



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

PUESTA EN FUNCIONAMIENTO DEL
ROBOT PARALELO NOVINT FALCON

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO ELÉCTRICO-ELECTRÓNICO

PRESENTA:

IVÁN TORRES RODRÍGUEZ

DIRECTOR DE TESIS:

M.I. JOSÉ DANIEL CASTRO DÍAZ



Ciudad Universitaria CD. MX.

Abril 2017



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Jurado asignado:

Presidente:	DR. MARCO ANTONIO ARTEAGA PÉREZ
Secretario:	DR. PAUL ROLANDO MAYA ORTÍZ
Vocal:	M.I. JOSÉ DANIEL CASTRO DÍAZ
1 ^{er} Suplente:	DR. HOOVER MUJICA ORTEGA
2 ^{do} Suplente:	M.I. MAURO GILBERTO LÓPEZ RODRÍGUEZ

Esta tesis se realizó en el Laboratorio de Robótica del Departamento de Control de la Facultad de Ingeniería.

Director de tesis:

M.I. JOSÉ DANIEL CASTRO DÍAZ

A mi madre, Raquel, que con tu esfuerzo, dedicación y cariño me has dado la oportunidad de vivir una gran vida. Sin ti nada de ésto sería posible.

A mi padre, Juan, por todas las experiencias que compartimos.

Agradecimientos

Al pueblo mexicano por darme la oportunidad de obtener educación a través de nuestra máxima casa de estudios la Universidad Nacional Autónoma de México.

A mis profesores de la Facultad que con su ejemplo me motivaron a ser mejor cada día.

A los miembros del jurado por el tiempo dedicado a la revisión de este trabajo.

A mi asesor Daniel por la dedicación, confianza y apoyo que me concedió durante la elaboración de este trabajo y quien además de ser un mentor es un buen amigo.

A mis compañeros de la Facultad y del Laboratorio de Robótica por todas las experiencias que compartimos en esta etapa de nuestras vidas, y sobre todo a aquellos con los que comparto una amistad.

A mis compañeros de vida en especial a Mariana por su apoyo y cariño pero sobre todo, por ser fuente de motivación e inspiración en mi vida. Sin dejar de mencionar a Ometeotl por acompañarme al escribir este trabajo.

Índice general

1. Introducción	1
1.1. Motivación	2
1.2. Definición del problema	2
1.3. Estado del arte	3
1.3.1. Robots Paralelos	3
1.3.2. Interfaces hápticas	5
1.4. Contribuciones	7
1.5. Organización de la tesis	8
2. Análisis cinemático y dinámico	9
2.1. Geometría del robot <i>Novint Falcon</i>	9
2.2. Cinemática Directa	12
2.3. Cinemática Inversa	13
2.4. Matriz Jacobiana	15
2.4.1. Singularidades en Cinemática Directa	18
2.4.2. Singularidades en Cinemática Inversa	18
2.5. Modelo Dinámico	19
2.5.1. Regresor de parámetros	22
3. Control del robot <i>Novint Falcon</i>	25
3.1. Control PID	26
3.1.1. Sintonización de controlador PID	27
3.2. Control Adaptable	28
3.3. Diseño de trayectorias	32

3.3.1.	Trayectoria punto a punto	33
3.3.2.	Lemniscata de Bernoulli	34
4.	Implementación y evaluación experimental	37
4.1.	Implementación con <i>Haptik library</i>	37
4.1.1.	Controlador PID en <i>C++</i>	38
4.1.2.	Controlador Adaptable en <i>Simulink</i>	39
4.1.3.	Uso del diferenciador de Levant	40
4.2.	Evaluación experimental	40
4.2.1.	Resultados de controlador PID	41
4.2.2.	Resultados de controlador adaptable	41
4.3.	Análisis de resultados	46
5.	Robot <i>Novint Falcon</i> como dispositivo háptico	49
5.1.	Sistema de realidad virtual	49
5.2.	Renderizado háptico	51
5.2.1.	Detección de Colisiones	52
5.2.2.	Control de fuerza	52
5.3.	Implementación mediante HDAL	53
6.	Conclusiones	55
6.1.	Trabajo a futuro	56
A.	Propiedades de matrices del sistema	57
B.	Diferenciador de Levant	59
C.	Guía de instalación	61
	Bibliografía	65

Índice de figuras

1.1. Robot <i>Novint Falcon</i>	2
1.2. Tipos de cadenas cinemáticas	4
1.3. Mecanismo de entretenimiento patentado por Gwinnet [8]	5
1.4. Plataforma de Stewart	5
1.5. Robot Delta patentado por Tsai [13]	5
1.6. Dispositivo Grope-3 en uso [3]	6
1.7. Exoesqueleto Rutgers Master Glove [4]	6
1.8. Interfaz háptica con forma de bolígrafo [11]	7
1.9. Robot <i>Omni Phantom</i>	7
2.1. Sistemas de Referencia Base y Háptico [28]	10
2.2. Sistemas de Referencia Base B y de cada extremidad T_i	10
2.3. Diagrama geométrico de la i -ésima extremidad.	12
2.4. Diagrama de la i -ésima extremidad.	16
3.1. Esquema genérico de control	26
3.2. Tipos de respuesta	27
3.3. Respuesta experimental en lazo cerrado con ganancia k_{cr}	29
3.4. Trayectoria de Lemniscata de Bernoulli con $a = 20[mm]$ y $b = 10[mm]$	35
4.1. Comunicación de robot <i>Novint Falcon</i>	38
4.2. Diagrama de implementación de controlador PID.	39
4.3. Diagrama de implementación de controlador adaptable.	40
4.4. Posición articular real y deseada PID	42
4.5. Error articular PID	43

4.6. Posición articular real y deseada en controlador adaptable	44
4.7. Error articular en controlador adaptable	45
4.8. Error articular en controlador adaptable a 100 segundos	47
4.9. Parámetros estimados en 200 segundos	48
5.1. Ambiente virtual	50
5.2. Arquitectura básica de una aplicación de realidad virtual con háptica	51
5.3. Esquema de renderizado háptico	51
5.4. Secuencia de interacción con sistema de realidad virtual.	54
C.1. Asistente de instalación <i>Falcon Setup</i>	62
C.2. Asistente de instalación HDAL	62
C.3. Carpeta <i>C:/Program Files/Novint</i>	63
C.4. Asistente de instalación <i>Haptik Library</i>	63

*The people who are crazy enough to think they can change the world
are the ones who do.*

Anuncio *Think Different* Apple, 1997

Capítulo 1

Introducción

Desde la aparición de las máquinas automáticas, científicos e ingenieros de todo el mundo dedican sus esfuerzos a analizar, diseñar, construir y controlar mecanismos cada vez más adaptados al entorno y quehaceres del hombre. Como ejemplo los robots, que en la industria ejecutan procesos de manufactura rápidos y eficientes obteniendo una mayor producción a un menor costo. Debido a los beneficios que generan, su demanda se encuentra en aumento y para el año 2019 se espera que el número de unidades en todo el mundo sea de 2,6 millones [9]. Lo anterior ha permitido una rápida evolución de esta tecnología trayendo como consecuencia que sus precios sean cada vez más competitivos, permitiendo su aplicación en países en desarrollo dando lugar a la llamada *democratización de la robótica*.

Un campo que se beneficia directamente con esta democratización es la realidad virtual ya que normalmente se utilizan robots como dispositivos hápticos. En general los sistemas de realidad virtual consisten en una simulación por computadora, la cual sensa el estado físico del usuario (posición, temperatura, frecuencia cardiaca, etc.) y posteriormente lo retroalimentan mediante estímulos sensoriales con el fin de crear una experiencia interactiva. Normalmente los sistemas de realidad virtual hacen retroalimentación visual por medio de pantallas, auditiva por medio de parlantes y más recientemente táctil por medio de dispositivos hápticos. La tecnología háptica recrea la sensación del tacto mediante la aplicación de fuerzas, vibraciones o movimientos ejecutados por dichos dispositivos.

El robot *Novint Falcon* que se muestra en la Figura 1.1 fue uno de los primeros

Figura 1.1: Robot *Novint Falcon*

dispositivos hápticos lanzados al mercado de consumo. Fabricado por la empresa *Novint*, es un dispositivo háptico orientado a la industria de los videojuegos, aunque por su diseño es posible usarlo como robot manipulador. Su producción en masa lo hacen el dispositivo háptico más económico del mercado con un precio menor a los 250 USD, resaltando que antes de su lanzamiento dispositivos similares eran sumamente costosos.

1.1. Motivación

El robot *Novint Falcon* fue recientemente adquirido por el Laboratorio de Robótica del Departamento de Control de la División de Ingeniería Eléctrica de la Facultad de Ingeniería, por lo que requiere de su análisis y puesta en funcionamiento. En el Laboratorio de Robótica la docencia e investigación son realizados con robots de estructura en serie. Esta tesis al presentar un robot con estructura paralela aporta el conocimiento necesario para una primera aproximación a este tipo de mecanismos.

1.2. Definición del problema

El problema a resolver en este trabajo es realizar la puesta en funcionamiento del robot paralelo *Novint Falcon* explorando sus capacidades hápticas para que en un

futuro se puedan desarrollar aplicaciones complejas, como por ejemplo su integración en simuladores de cirugía con retroalimentación de fuerza. Por lo tanto, resolverlo implica plantear los siguientes objetivos:

- Realizar el análisis de la cinemática y dinámica del robot paralelo en configuración delta *Novint Falcon*.
- Evaluar experimentalmente dos esquemas de control en el robot *Novint Falcon*.
- Realizar un programa que demuestre el uso del robot *Novint Falcon* como dispositivo háptico.
- Elaborar un manual y proporcionar los archivos necesarios para la puesta en funcionamiento del robot *Novint Falcon*.

1.3. Estado del arte

Los robots pueden ser clasificados según su estructura cinemática ya sea en serie o en paralelo. Se clasifican en serie si la secuencia de eslabones que lo componen, forma una cadena abierta como se muestra en la Figura 1.2(a). En cambio el robot de la Figura 1.2(b) es paralelo ya que su estructura cinemática forma una cadena cerrada. Este tipo de robots tienen la ventaja de poseer alta rigidez, baja inercia y una gran capacidad de carga, sin embargo sus desventajas más notables son su pequeño espacio de trabajo y su complejo mecanismo [14]. El robot *Novint Falcon* presenta más de una cadena cinemática cerrada y de acuerdo con su estructura se trata de un robot paralelo.

A continuación se describen las investigaciones más recientes acerca del tema, comenzando por lo general con robots paralelos, para terminar en lo particular con interfaces hápticas y los últimos trabajos realizados sobre el robot *Novint Falcon*.

1.3.1. Robots Paralelos

El primer trabajo teórico sobre un mecanismo paralelo se remonta a 1928 y fue presentado por Gwinnet [8] siendo patentado como un dispositivo de entretenimiento tal y como se ilustra en la Figura 1.3 pero que nunca fue llevado a la realidad. Más

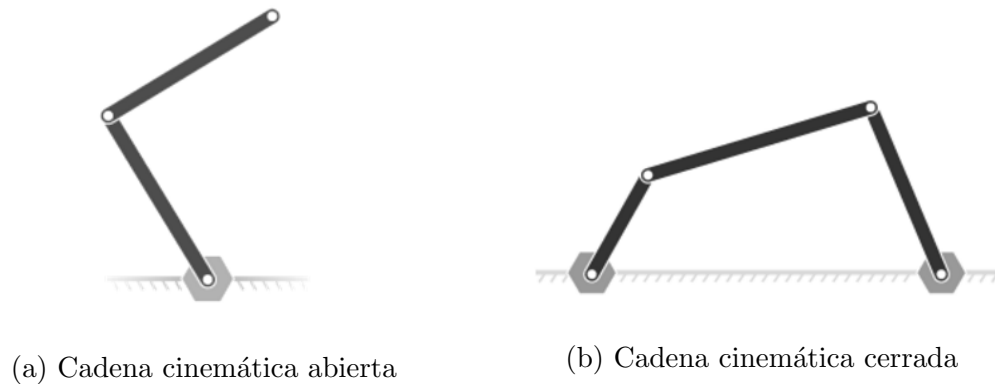


Figura 1.2: Tipos de cadenas cinemáticas

tarde en 1942 Pollard [21] diseñó el primer robot paralelo industrial. Cinco años después el Dr. Eric Gough inventó otro robot paralelo de seis grados de libertad que revolucionó la industria de la robótica: el primer hexápodo octaédrico, pero fue hasta 1962 cuando fue aplicado en un aparato de pruebas de neumáticos [6]. A partir del diseño de Gough, en 1965 Stewart publicó un artículo que describía una plataforma de seis grados de libertad para ser usada como simulador de vuelo [27]. Este trabajo tuvo un significativo impacto en el posterior desarrollo de mecanismos paralelos y fue el responsable de popularizar el diseño de Gough en la academia. En la Figura 1.4 se muestra la plataforma de Stewart.

Fueron Gough y Stewart los pioneros de los robots paralelos y sentaron un precedente para las futuras invenciones de este tipo de mecanismos, los cuales fueron aplicados en simuladores de vuelo y de manejo, dispositivos médicos, y más recientemente han sido usados para el desarrollo de máquinas de precisión [29]. Años más tarde llegó otro robot paralelo que marcó un paradigma de diseño, se trata del robot Delta patentado en el año 1990 por Clavel [5]. Consiste en una plataforma móvil conectada a una base fija en forma de triángulo equilátero a través de tres brazos los cuales se encuentran actuados mediante articulaciones de revolución ubicadas en la base. El movimiento es transmitido a la plataforma mediante un par de paralelogramos conectados a cada brazo, dando como resultado tres grados de libertad. A diferencia de la plataforma de Stewart sus dimensiones son más compactas, pues su objetivo era la manipulación de objetos pequeños a gran velocidad, principalmente

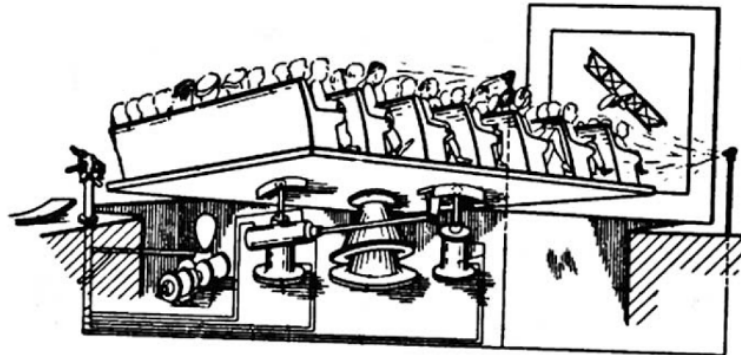


Figura 1.3: Mecanismo de entretenimiento patentado por Gwinnet [8]

en el área de empaquetado.

La configuración Delta inspiró el diseño del robot *Novint Falcon*, muy similar al propuesto por Tsai [13] en el cual las articulaciones esféricas del diseño de Clavel [5] son reemplazadas por articulaciones de revolución de un solo grado de libertad como se muestra en la Figura 1.5. Este diseño permite que la posición y orientación del manipulador estén desacopladas; además de que el uso de articulaciones esféricas reduce su costo.

1.3.2. Interfaces hápticas

Uno de los pioneros en el desarrollo de interfaces hápticas fue Brooks quien en 1967 inició un proyecto para desarrollar un dispositivo visual-háptico de campos de



Figura 1.4: Plataforma de Stewart

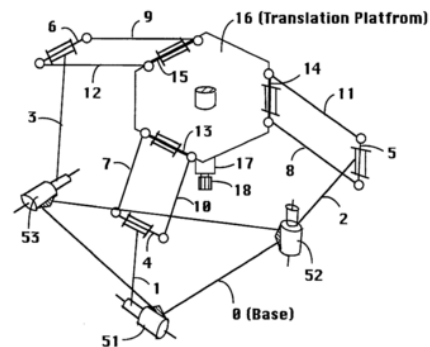


Figura 1.5: Robot Delta patentado por Tsai [13]

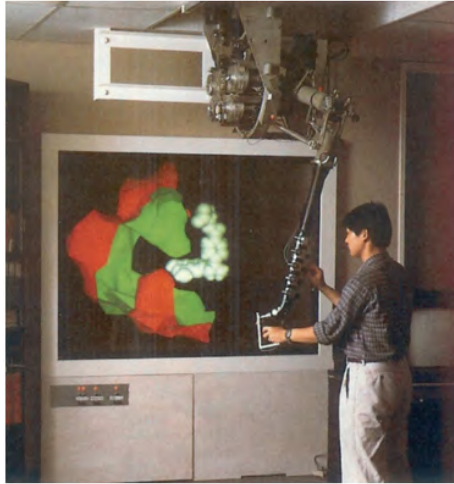


Figura 1.6: Dispositivo Gropo-3 en uso [3]



Figura 1.7: Exoesqueleto Rutgers Master Glove [4]

fuerza de seis grados de libertad para interactuar con moléculas de proteínas [3]. Su primer prototipo fue el Gropo-1 publicado en 1971 [2]. Cuenta con una plataforma de dos grados de libertad actuada con dos servomotores que brindan retroalimentación de fuerza. Más tarde, en la década de los 80, presenta el sistema final Gropo-3, el primer dispositivo háptico de seis grados de libertad con retroalimentación ilustrado en la Figura 1.6.

En la década de los noventa hubo un gran actividad en el desarrollo de interfaces hápticas. En el año de 1992 Burdea [4] presentó un exoesqueleto para su uso en realidad virtual. Un exoesqueleto es un conjunto de actuadores sujetos a alguna parte del cuerpo, por ejemplo la mano. Este mecanismo puede aplicar fuerza a los dedos, pero no puede generar una fuerza con respecto a la tierra como la gravedad o la representación de un muro virtual. En la Figura 1.7 se ilustra el exoesqueleto Rutgers Master diseñado por Burdea. Los usuarios de exoesqueletos experimentaron problemas al colocarse o quitarse estos dispositivos, esta desventaja impidió su aplicación en un uso práctico.

Ante la problemática de los exoesqueletos, en 1993 Iwata [11] propuso una alternativa que consistía en un dispositivo basado en la manipulación de una herramienta

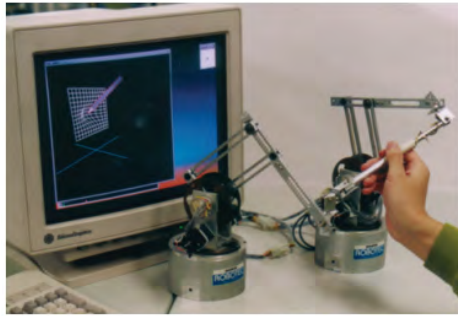


Figura 1.8: Interfaz háptica con forma de bolígrafo [11]



Figura 1.9: Robot *Omni Phantom*

con la forma de bolígrafo. El dispositivo contaba con seis grados de libertad, la realimentación de fuerza se lograba con el uso de dos robots manipuladores de tres grados de libertad como se ilustra en la Figura 1.8. Posteriormente se presentaron diseños similares al propuesto por Iwata; en 1994 investigadores del Massachusetts Institute of Technology presentaron el dispositivo *PHANToM* [23], se trata de un robot que tiene tres grados de libertad con realimentación de fuerza y tres más sin ella, sumando en total seis grados de libertad. El gran auge de los videojuegos y la realidad virtual permitió que se lanzaran al mercado de consumo dispositivos hápticos como el robot *Omni Phantom* de *Sensable* con un diseño similar al *PHANToM* del MIT, ilustrado en la Figura 1.9. Más adelante en el año 2007 sale al mercado el robot *Novint Falcon* el primer dispositivo de su tipo por tratarse de un robot paralelo tipo delta.

Desde que el robot *Novint Falcon* fue lanzado al mercado ha sido reportado por diversos autores como Panarese [19] y Yang [28], los dos orientados a su aplicación como dispositivo háptico. La literatura acerca del análisis cinemático y dinámico del robot *Novint Falcon* es escasa, siendo el trabajo de Michelotti [15] el más relevante aunque sólo reporta el análisis cinemático, dejando a un lado el dinámico.

1.4. Contribuciones

Las principales contribuciones de esta tesis son:

- La puesta en funcionamiento del dispositivo háptico *Novint Falcon*, y los programas necesarios para la implementación de los esquemas de control propuestos.
- La obtención de las ecuaciones que describen el comportamiento cinemático y dinámico del robot *Novint Falcon*.
- La implementación de un sistema de realidad virtual con interacción háptica mediante el robot *Novint Falcon*.

1.5. Organización de la tesis

La estructura de este trabajo se agrupa en cuatro partes, la primera contenida en el Capítulo 1, donde se presenta la problemática a resolver y se introduce al tema. Los Capítulos 2, 3 y 4 conforman la puesta en funcionamiento del robot *Novint Falcon*, comenzando en el Capítulo 2 donde se realiza el análisis cinemático y dinámico, continuando en el Capítulo 3 donde se proponen los esquemas de control y finalizando con su implementación en el Capítulo 4. El Capítulo 5 contiene la tercera parte, la cual expone la interacción del robot *Novint Falcon* con un ambiente virtual. Finalmente se presentan las conclusiones del trabajo.

Capítulo 2

Análisis cinemático y dinámico

En este capítulo se realiza el análisis cinemático y dinámico del robot Novint Falcon con el fin de obtener un modelo matemático que reproduzca fielmente su dinámica. Posteriormente se empleará en la implementación de dos esquemas de control.

2.1. Geometría del robot *Novint Falcon*

Generalmente el análisis de robots de cadena cinemática abierta comienza con la aplicación de la convención de Denavit-Hartenberg. En los robots de cadena cerrada la aplicación de dicha convención no siempre resulta conveniente ya que al tener más de una cadena cinemática la solución se torna complicada. Normalmente para este tipo de robots se realiza un análisis geométrico previo para cada extremidad, lo que a continuación se presenta.

La posición del efector final descrita por el punto P se obtiene mediante alguna API. Una API (*Application Programming Interface*) es un conjunto de rutinas y procedimientos que ofrecen una interfaz para interactuar con el robot, misma que proporciona la posición del efector final en coordenadas del sistema de referencia háptico $H(x, y, z)$. Su origen se encuentra a $151[mm]$ sobre el eje w del sistema de referencia base $B(u, v, w)$, cuya ubicación está en el centro de la plataforma fija tal como se observa en la Figura 2.1. Lo anterior implica una transformación de coordenadas para pasar del sistema háptico H al sistema base B .

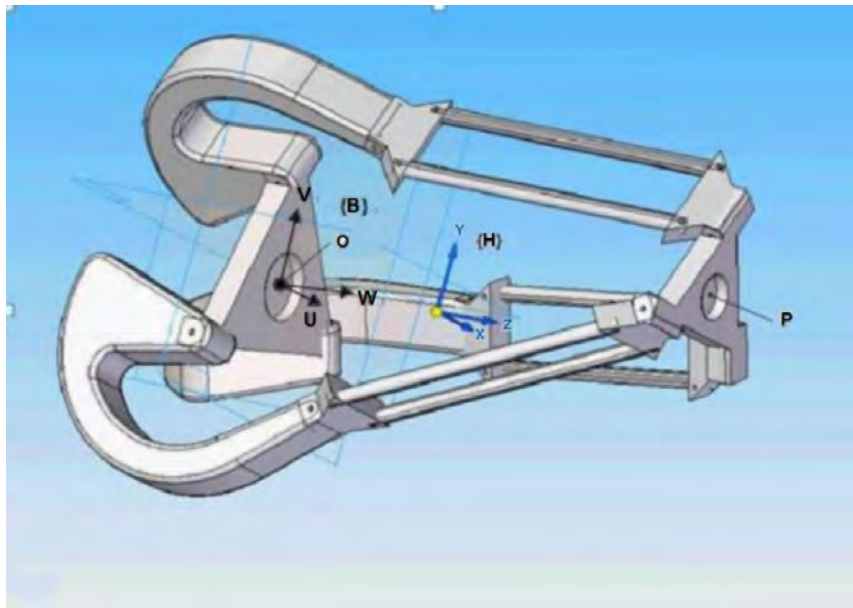


Figura 2.1: Sistemas de Referencia Base y Háptico [28]

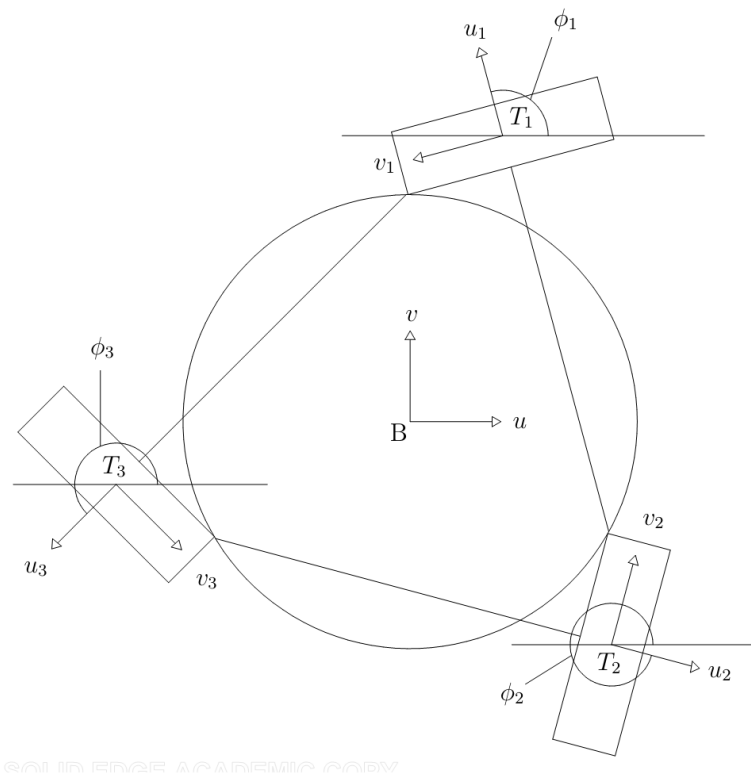


Figura 2.2: Sistemas de Referencia Base B y de cada extremidad T_i

Dada la posición del efector final P en el sistema de referencia háptico y asumiendo que el origen de éste se encuentra en la posición (x_0, y_0, z_0) del sistema de referencia base, es posible realizar la transformación de coordenadas del punto $P(p_x, p_y, p_z)$ al sistema de referencia base mediante:

$$\begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (2.1)$$

donde $(x_0, y_0, z_0) = (0, 0, 151)[mm]$.

Se requiere del uso de tres sistemas de referencia más: $T_i(u_i, v_i, w_i)$ para $i = 1, 2, 3$, los cuales se encuentran en cada articulación actuada de la base, con cada eje u_i perpendicular al eje de la articulación y a un ángulo ϕ_i del eje u del sistema base, como se ilustra en la Figura 2.2.

Las transformaciones del sistema de referencia base B a los sistemas de referencia T_i son similares a las presentadas por Stamper [26], con ligeras diferencias debidas a la existencia de una desviación s en la base y otra más en la plataforma móvil. Éstas se obtienen usando la siguiente relación:

$$\begin{bmatrix} p_{ui} \\ p_{vi} \\ p_{wi} \end{bmatrix} = \begin{bmatrix} \cos(\phi_i) & \sin(\phi_i) & 0 \\ -\sin(\phi_i) & \cos(\phi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix} + \begin{bmatrix} -r \\ s \\ 0 \end{bmatrix} \quad i = 1, 2, 3 \quad (2.2)$$

donde $(\phi_1, \phi_2, \phi_3) = (105, 345, 225)[^\circ]$ debido a la geometría del robot.

Lo siguiente es obtener las expresiones para p_{ui} , p_{vi} y p_{wi} en función de los ángulos de cada articulación a partir de las relaciones geométricas presentadas en el diagrama de la Figura 2.3. De esta forma se obtiene:

$$p_{ui} = a \cos(\theta_{1i}) - c + [d + e + b \sin(\theta_{3i})] \cos(\theta_{2i}) \quad (2.3)$$

$$p_{vi} = b \cos(\theta_{3i}) + f \quad (2.4)$$

$$p_{wi} = a \sin(\theta_{1i}) + [d + e + b \sin(\theta_{3i})] \sin(\theta_{2i}) \quad (2.5)$$

para $i = 1, 2, 3$. El valor de las longitudes $a - f$ se muestran en la Tabla 2.1.

Segmento	a	b	c	d	e	f	r	s
Longitud en mm	60	102.5	15.7	11.5	11.5	26.2	36.3	27.2

Tabla 2.1: Tabla de longitudes geométricas.

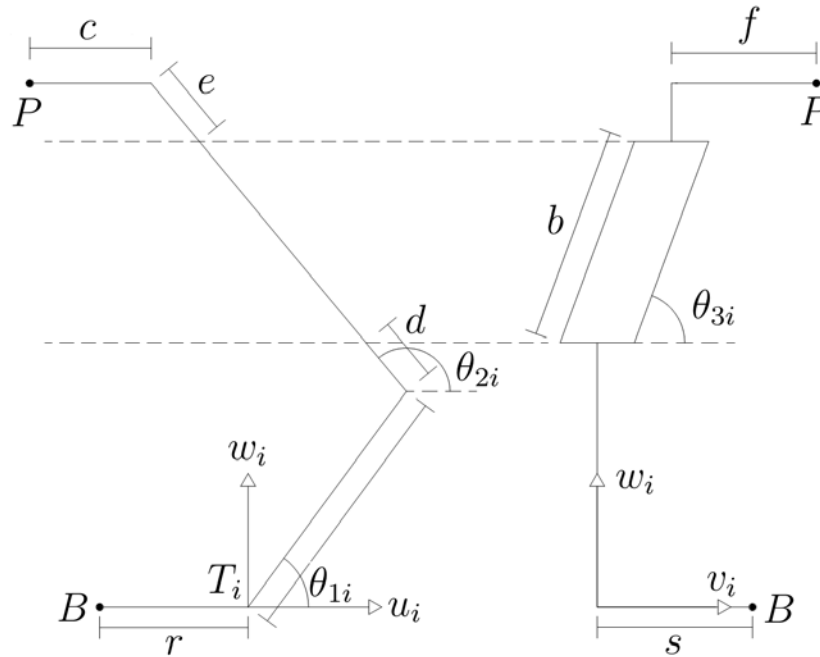


Figura 2.3: Diagrama geométrico de la i -ésima extremidad.

2.2. Cinemática Directa

El objetivo de la cinemática directa es realizar el mapeo desde variables articulares hasta la posición del efector final en coordenadas cartesianas. En este caso, dados los ángulos conocidos de las articulaciones en la base θ_{11} , θ_{21} y θ_{31} , se calcula la posición del efector final $P(x, y, z)$. A diferencia de los robots en serie, obtener la cinemática directa de los robots paralelos es más complicado que obtener la cinemática inversa.

Partiendo de las ecuaciones (2.3), (2.4) y (2.5) obtenidas en la sección 2.1 que describen las relaciones geométricas del punto P para cada extremidad en el sistema de coordenadas $T_i(u_i, v_i, w_i)$ para $i = 1, 2, 3$, es necesario pasar al sistema de referencia

háptico $H(x, y, z)$ substituyendo p_{ui} , p_{vi} y p_{wi} en la ecuación (2.2) y posteriormente en la (2.1). Lo anterior da como resultado un sistema de nueve ecuaciones con nueve incógnitas:

$$p_x \cos(\phi_i) + p_y \sin(\phi_i) - a \cos(\theta_{1i}) - [d + e + b \sin(\theta_{3i})] \cos(\theta_{2i}) + c - r = 0 \quad (2.6)$$

$$p_y \cos(\phi_i) - p_x \sin(\phi_i) - b \cos(\theta_{3i}) + s - f = 0 \quad (2.7)$$

$$p_z + z_0 - a \sin(\theta_{1i}) - [d + e + b \sin(\theta_{3i})] \sin(\theta_{2i}) = 0 \quad (2.8)$$

para $i = 1, 2$ y 3 . La solución a este conjunto de ecuaciones resuelve la cinemática directa. Para determinar dicha solución las ecuaciones son manipuladas para producir dos ecuaciones en función del tangente del ángulo medio de θ_{31} y θ_{32} , las dos ecuaciones resultantes se pueden resolver mediante tres diversas aproximaciones algebraicas como propone Ragahavan [22]: eliminación dialítica de Sylvester, continuación polinómica y bases de Grobner. La solución por medio de eliminación dialítica de Sylvester puede ser consultada a detalle en el trabajo de Stamper [26].

La API del robot *Novint Falcon* proporciona directamente la posición del efector final a partir del sistema de referencia háptico, por lo que en este trabajo no fue necesario resolver la cinemática directa del robot.

2.3. Cinemática Inversa

La cinemática inversa estudia el mapeo de coordenadas cartesianas a variables articulares. En este caso, dada la posición del efector final P se calcula un conjunto de tres ángulos (θ_{i1} , θ_{i2} y θ_{i3}) por cada uno de los brazos $i = 1, 2, 3$.

Existen dos soluciones para θ_{3i} a partir de la ecuación (2.4):

$$\theta_{3i} = \pm \arccos\left(\frac{p_{vi} - f}{b}\right). \quad (2.9)$$

Debido a su geometría el robot *Novint Falcon* sólo puede presentar valores positivos de θ_{3i} . El cuadrado de las ecuaciones (2.3) y (2.5) pueden sumarse entre sí para cancelar θ_{2i} de acuerdo al Teorema de Pitágoras, resultando lo siguiente:

$$(p_{ui} + c)^2 + p_{wi}^2 + a^2 - 2a(p_{ui} + c) \cos(\theta_{1i}) - 2ap_{wi} \sin(\theta_{1i}) = (d + e)^2 + 2(e + d)b \sin(\theta_{3i}) + b^2 \sin^2(\theta_{3i}). \quad (2.10)$$

Para encontrar una solución a la ecuación (2.10), ésta se lleva a una forma polinomial definiendo la siguiente variable:

$$\alpha_i = \tan\left(\frac{\theta_{1i}}{2}\right), \quad (2.11)$$

considerando las siguientes relaciones trigonométricas:

$$\sin(\theta_{1i}) = \frac{2\alpha_i}{1 + \alpha_i^2}, \quad (2.12)$$

$$\cos(\theta_{1i}) = \frac{1 - \alpha_i^2}{1 + \alpha_i^2} \quad (2.13)$$

y substituyéndolas en la ecuación (2.10) se obtiene:

$$A_{2i}\alpha_i^2 + A_{1i}\alpha_i + A_{0i} = 0 \quad (2.14)$$

donde

$$A_{0i} = (a - c)^2 - (d + e)^2 + p_{wi}^2 + p_{ui}^2 + 2p_{ui}(c - a) - 2b \sin(\theta_{3i})(d + e) - b^2 \sin^2(\theta_{3i}),$$

$$A_{1i} = -4ap_{wi} \quad \text{y}$$

$$A_{2i} = (a + c)^2 - (d + e)^2 + p_{wi}^2 + p_{ui}^2 + 2p_{ui}(c + a) - 2b \sin(\theta_{3i})(d + e) - b^2 \sin^2(\theta_{3i}).$$

Existen dos posibles soluciones para α_i en la ecuación (2.14) dadas por:

$$\alpha_i = \frac{-A_{1i} \pm \sqrt{A_{1i}^2 - 4A_{0i}A_{2i}}}{2A_{2i}}. \quad (2.15)$$

Debido a la geometría del robot Falcon, sólo es válida una solución:

$$\alpha_i = \frac{-A_{1i} - \sqrt{A_{1i}^2 - 4A_{0i}A_{2i}}}{2A_{2i}}. \quad (2.16)$$

Al conocer el valor de α_i se obtiene el de θ_{1i} :

$$\theta_{1i} = 2 \arctan(\alpha_i). \quad (2.17)$$

Para finalizar se obtiene θ_{2i} a partir de la ecuación (2.3):

$$\theta_{2i} = \arccos \left(\frac{p_{ui} - a \cos(\theta_{1i}) + c}{d + e + b \sin(\theta_{3i})} \right). \quad (2.18)$$

2.4. Matriz Jacobiana

En un robot paralelo no todas las articulaciones pueden controlarse independientemente debido a su estructura de cadena cerrada, de esta forma algunas articulaciones están actuadas mientras otras no. El robot *Novint Falcon* presenta tres articulaciones actuadas y todas se encuentran en la base (θ_{11} , θ_{12} y θ_{13}), por lo que estos tres ángulos forman el vector \mathbf{q} .

La matriz Jacobiana de un robot paralelo representa la transformación de la velocidad del efector final $\dot{\mathbf{x}}$ a la velocidad de las articulaciones actuadas $\dot{\mathbf{q}}$:

$$\dot{\mathbf{q}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{x}} \quad (2.19)$$

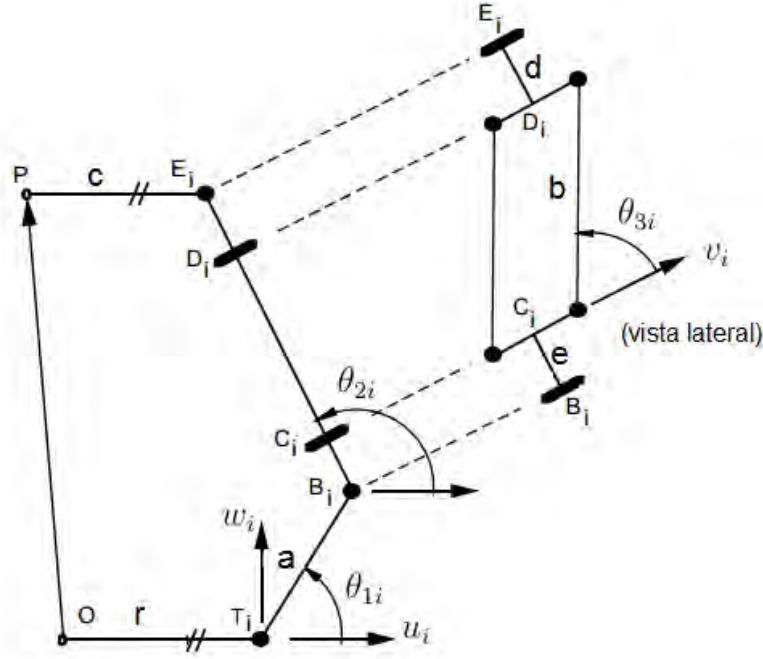
en la cual $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{n \times m}$ es la matriz Jacobiana, donde n es la dimensión del vector $\dot{\mathbf{x}}$ y m la dimensión del vector $\dot{\mathbf{q}}$. Puede notarse que la matriz Jacobiana de un robot paralelo es inversa a la de un robot en serie donde la transformación va de la velocidad de las articulaciones a la velocidad del efector final.

Las restricciones cinemáticas impuestas por las extremidades pueden ser escritas de forma general:

$$F(\mathbf{q}, \mathbf{x}) = 0, \quad (2.20)$$

donde F es una función implícita n -dimensional de \mathbf{q} y \mathbf{x} . Diferenciando la ecuación (2.20) con respecto al tiempo, resulta en la siguiente relación:

$$\mathbf{J}_x(\mathbf{q})\dot{\mathbf{x}} = \mathbf{J}_q(\mathbf{q})\dot{\mathbf{q}} \quad (2.21)$$


 Figura 2.4: Diagrama de la i -ésima extremidad.

donde

$$\mathbf{J}_x(\mathbf{q}) = \frac{\partial F}{\partial \mathbf{x}} \quad y \quad \mathbf{J}_q(\mathbf{q}) = -\frac{\partial F}{\partial \mathbf{q}}.$$

El primer paso para obtener las matrices Jacobianas de la ecuación (2.21), es escribir la relación de la cadena cinemática cerrada para la i -ésima extremidad en coordenadas del sistema de referencia $T_i(u_i, v_i, w_i)$ para $i = 1, 2, 3$ con base en el diagrama de la Figura 2.4:

$$\overline{OT_i} + \overline{TB_i} = \overline{OP_i} + \overline{PE_i} + \overline{ED_i} + \overline{DC_i} + \overline{CB_i}. \quad (2.22)$$

Derivando la ecuación (2.22) con respecto al tiempo y expresando el resultado en coordenadas del sistema de referencia T_i se tiene:

$$\overline{w_{1i}} \times \overline{TB_i} = \dot{P}_{T_i} + \overline{\omega_{3i}} \times \overline{DC_i} + \overline{\omega_{2i}} \times (\overline{ED_i} + \overline{CB_i}) \quad (2.23)$$

donde $\dot{P}_{T_i} = [\dot{p}_{ui} \quad \dot{p}_{vi} \quad \dot{p}_{wi}]^T$ y ω_{ni} es la velocidad angular del n -ésimo eslabón de la

i -ésima extremidad definida como:

$$\overline{\omega}_{1i} = \begin{bmatrix} 0 \\ -\dot{\theta}_{1i} \\ 0 \end{bmatrix}, \quad \overline{\omega}_{2i} = \begin{bmatrix} 0 \\ -\dot{\theta}_{2i} \\ 0 \end{bmatrix} \quad \text{y} \quad \overline{\omega}_{3i} = \begin{bmatrix} \dot{\theta}_{3i} \sin(\theta_{2i}) \\ -\dot{\theta}_{2i} \\ -\dot{\theta}_{3i} \cos(\theta_{2i}) \end{bmatrix}. \quad (2.24)$$

Escribiendo la ecuación (2.23) en términos de los parámetros de eslabones y de ángulos de articulaciones, se tiene:

$$\begin{bmatrix} -a\dot{\theta}_{1i} \sin(\theta_{1i}) \\ 0 \\ a\dot{\theta}_{1i} \cos(\theta_{1i}) \end{bmatrix} = \begin{bmatrix} \dot{p}_{ui} - b\dot{\theta}_{3i} \cos(\theta_{2i}) \cos(\theta_{3i}) + \dot{\theta}_{2i} \sin(\theta_{2i})[d + e + b \sin(\theta_{3i})] \\ \dot{p}_{vi} + b\dot{\theta}_{3i} \sin(\theta_{3i}) \\ \dot{p}_{wi} - b\dot{\theta}_{3i} \sin(\theta_{2i}) \cos(\theta_{3i}) - \dot{\theta}_{2i} \cos(\theta_{2i})[d + e + b \sin(\theta_{3i})] \end{bmatrix}. \quad (2.25)$$

Considerando la ecuación (2.25) como un sistema de tres ecuaciones con tres incógnitas $(\theta_{1i}, \theta_{2i}, \theta_{3i})$ y resolviendo para $\dot{\theta}_{1i}$ se obtiene:

$$\dot{\theta}_{1i} = \frac{\dot{p}_{ui} \cos(\theta_{2i}) + \dot{p}_{vi} \cot(\theta_{3i}) + \dot{p}_{wi} \sin(\theta_{2i})}{a \sin(\theta_{2i} - \theta_{1i})}. \quad (2.26)$$

La ecuación (2.26) es transformada desde el sistema de referencia $T_i(u_i, v_i, w_i)$ de cada extremidad $i = 1, 2, 3$ hacia el sistema de referencia base $B(u, v, w)$ usando la relación:

$$\begin{bmatrix} \dot{p}_{ui} \\ \dot{p}_{vi} \\ \dot{p}_{wi} \end{bmatrix} = \begin{bmatrix} \cos(\phi_i) & \sin(\phi_i) & 0 \\ -\sin(\phi_i) & \cos(\phi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{p}_u \\ \dot{p}_v \\ \dot{p}_w \end{bmatrix}. \quad (2.27)$$

Las ecuaciones resultantes se reordenan para obtener las matrices Jacobianas:

$$\mathbf{J}_x \begin{bmatrix} \dot{p}_u \\ \dot{p}_v \\ \dot{p}_w \end{bmatrix} = \mathbf{J}_q \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} \quad (2.28)$$

donde

$$\mathbf{J}_x = \begin{bmatrix} J_{x11} & J_{x12} & J_{x13} \\ J_{x21} & J_{x22} & J_{x23} \\ J_{x31} & J_{x32} & J_{x33} \end{bmatrix}, \quad \mathbf{J}_q = \begin{bmatrix} J_{q1} & 0 & 0 \\ 0 & J_{q2} & 0 \\ 0 & 0 & J_{q3} \end{bmatrix},$$

$$\begin{aligned}
J_{x_{i1}} &= \cos(\theta_{2i}) \sin(\theta_{3i}) \cos(\phi_i) - \cos(\theta_{3i}) \sin(\phi_i), \\
J_{x_{i2}} &= \cos(\theta_{3i}) \cos(\phi_i) + \cos(\theta_{2i}) \sin(\theta_{3i}) \sin(\phi_i), \\
J_{x_{i3}} &= \sin(\theta_{2i}) \sin(\theta_{3i}) \quad \text{y} \\
J_{q_i} &= a \sin(\theta_{2i} - \theta_{1i}) \sin(\theta_{3i})
\end{aligned}$$

para $i = 1, 2$ y 3 .

Finalmente se forma la matriz Jacobiana general $\mathbf{J}(\mathbf{q})$ de la ecuación (2.19), mediante la siguiente expresión:

$$\mathbf{J}(\mathbf{q}) = \mathbf{J}_q^{-1} \mathbf{J}_x. \quad (2.29)$$

Debido a su naturaleza la matriz Jacobiana $\mathbf{J}(\mathbf{q})$ no existe cuando se presenta una singularidad en la cinemática inversa, y de igual forma su inversa $\mathbf{J}^{-1}(\mathbf{q})$ no existe en presencia de una singularidad en la cinemática directa.

2.4.1. Singularidades en Cinemática Directa

Una singularidad en la cinemática directa ocurre cuando el determinante de \mathbf{J}_x es igual a cero,

$$\det(\mathbf{J}_x) = 0. \quad (2.30)$$

Cuando \mathbf{J}_x es singular y el espacio nulo de \mathbf{J}_x es diferente de cero, existen algunos vectores de $\dot{\mathbf{x}}$ que resultan en vectores nulos de $\dot{\mathbf{q}}$. Consecuentemente el robot gana uno o más grados de libertad, esto es que la plataforma sigue teniendo movimiento aún cuando los actuadores se encuentran inmóviles. Esto demuestra que existen fuerzas aplicadas al efector final que los actuadores no pueden resistir cuando el robot se encuentra en esta posición singular. Esta singularidad generalmente ocurre cuando diferentes soluciones de la cinemática directa convergen.

2.4.2. Singularidades en Cinemática Inversa

Una singularidad en la cinemática inversa ocurre cuando el determinante de \mathbf{J}_q toma el valor de cero,

$$\det(\mathbf{J}_q) = 0. \quad (2.31)$$

Cuando \mathbf{J}_q es singular y el espacio nulo de \mathbf{J}_q es diferente de cero, existen algunos vectores de $\dot{\mathbf{q}}$ que resultan en vectores nulos de $\dot{\mathbf{x}}$. Esto causa que el robot pierda uno o más grados de libertad, debido a que movimientos en ciertas direcciones no pueden ser realizados. La singularidad generalmente ocurre en los límites del espacio de trabajo, donde diferentes soluciones de la cinemática inversa convergen.

2.5. Modelo Dinámico

Diversas aproximaciones han sido utilizadas para caracterizar la dinámica de robots paralelos, dentro de las más comunes se encuentra el basado en la formulación de Lagrange [18], la aplicación del principio de Hamilton [17] y empleando directamente las ecuaciones de movimiento de Newton [7]. En este trabajo se realiza la aproximación basada en las ecuaciones de movimiento de Newton-Euler debido a que requiere un menor trabajo computacional que sus homónimas. Cabe señalar que el tiempo en que se realiza el cálculo del momento aplicado a cada actuador debe ser menor al tiempo de muestreo del robot (1 milisegundo).

En este modelo dinámico se desprecia la fricción en las articulaciones no actuadas. Se asume que la masa m_b de cada una de las bielas ubicadas en los puntos C_i y D_i de la Figura 2.4 son iguales entre sí y se concentran en las articulaciones B_i y E_i . Se desprecia la masa de las barras del paralelogramo, esto no afecta la fidelidad del modelo ya que la dinámica de las articulaciones recrea parte del comportamiento de las barras.

La primer ecuación de movimiento es obtenida mediante la suma de momentos aplicados a la articulación actuada de la i -ésima extremidad:

$$M_{AI_i} = I_A \ddot{\theta}_{1i} + \tilde{\tau}_i \quad i = 1, 2, 3 \quad (2.32)$$

donde M_{AI_i} es la suma de los momentos en la articulación T_i causados por la inercia rotacional y traslacional del robot, I_A es el momento de inercia debido a el eslabón de entrada, el rotor del motor, y una de las dos bielas del paralelogramo, por último

$\tilde{\tau}_i$ es el i -ésimo elemento de una matriz de cargas inerciales debido a la plataforma móvil y la segunda biela del paralelogramo de cada extremidad. Las relaciones para I_A y $\tilde{\boldsymbol{\tau}}$ son las siguientes:

$$I_A = I_r + \frac{1}{3}m_a a^2 + m_b a^2 \quad (2.33)$$

donde I_r es el momento de inercia del rotor, m_a es la masa del eslabón de entrada, m_b es la masa de una de las bielas del paralelogramo y a es la longitud del eslabón de entrada;

$$\tilde{\boldsymbol{\tau}} = (\mathbf{J}^T)^{-1}(3m_b + m_c)\ddot{\mathbf{P}}_B \quad (2.34)$$

donde m_c es la masa de la plataforma móvil, \mathbf{J} es la matriz Jacobiana y $\ddot{\mathbf{P}}_B$ es la aceleración de la plataforma móvil con respecto al sistema de referencia base.

La segunda ecuación de movimiento es obtenida al sumar los momentos aplicados en la articulación A_i causados por los motores, la fuerza gravitacional y la fricción viscosa:

$$\begin{aligned} \begin{bmatrix} M_{AT_1} \\ M_{AT_2} \\ M_{AT_3} \end{bmatrix} &= \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} + \frac{1}{2}am_a g \begin{bmatrix} \sin(\phi_1) \sin(\theta_{11}) \\ \sin(\phi_2) \sin(\theta_{12}) \\ \sin(\phi_3) \sin(\theta_{13}) \end{bmatrix} + am_b g \begin{bmatrix} \sin(\phi_1) \sin(\theta_{11}) \\ \sin(\phi_2) \sin(\theta_{12}) \\ \sin(\phi_3) \sin(\theta_{13}) \end{bmatrix} \\ &+ (\mathbf{J}^T)^{-1}m \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} - b_A \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} \end{aligned} \quad (2.35)$$

donde τ_i es el momento aplicado por el actuador de la i -ésima extremidad, $m = 3m_b + m_c$, y b_A es el coeficiente de fricción viscosa.

De acuerdo a la Segunda Ley de Newton los momentos en la articulación A_i se relacionan de la siguiente forma:

$$\begin{bmatrix} M_{AT_1} \\ M_{AT_2} \\ M_{AT_3} \end{bmatrix} = \begin{bmatrix} M_{AI_1} \\ M_{AI_2} \\ M_{AI_3} \end{bmatrix}. \quad (2.36)$$

Substituyendo las ecuaciones (2.32) y (2.35) en la ecuación (2.36) y resolviendo para los momentos aplicados por los actuadores se tiene:

$$\begin{aligned}
\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} &= -ag\left(\frac{1}{2}m_a + m_b\right) \begin{bmatrix} \sin(\phi_1) \sin(\theta_{11}) \\ \sin(\phi_2) \sin(\theta_{12}) \\ \sin(\phi_3) \sin(\theta_{13}) \end{bmatrix} - (\mathbf{J}^T)^{-1}m \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} + b_A \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} \\
&+ I_A \begin{bmatrix} \ddot{\theta}_{11} \\ \ddot{\theta}_{12} \\ \ddot{\theta}_{13} \end{bmatrix} + (\mathbf{J}^T)^{-1}m\ddot{\mathbf{P}}_B.
\end{aligned} \tag{2.37}$$

Se puede encontrar una relación entre la aceleración de la plataforma móvil y la aceleración angular derivando la matriz Jacobiana de la ecuación (2.19):

$$\ddot{\mathbf{P}}_B = \mathbf{J}^{-1} \begin{bmatrix} \ddot{\theta}_{11} \\ \ddot{\theta}_{12} \\ \ddot{\theta}_{13} \end{bmatrix} + \frac{d}{dt}(\mathbf{J}^{-1}) \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix}. \tag{2.38}$$

Substituyendo la ecuación (2.38) en la ecuación (2.37) y agrupando los términos se tiene:

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{2.39}$$

donde

$$\mathbf{q} = \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix},$$

$$\mathbf{H}(\mathbf{q}) = I_A \mathbf{I} + (\mathbf{J}^T)^{-1}m\mathbf{J}^{-1},$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = (\mathbf{J}^T)^{-1}m\frac{d}{dt}(\mathbf{J}^{-1}),$$

$$\mathbf{D} = b_A \mathbf{I} \quad \text{y}$$

$$\mathbf{g}(\mathbf{q}) = -ag\left(\frac{1}{2}m_a + m_b\right) \begin{bmatrix} \sin(\phi_1) \sin(\theta_{11}) \\ \sin(\phi_2) \sin(\theta_{12}) \\ \sin(\phi_3) \sin(\theta_{13}) \end{bmatrix} - (\mathbf{J}^T)^{-1}m \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix}.$$

La ecuación (2.39) describe la dinámica del robot *Novint Falcon* donde \mathbf{q} es el vector de variables articulares, $\boldsymbol{\tau}$ se conforma por los momentos de fuerza aplicados en cada articulación, $\mathbf{H}(\mathbf{q})$ es la matriz de inercia y presenta la característica de ser

simétrica positiva, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ es la matriz de Coriolis, \mathbf{D} es una matriz en la cual su diagonal principal contiene los coeficientes de fricción, y por último $\mathbf{g}(\mathbf{q})$ es el vector de gravedad.

2.5.1. Regresor de parámetros

El modelo dinámico descrito en la ecuación (2.39) puede ser representado como el producto entre una matriz \mathbf{Y} conocida como regresor y el vector de parámetros $\boldsymbol{\theta}$, de la siguiente forma:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\theta} \quad (2.40)$$

donde $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ es conocida ya que depende de las señales medibles del sistema, y puede ser expresada de la siguiente manera:

$$\mathbf{Y} = [\mathbf{Y}_H + \mathbf{Y}_C \quad \mathbf{Y}_D \quad \mathbf{Y}_g] \quad (2.41)$$

donde

$$\begin{aligned} \mathbf{Y}_H &= [\ddot{\mathbf{q}} \quad (\mathbf{J}^T)^{-1}\mathbf{J}^{-1}\ddot{\mathbf{q}}], \\ \mathbf{Y}_C &= \left[\mathbf{0} \quad (\mathbf{J}^T)^{-1}\frac{d}{dt}(\mathbf{J}^{-1})\dot{\mathbf{q}} \right], \\ \mathbf{Y}_D &= [\dot{\mathbf{q}}] \quad \text{y} \\ \mathbf{Y}_g &= \begin{bmatrix} -\sin(\phi_1)\sin(\theta_{11}) \\ -\sin(\phi_2)\sin(\theta_{12}) \\ -\sin(\phi_3)\sin(\theta_{13}) \end{bmatrix} (\mathbf{J}^T)^{-1} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}. \end{aligned}$$

La suma entre las matrices \mathbf{Y}_H y \mathbf{Y}_C se debe a que la matriz de inercia y la matriz de Coriolis comparten parámetros. Por último el vector de parámetros es el siguiente:

$$\boldsymbol{\theta} = \begin{pmatrix} I_A \\ m \\ b_A \\ ag(\frac{1}{2}m_a + m_b) \\ mg \end{pmatrix}. \quad (2.42)$$

Se puede observar que las matrices $\mathbf{H}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, \mathbf{D} y $\mathbf{g}(\mathbf{q})$ son linealmente dependientes de $\boldsymbol{\theta}$. La representación del modelo del sistema en esta forma es útil en presencia de variación paramétrica.

Capítulo 3

Control del robot *Novint Falcon*

En este capítulo se proponen dos controladores para seguimiento de trayectoria con el fin de ser aplicados al robot *Novint Falcon* y posteriormente comparar su desempeño.

El primero es un controlador Proporcional-Integral-Derivativo o PID, el cual se aplica a sistemas de una sola entrada y una sola salida (SISO por sus siglas en inglés) por lo que se considera a cada actuador como independiente, ignorando el acoplamiento y la naturaleza no lineal del robot. La implementación de este tipo de controlador conlleva un desempeño razonablemente satisfactorio cuando la dinámica del robot es dominada por la inercia interna de los motores, esto se presenta con una relación de transmisión alta entre el motor y el eslabón actuado.

El segundo es un controlador adaptable el cual se elige debido a que algunos de los parámetros del robot *Novint Falcon* no son especificados por el fabricante y son difíciles de medir, además de que es robusto ante variaciones paramétricas. Este tipo de controlador requiere de un modelo dinámico del sistema ya que a diferencia del PID, toma en cuenta el acoplamiento y la no linealidad del robot. Por consiguiente tiene el potencial para obtener un mejor desempeño. Sin embargo una de sus desventajas es un mayor tiempo de procesamiento, por lo que resulta inconveniente implementarse en procesadores de baja velocidad.

En la Figura 3.1 se presenta un esquema de control genérico, se pueden identificar cuatro señales: la señal de salida $y(t)$ que representa el estado actual de la planta, la señal de referencia $y_d(t)$ que define el estado deseado, la señal de error $e(t)$ la cual se

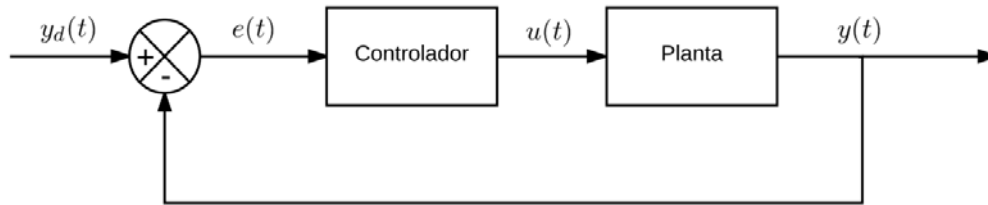


Figura 3.1: Esquema genérico de control

define como la diferencia entre el estado actual y el estado deseado

$$e(t) = y_d(t) - y(t),$$

por último la señal de control $u(t)$ se aplica directamente a la planta con el fin de que el error tienda a cero de forma asintótica y en tiempo finito.

3.1. Control PID

El control PID es el esquema de control más usado en la industria debido a su fácil implementación en un gran rango de aplicaciones. Se compone de la suma de tres términos. El primer término es proporcional al error, y se denomina control proporcional (P). El segundo término es proporcional a la integral del error, depende de los errores pasados y es denominado control integral (I). Por último se tiene un término proporcional a la derivada del error, el cual es una predicción de los errores futuros y se denomina control derivativo (D).

En el caso del robot *Novint Falcon* se considera el control para cada articulación dado que el controlador PID es aplicable a sistemas SISO y el robot cuenta con tres grados de libertad por lo se requieren tres lazos de control independientes, uno por cada actuador. La posición de referencia $q_d(t)$ de cada articulación actuada es determinada por las trayectorias propuestas.

El controlador PID recibe una señal de error $e(t)$ y genera una señal de control $u(t)$, definida por:

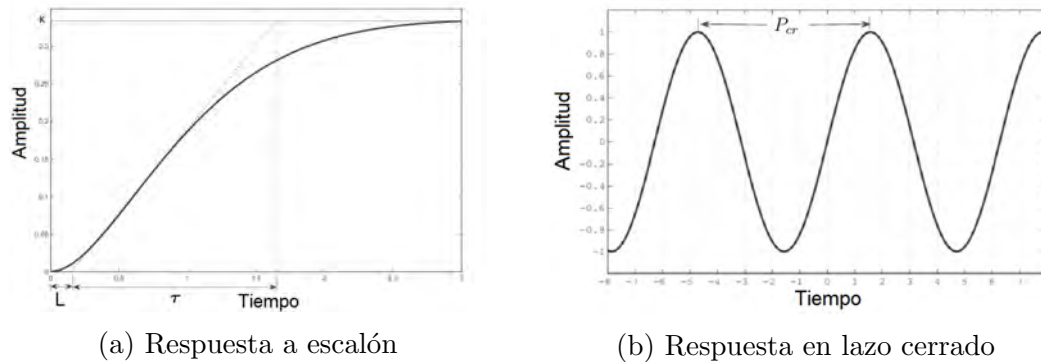


Figura 3.2: Tipos de respuesta

$$u(t) = k_p \left(e(t) + \frac{1}{h_i} \int_{-\infty}^t e(t) dt + h_d \frac{de(t)}{dt} \right) \quad (3.1)$$

donde k_p es la ganancia proporcional, h_i es el tiempo integral y h_d es el tiempo derivativo.

La clave para obtener un mejor desempeño en este tipo de controlador es seleccionar las ganancias adecuadas que determinan la influencia de cada término en la señal de control.

3.1.1. Sintonización de controlador PID

El proceso de sintonización de un controlador PID consiste en establecer el valor que deben tener los parámetros de ganancia k_p , h_i y h_d para que la señal de error tienda a cero a medida que pasa el tiempo. Los métodos de sintonización se basan en pruebas experimentales en donde se analiza la respuesta del sistema a una señal escalón, es por esto que las ganancias obtenidas pueden no ser definitivas, requiriendo un ajuste fino para obtener un comportamiento adecuado.

En esta tesis se utilizan las reglas de Ziegler y Nichols las cuales, dada su sencillez y disponibilidad son una opción viable. Las reglas definen dos métodos.

El primer método se aplica cuando la respuesta de la planta a una señal escalón tiene el aspecto de una curva en forma de S como se observa en la Figura 3.2(a). El segundo método se aplica a procesos en lazo cerrado con una ganancia crítica que permita una respuesta con oscilación sostenida, como se muestra en la Figura 3.2(b).

Controlador	k_p	h_i	h_d
P	$0.5k_{cr}$	∞	0
PI	$0.45k_{cr}$	$0.5P_{cr}$	0
PID	$0.6k_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Tabla 3.1: Tabla de valores de sintonización.

Ganancia	q_1	q_2	q_3
k_p	3.276	4.602	4.290
h_i [ms]	160	163	175
h_d [ms]	28.1	28.7	31

Tabla 3.2: Tabla de ganancias de controlador PID.

Debido a que la señal de entrada del robot *Novint Falcon* es de fuerza y el recorrido de su efector final se encuentra acotado, no es posible observar una respuesta a escalón por lo que se aplicó el segundo método. En primer lugar se aumentó gradualmente la ganancia hasta obtener una oscilación continua como respuesta, dicha ganancia crítica se denomina k_{cr} . Una vez obtenida una respuesta, se midió el período crítico de dicha oscilación P_{cr} , por último con los valores de k_{cr} y P_{cr} se calcularon los parámetros del controlador PID de acuerdo a la Tabla 3.1 definida en las reglas de Ziegler Nichols. Tras realizar el experimento se obtuvieron las gráficas de la Figura 3.3 donde se puede ver la respuesta con oscilación sostenida de cada una de las articulaciones actuadas del robot.

Las ganancias críticas obtenidas experimentalmente para cada articulación son las siguientes: $k_{cr1} = 4.2$, $k_{cr2} = 5.9$, $k_{cr3} = 5.5$. A partir de las gráficas obtenidas es posible calcular el período crítico para cada articulación: $P_{cr1} = 172[ms]$, $P_{cr2} = 175.6[ms]$, $P_{cr3} = 188[ms]$. Finalmente en la Tabla 3.2 se muestran los valores seleccionados para cada ganancia.

3.2. Control Adaptable

En algunas ocasiones se tiene incertidumbre e incluso variación en los parámetros del sistema a controlar. Para poder lidiar con estos problemas, alrededor de la década

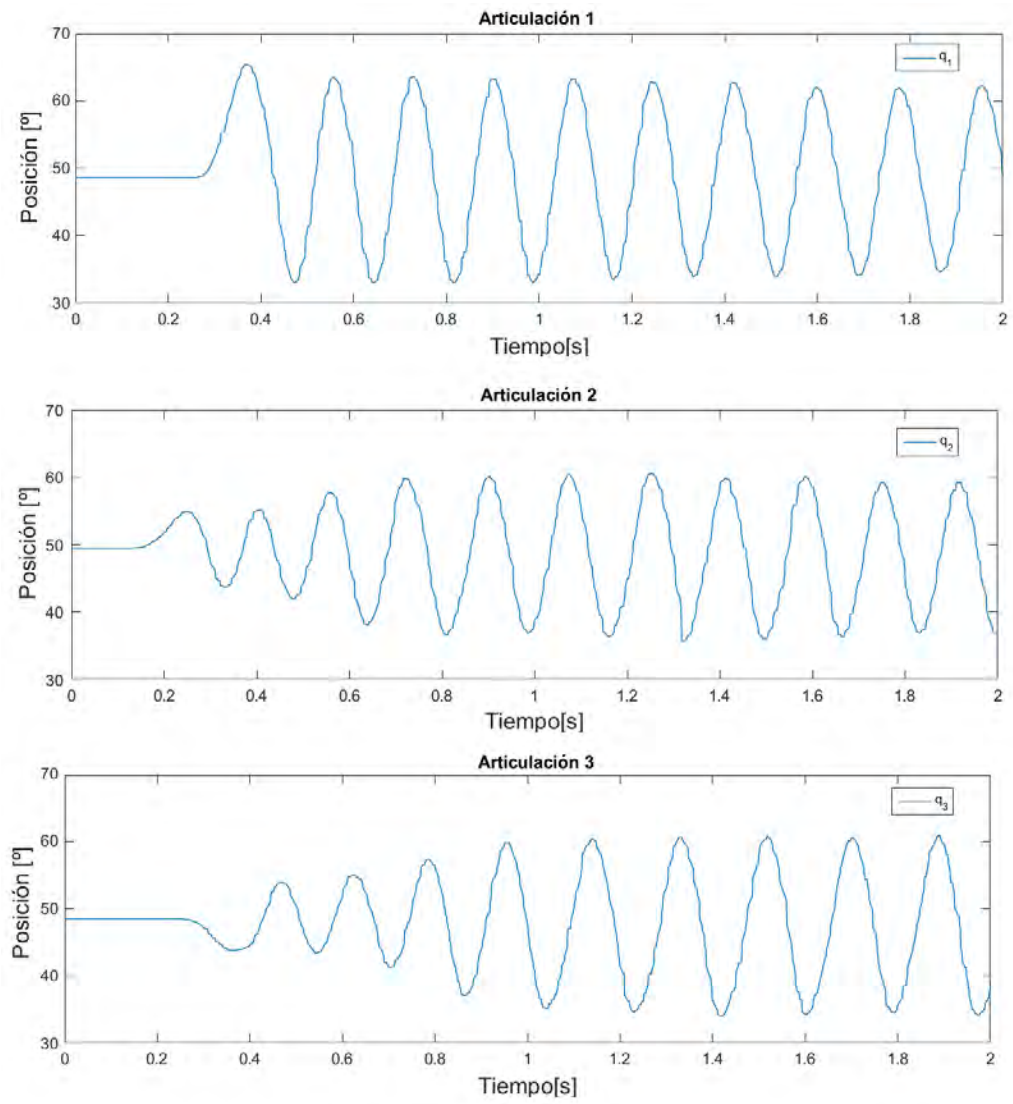


Figura 3.3: Respuesta experimental en lazo cerrado con ganancia k_{cr}

de 1950 se desarrolló el control adaptable el cual realiza una estimación paramétrica en línea basado en las señales de salida del sistema. El control adaptable ha sido usado principalmente en el control de aeronaves ya que al volar a diferentes velocidades y altitudes se presenta variación paramétrica [1]. En la robótica también juega un papel importante ya que usualmente los robots ejecutan tareas donde se requiere levantar objetos de diferente peso y tamaño.

La característica de este tipo de controlador es que no requiere información sobre los parámetros del sistema por lo que deben ser obtenidos a partir de una ley de adaptación. Para su diseño se requieren seguir los siguientes pasos:

1. Elegir una ley de control que contenga los parámetros a estimar.
2. Elegir una ley de adaptación para estimar los parámetros del sistema.
3. Analizar la convergencia del sistema de control resultante.

A continuación se diseña un controlador adaptable de seguimiento de trayectoria para el robot *Novint Falcon* haciendo uso de estos tres pasos. Considerando el modelo de la ecuación (2.39) y definiendo los errores de seguimiento y paramétricos como

$$\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$$

donde \mathbf{q} es el vector de variables articulares y \mathbf{q}_d representa la trayectoria deseada la cual debe ser acotada al igual que su primera y segunda derivada,

$$\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$$

donde $\boldsymbol{\theta}$ es el vector de parámetros y $(\hat{\cdot})$ denota el estimado de (\cdot) . Es necesario establecer una *referencia de velocidad* definida como

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

donde $\boldsymbol{\Lambda}$ es una matriz simétrica definida positiva y un término de *error de velocidad* definido como

$$\mathbf{s} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r.$$

Debido a que la estabilidad depende tanto de la ley de control como de la ley de adaptación, es conveniente coordinar su elección por medio de la siguiente función candidata de Lyapunov [24]:

$$V(t) = \frac{1}{2}[\mathbf{s}^T \mathbf{H} \mathbf{s} + \tilde{\boldsymbol{\theta}}^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\theta}}] \quad (3.2)$$

donde $\boldsymbol{\Gamma}$ es una matriz simétrica definida positiva. Derivando esta última ecuación se tiene:

$$\dot{V}(t) = \mathbf{s}^T (\mathbf{H} \ddot{\mathbf{q}} - \mathbf{H} \ddot{\mathbf{q}}_r) + \frac{1}{2} \mathbf{s}^T \dot{\mathbf{H}} \mathbf{s} + \dot{\tilde{\boldsymbol{\theta}}}^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\theta}}. \quad (3.3)$$

De la ecuación (2.39) se sabe que:

$$\mathbf{H} \ddot{\mathbf{q}} = \boldsymbol{\tau} - \mathbf{C} \dot{\mathbf{q}} - \mathbf{D} \dot{\mathbf{q}} - \mathbf{g}. \quad (3.4)$$

Substituyendo (3.4) en (3.3) y debido a que la matriz $\dot{\mathbf{H}} - 2\mathbf{C}$ es antisimétrica como se demuestra en el Apéndice A y la forma cuadrática asociada a cualquier matriz antisimétrica es nula, se elimina el término $\frac{1}{2} \mathbf{s}^T \dot{\mathbf{H}} \mathbf{s}$ resultando en:

$$\dot{V}(t) = \mathbf{s}^T (\boldsymbol{\tau} - \mathbf{H} \ddot{\mathbf{q}}_r - \mathbf{C} \dot{\mathbf{q}}_r - \mathbf{D} \dot{\mathbf{q}}_r - \mathbf{g} - \mathbf{D} \mathbf{s}) + \dot{\tilde{\boldsymbol{\theta}}}^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\theta}} \quad (3.5)$$

De manera similar al regresor de parámetros del Capítulo 2, el modelo dinámico se puede expresar de la forma:

$$\mathbf{H}(\mathbf{q}) \ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_r + \mathbf{D} \dot{\mathbf{q}}_r + \mathbf{g}(\mathbf{q}) = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r) \boldsymbol{\theta} \quad (3.6)$$

donde la matriz $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)$ es parecida a la presentada en la ecuación (2.41), la única diferencia es el reemplazo de los vectores $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ por $\dot{\mathbf{q}}_r$ y $\ddot{\mathbf{q}}_r$, respectivamente.

A continuación se propone la ley de control:

$$\boldsymbol{\tau} = \mathbf{Y} \hat{\boldsymbol{\theta}} - \mathbf{K} \mathbf{s} \quad (3.7)$$

la cual incluye un primer término $\mathbf{Y} \hat{\boldsymbol{\theta}}$ para eliminar la dinámica del sistema y un segundo término $\mathbf{K} \mathbf{s}$ el cual introduce un lazo de control PD. Substituyendo (3.7) en (3.5):

$$\dot{V}(t) = \mathbf{s}^T \mathbf{Y} \tilde{\boldsymbol{\theta}} - \mathbf{s}^T (\mathbf{K} + \mathbf{D}) \mathbf{s} + \dot{\tilde{\boldsymbol{\theta}}}^T \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\theta}}. \quad (3.8)$$

Por último se elige una ley de adaptación conveniente, por lo que se propone:

$$\dot{\boldsymbol{\theta}} = -\boldsymbol{\Gamma}\mathbf{Y}^T\mathbf{s} \quad (3.9)$$

resultando (3.18) en:

$$\dot{V}(t) = -\mathbf{s}^T(\mathbf{K} + \mathbf{D})\mathbf{s} \quad (3.10)$$

donde se observa que $\dot{V}(t)$ es una forma cuadrática negativa semidefinida.

Para el análisis de estabilidad en sistemas dinámicos variables en el tiempo se emplea el siguiente Corolario derivado del lema de Barbalat

Corolario 3.1 *Si una función escalar $V(x, t)$ satisface las siguientes condiciones*

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

Entonces $\dot{V}(x, t) \rightarrow 0$ cuando $t \rightarrow \infty$.

La función candidata de Lyapunov (3.2) y su derivada cumplen con las condiciones anteriores por lo tanto el error de velocidad converge a $\mathbf{s} \rightarrow \mathbf{0}$, como también lo hace $\dot{\tilde{\mathbf{q}}}$ y $\tilde{\mathbf{q}}$ cuando $t \rightarrow \infty$.

3.3. Diseño de trayectorias

Al implementar una ley de control se busca que la planta tenga una respuesta igual a una referencia previamente definida, ésta puede ser variante o invariante en el tiempo. Una referencia invariante en el tiempo se utiliza en sistemas donde se necesita tener una regulación constante, por ejemplo, el control de temperatura de un edificio o el de velocidad en un automóvil. Una referencia variante en el tiempo se emplea en sistemas en los cuales se requiere un comportamiento más complejo como cuando se busca el seguimiento de trayectoria de un robot.

En esta tesis se proponen dos trayectorias para el robot Novint Falcon: La primer trayectoria es punto a punto con la finalidad de que el experimento pueda iniciar en

cualquier lugar del espacio de trabajo, el cual representa el punto de inicio, mientras que el punto final se encuentra en el origen del sistema de referencia háptico. La segunda trayectoria está definida como una lemniscata en un plano paralelo al formado por xy .

Las articulaciones que seguirán la trayectoria son las que se encuentran más próximas a la base (θ_{1i}), por lo que se establece la siguiente relación: $q_i = \theta_{1i}$ para $i = 1, 2, 3$.

Existen diversos métodos para diseño de trayectorias dependiendo de las características deseadas. A continuación se describen dos métodos a utilizar de acuerdo a las trayectorias propuestas.

3.3.1. Trayectoria punto a punto

En el caso de la primer trayectoria se utiliza un polinomio con la finalidad de trazar una curva suave desde un punto inicial q_0 hasta un punto final q_f . Se considera trazar la trayectoria para una sola articulación dado que son independientes y exactamente de la misma forma. En primer lugar se establecen las siguientes restricciones de posición, velocidad y aceleración:

$$\begin{aligned} q_0 &= q(t_0) & q_f &= q(t_f) \\ \dot{q}_0 &= 0 & \dot{q}_f &= 0 \\ \ddot{q}_0 &= 0 & \ddot{q}_f &= 0. \end{aligned}$$

Se plantea un polinomio con seis coeficientes independientes tales que satisfagan dichas restricciones, por lo tanto se considera uno de quinto orden de la forma:

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (3.11)$$

donde la velocidad y la aceleración están dadas por:

$$\dot{q}(t) = a_1 + 2a_2t + 3a_3t^2 + 4a_4t^3 + 5a_5t^4 \quad (3.12)$$

$$\ddot{q}(t) = 2a_2 + 6a_3t + 12a_4t^2 + 20a_5t^3. \quad (3.13)$$

Aplicando las restricciones de velocidad y aceleración a las ecuaciones (3.11), (3.12) y (3.13) se obtienen seis ecuaciones con seis incógnitas:

$$q(t_0) = a_0 \quad (3.14)$$

$$q(t_f) = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \quad (3.15)$$

$$0 = a_1 \quad (3.16)$$

$$0 = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \quad (3.17)$$

$$0 = a_2 \quad (3.18)$$

$$0 = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \quad (3.19)$$

De las ecuaciones (3.14), (3.16) y (3.18) se obtienen los coeficientes a_0 , a_1 y a_2 , restando un sistema de tres ecuaciones con tres incógnitas. Resolviendo dicho sistema y substituyendo en la ecuación (3.11) se tiene:

$$q(t) = q(t_0) + \frac{10[q(t_f) - q(t_0)]}{t_f^3} t^3 - \frac{15[q(t_f) - q(t_0)]}{t_f^4} t^4 + \frac{6[q(t_f) - q(t_0)]}{t_f^5} t^5 \quad (3.20)$$

que es la ecuación que define la trayectoria en el intervalo de t_0 a t_f .

3.3.2. Lemniscata de Bernoulli

La segunda trayectoria propuesta se trata de una Lemniscata de Bernoulli la cual presenta una curva similar al símbolo de infinito. Se traza en un plano paralelo a xz , el cual se encuentra a una distancia variable del origen del sistema de referencia háptico H , como se observa en la Figura 3.4. A diferencia de la trayectoria anterior, requiere del uso de la cinemática inversa para determinar la posición de las articulaciones q_i .

Para trazar la curva se emplean las ecuaciones paramétricas de la lemniscata de Bernoulli para el eje x y para el eje z , mientras que el valor del eje y varía de forma sinusoidal, con la mitad de velocidad angular empleada por las ecuaciones paramétricas anteriores. Por lo tanto las ecuaciones paramétricas de cada uno de los

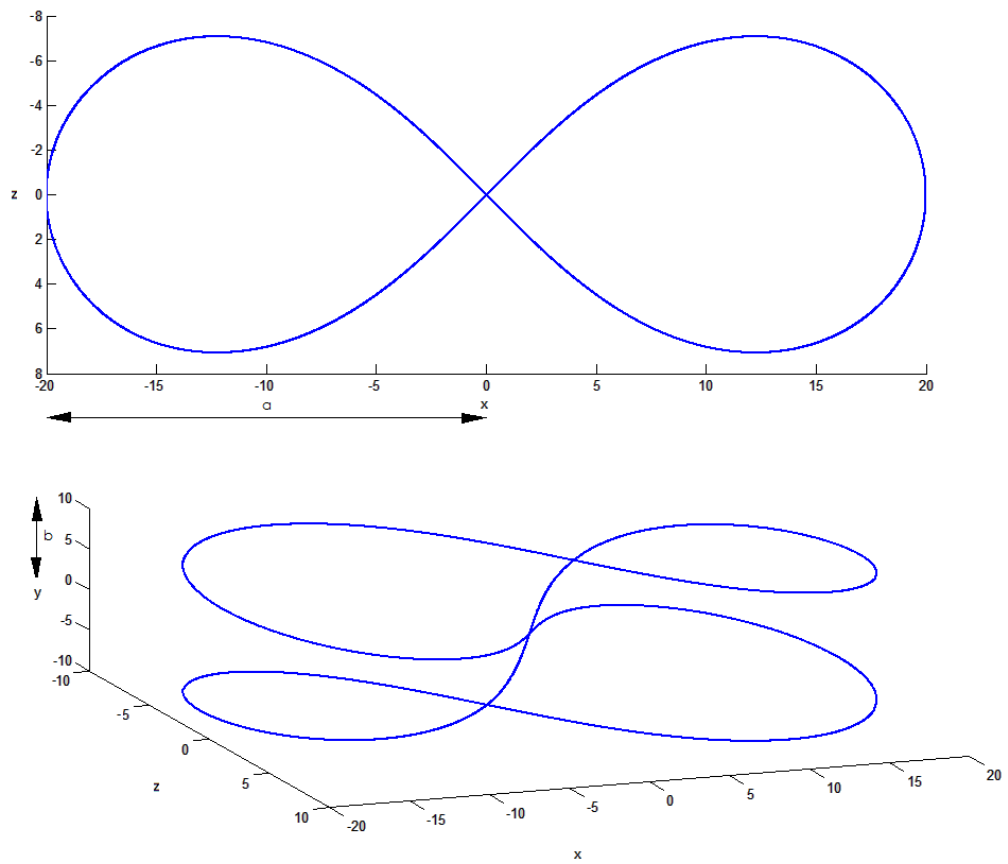


Figura 3.4: Trayectoria de Lemniscata de Bernoulli con $a = 20[mm]$ y $b = 10[mm]$

ejes son:

$$x = \frac{a \cos(\omega t)}{1 + \sin^2(\omega t)} \quad (3.21)$$

$$y = b \cos(0.5\omega t) \quad (3.22)$$

$$z = \frac{a \sin(\omega t) \cos(\omega t)}{1 + \sin^2(\omega t)} \quad (3.23)$$

donde a es la mitad del ancho de la Lemniscata , b es la profundidad en el eje y y ω representa la velocidad angular.

Con el contenido de este capítulo se tiene todo lo necesario para implementar los controladores de forma experimental con base en el esquema de control en lazo cerrado de la Figura 3.1.

Capítulo 4

Implementación y evaluación experimental

Para realizar los experimentos con los algoritmos de control en el robot *Novint Falcon* se utilizó solamente una PC con *Windows 7*. La interfaz entre el robot y un lenguaje de programación se realizó por medio de la biblioteca *Haptik*. Se elaboró una guía detallando el proceso seguido para la instalación requerida para la puesta en funcionamiento del robot *Novint Falcon* en el sistema *Windows* el cual puede ser consultado en el Apéndice C. Para mostrar la versatilidad de dicha biblioteca se realizaron dos programas, el primero de ellos se programó en lenguaje *C++*, el segundo se basa en el entorno de programación *Simulink* de *Matlab*. Posteriormente, se realizó una validación experimental de los algoritmos de control propuestos en el Capítulo 3 con el fin de probar su rendimiento y comparar las ventajas y desventajas de cada uno.

4.1. Implementación con *Haptik library*

La biblioteca *Haptik* ofrece una capa de abstracción de hardware que permite acceder a diversos dispositivos hápticos entre ellos el robot *Novint Falcon*, además es de código abierto por lo que es una herramienta ideal para la implementación de los controladores. *Haptik* es independiente de cualquier otro software por lo que presenta una gran variedad de plataformas con las que es compatible como *OpenGL*

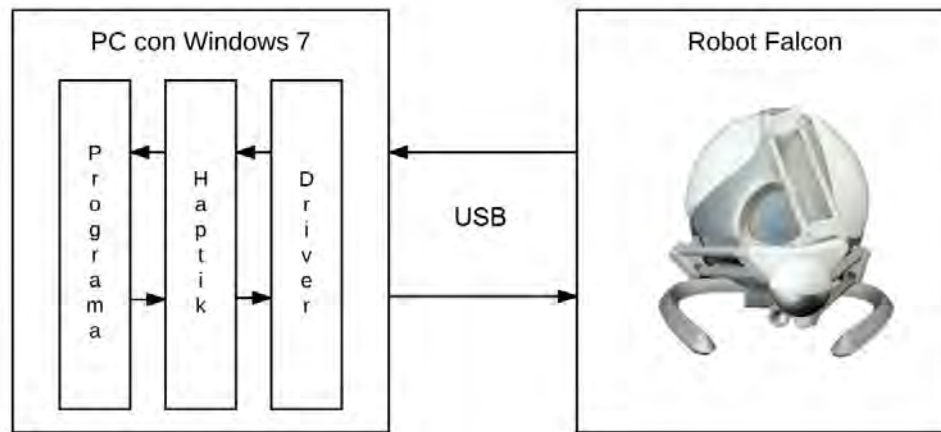


Figura 4.1: Comunicación de robot *Novint Falcon*

y *DirectX*. La biblioteca contiene una API para lenguaje *C++* y ofrece también acceso a ambientes como *Matlab* y *Simulink*.

En la Figura 4.1 se muestra la interfaz entre los programas y el robot *Novint Falcon*. La conexión física entre el robot y la PC se realiza mediante USB 2.0, y la interacción con el sistema operativo se realiza por medio del *driver* del dispositivo, el cual es proporcionado directamente por el fabricante.

4.1.1. Controlador PID en *C++*

El primer programa se basa en *C++*, lo cual le da una gran portabilidad pero al mismo tiempo puede resultar complicado al querer implementar algoritmos de control complejos como el control adaptable propuesto en el Capítulo 3. En cambio, al implementar algoritmos sencillos como es el caso del controlador PID resulta una buena opción, por lo que este último se implementó en este lenguaje.

La biblioteca *Haptik* provee la posición del efector final en el sistema de referencia háptico H , por lo que es necesario programar la cinemática inversa para obtener la posición articular. Por otra parte, *Haptik* tiene como entrada la fuerza en el efector final y dado que el algoritmo de control PID calcula el momento de fuerza, es

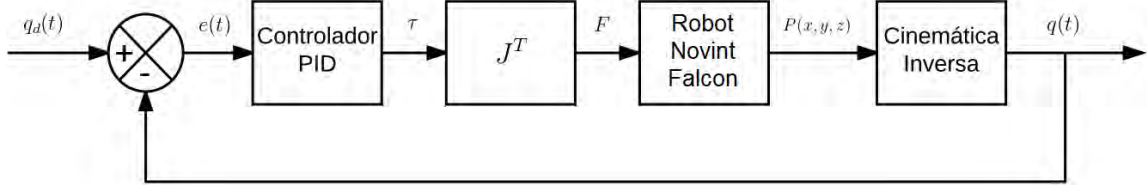


Figura 4.2: Diagrama de implementación de controlador PID.

necesario programar la siguiente relación:

$$\mathbf{F} = \mathbf{J}^T(\mathbf{q})\boldsymbol{\tau} \quad (4.1)$$

donde $\mathbf{J}^T(\mathbf{q})$ es el Jacobiano transpuesto, $\boldsymbol{\tau}$ es el momento de fuerza calculado por el controlador y \mathbf{F} es la fuerza en el efector final.

En la Figura 4.2 se muestra la implementación del controlador PID.

4.1.2. Controlador Adaptable en *Simulink*

El segundo programa contiene el controlador adaptable y está hecho en *Simulink* de *Matlab* para facilitar su programación. La programación en *Simulink* de un algoritmo de control complejo como en el caso del control adaptable se simplifica al permitir el uso de métodos numéricos preprogramados de *Matlab* como el usado en la inversión de matrices. En este controlador se requiere la medición de la velocidad de las articulaciones pero la biblioteca *Haptik* sólo proporciona la posición, para obtener la velocidad se hace uso de un diferenciador de Levant detallado en la siguiente sección. La ley de control utilizada en este programa es la descrita en la ecuación (3.7), mientras que la ley de adaptación implementada se encuentra en la ecuación (3.9).

El diagrama de la implementación del controlador adaptable se muestra en la Figura 4.3.

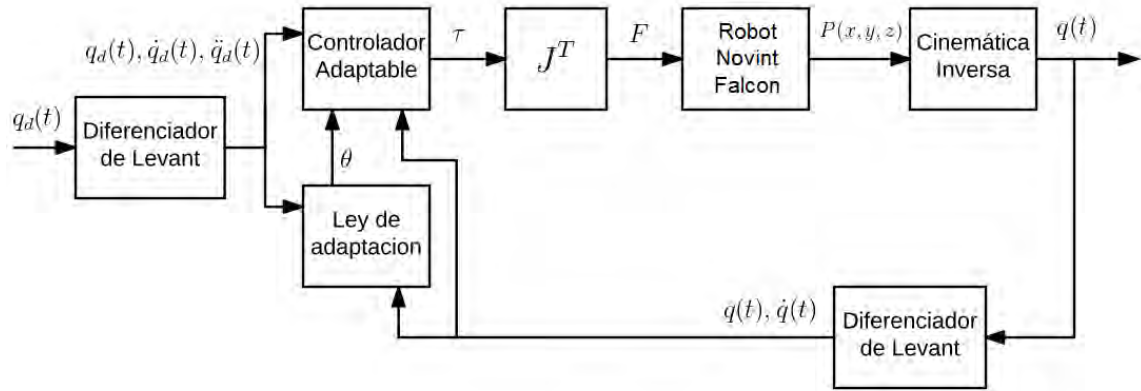


Figura 4.3: Diagrama de implementación de controlador adaptable.

4.1.3. Uso del diferenciador de Levant

La API que ofrece *Haptik* sólo permite la lectura de la posición, en el caso del controlador PID no es necesario medir la velocidad articular, pero al contrario en el caso del controlador adaptable sí lo es. Para obtener la velocidad basta con calcular la derivada de la posición, llevar a cabo dicha tarea usando el algoritmo que ofrece *Simulink* no es conveniente debido al ruido generado en la medición. Por fortuna existen otros métodos que son robustos ante el ruido, como el diferenciador propuesto por Levant [12] y que se detalla en el Apéndice B.

4.2. Evaluación experimental

Para evaluar a los controladores se realiza un experimento basado en el seguimiento de las trayectorias del Capítulo 3, donde:

- La duración de la primer trayectoria punto a punto es de cuatro segundos. A partir de este momento se realiza la Lemniscata de manera cíclica hasta los 30 segundos.
- Se establece una velocidad angular de $\omega = 0.5\pi[\text{rad}/\text{s}]$ para la segunda trayectoria, adicionando un desfase de $\pi/2$ con el fin de que cuando $t = 4[\text{s}]$ ésta se

encuentre en el origen.

- Las dimensiones de la Lemniscata son $a = 20[mm]$ y $b = 10[mm]$.

El experimento es realizado tanto con en el controlador PID como con en el control adaptable.

4.2.1. Resultados de controlador PID

El experimento con el controlador PID se realiza con el programa de la sección 4.1.1. Las ganancias se seleccionaron a partir de los valores obtenidos por medio de la sintonización, los cuales se muestran en la Tabla 3.2 del Capítulo 3.

En la Figura 4.4 se observa que la trayectoria q se asemeja a la deseada q_d para cada una de las articulaciones. El desempeño de este controlador se muestra en la Figura 4.5 donde el error más grande es de 3.25° . De igual forma es posible observar sobresaltos en el error, éstos son debido al ruido de medición en combinación al uso de un algoritmo simple para obtener la derivada en el control PID.

4.2.2. Resultados de controlador adaptable

El experimento para el controlador adaptable se realizó con el modelo en *Simulink* de la sección 4.1.2. En la Figura 4.6 se observa el seguimiento de las trayectorias propuestas usando este controlador.

El rendimiento de este controlador se observa en la Figura 4.7 donde el error articular no pasa los 3° , además el error claramente se acerca a cero a medida que pasa el tiempo. A diferencia del controlador anterior, no se presentan sobrepasos en el error, ya que a pesar de que existe ruido en la medición, en este controlador se emplea el diferenciador de Levant el cual es robusto ante ruido y teóricamente exacto. En la Figura 4.8 se muestra el error articular a 100 segundos donde se muestra la tendencia a cero a medida que pasa el tiempo.

La estimación del vector de parámetros de la ecuación (2.42) se presenta en la Figura 4.9. Es posible observar que el valor del parámetro estimado θ_1 es negativo, lo anterior se debe a que la trayectoria propuesta no brinda suficiente información para estimar este parámetro como menciona Ioannou [10]. A pesar de no lograr la

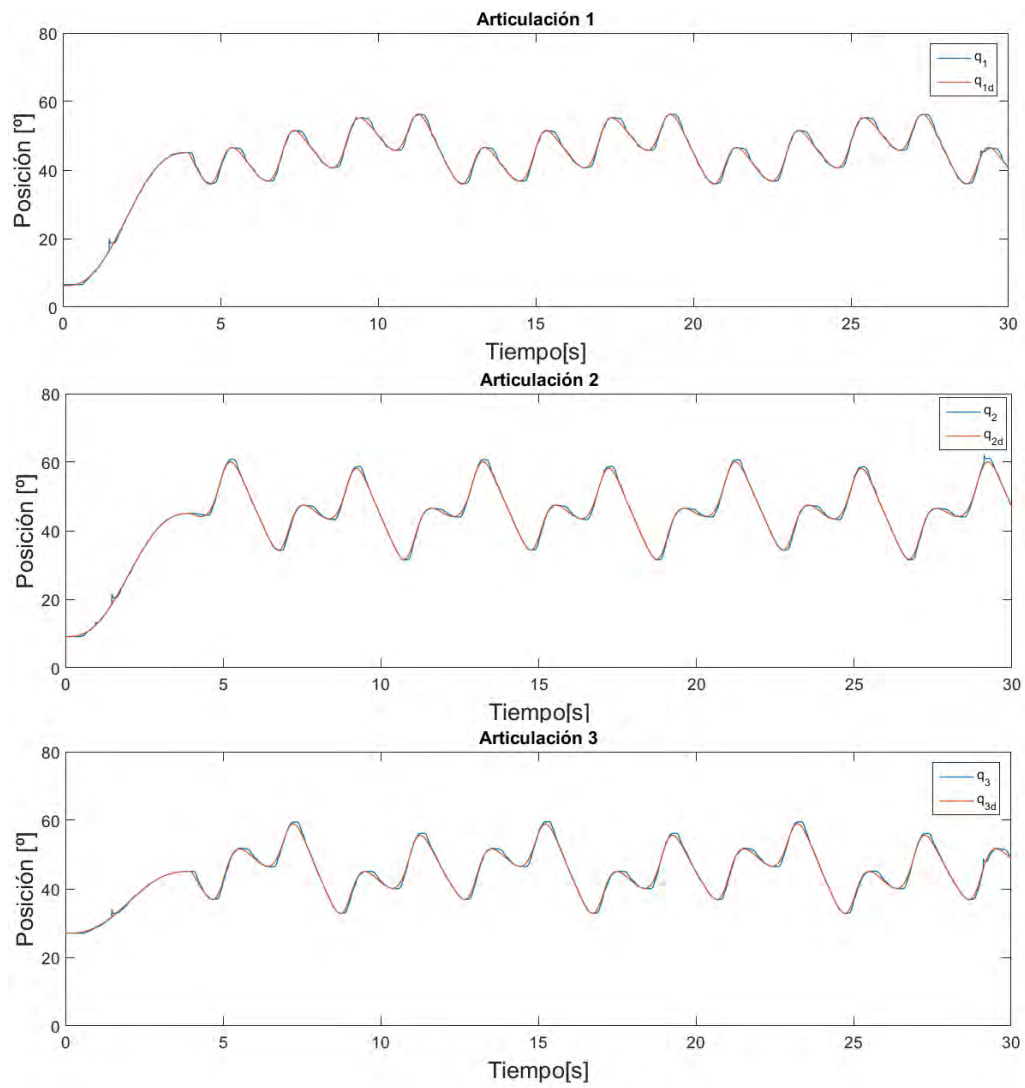


Figura 4.4: Posición articular real y deseada PID

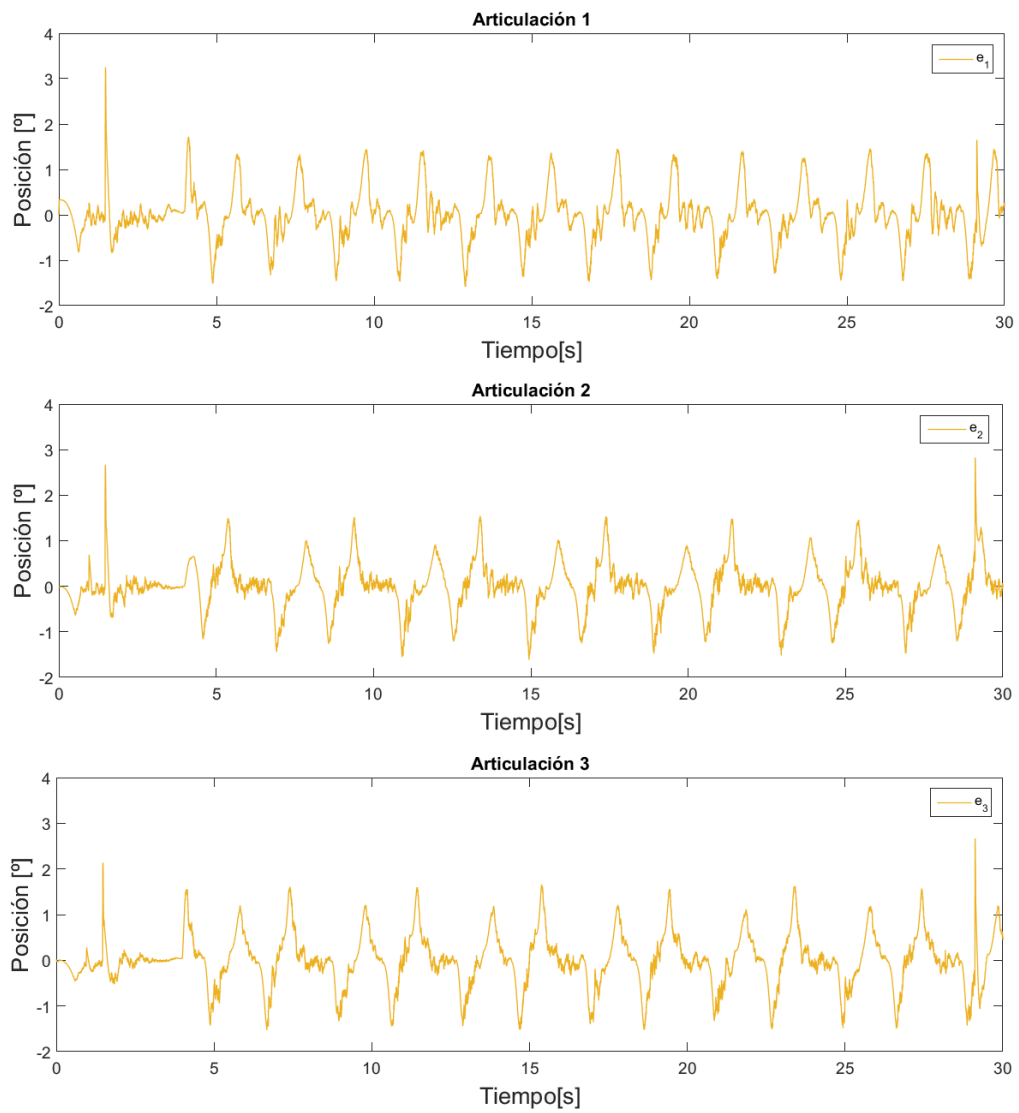


Figura 4.5: Error articular PID

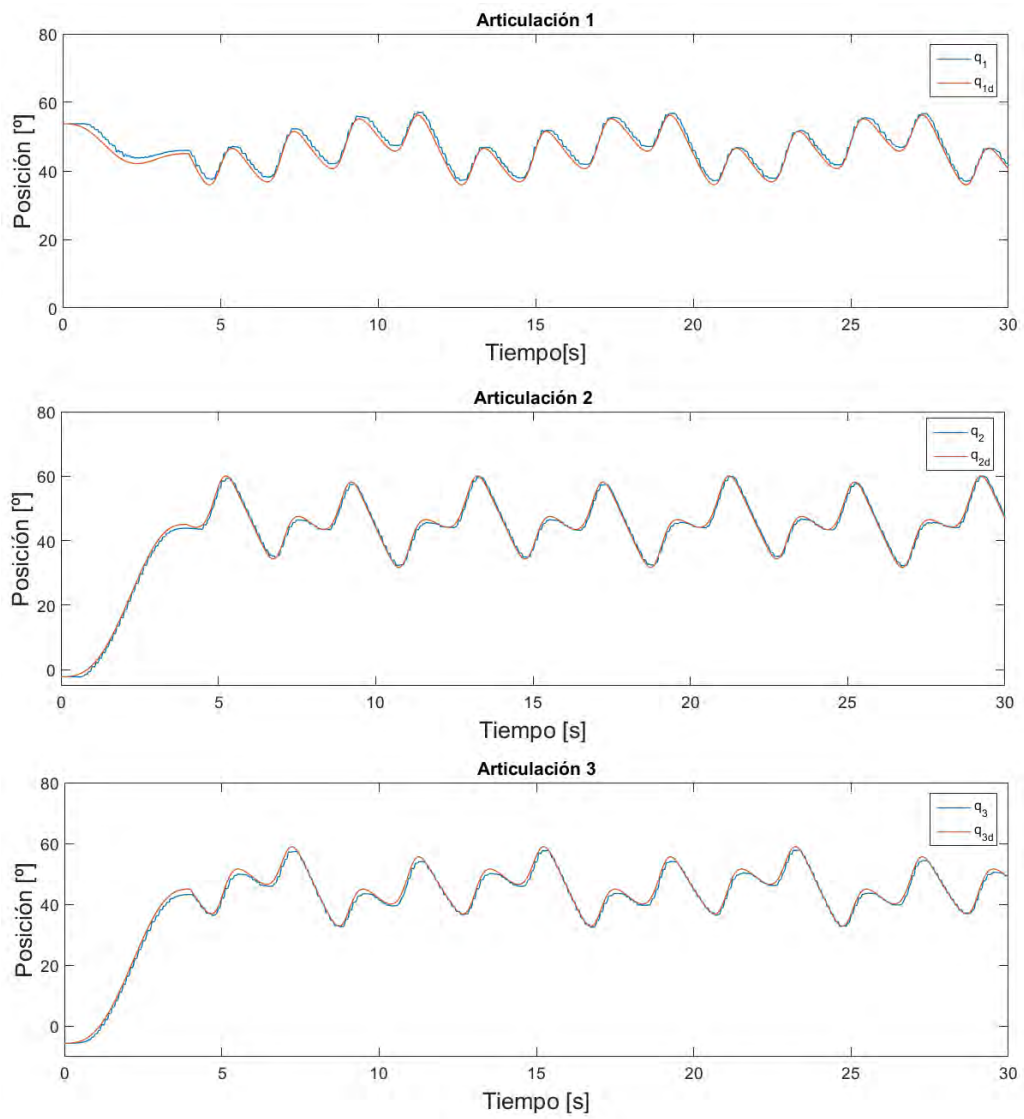


Figura 4.6: Posición articular real y deseada en controlador adaptable

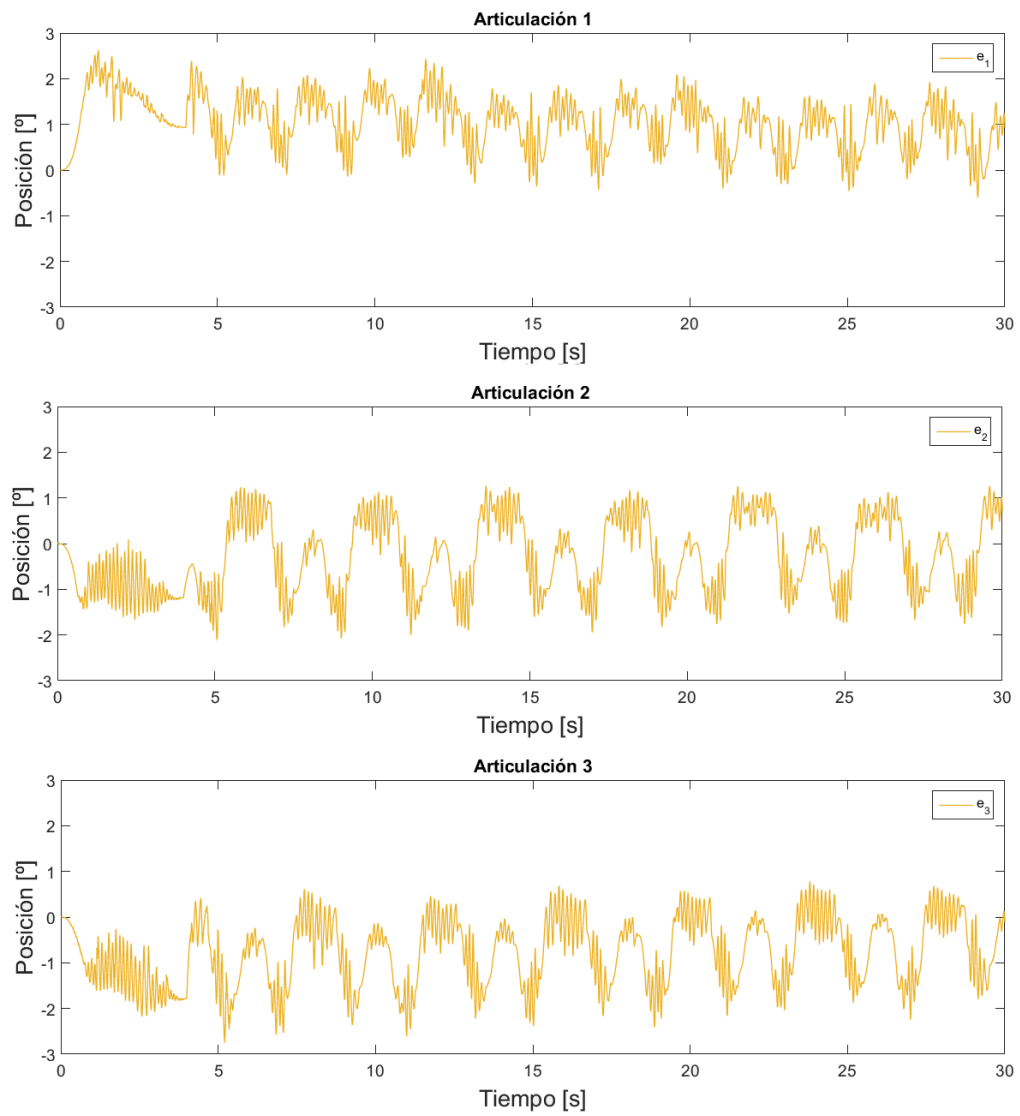


Figura 4.7: Error articular en controlador adaptable

estimación de todos los parámetros el objetivo de esta tesis es controlar el sistema, por lo que los resultados son satisfactorios.

4.3. Análisis de resultados

Para poder realizar una comparación analítica de los controladores, se utiliza la raíz del error cuadrático medio (RMSE por sus siglas en inglés) como indicador. Se define el RMSE como:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (q - q_d)^2} \quad (4.2)$$

donde n es el número de muestras. En la Tabla 4.1 se muestra el RMSE de cada articulación tanto para el experimento del controlador PID como del controlador adaptable, también se incluye el error absoluto máximo.

Controlador	$RMSE[^\circ]$			$ e_{max} [^\circ]$		
	q_1	q_2	q_3	q_1	q_2	q_3
PID	0.56	0.52	0.57	3.25	2.82	2.66
Adaptable	1.2	0.69	0.9	2.62	1.69	2.34

Tabla 4.1: Tabla de indicadores.

De acuerdo a los indicadores de la Tabla 4.1 el RMSE del controlador PID es menor que el del controlador adaptable, lo que indica un mejor desempeño del controlador PID frente al adaptable. Pero si se observa la Figura 4.8 el mejor rendimiento del controlador adaptable se tiene después de 70 segundos transcurridos, lo que invita a hacer una comparación entre el controlador PID y este último intervalo.

Controlador	$RMSE[^\circ]$			$ e_{max} [^\circ]$		
	q_1	q_2	q_3	q_1	q_2	q_3
PID	0.56	0.52	0.57	3.25	2.82	2.66
Adaptable	0.51	0.56	0.55	1.41	1.23	1.4

Tabla 4.2: Tabla de indicadores últimos 30 segundos.

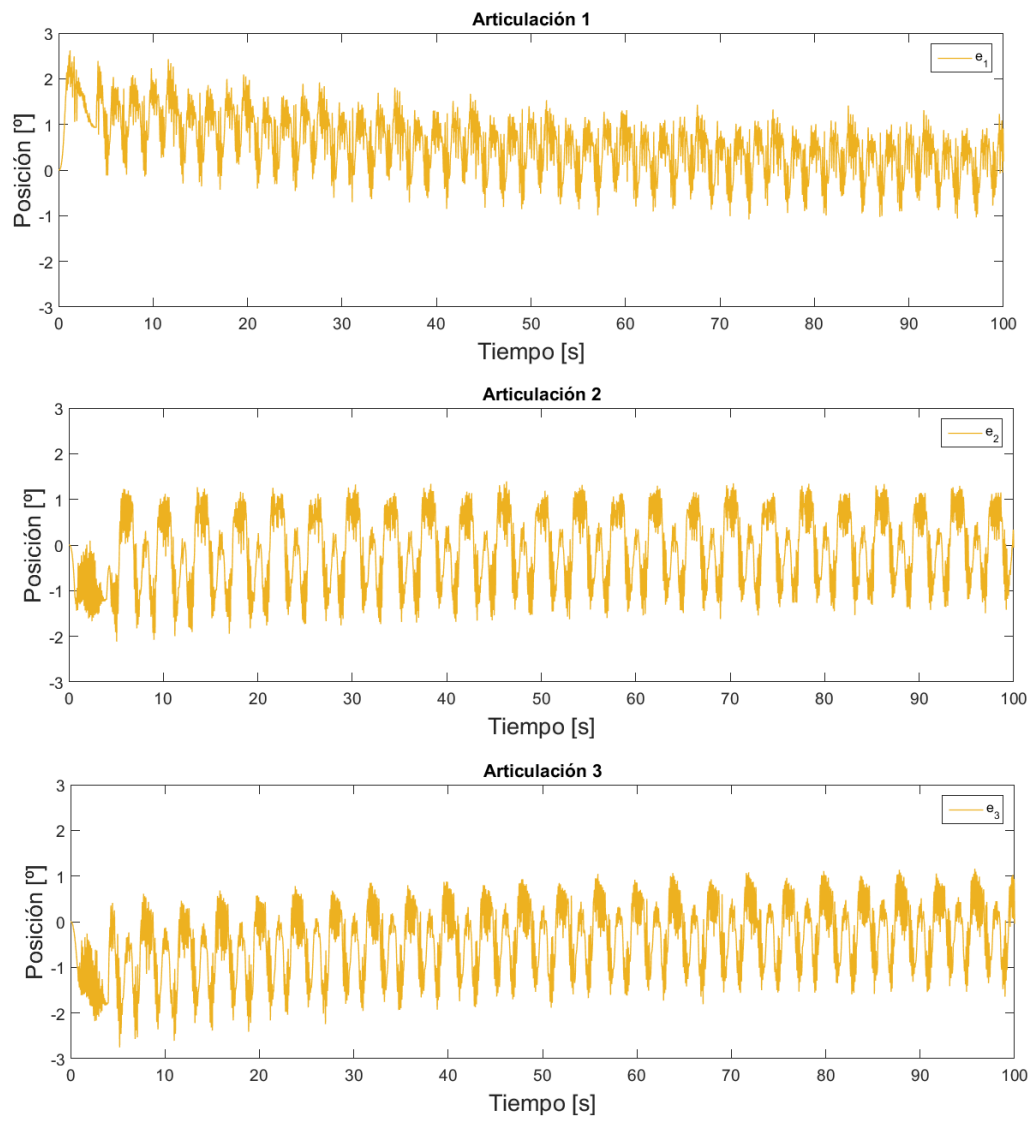


Figura 4.8: Error articular en controlador adaptable a 100 segundos

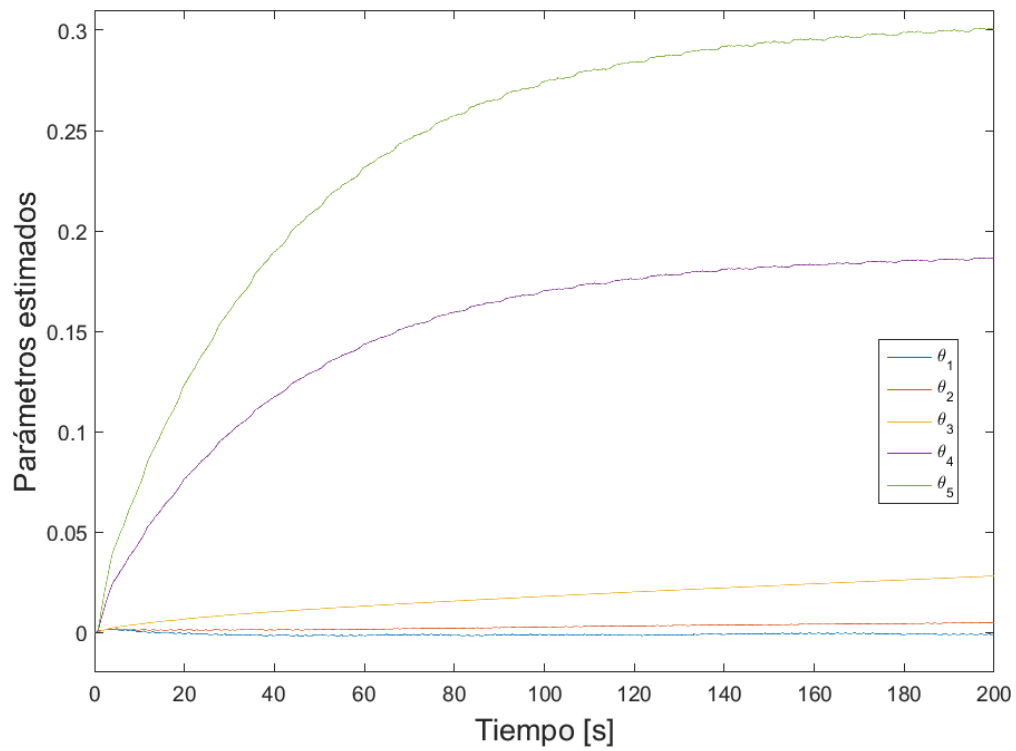


Figura 4.9: Parámetros estimados en 200 segundos

De la Tabla 4.2 se observa que en el intervalo de 70 a 100 segundos del experimento, el controlador adaptable presenta un 8% y 4% menos RMSE con respecto al controlador PID para la articulación uno y tres respectivamente, mientras que la articulación dos es un 6.5% mayor. En cuanto a el error absoluto el controlador adaptable genera un error 56%, 55% y 47% menor con respecto al PID para la articulación 1, 2 y 3 respectivamente. De esta manera se concluye que el controlador adaptable tiene un rendimiento ligeramente mejor que el controlador PID, pero sólo hasta después de transcurridos 70 segundos del experimento.

Capítulo 5

Robot *Novint Falcon* como dispositivo háptico

Como se ha mencionado, el robot *Novint Falcon* fue diseñado para ser usado como dispositivo háptico específicamente en videojuegos, siendo el primer dispositivo de su tipo en ser lanzado al mercado de consumo. En este capítulo se exploran las capacidades hápticas del robot *Novint Falcon* con el desarrollo de un sistema de realidad virtual.

5.1. Sistema de realidad virtual

Un sistema de realidad virtual permite la interacción entre el usuario y un ambiente virtual. El objetivo es que dicha interacción involucre a los cinco sentidos, pero en la actualidad los sistemas de realidad virtual sólo interactúan mediante visión, sonido y más recientemente mediante el tacto.

Un ambiente virtual consiste en elementos (objetos o personajes) que son representados mediante estímulos sensoriales de distintos tipos. Los objetos en un ambiente virtual al igual que en la realidad contienen propiedades como forma, peso, color, textura, densidad y temperatura. Estas propiedades son percibidas mediante diferentes estímulos sensoriales por ejemplo, el color se percibe mediante estímulos visuales, mientras que la forma del objeto se percibe mediante estímulos hápticos [16]. El ambiente virtual implementado en esta tesis se observa en la Figura 5.1 y se basa

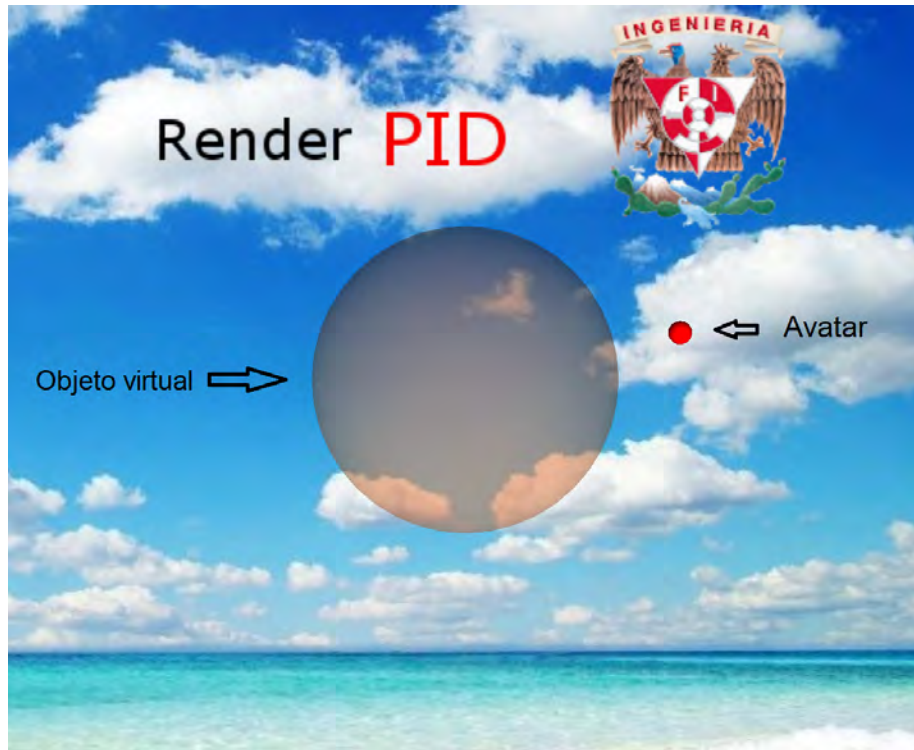


Figura 5.1: Ambiente virtual

en una esfera fija ubicada en el centro de la escena más un *avatar*. Usualmente se define como *avatar* a la representación virtual del dispositivo háptico que el usuario manipula a través del cuál ocurre la interacción física con el ambiente virtual. La elección del *avatar* depende de lo que se está simulando, en nuestro caso se optó por una pequeña esfera de color rojo.

La arquitectura del sistema de realidad virtual implementado en el robot *Novint Falcon* se muestra en la Figura 5.2. Los elementos principales que la integran son tres: el motor de simulación, el renderizado visual y háptico y por último, los transductores. El motor de simulación se encarga del comportamiento del entorno virtual en todo momento. El renderizado visual y háptico calcula las respuestas visuales y hápticas del entorno virtual hacia el usuario. Finalmente los transductores son los encargados de convertir las señales de video y fuerza en estímulos que puedan ser percibidos por el usuario. Los transductores son: para la señal de video una pantalla y para la señal de fuerza el robot *Novint Falcon*.

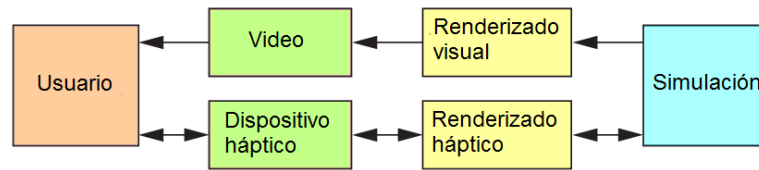


Figura 5.2: Arquitectura básica de una aplicación de realidad virtual con háptica

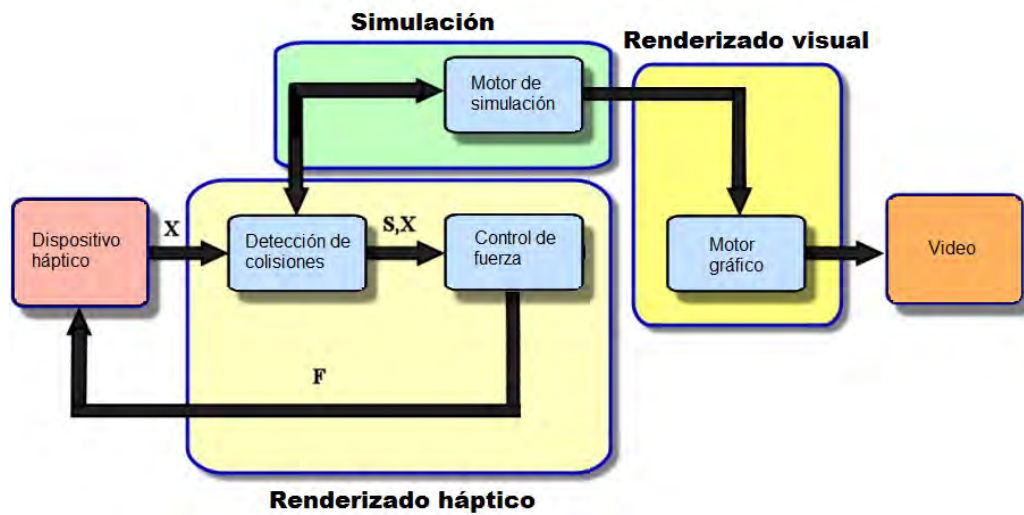


Figura 5.3: Esquema de renderizado háptico

Es importante notar que el canal de información de video es unidireccional, es decir la información sólo se transmite en un sentido (hacia el usuario). Por otra parte el canal de información háptica es bidireccional, la información es transmitida desde y hacia el usuario, siendo ésta su característica más relevante.

5.2. Renderizado háptico

La parte central de un sistema de realidad virtual en cuanto la estimulación táctil es el renderizado háptico. Sus algoritmos son responsables de calcular la correcta interacción de fuerzas entre el dispositivo háptico y el ambiente virtual. En la Figura 5.3 se ilustran los elementos que componen el renderizado háptico. Se pueden identificar dos bloques principales: El algoritmo de detección de colisiones y el de control de fuerza.

5.2.1. Detección de Colisiones

El algoritmo de detección de colisiones se encarga de procesar cuando existe algún contacto entre el avatar y el objeto virtual, proporcionando información sobre dónde y cuándo ocurren. El tipo de algoritmo a implementar se encuentra estrechamente relacionado con la geometría del objeto. Gran parte de los métodos se originaron a partir de la computación gráfica, algunos de los más utilizados son: superficie implícita, superficie paramétrica y modelo poligonal.

Dado que el único objeto interactivo en el ambiente virtual propuesto es una esfera, el método utilizado es el de superficie implícita que consiste en definir la superficie frontera mediante una función $f(x, y, z) = 0$. De esta forma se puede saber si algo se encuentra dentro $f(x, y, z) < 0$, o fuera de la superficie $f(x, y, z) > 0$. La función que define la esfera es la siguiente:

$$f(x, y, z) = x^2 + y^2 + z^2 - R^2 \quad (5.1)$$

donde R es el radio de la esfera.

5.2.2. Control de fuerza

Este proceso se encarga de calcular la fuerza resultante de la interacción entre el *avatar* y el objeto virtual ante la presencia de una colisión. En general podemos distinguir dos modelos de interacción. El primer modelo es llamado de impedancia el cual la interfaz háptica mide el desplazamiento y la simulación calcula la fuerza correspondiente. El segundo modelo es llamado de admitancia el cual mide la fuerza entre el usuario y el dispositivo háptico y la simulación calcula el desplazamiento.

El robot *Novint Falcon* puede medir el desplazamiento pero no fuerza por lo que se realizó un control por impedancia, donde la señal entrada del controlador es el desplazamiento y la señal de salida es fuerza. Para el programa se utilizarán tres algoritmos de respuesta basados en el esquema de control PID de la ecuación 3.1:

$$u(t) = k_p \left(e(t) + \frac{1}{h_i} \int_{-\infty}^t e(t) dt + h_d \frac{de(t)}{dt} \right)$$

- El primero de ellos se trata de un control P, el cual se basa en el algoritmo PID

con $h_d = 0$ y $h_i \rightarrow \infty$. Este algoritmo recrea el comportamiento de un resorte con una constante elástica k_p .

- El segundo es un control PD, de igual manera se basa en el algoritmo del controlador PID pero con $h_i \rightarrow \infty$. Emula el comportamiento de un sistema resorte-amortiguador donde el producto $k_p h_d$ representa la constante de amortiguamiento.
- El último es un control PID. El cual agrega un término integral al sistema resorte-amortiguador, físicamente no tiene una representación.

Los algoritmos requieren de la posición del avatar $f(x, y, z)$ del cual se obtiene el error definido como

$$e(t) = R - \sqrt{R^2 + f(x, y, z)}.$$

La respuesta de los algoritmos $u(t)$ representa el módulo del vector de fuerza aplicado directamente al usuario mediante el dispositivo háptico, para obtener dicho vector basta con multiplicar su módulo por el vector unitario que parte desde el origen hacia la posición del *avatar*:

$$\mathbf{F} = \frac{u(t)}{\sqrt{x^2 + y^2 + z^2}} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

5.3. Implementación mediante HDAL

A diferencia de la implementación realizada en el Capítulo 4 en ésta no se hace uso de la librería *Haptik*, en cambio se utiliza la API proporcionada directamente por el fabricante Novint llamada HDAL (*Haptic Device Abstraction Layer*). Ofrece soporte para una gran variedad de dispositivos con y sin retroalimentación de fuerza y por otra parte es una API creada para su implementación únicamente en lenguaje *C++*. Se utilizó una PC con *Windows 7* y al igual que la librería *Haptik*, HDAL es el puente entre el driver y el programa y la conexión física de la PC al dispositivo es mediante USB 2.0.

El programa se realizó en lenguaje *C++* y se compone por dos partes principales: el renderizado visual y el renderizado háptico. El renderizado gráfico se realizó con

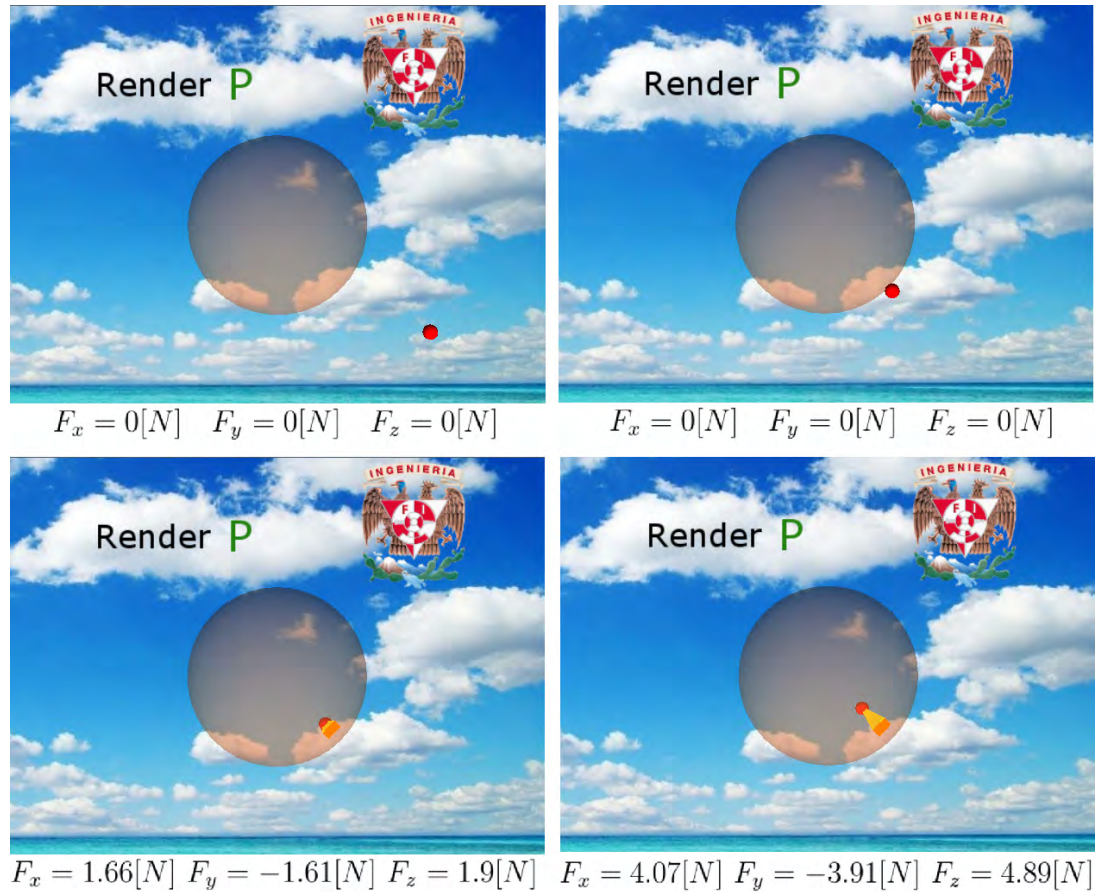


Figura 5.4: Secuencia de interacción con sistema de realidad virtual.

el uso de la especificación *OpenGL* 4.5 y la biblioteca *GLUT*. Por otro lado la API *HDAL* ofrece una función síncrona con una frecuencia de $1[kHz]$ y es aquí donde se realiza el renderizado háptico, es decir el cálculo de las fuerzas con las que interacciona el dispositivo, además proporciona una función asíncrona para comunicarse con el motor de simulación.

En la Figura 5.4 se aprecia la interacción entre el usuario y el ambiente virtual donde el tipo de renderizado utilizado es proporcional (P) con una constante $k_p = 984$ y la fuerza está dada en el sistema de referencia háptico $H(x, y, z)$.

Capítulo 6

Conclusiones

La elaboración de esta tesis involucró tanto trabajo teórico como trabajo experimental. Dentro del trabajo teórico se presentó un análisis de la cinemática y dinámica del robot *Novint Falcon*, el cual brindó las bases necesarias para su puesta en funcionamiento. Además el modelo dinámico propuesto en este trabajo deja un precedente importante ya que no se tenía reporte de alguno en la literatura.

Como parte del trabajo experimental se logró exitosamente la implementación de los esquemas de control propuestos (PID y control adaptable) logrando un satisfactorio seguimiento de trayectoria. Para documentar el procedimiento de instalación y configuración de *drivers* y APIs requeridas para la implementación se elaboró el manual contenido en el Apéndice C.

Es conveniente señalar que el desempeño del controlador PID cumplió las expectativas, en cambio el controlador adaptable las cumple sólo después de 70 segundos. La causa por la que el controlador adaptable no cumplió con las expectativas fue el seguimiento de una trayectoria con la que el sistema es incapaz de estimar correctamente todos los parámetros. Por lo anterior se determina que el seguimiento de trayectoria empleando un controlador PID, no depende de que la trayectoria sea lo suficientemente rica en frecuencias, manteniendo el error acotado a lo largo del experimento, mientras que el desempeño del controlador adaptable depende de que la trayectoria contenga las suficientes frecuencias para estimar los parámetros, con la ventaja de disminuir el error conforme el tiempo avanza.

Aparte de haber probado que el robot *Novint Falcon* es una plataforma expermen-

tal adecuada para implementar algoritmos de control, se demostraron las capacidades hápticas del dispositivo mediante la implementación de un sistema de realidad virtual, donde se constataron las características del robot paralelo como su gran fuerza y pequeño espacio de trabajo.

Finalmente, con este trabajo se demostró el valor del robot *Novint Falcon* el cual es uno de los robots más económicos del mercado lo que le da un gran potencial para el desarrollo de tecnología háptica que antes, por su alto costo no estaba al alcance de una gran parte de la comunidad científica.

6.1. Trabajo a futuro

Este trabajo deja un precedente para el uso de este robot y su utilidad como dispositivo háptico, siendo esta última una aplicación con un futuro favorable ya que la tecnología háptica se encuentra en su inicio y se espera un crecimiento exponencial en los próximos años, por lo que se propone:

- Realizar una identificación paramétrica con una trayectoria que permita estimar todos los parámetros.
- Realizar un experimento con variación paramétrica para comprobar la robustez del controlador adaptable.
- Continuar con el desarrollo de sistemas de realidad virtual para el dispositivo, como sistemas de teleoperación, simuladores de cirugía o rehabilitación médica por mencionar algunos.

Apéndice A

Propiedades de matrices del sistema

Las matrices de la ecuación que describe la dinámica del sistema (2.39) presenta las siguientes propiedades:

Propiedad A.1 *La matriz $\mathbf{H}(\mathbf{q})$ es simétrica positiva definida.*

Propiedad A.2 *La matriz $\dot{\mathbf{H}}(\mathbf{q}, \dot{\mathbf{q}}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ es antisimétrica.*

Demostración [20].

Cada elemento de $\dot{\mathbf{H}}(\mathbf{q}, \dot{\mathbf{q}})$ satisface

$$\dot{h}_{ij} = \sum_{k=1}^n \frac{\partial h_{ij}}{\partial q_k} \dot{q}_k \quad i, j = 1, 2, \dots, n. \quad (\text{A.1})$$

Sean los símbolos de Christoffel [25]:

$$c_{kji} = \frac{1}{2} \left(\frac{\partial h_{ij}}{\partial q_k} + \frac{\partial h_{ik}}{\partial q_j} - \frac{\partial h_{kj}}{\partial q_i} \right) = c_{jki} \quad (\text{A.2})$$

de esta manera cada elemento de $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ esta dado por:

$$c_{ij} = \sum_{k=1}^n c_{kji} \dot{q}_k \quad i, j = 1, 2, \dots, n. \quad (\text{A.3})$$

Se calculan los elementos de $\dot{\mathbf{H}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ como:

$$\begin{aligned} n_{ij} &= \dot{h}_{ij} - 2c_{ij} \\ &= \sum_{k=1}^n \left(\frac{\partial h_{ij}}{\partial q_k} - \left(\frac{\partial h_{ij}}{\partial q_k} + \frac{\partial h_{ik}}{\partial q_j} - \frac{\partial h_{kj}}{\partial q_i} \right) \right) \dot{q}_k \\ &= \sum_{k=1}^n \left(\frac{\partial h_{kj}}{\partial q_i} - \frac{\partial h_{ik}}{\partial q_j} \right) \dot{q}_k. \end{aligned}$$

Dado que la matriz de inercia es simétrica $h_{ij} = h_{ji}$, intercambiando los índices i y j se tiene:

$$n_{ji} = \sum_{k=1}^n \left(\frac{\partial h_{ik}}{\partial q_j} - \frac{\partial h_{jk}}{\partial q_i} \right) \dot{q}_k \quad (\text{A.4})$$

$$= -n_{ij} \quad (\text{A.5})$$

de esta manera concluye la demostración.

Apéndice B

Diferenciador de Levant

Sea $f(t)$ una señal con ruido de medición de una señal suave $f_0(t)$ teniendo una constante Lipschitz conocida $L > 0$ para su n derivada. Se asume que el ruido de medición $f(t) - f_0(t)$ es acotado.

Denotando como $D_{n-1}(f(\cdot), L)$ al diferenciador de $n - 1$ orden el cual produce las salidas $D_{n-1}^i(f(\cdot), L)$, con $i = 0, 1, \dots, n - 1$ que representan las estimaciones de $f_0, \dot{f}_0, \dots, f_0^{n-1}$. El diferenciador de n orden tiene la salidas $z_i = D_n^i(f(\cdot), L)$, con $i = 0, 1, \dots, n$, definido recursivamente como:

$$z_1 = D_{n-1}^0(v(\cdot), L), \dots, z_n = D_{n-1}^{n-1}(v(\cdot), L) \quad (\text{B.1})$$

donde

$$\dot{z}_0 = v, \quad v = -\lambda_n L^{1/(n+1)} |z_0 - f(t)|^{n/(n+1)} \text{sign}(z_0 - f(t)) + z_1.$$

El diferenciador $D_0(f(\cdot), L)$ es un filtro no lineal de la siguiente forma:

$$\dot{z} = -\lambda_0 L \text{sign}(z - f(t)). \quad (\text{B.2})$$

Por lo que el diferenciador de n orden tiene la forma:

$$\begin{aligned} \dot{z}_0 &= v_0, & v_0 &= -\lambda_n L^{1/(n+1)} |z_0 - f(t)|^{n/(n+1)} \text{sign}(z_0 - f(t)) + z_1, \\ \dot{z}_1 &= v_1, & v_1 &= -\lambda_{n-1} L^{1/n} |z_1 - v_0|^{(n-1)/n} \text{sign}(z_1 - v_0) + z_2, \end{aligned}$$

$$\begin{aligned} & \vdots \\ \dot{z}_i = v_i, \quad v_i = -\lambda_i L^{1/(n-i+1)} |z_i - v_{i-1}|^{(n-i)/(n-i+1)} \text{sign}(z_i - v_{i-1}) + z_{i+1}, \quad (\text{B.3}) \end{aligned}$$

$$\begin{aligned} & \vdots \\ \dot{z}_{n-1} = v_{n-1}, \quad v_{n-1} = -\lambda_1 L^{1/2} |z_1 - v_0|^{(n-1)/n} \text{sign}(z_1 - v_0) + z_2, \end{aligned}$$

$$\dot{z}_n = -\lambda_0 L \text{sign}(z_n - v_{n-1}).$$

Para determinar el valor de las ganancias λ_i el autor propone dos combinaciones para diferenciadores de orden $n = 5$ o menores. Una combinación es $\lambda_0 = 1.1$, $\lambda_1 = 1.5$, $\lambda_2 = 3$, $\lambda_3 = 5$, $\lambda_4 = 8$, $\lambda_5 = 12$, la otra combinación es $\lambda_0 = 1.1$, $\lambda_1 = 1.5$, $\lambda_2 = 2$, $\lambda_3 = 3$, $\lambda_4 = 5$, $\lambda_5 = 8$.

Apéndice C

Guía de instalación

En esta guía se muestran los pasos a seguir para llevar a cabo la instalación de los recursos indispensables para la puesta en funcionamiento del robot *Novint Falcon*. Los archivos necesarios para llevar a cabo la instalación que describe este manual son:

- `setup.Falcon.v3.1.4.a.exe`
- `2008-08-20_Novint_HDAL_SDK_2.1.3_setup.exe`
- `Haptik.Library.1.1rc1.win32.x86.20071009.exe`

Los cuales se encuentran en la siguiente dirección web:

Dirección URL C.1 <http://www.linkbucks.com/AkePm>

Primer paso: instalar el *driver* que proporciona el fabricante para ello se ejecuta el archivo `setup.Falcon.v3.1.4.a.exe` y se abrirá un asistente de instalación como se muestra en la Figura C.1 se recomienda usar las opciones por defecto, es importante verificar que la ruta de instalación sea: *C:/Program Files/Novint/Falcon*.

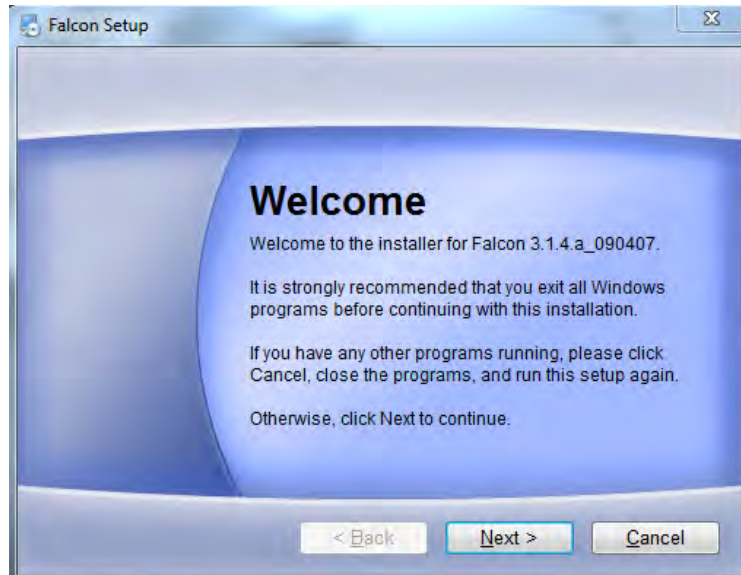


Figura C.1: Asistente de instalación *Falcon Setup*

Segundo paso: instalar la API HDAL que brinda el fabricante, se ejecuta el archivo *2008-08-20_Novint_HDAL_SDK_2.1.3_setup.exe* de manera similar al paso anterior se inicia un asistente de instalación como se muestra en la Figura C.2, se dejan las opciones por defecto y se procede a su instalación.



Figura C.2: Asistente de instalación HDAL

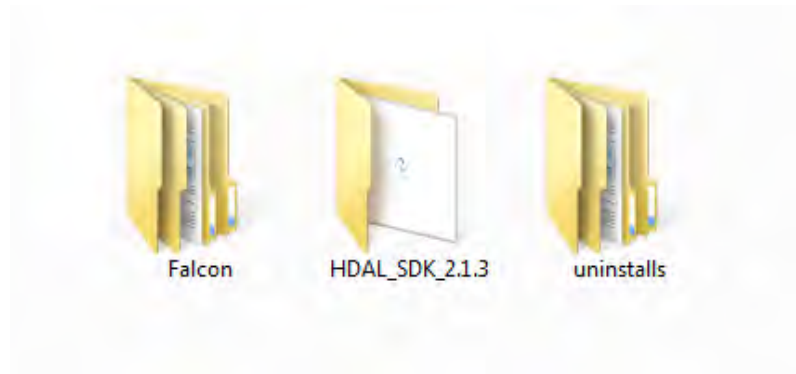


Figura C.3: Carpeta *C:/Program Files/Novint*



Figura C.4: Asistente de instalación *Haptik Library*

Si hasta este punto la instalación ha sido correcta se debe tener en la ruta *C:/Program Files/Novint* el contenido mostrado en la Figura C.3.

Tercer paso: se ejecuta el archivo *Haptik.Library.1.1rc1.win32.x86.20071009.exe* se abrirá el instalador mostrado en la Figura C.4, se dejan las opciones por defecto y se instalará en la ruta *C:/Program Files/Siena Robotics and Systems Lab/Haptik Library*. Con ésto finaliza la instalación.

Hola mundo: las APIs incluyen algunos ejemplos, en el caso de la API HDAL contiene una aplicación de realidad virtual en la ruta *C:/Program Files/Novint/HDAL_SDK_2.1.3/examples/Basic/vs2005/basic_opengl* donde se encuentra el proyecto *basic_opengl.vcproj* para Visual Studio. Solo se requiere abrir el proyecto, compilarlo y ejecutar el programa de realidad virtual con interacción háptica.

De manera similar la API *Haptik* contiene ejemplos para *C++* y para *Matlab*. Para una correcta configuración en esta última plataforma se recomienda seguir las instrucciones de la siguiente dirección web:

Dirección URL C.2 <https://manavkataria.wordpress.com/2009/11/26/tutorial-using-novint-falcon-with-matlab>

Bibliografía

- [1] Astrom, K. (2008). *Adaptive Control*. Dover Publications.
- [2] Brooks, F. y Batter, J. (1971). Grope-1: A computer display to sense of feel. *Proceedings of ITIP Congress*, pp. 759–763.
- [3] Brooks, F., Ouh-Young, M., Batter, J. J., y Kilpatrick, P. J. (1990). Project grope haptic displays for scientific visualization. *Computer Graphics*, 24:4.
- [4] Burdea, G., Zhuang, J., Roskos, E., Silver, D., y Langrana, N. (1992). A portable dextrous master with force feedback. *Presence*, 1:1.
- [5] Clavel, R. (1990). *Device for the Movement and positioning of an element in space*. United States Patent Number: 0250470.
- [6] Gough, V. E. y Whitehall, S. G. (1962). Universal tyre test machine. *Proc. 9th International Technical Congress F.I.S.I.T.A.*, pp. 117–137.
- [7] Guglielmetti, P. y Longchamp, R. (1994). A closed form inverse dynamics model of the delta parallel robot. *Proceedings from the International Federation of Automatic Control Conference on Robot Control 1994*, pp. 39–44.
- [8] Gwinnet, J. E. (1928). *Amusement device*. United States Patent Number: 1789680.
- [9] International Federation of Robotics (2016). *World Robotics Report*. <https://ifr.org/ifr-press-releases/news/world-robotic-report-2016>. Consultado: 2017-01-20.
- [10] Ioannou, P. A. (2012). *Robust Adaptive Control*. Courier Corporation.

-
- [11] Iwata, H. (1993). Pen-based haptic virtual environment. *Proc of IEEE VRAIS93*.
- [12] Levant, A. (1998). Robust exact differentiation via sliding mode technique. *Automatica*, 34:379–384.
- [13] Lung-Wen, T. (1997). *Multi-degree-of-freedom mechanisms for machine tools and the like*. United States Patent Number: 5656905.
- [14] Lung-Wen, T. (1999). *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. John Wiley & Sons.
- [15] Michelotti, M. B. (2011). *Application of the novint falcon haptic device as an actuator in real-time control*. Thesis for Master Degree, University of Illinois.
- [16] Mihelj, M. y Podobnik, J. (2012). *Haptics for Virtual Reality and Teleoperation*. Springer.
- [17] Miller, K. (1995). Experimental verification of modeling of delta robot dynamics by direct application of hamilton's principle. *IEEE International Conference on robotics and Automation*, pp. 532–537.
- [18] Miller, K. y Clavel, R. (1992). The lagrange-based model of delta-4 robot dynamics. *Robotersysteme*, 8:49–54.
- [19] Panarese, A. y Edin, B. B. (2010). A modified low-cost haptic interface as a tool for complex tactile stimulation. *Medical Engineering and Physics*, 33:386–390.
- [20] Pliego-Jiménez, J. (2010). *Análisis del movimiento de un robot bípedo de siete grados de libertad*. Tesis de Licenciatura, Universidad Nacional Autónoma de México.
- [21] Pollard, W. (1942). *Position-controlling apparatus*. United States Patent Number: 2286571.
- [22] Ragahavan, M. y Roth, B. (1995). Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators. *Transactions of the ASME Special 50th Anniversary Design Issue*, 117:71–79.

-
- [23] Salisbury, K. y Massie, T. (1994). The phantom haptic interface: A device for proving virtual objects. *ASME Winter Annual Meeting*, 55:295–300.
- [24] Slotine, J. E. (1991). *Applied Nonlinear Control*. Prentice Hall.
- [25] Spong, M. W. y Vidyagaras, M. (1989). *Robot Dynamics and Control*. John Wiley & Sons.
- [26] Stamper, R. E. (1997). *A three degree of freedom parallel manipulator with only translational degrees of freedom*. Ph.D dissertation, Univ. Maryland.
- [27] Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the Institution of Mechanical Engineers*, 180:371 – 386.
- [28] Yang, Y. (2013). *Exploring haptic feedback as an aid in design*. Thesis for Master Degree, Nanjing University of Aeronautics and Astronautics.
- [29] Zhang, D. (2009). *Parallel Robotic Machine Tools*. Springer Science and Business Media.