theta}_{b_3} c_{ba3} \cos \theta_{a_3}) & r_{3p}(-\dot{\theta}_{b_3} \sin \theta_{b_3} + \dot{\theta}_{b_3} c_{ba3} \sin \theta_{a_3}) \end{bmatrix} \quad (2.24)$$

- con $c_{bai} = \cos(\theta_{bi} - \theta_{ai})$.

Considerando que la fricción en las uniones pasivas es despreciable, puede simplificarse. Haciendo τ_a y f_a los torques en los actuadores y las fricciones en las uniones activas. 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.

$$S = \begin{bmatrix} r_1 \cos \theta_{b1} & r_1 \sin \theta_{b1} \\ r_2 \cos \theta_{b2} & r_2 \sin \theta_{b2} \\ r_3 \cos \theta_{b3} & r_3 \sin \theta_{b3} \end{bmatrix}$$

y sustituyendo (3.19) y (3.18) en (3.20) tenemos:

$$\widetilde{M}(z)\ddot{p} + \widetilde{C}(z, \dot{z})\dot{p} = S^T \tau - S^T f(\dot{z}) \quad (2.25)$$

donde:

- $\widetilde{M} = W^T M W$ representa la matriz de inercia del espacio de tareas.
- $\widetilde{C} = W^T (M \dot{W} + C W)$ 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 \widetilde{M} y \widetilde{C} cumplen las propiedades:

- \widetilde{M} es una matriz simétrica definida positiva.

2. MODELO MATEMÁTICO DEL ROBOT PARALELO

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

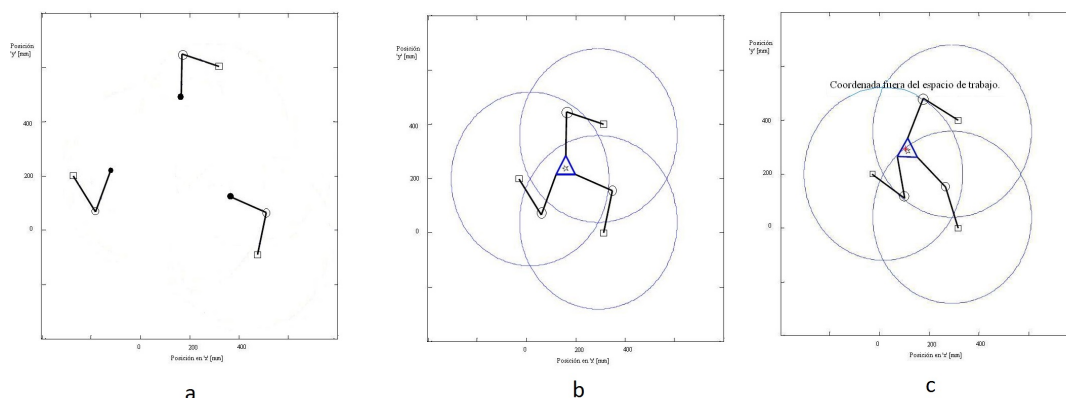


Figura 2.3: Configuraciones del robot paralelo: - a - eslabones separados, b - eslabones unidos en el efector final, c - efector final cerca de los límites del área de trabajo.

En la Figura 2.3, primero se muestran los tres brazos del robot separados, después al considerar que estos se unen se puede apreciar un ejemplo de cuando el efector final se encuentra dentro del espacio de trabajo y también cuando el efector final se acerca fuera del espacio de trabajo, que como se muestra en la Figura 2.3 el espacio está definido por la intersección de los tres círculos, círculos que describen el espacio de trabajo de cada cadena independiente si éstas no estuvieran unidas en el efector final.

2.4.1 Modelo en espacio de estados

Con el objetivo de aplicar las técnicas y análisis de control al sistema, transformaremos la expresión (2.25) a la forma de Cauchy. Considerando el cambio de variables:

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

Por tanto, el modelo queda expresado de la siguiente manera:

$$\dot{x} = A(z, \dot{z})x + B(z)u \quad (2.26)$$

$$y = (\theta a_1, \theta a_2, \theta a_3)^T \quad (2.27)$$

Donde:

- $x = (x_1, x_2, x_3, x_4)^T$ Es el vector de estados (posiciones y velocidades del efector final).
- $y = (\theta a_1, \theta a_2, \theta a_3)^T$ Es el vector de mediciones (posición angular de las uniones activas).
- $u = (u_1, u_2, u_3)$ Es el control expresado como el torque de los actuadores.

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

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & A_1[1, 1] & A_1[1, 2] \\ 0 & 0 & A_1[2, 1] & A_1[2, 2] \end{pmatrix}, A_1 = -(\tilde{M})^{-1}\tilde{C}, \quad (2.28)$$

$$B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ B_1[1, 1] & B_1[1, 2] & B_1[1, 3] \\ B_1[2, 1] & B_1[2, 2] & B_1[1, 3] \end{pmatrix}, B_1 = -(\tilde{M})^{-1}S^T. \quad (2.29)$$

2.5 Definición del error y superficie deslizando

Sea $p^d(t)$ la trayectoria deseada del efector final en el espacio de tareas, entonces el error de seguimiento se puede calcular como:

$$e = p - p_d, \quad (2.30)$$

2. MODELO MATEMÁTICO DEL ROBOT PARALELO

se puede definir una variable de deslizamiento s de la siguiente manera:

$$s = \dot{e} - Qe, \quad (2.31)$$

en donde Q es una matriz definida positiva. Si tenemos $s \equiv 0$ representa un conjunto de ecuaciones diferenciales lineales cuya única solución es $e \equiv 0$, dadas las condiciones iniciales $p^d(0) = p(0)$. Así entonces el problema de convergencia de e y \dot{e} puede reducirse a mantener el vector s en cero. Con (2.31), \dot{s} queda:

$$\dot{s} = \ddot{e} + Q\dot{e} \quad (2.32)$$

La referencia para el seguimiento de velocidad \dot{p}^r y aceleración \ddot{p}^r puede definirse así

$$\begin{aligned} \dot{p}^r &= \dot{p} + s = \dot{p}^d + Qe \\ \ddot{p}^r &= \ddot{p} + \dot{s} = \ddot{p}^d + Q\dot{e} \end{aligned}$$

2.6 Estabilidad en lazo cerrado con control PD

Esta sección está basada en [1]. Para mostrar las propiedades de la ley de control nominal (PD) del sistema en operación. Consideremos la siguiente función candidata de Lyapunov (2.33),

$$V = \frac{1}{2} \dot{z}^T M(z) \dot{z} + \frac{1}{2} \tilde{z}^T K_p \tilde{z} \quad (2.33)$$

de la cual

- $\frac{1}{2} \dot{z}^T M(z) \dot{z}$ Energía cinética del robot.
- $\frac{1}{2} \tilde{z}^T K_p \tilde{z}$ 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 $M(z) \ddot{z}$;

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

donde en la penúltima ecuación se ha utilizado la propiedad de $\dot{M}(z) - 2C(z, \dot{z})$ 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 termino gravitacional $g(z)$ o en nuestro caso, podemos añadir otro tipo de perturbaciones, usando la notación $\delta(z, \dot{z})$ (fricción seca-viscosa), tenemos

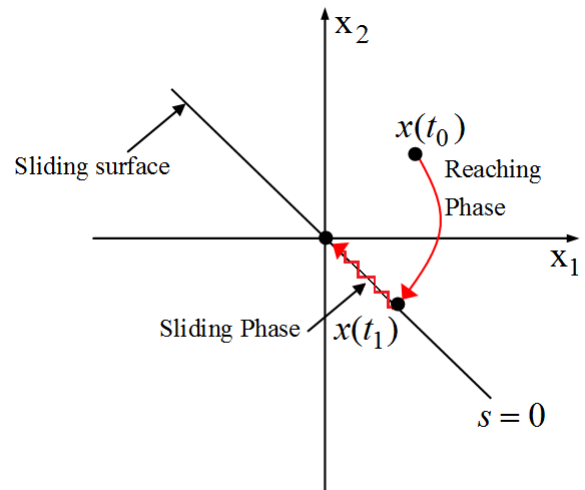
$$\dot{V} = \dot{z}^T (u - \delta(t, z, \dot{z}) - K_p \tilde{z})$$

2. MODELO MATEMÁTICO DEL ROBOT PARALELO

Algoritmos por modos deslizantes

En esta sección se presentan los algoritmos estándares por modos deslizantes. Las leyes de control correspondientes a cada algoritmo, además de las funciones de Lyapunov con las que estos cuentan y que son herramienta para estudiar bajo que condiciones cada algoritmo es convergente. En adición a los algoritmos estándares se mencionan dos nuevos recientemente desarrollados: el CID y 3-STA

(cuya definición se abundará al final del capítulo).



3.1 Sistema mecánico de segundo orden con incertidumbre acoplada

Para todos los algoritmos que se mencionarán en esta sección, se considerará como sistema a un doble integrador, con una incertidumbre acoplada al control. Es decir vamos a considerar cada brazo del robot como un doble integrador, las incertidumbres (para poder ajustar las ganancias en cada caso) será sobreestimada con ayuda de los parámetros y modelos del robot paralelo.

$$\dot{x}_1 = x_2 \quad (3.1)$$

$$\dot{x}_2 = u + \delta(x_1, x_2, t) \quad (3.2)$$

Con la consideracion siguiente (para los algoritmos Super-Twisting, 3-STA y CID):

- $\delta(x_1, x_2, t)$ es Lipschitz conociendo alguna constante tal que $\Delta > \left| \frac{d}{dt} \delta(x_1, x_2, t) \right|$.

En el caso del algoritmo de primer orden y Twisting, se hace la consideración:

- $\delta(x_1, x_2, t)$ está acotada, esto es $D > |\delta(x_1, x_2, t)|$.

Como se puede apreciar de éste sistema, el cual considera los estados como posición y velocidad, no está relacionado estrictamente con el modelo presentado del robot, sin embargo es útil ya que contamos con las transformaciones de posición cartesiana a angulares, por tanto en realidad se estarían implementando tres leyes de control, una a cada actuador. Ya que como se vio en la sección anterior, el modelo tiene un grado relativo igual a dos, el mismo que éste sistema simplificado, lo que los conecta son las perturbaciones y las señales de posición/velocidad.

3.2 Cambio de notación

En el algoritmo CID que se presentará más adelante, se usa el cambio de notación (3.3). Debido a que las expresiones se vuelven demasiado largas. Para una variable real $z \in \mathbb{R}$ elevada a una potencia $p \in \mathbb{R}$

$$[z]^p = |z|^p \text{sign}(z) \quad (3.3)$$

Para dejar más claro, aquí algunos ejemplos de la nueva notación,

$$\begin{aligned} [z]^0 &= \text{sign}(z) \\ [z]^0 z^p &= |z|^p \\ [z]^p [z]^q &= |z|^p \text{sign}(z) |z|^q \text{sign}(z) = |z|^{p+q} \end{aligned}$$

3.3 Algoritmo convencional de primer orden

El algoritmo dado por la expresión (3.4),

$$u_1 = -\rho \text{sign}(s) \quad (3.4)$$

con $\rho = D + \frac{\alpha}{\sqrt{2}}$, $D > |\delta(z, \dot{z})|$ y $\alpha > 0$ provee de convergencia en tiempo finito de las variables de deslizamiento y convergencia exponencial de los estados.

3.3.1 Condiciones de convergencia mediante función de Lyapunov

La ley de control (3.4) provee de convergencia asintótica a los estados, y de convergencia en tiempo finito a la superficie deslizante, la función candidata de Lyapunov está en la expresión (3.5), donde $s = x + qx$ es la superficie de deslizamiento.

$$V = \frac{1}{2} s^2 \quad (3.5)$$

$$\dot{V} \leq -\alpha V^{\frac{1}{2}}, \alpha > 0 \quad (3.6)$$

$$u_1 = -qx_2 - \rho \text{sign}(s) \quad (3.7)$$

$$\rho = D + \frac{\alpha}{\sqrt{2}} \quad (3.8)$$

3. ALGORITMOS POR MODOS DESLIZANTES

si se elige la ganancia (3.8) entonces la ley de control (3.7) hace $\sigma = 0$ en tiempo finito.

3.4 Modos deslizantes integrales (MDI)

La estructura del controlador es:

$$u(t) = u_0(t) + u_1(t),$$

donde :

- $u_0(t)$ es la parte nominal para la cual se garantiza estabilidad sin perturbaciones. $\delta(z, \dot{z}, t)$.
- $u_1(t)$ es el modo deslizante integral que compensa las incertidumbres desde $t = 0$.

Recordando en nuestro caso, u_0 es el PD, sin embargo podría ser un PID, o una retro de estados por LQR. El objetivo es alcanzar la dinámica ideal en lazo cerrado, a pesar de presencia de estas incertidumbres δ .

3.4.1 Caso lineal

$$\dot{x}(t) = Ax(t) + B(u_0(t) + u_1(t)) + B\delta(x, t),$$

la función σ se define como

$$\sigma(x) = G(x(t) - x(0)) - G \int_0^t (Ax(\tau) + Bu_0(\tau)) d\tau$$

La matriz G debe cumplir con la condición $\det(GB) \neq 0$

3.5 Super-Twisting

El Super-Twisting nos da una señal de control continua, requiere superficie de deslizamiento, además de las condiciones necesarias. El algoritmo provee de convergencia en tiempo finito de las superficies

y convergencia exponencial de los estados de posición y velocidad (3.1),

$$\begin{cases} u_1 = -k_1|s|^{\frac{1}{2}}\text{sign}(s) + w \\ \dot{w} = k_3\text{sign}(s) \end{cases} \quad (3.9)$$

Una posible elección de las constantes es $\Delta > |\frac{d}{dt}\delta(t)|$ y $k_1 = 1.5\sqrt{\Delta}$, $k_3 = 1.1\Delta$.

3.5.1 Condiciones necesarias para la convergencia del algoritmo Super-Twisting con función de Lyapunov

En [18] se presenta por primera vez una función de Lyapunov para modos deslizantes de segundo orden, ésta es una herramienta muy útil para estudiar condiciones y propiedades de convergencia. Dicha función se expresa en la ecuación (3.10), o en su forma cuadrática, que es $V(x) = \zeta^T P \zeta$, con $\zeta^T = [|x_1|^{\frac{1}{2}}\text{sign}(x_1)x_2]$ y P definida en (3.11).

$$V(x) = 2k_3|x_1| + \frac{1}{2}x_2^2 + \frac{1}{2}(k_1|x_1|^{\frac{1}{2}}\text{sign}(x_1) - x_2)^{\frac{1}{2}} \quad (3.10)$$

$$P = \frac{1}{2} \begin{bmatrix} 4k_3 + k_1^2 & -k_1 \\ -k_1 & 2 \end{bmatrix} \quad (3.11)$$

Si se acotan los terminos de la perturbación como en la expresión (3.12), entonces las condiciones de las desigualdades en (3.13) hacen al algoritmo (5.4) asintóticamente estable, en nuestro caso particular $|\varrho_2(t)| = |\frac{d}{dt}\delta(t)|$ y $\varrho_1 = 0$.

$$|\varrho_1| \leq \delta_1|x_1|^{\frac{1}{2}}, |\varrho_2| \leq \delta_2 \quad (3.12)$$

$$k_1 > 2\delta_1 \quad (3.13)$$

$$k_3 > k_1 \frac{5\delta_1 k_1 + 6\delta_2 + 4(\delta_1 + \delta_2/k_1)^2}{2(k_1 - 2\delta_1)} \quad (3.14)$$

3.6 Twisting

El Twisting es un algoritmo por modos deslizantes de segundo orden, este a diferencia de los dos antes mencionados, puede ser aplicado directamente sin necesidad de una superficie de deslizamiento, aunque el chattering en la señal de control aparece de nuevo

$$u_1 = -\alpha \text{sign}(e) - \beta \text{sgn}(\dot{e}) \quad (3.15)$$

con $D > |\delta(z, \dot{z})|$ y $\alpha - D > \beta > D$ provee convergencia en tiempo finito de los estados.

3.6.1 Condiciones de convergencia con función de Lyapunov

En el trabajo [16] se presenta el análisis para convergencia en tiempo finito para sistemas mecánicos de segundo orden. Con la función de Lyapunov (3.16) se puede concluir bajo el análisis presentado en el artículo que bajo las condiciones (4.7) y (4.7), con $\gamma > 0$ el algoritmo provee de convergencia en tiempo finito a los estados.

$$V(x_1, x_2) = \alpha^2 x^2 + \gamma |x_1|^{\frac{3}{2}} \text{sign}(x_1) x_2 + \alpha |x_1| x_2^2 + \frac{1}{4} x_2^4 \quad (3.16)$$

$$\gamma < \frac{4\sqrt{2}}{3} (\beta - D) \sqrt{\alpha} \quad (3.17)$$

$$\alpha - D > \beta > D \quad (3.18)$$

3.7 Control Integral Discontinuo (CID)

$$u_1 = -k_1|e_1|^{\frac{1}{3}}\text{sign}(e_1) - k_2|e_2|^{\frac{1}{2}}\text{sign}(e_2) + L + \ddot{r}(t) \quad (3.19)$$

$$\dot{L} = -k_3\text{sign}(e_1) \quad (3.20)$$

k_1, k_2, k_3 son ganancias que, diseñadas apropiadamente hacen al algoritmo con las siguientes propiedades.

1. Convergencia en tiempo finito.
2. Señal de control continua.
3. Se requiere información de los estados de posición y velocidad: x, \dot{x} .

3.7.1 Condiciones necesarias para la convergencia del algoritmo CID

El algoritmo super twisting de tercer orden cuenta con una función de Lyapunov, que para asegurar su convergencia se tienen que cumplir condiciones de positividad definida y negatividad definida para su derivada. En el trabajo de [24] y [25] presenta un método para el algoritmo, además de estas condiciones. Para positividad definida la expresión (3.21) es condición necesaria y suficiente, y para que $V(x) > 0$ cuando $\gamma_{13} > 0$ para todo $\gamma_1, \gamma_2 > 0$.

$$\begin{aligned} V(x) &= \xi^T \Gamma \xi + \gamma_{13} x_1 x_3 \\ \xi^T &= [[x_1]^2 \quad x_2 \quad [x_3]^2] \\ \Gamma &= \begin{bmatrix} \gamma_1 & 0 & 0 \\ 0 & \gamma_2 & -\frac{1}{2}\gamma_{23} \\ 0 & -\frac{1}{2}\gamma_{23} & \gamma_3 \end{bmatrix} \\ 0 &< \frac{3^3 \gamma_2 \gamma_{13}^4}{4^3 \gamma_1^3} < \gamma_2 \gamma_3 - \gamma_{23}^2 \end{aligned} \quad (3.21)$$

3. ALGORITMOS POR MODOS DESLIZANTES

Mientras que para negatividad definida, las condiciones se resumen en estas desigualdades:

$$\Delta > k_3, \quad (3.22)$$

$$0 < \phi(\alpha, \lambda) < v(\alpha), \quad (3.23)$$

$$0 < \alpha < 2 \frac{\gamma_2}{\gamma_{23}} - \frac{1}{\gamma_{23} k_2} \sqrt{\frac{32 |\gamma_2 k_1 - \frac{2}{3} \gamma_1|^3}{27 \gamma_{13} (k_3 - \Delta)}} \quad (3.24)$$

donde:

$$v(\alpha) = \frac{1}{\gamma_{23} k_2} (\gamma_{23} - 4 \gamma_3 (k_3 + \Delta) - \sqrt{\Pi}) \quad (3.25)$$

y

$$\Pi = \frac{4 \gamma_{23}^3 k_1^3 (2 \gamma_2 k_2 - \gamma_{23} k_2 \alpha)^2}{27 \gamma_{13} (k_3 - \Delta) (2 \gamma_2 k_2 - \gamma_{23} \alpha)^2 - 32 |\gamma_2 k_1 - \frac{2}{3} \gamma_1|^3}. \quad (3.26)$$

Además para todo este conjunto de parámetros existe un escalamiento para ajustar el conjunto de parámetros con respecto a una Δ conocida a partir de un caso nominal $\Delta = 0$, el escalamiento es $(k_1, k_2, k_3) \longrightarrow (l_1 k_1, l_1^{\frac{3}{4}} k_2, l_1^{\frac{3}{2}} k_3)$, $(\gamma_1, \gamma_2, \gamma_3, \gamma_{13}, \gamma_{23}) \longrightarrow (l_1^{-1} \gamma_1, l_1^{-2} \gamma_2, l_1^{-5} \gamma_3, l_1^{-2} \gamma_{13}, l_1^{-\frac{3}{2}} \gamma_{23})$.

3.7.2 Función $\psi(\alpha, \lambda)$

La función $\psi(\alpha, \lambda)$ cumple con $\psi(\alpha, \lambda) \leq \phi(\alpha)$ en las desigualdades anteriores (3.23). Para más detalles acerca de dichas desigualdades se puede consultar en [24]. La función está definida de la siguiente manera:

$$\psi(\alpha, \lambda) = \begin{cases} \max(0, \psi_1(\alpha, \lambda)) & \lambda \geq -\sqrt{3\alpha} \\ \max(0, \psi_2(\alpha, \lambda)) & \lambda < -\sqrt{3\alpha} \end{cases} \quad (3.27)$$

donde

$$\psi_1(\alpha, \lambda) = -\alpha \left(\frac{\lambda + \sqrt{|\lambda|^2 + 3\alpha}}{3\alpha} \right)^3 + \lambda \left(\frac{\lambda + \sqrt{|\lambda|^2 + 3\alpha}}{3\alpha} \right)^2 + \left(\frac{\lambda + \sqrt{|\lambda|^2 + 3\alpha}}{3\alpha} \right)$$

$$\psi_2(\alpha, \lambda) = \alpha \left(\frac{\lambda - \sqrt{|\lambda|^2 - 3\alpha}}{3\alpha} \right)^3 - \lambda \left(\frac{\lambda - \sqrt{|\lambda|^2 - 3\alpha}}{3\alpha} \right)^2 + \left(\frac{\lambda - \sqrt{|\lambda|^2 - 3\alpha}}{3\alpha} \right)$$

de la cual se sabe que λ puede tomar dos valores diferentes, esto es

$$\lambda_1 = 2\gamma_2 + \gamma_{13} + 2\gamma_{23}(k_3 + \Delta)$$

$$\lambda_2 = 2\gamma_2 + \gamma_{13} - 2\gamma_{23}(k_3 + \Delta)$$

3.8 Super-Twisting Tercer Orden (3-STA)

Este nuevo algoritmo es un trabajo conjunto, entre los autores de México e India [26]. Aún antes de ser publicado el trabajo referente a éste artículo se trabajó a la par de su desarrollo. Las propiedades mencionadas en la primera versión en principio se conservan, si se diseñan apropiadamente las ganancias k_1, k_2, k_3

$$u_1 = -k_1|\phi|^{\frac{1}{2}}\text{sign}(\phi) + L \quad (3.28)$$

$$\dot{L} = -k_3\text{sign}(\phi) \quad (3.29)$$

$$\phi = \dot{e} + k_2|e|^{\frac{2}{3}}\text{sign}(e) \quad (3.30)$$

3.8.1 Condiciones necesarias para la convergencia del algoritmo 3-STA

Al igual que el anterior algoritmo, este también cuenta con una función de Lyapunov, de la cual podemos resumir las siguientes condiciones para que este sea convergente. Para el caso perturbado:

3. ALGORITMOS POR MODOS DESLIZANTES

$$\dot{\delta} = \rho, \quad |\rho| \leq \Delta \neq 0$$

y con el escalamiento $l_2 > 0$

$$(k_1, k_2, k_3) \longrightarrow (l_2^3 k_1, l_2^2 k_2, l_2^6 k_3)$$

Además del juego de desigualdades,

$$\begin{aligned} p_1 + p_2 k_2^2 &> k_2 p_{12} \\ p_{12} &= 2p_2 k_2 \\ p_{12} &> 2p_{13} k_3 \\ 2p_2 &> 2p_{23} k_3 \\ k_3 &> 0 \end{aligned}$$

Desigualdades satisfechas por $\alpha_1, \alpha_2, \vartheta(\alpha_1), \vartheta(\alpha_2)$, esto es:

$$\begin{aligned} q_1 k_1^2 p_{12} - k_1 p_{12} - \sqrt{\frac{2^2 q_4^3}{3^2 (p_{23} - q_6)}} &> \alpha_1 > 0 \\ \vartheta &\geq \beta(\lambda, \alpha_1) \\ \frac{2k_1^2 p_2 p_{23} - \alpha_2}{k_1^3 p_{23} p_{12}} &> \nu(\alpha_1) > 0 \end{aligned}$$

y también,

$$\begin{aligned} 2k_1^2 p_2 p_{23} &> \alpha_2 > 0 \\ \vartheta(\alpha_2) &\geq \max\{\beta(\lambda_1, \alpha_2), \beta(\lambda_2, \alpha_2)\} \\ \frac{1}{(k_1 p_{12})^2} (p_{23} - |q_6| - \frac{2^2 |q_4|^3}{3^3 (q_1 k_1 - \frac{\alpha_1}{k_1 p_{12}})^2}) &> \vartheta(\alpha_2) > 0 \end{aligned}$$

3.9 Diferenciador robusto exacto

Una alternativa para obtener la señal de velocidad requerida para los algoritmos, es el uso del diferenciador de segundo orden.

$$\dot{\hat{x}}_1 = -\lambda_3 L_d^{\frac{1}{3}} |\hat{x}_1 - x_1|^{\frac{2}{3}} \text{sign}(\hat{x}_1 - x_1) + \hat{x}_2 \quad (3.31)$$

$$\dot{\hat{x}}_2 = -\lambda_2 L_d^{\frac{1}{2}} |\hat{x}_2 - \dot{\hat{x}}_1|^{\frac{1}{2}} \text{sign}(\hat{x}_2 - \dot{\hat{x}}_1) + \hat{x}_3 \quad (3.32)$$

$$\dot{\hat{x}}_3 = -\lambda_1 L_d \text{sign}(x_3 - \dot{\hat{x}}_2). \quad (3.33)$$

Donde x_1 , x_2 y x_3 son los estados que corresponden a (3.1). Eligiendo las apropiadas ganancias λ_i [19] (Levant). Las ganancias $L_d > \ddot{z}_i$ se eligen de manera distinta para cada motor, ya que son tres las señales de posición a diferenciar. Este diferenciador nos ofrece una señal continua de velocidad.

3. ALGORITMOS POR MODOS DESLIZANTES

Ajuste de ganancias para controladores

4.1 Seguimiento en espacio de uniones

Utilizando la cinemática inversa proporcionada por el manual del equipo, también mencionada en el Capítulo 2 de éste trabajo. Es posible obtener mediante transformaciones los valores de las uniones activas $(\theta_{a1}, \theta_{a2}, \theta_{a3})$, a partir de la trayectoria deseada (p_x, p_y) . En éste Capítulo a menudo se harán referencia a gráficas con las variables θ_{ai} , y definiendo $\theta_{di}(t)$ como el equivalente en el espacio de las uniones de $p_d(t)$. Se presentan las siguientes ecuaciones:

$$e_{ai} = \theta_{ai} - \theta_{di}(t) \quad (4.1)$$

$$\dot{e}_{ai} = \dot{\theta}_{ai} - \dot{\theta}_{di}(t) \quad (4.2)$$

4.2 Superficie deslizante

4.2.1 Superficie estandar

Usualmente se diseña de la siguiente manera, retomando (2.31), eligiendo a $Q = I$, la dinámica deseada es

$$s = \dot{e} - Qe = \begin{bmatrix} \dot{e}_{px} + e_{px} \\ \dot{e}_{py} + e_{py} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.3)$$

. De esta manera las soluciones para (4.3) son

$$e_{px}(t) = p_x(0)\exp(-t), \quad \dot{e}_{px}(t) = -p_x(0)\exp(-t) \quad (4.4)$$

$$e_{py}(t) = p_y(0)\exp(-t), \quad \dot{e}_{py}(t) = -p_y(0)\exp(-t) \quad (4.5)$$

4.2.2 Superficie para MDI

En el caso de usar modos deslizantes integrales, acorde a nuestra estructura de control sumando términos con el control lineal, recordando la superficie a elegir es:

$$\sigma(x) = G(x(t) - x(0)) - G \int_0^t (Ax(\tau) + Bu_0(\tau))d\tau$$

una manera de elegir la matriz G , para cumplir con $\det(GB) \neq 0$ es la siguiente:

$$G = B^{-1}(z, \dot{z})$$

y con esto:

$$GB = I_{2 \times 2}$$

Regresando a la superficie, la dinámica es:

$$\dot{\sigma} = G[Ax + Bu + \delta - Ax - Bu_0]$$

$$\dot{\sigma} = G[Ax + B(u_0 + u_1 + \delta) - Ax - Bu_0] = GB[u_1 + \delta]$$

y puesto que elegimos $GB = I_{2 \times 2}$ entonces para que la superficie se mantenga en el origen, queda el término por modos deslizantes agregado u_1 , y las perturbaciones o incertidumbres no modeladas.

$$\dot{\sigma} = u_1 + \delta(z, \dot{z}).$$

Finalmente la superficie en términos del error queda como en la ecuación (4.6).

$$\sigma = G[e(t) - e(t_0) - \int_{t_0}^t (Ae + Bu_0)d\tau] \quad (4.6)$$

en la última expresión, es claro ver que en la utilización de Modos Deslizantes Integrales, los cuales al parecer se adaptan correctamente a nuestro esquema, no resultan tan convenientes ya que en términos de implementación resulta poco práctico requerir del conocimiento de las matrices A y B del sistema (junto con todos sus parámetros). Por esta razón no se presentan los resultados obtenidos con ésta técnica, en simulación no resulta problema, pero el llevarlo a la implementación sí lo es.

4.3 Cota de la perturbación

Para los casos del Algoritmo Convencional de primer orden, Twisting y el HOSM, es suficiente con conocer la cota de la dinámica no conocida o perturbación. En este caso podemos considerar tanto fricción seca como fricción viscosa. Los modelos de estas perturbaciones, presentados en el Capítulo 2, presentan sobrepasos en el sistema nominal, sin embargo tomamos la fase después de ese fenómeno. Como se puede observar en la gráfica de la Figura 4.1 y cuyos valores para las fronteras están en la Tabla 4.1.

$$f_{ai} = \text{sign}(\dot{\theta}_{ai})f_{ci} + f_{vi}\dot{\theta}_{ai} \quad (4.7)$$

4.4 Cota de la derivada de perturbación

Para el caso de los algoritmos Súper Twisting, CID y 3-STA, es necesario conocer la cota de la derivada de la perturbación, es decir, qué tan rápido cambia o varía la fricción. En el caso de la derivada de la fricción seca no se puede hacer nada ya que su modelo implica una función signo, la cuál teóricamente tiene un cambio infinitamente

4. AJUSTE DE GANANCIAS PARA CONTROLADORES

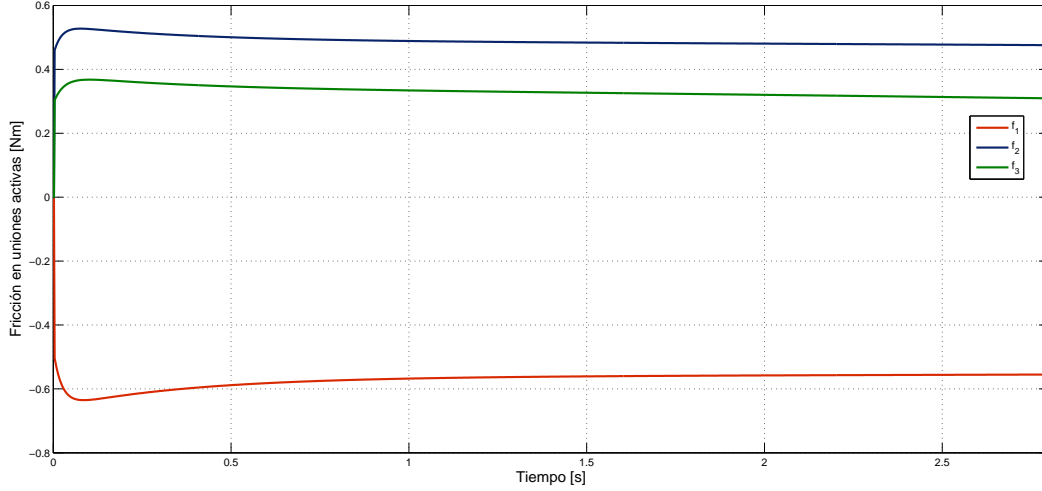


Figura 4.1: Fricción en uniones activas - Se muestran en esta grafica obtenida por simulación la fricción

rápido cuando la velocidad cruza por cero. Tomamos la ecuación de la fricción viscosa, en las uniones activas considerandola como perturbación. Para acotar la derivada de la fricción viscosa, consideramos (4.8), y su derivada en (4.9).

$$\delta_i = f_{vi}\dot{\theta}_{ai} \quad (4.8)$$

$$\dot{\delta}_i = f_{vi}\ddot{\theta}_{ai} \quad (4.9)$$

Entonces requerimos conocer la cota de la ecuación (4.9), considerando que el coeficiente f_{vi} no varía mucho, la cota queda como en (4.11).

$$|\dot{\delta}_i| = |f_{vi}\ddot{\theta}_{ai}| \quad (4.10)$$

$$|\dot{\delta}_i| = f_{vi}|\ddot{\theta}_{ai}| \quad (4.11)$$

Para obtener el término $|\ddot{\theta}_{ai}|$ regresamos a la ecuación (2.5) del modelo matemático, ya que $\ddot{z} = (\ddot{\theta}_{a1}, \ddot{\theta}_{a2}, \ddot{\theta}_{a3}, \ddot{\theta}_{a4}, \ddot{\theta}_{a5}, \ddot{\theta}_{a6})$, para obtener después $\ddot{\theta}_{ai}$ que es el término que nos interesa. Despejando \ddot{z} ten-

emos la ecuación (4.12)

$$\ddot{z} = M_i^{-1}[\tau_i - C_i \dot{z}_i - f_i] \quad (4.12)$$

Podemos obtener una expresión analítica si calculamos el término M^{-1} , quedando expresada en (4.13).

$$M^{-1} = \frac{1}{\alpha_i \beta_i + \gamma_i^2 \cos^2(ab_i)} \begin{pmatrix} \beta_i & -\gamma_i \cos ab_i \\ \gamma_i \cos ab_i & \alpha_i \end{pmatrix} \quad (4.13)$$

Sustituyendo (4.13) en (4.12) tenemos la expresión (4.14)

$$\ddot{z} = \frac{1}{\alpha_i \beta_i + \gamma_i^2 \cos^2(ab_i)} \begin{bmatrix} \beta_i & -\gamma_i \cos ab_i \\ \gamma_i \cos ab_i & \alpha_i \end{bmatrix} \left(\begin{bmatrix} \tau_{ai} \\ \tau_{bi} \end{bmatrix} - \begin{bmatrix} 0 & (\gamma_i \sin a_i - b_i) \dot{b}_i \\ (\gamma_i \sin a_i - b_i) \dot{b}_i & 0 \end{bmatrix} \begin{bmatrix} \dot{z}_{ai} \\ \dot{z}_{bi} \end{bmatrix} - \begin{bmatrix} f_{ai} \\ f_{bi} \end{bmatrix} \right) \quad (4.14)$$

desarrollando algunos terminos,

$$\ddot{z} = \frac{1}{\alpha_i \beta_i + \gamma_i^2 \cos^2(ab_i)} \left(\begin{bmatrix} \beta_i \tau_{ai} - \gamma_i \cos(ab_i) \tau_{bi} \\ \gamma_i \cos(ab_i) \tau_{ai} + \alpha_i \tau_{bi} \end{bmatrix} - \begin{bmatrix} -\gamma_i^2 \cos(ab_i) \sin(ab_i) \dot{\theta}_{ai} & \beta_i (\gamma_i \sin(ab_i)) \dot{\theta}_{bi} \\ -\alpha_i (\gamma_i \sin(ab_i)) \dot{\theta}_{ai} & \gamma_i^2 \cos(ab_i) \sin(ab_i) \dot{\theta}_{bi} \end{bmatrix} \begin{bmatrix} \dot{z}_{ai} \\ \dot{z}_{bi} \end{bmatrix} - \begin{bmatrix} \beta_i f_{ai} - \gamma_i \cos(ab_i) f_{bi} \\ \gamma_i \cos(ab_i) f_{ai} + \alpha_i f_{bi} \end{bmatrix} \right) \quad (4.15)$$

$$\ddot{z} = \frac{1}{\alpha_i \beta_i + \gamma_i^2 \cos^2(ab_i)} \left[\begin{aligned} &\beta_i \tau_{ai} - \gamma_i c_{abi} \tau_{bi} + \gamma_i^2 c_{abi} s_{abi} \dot{\theta}_{ai}^2 - \beta_i (\gamma_i s_{abi}) \dot{\theta}_{bi}^2 - \beta_i f_{ai} + \gamma_i c_{abi} f_{bi} \\ &\gamma_i c_{abi} \tau_{ai} + \alpha_i \tau_{bi} + \alpha_i (\gamma_i s_{abi}) \dot{\theta}_{ai}^2 - \gamma_i^2 c_{abi} s_{abi} \dot{\theta}_{bi}^2 - \gamma_i c_{abi} f_{ai} - \alpha_i f_{bi} \end{aligned} \right]. \quad (4.16)$$

Interesa sólo el término $\ddot{\theta}_{ai}$ para acotar la derivada de la fricción viscosa. Además en nuestro sistema las uniones pasivas no están actuadas, esto es $\tau_{bi} = 0$, consideramos también que la fricción en las uniones pasivas es despreciable, f_{bi} , con lo anterior esta manera, sólo tomamos el primer renglón de la expresión (4.16), y con las consideraciones antes mencionadas queda expresada en (4.17).

$$\ddot{\theta}_{ai} = \frac{1}{\alpha_i \beta_i + \gamma_i^2 \cos^2(ab_i)} (\beta_i \tau_{ai} + \gamma_i^2 \cos(ab_i) \sin(ab_i) \dot{\theta}_{ai}^2 - \beta_i (\gamma_i \sin(ab_i)) \dot{\theta}_{bi}^2 - \beta_i f_{ai}) \quad (4.17)$$

consideramos conocidos los parámetros α_i , β_i , además los términos $\dot{\theta}_{ai}^2$ se obtienen diferenciando y las velocidades de las uniones pasivas θ_{bi} se pueden obtener mediante las funciones de cinemática mencionadas en el modelo. Ahora podemos buscar una expresión analítica que nos asegure que existe una frontera para la cota de la derivada.

$$|\ddot{\theta}_{ai}| = \left| \frac{1}{\alpha_i \beta_i + \gamma_i^2 \cos^2(ab_i)} \right| \left| (\beta_i \tau_{ai} + \gamma_i^2 \cos(ab_i) \sin(ab_i) \dot{\theta}_{ai}^2 - \beta_i (\gamma_i \sin(ab_i)) \dot{\theta}_{bi}^2 - \beta_i f_{ai}) \right| \quad (4.18)$$

$$|\ddot{\theta}_{ai}| \leq \left| \frac{1}{\alpha_i \beta_i + \gamma_i^2 \cos^2(ab_i)} \right| \left(|\beta_i \tau_{ai}| + |\gamma_i^2 \cos(ab_i) \sin(ab_i) \dot{\theta}_{ai}^2| + |\beta_i (\gamma_i \sin(ab_i)) \dot{\theta}_{bi}^2| + |\beta_i f_{ai}| \right) \quad (4.19)$$

$$|\ddot{\theta}_{ai}| \leq \left| \frac{1}{\alpha_i \beta_i + \gamma_i^2 \cos^2(ab_i)} \right| (\beta_i |\tau_{ai}| + \gamma_i^2 \dot{\theta}_{ai}^2 + \beta_i \gamma_i \dot{\theta}_{bi}^2 + \beta_i f_{ai}) \quad (4.20)$$

Podemos conocer esa cota calculandola, con las simulaciones que se han desarrollado en matlab, utilizando el modelo matemático. El valor numérico para el experimento considerado toma valores aproximadamente entre: $|\dot{\delta}_i| = f_v |\ddot{\theta}_i| \approx 0.01 \sim 0.02$. Sólo que al inicio se presenta un fenómeno de sobrepaso muy grande, como 10 veces más del valor que se ha mencionado, sin embargo por las propiedades del control nominal PD este sobrepaso es compensado rápidamente.

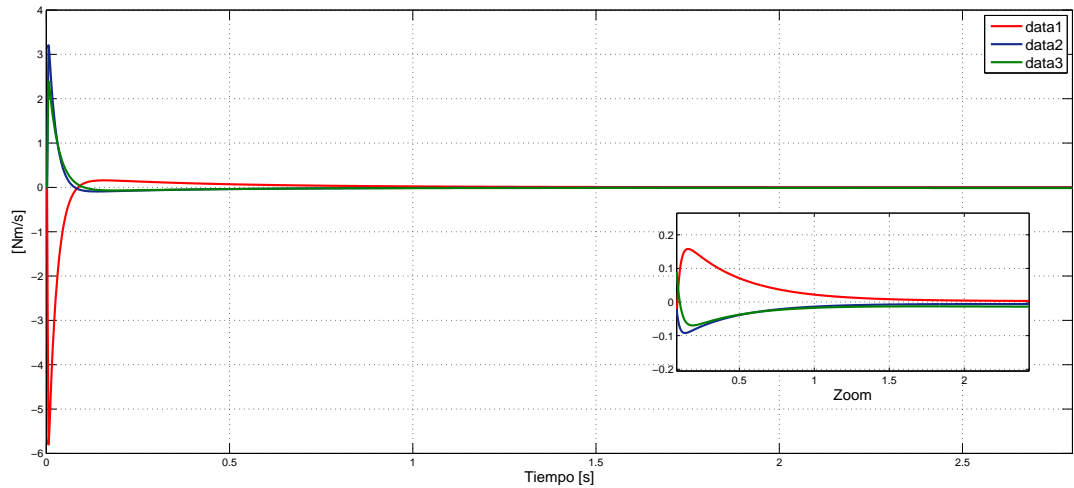


Figura 4.2: Derivada de la perturbación - Gráfica obtenida de simulación, calculando la derivada de la fricción

4.5 Valores de las constantes obtenidas

D_1	D_2	D_3	Δ_1	Δ_2	Δ_3
0.65	0.6	0.4	0.2	0.15	0.1

Tabla 4.1: Cotas consideradas, calculadas a partir de simulación.

De las constantes mostradas en la tabla 4.1, se toma $D_{max} = D$ y $\Delta_{max} = \Delta$ (la mayor) como referencia para satisfacer las condiciones que deben cumplir los controladores para que estos sean convergentes. Lo anterior es por que se desea hacer un seguimiento hacia un punto o dirección arbitraria dentro del espacio de trabajo del robot, esto implica que los motores no siempre harán un esfuerzo mínimo. Para el algoritmo de primer orden se ha elegido como ganancia $\rho = 0.8$, para el algoritmo Twisting las ganancias que satisfacen las condiciones de (4.7) se eligieron como $\alpha = 1.45$, $\beta = 0.7$. Por otro lado las ganancias para el super twisting se pueden calcular a partir de la cota de la derivada de la perturbación, en el capítulo anterior las expresiones se encuentran en función de un término que se toma como $\Delta = 0.1$, eligiendo este valor es posible asegurar que el algoritmo es convergente.

D_1	D_2	D_3	Δ_1	Δ_2	Δ_3
0.65	0.6	0.4	0.2	0.15	0.1

Tabla 4.2: Constantes de los controladores.

4.6 Ajustes Super-Twisting

En un primer caso se hacen ajustes considerando estas expresiones:

$$k_1 = 1.5\sqrt{\Delta_i}$$

y

$$k_2 = 1.1|\Delta_i|$$

Es de mencionar que se pueden ajustar las ganancias considerando las condiciones a partir función de Lyapunov, que aunque son muy parecidas también tienen cierta relación. Pero no se hace por que no fue necesario, con ésta elección de ganancias es suficiente para que al menos en este sistema bajo las condiciones consideradas, el algoritmo funcione.

4.7 Ajuste Twisting

Recordando las condiciones y de la sección anterior de donde:

$$\alpha - D_i > \beta > D_i$$

Y por ejemplo, para un caso particular D_i , (0.65, 0.6, 0.4), se puede verificar que ganancias que satisfacen las condiciones de (4.7) pueden ser $\alpha = 1.45$, $\beta = 0.7$, y con esto queda

$$\gamma < \frac{4\sqrt{2}}{3}(\beta - D_i)\sqrt{\alpha} < 0.1135$$

4.8 Ajuste para 3-STA y CID

El primer valor que hay que ajustar es k_3 , ya que debe cumplir $k_3 > \Delta$ se puede elegir $k_3 = 0.15$, después se deben de sintonizar k_2 y k_3 incrementándolas de tal manera que el sistema tenga una respuesta deseada. Una manera rápida de hacer esto es utilizar parámetros que satisfacen el caso nominal ($\Delta = 0$), y a partir de éste hacer un ajuste de L para una Δ ya conocida. La manera de sencilla de comprobar este diseño se es graficamente. Las condiciones son satisfechas si la función Ψ pasa por debajo de la función v . La comprobación grafica sirve para no tener que buscar todos los parámetros que cumplan con las desigualdades, basta con reescalar los ya existentes, en ambos casos los parámetros y ganancias. Este escalamiento, mencionado en la Sección 3.5.1 (l_1 para versión 1 y l_2 para versión 2).

4. AJUSTE DE GANANCIAS PARA CONTROLADORES

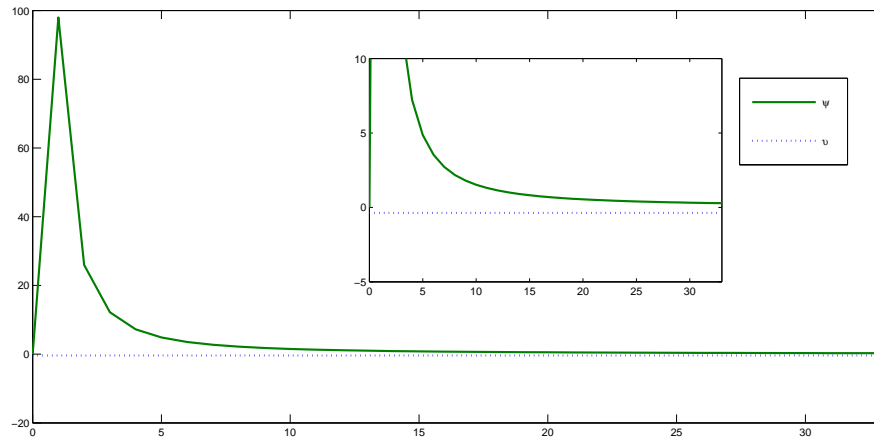


Figura 4.3: Función Ψ 1 - No siempre cumple la desigualdad, esta grafica es un ejemplo cuando la función Ψ no pasa por debajo de ν .

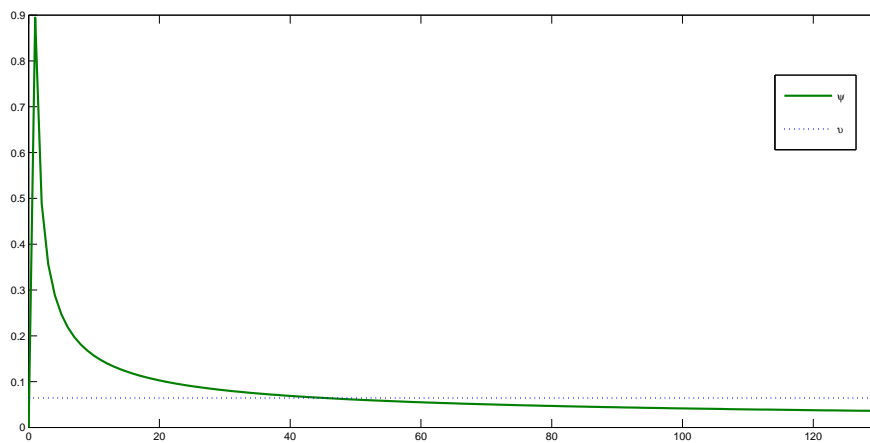


Figura 4.4: Función Ψ 2 - En esta gráfica la desigualdad es satisfecha.

4.9 Diferenciador con el experimento nominal

Antes de enviar las señales de velocidad a los algoritmos discontinuos, es necesario asegurarse de que el diferenciador funcione correctamente. En la Figura 4.5 podemos ver las posiciones de los tres motores que serán diferenciadas, se trata de un experimento en el que el efector final del robot sigue una línea recta y después regresa a la posición inicial. En la Figura 4.6 se muestran los resultados de aplicar el diferenciador de segundo orden, aquí lo importante es que se obtiene una señal continua, lo cuál es condición necesaria en todos los algoritmos.

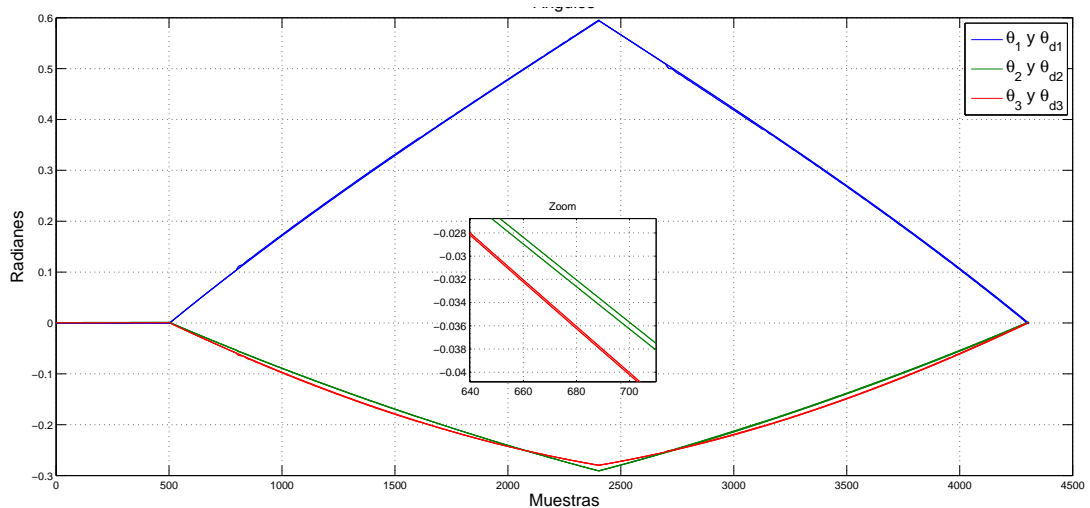


Figura 4.5: Posiciones angulares deseadas y de los actuadores - Experimento de seguimiento de línea recta, con control nominal

4. AJUSTE DE GANANCIAS PARA CONTROLADORES

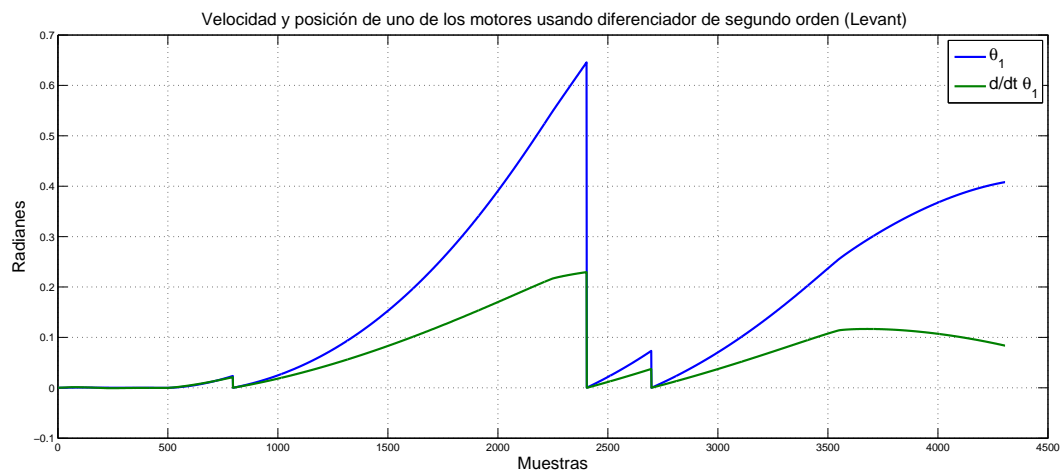


Figura 4.6: Resultados de diferenciador - Se muestran los resultados del diferenciador aplicado a un sólo motor, se puede apreciar su derivada continua

Resultados de simulación

En este capítulo nos enfocamos en los resultados obtenidos por simulación numérica. Se simuló la dinámica completa del sistema descrito en el Capítulo 2 de este trabajo. El entorno de simulación usado es MATLAB simulink. Este entorno cuenta con una amplia gama de funciones y librerías que facilitan la reproducción numérica de nuestro robot paralelo. El diagrama de bloques utilizado se muestra en la Figura 5.1.

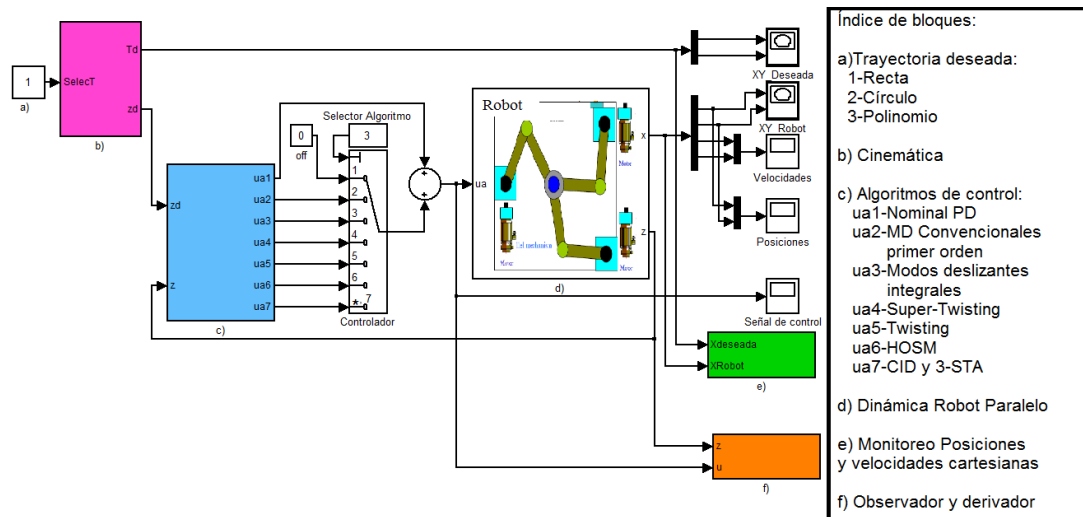


Figura 5.1: Diagrama Simulink - Estos son los bloques que reproducen en simulación numérica a los experimentos reales.

5. RESULTADOS DE SIMULACIÓN

5.1 Experimento a realizar

Los experimentos tanto en simulaciones como en implementación consisten en el seguimiento de curvas paramétricas. En esta sección se muestran sólo los resultados de implementar el seguimiento de una trayectoria recta a velocidad constante. Lo ideal es que el efector final del robot la siga sin desviarse demasiado del error de posición y velocidad. La curva generada para el caso de la línea recta se muestra en la Figura 5.2.

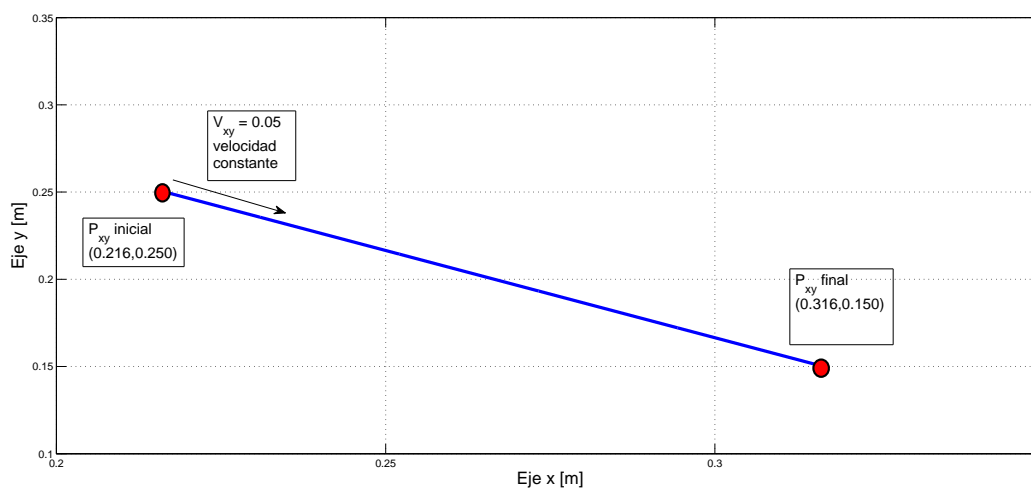


Figura 5.2: Trayectoria recta - Se muestra la trayectoria recta que debe seguir el robot a una velocidad constante, la velocidad de cada eje es calculada a partir de la magnitud deseada, con ayuda de relaciones trigonométricas

5.2 Parámetros usados

En esta sección se presentan los parámetros usados tanto para las simulaciones, en la Tabla 5.1 se encuentran los valores de la dinámica del robot,

De la Tabla 5.1 se tiene,

- r_i es la distancia entre el centro de masa y la unión.

	masa [kg]	longitud de eslabones [m]	r_i [m]	J_i [kg.m ²]
L_{a1}	1.2525	0.2440	0.1156	0.0124
L_{a2}	1.3663	0.2440	0.0067	0.0122
L_{a3}	1.3663	0.2440	0.0067	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 del robot paralelo - Estos son los parámetros que fueron introducidos a la simulación, corresponden a los valores reales.

- J_i es el momento de inercia relativo al centro de masa.

Los parametros de simulación fueron los mismos para todos los algoritmos, son los siguientes:

- **Tiempo de simulación:** Variable dependiendo la trayectoria a seguir (entre 2 y 4 segundos).
- **Opciones de solución:** Fixed-step (Paso fijo)
- **Método de solución:** ode1 (Euler)
- **Paso de muestreo:** 0.0002 [s] (igual que en la tarjeta PCI).

5.3 Modos deslizantes convencionales

En la Figura 5.3 se aprecian los resultados de haber probado, bajo las condiciones apropiadas, el algoritmo de primero orden propuesto. Aunque al principio las constantes se tomaron con base en el valor numérico calculado, al final se hicieron algunos ajustes para obtener el resultado deseado, la diferencia no fue mucha, pero esto se debe a las dinámicas no modeladas y a la inexactitud del control nominal. En la simulación se aprecia el chattering del control, este fenómeno de la conmutación del actuador a altas frecuencias no es desable en implementación real. También se pueden apreciar los resultados de convergencia en tiempo finito a la superficie deslizante.

5. RESULTADOS DE SIMULACIÓN

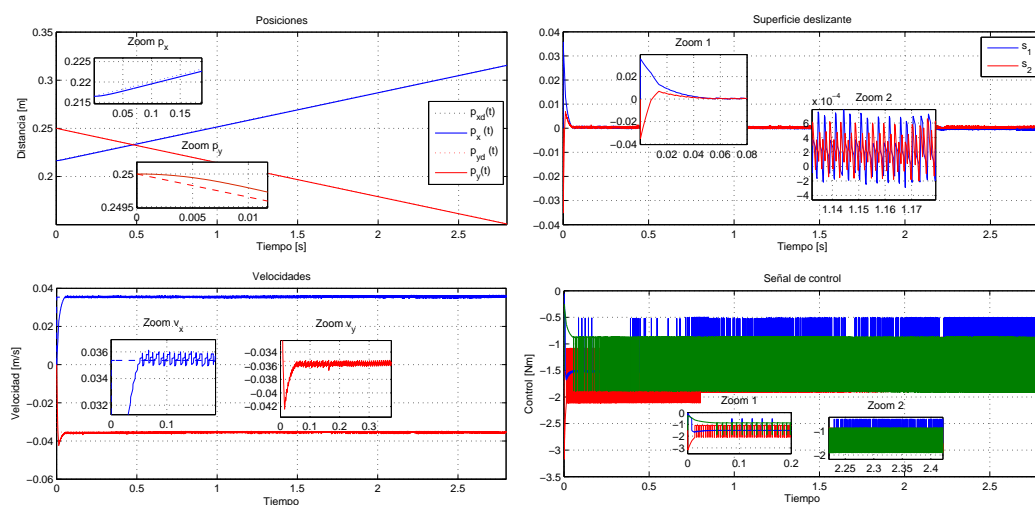


Figura 5.3: Modos deslizantes convencionales Simulación - En esta Figura se muestran los resultados de diseño por simulación de los modos deslizantes convencionales sobre el robot paralelo

5.4 Super-Twisting

En la Figura 5.4 se aprecian los resultados obtenidos de probar numéricamente el algoritmo Super-Twisting, como era de esperarse, se pueden apreciar convergencia asintótica de los estados. Una señal de control continua, gracias al integrador. Pero también se tiene una convergencia en tiempo finito de la superficie de deslizamiento.

5.5 Twisting

Para el caso del algoritmo twisting, se pueden apreciar en estos resultados de simulación, presentados en la Figura 5.5 dos aspectos importantes: Convergencia en tiempo finito de los estados y señal de control discontinua. No es necesaria la utilización de una superficie de deslizamiento como en los casos anteriores, pero el costo, muy visible en las graficas presentadas, es la aparición de nuevo del chattering.

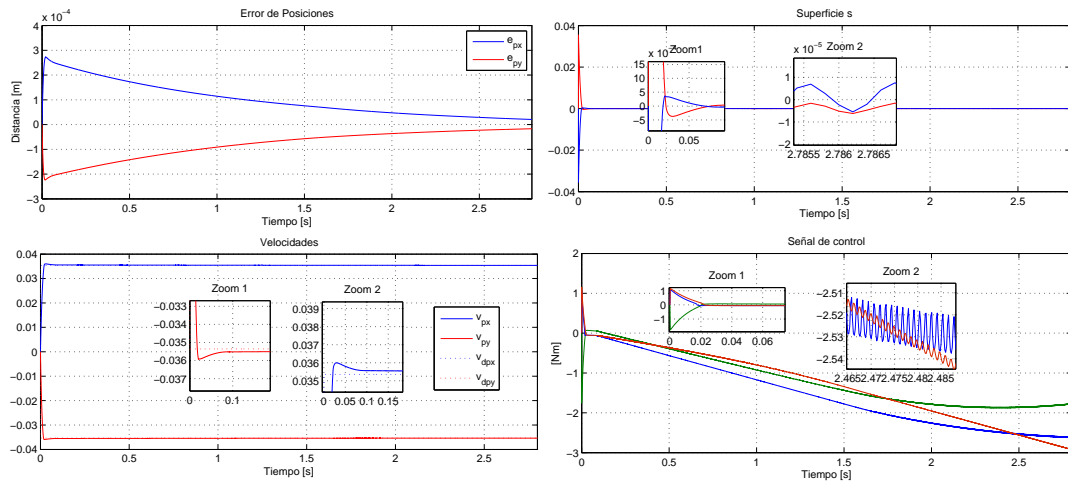


Figura 5.4: Super Twisting Simulación - En esta Figura se muestran los resultados de diseño por simulación del super twisting sobre el robot paralelo

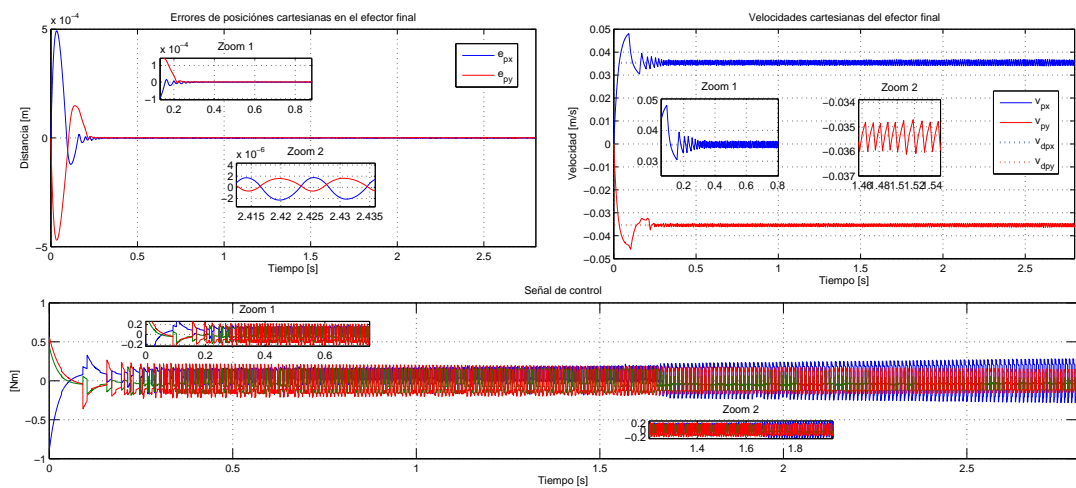


Figura 5.5: Twisting Simulación - En esta Figura se muestran los resultados de diseño por simulación del twisting sobre el robot paralelo

5. RESULTADOS DE SIMULACIÓN

5.6 Control Integral Discontinuo (CID)

Para este algoritmo, como se puede apreciar en la Figura 5.6 se tienen resultados interesantes. Resultados tales como: Convergencia en tiempo finito de los estados, sin la utilización de una superficie de deslizamiento, señal de control absolutamente continua. Además de todo esto sólomente se está utilizando información de la posición y la velocidad, igual que el control nominal.

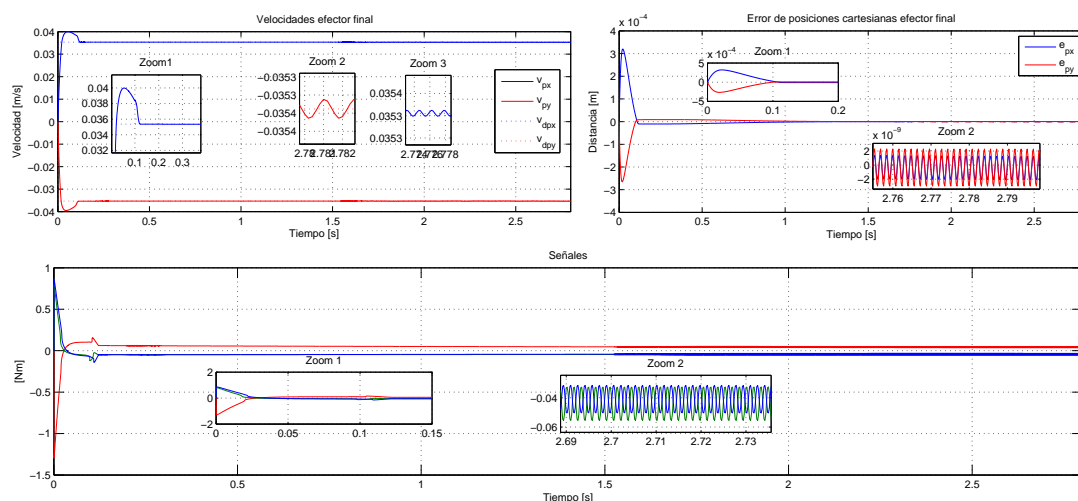


Figura 5.6: CID Simulación - Se muestran los resultados de diseño por simulación de CID sobre el robot paralelo

5.7 Super-Twisting Tercer Orden (3-STA)

Para este algoritmo, como se puede apreciar en la Figura 5.7 se tienen resultados interesantes. Resultados tales como: Convergencia en tiempo finito de los estados, sin la utilización de una superficie de deslizamiento, señal de control absolutamente continua. Además de todo esto sólomente se está utilizando información de la posición y la velocidad, igual que el control nominal.

5.7 Super-Twisting Tercer Orden (3-STA)

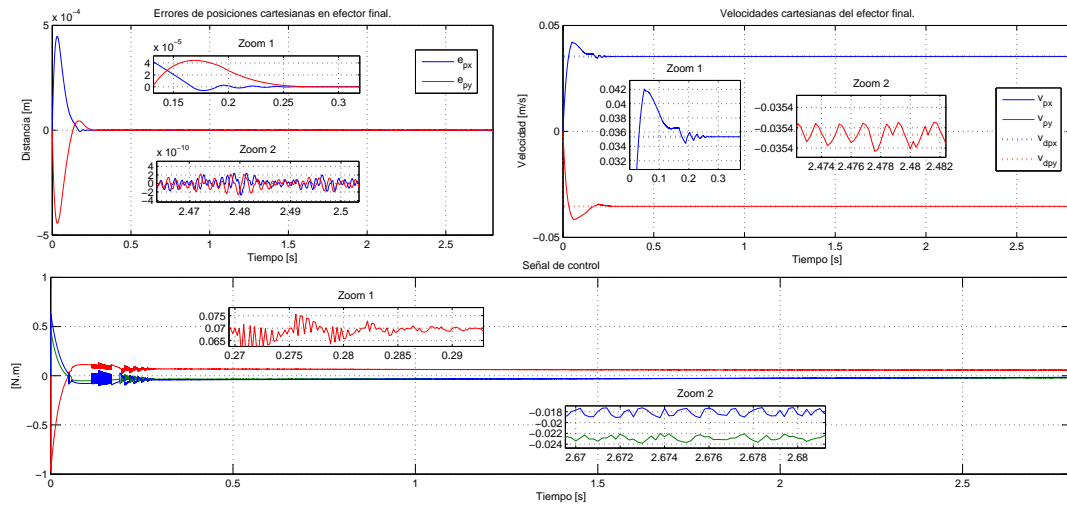


Figura 5.7: 3-STA Simulación - Se muestran los resultados de diseño por simulación del Super-Twisting tercer orden (versión 2) sobre el robot paralelo

5. RESULTADOS DE SIMULACIÓN

Experimentos

En esta sección se presentan los resultados experimentales con el equipo de prueba, donde existen factores no considerados que hacen al modelo más incierto. Factores como el tiempo de muestreo, ruido de los sensores, tiempo respuesta de los actuadores. Se muestran los resultados primero sin diferenciador y después con el uso de diferenciador. En la Figura 6.1 se muestra una fotografía del laboratorio de modos deslizantes (Posgrado Facultad de Ingeniería UNAM) en dónde se desarrollan las pruebas, se puede apreciar tanto la PC, equipo de amplificadores (debajo de la mesa), y arriba los mecanismos del robot paralelo.

6.1 Sistema de prueba GPM 2002

En la Figura 6.2 se presenta un diagrama a bloques del sistema completo de experimentación. La maquinaria del robot paralelo es de la empresa Googol Tech, modelo GPM-2002, la tarjeta de adquisición de datos GT-400-SV es de la misma empresa. La etapa de potencia, los servoamplificadores PY2 además de los servomotores son de la empresa Sanyo Denki Co. En los apéndices se mencionan algunos detalles acerca de cada bloque, detalles como algunos ajustes que fueron necesarios hacer para una correcta aplicación experimental.

6. EXPERIMENTOS

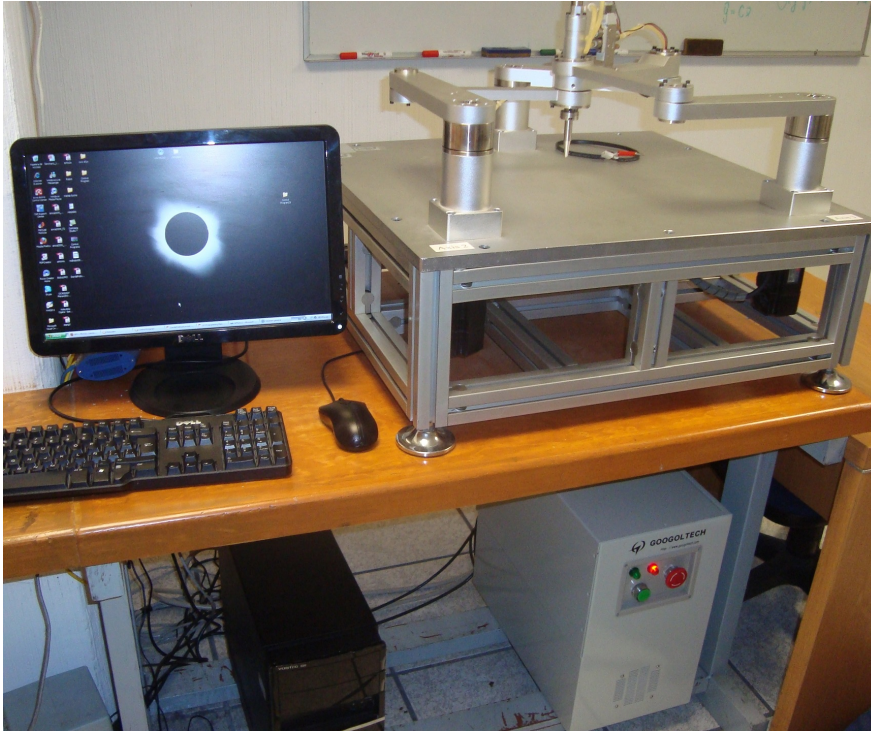


Figura 6.1: Sistema experimental Robot Paralelo - Fotografía del laboratorio de modos deslizantes (Posgrado Facultad de Ingeniería UNAM) en dónde se desarrollan las pruebas.

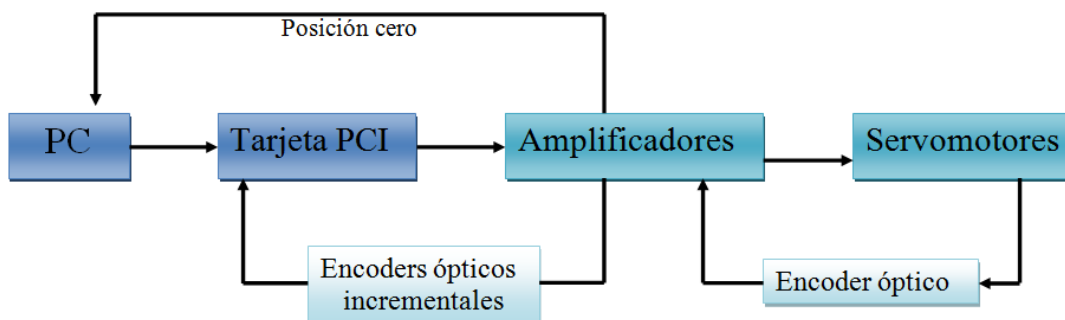


Figura 6.2: Sistema experimental diagrama a bloques 1 - Esquema con la tarjeta, los amplificadores y servomotores.

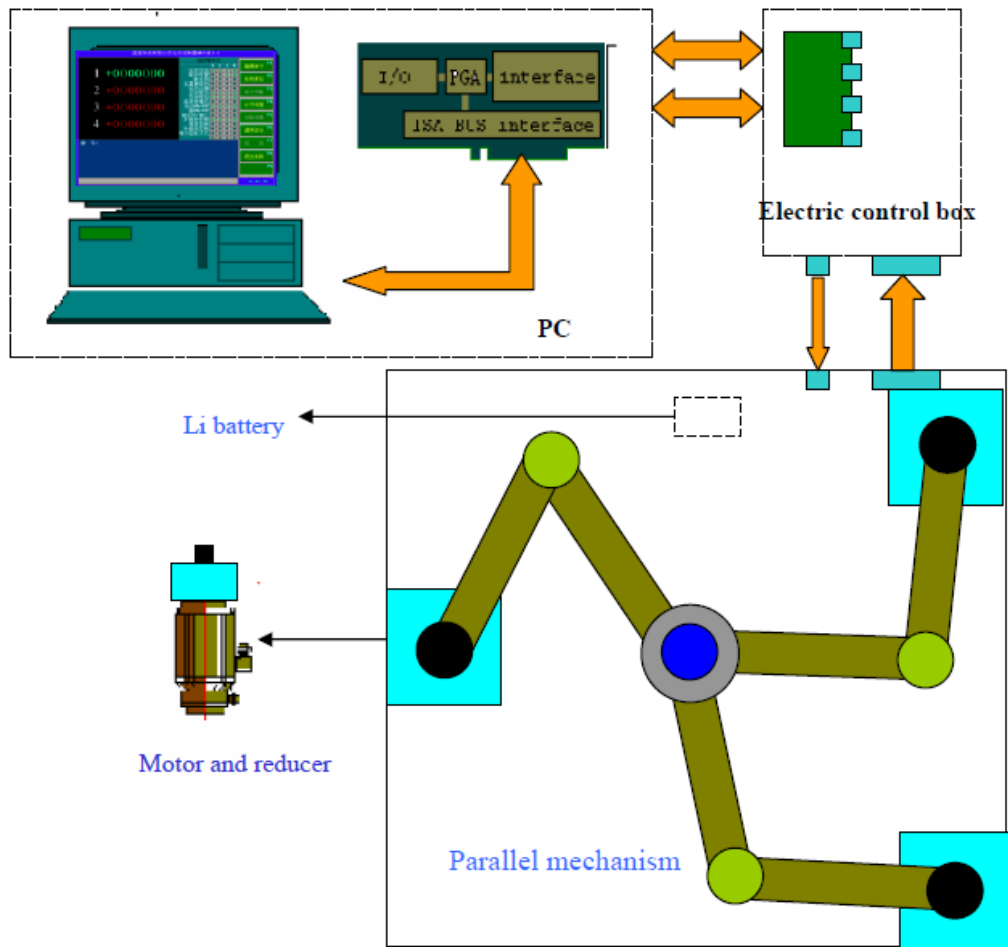


Figura 6.3: Sistema experimental diagrama a bloques 2 - En este diagrama a bloques se muestran todos los elementos que conforman el prototipo de prueba.

6.2 Experimento nominal PD

En este primer experimento, se muestran los resultados que se obtienen de las lecturas de los encoders. El experimento consiste en el seguimiento de una línea recta, para el caso del efector final (igual que como se describió al inicio del Capítulo anterior de simulaciones), sin embargo hay que mencionar que ésta trayectoria es generada en términos de posiciones cartesianas, y después obtenemos las transformaciones equivalentes para las posiciones angulares de motores, dichas posiciones son las que se muestran en la gráfica puesto que ahí se encuentran los sensores de posición. En la gráfica de la Figura 6.4 es visible como se generan errores de posición durante el experimento, aunque también es notable que las lecturas de los errores no inician en cero, esto se atribuye a que la calibración al iniciar el experimento no siempre es exacta. A partir de esta sección en adelante, se mostrará lo que sucede al añadir un término de corrección al nominal. Es de destacar que éste algoritmo nominal PD **no** se procesa en el entorno C++ en el que trabajamos para los demás algoritmos, si no que se encuentra por default en la tarjeta PCI, lo cuál lo hace trabajar a diferente tiempo de muestreo que el que se trabaja en windows. Las velocidades para los experimentos sin diferenciados de segundo orden, se obtienen diferenciando el valor actual menos el valor en el instante anterior, una manera sencilla al igual que como viene implementada en la tarjeta PCI, se puede ver el diagrama al final en los apéndices.

6.3 Algoritmo de primer orden

El valor máximo que toman los errores de posición son $[0.2948, 0.0167, 0.0887][rad]$ desde que inicia el experimento hasta que termina, mientras que toma los valores de $[0.0783, 0.0167, 0.0059][rad]$ una vez que las trayectorias se mantienen en el modo deslizante. En la Figura 6.5 se muestran las lecturas de errores de posición angular de los motores,

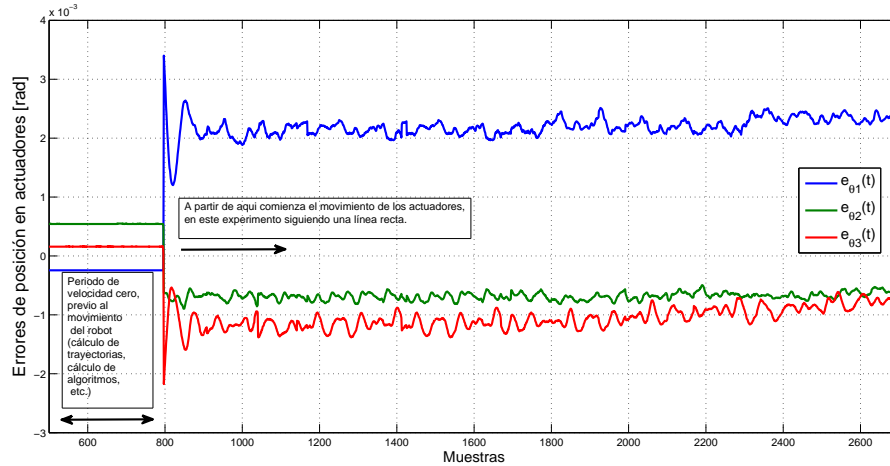


Figura 6.4: Salida medida, experimento nominal PD - Se muestran las lecturas de errores de posición angular de los motores durante seguimiento de una línea recta.

durante seguimiento de una línea recta, es visible cómo el algoritmo no logra estabilizar de manera adecuada para nuestro propósito de control.

6.4 Super-Twisting

En este experimento, cuyos resultados mostrados en la Figura 6.7, se muestran las lecturas de errores de posición angular de los motores durante seguimiento de una línea recta. Para éste experimento es claro como mejora notablemente el desempeño del algoritmo para la tarea de control. Y además en la Figura 6.8, por fin obtenemos una señal de control continua, aprecian las señales de control adicionadas al nominal calculadas por el algoritmo Super-Twisting. El valor máximo que toman los errores de posición son $[0.3285, 0.0167, 0.0751][rad]$ desde que inicia el experimento hasta que termina, mientras que toma los valores de $[0.0382, 0.0054, 0.0076][rad]$ cuando las trayectorias se mantienen en el modo deslizante.

6. EXPERIMENTOS

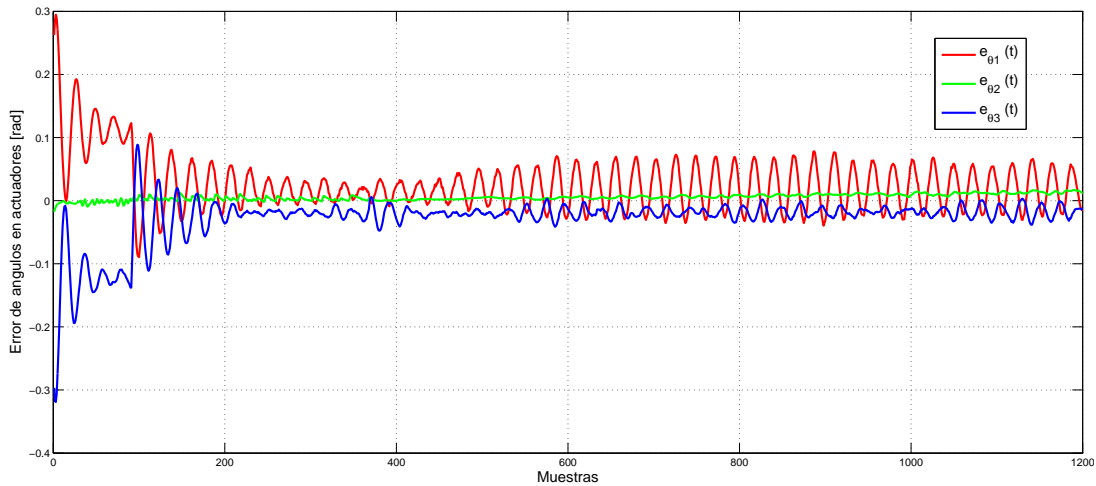


Figura 6.5: Salida medida, experimento algoritmo de primer orden - Se muestran las lecturas de errores de posición angular de los motores durante seguimiento de una línea recta.

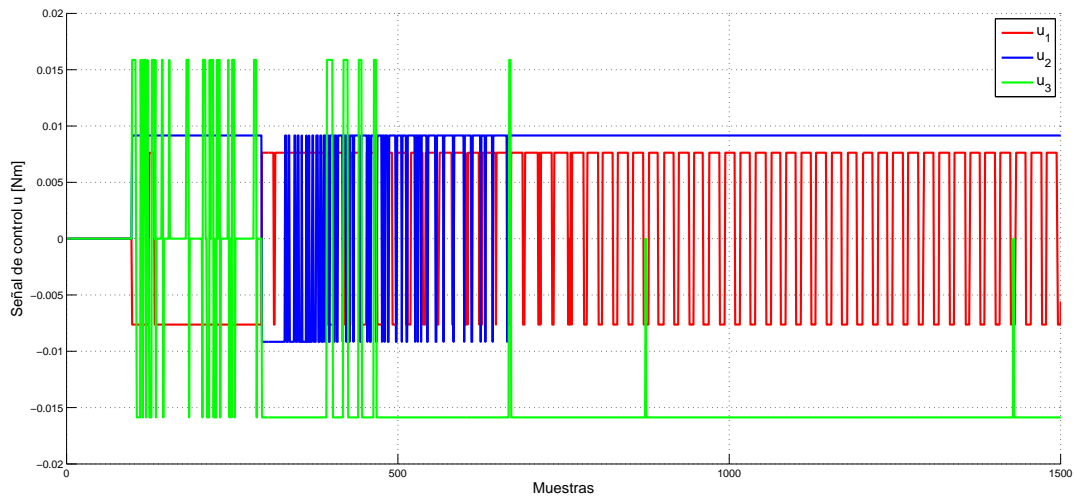


Figura 6.6: Señal de control experimento algoritmo primer orden - Se muestran señales de control adicionales al nominal calculadas por el algoritmo por modos deslizantes convencionales de primer orden, sin el uso diferenciador de segundo orden.

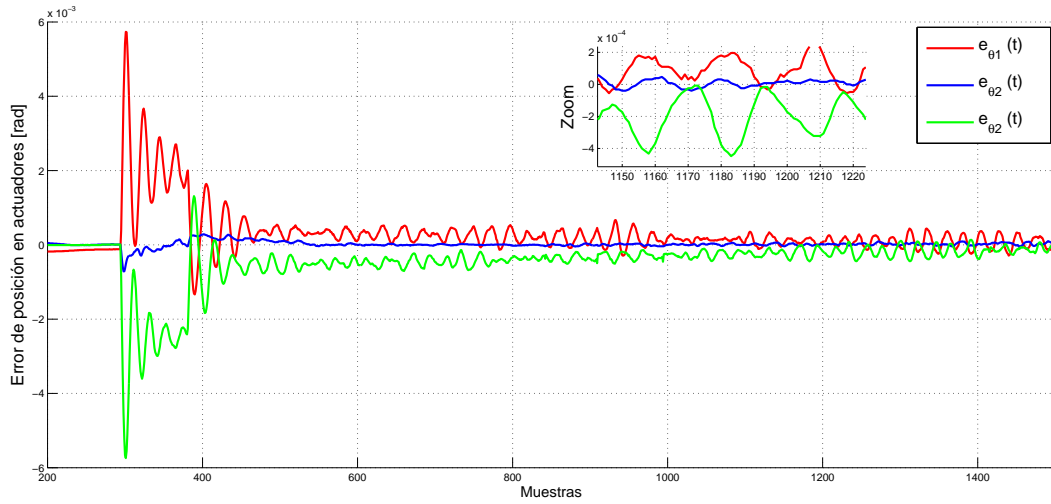


Figura 6.7: Salida medida, experimento algoritmo super-twisting - Se muestran las lecturas de errores de posición angular de los motores durante seguimiento de una línea recta, sin el uso de diferenciador de segundo orden.

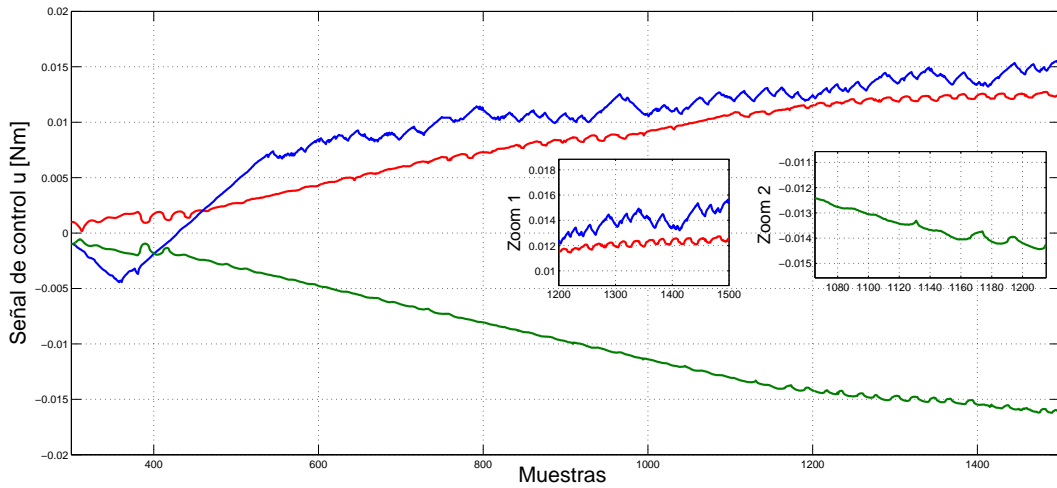


Figura 6.8: Señal de control experimento super-twisting - Se muestran señales de control adicionadas al nominal calculadas por el algoritmo por el algoritmo Super-Twisting, sin el uso de diferenciador de segundo orden.

6. EXPERIMENTOS

6.5 Resultados de 3-STA

Para estos experimentos se utilizó el método de integración del rectángulo. Ya que el principio de funcionamiento de las trayectorias comandadas al robot consisten en un ciclo 'for', los resultados de la gráfica en la Figura 6.9 de la cual se observan las lecturas de errores de posición angular de los motores durante seguimiento de una línea recta. Se aprecia como el error es atenuado, por otro lado en la Figura 6.10 se aprecian las inyecciones de control, las conmutaciones debidas a la función discontinua son filtradas por el sistema mecánico. El valor máximo que toman los errores de posición son $[0.2403, 0.1196, 0.0274][rad]$ desde que inicia el experimento hasta que termina, mientras que toma los valores de $[0.0121, 0.02225, 0.0144][rad]$ una vez que las trayectorias se mantienen en el modo deslizante.

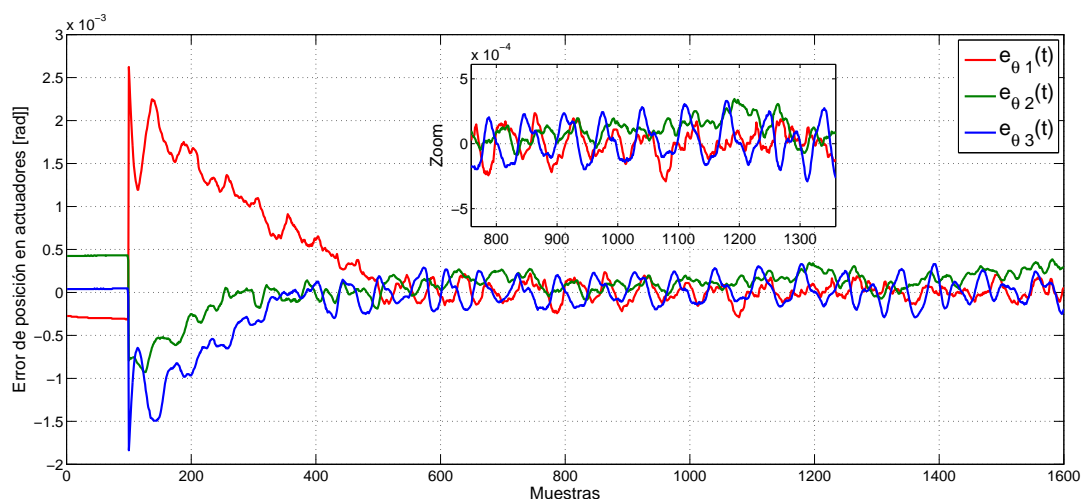


Figura 6.9: Salida medida experimento 3-STA - Se muestran las lecturas de errores de posición angular de los motores durante seguimiento de una línea recta, sin el uso de diferenciador de segundo orden.

6.6 Resultados de Control Integral Discontinuo (CID)

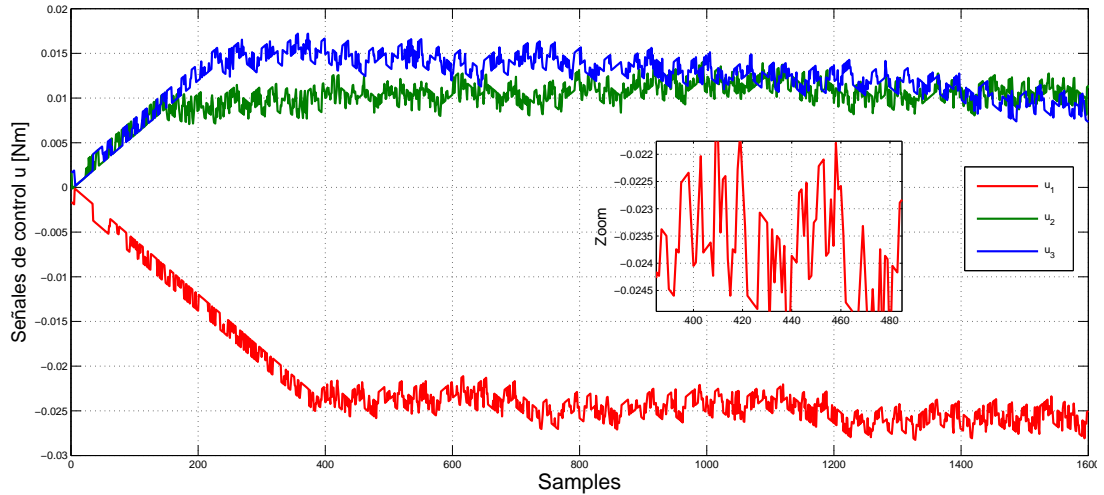


Figura 6.10: Señal de control 3-STA - Se muestran señales de control adicionadas al nominal calculadas por el algoritmo Super-Twisting de tercer orden, sin el uso de diferenciador de segundo orden.

6.6 Resultados de Control Integral Discontinuo (CID)

De igual manera que el algoritmo anterior método de integración del rectángulo. Ya que ambos algoritmos requieren la utilización de la integral. Los resultados de la gráfica en la Figura 6.11 en la cual de nuevo se muestran las lecturas de errores de posición angular de los motores. Dichas gráficas permiten apreciar como el error es atenuado, por otro lado en la Figura 6.12 se aprecian las inyecciones de control, en este caso, las rampas como dientes de sierra corresponderían a la integral de la función signo, este es el “control absolutamente continuo”. El valor máximo que toman los errores de posición son $[0.1503, 0.0796, 0.0374][rad]$ desde que inicia el experimento hasta que termina, mientras que toma los valores de $[0.0135, 0.0248, 0.0216][rad]$ una vez que las trayectorias se mantienen en el modo deslizante.

6. EXPERIMENTOS

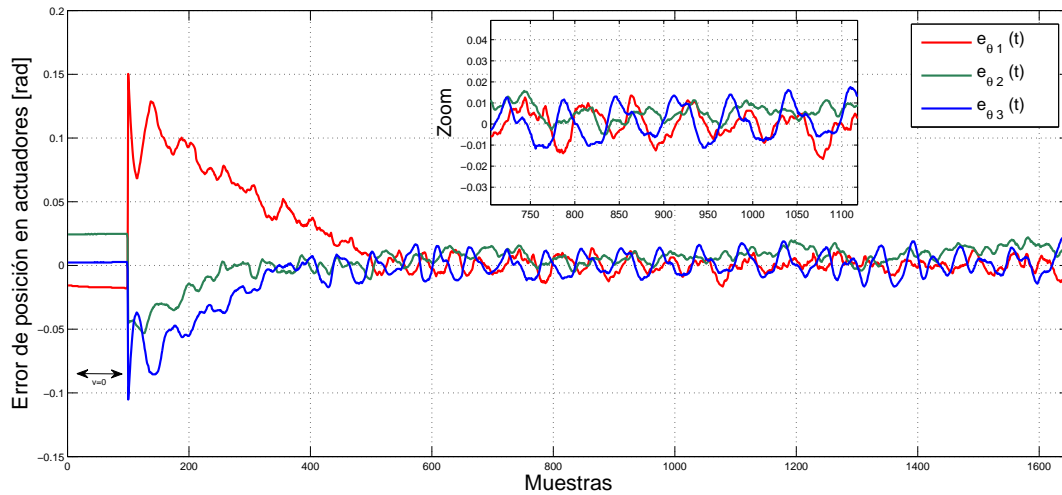


Figura 6.11: Salida medida experimento CID - Se muestran las lecturas de errores de posición angular de los motores, sin el uso de diferenciador de segundo orden.

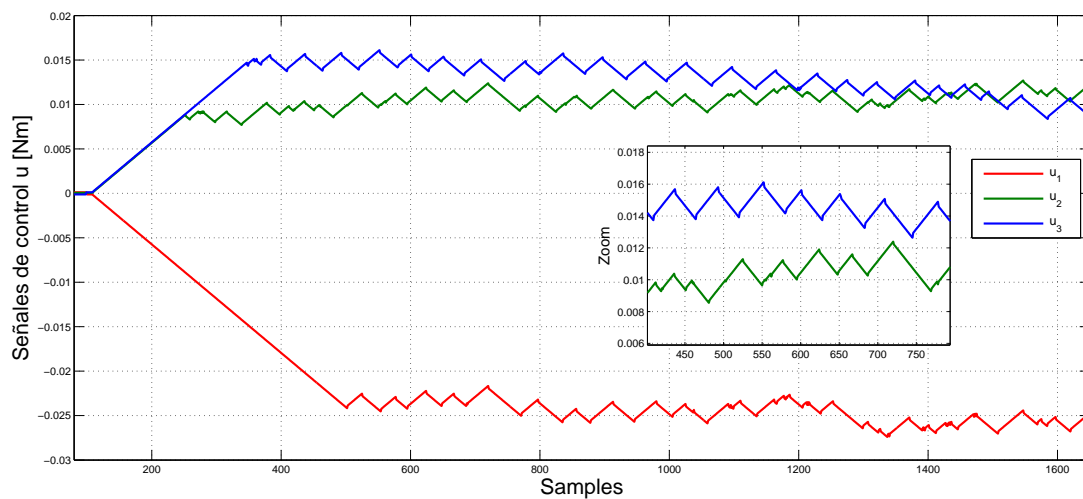


Figura 6.12: Señal de control CID - Se muestran señales de control adicionadas al nominal, sin el uso de diferenciador de segundo orden.

6.6 Resultados de Control Integral Discontinuo (CID)

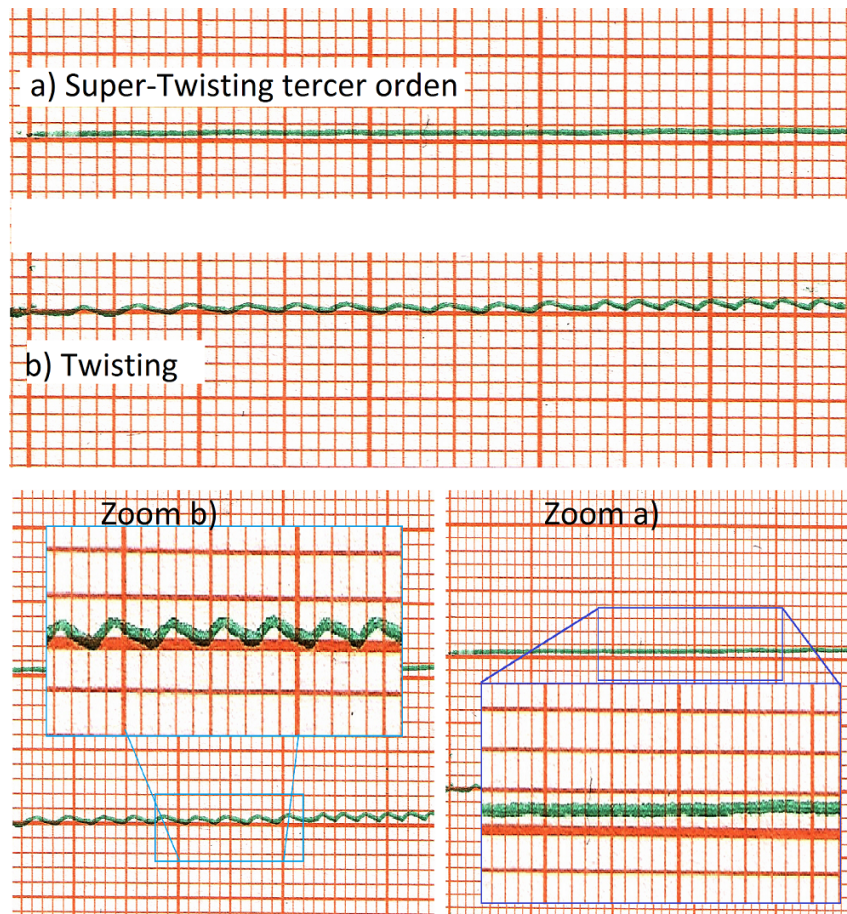


Figura 6.13: Experimentos en papel - Seguimiento de trayectorias rectas, trazos del efector final con algoritmo 3-STA (continuo) y discontinuo (Twisting).

6. EXPERIMENTOS

6.7 Experimentos con Diferenciador de Segundo Orden

En estas dos subsecciones, se muestran los resultados al combinar la utilización de un diferenciador de segundo orden. Se mostró anteriormente este diferenciador nos ofrece una señal continua de la velocidad, utilizando sólo la posición como información.

6.7.1 3-STA y diferenciador

El uso del diferenciador tuvo efectos notables sobre todo en este algoritmo. Debido a que éste contiene la señal de velocidad dentro de la discontinuidad, gracias al diferenciador robusto por modos deslizantes ahora es posible contar con una señal más confiable. Al ser un diferenciador de segundo orden, la señal de velocidad que necesitan los controladores es absolutamente continua. En la gráfica de la Figura 6.14 se puede observar como la señal tiene menos ruido antes de llegar a alguna banda y empezar a conmutar. Cambios también son notables en la señal de control, como se puede ver en la Figura 6.15, pero esos cambios serán analizados y cuantificados con más detalle en la siguiente sección.

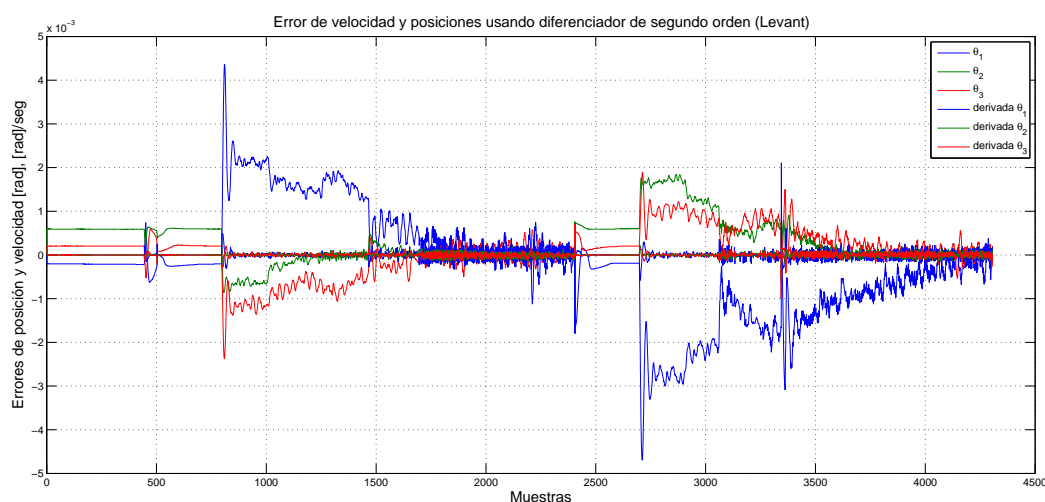


Figura 6.14: 3-STA y diferenciador de segundo orden - Se muestran las lecturas de errores de posición angular de los motores, con diferenciador de segundo orden.

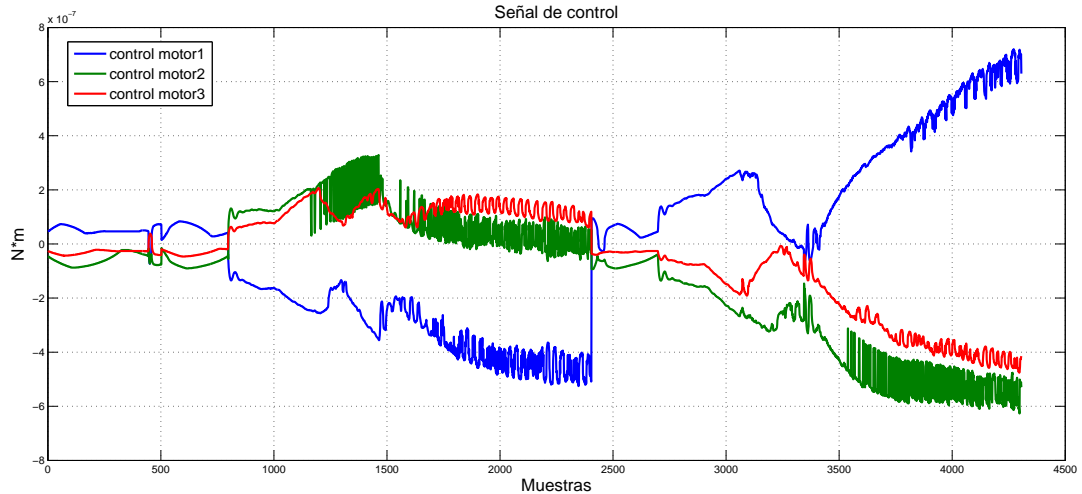


Figura 6.15: Control 3-STA con diferenciador segundo orden - Se muestran señales de control adicionadas al nominal, utilizando diferenciador de segundo orden.

6.7.2 Control integral discontinuo y diferenciador

Es notable, sobre todo en las señales de control, cambian a trayectorias más suaves al principio. Lo anterior se puede apreciar en la Figura 6.15, para el caso del control 3-STA con diferenciador segundo orden y también en la Figura 6.17 que corresponde al control CID con diferenciador segundo orden. En ambos casos para las señales de control, ya alcanzado el modo deslizante las conmutaciones causadas por la función *sign* (que es integrada), son más rápidas. Se puede apreciar que el comportamiento en las lecturas de las señales de encoders, correspondientes a los errores de posición, no hay una notable mejoría, como se puede ver en las Figuras 6.14 y 6.16, sin embargo éste comentario se analizará con más detalle en el siguiente Capítulo.

6.8 Múltiples líneas

Para finalizar se presentan los resultados tras implementar consecutivamente diferentes trayectorias. Se puede observar la precisión lograda totalmente libre de *chattering* en la Figura 6.18. Y las lec-

6. EXPERIMENTOS

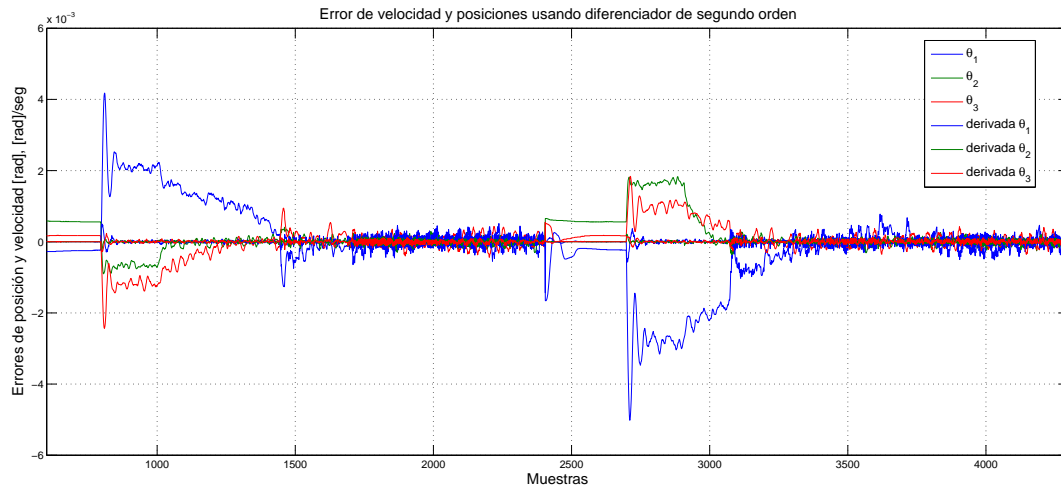


Figura 6.16: CID y diferenciador de segundo orden - Se muestran las lecturas de errores de posición angular de los motores, con diferenciador de segundo orden.

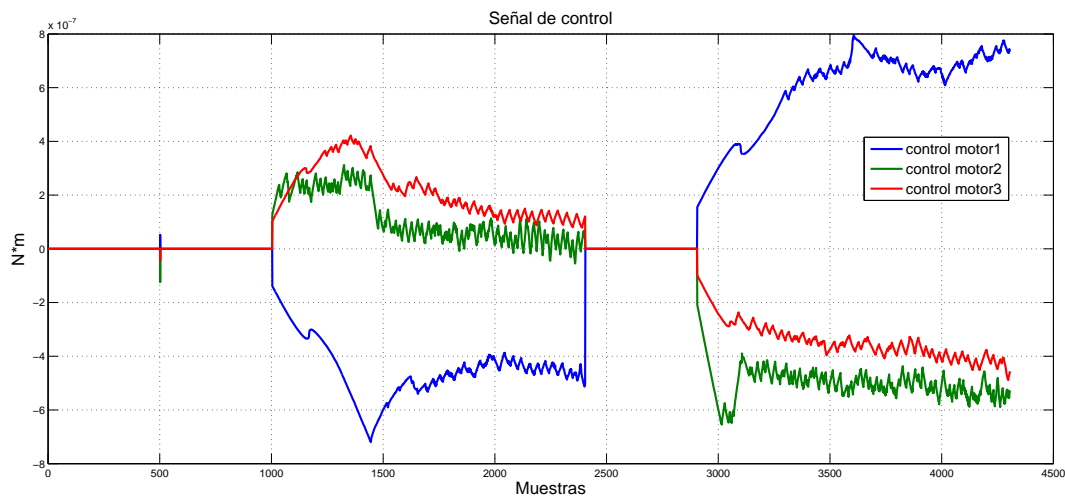


Figura 6.17: Control CID con diferenciador segundo orden - Se muestran señales de control adicionadas al nominal, con diferenciador de segundo orden.

turas de los encoders en cada motor (errores de posición), se pueden ver en las Figuras 6.19 y 6.20.

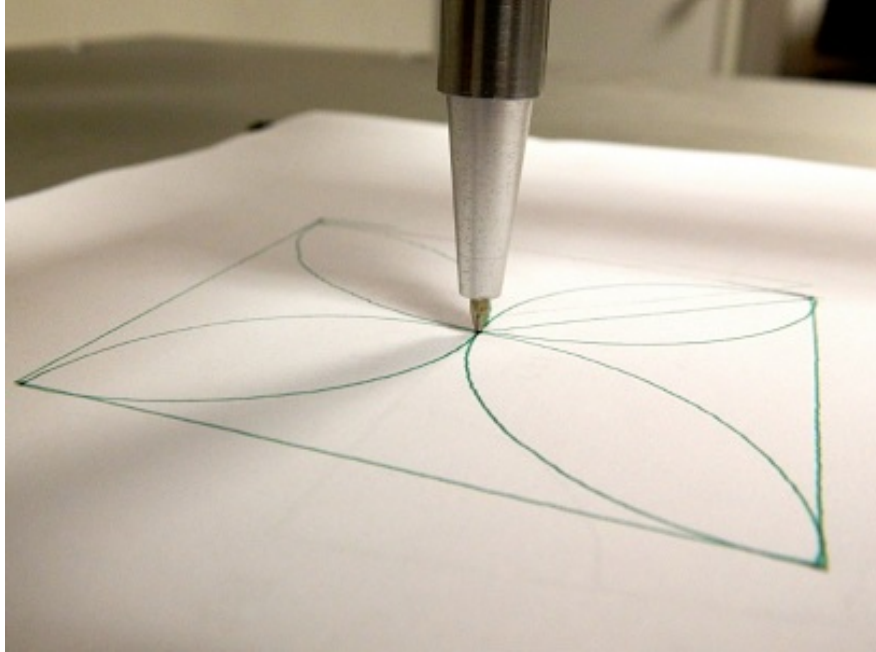


Figura 6.18: Experimentos con trayectorias consecutivas - Seguimiento de trayectorias rectas y circulares.

6. EXPERIMENTOS

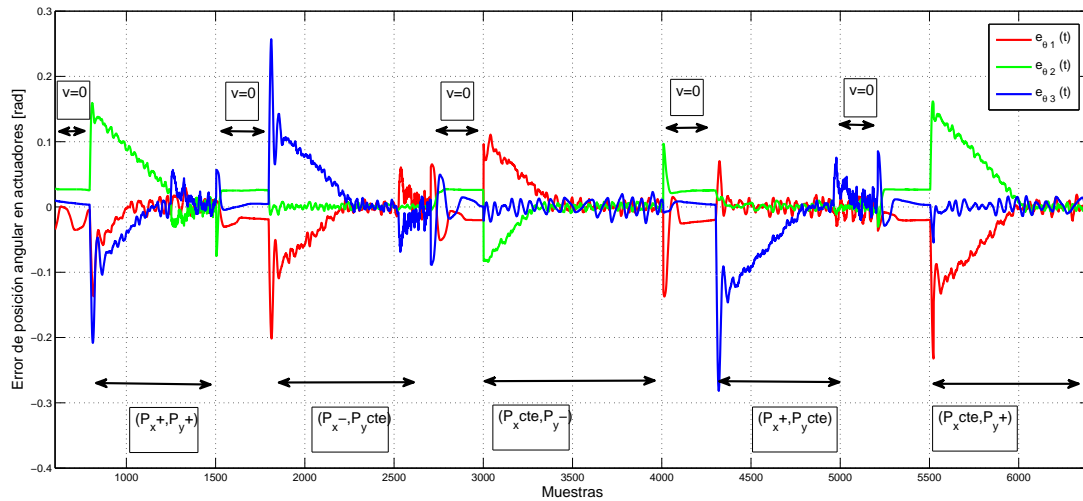


Figura 6.19: Experimento múltiples líneas CID - Seguimiento de trayectorias rectas en diferentes direcciones, se puede apreciar como el error es corregido en todos los casos, sin el uso de diferenciador de segundo orden.

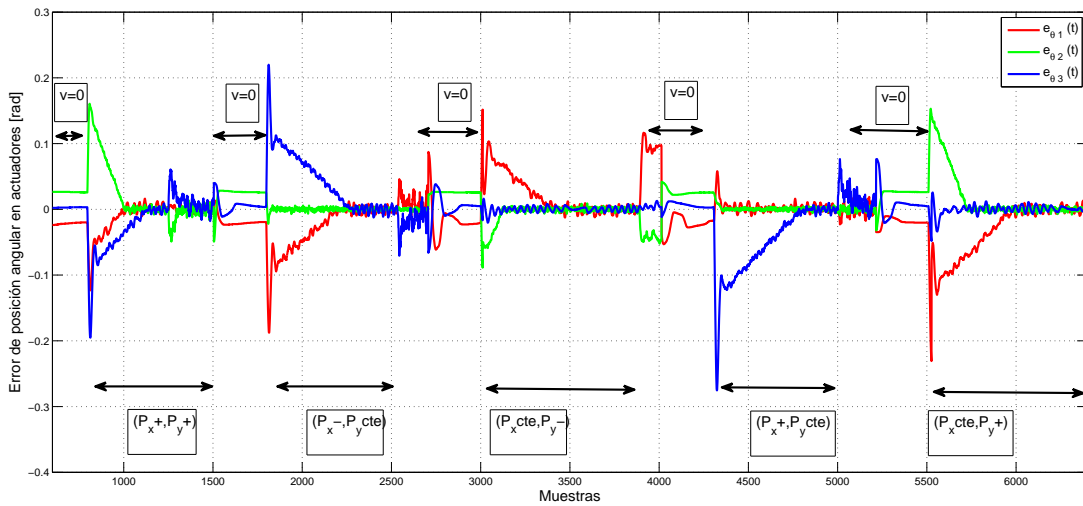


Figura 6.20: Experimento múltiples líneas 3-STA - Seguimiento de trayectorias rectas en diferentes direcciones, se puede apreciar como el error es corregido en todos los casos, sin el uso de diferenciador de segundo orden.

Conclusiones

7.1 Análisis de resultados

7.1.1 Errores de seguimiento

Para evaluar el desempeño de los algoritmos en términos de la eficiencia del seguimiento en la tarea de control, consideraremos el error de posición, además es la única medición directa que obtenemos a partir de los sensores en actuadores. En la Figura 7.1 se muestran los tipos de error a considerar, durante el seguimiento de trayectoria, tomando como ejemplo la lectura de error del experimento con CID, todos son considerados para evaluar el desempeño de la tarea de control. Hay un caso especial, que en el super-twisting ó primer orden, donde no se toma en cuenta el error en estado estable (e_{ss}) que se muestra en la Figura 7.1, ya que teóricamente el estado (posición) nunca converge a una banda, debido a su comportamiento asintótico en los estados. En resumen los errores considerados son:

- Error en estado estable.
- Error máximo.
- RSME.

En esta sección se muestran algunas tablas que nos permiten dar algunas conclusiones acerca del desempeño de cada algoritmo. El primer punto comparativo es el error acumulado a lo largo de cada

7. CONCLUSIONES

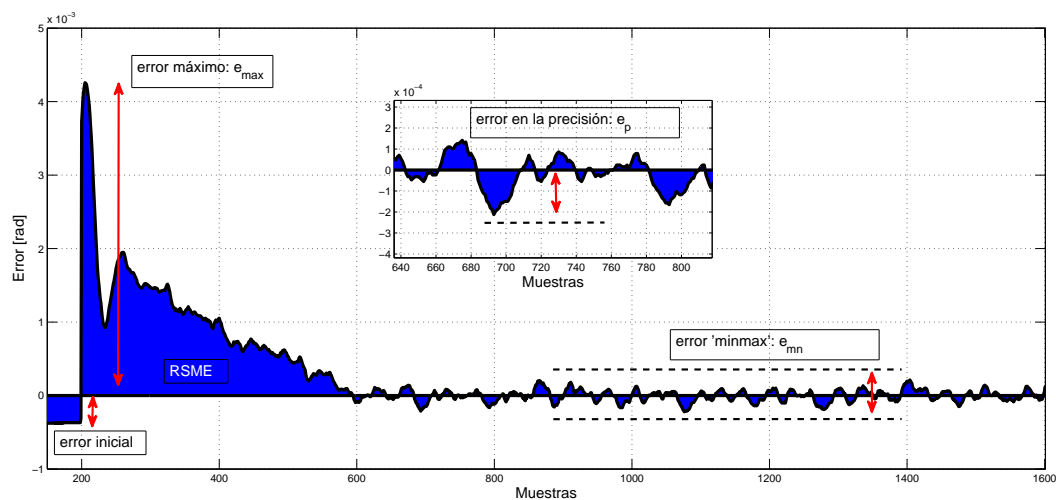


Figura 7.1: Errores considerados - Se muestran los errores durante el seguimiento de trayectoria, todos son considerados para evaluar el desempeño de la tarea de control.

experimento, esta medida del error es conocida por sus siglas en inglés como RSME (*root-square mean error*). En la Tabla 7.5 se muestran los resultados de obtener estos valores a partir de resultados experimentales, de la cual se puede leer que el mejor desempeño fue el que resultó de utilizar el CID sin diferenciador, seguido por muy cerca el 3-STA sin diferenciador. La utilización del diferenciador de segundo orden en estos algoritmos no significó un mejor desempeño en términos del RSME.

Algoritmo	RSME $\sqrt{\sum_i^N \ e_{\theta_i}\ ^2}$ [rad]
Nominal	0.111
Primer Orden	0.0413
Super-Twisting	0.0391
CID	0.0321
3-STA	0.0360
CID/3STA con diferenciador	0.0483/0.0563

Tabla 7.1: RSME (root-mean square error). Esta medida del error nos da una idea de qué tan eficiente se realiza la tarea de control durante todo el seguimiento.

La gráfica de la Figura 7.2, es un acercamiento detallado de la lectura del error generado para un solo motor. Tomando ese valor

nos podemos dar una idea de cuanto están afectando los algoritmos al comportamiento de los motores, sin importar que estos convergan o no a cero, no se toma el caso del PD nominal pues nunca converge, y la banda de errores es calculada con ayuda de Matlab para los demás algoritmos. Estos valores son presentados en la Tabla 7.2, de la cual podemos concluir que cuando añadimos los algoritmos CID y 3-STA. Es importante comparar esta banda de tolerancia, para estar seguro que no suceda como con el algoritmo de primer orden, en donde se inducen oscilaciones.

	$ e_{ss1} $ [rad]	$ e_{ss2} $ [rad]	$ e_{ss3} $ [rad]
Nominal	5×10^{-4}	4×10^{-4}	7.9×10^{-4}
Primer orden	0.0051	0.004841	0.0056
C.I.D.	5.25×10^{-4}	5.732×10^{-4}	6.6250×10^{-4}
3-STA	4.31×10^{-4}	4.626×10^{-4}	5.1×10^{-4}

Tabla 7.2: Errores en cada motor con $e_{ss} = e_{imax} - e_{imin}$ pasado $t > t_1$. Errores en estado estable una vez que se mantienen en alguna banda en la que pasado un tiempo, éstos no presentan un cambio significativo, excepto para el algoritmo de Primer Orden.

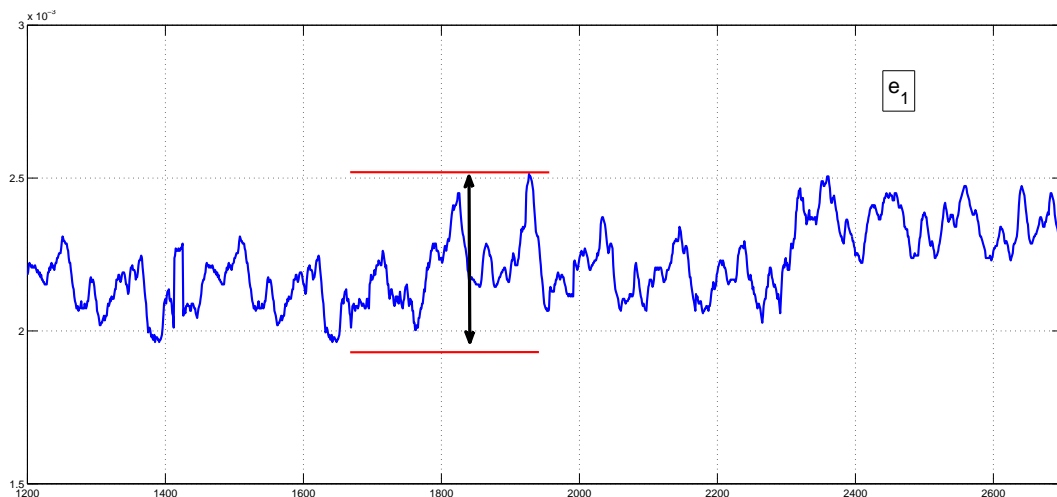


Figura 7.2: $e_{ss} = e_{imax} - e_{imin}$ pasado $t > t_1$ - Banda de tolerancia para el error generado, esta gráfica corresponde a la lectura del error en el experimento nominal, es importante comparar esta banda de tolerancia, para estar seguro que no suceda como con el algoritmo de primer orden, en donde se inducen oscilaciones.

7. CONCLUSIONES

Otra característica que podemos analizar de las gráficas es el valor del error máximo del seguimiento, aún considerando el sobrepaso antes de que el error se empiece a atenuar (a excepción del experimento nominal).

	$ e_{\theta 1} _{max}$ [rad]	$ e_{\theta 2} _{max}$ [rad]	$ e_{\theta 3} _{max}$ [rad]
Primer orden	0.2948	0.167	0.0887
Super-Twisting	0.3285	0.0167	0.0751
CID	0.1503	0.0796	0.0374
3-STA	0.2403	0.1196	0.0274
CID/3-STA con diferenciador	0.0042/0.0044	0.0018/0.0018	0.0024/0.0024

Tabla 7.3: Valor máximo del error en cada experimento, para cada motor, seguimiento de línea recta: $|e_{\theta}|_{max}$.

7.1.2 Señales de control

En las Figuras 7.3 y 7.4 se muestran las señales de control adicionales al nominal, por algoritmos de modos deslizantes continuos. Algo que hay que comentar de éstas gráficas es la amplitud del control. La Figura 7.4 es la misma que 7.3, pero considerando sólo un motor, pues de esta manera se puede apreciar mejor cómo el uso del diferenciador, permite que el controlador tenga una menor amplitud, lo cual se puede observar para los casos del CID y 3-STA. Sin embargo para el caso del Super-twisting, la señal nunca converge a alguna banda, más bien tiene un comportamiento en el que crece indefinidamente, teóricamente se debe aproximar a alguna asíntota, pero en la práctica éste es el comportamiento.

Con base en las gráficas presentadas, se muestra en la Tabla 7.4 un listado de los valores máximos que toman las señales de control. La que menos amplitud tiene es el Super-Twisting, pero también hay que señalar que para CID y 3-STA, agregar un diferenciador disminuye la amplitud que la señal de control requiere.

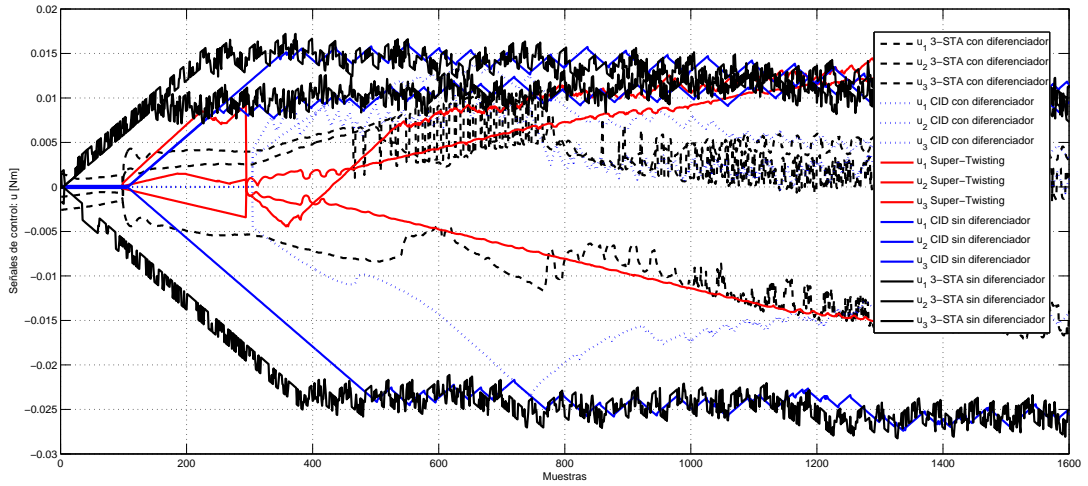


Figura 7.3: Señales de control continuas - Se muestran todas las señales de control (continuas) implementadas, para el caso del experimento de la línea recta.

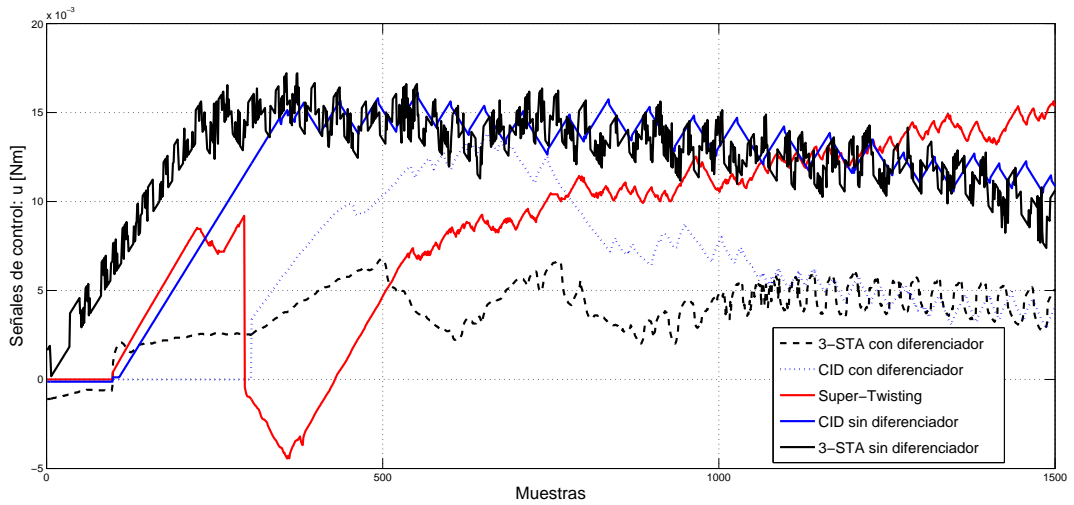


Figura 7.4: Señales de control continuas (sólo un motor) - Se muestra ahora sólo una señal de control, correspondiente al motor 1 del robot. Para mostrar mejor el diferente comportamiento de cada algoritmo.

Algoritmo	$ u_1 _{max}$ [Nm]	$ u_2 _{max}$ [Nm]	$ u_3 _{max}$ [Nm]
Super-Twisting	0.0127	0.0156	0.0163
CID	0.0274	0.0127	0.0161
3-STA	0.0283	0.0192	0.0194
CID/3-STA con diferenciador	0.0236/0.0172	0.0196/0.0108	0.0138/0.0068

Tabla 7.4: Valor máximo de la señal de control en cada motor, para un caso de seguimiento de línea recta.

7. CONCLUSIONES

Algoritmo	Tiempo [s]
Nominal	4.61
Primer Orden	4.631
Super-Twisting	4.649
CID	4.7321
3-STA	4.7360
CID/3-STA con diferenciador	5.104/5.12

Tabla 7.5: Tiempos de ejecución. Se presentan los tiempos que tardó cada experimento en hacer la tarea de control, para el caso de la línea recta.

7.1.3 Tiempos de ejecución

7.2 Comentarios finales

Una de las motivaciones principales al utilizar los algoritmos por modos deslizantes es su robustez ante incertidumbres y perturbaciones, utilizando poca información del sistema. Refiriéndonos a los últimos algoritmos probados, es claro que aunque estos tengan la propiedad de convergencia en tiempo finito de los estados, además de una señal de control continua, sus desventajas podrían ser la difícil sintonización, además de que requieren también conocer un poco de la dinámica de la perturbación. Por tanto el control de este sistema como una caja negra, o dinámica desconocida, aún queda por explorar. Una de las diferencias principales entre el 3-STA y CID es que el primero ofrece un método más simple (al menos al inicio de este trabajo así lo fue, sin embargo como se desarrollo a la par los autores pudieron hacer nueva actualización del algoritmo), y por tanto más fácil de implementar, ya que requiere la utilización de menos constantes si se quiere obtener todos los parámetros de la función de Lyapunov. Un aspecto muy notable es que el CID contiene sólo a la posición en la discontinuidad, a diferencia del 3-STA que contiene una combinación lineal de posición y velocidad (muy parecido al algoritmo terminal por modos deslizantes). Esto puede ser conveniente dependiendo de la aplicación, que para nuestro caso en el robot paralelo, ha sido más conveniente que la

discontinuidad sólo aparezca la posición, ya que no se cuenta con un sensor directo de velocidad y el switcheo ocasionado por la función signo puede no ser el apropiado. Lo cual requeriría el uso de métodos más sofisticados vía software para la obtención de la velocidad. Diferencias pueden ser las siguientes enlistandolas:

- Cambios estructurales en los algoritmos, es decir: la discontinuidad de la función *sign* del CID contiene solamente a la posición, y el 3-STA contiene posición y velocidad.
- El algoritmo 3-STA contiene más parámetros en su función de Lyapunov, lo cual puede hacer más tedioso la búsqueda de estos, si es el caso de ser necesario.

En la experimentación sin condiciones ideales, también nos encontramos con otro tipo de fenómenos que afectan el desempeño de los controladores. Volviendo a hacer referencia a los algoritmos CID y 3-STA, podemos mencionar que se ha utilizado uno de los métodos de integración más simples, incluso también la diferenciación para la obtención de la velocidad. La obtención de la velocidad también podría obtenerse con otros métodos más eficientes, como el uso de observadores o diferenciadores, pero que requieren más tiempo de trabajo por su complejidad. Algunos puntos a mencionar son:

- Utilización de métodos para obtención de derivadas, en este caso velocidad y/o aceleración dependiendo el caso.
- Métodos de integración que requiera el algoritmo empleado (Super-Twisting, 3-STA, HOSM con integrador).

7. CONCLUSIONES

Algoritmo	Precisión	Chattering/Control	Información	Perturbación $\delta(t)$
M.D. Convencionales	μ	✓/Discontinuo	x, \dot{x}	$ \delta(t) \leq D$
Súper-Twisting	μ	Continuo	x, \dot{x}	$ \frac{d}{dt}\delta(t) \leq \Delta$
Twisting	μ^2	✓/Discontinuo	x, \dot{x}	$ \delta(t) \leq D$
Tercer Orden	μ^3	Continuo	x, \dot{x}, \ddot{x}	$ \delta(t) \leq D$
3-STA, CID	μ^3	Continuo	x, \dot{x}	$ \frac{d}{dt}\delta(t) \leq \Delta$

Tabla 7.6: Propiedades de los controladores implementados.

Apendices

A.1 Tarjeta GT-400-SV

Esta tarjeta cuenta con una librería de instrucciones para diferentes aplicaciones que requieran la utilización de un controlador de movimiento. Los siguientes detalles técnicos de implementación: tiempo de muestreo, filtros digitales, programación de tareas e instrucciones de control, adquisición de señales, todos esos detalles ac-

erca de la implementación vía programación de los algoritmos se puede encontrar en [27].

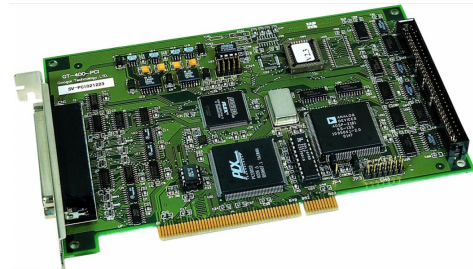


Figura A.1: Tarjeta GT-400-SV

A.2 Filtro digital

Mediante el filtro digital la PC envía la señal de salida que reciben los servoamplificadores. La inyección de control adicionada al algoritmo nominal de fábrica se implementó mediante este filtro digital. Se utiliza la señal correspondiente a “Net difference compensation”, mediante la instrucción *GTSetMtr_Bias*, para más detalle se puede encontrar en [27] (pag. 14).

A. APENDICES

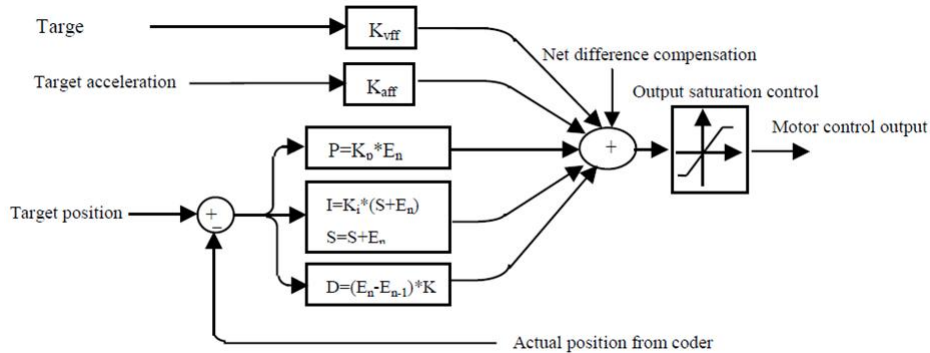


Figura A.2: Filtro digital

A.3 Servoamplificadores

La etapa de potencia corresponde a los servoamplificadores PY2. También para la calibración de los encoders y tener una posición cero se debe conectar este sistema directamente a la PC mediante una comunicación RS232. La comunicación RS232 permite también limpiar alarmas que por seguridad frecuentemente estos equipos blo-

quean automáticamente.

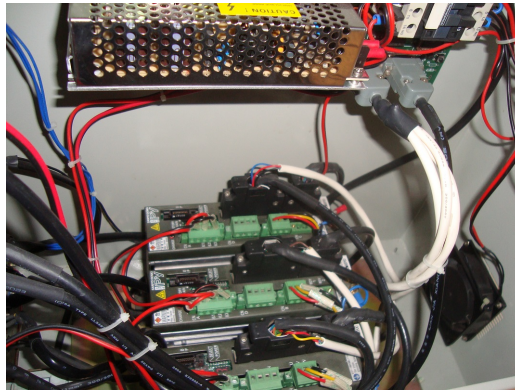


Figura A.3: Sistema de amplificadores

Se han modificado algunos parámetros del servoamplificador, en la Figura A.4 se puede apreciar con línea roja, la señal utilizada para poder inyectar una señal de torque enviada desde la PC mediante la tarjeta SV. La línea roja muestra como al final esta señal acaba llegando al servomotor. La relación voltaje/torque también fue modificada ya que esta puede ser configurable mediante el software especial con el que cuentan estos servoamplificadores.

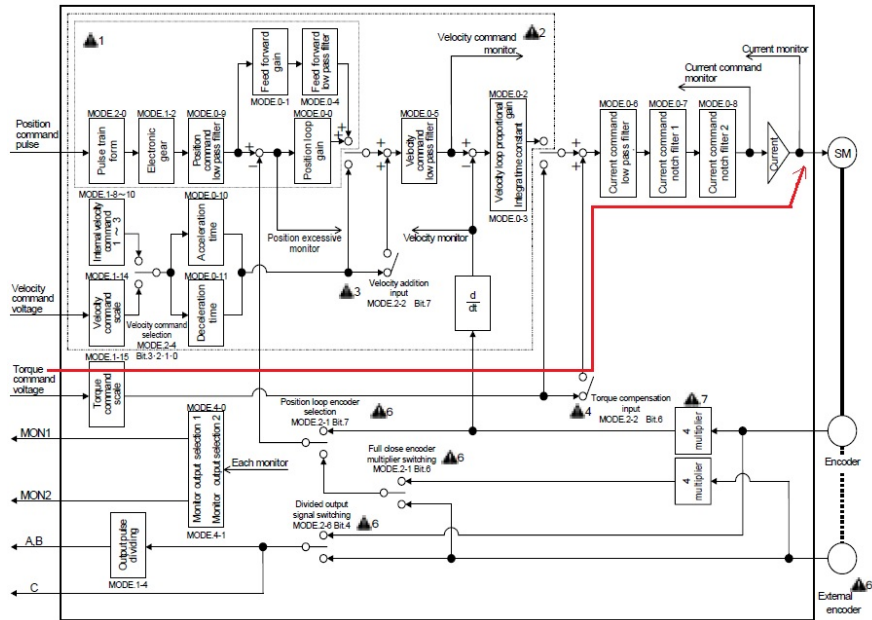


Figura A.4: Parámetros del amplificador PY2 -

A.4 Relación voltaje-torque

Existe relación voltaje-torque generada por el motor a partir de la señal enviada. Esta relación se utilizó para tener certeza de que el modelo de la planta que considera torque, sea congruente con la implementación del sistema real. Originalmente se encontraba en una escala del 50%, se reconfiguro a 100% para fines prácticos.

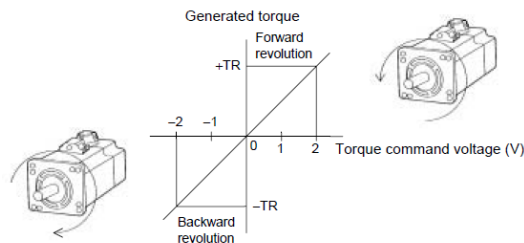


Fig. 9-17 Torque Command - Generated Torque

Figura A.5: La relación voltaje torque es configurable en el sistema de servoamplificadores gracias al controlador interno -

A. APENDICES

A.5 Entorno de programación C++

Para la aplicación de los algoritmos de control a la tarjeta GT-400-SV, se utilizó el entorno de programación Microsoft C++ [[28]], utilizando la plataforma Visual Studio 2010.

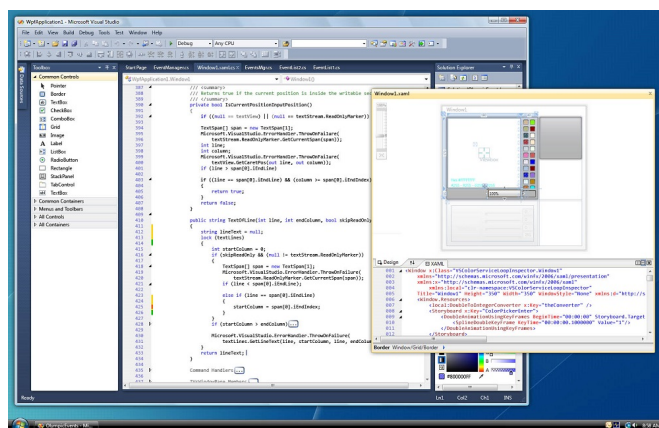


Figura A.6: Entorno C++ - Programación de algoritmos en el entorno Visual 2010

Bibliografía

- [1] MARK W. SPONG, SETH HUTCHINSON, AND M. VIDYASAGAR. *A Mathematical Introduction to Robotic Manipulation*. 1994. 1, 22
- [2] S. SHANKAR SASTRY RICHARD M. MURRAY, ZEXIANG LI. **A Mathematical Introduction to Robotic Manipulation**. *CRC Press*, 1994. 1
- [3] J.-P-MERLET. *Parallel Robots*. Springer, 2006. 1
- [4] GRIGORE GOGU. *Structural Synthesis of Parallel Robots*. Springer, 2008. 1
- [5] PHILIPPE BEGON, FRANCOIS PIERROT, AND PIERRE DAUCHEZ. **Fuzzy Sliding Mode Control of a Fast Parallel Robot**. *IEEE International Conference on Robotics and Automation*, 1995. 2
- [6] S. KICK AND W. SCHUMACHER. **A Parallel x-y Manipulator with Actuation Redundancy for High-Speed and Active-Stiffness Applications**. *IEEE Proceedings International conference on Robotics & Automation*, 1998. 2
- [7] HUI CHENG, YIU-KUEN YIU, AND ZEXIANG LI. **Dynamics and Control of Redundant Actuated Parallel Manipulators**. *IEEE ASME TRANSACTIONS ON MECHATRONICS*, 2003. 2
- [8] TRUNG DO THAN, JENS KOTLARSKI, BODO HEIMANN, AND TOBIAS ORTMAIER. **On the Inverse Dynamics Problem of General Parallel Robots**. *Proceedings of the 2009 IEEE International Conference Mechatronich, Málaga*, 2009. 2
- [9] WEIWEI SHANG AND SHUANG CONG. **Nonlinear adaptive task space control for a 2-DOF redundantly actuated parallel manipulator**. *Springer*, 2010. 2
- [10] WEIWEI SHANG, SHUANG CONG, YAOXIN ZHANG, AND YANYANG LIANG. **Active Joint Synchronization Control for a 2-DOF Redundantly Actuated Parallel Manipulator**. *IEEE Transactions on Control Systems Technology*, **17 No. 2**, 2009. 2

BIBLIOGRAFÍA

- [11] WEIWEI SHANG AND SHUANG CONG. **Adaptive Compensation of Dynamics and Friction for a Planar Parallel Manipulator with Redundant Actuation.** *Proceedings of the 2010 IEEE International Conference on Robotics and Biomimetics*, 2010. 2
- [12] JENS KOTLARSKI, BODO HEIMANN, AND TOBIAS ORTMAIER. **Influence of Kinematik Redundancy on the singularity-free workspace of parallel kinematik machines.** *Higher Education press and Springer-Verlag Berlin Heidelberg*, 2012. 3
- [13] C. HANSEN, J. KOTLARSKI, AND T. ORTMAIER. **Optimal motion planning for energy efficient multi-axis applications.** *International Journal of Mechatronics and Automation*, 4, 2014. 3
- [14] V.I. UTKIN. *Sliding Modes in Control Optimization.* Springer, Heidelberg, 1992. 3
- [15] SV EMELYANOV, SK KOROVIN, , AND LV LEVANTOVSKY. **Second order sliding modes in controlling uncertain systems.** *Soviet Journal of Compute and System Science*,, 1986. 3
- [16] RAÚL SANTISTEBAN, LEONID FRIDMAN, AND JAIME MORENO. **Finite-time convergence analysis for Twisting controller via a strict Lyapunov function.** *International Workshop on Variable Structure Systems Mexico City*, 2010. 3, 30
- [17] A. LEVANT. **Sliding order and sliding accuracy in sliding mode control.** *International Journal of Control*, 58:1247–1263, 1993. 4
- [18] JAIME A. MORENO AND MARISOL OSORIO. **A Lyapunov approach to second-order sliding mode controllers and observers.** *Proceedings of the 47th IEEE Conference on Decision and Control Cancun, Mexico*, 2008. 4, 29
- [19] A. LEVANT. **Robust Exact Differentiation via Sliding Mode Technique.** *Automatica*, 1998. 4, 35
- [20] A. LEVANT. **Higher-order Sliding Modes, Differentiation and Output-feedback Control.** *International Journal Control*, 2003. 4
- [21] V. UTKIN, J. GULDNER, AND J. SHI. *Sliding Mode control in Electro-Mechanical Systems.* CRC Press, 2009. 4
- [22] Y. SHTESSEL, C. EDWARDS, L. FRIDMAN, AND A. LEVANT. *Sliding Mode Control and Observation.* Springer, 2013. 4
- [23] L. MEIROVICHM. **Methods of analytical dynamics.** *McGaw-Hill*, 1970. 18
- [24] CÉSAR AGUSTÍN ZAMORA SALAZAR. *Control Integral Doscontinuo Para Sistemas de Segundo Orden.* Master’s thesis, Universidad Nacional Autónoma de México, 2013. 31, 32

- [25] CÉSAR ZAMORA AND JAIME MORENO. **Control integral discontinuo para sistemas mecánicos.** In *Congreso Nacional de Control Automático CNCA 2013, Ensenada México.*, 2013. 31
- [26] S. KAMAL, A. CHALANGA, B. BANDYOPADHYAY, J.A. MORENO, AND L. FRIDMAN. **Third Order Sliding Mode Super-Twisting Algorithm.** 2014. 33
- [27] GOOGOL TECHNOLOGY. **Programming Manual for GT Motion Controller.** 2003. 81
- [28] BJARNE STROUSTRUP. **The C++ Programming Language: Special Edition.** 2001. 84