



UNIVERSIDAD AUTÓNOMA
DEL ESTADO DE HIDALGO



INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

CENTRO DE INVESTIGACIÓN EN TECNOLOGÍAS DE INFORMACIÓN Y SISTEMAS

MAESTRÍA EN AUTOMATIZACIÓN Y CONTROL

CONTROL DIFUSO ADAPTABLE PARA SISTEMAS DE INTERACCIÓN
HOMBRE-ROBOT

T E S I S

QUE PARA OBTENER EL GRADO DE MAESTRO EN
CIENCIAS EN AUTOMATIZACIÓN Y CONTROL

PRESENTA:

MIGUEL ÁNGEL HOYOS LEÓN

ASESORES:

DR. RAÚL VILLAFUERTE SEGURA

DR. OMAR ARTURO DOMÍNGUEZ RAMÍREZ

PACHUCA HGO., MÉXICO 18 DE ENERO DE 2016



Miguel Ángel Hoyos León

**Control Difuso Adaptable
para Sistemas de Interacción
Hombre-Robot**







UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
Instituto de Ciencias Básicas e Ingeniería
Institute of Basic Sciences and Engineering
Área Académica de Computación y Electrónica
Computer Science and Electronics Department


Mineral de la Reforma, Hgo., a 15 de enero de 2016
 Oficio No. MCAC02_2016

Lsc. Miguel Ángel Hoyos León
 PRESENTE

Por medio de la presente y en mi calidad de coordinador de la Maestría en Ciencias en Automatización y Control, del Área Académica de Computación y Electrónica (AACyE) de la Universidad Autónoma del Estado de Hidalgo (UAEH), me es grato informarle que el Jurado asignado para la revisión de su trabajo de tesis titulado: **"Control difuso adaptable para sistemas de interacción hombre-robot"**, dirigido por el Dr. Raúl Villafuerte Segura y el Dr. Omar Arturo Domínguez Ramírez, que para obtener el grado de Maestro en Ciencias en Automatización y Control fue presentado por usted, ha tenido a bien en reunión de sinodales, autorizarlo para impresión. A continuación se integran las firmas de conformidad de los integrantes del Jurado:

Dr. Joel Suárez Cansino	(Presidente)	AACyE-UAEH	
Dr. Omar López Ortega	(Secretario)	AACyE-UAEH	
Dr. Raúl Villafuerte Segura	(Vocal)	AACyE-UAEH	
Dr. Omar Arturo Domínguez Ramírez	(Suplente 1)	AACyE-UAEH	

Atentamente
 "Amor, Orden y Progreso"


Dr. Omar Jacobo Santos Sánchez
 Coordinador de la Maestría en Ciencias en Automatización y Control
 Universidad Autónoma del Estado de Hidalgo

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO



Instituto de Ciencias Básicas e Ingeniería
 Área Académica de Computación y Electrónica

c.c.p. Dr. Orlando Ávila Pozos, Director del Instituto de Ciencias Básicas e Ingeniería
 c.c.p. Dr. Hugo Romero Trejo, Jefe del Área Académica de Computación y Electrónica
 c.c.p. Expediente/ apl

Ciudad del Conocimiento
 Carretera Pachuca - Tulancingo km. 4.5
 Colonia Carboneras
 Mineral de la Reforma, Hidalgo, México, C.P. 42184
 Tel. +52 771 7172000 exts. 2250 y 2251
 omarj@uaeh.edu.mx



*El presente trabajo es el fruto del
esfuerzo de dos años, está dedicado a mis padres,
hermanos, hijos y esposa, quienes me han
enseñado, educado, me han dado su cariño,
comprensión y me han guiado por el buen camino.
Por todo eso y más, gracias.*

Gracias a

CONACYT por la beca de Maestría
otorgada durante el periodo
Agosto 2013 - Junio 2015, con CVU
543440

CITIS-UAEH, por la formación
académica brindada durante el
posgrado y el apoyo para la
realización de la presente memoria.

Agradecimientos

Agradezco a Dios, por la vida y por la excelente familia que tengo. Agradezco a mis padres y hermanos por todo el amor que me han dado, pilares importantes para mi desarrollo personal y profesional.

Agradezco a la Universidad Autónoma del Estado de Hidalgo, por ser la institución que me ha formado humana y profesionalmente.

Gracias a mi esposa, e hijos por el apoyo y amor que me han dado durante esta etapa tan importante en mi vida. Gracias a mis asesores de tesis, Dr. Raúl Villafuerte Segura y el Dr. Omar. A. Ramírez Domínguez por su tiempo, dedicación y conocimiento, a sus aportaciones para lograr este trabajo de investigación. A todos los profesores involucrados en mi formación académica .

Gracias a todos mis compañero y amigos de la maestría (Tere, Gil, Omar y Mario) por estar conmigo en una etapa tan bonita e irreplicable de mi vida. A todos los que han formado parte en mi vida gracias.

Resumen

Los sistemas robóticos, corresponden a sistemas electrónicos sujetos a diversas fuerzas interactivas modeladas y no modeladas. Cuando un robot, tiene interacción con el ambiente, el nivel de incertidumbre es alto, y no permite identificar variables que determinan completamente el comportamiento dinámico del sistema. Cuando el sistema robótico en estudio interactúa físicamente con un humano, el sistema opera en condiciones de incertidumbre provocando un deficiente desempeño. Los sistemas de control adaptables difusos ofrecen herramientas y técnicas útiles para mejorar el funcionamiento de esta clase de sistemas en condiciones de incertidumbre.

Por lo cual, en el presente trabajo de tesis se evalúa una estrategia de control difusa adaptable diseñada para mejorar la interacción entre Hombre-Robot. Así, dado que se asume desconocida la dinámica del robot, se emplea el enfoque de Mamdani para aproximar su modelo matemático. Además, para compensar las variaciones paramétricas y/o incertidumbres de la interacción Hombre-Robot, se propone la inclusión de una ley de adaptación. Por último, implementaciones vía simulación y vía experimentación son aplicados sobre un dispositivo háptico conocido como PHANToM Premium 1.0. en interacción con un operador humano, para verificar la eficiencia de los resultados teóricos propuestos en el presente trabajo.

Abstract

Robotic systems can be seen as electronic systems subject to various modeled and unmodeled interactive forces. When a robot interacts with the environment, the level of uncertainty is high and it is difficult to identify all variables that determine the dynamic behavior of the system. Clearly, when a robotic system physically interact with a human, the system operates in conditions of high uncertainty, it leads to poor performance in the system response. Adaptive fuzzy control systems provide tools and techniques to improve the performance of such systems under uncertainty.

Therefore, in this thesis the design of one adaptive fuzzy control to improve Human-Robot interaction is evaluated. Here, tools of fuzzy systems of type Mandani are used to approximate the unknown dynamics of the robot. Moreover, to compensate for the parametric variations and/or uncertainties of human-robot interaction an adaptation law is included in the control law. Finally, to verify the efficiency of the theoretical results proposed in this work, an implementation (simulation and experiments) of controller on the PHANToM Premium 1.0 haptic device with interaction of a human operator is presented.

Índice general

Agradecimientos	V
Resumen	VII
Abstract	IX
Acronimos	XI
Índice general	XI
Índice de figuras	XIII
Índice de tablas	XV
1. Introducción	1
1.1. Antecedentes	1
1.2. Planteamiento del problema	3
1.3. Hipótesis	3
1.4. Justificación	4
1.5. Objetivo General	4
1.6. Objetivos específicos.	4
1.7. Metodología	5
1.8. Organización	5
2. Antecedentes principales	7
2.1. Sistemas difusos	8
2.2. Plataforma experimental	10
2.2.1. Modelo dinámico	11
2.2.2. Modelo cinemático directo de posición	12
2.2.3. Modelo cinemático inverso de posición	13
2.2.4. Modelo cinemático directo de velocidad	14
2.2.5. Modelo cinemático inverso de velocidad	14
2.2.6. Modelo cinemático directo de aceleración	15
2.2.7. Modelo cinemático inverso de aceleración	16
2.2.8. Parámetros PHANToM Premium 1.0	16
2.2.9. Propiedades dinámicas PHANToM Premium 1.0	16
2.3. Herramientas matemáticas	17

2.4. Herramientas computacionales	19
2.4.1. Matlab	19
2.4.2. Visual Studio	21
2.5. Conclusiones de capítulo	21
3. Resultados principales	23
3.1. Diseño de la ley de control difusa adaptable	23
3.1.1. Primer esquema de control: Par calculado	24
3.1.2. Segundo esquema de control: Par calculado con identificación Difusa con aproximación de $F(x)$ y $G(x)$	25
3.1.3. Tercer esquema de control: Par calculado con identificación Difusa y ley de adaptación	26
3.2. Implementación de resultados vía simulación	30
3.2.1. Identificación del sistema	30
3.2.2. Aproximación de $F(x)$	31
3.3. Aproximación de $G(x)$	33
3.3.1. Ley de adaptación	38
3.3.2. Desempeño del sistema en tarea de seguimiento en simulación	39
3.3.3. Desempeño del sistema en tarea de seguimiento en PHANToM Pre- mium 1.0	41
4. Conclusiones y trabajos futuros	45
4.1. Conclusiones	45
4.2. Trabajos futuros	46
Bibliografía	47
A. Código C++ PHANToM Premium 1.0	53
A.1. Código en C	53
B. Código Matlab para Simulación	77
B.1. Trayectoria deseada	77
B.2. Modelo cinemático directo	79
B.3. Modelo PHANToM Premium 1.0	80
B.4. Control τ_c	81
B.5. Control τ_0	82
B.6. Control τ_r	82
B.7. $\dot{\delta}$	83
B.8. Error de seguimiento filtrado	84

Índice de figuras

2.1. Diagrama de bloques de control difuso.	8
2.2. PHANToM Premium 1.0.	10
2.3. Representación MCDP.	12
2.4. Software Matlab.	20
2.5. Entorno de programación Visual Studio.	21
3.1. Funciones de pertenencia para posición articular q_2	31
3.2. Función de pertenencia para posición articular q_3	32
3.3. Funciones de pertenencia para posición articular q_2	33
3.4. Desempeño en espacio de trabajo.	39
3.5. Posición angular real y deseada.	40
3.6. Error de posición articular.	40
3.7. Posición angular.	41
3.8. Error de posición articular.	42
3.9. Trayectoria en el espacio de trabajo.	42
3.10. Par de control articular.	43
3.11. Parámetros θ	43

Índice de tablas

2.1. Parámetros PHANToM Premium 1.0.	16
3.1. Parámetros de funciones de pertenencia.	33

Capítulo 1

Introducción

En este capítulo se muestra una descripción general del proyecto realizado, para tener una mejor comprensión de la investigación desarrollada se presentan antecedentes, planteamiento del problema, hipótesis, se hace la justificación, se plantean los objetivos, y se menciona la metodología.

1.1. Antecedentes

Los sistemas de interacción hombre-máquina, y particularmente sistemas de interacción hombre-robot, nacen como una necesidad de resolver problemas de inmersión virtual activa (ambientes virtuales dinámicos deformables), para recrear escenarios remotos en la estación del usuario, teleoperar sistemas robóticos complejos ante la existencia de riesgos humanos, establecer plataformas para evaluación biomecánica y entrenamiento, así como sistemas de visualización científica interactiva y simulación de eventos dinámicos complejos [1], [2], [3], [4]. La comunidad científica y académica del área, propone continuamente diversos esquemas y plataformas para lograr metas como las ya descritas, y promueve el diseño a partir de los alcances de la tarea y aplicación. Una plataforma robótica está compuesta por los subsistemas siguientes:

- * Electromecánico.
- * Percepción.
- * Comunicación.
- * Decisión.

* Control.

La relación entre ellos, las tecnologías de integración y los procedimientos algorítmicos empleados contribuyen en beneficio del desempeño y garantizan eficiencia operativa [5]. Sin embargo, cuando estos sistemas interactúan con fuerzas externas, las condiciones de operación cambian radicalmente y la diferencia se describe a continuación:

- Sistema robótico en movimiento libre: no existe presencia de fuerzas externas (es requerido el control de posición y velocidad).
- Sistema robótico en movimiento restringido: presencia de una superficie de contacto conocida, o calculada en tiempo real en interacción con el efector final (es requerido el control de posición, velocidad y fuerza de contacto).

En la última década se ha prestado especial atención al diseño, construcción y control de sistemas robóticos que permitan al ser humano interactuar activamente con un ambiente de visualización virtual, incorporando estímulos táctiles y así permitir al ser humano tener una percepción de mundos virtuales recreados a partir de ambientes artificiales y/o reales, [1], [6], [7]. A este tipo de sistemas se les conoce como interfaces hápticas, las cuales están constituidas por un operador humano, un ambiente de visualización virtual y un sistema electromecánico denominado dispositivo háptico.

Las estrategias de control y planificación para cada uno de los casos, consideran en su diseño a la dinámica de fuerzas de interacción y en el mejor de los casos a la incertidumbre presente en la carga en el efector final, en las fuentes de energía empleadas en interfaces electrónicas de potencia, y en presencia de objetos susceptibles de colisión en el espacio de trabajo (ambiente del robot). Los cambios en la dinámica más relevantes corresponden a la tribología articular, dependen de la fricción dinámica compleja y de las vibraciones mecánicas; éstas pueden ser reducidas, compensadas o estabilizadas a través de estrategias de control de movimiento y fuerza [8], [9]. Cuando el sistema robótico tiene contacto con el operador humano, el esquema corresponde a una interfaz háptica, y se asocia a un tipo especial de sistema robótico con movimiento restringido; en estas condiciones la dinámica del humano representa incertidumbre a la dinámica del robot en lazo cerrado, y aunque en condiciones de salud del operador, ésta puede ser acotada a través de una estrategia de control robusta, las condiciones no son las mismas bajo circunstancias de una discapacidad motriz (movimiento involuntario activo, movimiento espástico, o movimiento pasivo) [10]. En la literatura, se abordan esquemas de diseño e integración de plataformas experimentales con diversos propósitos, que atienden situaciones de este tipo, [11], [12], [13], [14]. En [15] se presenta una herramienta integral basada en redes ZigBee para

rehabilitación robótica de miembro superior. En este trabajo de investigación se presenta una modificación y una aplicación de los resultados teóricos desarrollados en [16], [17], [18]. A diferencia de los resultados previos, en la presente propuesta se modifica el error de seguimiento filtrado para obtener una ley de control difuso adaptable que mejora el desempeño de los sistemas de interacción hombre-robot en tareas de seguimiento, además se presenta la implementación de los resultados teóricos obtenidos sobre un dispositivo háptico PHANToM Premium 1.0. El desempeño apropiado de la interacción hombre-robot es garantizado mediante un análisis de estabilidad del sistema en lazo cerrado en el marco de Lyapunov.

1.2. Planteamiento del problema

Sea un sistema no lineal MIMO de la forma:

$$\begin{aligned} y_1^{(r_1)} &= f_1(x) + \sum_{j=1}^p g_{1j}(x)u_j, \\ &\vdots \\ y_p^{(r_p)} &= f_p(x) + \sum_{j=1}^p g_{pj}(x)u_j, \end{aligned} \tag{1.1}$$

donde r_i es el orden del sistema, el vector de salida $y = [y_1, \dots, y_p]^T$, el vector de estado se asume como disponible para su medición $x = [y_1, \dot{y}_1, \dots, y_1^{(r_1-1)}, \dots, y_p, \dot{y}_p, \dots, y_p^{(r_p-1)}]^T$, el vector de entradas de control $u = [u_1, \dots, u_p]^T$ y $f_i(x), g_{ij}(x), i, j = 1, \dots, p$ son funciones no lineales, suaves y desconocidas.

La mayoría de los controles aplicados a la interacción física entre hombre-robot valoran el desempeño del robot en lazo cerrado, durante la interacción se pueden medir energía y la potencia que demanda, sin embargo esto obliga al dispositivo a trabajar bajo ciertas condiciones para extender la vida útil de sus componentes pero esto no garantiza confort al operador humano cuando se ejecuten tareas de exploración o guiado háptico y si las condiciones no favorecen al usuario, el dispositivo realiza la tarea inadecuadamente o la da por terminada.

1.3. Hipótesis

El empleo de estrategia de control difuso adaptables en tareas de seguimiento, pueden mejorar el desempeño en sistemas de interacción física hombre-robot en comparación al

uso de controladores clásicos. Dado un sistema no lineal MIMO de la forma

$$\begin{aligned} y_1^{(r1)} &= f_1(x) + \sum_{j=1}^p g_{1j}(x)u_j, \\ &\vdots \\ y_p^{(rp)} &= f_p(x) + \sum_{j=1}^p g_{pj}(x)u_j, \end{aligned}$$

donde las funciones $f_i(x)$ y $g_{ij}(x)$, $i = 1, \dots, p$, se asumen como desconocidas, es posible obtener un controlador difuso adaptable que estabilice el sistema en lazo cerrado y garantice el seguimiento de la trayectoria deseada.

1.4. Justificación

Actualmente las leyes de control clásicas propuestas en diferentes literaturas para manipular un sistema robótico resulta eficiente bajo condiciones ideales, sin embargo cuando estos dispositivos interactúan físicamente con un operador humano el sistema opera bajo ciertas condiciones de incertidumbre provocando un mal desempeño en la ley de control. La inteligencia artificial, como son los sistemas difusos, ofrecen herramientas y técnicas útiles para mejorar el funcionamiento y desempeño de los sistemas robóticos bajo condiciones de incertidumbre. Por lo cual, en el presente trabajo de investigación se diseña e implementa una ley de control difusa para mejorar el desempeño de interacción hombre-robot, el cual se implementa a un dispositivo háptico PHANToM Premium 1.0 en tareas de guiado háptico sujetas a incertidumbres dinámicas en lazo.

1.5. Objetivo General

Diseñar una estrategia de control difuso adaptable para mejorar el desempeño en la interacción física entre hombre-robot con un dispositivo háptico PHANToM Premium 1.0 en tareas de guiado háptico sujeta a incertidumbre que se genera en la dinámica cuando un operador humano esté en lazo.

1.6. Objetivos específicos.

Los objetivos específicos de la tesis son los siguientes:

1. Realizar una revisión bibliográfica de sistemas robóticos como los dispositivos hápticos empleando motores de búsqueda para conocer esta clase de sistemas.

2. Estudiar sistemas robóticos, para obtención de modelos matemáticos mediante Euler-Lagrange.
3. Estudiar sistemas difusos, para aproximar el modelo, mediante sistemas difusos en el enfoque de Mamdani.
4. Agregar una ley de adaptación, para compensar la incertidumbre hombre-robot, mediante mínimos cuadrados recursivos.
5. Estudiar el modelo dinámico PHANToM Premium 1.0, para simulación, mediante Matlab y Simulink.
6. Implementar la ley de control difuso adaptable en PHANToM Premium para evaluar estrategia de control, mediante codificación en lenguaje C++.

1.7. Metodología

La metodología empleada en el presente trabajo de investigación inicia con el estudio de conocimientos teóricos sobre Euler-Lagrange para obtener el modelo matemático del dispositivo háptico, utilizar técnicas de control difuso en el marco de Mamdani para diseñar una ley de control difuso. Se integrará una ley de adaptación con el fin de compensar incertidumbre generada al momento de que un operador humano interactúe con el dispositivo háptico. Para simular el dispositivo háptico se hace uso de Matlab y su librería de programación Simulink, para poder ser implementado en el dispositivo háptico.

1.8. Organización

El presente trabajo de investigación se organiza en 4 capítulos. En el **capítulo 2** se presentan las herramientas matemáticas, definiciones, teoremas, que se ocupan en el momento de diseñar la estrategia de control difuso adaptable, también se describe la plataforma experimental PHANToM Premium 1.0, sus características principales, tanto como su modelo cinemático y dinámico. Se describen las herramientas computacionales que se usan tanto para simular el dispositivo háptico y cómo programar el PHANToM Premium 1.0.

Se continúa con el **capítulo 3** el cual se dedica a desarrollar la ley de control difuso adaptable, presentar el análisis de estabilidad en el sentido de Lyapunov, la convergencia que se necesita tener con la trayectoria deseada y cómo se propone la ley de adaptación,

demás es parte fundamental de este trabajo, ya que corresponde a los resultados numéricos y experimentales realizados en PHANToM Premium 1.0 que se obtuvo de validar la ley de control obtenida con ayuda de las herramientas que se abordan en **capítulo 2**. Se concluye el capítulo con comentarios y conclusiones.

Para finalizar en el **capítulo 4** se presentan resultados y conclusiones que se generan en este trabajo de investigación, se agregan los trabajos futuros para dar seguimiento a esta línea de investigación a corto y mediano plazo.

Capítulo 2

Antecedentes principales

En este capítulo se presenta los conceptos básicos, además el modelo cinemático y dinámico de la plataforma experimental y también las herramientas tanto matemáticas como computacionales que se necesitan para poder desarrollar el presente trabajo de investigación.

La lógica difusa es considerada como una extensión de la lógica clásica y fue propuesta por Lotfi A. Zadeh en 1965 en la Universidad de California en Berkeley, basado en el artículo "*Fuzzy sets*" [42]. A diferencia de los conjuntos clásicos que se le asigna un valor si pertenece o no pertenece al conjunto, la lógica difusa utiliza el grado de pertenencia en un intervalo de $[0, 1]$ de acuerdo al problema que se está abordando, por ejemplo "*la temperatura es baja*", "*la velocidad es alta*". Puede entenderse a lo difuso como la posibilidad de asignarle más valores de verdad a los conjuntos clásicos de "*falso*" y "*verdadero*". Por estas razones es que ciertos sistemas de lógica difusa aparentan una labor de aprendizaje, y son excelentes mecanismos de control de procesos, es decir, los controladores difusos son capaces de estabilizar sistemas no lineales [43]. Cabe destacar que a medida que la complejidad de un problema crece la posibilidad de analizarlo en términos precisos disminuye considerablemente. Se debe hacer notar que el modelo matemático es una aproximación de un sistema físico y que no puede representar todas las variables dinámicas del sistema pues resultan en la mayoría de los casos muy difíciles de modelar [37]. Cuando se habla de adaptación, se habla de una característica fundamental que poseen los organismos vivos, ya que tratan de mantener un equilibrio físico en condiciones ambientales variables. En el área de control se dice que un sistema mide las características dinámicas de la planta y puede hacer variar los parámetros ajustables del sistema. En la Figura 2.1 se ilustra el diagrama de bloques y sus componentes básicos de un sistema

difuso.

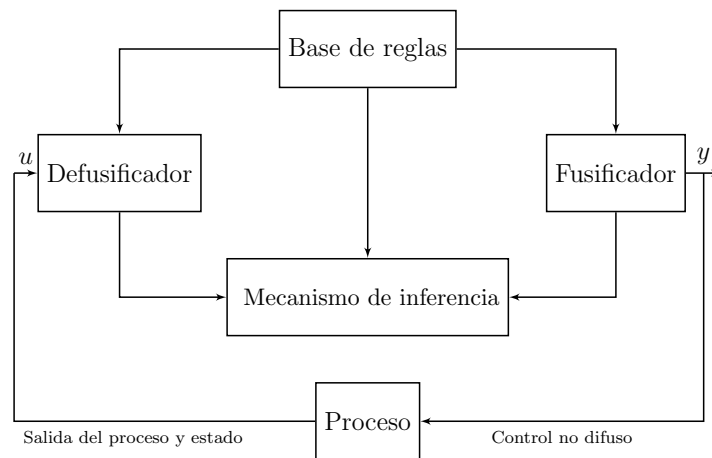


Figura 2.1: Diagrama de bloques de control difuso.

2.1. Sistemas difusos

Los sistemas difusos se caracterizan por su estructura que consta de tres bloques principales y están basados en conocimiento de un experto. La base del conocimiento consiste en reglas difusas con un *antecedente* y *consecuente*. Un ejemplo de estas premisas de tipo **SI-ENTONCES** es:

SI la *velocidad* del auto es *alta*, **ENTONCES** *bajar* la *velocidad*.

A continuación se mencionan las tres partes principales de los sistemas difusos.

Fusificación

La fusificación se utiliza para cuantificar la información de la base de reglas, en otras palabras, es el proceso de asignar o calcular un valor que represente un grado de pertenencia para todos los conjuntos difusos definidos, sobre alguna variable de entrada, dado que el valor actual de dicha variable no se toma como una cantidad difusa.

El fusificador más utilizado en el área de control automático es el solitón cuya función de membresía está dado por:

$$\mu_A(x) = \begin{cases} 1 & \text{si } x = A \\ 0 & \text{en otro caso.} \end{cases} \quad (2.1)$$

Inferencia difusa

El mecanismo de inferencia difusa es la unidad central de cualquier controlador difuso. De manera general el comportamiento dinámico es caracterizado por un conjunto de reglas difusas del tipo:

$$\text{Si } x \text{ es } A, \text{ entonces } y \text{ es } B, \quad (2.2)$$

donde A y B son valores lingüísticos definidos por el universo de discurso U_x y U_y , respectivamente.

Una acción de control es determinada por las entradas observadas, las cuales representan el estado del proceso a ser controlado mediante las reglas difusas de control.

Los métodos más utilizados de mecanismos de inferencia son los siguientes:

- Implicación Dienes-Rescher [44]: Esta implicación es una relación difusa llamada Q_D en $A \times B$ con función de membresía

$$\mu_{Q_D}(x, y) = \text{máx}[1 - \mu_A(x), \mu_B(y)]. \quad (2.3)$$

- Implicación de Lukasiewicz [23]: La relación difusa está dada por Q_L en $A \times B$ con función de membresía

$$\mu_{Q_L}(x, y) = \text{mín}[1, 1 - \mu_A(x) + \mu_B(y)]. \quad (2.4)$$

- Implicación de Zadeh [23], [45], [46]: La regla difusa (2.2) es interpretada como una relación difusa dada Q_z en $A \times B$ con función de membresía

$$\mu_{Q_z}(x, y) = \text{máx}[\text{mín}(\mu_A(x), \mu_B(y)), 1 - \mu_A(x)]. \quad (2.5)$$

- Implicación de Gödel [23], [25]: La regla difusa (2.2) es interpretada como una relación Q_G en $A \times B$ con función de membresía

$$\mu_{Q_G}(x) = \begin{cases} 1 & \text{si } \mu_A(x) \leq \mu_B(x), \\ \mu_B(x) & \text{en otro caso.} \end{cases} \quad (2.6)$$

- Implicación Mamdani [23],[25],[47]: Aquí la regla difusa 2.2 es interpretada como una relación difusa dada Q_M en $A \times B$ con función de membresía

$$\mu_{Q_M}(x, y) = \text{mín}[\mu_A, \mu_B], \mu_{Q_M} = \mu_A * \mu_B. \quad (2.7)$$

Las cinco ecuaciones anteriormente mencionadas tienen diferente interpretación de la regla difusa (2.2), generalmente la más utilizada en aplicaciones en los sistemas difusos es la Mamdani, ecuación (2.7).

Defusificación

La defusificación es un proceso matemático usado para convertir un conjunto difuso en un número real, (el cual es la salida de la inferencia difusa). La tarea del defusificador consiste en reducir los valores de pertenencia que se asocian con la salida de datos numéricos de la solución del problema.

2.2. Plataforma experimental

En esta sección se presenta una descripción de la plataforma de trabajo, el dispositivo háptico PHANToM Premium 1.0 (2.2), el modelo matemático, y los modelos de cinemáticos inverso de posición, directo de posición, inverso de velocidad y directo de velocidad. PHANToM Premium 1.0 es un dispositivo háptico, el cual permite la interacción hombre-máquina con el objetivo de poder tener tacto con un ambiente virtual. Un dispositivo háptico que cuente con una cadena de eslabones articulados puede ser modelado como un robot manipulador utilizando las mismas técnicas que con un robot manipulador [33], [17]. La interfaz PHANToM Premium 1.0 es un dispositivo háptico que se puede configurar de acuerdo a la interacción que se realice con el operador humano, ésta puede ser tipo mayordomo. El dispositivo háptico PHANToM Premium 1.0 puede ser alimentado por una fuente de $50Hz$ y $60Hz$ en C.A., a 115 y 230 voltios, con corriente de 5 y 2.5 amperios respectivamente, la fuerza máxima es de $8.5N$, cuenta con detección de posiciones x, y, z , la resolución nominal por pulso es de $0.03mm$ y el protocolo de comunicación es por puerto paralelo [36].

El modelo matemático de un robot manipulador está basado en la cinemática y en la dinámica de su mecanismo de eslabones articulados.

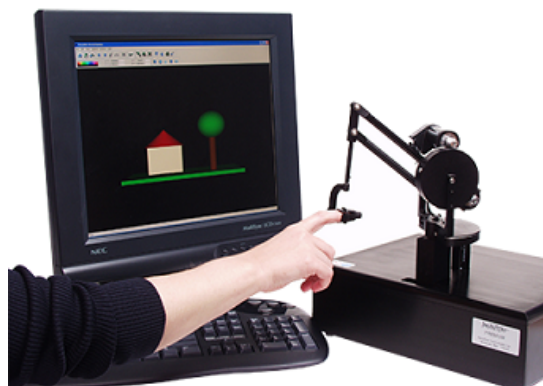


Figura 2.2: PHANToM Premium 1.0.

2.2.1. Modelo dinámico

La cinemática estudia el movimiento del robot con respecto a un sistema de referencia seleccionado. El estudio que realiza la cinemática se interesa por la descripción analítica del movimiento espacial del robot, y en particular las relaciones existentes entre la posición y orientación del efector final y los valores que toman sus coordenadas articulares [9].

$$\tau = H(q)\ddot{\mathbf{q}} + C(q, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(q), \quad (2.8)$$

donde $H(q)$ es la matriz de fuerzas inerciales, $C(q, \dot{\mathbf{q}})\dot{\mathbf{q}}$ es la matriz de Coriolis y $G(q)$ es el vector de fuerza de gravedad [48].

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & 0 & c_{23} \\ c_{13} & c_{32} & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ g_2 \\ g_3 \end{bmatrix},$$

$$m_{11} = [(0.5l_1^2 + 0.125l_2^2)ma + (0.125l_1^2)mc] + 0.125l_1^2(4ma + mc)\cos(2q_2) + m'_{11},$$

$$m'_{11} = (-0.125l_2^2ma + 4l_3^2mc)\cos(2q_3) + 0.125l_1(l_2ma + l_3mc)\cos(q_2)\sin(q_3),$$

$$m_{22} = l_1^2(ma + 0.25mc),$$

$$m_{23} = -0.5l_1(l_2ma + l_3mc)\sin(q_2 - q_3),$$

$$m_{32} = m_{23},$$

$$m_{33} = 0.25l^2ma + l_3^2mc,$$

$$c_{11} = 0.25[-2\sin(q_2)[l_1^2(4ma + mc)\cos(q_2)] + 2l_1(l_2ma + l_3mc)\sin(q_3)\dot{q}_3] + c'_{11},$$

$$c'_{11} = 0.252\cos(q_3)[2l_1(l_2ma + l_3mc)\cos(q_2) + (l_2^2ma + 4l_3mc)\sin(q_3)]\dot{q}_3,$$

$$c_{12} = -0.25[l_1^2(4ma + mc)\sin(2q_2) + 4l_1(l_2ma + l_3mc)\sin(q_2)\sin(q_3)]\dot{q}_1,$$

$$c_{13} = -0.125[-4l_1(l_2ma + l_3mc)\cos(q_2)\cos(q_3) - (l_2^2ma + 4l_3^2mc)\sin(2q_3)]\dot{q}_1,$$

$$c_{21} = -c_{12},$$

$$c_{23} = 0.5l_1(l_2ma + l_3mc)\cos(q_2 - q_3)\dot{q}_3,$$

$$c_{31} = -c_{13},$$

$$c_{33} = 0.5l_1(l_2ma + l_3mc)\cos(q_2 - q_3)\dot{q}_2.$$

2.2.2. Modelo cinemático directo de posición

Para la localización de un objeto en el espacio tridimensional con respecto a un objeto de referencia fijo se utiliza fundamentalmente el álgebra vectorial y matricial [49]. La

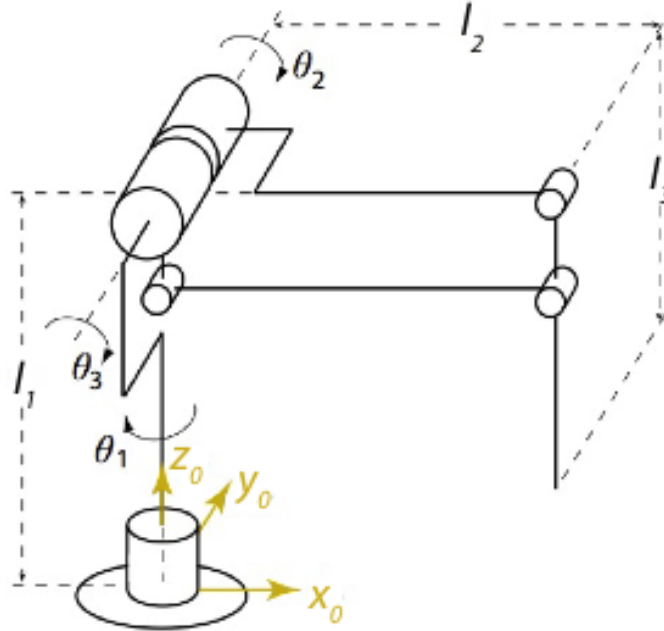


Figura 2.3: Representación MCDP.

resolución del problema de la cinemática directa consiste en encontrar la relación que permita conocer la localización del efector final a partir de los valores de sus coordenadas articulares [50]. La ecuación siguiente describe al modelo cinemático directo de posición:

$$x = f(q), \quad (2.9)$$

donde $x \in \mathbb{R}^{n \times 1}$ son las coordenadas del sistema, $q \in \mathbb{R}^{n \times 1}$ es el vector de variables articulares, n es el número de de variables articulares.

El modelo cinemático directo de posición para el PHANToM Premium 1.0 está descrito por la siguiente ecuación:

$$\begin{aligned} x &= s_1(l_1c_2 + l_2s_3), \\ y &= l_2 - l_2c_3 + l_1s_2, \\ z &= -l_1 + c_1(l_1c_2 + l_2s_3). \end{aligned}$$

donde $s_1 = \text{sen}(q_1)$, $s_2 = \text{sen}(q_2)$, $s_3 = \text{sen}(q_3)$, $c_1 = \text{cos}(q_1)$, $c_2 = \text{cos}(q_2)$, $c_3 = \text{cos}(q_3)$.

2.2.3. Modelo cinemático inverso de posición

La cinemática inversa consiste en hallar los valores en que deben orientarse las coordenadas articulares $q = [q_1, q_2, \dots, q_n]^T$ para que el efector final se posicione y se oriente en una localización previamente establecida. El dispositivo háptico PHANToM Premium 1.0 cuenta con 3 GDL, tiene una estructura plana, este hecho facilita la resolución del problema [51], [52].

La cinemática inversa presenta más problemas que la directa, se mencionan algunos problemas:

- Un punto previamente fijado, por su lejanía está fuera del alcance del dispositivo háptico.
- Un punto está fuera del alcance del dispositivo háptico, debido a que sus eslabones chocan entre si (problemas geométricos).
- Un punto puede alcanzarse mediante dos o más configuraciones o valores del vector de articulación.

El modelo cinemático inverso de posición corresponde a la inversa de f^{-1} del modelo cinemático, generalmente un robot se controla en el espacio de las variables de articulación, mientras que el efector final suele expresarse en el espacio de coordenadas cartesianas. Es por eso y con el fin de controlar la posición y orientación del efector final de un robot para alcanzar un objeto, es más frecuente el uso de la solución al problema de la cinemática inversa, el cual se describe como:

$$q = f^{-1}(x), \quad (2.10)$$

donde $x \in \mathbb{R}^{n \times 1}$ son las coordenadas del sistema, $q \in \mathbb{R}^{n \times 1}$ es el vector de variables articulares, n es el número de variables articulares. Para el caso del dispositivo háptico se asume $n = m$, lo cual quiere decir que existe un número finito de soluciones de la matriz Jacobiana fuera de singularidades y puede existir infinito número de soluciones cuando la matriz Jacobiana es singular. Para el dispositivo háptico PHANToM Premium 1.0 el

modelo cinemático inverso de posición está dado por:

$$\begin{aligned}
\psi_1 &= \tan^{-1} 2(x, z + l_1) \\
R &= \sqrt{x^2 + (z + l_1)^2} \\
r &= \sqrt{x^2 + (z + l_1)^2 + (y - l_2)^2} \\
\beta &= \tanh^{-1} 2(y - l_2 + R) \\
\gamma &= \cos^{-1} \left(\frac{l_1^2 + r^2 - l_2^2}{2l_1 r} \right) \\
\psi_2 &= \gamma + \beta \\
r^2 &= l_1^2 + l_2^2 - 2l_1 l_2 \cos(\alpha) \\
\alpha &= \cos^{-1} \left(\frac{l_1^2 + l_2^2 - r^2}{2l_1 l_2} \right). \\
\psi_3 &= \psi_2 + \alpha - \frac{\pi}{2}.
\end{aligned}$$

2.2.4. Modelo cinemático directo de velocidad

El modelo cinemático directo de velocidad consiste en una relación entre las velocidades angulares y la velocidad operacional. La matriz Jacobiana depende de la configuración del robot y permite detectar singularidades, para obtener el modelo cinemático directo de velocidad se deriva la ecuación (2.9) y se obtiene [9]

$$\dot{x} = J(q)\dot{q}, \quad (2.11)$$

donde $\dot{x} \in \mathbb{R}^{n \times 1}$ son las coordenadas del sistema, $q \in \mathbb{R}^{n \times 1}$ es el vector de variables articulares, n es el número de variables articulares.

El Phantom Premium 1.0. tiene un modelo cinemático directo de velocidad descrito por [53]

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} l_1 c_1 c_2 + l_2 s_3 c_1 & -l_1 s_1 s_2 & l_2 s_2 c_3 \\ 0 & l_1 c_2 & l_2 s_3 \\ -(l_1 s_1 c_2 + l_2 s_1 s_3) & -l_1 s_2 c_1 & l_2 c_1 c_3 \end{bmatrix} \begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \end{bmatrix}.$$

2.2.5. Modelo cinemático inverso de velocidad

El modelo cinemático inverso de velocidad está descrito por:

$$\dot{\psi} = J^{-1}\dot{x},$$

$$J^{-1} = \frac{\text{adj}(J)^T}{\det(J)},$$

$$\text{adj}(J) = \begin{bmatrix} l_1 l_2 c_1 c_{2-3} & 0 & -l_1 l_2 s_1 c_2 c_{2-3} \\ -l_2 s_1 s_3 K & l_2 c_3 K & -l_2 s_3 c_1 K \\ l_1 s_1 c_2 K & l_1 s_2 K & l_1 c_1 c_2 K \end{bmatrix},$$

con $K = l_1 c_2 + l_2 s_3$ y $\det(J) = l_1 l_2 l_1 s_2 s_3 c_2 + l_1 c_2^3 c_3 + l_2 s_3 c_2 c_3 - l_2 s_2 c_3^2$, donde $\det(J)$ es el determinante del Jacobiano, $\text{adj}(J)$ es la matriz adjunta del Jacobiano.

2.2.6. Modelo cinemático directo de aceleración

El modelo se obtiene de derivar respecto del tiempo a la ecuación (2.11), este modelo permite conocer los cambios de velocidad y se rige por la siguiente ecuación:

$$\ddot{x} = j(q)\ddot{q} + \dot{j}\dot{q}, \quad (2.12)$$

donde el vector de coordenadas de aceleración operacionales está dado por $\ddot{x} \in \mathbb{R}^n$, $\ddot{q} \in \mathbb{R}^n$ es el vector de aceleraciones articulares y $\dot{j}(\dot{q}) \in \mathbb{R}^{n \times n}$ se define como la derivada temporal de la matriz Jacobiana.

Para el caso del PHANToM Premium 1.0 la ecuación (2.12), $\dot{j}(\dot{q})$ está descrita a continuación:

$$\dot{J}(\dot{q}) = \begin{bmatrix} \dot{j}_{11} & \dot{j}_{12} & \dot{j}_{13} \\ \dot{j}_{21} & \dot{j}_{22} & \dot{j}_{23} \\ \dot{j}_{31} & \dot{j}_{32} & \dot{j}_{33} \end{bmatrix}, \quad (2.13)$$

donde

$$\begin{aligned} \dot{j}_{11} &= -(l_1 s_1 c_1 + l_2 s_1 s_3) \dot{q}_1 - l_1 c_2 s_1 \dot{q}_2 + l_1 c_1 - c_3 \dot{q}_3, \\ \dot{j}_{12} &= -l_1 s_2 c_1 \dot{q}_1 - l_1 s_2 c_1 \dot{q}_2, \\ \dot{j}_{13} &= l_2 c_1 c_3 \dot{q}_1 - l_2 s_1 s_3 \dot{q}_3, \\ \dot{j}_{21} &= 0, \\ \dot{j}_{22} &= -l_1 s_2 \dot{q}_2, \\ \dot{j}_{23} &= (l_1 s_1 c_1 + l_2 s_1 s_3) \dot{q}_1 + l_1 c_2 s_1 \dot{q}_2 + l_1 c_1 - c_3 \dot{q}_3, \\ \dot{j}_{31} &= -l_2 c_1 c_3 \dot{q}_1 + l_2 s_1 s_3 \dot{q}_3, \\ \dot{j}_{32} &= l_1 s_2 c_1 \dot{q}_3 - l_2 s_1 c_2 \dot{q}_1, \\ \dot{j}_{33} &= -l_2 c_1 s_3 \dot{q}_1 - l_2 s_3 c_1 \dot{q}_3. \end{aligned}$$

2.2.7. Modelo cinemático inverso de aceleración

El modelo cinemático inverso de aceleración se deduce de la ecuación (2.12) y se representa de la siguiente manera

$$\ddot{q} = J^{-1}(q) \left[\ddot{x} - \dot{J}(\dot{q}) \right]. \quad (2.14)$$

2.2.8. Parámetros PHANToM Premium 1.0

Los parámetros se describen en la siguiente tabla

Parámetro	Valor	Unidad
m_a	$17.5 \times 10_{-3}$	Kg
m_c	$10.4 \times 10_{-3}$	Kg
m_{be}	0.2214	Kg
m_{df}	0.01106	Kg
L_1	139.7	mm
L_2	139.7	mm
L_3	32.5	mm
L_4	36.8	mm
L_5	52.7	mm
g	9.81	$\frac{m}{s^2}$

Tabla 2.1: Parámetros PHANToM Premium 1.0.

2.2.9. Propiedades dinámicas PHANToM Premium 1.0

Existen diversas propiedades que el dispositivo háptico cumple, a continuación se mencionan sólo las que se emplean en este trabajo de investigación [43], [54]. **Propiedad 1**, la matriz de inercias es definida positiva [35], [55]:

$$x^T H(q)x > 0. \quad (2.15)$$

Propiedad 2, antisimetría dice que existe una relación entre la matriz de inercias $H(q)$ y la matriz de Coriolis $C(q, \dot{q})$ [55], [35], [48].

$$x^T \left(\dot{H}(q) - 2C(q, \dot{q}) \right) x \approx 0. \quad (2.16)$$

Propiedad 3, la matriz de fuerzas inerciales es simétrica [55], [35].

$$H(q) = H(q)^T. \quad (2.17)$$

2.3. Herramientas matemáticas

En esta sección se presentan las herramientas matemáticas más importantes que se utilizan en el presente trabajo de tesis [56], [31], [57], [50].

Definición 2.1 Considerar el sistema de coordenadas generalizadas q y sus derivadas \dot{q} y \ddot{q} [19], [57], [47],

$$\ddot{q} = f(q, \dot{q}, \tau), \quad (2.18)$$

suponiendo que el sistema tiene restricciones en sus límites de operación. Si la condición de restricción puede ser expresada como una ecuación que relacionen a las coordenadas generalizadas de la forma:

$$h(q, t) = 0. \quad (2.19)$$

Definición 2.2 [53] Un punto $x = x^*$ en espacio de estados es un punto de equilibrio del sistema $\dot{x} = f_0(\bar{x}) + f_1(\bar{x})u$, con $u = 0$, si tiene la propiedad de que la condición inicial del estado x^* , permanece en x^* para todo tiempo t , es decir, x^* es punto de equilibrio del sistema $\dot{x} = f_0(\bar{x}) + f_1(\bar{x})u$.

Definición 2.3 [42], [45] Se considera el sistema en coordenadas generalizadas y sus derivadas \dot{q} y \ddot{q} ,

$$\ddot{q} = f(q, \dot{q}, \tau), \quad (2.20)$$

suponiendo que el sistema tiene restricciones en sus límites de operación. Si la condición de restricción puede ser expresada como una ecuación que relaciona a las coordenadas generalizadas de la forma.

$$h(q, t) = 0, \quad (2.21)$$

entonces la restricción es llamada holonómica y puede ser integrable.

Definición 2.4 [42], [58] Si un punto de equilibrio asintóticamente o exponencialmente estable para cualquier condición inicial, el punto de equilibrio es llamado asintóticamente estable en la trayectoria. También es llamado globalmente estable.

Definición 2.5 [42], [58], [47] Una función escalar continua $V(x)$ es llamada definida positiva localmente si, $V(x) = 0$ y sobre la bola B_{R_0} :

$$\forall x \neq 0, V(x) > 0. \quad (2.22)$$

Considere el sistema de n entradas y p entradas

$$\dot{x} = Ax + Bu, \quad (2.23)$$

con las matrices constantes $A \in \mathbb{R}^{n \times n}$ y $B \in \mathbb{R}^{n \times p}$

Definición 2.6 (Controlabilidad) La ecuación de estados (2.23) o el par (A, B) , se dice controlable si para cualquier estado inicial $x(0) = x_0 \in \mathbb{R}^n$ y cualquier estado final $x_1 \in \mathbb{R}^n$, existe una entrada que transfiere el estado x de x_0 a x_1 en tiempo finito. En caso contrario la ecuación (2.23), o el par (A, B) , se dice no controlable [54], [59], [60].

Considerando el sistema lineal:

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du \end{aligned}$$

donde $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$, $C \in \mathbb{R}^{q \times n}$, $D \in \mathbb{R}^{q \times p}$.

Teorema 2.1 La ecuación de estado (2.23) es observable si para cualquier estado inicial $x(0)$ desconocido, existe un tiempo finito t_1 tal que el conocimiento de la entrada u y la salida y sobre el intervalo $[0, t_1]$ es suficiente para determinar en forma única el estado inicial $x(0)$. En caso contrario el sistema no es observable [59], [62].

Teorema 2.2 Sea $\phi(\cdot)$ una función continua, monótona creciente y acotada. Denotemos I_n el hipercubo unitario n -dimensional $[0, 1]^n$ y por $C(I_n)$ el espacio de las funciones continuas sobre I_n . Entonces, dado una función $f \in C(I_n)$ y un $\epsilon > 0$, existe un entero m y conjunto de constantes reales, α_i, θ_j y w_{ij} , donde $i = 1, \dots, m$, $j = 1, \dots, n$ tales que se puede definir [54], [59], [60]

$$F(x_1, \dots, x_n) = \sum_{i=1}^m \alpha_i \phi \left(\sum_{j=1}^n w_{ij} x_j - \theta_i \right) \quad (2.24)$$

como una aproximación de la función f , que cumple $\| F(x_1, \dots, x_n) - f(x_1, \dots, x_n) \| < \epsilon$, $\forall (x_1, \dots, x_n) \in I_n$.

Teorema 2.3 [49], [50], [51] Sea $x = 0$ un punto de equilibrio de $\dot{x} = f(x)$. Sea $V D: \rightarrow \mathbb{R}$ una función diferenciable tal que:

- $V(x)$.
- $V(x) > 0$ para $x \in D - \{0\}$.
- $\dot{V}(x(t)) = -w(\|x(t)\|) \leq 0$ para todo $x \in \{D\}$ y $w(\|x\|)$ definida positiva. Entonces $x = 0$ es estable.

Se emplean las ecuaciones de Euler-Lagrange para obtener un modelo. Para obtener de un modelo dinámico de un robot manipulador por medio de las ecuaciones de Euler-Lagrange se describen los siguientes pasos:

- Calcular energía cinética total $\mathcal{K}(q, \dot{q})$.
- Calcular energía potencial total $\mathcal{U}(q)$.
- Calcular el Lagrangiano: \mathcal{L} .
- Desarrollar las ecuaciones de Euler-Lagrange para obtener

$$\mathcal{E}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) + \mathcal{U}(q), \quad (2.25)$$

donde \mathcal{E} es la suma de las energías que se conoce como Hamiltoniano, la energía cinética $\mathcal{K}(q, \dot{q})$, energía potencial $\mathcal{U}(q)$ y $q = [q_1, q_2, \dots, q_n]^T$ es una función explícita del tiempo y corresponde al vector de coordenadas generalizadas [44], [48], [61], [62].

2.4. Herramientas computacionales

2.4.1. Matlab

MATLAB es el nombre abreviado de “MATrix LABoratory”. Es un programa para realizar cálculos numéricos con vectores y matrices, y por tanto se puede trabajar también con números escalares (tanto reales como complejos), con cadenas de caracteres y con otras estructuras de información más complejas. Matlab es un lenguaje de alto rendimiento para cálculos técnicos, es al mismo tiempo un entorno y un lenguaje de programación. Uno de sus puntos fuertes es que permite construir sus propias herramientas reutilizables. Podemos crear fácilmente nuestras propias funciones y programas especiales (conocidos

como M-archivos) en código Matlab, los podemos agrupar en Toolbox (también llamadas librerías): colección especializada de M-archivos para trabajar en clases particulares de problemas. Matlab, aparte del cálculo matricial y álgebra lineal, también puede manejar polinomios, funciones, ecuaciones diferenciales ordinarias, gráficos. [1] Una de las librerías que se emplea de Matlab es la de Simulink que es una aplicación que permite construir y simular modelos de sistemas físicos y sistemas de control mediante diagramas de bloques. El comportamiento de dichos sistemas se define mediante funciones de transferencia, operaciones matemáticas, elementos de Matlab y señales predefinidas de todo tipo. Simulink dispone de una serie de utilidades que facilitan la visualización, análisis y guardado de los resultados de simulación.



Figura 2.4: Software Matlab.

2.4.2. Visual Studio

Para programar el dispositivo háptico PHANToM Premium 1.0 se utiliza una alternativa de programación como es Visual Studio, en cual es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y hace más sencilla la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML [13].



Figura 2.5: Entorno de programación Visual Studio.

2.5. Conclusiones de capítulo

En este capítulo se hizo una descripción de las herramientas empleadas para desarrollar este tema de investigación. Se han presentado los conceptos de la teoría de los sistemas difusos, los métodos matemáticos para la obtención del modelo matemático de una interfaz háptica de tres grados de libertad. Se detalla la obtención del modelo cinemático de posición, velocidad y aceleración.

Capítulo 3

Resultados principales

En este capítulo se presentan los resultados principales del presente trabajo de investigación, en el cual se aborda el diseño de una ley de control difusa adaptable para una clase de sistemas no lineales.

3.1. Diseño de la ley de control difusa adaptable

En esta sección se considera un sistema no lineal MIMO en la ecuación (3.1), el cual se emplea para desarrollar la metodología descrito en las secciones posteriores.

$$\begin{aligned} y_1^{(r_1)} &= \mathbf{f}_1(x) + \sum_{j=1}^p g_{1j}(x)\mathbf{u}_j, \\ &\vdots \\ y_p^{(r_p)} &= \mathbf{f}_p(x) + \sum_{j=1}^p g_{pj}(x)\mathbf{u}_j \end{aligned} \tag{3.1}$$

donde r_i es el orden del sistema, el vector de salida $y = [y_1, \dots, y_p]^T$, el vector de estado se asume como disponible para su medición $\mathbf{x} = [y_1, \dot{y}_1, \dots, y_1^{(r_p-1)}, \dots, y_p, \dot{y}_p, \dots, y_p^{(r_p-1)}]^T$, el vector de entradas de control $\mathbf{u} = [u_1, \dots, u_p]^T$ y $f_i(x), g_{ij}(x), i, j = 1, \dots, p$ son funciones no lineales, suaves y desconocidas.

El vector de salida deseada se define como $\mathbf{y}_d(t) = [y_{d1}(t), \dots, y_{dp}(t)]^T$, el cual representa la trayectoria deseada del sistema y de la ecuación (3.1) se puede reescribir el sistema con:

$$\begin{aligned} \mathbf{y}^{(r)} &= [y_1^{r_1}, \dots, y_p^{r_p}]^T \\ \mathbf{F}(\mathbf{x}) &= [f_1(x), \dots, f_p(x)]^T \\ G(x) &= \begin{bmatrix} g_{11}(x) & \cdots & g_{1p}(x) \\ \vdots & \ddots & \vdots \\ g_{p1}(x) & \cdots & g_{pp}(x) \end{bmatrix}, \end{aligned}$$

como:

$$\mathbf{y}^{(r)} = \mathbf{F}(\mathbf{x}) + G(x)\mathbf{u}. \quad (3.2)$$

La forma del sistema no lineal de la ecuación (3.1) puede aplicarse ésta metodología, sin embargo existe una restricción en la matriz $G(x)$, la cual debe ser definida positiva, esto limita a los sistemas que puede emplearse esta metodología. Uno de los varios sistemas que cumplen con la restricción son los sistemas robóticos, como se ve en la SECCIÓN PROPIEDADES PHANTOM. Además el vector de salidas $y_{di}(t)$, $i = 1, \dots, p$ son funciones del tiempo, conocidas, acotadas y r_i veces diferenciables cuyas derivadas también son acotadas y conocidas.

3.1.1. Primer esquema de control: Par calculado

Para empezar a diseñar el control difuso adaptable es necesario definir un error de seguimiento (3.3), con el cual se definirá la ecuación (3.4),

$$\begin{aligned} e_1(t) &= y_{d1}(t) - y_1(t), \\ &\vdots \\ e_p(t) &= y_{dp}(t) - y_p(t), \end{aligned} \quad (3.3)$$

y el error de seguimiento filtrado como:

$$\begin{aligned} s_1(t) &= \left(\frac{d}{dt} + \lambda_1\right)^{(r_1-1)} e_1(t) + \alpha_1 \int_0^t e_1(\theta) d\theta + \gamma_1 \int_0^t \int_0^t e_1(\theta) d\theta \\ &\vdots \\ s_p(t) &= \left(\frac{d}{dt} + \lambda_p\right)^{(r_p-1)} e_p(t) + \alpha_p \int_0^t e_p(\theta) d\theta + \gamma_p \int_0^t \int_0^t e_p(\theta) d\theta, \end{aligned} \quad (3.4)$$

donde $\alpha, \gamma, \lambda > 0$ para $i = 1, \dots, p$. Ahora para que la ecuación (3.4) converja a 0 cada uno de los términos del lado derecho de la ecuación deben de ser 0, entonces, la ecuación converge asintóticamente. La derivada temporal del error de seguimiento filtrado se definen como:

$$\begin{aligned} \dot{S}_1 &= v_1 - \mathbf{f}_1(x) - \sum_{j=1}^p g_{1j}(x)\mathbf{u}_j, \\ &\vdots \\ \dot{S}_p &= v_p - \mathbf{f}_p(x) - \sum_{j=1}^p g_{pj}(x)\mathbf{u}_j, \end{aligned} \quad (3.5)$$

donde v_i , $i = 1, \dots, p$, están definidas como:

$$\begin{aligned} v_1 &= y_{d1}^{(r_1)} + \beta_{1,r_1-1} e_p^{(r_1-1)} + \dots + \beta_{1,1} \dot{e}_1 + \alpha_1 e_1 + \gamma_1 \int_0^t e_1(\theta) d\theta \\ &\vdots \\ v_p &= y_{dp}^{(r_p)} + \beta_{1,r_1-1} e_p^{(r_p-1)} + \dots + \beta_{p,1} \dot{e}_p + \alpha_p e_p + \gamma_p \int_0^t e_p(\theta) d\theta, \end{aligned} \quad (3.6)$$

donde

$$\beta_{i,j} = \frac{(r_i - 1)!}{(r_i - j)!(j - 1)!} \lambda_i^{(r_i - j)}, \quad (3.7)$$

con $i = 1, \dots, p$ y $j = 1, \dots, r_i - 1$. Ahora para poder reescribir la ecuación (3.4), necesitamos denotar lo siguiente:

$$\begin{aligned} \mathbf{s}(t) &= [s_1(t) \dots s_p(t)]^T, \\ \mathbf{v}(t) &= [v_1(t) \dots v_p(t)]^T, \end{aligned} \quad (3.8)$$

y la ecuación queda como:

$$\dot{\mathbf{s}} = \mathbf{v} - \mathbf{F}(x) - G(x)\mathbf{u}. \quad (3.9)$$

Las funciones $\mathbf{F}(x)$ y $G(x)$ se asumen como conocidas, por lo tanto podemos establecer la siguiente ley de control

$$u = G^{-1}(x) (-\mathbf{F}(x) + \mathbf{v} + K_0\mathbf{S}), \quad (3.10)$$

donde $K_{0i} = \text{diag}[k_{01}, \dots, k_{0p}]$ con $K_0 > 0$, $i = 1, \dots, p$. Se sustituye la ecuación (3.10) en (3.9) y se obtiene:

$$\dot{s} = K_0 s(t). \quad (3.11)$$

Esto implica que $s_i(t) \rightarrow 0$ como $t \rightarrow \infty$, en otras palabras, $e_i(t) \rightarrow 0$ como $t \rightarrow \infty$. Como ya se había mencionado hasta la ecuación (3.10) puede aplicarse de forma directa a un sistema que cumpla con cierta restricción, pero ahora se aprovecha las virtudes que ofrecen los sistemas difusos para aproximar a cada una de las funciones no lineales $f_i(x)$ y $g_{ij}(x)$.

3.1.2. Segundo esquema de control: Par calculado con identificación Difusa con aproximación de $F(x)$ y $G(x)$

Un sistema difuso Mamdani se caracteriza por poseer un conjunto de reglas del tipo **si-entonces** con la estructura siguiente:

$$R^k : \text{si } x_1 \text{ está en } F_1^k \text{ y } \dots \text{ y } x_n \text{ está en } F_n^k \text{ entonces } y(x) \text{ está en } C^k, k = 1, \dots, r,$$

donde r es el número de reglas difusas, C^k es un conjunto difuso definido en \mathbb{R} , $y \in \mathbb{R}$ es la variable de salida, F_1^k, \dots, F_n^k son conjuntos difusos asociados a la k -ésima regla y el vector de entradas $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$. Los sistemas difusos de tipo Mamdani han demostrado ser herramientas muy útiles para aproximar funciones no lineales. Al utilizar

el fusificador solitón, la operación de producto como motor de inferencia, y el defusificador por centro promedio, la salida final del sistema difuso se obtiene mediante:

$$y(x) = \frac{\sum_{k=1}^r \mu_k(x) d_k}{\sum_{k=1}^r \mu_k(x)}, \quad (3.12)$$

donde

$$\mu_k(x) = \prod_{i=1}^n \mu_{F_i^k}(x_i),$$

observe que $\mu_{F_i^k}(x)$ es el valor de pertenencia de x_i en el conjunto difuso F_i^k y d_k es el punto donde la función de pertenencia C^k alcanza su máximo valor. Observe que la salida $y(x)$, dada por la ecuación (3.12) se puede reescribir de la siguiente manera:

$$w_k(x) = \frac{\mu_k(x)}{\sum_{j=1}^r \mu_j(x)}.$$

Cabe mencionar que el sistema difuso está bien definido, ya que $\sum_{j=1}^r \mu_j(x) \neq 0$ para todo $x \in U$. Un sistema difuso de la forma (3.12) pueden aproximar funciones no lineales continuas sobre un conjunto compacto con cierto grado arbitrario de precisión en función del número de reglas difusas consideradas, pero no significa que mientras más reglas difusas mejore la aproximación.

3.1.3. Tercer esquema de control: Par calculado con identificación Difusa y ley de adaptación

En esta sección se emplea sistemas difusos para aproximar a las funciones $f_i(x)$ y $g_{ij}(x)$ que se asumen como desconocidas, después éstas aproximaciones se utilizan para desarrollar un control adaptable que establezca al sistema cumpliendo con el criterio del método indirecto de Lyapunov. Sean las funciones no lineales $f_i(x)$ y $g_{ji}(x)$ aproximadas por los sistemas difusos en un conjunto cerrado y acotado, es decir, compacto D_x de la forma:

$$\begin{aligned} \hat{f}_i(x, \theta_{f_i}) &= \mathbf{w}_{f_i}^T(x) \theta_{f_i}, & i &= 1, \dots, p, \\ \hat{g}_{ij}(x, \theta_{g_{ij}}) &= \mathbf{w}_{g_{ij}}^T(x) \theta_{g_{ij}}, & i, j &= 1, \dots, p, \end{aligned}$$

donde $\mathbf{w}_{f_i}(x)$ y $\mathbf{w}_{g_{ij}}(x)$ son vectores base de funciones, establecidas por el diseñador, $\theta_{f_i}(x)$ y $\theta_{g_{ij}}(x)$ son los parámetros ajustables correspondientes de cada sistema difuso.

Se define

$$\theta_{f_i}^* = \arg \min_{\theta_{f_i}} \left[\sup_{x \in D_x} \left| f_i(x) - \hat{f}_i(x, \theta_{f_i}) \right| \right], \quad i = 1, \dots, p,$$

$$\theta_{g_{ij}}^* = \arg \min_{\theta_{g_{ij}}} \left[\sup_{x \in D_x} \left| g_{ij}(x) - \hat{g}_{ij}(x, \theta_{g_{ij}}) \right| \right], \quad i, j = 1, \dots, p,$$

con los parámetros óptimos de θ_{f_i} y $\theta_{g_{ij}}$ respectivamente. Cabe mencionar que los parámetros óptimos $\theta_{f_i}^*$ y $\theta_{g_{ij}}^*$ son constantes que se introducen únicamente con propósito de análisis cuyos valores no se necesitan para implementarlos en la plataforma experimental. Sea los errores de estimación de los parámetros:

$$\tilde{\theta}_{f_i} = \theta_{f_i}^* - \theta_{f_i}, \quad i = 1, \dots, p, \quad \tilde{\theta}_{g_{ij}} = \theta_{g_{ij}}^* - \theta_{g_{ij}}, \quad i, j = 1, \dots, p,$$

y el error mínimo de aproximación difusa que se obtiene de la diferencia de los parámetros óptimos y del error de aproximación, esta dada por la siguiente ecuación:

$$\tilde{\theta}_{\mathbf{f}_i} = \theta_{\mathbf{f}_i}^* - \theta_{\mathbf{f}_i},$$

$$\tilde{\theta}_{g_{ij}} = \theta_{g_{ij}}^* - \theta_{g_{ij}} + + + \mathbf{a}.$$

En [10] se demuestra que los sistemas difusos cumplen con el teorema de aproximación universal sobre un conjunto compacto D_x , ya que el conjunto compacto es lo suficientemente grande para mantener dentro la variable de estado x cuando el sistema se encuentre en lazo cerrado, se puede decir que el error de aproximación mínimo está acotado para todo x que está en D_x , en otras palabras,

$$|\varepsilon_{\mathbf{f}_i}(x)| \leq \bar{\varepsilon}_{\mathbf{f}_i}(x), \quad |\varepsilon_{g_{ij}}(x)| \leq \bar{\varepsilon}_{g_{ij}}(x), \quad \forall x \in D_x, \quad (3.13)$$

donde $\bar{\varepsilon}_{g_{ij}}$ y $\bar{\varepsilon}_{\mathbf{f}_i}$ son constantes proporcionadas por el diseñador.

Sea

$$\bar{F}(x, \theta_f = [\bar{f}_1(x, \theta_{\mathbf{f}_1}), \dots, \bar{f}_p(x, \theta_{\mathbf{f}_p})]^T,$$

$$G(x, \theta_g = \begin{bmatrix} g_{11}(x) & \cdots & g_{1p}(x) \\ \vdots & \ddots & \vdots \\ g_{p1}(x) & \cdots & g_{pp}(x) \end{bmatrix},$$

$$\varepsilon_f(x) = [\varepsilon_{\mathbf{f}_1}(x), \dots, \varepsilon_{\mathbf{f}_p}(x)]^T,$$

$$\varepsilon_g(x) = \begin{bmatrix} \varepsilon_{g_{11}}(x) & \cdots & \varepsilon_{g_{1p}}(x) \\ \vdots & \ddots & \vdots \\ \varepsilon_{g_{p1}}(x) & \cdots & \varepsilon_{g_{pp}}(x) \end{bmatrix},$$

$$\bar{\epsilon}_f(x) = [\bar{\epsilon}_{f_1}(x), \dots, \bar{\epsilon}_{f_p}(x)]^T$$

$$\bar{\epsilon}_g(x) = \begin{bmatrix} \bar{\epsilon}_{g_{11}}(x) & \cdots & \bar{\epsilon}_{g_{1p}}(x) \\ \vdots & \ddots & \vdots \\ \bar{\epsilon}_{g_{p1}}(x) & \cdots & \bar{\epsilon}_{g_{pp}}(x) \end{bmatrix},$$

$$\theta_f = [\theta_{f_1}, \dots, \theta_{f_p}]^T,$$

$$\theta_g = [\theta_{g_1}, \dots, \theta_{g_p}]^T,$$

$$\theta_g = [\theta_{g_1}, \dots, \theta_{g_p}]^T,$$

Entonces, se puede establecer lo siguiente:

$$\begin{aligned} F(x) - \bar{F}(x, \theta_f) &= \bar{F}(x, \theta_f^*) - \bar{F}(x, \theta_f) + \bar{\epsilon}_f(x), \\ G(x) - \bar{G}(x, \theta_g) &= \bar{G}(x, \theta_g^*) - \bar{G}(x, \theta_g) + \bar{\epsilon}_g(x). \end{aligned}$$

Se considera el controlador de la ecuación (3.10), pero las funciones no lineales de $F(x)$ y $G(x)$ se sustituyen por las ecuaciones difusas obtenidas $\hat{F}(x, \theta_f)$ y $\hat{G}(x, \theta_g)$, el controlador se puede escribir como:

$$u = \hat{G}(x, \theta_g)^{-1}(-\hat{F}(x, \theta_f) + v + K_0 S). \quad (3.14)$$

La ley de control (3.14) puede ser aplicada directamente, pero ya que los parámetros θ_g son calculados en líneas, no se puede garantizar la existencia de la matriz $\hat{G}(x, \theta_g)^{-1}$, la cual se define como:

$$\hat{G}^T(x, \theta_g)[\epsilon_0 I_p + \hat{G}(x, \theta_g)\hat{G}^T(x, \theta_g)]^{-1}, \quad (3.15)$$

donde ϵ_0 es un valor positivo y muy pequeño, $I_p \in \mathbb{R}^{n \times n}$ es una matriz identidad, la ecuación (3.15), está bien definida, inclusive cuando $\hat{G}(x, \theta_g)$ sea singular.

Considerando la ecuación (3.14) se reemplaza $\hat{G}(x, \theta_g)^{-1}$ por la ecuación (3.15) y se obtiene:

$$u = \hat{G}^T(x, \theta_g)[\epsilon_0 I_p + \hat{G}(x, \theta_g)\hat{G}^T(x, \theta_g)]^{-1}(-\hat{f}(x, \theta_f) + v + K_0 s). \quad (3.16)$$

Dado la ecuación (3.16), note que, siempre está bien definida, pero por sí sola no puede garantizar la estabilidad del sistema en lazo cerrado, debido, principalmente al error de aproximación de las funciones $F(x)$ y $G(x)$ y de la aproximación de $\hat{G}(x, \theta_g)^{-1}$ con la inversa regularizada, con el fin de cancelar dicho error se le propone un término más a la ley de control y finalmente la ley de control se propone como:

$$u = u_c + u_r.$$

donde

$$u_c = \hat{G}^T(x, \theta_g) \left[\varepsilon_o I_p + \hat{G}(x, \theta_g) \hat{G}^T(x, \theta_g) \right]^{-1} (-\hat{\mathbf{f}}(x, \theta_f) + v + K_o s), \quad (3.17)$$

$$u_r = s \left(\frac{s^T(Q_0 + Q_1 + Q_2)s + \bar{\varepsilon}_f^T Q_0^{-1} \bar{\varepsilon}_f + (\bar{\varepsilon}_g u_c)^T Q_1^{-1} (\bar{\varepsilon}_g u_c + u_0^T Q_2^{-1} u_0)}{2\sigma_0 \|s\|^2 + \delta} \right), \quad (3.18)$$

donde

$$u_0 = \varepsilon_0 \left[\varepsilon_o I_p + \hat{G}(x, \theta_g) \hat{G}^T(x, \theta_g) \right]^{-1} (-\hat{\mathbf{f}}(x, \theta_f) + v + K_o s), \quad (3.19)$$

Q_0, Q_1, Q_2 son matrices definidas positivas, cuyo valor queda a la elección del diseñador, σ_0 es una constante positiva, que se asocia con el valor más pequeño de la matriz $G(x)$. Las leyes de adaptación se proponen de la siguiente manera:

$$\dot{\theta}_f = -N_f^{-1} w_f s, \quad (3.20)$$

$$\dot{\theta}_{g_i} = -N_{g_i}^{-1} w_{g_i} s u_i, \quad (3.21)$$

$$(3.22)$$

$$\dot{\delta} = -\eta_0 \frac{\psi}{\sigma_0 \|s\|^2} + \delta,$$

con

$$N_f = \text{diag} \left[\frac{1}{n_{f_1}}, \dots, \frac{1}{n_{f_1}} \right],$$

$$N_{g_i} = \text{diag} \left[\frac{1}{n_{g_{1i}}}, \dots, \frac{1}{n_{g_{1p}}} \right], i = 1, \dots, p,$$

$$w_f = \begin{bmatrix} w_{f_1}^T & 0 & \dots & 0 \\ 0 & w_{f_2}^T & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & w_{f_p}^T \end{bmatrix},$$

$$w_{g_i} = \begin{bmatrix} w_{g_{1i}}^T & 0 & \dots & 0 \\ 0 & \ddots & \vdots & \vdots \\ \vdots & 0 & w_{g_{(p-1)i}}^T & 0 \\ 0 & 0 & \dots & w_{g_{pi}}^T \end{bmatrix},$$

$$\psi = \frac{1}{2} \left[s^T(Q_0 + Q_1 + Q_2)s + \bar{\varepsilon}_f^T Q_0^{-1} \bar{\varepsilon}_f + (\bar{\varepsilon}_g u_c)^T Q_1^{-1} (\bar{\varepsilon}_g u_c + u_0^T Q_2^{-1} u_0) \right],$$

donde n_{f_i} y $n_{g_{ij}}$ son constantes positivas.

3.2. Implementación de resultados vía simulación

En esta sección se presentan los resultados experimentales, que corresponden al control difuso descrito anteriormente. La ley de control obtenida se programa y valida al dispositivo háptico PHANToM Premium 1.0.

Se hace una validación del espacio de trabajo del dispositivo PHANToM Premium 1.0, con la trayectoria deseada para poder conocer los posición del efector final y poder definir las funciones de pertenencia de acuerdo a su posición.

Se presentan los resultados experimentales obtenidos de PHANToM Premium 1.0, al igual que en la simulación primero se presentan los resultados obtenidos cuando la dinámica del sistema se asume como conocida y después los resultados obtenidos cuando la dinámica del sistema se asume como desconocido y con la ley de adaptación agregada.

Se sabe para la trayectoria deseada, el eslabón 1 está entre $[0, 0.22]$ radianes, eslabón 2 está entre $[0, 1.15]$ radianes y eslabón 3 está entre $[-0.02, 0.55]$ radianes.

3.2.1. Identificación del sistema

En esta sección se hace uso de los sistemas difusos en el marco de Mamdani para aproximar a las funciones no lineales, se retoma al sistema no lineal de la forma $\mathbf{y}^{(r)} = \mathbf{F}(x) + G(g)$ y la ecuación (2.8), observe que el sistema se puede reescribir de la forma

$$\ddot{\mathbf{q}} = \underbrace{-H^{-1}(\mathbf{q})(C(q, \dot{q})\dot{\mathbf{q}} + \mathbf{G}(q))}_{F(x)} + \underbrace{H^{-1}(q)}_{G(x)} \tau, \quad (3.23)$$

donde los valores de $C(q, \dot{q})\dot{q}$, $G(x)$ y H^{-1} están descritos en (2.2.1).

De la ecuación (3.23) se sabe que el vector de estados, el cual se asume disponible para su medición es $x = [q_1, \dot{q}_1, q_2, \dot{q}_2, q_3, \dot{q}_3]^T$, el vector de entradas $u = [u_1, u_2, u_3]^T$ y el vector de salidas $y = [q_1, q_2, q_3]^T$ y

$$F(x) = [f_1(x), f_2(x), f_3(x)]^T,$$

$$G(x) = \begin{bmatrix} g_{11}(x) & 0 & 0 \\ 0 & g_{22}(x) & g_{23}(x) \\ 0 & g_{32}(x) & g_{33}(x) \end{bmatrix},$$

cabe recordar que son funciones no lineales y que se asumen como desconocidas.

Como ya se mencionó en el capítulo 2, una de las propiedades del PHANToM Premium 1.0 es que la función $G(x)$ es definida positiva.

3.2.2. Aproximación de $F(\mathbf{x})$

Los sistemas difusos de tipo Mamdani permiten establecer el número de reglas difusas con base en la experiencia de la persona que diseñe el controlador, sin tomar en cuenta alguna metodología se tomaron funciones de pertenencia que con base en la experimentación es la que mejor desempeño tiene, para el caso del PHANToM Premium 1.0 se seleccionaron las funciones de pertenencias triangulares y dado que en la validación del espacio de trabajo se conoce el rango de valores de las articulaciones. A continuación se ilustra como se distribuyen las funciones de pertenencia.

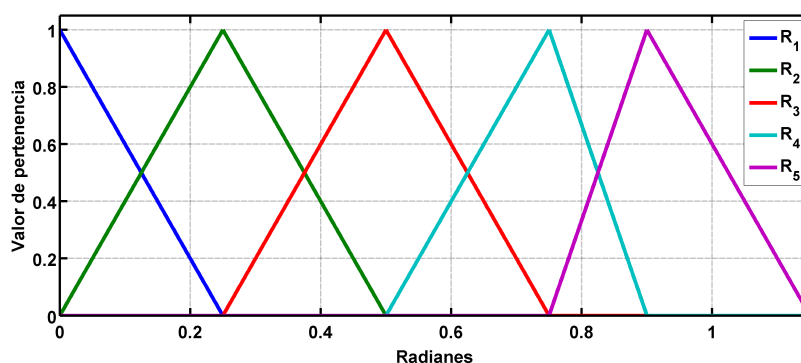


Figura 3.1: Funciones de pertenencia para posición articular q_2 .

Como se ha mencionado anteriormente el rango de operaciones del PHANToM Premium 1.0 está alrededor $q_2 \in [0, 1.12]$ y $q_3 \in [-0.02, 0.55]$ radianes, es por eso que se distribuyen las funciones de pertenencia de acuerdo al espacio de trabajo del PHANToM Premium 1.0. En la Tabla 3.2.2 se muestran los parámetros de las funciones de pertenencia, tomando en cuenta solamente los valores de la articulación 2 (q_2).

Regla	a	b	c
R_1	-0.125	0	0.125
R_2	0	0.125	0.250
R_3	0.125	0.250	0.375
R_4	0.250	0.375	0.5
R_5	0.375	0.5	0.625

Para distribuir las funciones de pertenencia con respecto a la articulación 3 (q_3), se distribuyen de acuerdo a los valores que se presentan en la Figura 3.3 y se ilustran en la Figura 3.2.

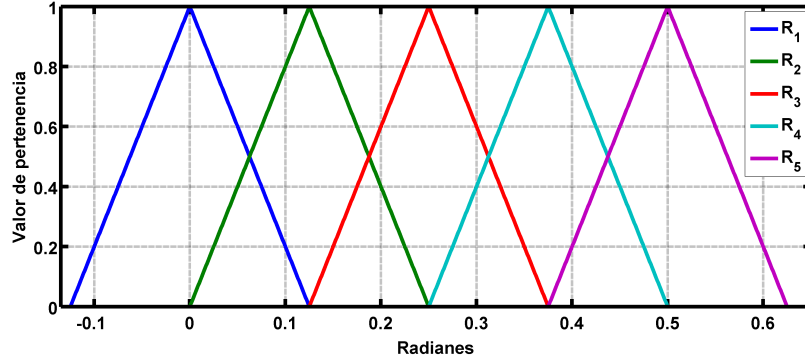


Figura 3.2: Función de pertenencia para posición articular q_3

En la Tabla 3.2.2 se muestran los parámetros que se utilizan para distribuir de acuerdo al movimiento de la articulación (q_3).

Regla	a	b	c
R_1	-0.25	0	0.25
R_2	0	0.25	0.50
R_3	0.25	0.50	0.75
R_4	0.50	0.75	1
R_5	0.75	1	1.15

Ahora se tiene la ecuación (3.12) la cual puede aproximar a $f_1(x)$, $f_2(x)$ y $f_3(x)$ con

$$f_i(x) = \mathbf{w}^T(x)\theta, \text{ con } i = 1, 2, 3,$$

donde $\theta = [d_1, d_2, \dots, d_5]^T$ es el vector de parámetros que se obtienen de la ley de adaptación, únicamente que se tiene que integrar (3.20) y $\mathbf{w}(x) = [w_1(x), w_2(x), \dots, w_5(x)]^T$

$$y(x) = \frac{\mu_k(x)d_k}{\sum_{k=1}^5 \mu_j(x)}, \quad K = 1, \dots, 5m, \quad (3.24)$$

donde

$$\mu_k(x) = \prod_{i=1}^5 \mu_{F_i^k}(x_i),$$

donde $\mu_{F_1^k}(x_1)$ es el valor de pertenencia de q_1 , $\mu_{F_2^k}(x_2)$ es el valor de pertenencia de \dot{q}_1 , $\mu_{F_3^k}(x_3)$ es el valor de pertenencia de q_2 , $\mu_{F_4^k}(x_4)$ es el valor de pertenencia de \dot{q}_2 , $\mu_{F_5^k}(x_5)$ es el valor de pertenencia de q_3 , $\mu_{F_6^k}(x_6)$ es el valor de pertenencia de \dot{q}_3 asociados para la

k -ésima regla. La ley de adaptación de la ecuación (3.20) se propone como:

$$\begin{aligned}\dot{\theta}_{f_1} &= -\eta_{f_1} w_{f_1}(x) s_1, \\ \dot{\theta}_{f_2} &= -\eta_{f_2} w_{f_2}(x) s_2, \\ \dot{\theta}_{f_3} &= -\eta_{f_3} w_{f_3}(x) s_3,\end{aligned}$$

donde $w_{f_i} = w(x) = [w_1, w_2, \dots, w_5]^T$, $i = 1, 2, 3$ y $\eta_{f_1} = 0.25, \eta_{f_2} = 0.20, \eta_{f_3} = 0.30$.

3.3. Aproximación de $G(x)$

Para aproximar a la función $G(x)$ se sabe por el análisis previo que $g_{11}(x)$, depende únicamente de la articulación q_2 . También se sabe por la validación del espacio de trabajo que q_2 está entre los valores $[0, 1.15]$ radianes y con estos hechos se proponen siete funciones de pertenencia triangulares como se muestra en la Figura 3.3,

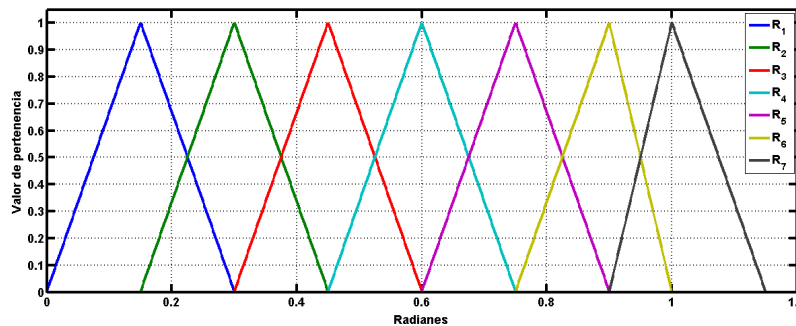


Figura 3.3: Funciones de pertenencia para posición articular q_2 .

donde los parámetros de las siete funciones de pertenencias están dadas por Tabla 3.1.

Tabla 3.1: Parámetros de funciones de pertenencia.

Regla	a	b	c
R_1	0	0.15	0.30
R_2	0.25	0.30	0.45
R_3	0.30	0.45	0.60
R_4	0.45	0.60	0.75
R_5	0.60	0.75	0.90
R_6	0.75	0.90	1
R_7	0.90	1	1.15

Aproximación $g_{11}(x)$

El sistema difuso para aproximar a la función $g_{11}(x)$ se propone con 7 funciones de pertenencia anteriormente mencionadas. Las reglas se mencionan a continuación:

- R_1 : Si $q_2(t)$ está *cerca* de 0.15 **entonces** $g_{11}(x)$ está en d_1 ,
- R_2 : Si $q_2(t)$ está *cerca* de 0.30 **entonces** $g_{11}(x)$ está en d_2 ,
- R_3 : Si $q_2(t)$ está *cerca* de 0.45 **entonces** $g_{11}(x)$ está en d_3 ,
- R_4 : Si $q_2(t)$ está *cerca* de 0.60 **entonces** $g_{11}(x)$ está en d_4 ,
- R_5 : Si $q_2(t)$ está *cerca* de 0.75 **entonces** $g_{11}(x)$ está en d_5 ,
- R_6 : Si $q_2(t)$ está *cerca* de 0.90 **entonces** $g_{11}(x)$ está en d_6 ,
- R_7 : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{11}(x)$ está en d_7 .

De tal forma la ecuación (3.20) puede aproximar a $g_{11}(x)$ con

$$\hat{g}_{11}(x) = \mathbf{w}^T(x)\theta, \quad (3.25)$$

donde el vector de parámetros del consecuente es $\theta = [d_1, d_2, \dots, d_7]^T$ y se calculan mediante la ley de adaptación (3.20) y el conjunto de funciones difusas base $w(x) = [w_1, w_2, \dots, w_7]^T$ que se definen como:

$$g_{11}(x) = \mathbf{w}^T(x)\theta, \quad (3.26)$$

donde $\theta = [d_1, d_2, \dots, d_7]^T$ es el vector de parámetros que se obtienen de la ley de adaptación (3.12) y $w(x) = [w_1(x), w_2(x), \dots, w_7(x)]^T$

$$y(x) = \frac{\mu_k(x)d_k}{\sum_{k=1}^7 \mu_j(x)}, \quad K = 1, \dots, 7, \quad (3.27)$$

donde

$$\mu_k(x) = \prod_{i=1}^7 \mu_{G_i^k}(x_i),$$

donde $\mu_{G_{11}^k}(x_1)$ es el valor de pertenencia de q_2 , es el valor de pertenencia de \dot{q}_2 asociados para la k -ésima regla. La ley de adaptación de la ecuación (3.20) se propone como:

$$\dot{\theta}_{g_{11}} = -\eta_{g_{11}} w_{g_{11}}(x) s_1 u_1,$$

donde $w_{g_{11}} = w(x) = [w_1, w_2, \dots, w_7]^T$ y $\eta_{g_{11}} = 0.20$.

Aproximación $g_{22}(x)$

El sistema difuso para aproximar a la función $g_{22}(x)$ se propone con doce funciones de pertenencia anteriormente mencionadas. Las reglas se mencionan a continuación:

- R_1 : Si $q_1(t)$ está *cerca* de 0.15 **entonces** $g_{22}(x)$ está en d_1 ,
- R_2 : Si $q_1(t)$ está *cerca* de 0.30 **entonces** $g_{22}(x)$ está en d_2 ,
- R_3 : Si $q_1(t)$ está *cerca* de 0.45 **entonces** $g_{22}(x)$ está en d_3 ,
- R_4 : Si $q_1(t)$ está *cerca* de 0.60 **entonces** $g_{22}(x)$ está en d_4 ,
- R_5 : Si $q_1(t)$ está *cerca* de 0.75 **entonces** $g_{22}(x)$ está en d_5 ,
- R_6 : Si $q_1(t)$ está *cerca* de 0.90 **entonces** $g_{22}(x)$ está en d_6 ,
- R_7 : Si $q_1(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_7 ,
- R_8 : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_8 ,
- R_9 : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_9 ,
- R_{10} : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_{10} ,
- R_{11} : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_{11} ,
- R_{12} : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_{12} .

De tal forma la ecuación (3.20) puede aproximar a $g_{22}(x)$ con

$$\hat{g}_{22}(x) = \mathbf{w}^T(x)\theta, \quad (3.28)$$

donde el vector de parámetros del consecuente es $\theta = [d_1, d_2, \dots, d_{12}]^T$ y se calculan mediante la ley de adaptación (3.20) y el conjunto de funciones difusas base $w(x) = [w_1, w_2, \dots, w_{12}]^T$ que se definen como:

$$g_{22}(x) = \mathbf{w}^T(x)\theta,$$

donde $\theta = [d_1, d_2, \dots, d_{12}]^T$ es el vector de parámetros que se obtienen de la ley de adaptación (3.12) y $w(x) = [w_1(x), w_2(x), \dots, w_{12}(x)]^T$

$$y(x) = \frac{\mu_k(x)d_k}{\sum_{k=1}^{11} \mu_j(x)}, K = 1, \dots, 12, \quad (3.29)$$

donde

$$\mu_k(x) = \prod_{i=1}^{12} \mu_{G_i^k}(x_i),$$

donde $\mu_{G_{22}^k}(x_1)$ es el valor de pertenencia de q_1 , es el valor de pertenencia de q_2 asociados para la k -ésima regla. La ley de adaptación de la ecuación (3.20) se propone como:

$$\dot{\theta}_{g_{22}} = -\eta_{g_{22}} w_{g_{22}}(x) s_2 u_2,$$

donde $w_{g_{22}} = w(x) = [w_1, w_2, \dots, w_7]^T$ y $\eta_{g_{22}} = 0.25$.

Aproximación $g_{23}(x)$ y $g_{32}(x)$

El sistema difuso para aproximar a la función $g_{23}(x)$ y $g_{32}(x)$. $G(x)$ al ser una matriz simétrica las funciones g_{23} y g_{32} son la misma. Se propone con doce funciones de pertenencia anteriormente mencionadas. Las reglas se mencionan a continuación:

- R_1 : Si $q_1(t)$ está *cerca* de 0.15 **entonces** $g_{22}(x)$ está en d_1 ,
- R_2 : Si $q_1(t)$ está *cerca* de 0.30 **entonces** $g_{22}(x)$ está en d_2 ,
- R_3 : Si $q_1(t)$ está *cerca* de 0.45 **entonces** $g_{22}(x)$ está en d_3 ,
- R_4 : Si $q_1(t)$ está *cerca* de 0.60 **entonces** $g_{22}(x)$ está en d_4 ,
- R_5 : Si $q_1(t)$ está *cerca* de 0.75 **entonces** $g_{22}(x)$ está en d_5 ,
- R_6 : Si $q_1(t)$ está *cerca* de 0.90 **entonces** $g_{22}(x)$ está en d_6 ,
- R_7 : Si $q_1(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_7 ,
- R_8 : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_8 ,
- R_9 : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_9 ,
- R_{10} : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_{10} ,
- R_{11} : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_{11} ,
- R_{12} : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{22}(x)$ está en d_{12} .

De tal forma la ecuación (3.20) puede aproximar a $g_{23}(x)$ y $g_{32}(x)$ con

$$\hat{g}_{23}(x) = \mathbf{w}^T(x)\theta, \quad (3.30)$$

$$\hat{g}_{32}(x) = \mathbf{w}^T(x)\theta, \quad (3.31)$$

donde el vector de parámetros del consecuente es $\theta = [d_1, d_2, \dots, d_{12}]^T$ y se calculan mediante la ley de adaptación (3.20) y el conjunto de funciones difusas base $w(x) = [w_1, w_2, \dots, w_{12}]^T$ que se definen como:

$$g_{23}(x) = \mathbf{w}^T(x)\theta, \quad (3.32)$$

$$g_{32}(x) = \mathbf{w}^T(x)\theta, \quad (3.33)$$

donde $\theta = [d_1, d_2, \dots, d_{12}]^T$ es el vector de parámetros que se obtienen de la ley de adaptación (3.12) y $w(x) = [w_1(x), w_2(x), \dots, w_{12}(x)]^T$

$$y(x) = \frac{\mu_k(x)d_k}{\sum_{k=1}^{12} \mu_j(x)}, K = 1, \dots, 12, \quad (3.34)$$

donde

$$\mu_k(x) = \prod_{i=1}^{12} \mu_{G_i^k}(x_i),$$

donde $\mu_{G_{22}^k}(x_1)$ es el valor de pertenencia de q_1 , es el valor de pertenencia de q_2 asociados para la k -ésima regla. La ley de adaptación de la ecuación (3.20) se propone como:

$$\begin{aligned} \dot{\theta}_{g_{23}} &= -\eta_{g_{22}} w_{g_{22}}(x) s_2 u_1, \\ \dot{\theta}_{g_{32}} &= -\eta_{g_{22}} w_{g_{22}}(x) s_2 u_1, \end{aligned}$$

donde $w_{g_{23}} = w(x) = [w_1, w_2, \dots, w_{12}]^T$, $w_{g_{32}} = w(x) = [w_1, w_2, \dots, w_{12}]^T$ y $\eta_{g_{23}} = 0.25$, $\eta_{g_{32}} = 0.25$.

Aproximación $g_{33}(x)$

El sistema difuso para aproximar a la función $g_{11}(x)$ se propone con 7 funciones de pertenencia anteriormente mencionadas. Las reglas se mencionan a continuación:

- R_1 : Si $q_2(t)$ está *cerca* de 0.15 **entonces** $g_{11}(x)$ está en d_1 ,
- R_2 : Si $q_2(t)$ está *cerca* de 0.30 **entonces** $g_{11}(x)$ está en d_2 ,
- R_3 : Si $q_2(t)$ está *cerca* de 0.45 **entonces** $g_{11}(x)$ está en d_3 ,
- R_4 : Si $q_2(t)$ está *cerca* de 0.60 **entonces** $g_{11}(x)$ está en d_4 ,
- R_5 : Si $q_2(t)$ está *cerca* de 0.75 **entonces** $g_{11}(x)$ está en d_5 ,
- R_6 : Si $q_2(t)$ está *cerca* de 0.90 **entonces** $g_{11}(x)$ está en d_6 ,
- R_7 : Si $q_2(t)$ está *cerca* de 1 **entonces** $g_{11}(x)$ está en d_7 .

De tal forma la ecuación (3.20) puede aproximar a $g_{33}(x)$ con

$$\hat{g}_{33}(x) = \mathbf{w}^T(x)\theta, \quad (3.35)$$

donde el vector de parámetros del consecuente es $\theta = [d_1, d_2, \dots, d_{12}]^T$ y se calculan mediante la ley de adaptación (3.20) y el conjunto de funciones difusas base $w(x) = [w_1, w_2, \dots, w_{12}]^T$ que se definen como:

$$g_{33}(x) = \mathbf{w}^T(x)\theta,$$

donde $\theta = [d_1, d_2, \dots, d_{12}]^T$ es el vector de parámetros que se obtienen de la ley de adaptación (3.12) y $w(x) = [w_1(x), w_2(x), \dots, w_{12}(x)]^T$

$$y(x) = \frac{\mu_k(x)d_k}{\sum_{k=1}^7 \mu_j(x)}, K = 1, \dots, 12, \quad (3.36)$$

donde

$$\mu_k(x) = \prod_{i=1}^{12} \mu_{G_i^k}(x_i),$$

donde $\mu_{G_{33}^k}(x_1)$ es el valor de pertenencia de q_2 , es el valor de pertenencia de \dot{q}_2 asociados para la k -ésima regla. La ley de adaptación de la ecuación (3.20) se propone como:

$$\dot{\theta}_{g_{33}} = -\eta_{g_{33}} w_{g_{33}}(x) s_3 u_3,$$

donde $w_{g_{33}} = w(x) = [w_1, w_2, \dots, w_{12}]^T$ y $\eta_{g_{33}} = 0.25$.

3.3.1. Ley de adaptación

La ley de control que se aplica al PHANToM Premium 1.0 está definida en las ecuaciones (3.17, 3.18, 3.19) los parámetros que se le asignan están dados a continuación:

Para los calcular a u_r que está dada en la ecuación (3.17) y a u_0 de la ecuación (3.19), se asignan los parámetros $\sigma_0 = 0.001$, $\bar{\epsilon}_f = [1, 1, 1]^T$,

$$K_0 = \begin{bmatrix} 350 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 180 \end{bmatrix},$$

$$\epsilon I_p = 0.001 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Para calcular a u_r de la ecuación (3.18) y u_0 de la ecuación (3.19), se emplean los siguientes parámetros:

$$\bar{\epsilon}_g = \begin{bmatrix} 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 \end{bmatrix},$$

$$Q_0 = \begin{bmatrix} 15 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 15 \end{bmatrix}, Q_1 = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}, Q_2 = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}.$$

3.3.2. Desempeño del sistema en tarea de seguimiento en simulación

Para poder implementar los resultados del capítulo anterior se hace uso del modelo matemático de la plataforma experimental PHANToM Premium 1.0 y la herramienta computacional Matlab 2014a y su librería de programación Simulink.

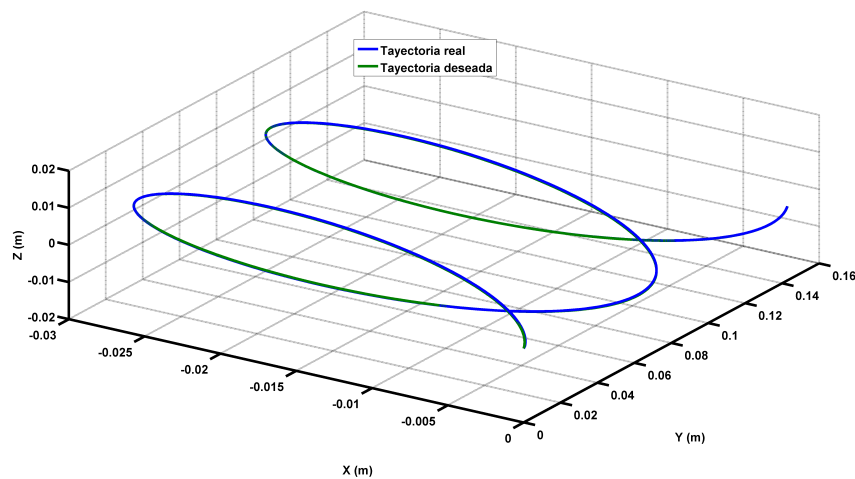


Figura 3.4: Desempeño en espacio de trabajo.

En la Figura 3.5 se ilustra el resultado obtenido de la posición angular de las tres articulaciones, la cual para el ángulo de la primera articulación se representa con q_1 , la señal deseada q_{d1} , segunda articulación q_2 , la señal deseada q_{d2} y para la tercera y última articulación q_3 , la señal deseada q_{d3} .

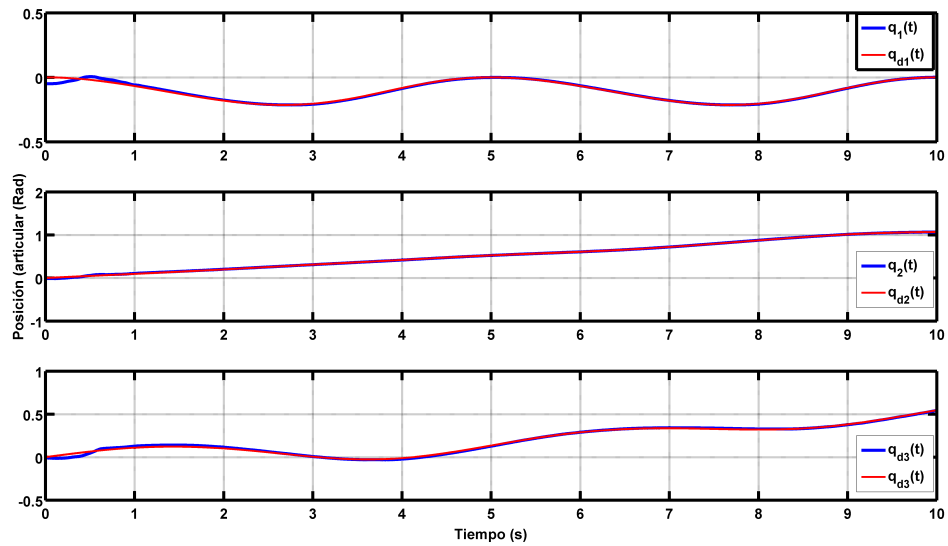


Figura 3.5: Posición angular real y deseada.

La Figura 3.6 se ilustra el error de posición articular de cada uno de los tres eslabones del PHANToM Premium 1.0.

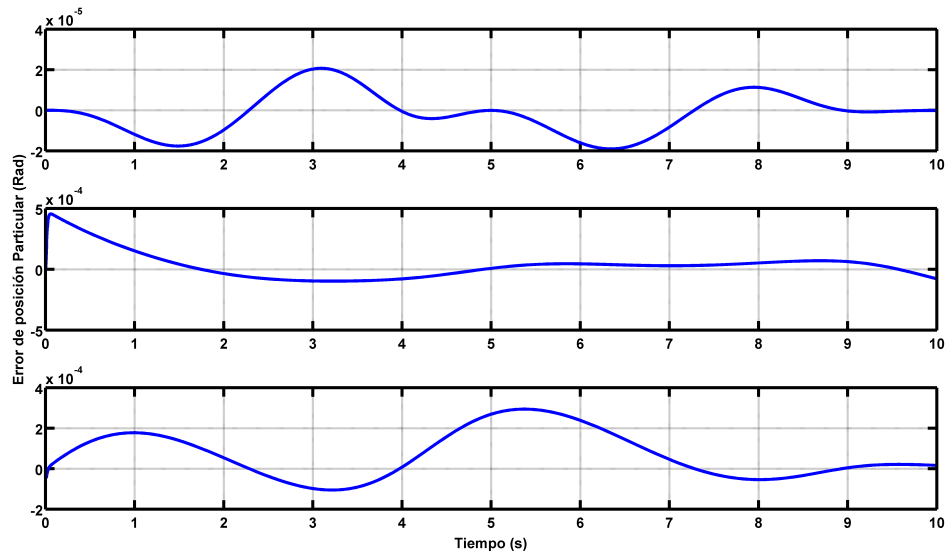


Figura 3.6: Error de posición articular.

3.3.3. Desempeño del sistema en tarea de seguimiento en PHANToM Premium 1.0

Como se puede observar en la Tabla 3.1 los parámetros $[\alpha, \lambda$ y $\gamma]$, para la articulación q_1 los valores son más grandes respecto a las otras articulaciones, esto se debe a que carga con el peso de las demás articulaciones y por eso sus ganancias deben ser más grandes.

En la Figura 3.7 se ilustra el resultado obtenido de la posición angular de las tres articulaciones, la cual para el ángulo de la primer articulación se representa con q_1 , la señal deseada q_{d1} , segunda articulación q_2 , la señal deseada q_{d2} y para la tercera y ultima articulación q_3 , la señal deseada q_{d3} .

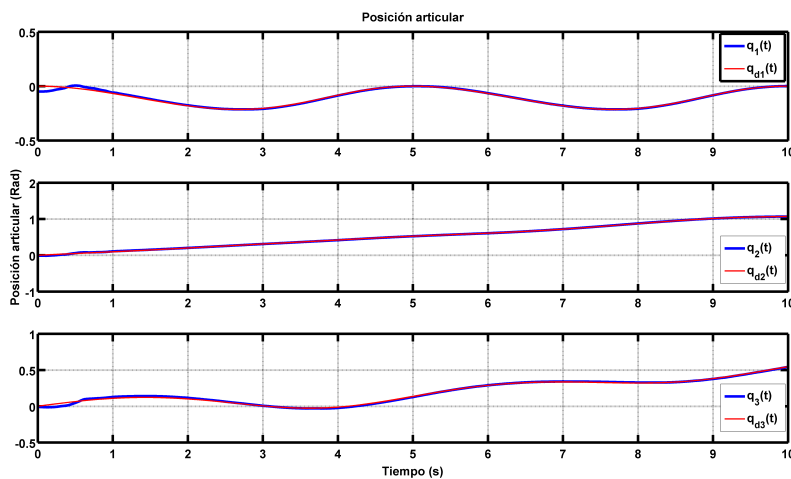


Figura 3.7: Posición angular.

La Figura 3.8 se ilustra el error de posición articular de cada uno de los tres eslabones del PHANToM Premium 1.0.

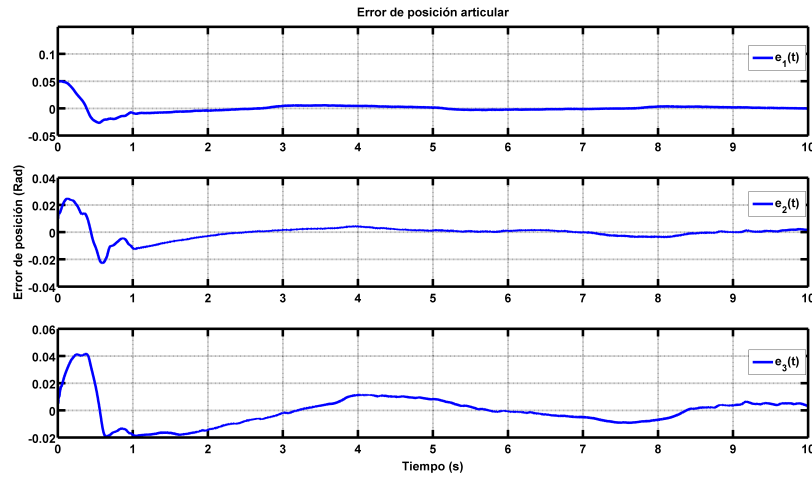


Figura 3.8: Error de posición articular.

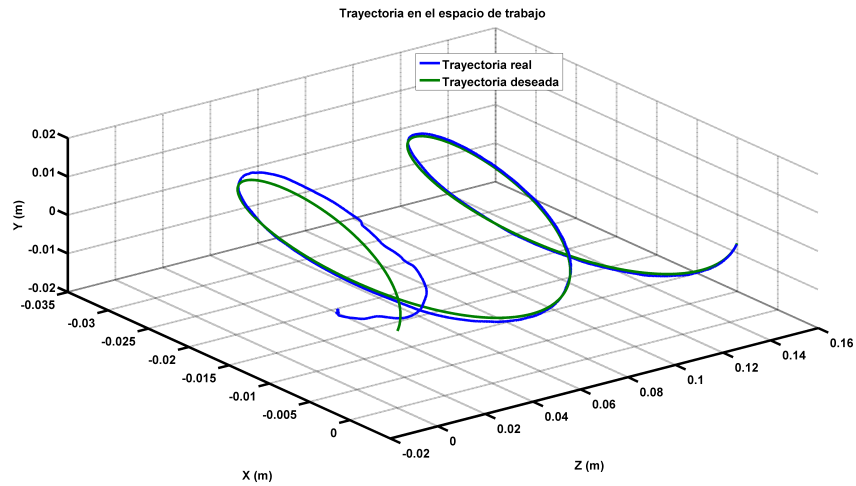


Figura 3.9: Trayectoria en el espacio de trabajo.

En la Figura 3.10 ilustra el resultado obtenido de la ley de control aplicada sobre los actuadores del dispositivo háptico.

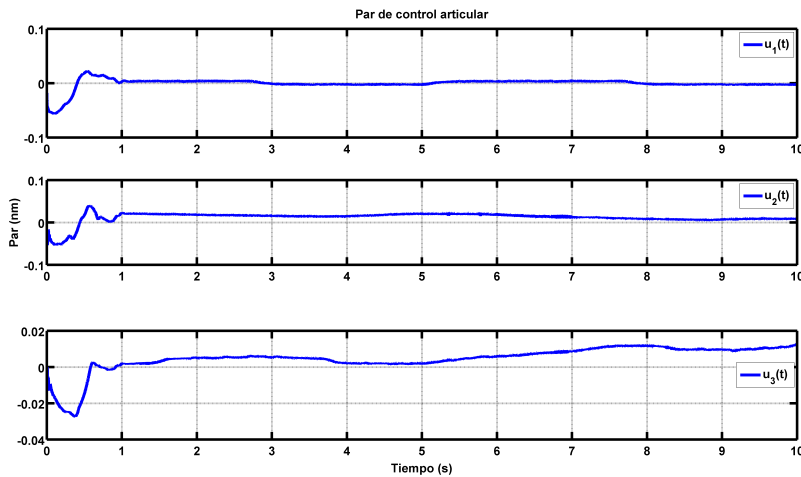


Figura 3.10: Par de control articular.

Los parámetros θ se ilustran en la Figura 3.11, es el resultado de los valores de la función $G(x)$ y sus elementos que se aproximan mediante sistemas difusos.

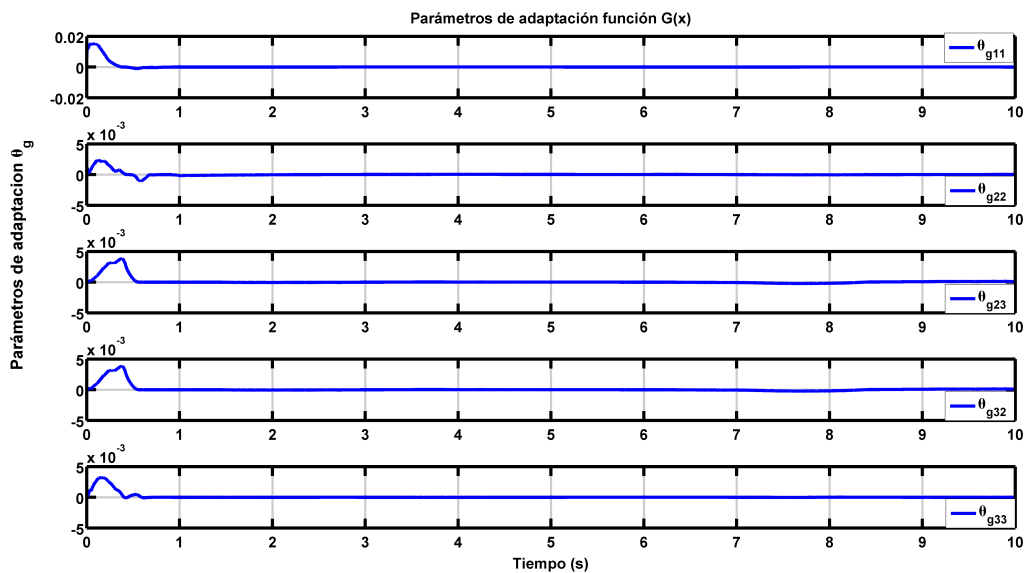


Figura 3.11: Parámetros θ .

Capítulo 4

Conclusiones y trabajos futuros

En este capítulo se presentan las conclusiones obtenidas a partir de obtener la ley de control (3.10) obtenidas en el capítulo (3), su metodología para su implementación, cómo se sintonizaron las ganancias y parámetros requeridos, simulación y experimentación.

4.1. Conclusiones

A continuación se mencionan las conclusiones obtenidas:

- La ley de control obtenida cuando se asume el modelo matemático conocido resulta con un excelente desempeño al momento de realizar el seguimiento de la trayectoria deseada.
- Las ganancias de la ley de control se obtuvieron con base en la experimentación en PHANToM Premium 1.0.
- La metodología que se aplicó para aproximar a las funciones $F(x)$ y $G(x)$ son adecuadas, ya que cumple con el seguimiento de la trayectoria deseada.
- Se verificaron los resultados obtenidos en simulación del PHANToM Premium 1.0.
- Se verificaron los resultados obtenidos en la plataforma experimental PHANToM Premium 1.0.
- La convergencia entre la trayectoria deseada y el PHANToM Premium 1.0, a pesar de que existe perturbaciones causadas por el operador humano cuando está en lazo, dichas perturbaciones fueron realizadas de manera intencional.

- La ley de control aplicada directamente a los codificadores ópticos del PHANToM Premium no muestran mucha saturación, inclusive cuando se hace la trayectoria deseada con un operador en lazo.
- La ley de control adaptable tiene un magnífico desempeño entre diferentes operadores humanos con diferente grado de experiencia con el uso del dispositivo háptico.

La ventaja más importante que ofrece este controlador en el enfoque difuso es el asumir el modelo matemático como desconocido, ya que la mayoría de los controladores que existen necesitan conocer el modelo para poder aplicar la ley de control.

Por otra parte está la desventaja como la matriz de $G(x)$ debe de ser definida positiva lo cual limita al tipo de sistemas donde se pueden aplicar esta metodología.

Cabe destacar que la metodología empleada para sintonizar los parámetros fue mediante experimentación de prueba y error. Los parámetros primeramente se fueron sintonizando en las pruebas de simulación y se hizo un ajuste en las pruebas del PHANToM Premium 1.0, para después sincronizar las ganancias tanto en simulación como en la plataforma experimental.

4.2. Trabajos futuros

Los trabajos futuros que permitirán mejorar el desempeño del PHANToM Premium 1.0 y abarcar otros aspectos se proponen:

- Utilizar una metodología para sintonizar los parámetros.
- Mejorar la aproximación de las funciones $F(x)$ y $G(x)$, mediante elegir adecuadamente el número de reglas difusas.
- Mejorar la aproximación de las funciones $F(x)$ y $G(x)$ utilizando redes neuronales y sistemas difusos.
- Utilizar esta misma metodología para otra plataforma experimental que cumpla con la restricción de la matriz definida positiva de $G(x)$.

Bibliografía

- [1] F. K. Salisbury, B.F. Conti. *Haptic rendering: introductory concepts. IEEE Computer Graphics and Applications*, 24(2):24–32, 2004.
- [2] K. O. Dane Powell. *The task-dependent efficacy of shared-control haptic guidance paradigms. IEEE Computer Graphics and Applications*, 5(pp):208–219, 2012.
- [3] C. Burdea. *Haptic Feedback for Virtual Reality. John Wiley and Sons, Inc.*, 1(1):1–9, 1996.
- [4] Hannaford and S. Venema. *Virtual Environments and Advance Interface Design. Oxford University Press.*, 1(1):1–16, 1995.
- [5] W. Spong and M. Vidyasagar. *Robots Dynamics and Control. John Wiley and Sons.*, 1(1):1–16, 1989.
- [6] V. Parra Vega S. Cervantes and O. A. Dominguez Ramirez. *Dynamic coupling haptic suturing based on orthogonal decomposition. Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1(1):1–19, 2009.
- [7] C. Huegel and M. K. OMalley. *Visual versus haptic progressive guidance for training in a virtual dynamic task. Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1(1):1–19, 2009.
- [8] C. Huegel and M. K. OMalley. *Visual versus haptic progressive guidance for training in a virtual dynamic task. Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1(1):1–19, 2009.
- [9] O. A. Domínguez-Ramírez and V. Parra-Vega. *Active haptic interface with purposes of remote training. Proceedings of the 11th International Conference on Advanced Robotics*, 3(1):1588–1593, 2003.

-
- [10] O. A. Domínguez-Ramírez F. Daz and L. E. Ramos-Velasco. *Sliding computed torque control based on passivity for a haptic device: Phantom premium 1.0*. *World Automation Congress (WAC)*, 1(1):24–28, 2012.
- [11] O. A. Domínguez-Ramírez J. A. Turijan-Rivera, F. J. Ruz Snchez and P. V. V. *Modular platform for haptic guidance in paediatric rehabilitation of upper limb neuromuscular disabilities*. *Converging Clinical and Engineering Research on Neurorehabilitation Biosystems and Biorobotics*, 1(1):925–929, 2013.
- [12] G. Grindlay. *Haptic guidance benefits musical motor learning*. *Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 1(1):1–9, 2008.
- [13] K. Komoriya T. Kotoku and K. Tanie. *A force display system for virtual environments and its evaluation*. *Proceedings of IEEE International Workshop on Robot and Human Communication*, 1(1):246–251, 1994.
- [14] M. M. Boroujeni and A. Meghdari. *Haptic device application in persian calligraphy*. *International Conference on Computer and Automation Engineering*, 1(1):1–20, 2009.
- [15] A. Frizera Neto M. Frisoli A. A. Braidot, C. Cifuentes and A. Santiago. *ZigBee Wearable Sensor Development for Upper Limb Robotics Rehabilitation*. *IEEE Latin America Transactions*, 11(1):408–413, 2013.
- [16] M. Boucherit S. Labiod and G. T.M. *Adaptive fuzzy control of a class of mimo nonlinear systems*. *Fuzzy Sets and Systems*, 151(1):59–77, 2005.
- [17] F. J. Ruiz-Sanchez V. Parra-Vega J. A. Turijan-Rivera, F. A. Machorro-Fernandez and O. A. Dominguez-Ramirez. *Nasa-tlx assessment of modern close loop controllers in haptic guidance for assisted rehabilitation*, *springer-verlag berlin heidelberg*. *Springer-Verlag Berlin Heidelberg: Converging Clinical and Engineering Research on Neurorehabilitation Biosystems and Biorobotics*, 1(1):243–247, 2013.
- [18] M. Keehner D. Feygin and F. Tendick. *Haptic guidance: Experimental evaluation of a haptic training method for a perceptual motor skill*,. *Proceedings of the 10th Symp. On Haptic Interfaces For Virtual Envir. Teleoperator Sysys.*, 1(1):1–16, 2002.
- [19] T. Wang T. Shaocheng and J. Tang. *Fuzzy adaptive output tracking control of nonlinear systems*. *Fuzzy Sets Systems*, 111(1):169–181, 2000.

-
- [20] L. Wang and J. Mendel. *Fuzzy basis functions, universal approximation, and orthogonal least squares learning*. *IEEE Trans. on Neural Networks*, 3(1):807–814, 1992.
- [21] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(1965):338–353, 1965.
- [22] F. Saravia G. Gomez L. Mathé, A. Caverzasi and J. P. Pedroni. *Detection of Human-Robot Collision Using Kinetic*. *IEEE Latin America Transactions*, 11(1):143–148, 2013.
- [23] J. D. M. Santibáñez C. A. G. Gutiérrez, J. R. Reséndiz and G. M. Bobadilla. *A Model and Simulation of a Five-Degree-of-Freedom Robotic Arm for Mechatronic Course*. *IEEE Latin America Transactions*, 12(2):78–86, 2014.
- [24] M. Jamett C. Urrea J. Kern, Member and H. Torres. *Development of a neural controller applied in a 5 DOF robot redundant*. *IEEE Latin America Transactions*, 12(2):98–106, 2014.
- [25] K. Tanaka and M. Sano. *Fuzzy stability criterion of a class of nonlinear systems*. *Inform. sci.*, 71(1):3–26, 1993.
- [26] K. Tanaka H. Wang and M. Griffin. *An analytical framework of fuzzy modeling and control of nonlinear systems: Stability and design issues*. *Seattle: Proc. 1995 American Control Conference*, 1(1):2272–2276, 1995.
- [27] H. Wang and K. Tanaka. *An approach to fuzzy control of nonlinear systems: Stability and design issues*. *IEEE Trans. Fuzzy Syst.*, 4(1):14–23, 1996.
- [28] K. Tanaka H. Wang and M. Griffin. *Parallel distributed compensation of nonlinear systems by Takagi-Sugeno fuzzy model*. *Proc. FUZZ-IEEE/IFES'95*, 1(1):1995, 1995.
- [29] R. Yang and T. Sun. *Neuro-Fuzzy and Soft Computing*. *Prentice Hall*, 1(1):1–20, 1997.
- [30] B. Kosko and J. Bures. *Neural networks and fuzzy systems*. *J. Acoust. Soc. Am.*, 103(1):3131–3131, 1998.
- [31] M. Boucherit S. Labiod and T. Guerra. *Adaptive fuzzy control of a class of mimo nonlinear systems*. *Fuzzy Sets Systems*, 151(1):59–77, 2005.
- [32] C. Bin T. Shaocheng and W. Yongfu. *Fuzzy adaptive output feedback control for mimo nonlinear systems*. *Fuzzy Sets Systems*, 156(1):285–299, 2005.

- [33] T. Shaocheng and H. Li. *Fuzzy adaptive sliding-mode control for mimo nonlinear systems*. *IEEE Trans. Fuzzy Systems*, 11(1):354–360, 2003.
- [34] C. Park and Y. Cho. *T-s model based indirect adaptive fuzzy control using online parameter estimation*. *IEEE Trans. on Systems*, 34(1):2293–2301, 2004.
- [35] C. Chen. *Linear System Theory and Design*. New York: Oxford University Press, 1(3):1–8, 1999.
- [36] L. Wang. *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Englewood Cliffs, NJ.: Prentice Hall,, 1(1):203–221, 1999.
- [37] T. Massie and K. Salisbury. Modeling of the phantom haptic interface. [urlhttp://www.geomagic.com](http://www.geomagic.com), 1999.
- [38] adachi Adachi Y. Development of a haotic interface for virtual reality. *France Congress on Mechatronics*, 1994.
- [39] Brisenó Cerón Abraham. Análisis de movimiento superior para identificar. *5 Congreso Nacional de Mecatrónica y Tecnologías Inteligentes*, 2012.
- [40] Briseño Cerón Abraham. Sistemas robóticos complejos para interacción háptica con alto desempeño antropomórfico. *UPSIN*, 2012.
- [41] Wang L.X. A course in fuzzy systems and control. *Prentice Hall Inc*, 1997.
- [42] Vidyasagar M. Spong Mark W. Robot dynamics and control. *John Wiley and Sons*, 1989.
- [43] Lendek Z., Guerra Thierry Marie, Babuska Robert, and De Schutter Bart. *Stability analysis and nonlinear observer design using Takagi-Sugeno fuzzy models*, volume 262. Springer, 2010.
- [44] Vicente-Parra Domínguez-Ramírez. Diseño e integración de una interfaz háptica activa realista. *CINVESTAV México*, 2004.
- [45] Megumi Nakao Toshihiko Sato Tomohiro Kuroda Yasushi Masuda Osamu Oshiro Yoshihiro Kuroda, Makoto Hirai. Construction of training environment for surgical exclusion with a basic study of multi-finger haptic interaction. *EuroHaptics Conference, and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. *World Haptics*, 2007.

- [46] Nicosia S and Tomei P. A tracking controller for flexible joint robots using only link position feedback. *IEEE Transactions on Automatic Control*, 40(5):885–890, 1995.
- [47] Deitel. How to program in visual studio. *Springer, Edition 7*, 2009.
- [48] Jarillo-Silva Alejandro. Diseño e implementación de técnicas de control no lineal basadas en pasividad aplicadas a guiado háptico. *Universidad Autónoma del Estado de Hidalgo, México*, 2010.
- [49] Domínguez-Ramírez O. A. and V. Parra-Vega. Active haptic exploration of deformable objects. *Chapter 16 in DAAAM International Scientific Book 2006, B. Katalinic (Ed), published by DAAAM international, pp. 177-190*, 2006.
- [50] R. Samperio-Llano Domínguez-Ramírez, V. López-Morales. Resultados preliminares sobre interacción háptica en laberintos virtuales. *Universidad Autónoma del Estado de Hidalgo, México*, 2005.
- [51] Domínguez-Ramírez and Parra-Vega. Texture, roughness and shape haptic perception of deformable virtual objects with constrained lagrangian formulation. *IEEE/RSJ International Conference on Intelligent Robots and Systems, Proceedings of IROS*, 2003.
- [52] Shankar Sastry and Marc Bodson. Adaptive control: Stability, convergence, and robustness. *Prentice-Hall*, 1989-1994.
- [53] S.S. Murray R.M, Sastry. A mathematical introduction to robotic manipulation. *CRO PRESS, Inc Boca Raton. FL.*, 1994.
- [54] Joseph Bak and Donald J. Newman. *Complex Analysis*. Springer, tercera edición, 2010.
- [55] M. Sugeno and G. Kang. *Structure identification of fuzzy model. Fuzzy Sets Systems*, 28(1):329–346, 1986.
- [56] R. Tong. *A control engineering review of fuzzy systems. Automatica*, 13(1):559–569, 1977.
- [57] Adachi Y. Development of a haptic interface for virtual reality. *2nd Japan-France Congress on Mechatronics*, November, 1994.
- [58] Yoshikawa. Foundations of robotics: Analisis and control. *MIT Press*, 1990.

- [59] Shankar P. Bhattacharyya, H. Chapellat, and Lee H. Keel. *Robust Control: The Parametric Approach*. Prentice Hall, 1995.
- [60] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965.
- [61] F.L.Lewis, C. T. Abdallah, and D. M. Dawson. Control of robot manipulators. *Macmillan Publishing Company*, 1993.
- [62] Ordaz-Oliver M. Oscar. Diseño e implementación de estrategias de control no lineal con propósitos de guiado hápticos. *ITP, México*, 2009.

Apéndice A

Código C++ PHANToM Premium 1.0

A.1. Código en C

```
#include <stdio.h>
#include <stdlib.h>
#include "stdafx.h"
#include "wavelet.h"
#include <iostream>
#include <conio.h>
#include <assert.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <HD/hd.h>
#include <HDU/hduError.h>
#include <HDU/hduVector.h>
FILE *Dat;
FILE *Dat2;
FILE *Dat3;

using namespace std;
HDSchedulerHandle hBoyCallBack;

int menu, tiempoE, np=49, fp=7;
```

```

double ngCg11= 0.058, ngCg22= 0.058, ngCg23= 0.058, ngCg33= 0.058,
epsilon = 0.001;

double labda1, labda2, labda3, ro1, ro2, ro3, betha1, betha2, betha3,
alpha1, alpha2, alpha3, gamma1, gamma2, gamma3, KI1, KI2, KI3,
t = 0.0, h, k1, k2, k3, k4, medio=0.00, t0=0.0, Q0, Q1, Q2;

double wCg11[49], wCg22[49], wCg23[49], wCg33[49];
double auxCg11[49], auxCg22[49], auxCg23[49], auxCg33[49];
double pCg11[49], pCg22[49], pCg23[49], pCg33[49];
double muCg11[49], muCg22[49], muCg23[49], muCg33[49];
double auxe [6];
double muTri1Fg0=0, muTri2Fg0=0, muTri3Fg0=0, muTri4Fg0=0, muTri1Fg2
=0, muTri2Fg2=0, muTri3Fg2=0, muTri4Fg2=0, muTri1Fg3=0, muTri2Fg3
=0, muTri3Fg3=0, muTri4Fg3=0, w1Fg0=0, w2Fg0=0, w3Fg0=0, w4Fg0=0,
w1Fg1=0, w2Fg1=0, w3Fg1=0, w4Fg1=0, w1Fg2=0, w2Fg2=0, w3Fg2=0,
w4Fg2=0, w1Fg3=0, w2Fg3=0, w3Fg3=0, w4Fg3=0, sumaFg0=0, sumaFg1=0,
sumaFg2=0, sumaFg3, cg11, cg22, cg23, cg33, sumaCg11, sumaCg22,
sumaCg23, sumaCg33, aux1Fg0, aux2Fg0, aux3Fg0, aux4Fg0, aux1Fg2,
aux2Fg2, aux3Fg2, aux4Fg2, aux1Fg3, aux2Fg3, aux3Fg3, aux4Fg3;

long int contControl=1;

double triangular(double a, double b, double c, double x);
double trapezoidal(double a, double b, double c, double d, double x);
double gaussiana(double a, double b, double x);
double sigmoidal(double a, double b, double x);
double campana(double a, double b, double c, double x);
double Integral (double sv, double sv_1);

hduVector3Dd Tau[100000]; hduVector3Dd tau0[100000];
hduVector3Dd tauC[100000]; hduVector3Dd tauR[100000];
hduVector3Dd Deltaq[100000]; hduVector3Dd Deltaqp[100000];
hduVector3Dd e[100000]; hduVector3Dd ei[100000];
hduVector3Dd eii[100000]; hduVector3Dd ep[100000];
hduVector3Dd q0thetap[100000]; hduVector3Dd q0theta[100000];
hduVector3Dd q1thetap[100000]; hduVector3Dd q2thetap[100000];
hduVector3Dd q3thetap[100000]; hduVector3Dd q1theta[100000];

```

```

hduVector3Dd q2theta[100000]; hduVector3Dd q3theta[100000];
hduVector3Dd thetapCg11[100000]; hduVector3Dd thetaCg11[100000];
hduVector3Dd thetapCg22[100000]; hduVector3Dd thetaCg22[100000];
hduVector3Dd thetapCg23[100000]; hduVector3Dd thetaCg23[100000];
hduVector3Dd thetapCg33[100000]; hduVector3Dd thetaCg33[100000];

HDCallbackCode HDCALLBACK boyCallBack(void *data)
{ HDErrorInfo error;
  hduVector3Dd Fg, theta, position, positiond, Vel, Grav, q, res, qp,
    qpd, Wave_Va;
  hduVector3Dd force, V, s, v;
  hduVector3Dd qd, qppd, Acard, Vcard, Wave_Varef, Varef;
  HHD hHD = hdGetCurrentDevice();
  hdBeginFrame(hHD);
  hdGetDoublev(HD_CURRENT_POSITION, position);
  position=(position/1000);
  hdGetDoublev(HD_CURRENT_VELOCITY, Vel);
  Vel=(Vel/1000);
  memset(force, 0, sizeof(hduVector3Dd));
  t += 0.001;
  t0 += 0.001;
double medio=0, k1=0, k2=0, k3=0, k4=0, K=0, R, r, beta, gamm, alfa,
  aux1, Rd, rd, betad, gammd, alfad, aux1d, l1, l2, l3, l4, l5, g,
  ma, mc, mbe, mdf, pi = 3.1415926535; l1=0.1397; l2=0.1397; l3
  =0.0325; l4=0.0368; l5=0.0527; g=9.81; ma=0.0175; mc=0.0104; mbe
  =0.0214; mdf=0.1106;
double Jpd11, Jpd12, Jpd13, Jpd21, Jpd22, Jpd23, Jpd31, Jpd32, Jpd33, J11,
  J12, J13, J21, J22, J23, J31, J32, J33, DJ, JI11, JI12, JI13, JI21, JI22, JI23,
  JI31, JI32, JI33, JTI11, JTI12, JTI13, JTI21, JTI22, JTI23, JTI31, JTI32,
  JTI33, Jd11, Jd12, Jd13, Jd21, Jd22, Jd23, Jd31, Jd32, Jd33, DJd, JId11,
  JId12, JId13, JId21, JId22, JId23, JId31, JId32, JId33;
double m11, m22, m23, m32, m33, m11I, m22I, m23I, m32I, m33I, deno;

q[0]=atan2(position[0], position[2]+l1);
R=sqrt(pow(position[0],2) + pow((position[2]+l1),2));
r=sqrt(pow(position[0],2)+pow((position[1]-l2),2)+pow((position[2]+l1
),2));
beta=atan2(position[1]-l2,R);

```

```

aux1=(pow(l1 ,2) + pow(r ,2) - pow(l2 ,2) ) / (2*l1*r);
gamm=acos (aux1);
q[1]=gamm+beta;
alfa=acos ((pow(l1 ,2)+pow(l2 ,2)-pow(r ,2))/(2*l1*l2));
q[2]=q[1]+ alfa -pi /2;

J11= l1*cos (q [0]) *cos (q [1])+l2*sin (q [2]) *cos (q [0]) ;
J12=-l1*sin (q [0]) *sin (q [1]) ;
J13= l2*sin (q [0]) *cos (q [2]) ;
J21=0;
J22=l1*cos (q [1]) ;
J23=l2*sin (q [2]) ;
J31=-l1*sin (q [0]) *cos (q [1])-l2*sin (q [0]) *sin (q [2]) ;
J32=-l1*sin (q [1]) *cos (q [0]) ;
J33= l2*cos (q [0]) *cos (q [2]) ;
DJ=J11*J22*J33-J11*J32*J23+J12*J23*J31-J12*J21*J33+J13*J21*J32-J13*
    J22*J31 ;
JI11=(J22*J33-J32*J23)/DJ;
JI12=(-J12*J33+J13*J32)/DJ;
JI13=-(-J12*J23+J13*J22)/DJ;
JI21=-(-J23*J31+J21*J33)/DJ;
JI22=-(-J11*J33+J13*J31)/DJ;
JI23=(-J11*J23+J13*J21)/DJ;
JI31=(J21*J32-J22*J31)/DJ;
JI32=-(-J11*J32-J12*J31)/DJ;
JI33=(J11*J22-J12*J21)/DJ;
JTI11=JI11; JTI21=JI12; JTI31=JI13;
JTI12=JI21; JTI22=JI22; JTI32=JI23;
JTI13=JI31; JTI23=JI32; JTI33=JI33;

qp [0] = JI11*Vel [0]+ JI12*Vel [1]+ JI13*Vel [2];
qp [1] = JI21*Vel [0]+ JI22*Vel [1]+ JI23*Vel [2];
qp [2] = JI31*Vel [0]+ JI32*Vel [1]+ JI33*Vel [2];

double w, ts=5, ro , radio = 0.015;
double h1=-0.015,h2=0;
w=2*pi/ts;
ro=w*t;

```

```

positiond [0] = h1+radio*cos(ro); Vcard[0]=-radio*w*sin(ro); Acard
    [0]=-radio*(w*w)*cos(ro);
positiond [1] = t/70; Vcard [1] = 1/70; Acard [1] = 0;
positiond [2] = h2+radio*sin(ro); Vcard [2]= radio*w*cos(ro); Acard
    [2]=-radio*(w*w)*sin(ro);

qd[0]=atan2(positiond [0] , positiond [2]+l1);
Rd=sqrt(pow(positiond [0] ,2)+pow((positiond [2]+l1) ,2));
rd=sqrt(pow(positiond [0] ,2)+pow((positiond [1]-l2) ,2)+pow((positiond
    [2]+l1) ,2));
betad=atan2(positiond [1]-l2 ,Rd);
aux1d=(pow(l1 ,2)+pow(rd ,2)-pow(l2 ,2))/(2*l1*rd);
gammd=acos(aux1d);
qd[1]=gammd+betad;
alfad=acos((pow(l1 ,2)+pow(l2 ,2)-pow(rd ,2))/(2*l1*l2));
qd[2]=qd[1]+alfad-pi/2;

Jd11=l1*cos(qd [0]) *cos(qd [1]) +(l2*sin(qd [2]) *cos(qd [0]));
Jd12=-l1*sin(qd [0]) *sin(qd [1]);
Jd13=l2*sin(qd [0]) *cos(qd [2]);
Jd21= 0;
Jd22=l1*cos(qd [1]);
Jd23=l2*sin(qd [2]);
Jd31=-l1*sin(qd [0]) *cos(qd [1]) -(l2*sin(qd [0]) *sin(qd [2]));
Jd32=-l1*sin(qd [1]) *cos(qd [0]);
Jd33=l2*cos(qd [0]) *cos(qd [2]);
DJd=Jd11*Jd22*Jd33-Jd11*Jd32*Jd23+Jd12*Jd23*Jd31-Jd12*Jd21*Jd33+Jd13*
    Jd21*Jd32-Jd13*Jd22*Jd31;
JIId11=(Jd22*Jd33-Jd32*Jd23)/DJd;
JIId12=(-Jd12*Jd33+Jd13*Jd32)/DJd;
JIId13=-(-Jd12*Jd23+Jd13*Jd22)/DJd;
JIId21=-(-Jd23*Jd31+Jd21*Jd33)/DJd;
JIId22=-(-Jd11*Jd33+Jd13*Jd31)/DJd;
JIId23=(-Jd11*Jd23+Jd13*Jd21)/DJd;
JIId31=(Jd21*Jd32-Jd22*Jd31)/DJd;
JIId32=-(-Jd11*Jd32-Jd12*Jd31)/DJd;
JIId33=(Jd11*Jd22-Jd12*Jd21)/DJd;
qpd[0]=JIId11*Vcard [0]+JIId12*Vcard [1]+JIId13*Vcard [2];

```

```

qpd[1]=JId21*Vcard[0]+JId22*Vcard[1]+JId23*Vcard[2];
qpd[2]=JId31*Vcard[0]+JId32*Vcard[1]+JId33*Vcard[2];
Deltaq[contControl][0] = q[0] - qd[0];   Deltaqp[contControl][0] = qp
[0] - qpd[0];
Deltaq[contControl][1] = q[1] - qd[1];   Deltaqp[contControl][1] = qp
[1] - qpd[1];
Deltaq[contControl][2] = q[2] - qd[2];   Deltaqp[contControl][2] = qp
[2] - qpd[2];
e[contControl][0] = qd[0] - q[0]; ep[contControl][0]= -qp[0] + qpd[0]
;
e[contControl][1] = qd[1] - q[1]; ep[contControl][1]= -qp[1] + qpd[1]
;
e[contControl][2] = qd[2] - q[2]; ep[contControl][2]= -qp[2] + qpd[2]
;
Jpd11=-11*sin(qd[0])*cos(qd[1])*qpd[0]-12*sin(qd[0])*sin(qd[2])*qpd
[0]-11*sin(qd[1])*cos(qd[0])*qpd[1]+12*cos(qd[0])*cos(qd[2])*qpd
[2];
Jpd12=-11*sin(qd[1])*cos(qd[0])*qpd[0]-11*sin(qd[0])*cos(qd[1])*qpd
[1];
Jpd13=12*cos(qd[0])*cos(qd[2])*qpd[0]-12*sin(qd[0])*sin(qd[2])*qpd
[2];
Jpd21 = 0;
Jpd22 =-11*sin(qd[1])*qpd[1];
Jpd23 =12*cos(qd[2])*qpd[2];
Jpd31 =-11*cos(qd[0])*cos(qd[1])*qpd[0]-12*sin(qd[2])*cos(qd[0])*qpd
[0]+11*sin(qd[0])*sin(qd[1])*qpd[1]-12*sin(qd[0])*cos(qd[2])*qpd
[2];
Jpd32=11*sin(qd[0])*sin(qd[1])*qpd[0]-11*cos(qd[0])*cos(qd[1])*qpd
[1];
Jpd33=-12*sin(qd[0])*cos(qd[2])*qpd[0]-12*sin(qd[2])*cos(qd[0])*qpd
[2];

double m11e,m22e,m23e,m32e,m33e,m11Ie,m22Ie,m23Ie,m32Ie,m33Ie,denoe;
double c11,c12,c13,c21,c22,c23,c31,c32,c33,EC,EP,H,ECd,EPd,Hd,
Htilde;
m11=((0.5*pow(11,2)+0.125*pow(12,2))*ma+(0.125*pow(11,2)+0.5*
pow(13,2))*mc)+0.125*pow(11,2)*(4*ma+mc)*cos(2*q[1])
-0.125*(pow(12,2)*ma+4*pow(13,2)*mc)*cos(2*q[2])+0.125*(12

```

```

        *ma+l3*mc)*cos(q[1])*sin(q[2]); m22=pow(l1,2)*(ma+0.25*mc)
        ; m23=-0.5*l1*(l2*ma+l3*mc)*sin(q[1]-q[2]);
m32=m23; m33=0.25*pow(l2,2)*ma+pow(l3,2)*mc;
deno=m22*m33-m23*m32;
m11I=1/m11;
m22I=m33/deno;
m23I=-m23/deno;
m32I=-m32/deno;
m33I=m22/deno;
        m11e=((0.5*pow(l1,2)+0.125*pow(l2,2))*ma+(0.125*pow(l1,2)
        +0.5*pow(l3,2))*mc)+0.125*pow(l1,2)*(4*ma+mc)*cos(2*Deltaq
        [contControl][1])-0.125*(pow(l2,2)*ma+4*pow(l3,2)*mc)*cos
        (2*Deltaq[contControl][2])+0.125*l1*(l2*ma+l3*mc)*cos(
        Deltaq[contControl][1])*sin(Deltaq[contControl][2]);
m22e=pow(l1,2)*(ma+0.25*mc);
m23e=-0.5*l1*(l2*ma+l3*mc)*sin(Deltaq[contControl][1]-Deltaq[
        contControl][2]);
m32e=m23e;
m33e=0.25*pow(l2,2)*ma+pow(l3,2)*mc;
denoe=m22e*m33e-m23e*m32e;
m11Ie=1/m11e;
m22Ie=m33e/denoe;
m23Ie=-m23e/denoe;
m32Ie=-m32e/denoe;
m33Ie=m22e/denoe;        c11=0.125*(-2*sin(q[1])*(pow(l1,2)*(4*ma+mc)*
        cos(q[1])+2*l1*(l2*ma+l3*mc)*sin(q[2]))*qp[1])+0.25*cos(q[2])*(2*
        l1*(l2*ma+l3*mc)*cos(q[1])+(pow(l2,2)*ma+4*pow(l3,2)*mc)*sin(q[2])
        )*qp[2];
c12=-0.125*(pow(l1,2)*(4*ma+mc)*sin(2*q[1])+4*l1*(l2*ma+l3*mc)*sin(q
        [1])*sin(q[2]))*qp[0];
c13=-0.125*(-4*l1*(l2*ma+l3*mc)*cos(q[1])*cos(q[2])-(pow(l2,2)*ma+4*
        pow(l3,2)*mc)*sin(2*q[2]))*qp[0];
c21=-c12;
c22=0;
c23=0.5*l1*(l2*ma+l3*mc)*cos(q[1]-q[2])*qp[2];
c31=-c13;
c32=0.5*l1*(l2*ma+l3*mc)*cos(q[1]-q[2])*qp[1];
c33=0;

```

```

Grav[0]=0;
Grav[1]=(g*(11*(ma+0.5*mc)+15*mbe)*cos(q[1]))*-0.173;
Grav[2]=(g*(0.5*12*ma+13*mc-14*mdf)*sin(q[2]))*1.663;
EC=(0.5*m11*(pow(qp[0],2)))+(0.5*m22*(pow(qp[1],2)))+(0.5*m33*(pow(qp[2],2)))+(m32*(pow(qp[1],2))*(pow(qp[2],2)));
EP=0.0426*-0.173*sin(q[1])+0.0246*1.663*cos(q[2])-0.0246*1.663;
H=EC+EP;
ECd=(0.5*m11*(pow(qpd[0],2)))+(0.5*m22*(pow(qpd[1],2)))+(0.5*m33*(pow(qpd[2],2)))+(m32*(pow(qpd[1],2))*(pow(qpd[2],2)));
EPd=0.0426*-0.173*sin(qd[1])+0.0246*1.663*cos(qd[2])-0.0246*1.663;
Hd=ECd+EPd;
Htilde=H-Hd;
if(menu==2&& t<tiempoE)
{
qppd[0]=JId11*(Acard[0]-(Jpd11*qpd[0])+(Jpd12*qpd[1])+(Jpd13*qpd[2]))+JId12*(Acard[1]-(Jpd21*qpd[0])+(Jpd22*qpd[1])+(Jpd23*qpd[2]))+JId13*(Acard[2]-(Jpd31*qpd[0])+(Jpd32*qpd[1])+(Jpd33*qpd[2]));
qppd[1]=JId21*(Acard[0]-(Jpd11*qpd[0])+(Jpd12*qpd[1])+(Jpd13*qpd[2]))+JId22*(Acard[1]-(Jpd21*qpd[0])+(Jpd22*qpd[1])+(Jpd23*qpd[2]))+JId23*(Acard[2]-(Jpd31*qpd[0])+(Jpd32*qpd[1])+(Jpd33*qpd[2]));
qppd[2]=JId31*(Acard[0]-(Jpd11*qpd[0])+(Jpd12*qpd[1])+(Jpd13*qpd[2]))+JId32*(Acard[1]-(Jpd21*qpd[0])+(Jpd22*qpd[1])+(Jpd23*qpd[2]))+JId33*(Acard[2]-(Jpd31*qpd[0])+(Jpd32*qpd[1])+(Jpd33*qpd[2]));
e[0][0]=0; ei[0][0]=0; eii[0][0]=0;
e[0][1]=0; ei[0][1]=0; eii[0][1]=0;
e[0][2]=0; ei[0][2]=0; eii[0][2]=0;
for(int l=0; l<3; l++)
{
ei[contControl][l]=Integral((e[contControl][l]),e[contControl-1][l]);
ei[contControl][l]=auxe[l]+ei[contControl][l];
auxe[l]=ei[contControl][l];
}

for(int i=0; i<3; i++)
{

```



```

eii [contControl][i] = Integral((ei [contControl][i]), ei [contControl
-1][i]);
eii [contControl][i] = auxe[i+3] + eii [contControl][i];
auxe[i+3] =eii [contControl][i];
}

alpha1 = 70;    betha1 = 345; gamma1 = 80; labda1 = 110; KI1 = 350;
alpha2 = 63;    betha2 = 30;    gamma2 = 60; labda2 = 70; KI2 = 100;
alpha3 = 50; betha3 = 25;    gamma3 = 50; labda3 = 65; KI3 = 180;

s[0]=ep [contControl][0]+(labda1*ep [contControl][0] )+alpha1*ei [
contControl][0]+(gamma1*eii [contControl][0]);
s[1]=ep [contControl][1]+(labda2*ep [contControl][1] )+alpha2*ei [
contControl][1]+(gamma2*eii [contControl][1]);
s[2]=ep [contControl][2]+(labda3*ep [contControl][2] )+alpha3*ei [
contControl][2]+(gamma3*eii [contControl][2]);
v[0]=qppd[0]+(labda1*ep [contControl][0])+(alpha1*ep [contControl][0])+(
gamma1*ei [contControl][0]);
v[1]=qppd[1]+(labda2*ep [contControl][1])+(alpha2*ep [contControl][1])+(
gamma2*ei [contControl][1]);
v[2]=qppd[2]+(labda3*ep [contControl][2])+(alpha3*ep [contControl][2])+(
gamma3*ei [contControl][2]);

muTri1Fg0 = triangular(-0.8, 0, 0.5, q[0]);
muTri2Fg0 = triangular( 0, 0.18, 0.3, q[0]);
muTri3Fg0 = triangular(-0.18, 0.3, 0.4, q[0]);
muTri4Fg0 = trapezoidal( 0, (pi/6), (pi/2), (pi/2), q[2]);
sumaFg0 = muTri1Fg0 + muTri2Fg0 + muTri3Fg0 + muTri4Fg0;
w1Fg0 = ( muTri1Fg0/sumaFg0 );
w2Fg0 = ( muTri2Fg0/sumaFg0 );
w3Fg0 = ( muTri3Fg0/sumaFg0 );
w4Fg0 = ( muTri4Fg0/sumaFg0 );

q0thetap [contControl][0] = (-0.5*s[0]*w1Fg3 );
q0thetap [contControl][1] = (-0.5*s[0]*w2Fg3 );
q0thetap [contControl][2] = (-0.5*s[0]*w3Fg3 );
q0thetap [contControl][3] = (-0.5*s[0]*w4Fg3 );

```

```

q0theta [contControl][0] = Integral( q0thetap [contControl][0] ,
    q0thetap [contControl -1][0] );
q0theta [contControl][0] = aux1Fg0 + q0thetap [contControl][0];
aux1Fg0 = q0thetap [contControl][0];
q0theta [contControl][1] = Integral( q0thetap [contControl][1] ,
    q0thetap [contControl -1][1] );
q0theta [contControl][1] = aux2Fg0 + q0thetap [contControl][1];
aux2Fg0 = q0thetap [contControl][1];
q0theta [contControl][2] = Integral( q0thetap [contControl][2] ,
    q0thetap [contControl -1][2] );
q0theta [contControl][2] = aux3Fg0 + q0thetap [contControl][2];
aux3Fg0 = q0thetap [contControl][2];
q0theta [contControl][3] = Integral( q0thetap [contControl][3] ,
    q0thetap [contControl -1][3] );
q0theta [contControl][3] = aux4Fg0 + q0thetap [contControl][3];
aux4Fg0 = q0thetap [contControl][3];
Fg[0] = ( (w1Fg0*q0theta [contControl][0])+(w2Fg0*q0theta [contControl
    ][1])+(w3Fg0*q0theta [contControl][2])+(w4Fg0*q0theta [contControl
    ][3]) );
muTri1Fg3 = triangular(-0.8, 0, 0.5, q[2]);
muTri2Fg3 = triangular( 0, 0.18, 0.3, q[2]);
muTri3Fg3 = triangular(-0.18, 0.3, 0.4, q[2]);
muTri4Fg3 = trapezoidal( 0, (pi/6), (pi/2), (pi/2), q[2]);
sumaFg3 = muTri1Fg3 + muTri2Fg3 + muTri3Fg3 + muTri4Fg3;
w1Fg3 = ( muTri1Fg3/sumaFg3 );
w2Fg3 = ( muTri2Fg3/sumaFg3 );
w3Fg3 = ( muTri3Fg3/sumaFg3 );
w4Fg3 = ( muTri4Fg3/sumaFg3 );

q3thetap [contControl][0] = (-0.5*s[2]*w1Fg3 );
q3thetap [contControl][1] = (-0.5*s[2]*w2Fg3 );
q3thetap [contControl][2] = (-0.5*s[2]*w3Fg3 );
q3thetap [contControl][3] = (-0.5*s[2]*w4Fg3 );

q3theta [contControl][0] = Integral( q3thetap [contControl][0] ,
    q3thetap [contControl -1][0] );
q3theta [contControl][0] = aux1Fg3 + q3thetap [contControl][0];
aux1Fg3 = q3thetap [contControl][0];

```

```

q3theta[contControl][1] = Integral( q3thetap[contControl][1],
    q3thetap[contControl-1][1] );
q3theta[contControl][1] = aux2Fg3 + q3thetap[contControl][1];
aux2Fg3 = q3thetap[contControl][1];
q3theta[contControl][2] = Integral( q3thetap[contControl][2],
    q3thetap[contControl-1][2] );
q3theta[contControl][2] = aux3Fg3 + q3thetap[contControl][2];
aux3Fg3 = q3thetap[contControl][2];
q3theta[contControl][3] = Integral( q3thetap[contControl][3],
    q3thetap[contControl-1][3] );
q3theta[contControl][3] = aux4Fg3 + q3thetap[contControl][3];
aux4Fg3 = q3thetap[contControl][3];

Fg[2] = ( (w1Fg3*q3theta[contControl][0])+(w2Fg3*q3theta[contControl
][1])+(w3Fg3*q3theta[contControl [2])+(w4Fg3*q3theta[contControl
][3]) );

muTri1Fg2 = trapezoidal(-pi, 0, 0.35, 0.4, q[1]);
muTri2Fg2 = triangular( 0.3, 0.4, 0.6, q[1]);
muTri3Fg2 = triangular( 0.4, 0.6, 0.8, q[1]);
muTri4Fg2 = trapezoidal( 0.6, 1, (pi/2), pi, q[1]);
sumaFg2 = muTri1Fg2 + muTri2Fg2 + muTri3Fg2 + muTri4Fg2;
w1Fg2 = ( muTri1Fg2/sumaFg2 );
w2Fg2 = ( muTri2Fg2/sumaFg2 );
w3Fg2 = ( muTri3Fg2/sumaFg2 );
w4Fg2 = ( muTri4Fg2/sumaFg2 );

q1thetap[contControl][0] = (-0.5*s[1]*w1Fg2 );
q1thetap[contControl][1] = (-0.5*s[1]*w2Fg2 );
q1thetap[contControl][2] = (-0.5*s[1]*w3Fg2 );
q1thetap[contControl][3] = (-0.5*s[1]*w4Fg2 );

q1theta[contControl][0] = Integral( q1thetap[contControl][0],
    q1thetap[contControl-1][0] );
q1theta[contControl][0] = aux1Fg2 + q1thetap[contControl][0];
aux1Fg2 = q1thetap[contControl][0];

```

```

q1theta[contControl][1] = Integral( q1thetap[contControl][1],
    q1thetap[contControl-1][1] );
q1theta[contControl][1] = aux2Fg2 + q1thetap[contControl][1];
aux2Fg2 = q1thetap[contControl][1];

q1theta[contControl][2] = Integral( q1thetap[contControl][2],
    q1thetap[contControl-1][2] );
q1theta[contControl][2] = aux3Fg2 + q1thetap[contControl][2];
aux3Fg2 = q1thetap[contControl][2];

q1theta[contControl][3] = Integral( q1thetap[contControl][3],
    q1thetap[contControl-1][3] );
q1theta[contControl][3] = aux4Fg2 + q1thetap[contControl][3];
aux4Fg2 = q1thetap[contControl][3];

Fg[1] = ((w1Fg2*q3theta[contControl][0])+(w2Fg2*q3theta[contControl][1])+(w3Fg2*q3theta[contControl][2])+(w4Fg2*q3theta[contControl][3]));

muCg11[0] = triangular(-0.75, -0.5, -0.25, q[1]);
muCg11[1] = triangular(-0.5, -0.25, 0, q[1]);
muCg11[2] = triangular(-0.25, 0, 0.25, q[1]);
muCg11[3] = triangular(0, 0.25, 0.5, q[1]);
muCg11[4] = triangular(0.25, 0.5, 0.75, q[1]);
muCg11[5] = triangular(0.5, 0.75, 1, q[1]);
muCg11[6] = triangular(0.75, 1, 1.25, q[1]);
muCg11[7] = triangular(-0.02, -0.015, -0.01, q[2]);
muCg11[8] = triangular(-0.015, -0.01, -0.005, q[2]);
muCg11[9] = triangular(-0.01, -0.005, 0, q[2]);
muCg11[10] = triangular(-0.005, 0, 0.005, q[2]);
muCg11[11] = triangular(0, 0.005, 0.01, q[2]);
muCg11[12] = triangular(0.005, 0.01, 0.015, q[2]);
muCg11[13] = triangular(0.01, 0.015, 0.02, q[2]);

    int cont11 =0;
    for(int i = 0; i < np; i++)
    {

```

```

        if (cont11 == fp)
        { cont11 =0; }
        for (int j = 0; j < fp; j++)
        {pCg11[i] = muCg11[cont11]*muCg11[cont11+fp];}
        cont11++;
        sumaCg11 = sumaCg11+pCg11[i];
    }

    for(int i = 0; i < np; i++)
    { wCg11[i] = pCg11[i]/sumaCg11; }

    Tau[0][0]=1;
    Tau[1][0]=1;
    Tau[2][0]=1;

    for (int i =0; i < np; i++)
    {thetapCg11[contControl][i] = wCg11[i]*(ngCg11)*s[0]*Tau[
        contControl][0];}

    for(int j=0; j < np; j++)
    {
        thetaCg11[contControl][j] = Integral( thetapCg11[
            contControl][j], thetapCg11[contControl-1][j] );
        thetaCg11[contControl][j] = auxCg11[j] + thetaCg11[
            contControl][j];
        auxCg11[j] = thetaCg11[contControl][j];
    }

    for (int k = 0; k < np; k++)
    { cg11 = cg11 + wCg11[k]*thetaCg11[contControl][k];}

    for (int k =0; k < np; k++)
    {
        thetaCg22[0][k]=0;
        thetapCg22[0][k]=0;
    }

muCg22[0] = triangular(-0.75, -0.5, -0.25, q[1]);

```

```

muCg22[1] = triangular(-0.5, -0.25, 0, q[1]);
muCg22[2] = triangular(-0.25, 0, 0.25, q[1]);
muCg22[3] = triangular(0, 0.25, 0.5, q[1]);
muCg22[4] = triangular(0.25, 0.5, 0.75, q[1]);
muCg22[5] = triangular(0.5, 0.75, 1, q[1]);
muCg22[6] = triangular(0.75, 1, 1.25, q[1]);
muCg22[7] = triangular(-0.02, -0.015, -0.01, q[2]);
muCg22[8] = triangular(-0.015, -0.01, -0.005, q[2]);
muCg22[9] = triangular(-0.01, -0.005, 0, q[2]);
muCg22[10] = triangular(-0.005, 0, 0.005, q[2]);
muCg22[11] = triangular(0, 0.005, 0.01, q[2]);
muCg22[12] = triangular(0.005, 0.01, 0.015, q[2]);
muCg22[13] = triangular(0.01, 0.015, 0.02, q[2]);

int cont22 =0;
for(int i = 0; i < np; i++)
{ if (cont22 == fp)
    { cont22 =0; }
    for (int j = 0; j < fp; j++)
    { pCg22[i] = muCg22[cont22]*muCg22[cont22+fp];}

    cont22++;
    sumaCg22 = sumaCg22+pCg22[i];
}

for(int i = 0; i < np; i++)
{ wCg22[i] = pCg22[i]/sumaCg22;}

Tau[0][1]=1;
Tau[1][1]=1;
Tau[2][1]=1;

for (int i =0; i < np; i++)
{ thetapCg22[contControl][i] = wCg22[i]*(ngCg22)*s[1]*Tau[
    contControl][1];}

for(int j=0; j < np; j++)
{

```

```

        thetaCg22[contControl][j] = Integral( thetapCg22[contControl
            ][j], thetapCg22[contControl-1][j] );
        thetaCg22[contControl][j] = auxCg22[j]; + thetaCg22[
            contControl][j];
        auxCg22[j] = thetaCg22[contControl][j];
    }

    for (int k = 0; k < np; k++)
    { cg22 = cg22 + wCg22[k]*thetaCg22[contControl][k]; }

    for (int k =0; k < np; k++)
    {
        thetaCg23[0][k]=0;
        thetapCg23[0][k]=0;
    }

    muCg23[0] = triangular(-0.75, -0.5, -0.25, q[1]);
    muCg23[1] = triangular(-0.5, -0.25, 0, q[1]);
    muCg23[2] = triangular(-0.25, 0, 0.25, q[1]);
    muCg23[3] = triangular(0, 0.25, 0.5, q[1]);
    muCg23[4] = triangular(0.25, 0.5, 0.75, q[1]);
    muCg23[5] = triangular(0.5, 0.75, 1, q[1]);
    muCg23[6] = triangular(0.75, 1, 1.25, q[1]);
    muCg23[7] = triangular(-0.02, -0.015, -0.01, q[2]);
    muCg23[8] = triangular(-0.015, -0.01, -0.005, q[2]);
    muCg23[9] = triangular(-0.01, -0.005, 0, q[2]);
    muCg23[10] = triangular(-0.005, 0, 0.005, q[2]);
    muCg23[11] = triangular(0, 0.005, 0.01, q[2]);
    muCg23[12] = triangular(0.005, 0.01, 0.015, q[2]);
    muCg23[13] = triangular(0.01, 0.015, 0.02, q[2]);

    int cont23 =0;
    for(int i = 0; i < np; i++)
    {
        if (cont23 == fp)
        { cont23 =0; }
        for (int j = 0; j < fp; j++)
        { pCg23[i] = muCg23[cont23]*muCg23[cont23+fp]; }
    }

```

```

        cont23++;
        sumaCg23 = sumaCg23+pCg23[i];
    }

    for(int i = 0; i < np; i++)
    { wCg23[i] = pCg23[i]/sumaCg23;}

    Tau[0][2]=1;
    Tau[1][2]=1;
    Tau[2][2]=1;

    for (int i =0; i < np; i++)
    { thetapCg23[contControl][i] = wCg23[i]*(ngCg23)*s[1]*Tau[
        contControl][2]; }

    for(int j=0; j < np; j++)
    {
        thetaCg23[contControl][j] = Integral( thetapCg23[contControl
            ][j],thetapCg23[contControl-1][j] );
        thetaCg23[contControl][j] = auxCg23[j] + thetaCg23[
            contControl][j];
        auxCg23[j] = thetaCg23[contControl][j];
    }

    double aauuxx23=0.0;
    for (int k = 0; k < np; k++)
    {
        cg23 = aauuxx23 + wCg23[k]*thetaCg23[contControl][k];
        aauuxx23=cg23;
    }

    for (int k =0; k < np; k++)
    {
        thetaCg33[0][k]=0;
        thetapCg33[0][k]=0;
    }

    muCg33[0] = triangular(-0.75, -0.5, -0.25, q[1]);

```



```

muCg33[1] = triangular(-0.5, -0.25, 0, q[1]);
muCg33[2] = triangular(-0.25, 0, 0.25, q[1]);
muCg33[3] = triangular(0, 0.25, 0.5, q[1]);
muCg33[4] = triangular(0.25, 0.5, 0.75, q[1]);
muCg33[5] = triangular(0.5, 0.75, 1, q[1]);
muCg33[6] = triangular(0.75, 1, 1.25, q[1]);
muCg33[7] = triangular(-0.02, -0.015, -0.01, q[2]);
muCg33[8] = triangular(-0.015, -0.01, -0.005, q[2]);
muCg33[9] = triangular(-0.01, -0.005, 0, q[2]);
muCg33[10] = triangular(-0.005, 0, 0.005, q[2]);
muCg33[11] = triangular(0, 0.005, 0.01, q[2]);
muCg33[12] = triangular(0.005, 0.01, 0.015, q[2]);
muCg33[13] = triangular(0.01, 0.015, 0.02, q[2]);

int cont33 =0;
for(int i = 0; i < np; i++)
{
    if (cont33 == fp)
    { cont33 =0; }
    for (int j = 0; j < fp; j++)
    { pCg33[i] = muCg33[cont33]*muCg33[cont33+fp]; }
    cont33++;
    sumaCg33 = sumaCg33+pCg33[i];
}

for(int i = 0; i < np; i++)
{ wCg33[i] = pCg33[i]/sumaCg33;}

Tau[0][2]=1;
Tau[1][2]=1;
Tau[2][2]=1;

for (int i =0; i < np; i++)
{ thetapCg33[contControl][i] = wCg33[i]*(ngCg33)*s[2]*Tau[
    contControl][2]; }

for(int j=0; j < np; j++)
{

```

```

        thetaCg33[contControl][j] = Integral( thetapCg33[contControl
            ][j], thetapCg33[contControl-1][j] );
        thetaCg33[contControl][j] = auxCg33[j]; + thetaCg33[
            contControl][j];
        auxCg33[j] = thetaCg33[contControl][j];
    }
    double aauuxx=0.0;
    for (int k = 0; k < np; k++)
    {
        cg33 = aauuxx + wCg33[k]*thetaCg33[contControl][k];
        aauuxx= cg33;
    }
    double Gx11, Gx12, Gx13, Gx21, Gx22, Gx23, Gx31, Gx32, Gx33, invGx11
        , invGx12, invGx13, invGx21, invGx22, invGx23, invGx31, invGx32,
        invGx33, detGx, adjGx; Gx12 = Gx13 = Gx21 = Gx31 = invGx31 =
        invGx21 = invGx12 = invGx13 =0;
    Gx11 = epsilon * pow(cg11,2);
    Gx22 = epsilon * (pow(cg22,2)+pow(cg23,2));
    Gx23 = (cg22*cg23)+(cg23*cg33);
    Gx32 = Gx23;
    Gx33 = (pow(cg23,2)+pow(cg33,2));
    detGx = ( Gx11*Gx22*Gx33 ) - ( Gx11*Gx23*Gx32 );
    invGx11 = (( Gx22*Gx33 )-( Gx23*Gx32 ))/detGx;
    invGx12 = 0;
    invGx13 = 0;
    invGx21 = 0;
    invGx22 = ( Gx11*Gx33 )/detGx;
    invGx23 = ( -Gx11*Gx32 )/detGx;
    invGx31 = 0;
    invGx32 = invGx23;
    invGx12 = ( Gx11*Gx22 )/detGx;

    tauC[contControl][0]= 0;
    tauC[contControl][1]= 0 ;
    tauC[contControl][2]= 0;

    Tau[contControl][0] = cg11 *(-Fg[0] + v[0] + KI1*s[0]);

```

```

Tau[contControl][1] = cg22 *(-Fg[1] + v[1] + KI2*s[1]) + cg23 * (-
    Fg[2] + v[2] + KI3*s[2]);
Tau[contControl][2] = cg32 *(-Fg[1] + v[1] + KI2*s[1]) + cg33 * (-
    Fg[2] + v[2] + KI3*s[2]);
    }

fprintf(Dat, "%f_%f_%f_%f_%f_%f_%f_%f_%f_%f_%f_%f_%f_%f_
    %f_%f_%f_%f_%f_%f_%f_%f_%f_%f_%f_%f_%f_%f_
    %f_%f_%f_%f_%f_%f_%f_%f_%f_%f_%f_%f\n", t, e[contControl][0], e[
    contControl][1], e[contControl][2], ep[contControl][0], ep[
    contControl][1], ep[contControl][2], ei[contControl][0], ei[
    contControl][1], ei[contControl][2], eii[contControl][0], eii[
    contControl][1], eii[contControl][2], s[0], s[1], s[2], v[0], v
    [1], v[2], Tau[contControl][0], Tau[contControl][1], Tau[
    contControl][2], position[0], position[1], position[2], positiond
    [0], positiond[1], positiond[2], q[0], q[1], q[2], qd[0], qd[1],
    qd[2], H, Fg[0], Fg[1], Fg[1], Fg[2]);

fprintf(Dat2, "%f_%f_%f_%f_%f_%f_%f_%f_%f\n", t, cg11, cg22,
    cg23, cg33, m11, m22, m23, m33 );
fprintf(Dat3, "%f_%f_%f_%f_%f_%f\n", t, wCg33[0], Fg[0], Fg[1], Fg
    [2]);

res[0]=(JTI11*Tau[contControl][0])+(JTI12*Tau[contControl][1])+(JTI13
    *Tau[contControl][2]);
res[1]=(JTI21*Tau[contControl][0])+(JTI22*Tau[contControl][1])+(JTI23
    *Tau[contControl][2]);
res[2]=(JTI31*Tau[contControl][0])+(JTI32*Tau[contControl][1])+(JTI33
    *Tau[contControl][2]);

printf("%f_%f_%f_%f_%f\n", t, q[0], q[1], q[2]);

if(res[0]<5 && res[1]<5 && res[2]<5)
{
    res[0]=res[0];   res[1]=res[1];   res[2]=res[2];  }
else
{
    res[0]=0;       res[1]=0;       res[2]=0;        }
force[0]=res[0];   force[1]=res[1];   force[2]=res[2];
hdSetDoublev(HD_CURRENT_FORCE, force);

```

```
    contControl++;
    hdEndFrame(hHD);
    return HD_CALLBACK_CONTINUE;
}
int inicia()
{
    HDErrorInfo error;
    HHD hHD = hdInitDevice(HD_DEFAULT_DEVICE);
    if (HD_DEVICE_ERROR(error = hdGetError()))
    {
        printf("Error_1\n");
        return -1;
    }
    hBoyCallBack = hdScheduleAsynchronous(boyCallBack, 0,
        HD_MAX_SCHEDULER_PRIORITY);
    hdEnable(HD_FORCE_OUTPUT);
    hdStartScheduler();
    if (HD_DEVICE_ERROR(error = hdGetError()))
    {
        printf("Error_2\n");
        return -1;
    }
}
int _tmain(int argc, _TCHAR* argv[])
{
    printf("2.-_Fuzzy_Control_\n");
    printf("");
    system("PAUSE");
    cin >>= 2;
    tiempoE = 10;
    Dat = fopen("Datos.txt", "w");
    Dat2 = fopen("Datos2.txt", "w");
    Dat3 = fopen("Datos3.txt", "w");

    inicia();
    while (!_kbhit() && t < tiempoE)
    {}
    fclose(Dat);
```

```
        fclose(Dat2);
        fclose(Dat3);
        return 0;
}

//FUNCION PERTENENCIA TRIANGULAR
double triangular(double a, double b, double c, double x)
{
    //// b es el punto medio
    double mu;
    if (x >= c || x <= a)
    {
        mu = 0;
    }

    if (x > a && x <= b)
    {
        mu = (x - a)/(b - a);
    }

    if (x < c && x > b)
    {
        mu = (c - x)/(c - b);
    }
    return mu;
}

//FUNCION PERTENENCIA TRAPEZOIDAL
double trapezoidal(double a, double b, double c, double d, double x)
{
    double mu;
    if (x >= d || x <= a)
    {
        mu = 0;
    }

    if (x > a && x < b)
    {
```

```
        mu = (x - a)/(b - a);
    }

    if (x >= b && x <= c)
    {
        mu = 1;
    }

    if (x > c && x < d)
    {
        mu = (d - x)/(d - c);
    }
    return mu;
}

//FUNCION PERTENENCIA GAUSSIANA
double gaussiana(double a, double b, double x)
{
    double mu;
    mu = exp(-b*pow((x-a),2));
    return mu;
}

//FUNCION PERTENENCIA SIGMOIDAL
double sigmoidal(double a, double b, double x)
{
    double mu;
    mu = 1/(1 + exp( -a*(x-b) ) );
    return mu;
}

//FUNCION PERTENENCIA CAMPANA
double campana(double a, double b, double c, double x)
{
    double mu;
    mu = 1/( 1 + pow(abs((x-c)/a), (2*b)) );
    return mu;
}
```

```
//FUNCION PARA CALCULAR INTEGRAL  
double Integral (double sv, double sv_1)  
{  
    double k1, k2, k4, medio;  
    medio = (sv+sv_1)/2;  
    k1=sv_1;  
    k2=medio;  
    k4=sv;  
    return (0.001*(medio));  
}
```


Apéndice B

Código Matlab para Simulación

B.1. Trayectoria deseada

```
function [xd, yd, zd, q0d, q1d, q2d, qp0d, qp1d, qp2d, qpp0d, qpp1d,
        qpp2d] = fcn(t,l1, l2)
ts=5;
radio = 0.015;
w=2*pi/ts;
ro=w*t;
h0=-0.015;
h2=0;
xd = h0+radio*cos(ro);
yd = t/70;
zd = h2+radio*sin(ro);
xpd = -radio*w*sin(ro);
ypd = 1/70;
zpd = radio*w*cos(ro);
xppd = -radio*(w*w) * cos(ro);
yppd = 0;
zppd = -radio*(w*w) * sin(ro);

q0d=atan2(xd,zd+l1);
Rd=sqrt((xd^2) + ((zd+l1)^2));
rd=sqrt((xd^2) + ((yd-l2)^2) +((zd+l1)^2));
betad=atan2(yd-l2,Rd);
aux1d=((l1^2)+(rd^2)-(l2^2))/(2*l1*rd);
```

```

gammd=acos(aux1d);
q1d = gammd+betad;
alfad =acos((l1^2 + l2^2 -rd^2)/(2*l1*l2));
q2d = q1d+alfad -(pi/2);

Jd11= l1 * cos(q0d) * cos(q1d) + (l2 * sin(q2d) * cos(q0d));
Jd12= -l1 * sin(q0d) * sin(q1d);
Jd13= l2 * sin(q0d) * cos(q2d);
Jd21= 0;
Jd22= l1 * cos(q1d);
Jd23= l2 * sin(q2d);
Jd31= -l1 * sin(q0d) * cos(q1d) - (l2 * sin(q0d) * sin(q2d));
Jd32= -l1 * sin(q1d) * cos(q0d);
Jd33= l2 * cos(q0d) * cos(q2d);
DJd=Jd11*Jd22*Jd33-Jd11*Jd32*Jd23+Jd12*Jd23*Jd31-Jd12*Jd21*Jd33+Jd13*
    Jd21*Jd32-Jd13*Jd22*Jd31;

JI11=(Jd22*Jd33-Jd32*Jd23)/DJd;
JI12=(-Jd12*Jd33+Jd13*Jd32)/DJd;
JI13=-(-Jd12*Jd23+Jd13*Jd22)/DJd;
JI21=-(-Jd23*Jd31+Jd21*Jd33)/DJd;
JI22=-(-Jd11*Jd33+Jd13*Jd31)/DJd;
JI23= (-Jd11*Jd23+Jd13*Jd21)/DJd;
JI31=(Jd21*Jd32-Jd22*Jd31)/DJd;
JI32=-(-Jd11*Jd32-Jd12*Jd31)/DJd;
JI33=(Jd11*Jd22-Jd12*Jd21)/DJd;

qp0d =JI11*xpd+JI12*ypd+JI13*zpd;
qp1d =JI21*xpd+JI22*ypd+JI23*zpd;
qp2d =JI31*xpd+JI32*ypd+JI33*zpd;

Jpd11 = -l1 * sin(q0d) * cos(q1d) * qp0d - l2 * sin(q0d) * sin(q2d) *
    qp0d - l1 * sin(q1d) * cos(q0d) * qp1d + l2 * cos(q0d) * cos(q2d)
    ) * qp2d ;
Jpd12 = -l1 * sin(q1d) * cos(q0d) * qp0d - l1 * sin(q0d) * cos(q1d) *
    qp1d;
Jpd13 = l2 * cos(q0d) * cos(q2d) * qp0d - l2 * sin(q0d) * sin(q2d) *
    qp2d;

```

```

Jpd21 = 0;
Jpd22 = -l1 * sin(q1d) * qp1d;
Jpd23 = l2 * cos(q2d) * qp2d;

Jpd31 = -l1 * cos(q0d) * cos(q1d) * qp0d - l2 * sin(q2d) * cos(q0d) *
        qp0d + l1 * sin(q0d) * sin(q1d) * qp0d - l2 * sin(q0d) * cos(q2d)
        * qp2d ;
Jpd32 = l1 * sin(q0d) * sin(q1d) * qp0d - l1 * cos(q0d) * cos(q1d) *
        qp1d ;
Jpd33 = -l2 * sin(q0d) * cos(q2d) * qp0d - l2 * sin(q2d) * cos(q0d) *
        qp2d ;

qpp0d = JId11 * ( xppd - (Jpd11*qp0d) + (Jpd12*qp1d) + (Jpd13 * qp2d)
        ) + JId12 * ( yppd - (Jpd21*qp0d) + (Jpd22*qp1d) + (Jpd23 * qp2d)
        ) + JId13 * ( zppd - (Jpd31*qp0d) + (Jpd32*qp1d) + (Jpd33 * qp2d)
        );
qpp1d = JId21 * ( xppd - (Jpd11*qp0d) + (Jpd12*qp1d) + (Jpd13 * qp2d)
        ) + JId22 * ( yppd - (Jpd21*qp0d) + (Jpd22*qp1d) + (Jpd23 * qp2d)
        ) + JId23 * ( zppd - (Jpd31*qp0d) + (Jpd32*qp1d) + (Jpd33 * qp2d)
        );
qpp2d = JId31 * ( xppd - (Jpd11*qp0d) + (Jpd12*qp1d) + (Jpd13 * qp2d)
        ) + JId32 * ( yppd - (Jpd21*qp0d) + (Jpd22*qp1d) + (Jpd23 * qp2d)
        ) + JId33 * ( zppd - (Jpd31*qp0d) + (Jpd32*qp1d) + (Jpd33 * qp2d)
        );

```

B.2. Modelo cinemático directo

```

function [x, y, z]= fcn(l1 ,l2 ,q0 ,q1 ,q2)
S1=sin(q0);
S2=sin(q1);
S3=sin(q2);
C1=cos(q0);
C2=cos(q1);
C3=cos(q2);

x = S1*(l1*C2+l2*S3);

```

```
y = l2-12*C3+l1*S2;
z = -l1 + C1*(l1*C2+l2*S3);
```

B.3. Modelo PHANToM Premium 1.0

```
function [Qpp,Fx, g11, g22, g23, g33] = fcn(q1,q2,qp0,qp1,qp2,l1,l2,
    l3,l4,l5,ma, mc,mbe,mdf,tau1, tau2, tau3)

g=9.81; %Constante gravitacional
Qp=[qp0; qp1; qp2];
Tau=[tau1; tau2; tau3];

M11aux=(-0.125*(l2^2*ma+4*l3^2*mc)*cos(2*q2))+(0.125*(l2*ma+l3*mc)*
    cos(q1)*sin(q2));
M11=((0.5*l1^2+0.125*l2^2)*ma)+((0.125*l1^2+0.5*l3^2)*mc)+(0.125*l1
    ^2*(4*ma+mc)*cos(2*q1))+M11aux;
M22=l1^2*(ma+0.25*mc);
M23=-0.5*l1*(l2*ma+l3*mc)*sin(q1-q2);
M32=M23;
M33=0.25*l2^2*ma+l3^2*mc;
deno=M22*M33-M23*M32;
HM=[M11 0 0;0 M22 M23;0 0 M33];
InvM=inv(HM);

M11I=1/M11;
M12I=0;
M13I=0;
M21I=0;
M22I=M33/deno;
M23I=-M23/deno;
M31I=0;
M32I=-M32/deno;
M33I=M22/deno;
InvM=[M11I M12I M13I; M21I M22I M23I; M31I M32I M33I];

g11= M11I;
```

```

g22= M22I;
g23= M23I;
g33= M33I;

C11aux=0.25*cos(q2)*(2*l1*(l2*ma+l3*mc)*cos(q1)+...
(l2^2*ma+4*l3^2*mc)*sin(q2))*qp2;
C11=0.125*(-2*sin(q1)*(l1^2*(4*ma+mc)*cos(q1)+2*l1*(l2*ma+l3*mc)*sin(
q2))*qp1)+C11aux;
C12=-0.125*(l1^2*(4*ma+mc)*sin(2*q1)+4*l1*(l2*ma+l3*mc)*sin(q1)*sin(
q2))*qp0;
C13=-0.125*(-4*l1*(l2*ma+l3*mc)*cos(q1)*cos(q2)-(l2^2*ma+4*l3^2*mc)*
sin(2*q2))*qp0;
C21=-C12;
C22=0;
C23=0.5*l1*(l2*ma+l3*mc)*cos(q1-q2)*qp2;
C31=-C13;
C32=0.5*l1*(l2*ma+l3*mc)*cos(q1-q2)*qp1;
C33=0;
C=[C11 C12 C13; C21 C22 C23; C31 C32 C33];

cg2=-0.173;
cg3=1.663;
G11=0;
G21=(g*(l1*(ma+0.5*mc)+l5*mbe)*cos(q1))*cg2;
G31=(g*(0.5*l2*ma+l3*mc-l4*mdf)*sin(q2))*cg3;
G=[G11;G21;G31];

Fx=-InvM*(C*Qp+G);
Gx=InvM;
Qpp=Fx+(Gx*Tau);

```

B.4. Control τ_c

```

function taoC = fcn(Fx, G11, G22, G23, G33, v1, v2, v3, s1, s2, s3
,ko1, ko2, ko3, epsilonO)

Ko=[ko1 0 0 ; 0 ko2 0 ; 0 0 ko3];

```

```
v=[v1;v2;v3];
s=[s1;s2;s3];
Gx=[G11 0 0; 0 G22 G23; 0 G23 G33];
Ip=eye(3);
Inversa=inv(epsilonO*Ip + Gx*Gx');
taoC=Gx'*Inversa*(-Fx+v+Ko*s);
```

B.5. Control τ_0

```
function tao0 = fcn(Fx, G11, G22, G23, G33, v1, v2, v3, s1, s2, s3
    , ko1, ko2, ko3, epsilonO)

Ko=[ko1 0 0 ; 0 ko2 0 ; 0 0 ko3];
v=[v1;v2;v3];
s=[s1;s2;s3];
Gx=[G11 0 0; 0 G22 G23; 0 G23 G33];
Ip=eye(3);
Inversa=inv(epsilonO*Ip + Gx*Gx');
tao0=epsilonO*Inversa*(-Fx+v+Ko*s);
```

B.6. Control τ_r

```
function taoR=fcn(taoC1, taoC2, taoC3, tao01, tao02, tao03, s1, s2,
    s3, sigma0, delta, Qa, Qb, Qc)

Q0=Qa*eye(3);
iQ0=inv(Q0);
Q1=Qb*eye(3); %60 OK
iQ1=inv(Q1);
Q2=Qc*eye(3);
iQ2=inv(Q2);

s=[s1;s2;s3];
normas=norm(s);
```

```

taoC=[taoC1;taoC2;taoC3];
tao0=[tao01;tao02;tao03];

epsBF=[0.1;0.1;0.1];
epsBG=[0.1 0.1 0.1;0.1 0.1 0.1; 0.1 0.1 0.1];

taoR=s*((1/((2*sigma0*normas^2)+delta))*((s'*(Q0+Q1+Q2)*s)+(epsBF'*
    iQ0*epsBF) + ((epsBG*taoC)'*iQ1*(epsBG*taoC))+(tao0'*iQ2*tao0)));

```

B.7. δ

```

function deltap = fcn(taoC1, taoC2, taoC3, tao01, tao02, tao03, s1
    , s2, s3, sigma0, delta, Qa, Qb, Qc)

Q0=Qa*eye(3);
iQ0=inv(Q0);
Q1=Qb*eye(3); %60 OK
iQ1=inv(Q1);
Q2=Qc*eye(3);
iQ2=inv(Q2);

s=[s1;s2;s3];
normas=norm(s);

taoC=[taoC1;taoC2;taoC3];
tao0=[tao01;tao02;tao03];

epsBF=[.1;.1;.1];
epsBG=[.1 .1 .1;.1 .1 .1;.1 .1 .1];

niu=1;
p1= s'*(Q0+Q1+Q2)*s;
p2=_epsBF'* iQ0 *epsBF;
p3= (epsBG*taoC)'*iQ1*(epsBG*taoC);
p4=_tao0'*iQ2*tao0;
phi = p1++p2+p3+p4;

```

```
deltap = -niu*((0.5*phi)/(sigma0*normas^2+ delta));
```

B.8. Error de seguimiento filtrado

```
function [V, S] = fcn(qpp0d, qpp1d, qpp2d, e0, e1, e2, ep0, ep1, ep2,
    ei0, ei1, ei2, eii0, eii1, eii2)

lamda0= 100;  alfa0= 90;  gama0=50;
lamda1= 100;  alfa1= 90;  gama1=50;
lamda2= 100;  alfa2= 90;  gama2=50;

s0 = ep0 + (lamda0*e0) + (alfa0*ei0) + (gama0*eii0);
s1 = ep1 + (lamda1*e1) + (alfa1*ei1) + (gama1*eii1);
s2 = ep2 + (lamda2*e2) + (alfa2*ei2) + (gama2*eii2);

v0 = qpp0d + (lamda0*ep0) + (alfa0*e0) + (gama0*ei0);
v1 = qpp1d + (lamda1*ep1) + (alfa1*e1) + (gama1*ei1);
v2 = qpp2d + (lamda2*ep2) + (alfa2*e2) + (gama2*ei2);

S=[s0 ; s1 ; s2 ];
V=[v0 ; v1 ; v2 ];
```