



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA

**SISTEMA HÁPTICO PARA LA OPERACIÓN MAESTRO-
ESCLAVO DE UN ROBOT INDUSTRIAL**

TESIS

Que como parte de los requisitos para obtener el grado de:

MAESTRO EN CIENCIAS (MECATRÓNICA)

Presenta

Ing. Francisco Javier Mendoza Galindo

Dirigido por

Dr. Roque Alfredo Osornio Ríos

San Juan del Río, Querétaro, Noviembre de 2014



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Maestría en Ciencias (Mecatrónica)

Sistema Háptico para la Operación Maestro-Esclavo de un Robot Industrial

Tesis

Que como parte de los requisitos para obtener el Grado de
Maestría en Ciencias Mecatrónica

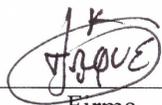
Presenta:

Ing. Francisco Javier Mendoza Galindo

Dirigido por:

Dr. Roque Alfredo Osornio Ríos

Dr. Roque Alfredo Osornio Ríos
Presidente



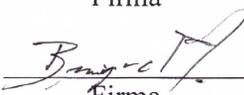
Firma

Dr. Luis Morales Velázquez
Secretario



Firma

M. en C. Benigno Muñoz Barrón
Vocal



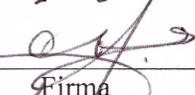
Firma

Dr. Jesús Rooney Rivera Guillén
Suplente



Firma

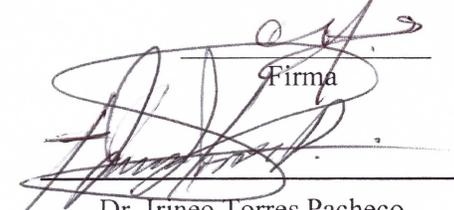
Dr. Carlos Santiago López Cajún
Suplente



Firma



Dr. Aurelio Domínguez González
Director de la Facultad de Ingeniería



Dr. Irineo Torres Pacheco
Director de Investigación y Posgrado

Resumen

Los robots cumplen con una importante función en el área industrial ya que realizan tareas de manera eficiente, sin embargo, han adquirido auge en campos como medicina donde son guiados por un operador para realizar intervenciones quirúrgicas. Este trabajo presenta la operación maestro-esclavo de un robot industrial utilizando un dispositivo háptico de 3 grados de libertad como maestro y un robot PUMA como esclavo. El uso de un dispositivo háptico permite que el usuario realice los movimientos de una manera natural ya que el dispositivo háptico se mueve a través del espacio. Se obtuvieron las ecuaciones de cinemática inversa para robots PUMA con el objetivo de que el movimiento del dispositivo háptico sea replicado lo más rápido posible y del mismo modo. Pese a que se tiene un maestro con menor número de grados de libertad que el esclavo, se mantiene fija la orientación del órgano terminal del robot PUMA. Para verificar el correcto funcionamiento, se realizó una interfaz utilizando Simulink para implementar el sistema háptico, realizando pruebas de seguridad con el dispositivo háptico y el manipulador, dicha interfaz puede ser utilizada para verificar otras trayectorias y observar el movimiento de los eslabones del robot PUMA. Para la implementación del robot PUMA real y el dispositivo háptico, se utilizó un controlador PID dentro de una implementación en FPGA. Finalmente se realizaron pruebas donde se evaluó el desempeño del sistema.

(Palabras clave: maestro-esclavo, dispositivo háptico, robot PUMA, cinemática inversa, FPGA)

Abstract

Robots play a key role on industrial applications since they execute predefined tasks efficiently; however, they have acquired importance in other research fields, such as medicine, as they are guided by an operator to execute surgical interventions. This work shows the master-slave operation for an industrial robot. The methodology uses a 3 degrees of freedom haptic device as the master, and a PUMA robot as slave. The use of the haptic device allows the user to develop movements in a natural way, because it moves inside the 3D space. In order to achieve the replication of the haptic movements in a quickly and accurately way, the inverse kinematics equations for PUMA robots are calculated. Since the haptic device has less degrees of freedom than the slave, the end effector orientation is fixed. To ensure that the system behaves in the desired way, an interface is designed and implemented using Simulink, where the inverse kinematic equations are implemented in order to design an algorithm for restricting the available workspace. Moreover, the interface can be used to test trajectories before they are programmed in the real-life robot, and the links movement can be observed. For the implementation in a real robot and haptic device, an FPGA-based proprietary processor is used to control the robot by employing a PID controller. Several tests are carried out in order to demonstrate the performance of the proposed controller.

(Keywords: master-slave, haptic device, PUMA robot, inverse kinematics, FPGA)

Dedicatorias

A mis padres y hermanos quienes me
han brindado apoyo incondicional
durante toda mi vida académica

Agradecimientos

Primero que nada agradezco a mi familia cuyo apoyo durante toda mi vida académica fue motivación para terminar la carrera, a mi asesor el doctor Roque que sus consejos y su orientación me ayudó a completar este escrito, a Benigno con quien viví muchas experiencias, quien fue mi mentor y un gran amigo durante toda la carrera, al doctor Luis quien me asesoró y aconsejó en los momentos de duda. Por su puesto a mis compañeros de generación Carlos e Irving con quienes compartí buenos y malos momentos en la carrera pero que al final, al apoyarnos mutuamente logramos salir adelante, también agradezco a todos nuestros compañeros de posgrado y profesores donde encontré muchos buenos amigos, finalmente agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca otorgada con el número de becario (289473) para realizar los estudios en la Universidad Autónoma de Querétaro.

Índice general

Resumen	1
Abstract.....	2
Dedicatorias	3
Agradecimientos.....	4
Índice general	5
Índice de figuras	7
Índice de tablas	9
CAPÍTULO I.....	10
INTRODUCCIÓN.....	10
1.1. Antecedentes.....	11
1.2. Descripción del problema.....	14
1.3. Hipótesis y Objetivos	14
1.3.1. Hipótesis	14
1.3.2. Objetivos.....	15
1.4. Justificación.....	15
1.5. Planteamiento general.....	16
CAPÍTULO II.....	18
FUNDAMENTACIÓN TEÓRICA	18
2.1. Estado del arte	18
2.2. Dispositivo Novint Falcon.....	20
2.3. La matriz de rotación.....	20
2.4. Manipuladores Seriales.....	21
2.6. Cinemática directa	24
2.7. Cinemática inversa	25
2.8. El problema de la Posición	26
2.9. El problema de la Orientación.....	27
2.10. Controlador PID	29

CAPÍTULO III	32
METODOLOGÍA.....	32
3.1. Documentación.....	33
3.2. Simulación.....	34
3.2.1. Implementación en Simulink.....	34
3.2.1.1. Traslación y Escalamiento.....	35
3.2.1.2. Inclusión de las ecuaciones de cinemática inversa.....	35
3.2.1.3. Compensación debido a la configuración.....	36
3.2.2. Diseño de un algoritmo de seguridad.....	36
3.2.3. Diseño del modelo virtual.....	37
3.3. Implementación.....	38
3.3.1. Implementación en C++.....	38
3.3.2. Diseño del controlador.....	39
3.3.2.1. Modelado del sistema.....	39
3.3.2.2. Sintonización del controlador.....	40
3.3.3. Implementación del sistema digital.....	41
3.3.3.1. Máquina principal.....	42
3.3.3.2. Módulo de comunicación.....	43
3.3.3.3. Sistema de control.....	44
3.3.3.4. Módulo de escritura y lectura de memoria.....	44
CAPÍTULO IV	46
PRUEBAS Y RESULTADOS	46
4.1. Simulación.....	46
4.2. Cálculo del periodo de muestreo.....	48
4.3. Pruebas de seguimiento.....	50
4.4. Pruebas finales.....	54
CAPÍTULO V	58
CONCLUSIONES.....	58
BIBLIOGRAFÍA.....	60
Apéndice A.....	64
Artículo publicado en congreso.....	64

Índice de figuras

FIGURA 1.1. Esquema de un sistema maestro-esclavo.....	10
FIGURA 1.2. Diagrama a bloques del trabajo propuesto.....	17
FIGURA 2.1. Algunos de los dispositivos hápticos disponibles en el mercado a) PHANTOM Premium 1.0, b) PHANTOM Omni y c) Falcon.....	20
FIGURA 2.2. Signo de sentido de giro a) α_i y b) θ_i	23
FIGURA 2.3. Diagrama esquemático de los primeros 3 ejes.....	26
FIGURA 2.4. Diagrama esquemático de los últimos tres ejes.....	28
FIGURA 2.5. a) Diagrama simplificado de un sistema en lazo cerrado y b) Diagrama a bloques de la estructura típica de un controlador PID.....	30
FIGURA 3.1. Metodología propuesta.....	33
FIGURA 3.2. Implementación en Simulink.....	34
FIGURA 3.3. a) Escalamiento de valores b) Implementación de la detección de objetos.....	35
FIGURA 3.4. Diseño de medida de seguridad.....	36
FIGURA 3.5. Diseño del manipulador virtual en V-Realm Builder.....	37
FIGURA 3.6. Implementación del sistema en C++.....	39
FIGURA 3.7. Implementación del algoritmo de mínimos cuadrados recursivos.....	40
FIGURA 3.8. Implementación del controlador.....	41
FIGURA 3.9. Implementación del sistema digital.....	42
FIGURA 3.10. Implementación de a) la máquina de estados principal y b) agrupamiento de datos.....	43
FIGURA 3.11. Módulo de comunicación USB.....	43
FIGURA 3.12. Esquema del sistema de control PID.....	44
FIGURA 3.13. Módulo de escritura y lectura de RAM.....	45
FIGURA 4.1. Sistema simulado.....	46
FIGURA 4.2. a) Lecturas de los ejes simulados, b) Fuerza ejercida por el Falcon.....	47
FIGURA 4.3. Implementación para obtener el retardo.....	48
FIGURA 4.4. Cuentas calculadas y enviadas hacia el FPGA.....	49
FIGURA 4.5. Retardos del sistema.....	49
FIGURA 4.6. Sistema dedicado a la sintonía y prueba de controladores.....	50
FIGURA 4.7. Prueba del controlador eje 1 a) Seguimiento de perfil, b) Error.....	51
FIGURA 4.8. Prueba del controlador eje 2 a) Seguimiento de perfil, b) Error.....	52
FIGURA 4.9. Prueba del controlador eje 3 a) Seguimiento de perfil, b) Error.....	52
FIGURA 4.10. Prueba del controlador eje 5 a) Seguimiento de perfil, b) Error.....	52
FIGURA 4.11. Prueba del controlador eje 6 a) Seguimiento de perfil, b) Error.....	53
FIGURA 4.12. Implementación del sistema completo.....	54
FIGURA 4.13. a) Trayectoria generada por el Falcon, b) Respuesta del sistema háptico.....	55

FIGURA 4.14. Prueba de trayectoria eje 1, a) Seguimiento, b) Error.....	55
FIGURA 4.15. Prueba de trayectoria eje 2, a) Seguimiento, b) Error.....	55
FIGURA 4.16. Prueba de trayectoria eje 3, a) Seguimiento, b) Error.....	56
FIGURA 4.17. Prueba de trayectoria eje 5, a) Seguimiento, b) Error.....	56
FIGURA 4.18. Prueba de trayectoria eje 6, a) Seguimiento, b) Error.....	56

Índice de tablas

TABLA 1. Parámetros D-H del robot ROMAT de los ejes 1, 2 y 3.....	26
TABLA 2. Parámetros D-H del robot ROMAT de los ejes 4, 5 y 6.....	28
TABLA 3. Relación para discretizar los valores P, I y D.....	31
TABLA 4. Equivalencia de los parámetros para el controlador PID discreto.....	31
TABLA 5. Controladores analógicos	51
TABLA 6. Error cuadrático medio para perfiles	53
TABLA 7. Error cuadrático medio prueba	56
TABLA 8. Abreviaturas utilizadas en este trabajo	63

CAPÍTULO I

INTRODUCCIÓN

En la actualidad los robots pre-programados cumplen con una importante función para el desarrollo de las actividades económicas e industriales ya que permiten realizar tareas simples de una manera rápida y eficiente. La mayoría de los robots se enfocan a tareas previamente diseñadas en ambientes estáticos; sin embargo, si el robot trabaja en ambientes dinámicos, será necesaria la intervención humana.

La tecnología háptica se basa en la realimentación de tacto sobre un dispositivo aplicando fuerzas, vibraciones o movimientos sobre el usuario (un diagrama general se presenta en la Figura 1.1); por otra parte, un sistema de control sobre un manipulador en una configuración maestro-esclavo busca el seguimiento del esclavo a los movimientos del maestro. Debido a que no se cuenta con un manipulador equipado con sensores de proximidad, en el presente trabajo cuando se hace mención de un sistema háptico, se refiere un modelo simulado ya que vía software se puede programar las funciones de los sensores; mientras debe entenderse una configuración maestro-esclavo a una implementación física que retribuye fuerza al dispositivo háptico (Falcon) con la función de los sensores programada.

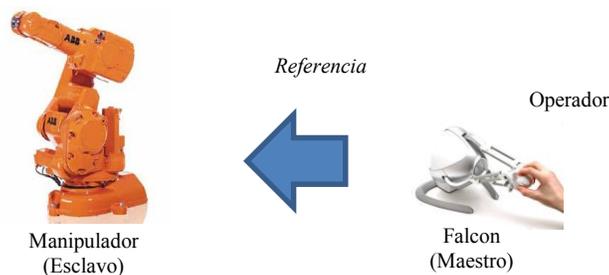


FIGURA 1.1. Esquema de un sistema maestro-esclavo.

La búsqueda de un control maestro-esclavo es el objetivo principal de este trabajo, pero también se busca el diseño de un entorno virtual como herramienta de simulación en el cual se pueda implementar el sistema háptico, para ello en el capítulo II, se hace una revisión general de conceptos de robótica y control, en el capítulo III se plantea la metodología

propuesta la cual consiste de una etapa de documentación, seguido de una etapa de simulación, para posteriormente realizar la implementación, en el capítulo IV se plantean cuatro casos de estudio con el objetivo de demostrar el correcto funcionamiento del sistema, finalmente en el capítulo V se discuten los resultados obtenidos así como las perspectivas de este trabajo.

1.1. Antecedentes

En la Universidad Autónoma de Querétaro se han realizado algunos trabajos enfocados al control de posicionamiento de manipuladores robóticos, empezando por el trabajo de Martínez-Prado et al., (2012) quienes desarrollaron un controlador de movimiento para aplicaciones robóticas basado en una plataforma FPGA (Field Programmable Gate Array, Arreglo de Compuertas Programables en Campo) enfocado a un prototipo de manipulador con 3 grados de libertad. Las implementaciones previas al uso de FPGA requerían de microprocesadores comerciales específicos lo que limitan las aplicaciones. Jaen-Cuellar et al., (2012) desarrollaron un controlador de movimiento aplicado a una fresadora con tres grados de libertad mediante el uso de curvas paramétricas, las cuales permiten realizar movimientos suaves con pocos datos; además, su implementación en hardware permite que la aplicación sea en línea. Para que el seguimiento sea apropiado es necesario contar con la ubicación específica del manipulador, por lo que en los trabajos de Rodríguez-Donate et al., (2012) y Muñoz-Barrón et al., (2014) se enfocaron en ubicar un robot tipo PUMA de 6 grados de libertad que a diferencia de las técnicas convencionales que utilizan sensores ópticos únicamente, cuya principal desventaja es el hecho de que no pueden percibir deformaciones ni juegos mecánicos, en su trabajo se aplica la fusión de sensores (encoder, giroscopio y acelerómetro), para medir la posición, velocidad angular y aceleración, respectivamente. El procesamiento se lleva a cabo por medio de un filtro Kalman que combina la contribución de información de cada sensor. Se demuestra que la fusión mejora la aproximación e información que cada uno de los sensores puede ofrecer. Debe destacarse que toda la implementación se realizó sobre una plataforma FPGA. En cuanto al área de control, Jaen-Cuellar et al., (2013) ofrecieron una nueva metodología para la sintonización de controlador PID por medio de la utilización de los métodos de respuesta en

frecuencia y margen de fase con algoritmos genéticos. La metodología se aplica a un robot industrial, presentando una mejora en cuanto a los parámetros de control respecto al método de margen de fase, que repercuten finalmente en el posicionamiento.

En cuanto a las herramientas de software, se han desarrollado varios simuladores de robots. López (2007) implementó un sistema háptico que consiste en un dispositivo háptico tipo lápiz y un entorno virtual que simula a un paciente, con el fin de entrenar a médicos en cirugías mínimamente invasivas. La mayoría de las herramientas de software para la conexión entre el háptico que se utilizará, (el dispositivo Falcon) y la computadora, las proporcionó la empresa fabricante Novint; sin embargo, se encuentran bajo plataformas de software obsoletas.

Actualmente los dispositivos hápticos han mejorado mucho en cuanto a su diseño y precisión, lo que ha permitido el desarrollo de un sin fin de trabajos. Morgan (2011) desarrolló un control háptico con un dispositivo Falcon para manipular una mano robótica que tenía implementado en sus dedos sensores de presión, que proporcionan información sobre la consistencia del objeto que se toma, además realizó un software que provee al usuario de información de los sensores en forma visual. Algunos trabajos mejoran el control mediante el remplazo del dispositivo háptico por otro manipulador, mediante un control bilateral, como los de Tanaka et al., (2009) y Mizoguchi et al., (2012), en el que un control bilateral obedece al principio de acción y reacción, y consiste en un manipulador “maestro” que controla a otro manipulador, “esclavo”. Debido a que se necesita controlar ambos manipuladores, el primero utilizó una implementación en FPGA para reducir el tiempo de procesamiento; mientras que la técnica del segundo se aplicó mediante un manipulador maestro pequeño, en este caso la investigación fue orientada al escalado de la velocidad del manipulador esclavo y a mejorar la sensibilidad por medio de un sensor giratorio. Los resultados obtenidos demostraron un incremento en la sensibilidad en el manipulador maestro. En ambos casos, la técnica de control bilateral pretende reproducir fielmente los movimientos del operador debido a que tanto maestro como esclavo tienen la misma cantidad de eslabones. En su trabajo, Chotiprayanakul y Liu (2009) explican el método para escalado y control en el caso de manipuladores largos controlados por dispositivos hápticos pequeños. Para evitar colisiones,

se utilizó un láser para indicar la distancia del robot hacia las paredes; de esta manera se evitan colisiones oponiendo mayor fuerza al operador según la cercanía hacia la pared.

Debido a que se ha reducido el costo de algunos sistemas hápticos se han incrementado el número de investigaciones de sistemas de control háptico, por ejemplo, orientados hacia el campo de la industria. Radi et al., (2009), se enfocaron en un sistema háptico dedicado al ensamble de piezas no solo en lugares donde interactúan máquinas sino que también se busca una cooperación hombre-máquina; se analizan los espacios de trabajo por cuestiones de seguridad, y también proponen una manera práctica de experimentar los movimientos del manipulador mediante el uso de una pared virtual. Gran parte de las aplicaciones maestro-esclavo se llevan a cabo en robots móviles utilizados para exploración. Mitsou et. al., (2006) utilizó una cámara en conjunto con un sistema maestro-esclavo para generar un mapa más detallado del ambiente. También se han desarrollado investigaciones buscando la teleoperación del sistema maestro-esclavo. Glamnik y Šafarič (2012) desarrollaron un software para el control de un robot KUKA KR 5 con un dispositivo háptico tipo lápiz. Utilizando los protocolos de comunicación FireWire y TCP/IP lograron un intercambio de datos entre el maestro y el esclavo en tiempo real utilizando el protocolo Ethernet. Las aplicaciones con sistemas hápticos se han expandido incluso al área de medicina donde se les ha utilizado en intervenciones quirúrgicas remotas como lo muestra el trabajo de Hyuk et al., (2009) quienes diseñaron un sistema maestro-esclavo completo de teleoperación, el cual permitió a un cirujano realizar una operación a 40 km de distancia. El sistema consistía de un dispositivo háptico comercial, una computadora conectada a internet y un robot industrial que fue adaptado con el instrumental necesario para realizar la intervención. Cabe destacar que el retraso en la respuesta fue muy bajo respecto al tiempo de operación, tan solo de 40 ms, el cual es crítico en este tipo de aplicaciones, sin embargo su diseño fue para una aplicación específica y su reproducción podría representar un alto costo.

Como se ha observado, el uso de manipuladores robóticos en configuración maestro-esclavo con dispositivos hápticos han tomado gran auge en los últimos años, esto se debe en gran medida al avance tecnológico en cuanto a precisión y poder de cómputo, prueba de ello es la gran variedad de aplicaciones que se llevan a cabo, principalmente en manipulación de objetos peligrosos y en el área de medicina. En la Universidad Autónoma de Querétaro se

han logrado grandes aportes en el área de instrumentación y control de robots de múltiples grados de libertad, buscando siempre una óptima implementación, pero a la fecha no se tienen antecedentes de trabajos con sistemas hápticos. Ampliando la línea de investigación, se propone un sistema de control háptico de bajo costo para un robot en configuración maestro-esclavo el cual puede mejorar en muchos aspectos la experiencia del operador al acercarlo al entorno real del robot.

1.2. Descripción del problema

La mayoría de dispositivos para manejar robots industriales son muy simples y no se ajustan al movimiento natural del cuerpo, ya que consisten en perillas o botones y eso dificulta y retrasa los movimientos del operador pues él debe pensar en movimientos a los cuales no está acostumbrado. Los dispositivos hápticos profesionales que son ergonómicos, tienden a ser 20 o 200 veces más costosos que el Falcon.

Pocas investigaciones tratan de resolver la cinemática inversa, debido a la cantidad de cálculos que son requeridos. La mayoría de los trabajos realizan asignación directa hacia los ejes, utilizando dispositivos hápticos con los mismos DoFs, lo cual simplifica la tarea pero incrementa el costo. Los trabajos que sí obtienen las ecuaciones de cinemática inversa manejan manipuladores con menos de 6 DoFs.

En cuanto a trabajos donde se implementan los sistemas maestro esclavo, usualmente se utiliza programación en software para el controlador, lo que conlleva a generar retardos causados por la carga computacional ejercida por otros programas en uso, lo que a la larga tiende a desincronizar el sistema de control.

La mayoría de las investigaciones reportadas en la literatura sólo se enfocan a la operatividad del manipulador, siendo pocos los trabajos que implementan alguna medida de seguridad.

1.3. Hipótesis y Objetivos

1.3.1. Hipótesis

Es posible desarrollar un sistema maestro-esclavo con un dispositivo háptico de bajo costo para el control de un robot industrial mediante las ecuaciones de cinemática inversa y

la implementación de control en tarjetas FPGA, para la mejora en el desempeño del robot en cuanto a su manejo y operación.

1.3.2. Objetivos

Objetivo general:

Desarrollar un sistema maestro-esclavo para la operación de un robot industrial tipo PUMA, mediante una plataforma de bajo costo “Falcon”.

Objetivos particulares:

- Realizar la conexión del dispositivo háptico con la PC, para obtener sus referencias.
- Obtener las ecuaciones de cinemática inversa para robots tipo PUMA con el fin de trabajar las referencias en coordenadas rectangulares.
- Desarrollar una interfaz gráfica utilizando Simulink para realizar pruebas sobre el Falcon y el manipulador.
- Implementar el sistema digital para controlar el manipulador mediante un controlador PID.
- Validar los resultados mediante pruebas de seguimiento, después utilizar el Falcon para observar su operatividad.

1.4. Justificación

La implementación de un dispositivo háptico como mando para el manejo de un manipulador agiliza y simplifica los movimientos del operador, ya que el movimiento se realiza de una forma más natural. La adaptación de un mando omnidireccional como lo es el Falcon, el cual originalmente se enfoca al área de videojuegos, mejora la experiencia del manejo y reduce los costos de la aplicación.

El implementar las ecuaciones de cinemática inversa en conjunto con el Falcon, permite trabajar los movimientos del manipulador a través del espacio euclidiano. De este modo, se observan en forma más clara los movimientos del manipulador y permite al usuario realizar movimientos fluidos.

La implementación de controladores dentro de un FPGA brinda mayor velocidad a los cálculos de los controladores, ya que las operaciones se realizan de forma paralela, lo que a su vez provee la sincronía necesaria que requiere la acción de control.

La adición de medidas de seguridad ayuda a prevenir daños materiales causados por descuidos humanos o por falta de experiencia, una de ellas es implementar una pared virtual que restrinja el movimiento dentro del espacio de trabajo.

La implementación de una simulación dentro de un entorno virtual siempre es de utilidad, puesto que al realizar un proceso se puede observar posibles fallas de trayectoria o espacio de trabajo, ya que un manipulador es un sistema altamente complejo en cuanto a su estructura física. Además, se puede utilizar para diversos tipos de pruebas ya sea para probar trayectorias o para simular un proceso con el dispositivo háptico, en el cual se le puede añadir el peso de la herramienta para proveer más realismo.

1.5. Planteamiento general

El propósito de esta investigación consiste en el desarrollo de un sistema maestro-esclavo utilizando un dispositivo háptico de tres DoFs para controlar un manipulador de seis DoFs, cuyos controladores digitales serán diseñados en una plataforma FPGA. Adicionalmente, se resuelve el problema de la cinemática inversa para una mayor comodidad en los movimientos y se propone el diseño de un entorno virtual con el fin de realizar pruebas y utilizar el dispositivo háptico para experimentar contacto con obstáculos.

El desarrollo general de este trabajo se basa en la conexión entre el dispositivo Falcon y la computadora, el obtener las ecuaciones de cinemática inversa, la creación de un entorno virtual, el desarrollo de un software en C++, descripción en hardware y pruebas finales. Cada una de dichas etapas se muestra en la Figura 1.2.

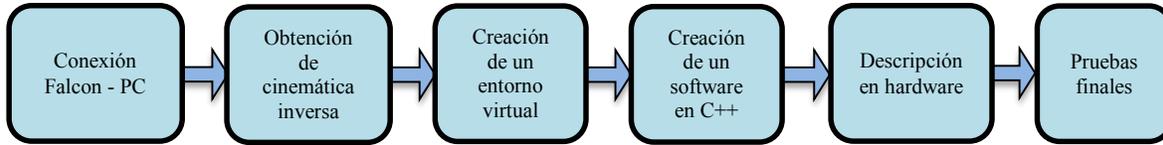


FIGURA 1.2. Diagrama a bloques del trabajo propuesto.

Conexión Falcon-PC: En esta etapa se realiza la conexión del Falcon hacia la computadora. A pesar de que el fabricante posee las herramientas de software para desarrolladores, éstas están bajo un software obsoleto por lo cual será necesario buscar alternativas.

Obtención de la cinemática inversa: En este paso se buscará la obtención de la cinemática inversa, es decir, la obtención de una relación de los ángulos requeridos en cada eje para una posición y orientación deseados.

Creación de entorno virtual: Para esta etapa se crea la interfaz gráfica que permita la simulación de los movimientos del manipulador dentro de un entorno virtual, utilizando el dispositivo háptico para guiarlo, así como la posibilidad de interacción con otros objetos.

Desarrollo de un software en C++: En esta etapa se realiza un software en C++, en el que se encuentran la comunicación con el Falcon, las ecuaciones de cinemática inversa, la conversión de ángulos a cuentas de encoder para ser enviadas por USB, la comunicación USB y un medio para almacenar los datos para posteriormente graficarlos.

Descripción en hardware: En esta etapa se busca integrar el módulo de comunicación hacia la computadora, los controladores PID requeridos, los drivers de los sensores, los drivers DAC, una memoria para almacenar los datos, entre otros bloques básicos requeridos. La herramienta utilizada para los diseños es el lenguaje descriptivo VHDL.

Pruebas finales: Por último el sistema se prueba, utilizando al Falcon como mando hacia el manipulador, verificando que los movimientos sean acordes al dispositivo háptico, se capturan los datos para demostrar el seguimiento del controlador.

CAPÍTULO II

FUNDAMENTACIÓN TEÓRICA

En este capítulo se hace una breve revisión a la historia tanto de los manipuladores como a los inicios de los primeros dispositivos con realimentación de fuerza y se dará una descripción del dispositivo háptico. También se introduce el tema de la geometría de manipuladores, comenzado con algunas propiedades de las matrices de rotación, conceptos básicos sobre manipuladores, las reglas para obtener los parámetros Denavit-Hartenberg (fundamental para el diseño), las ecuaciones de cinemática directa y la obtención de las ecuaciones de cinemática inversa, esta última, para ser utilizada a lo largo del trabajo. Finalmente se deja un apartado para revisar los controladores PID de los cuales se obtendrá la forma recursiva que será implementada de manera digital.

2.1. Estado del arte

La palabra robot fue usada por primera vez en el año de 1921, cuando el escritor checo Karel Capek (1890-1938) estrenó su obra Rossum's Universal Robot (R.U.R). Su origen es la palabra eslava robota, que quiere decir trabajo realizado de manera forzada (Rosheim, 1994). Los manipuladores han existido desde que el hombre creó la primera herramienta, pero no fue sino hasta finales de la Segunda Guerra Mundial que se desarrollaron los primeros manipuladores capaces de imitar el movimiento del brazo humano, debido a la necesidad de manipular recipientes con sustancias radioactivas.

La configuración de los primeros robots se enfocaba en diseños de tipo esférico y antropomórfico, usualmente para manipulaciones. Uno de los primeros manipuladores en aparecer es el tipo SCARA con 4 grados de libertad. A partir de ahí, el desarrollo de la robótica se ha basado en aumentar la movilidad, destreza y autonomía de las acciones. Una forma de lograrlo es aumentando los grados de libertad. Así, han aparecido manipuladores con mayor número de grados de libertad como el robot tipo PUMA de seis ejes tipo revoluta concebido en la universidad de Stanford en 1960 (Ángeles, 2007).

El desarrollo de manipuladores ha continuado tratando de imitar cada vez más al brazo humano y sus tres articulaciones (hombro, codo y muñeca) que producen un sistema de siete

grados de libertad, el primer robot con siete grados de libertad apareció en 1978, diseñado para alto desempeño y aplicaciones avanzadas en el espacio y sectores de defensa. Otro robot diseñado con siete grados de libertad es el FTS (Flight Telerobotic Servicer) desarrollado en 1986 para aplicaciones espaciales. El FTS requería de gran destreza, confiabilidad y precisión del orden de 0.025 mm.

El manipulador más parecido a un brazo humano es el Dextrous Arm debido a su antropomorfismo. Este robot fue patentado en 1989 y es de gran utilidad para aplicaciones específicas. Ha tenido un gran uso para investigaciones en universidades y centros de investigación (Rosheim, 1994).

Por otro lado, los primeros sistemas con retroalimentación de fuerza se construyeron en los 50's para manejar sustancias peligrosas a través de teleoperación. Mientras que la primera aplicación en un video juego fue desarrollada por SEGA en 1975 para un juego de conducción de motocicleta. El juego Moto-Cross implementó una retroalimentación háptica haciendo que el volante vibrase cuando ocurría una colisión. Más tarde, los juegos de carreras de autos incorporaron una retroalimentación háptica para simular una colisión o una vuelta muy cerrada. En 1997, apareció "Rumble Pack", un dispositivo auxiliar que se conectaba en los controles de Nintendo 64 y permitía una retroalimentación háptica. Además de video juegos, algunas instituciones dedicadas a la investigación utilizan dispositivos hápticos para aplicaciones de teleoperación. Actualmente existen dispositivos hápticos comerciales para desarrolladores, aquí se muestran tres de los más simbólicos en la Figura 2.1. El primero es el PHANTOM Premium 1.0 de (Sensable Technologies, Woburn, MA), el PHANTOM Omni (Sensable Technologies, Woburn, MA), y el Falcon (Novint, Albuquerque, NM). Los precios aproximados son de 20,000 dólares, 2,000 dólares y 200 dólares respectivamente (Salisbury et al., 2009).

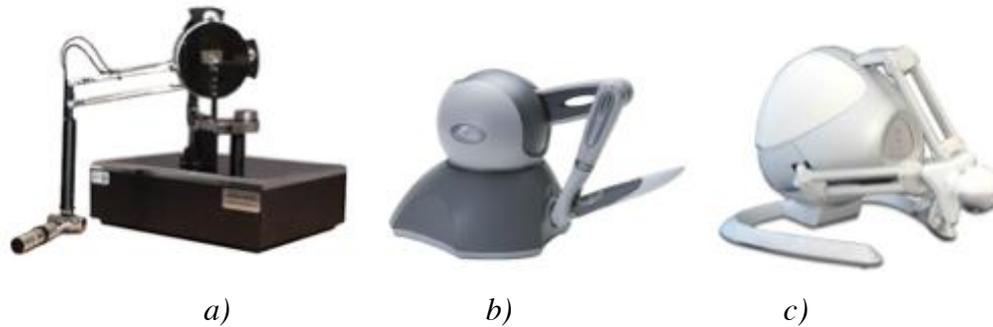


FIGURA 2.1. Algunos de los dispositivos hápticos disponibles en el mercado a) PHANTOM Premium 1.0, b) PHANTOM Omni y c) Falcon.

2.2. Dispositivo Novint Falcon

En 2007, el Novint Falcon (Figura 2.1c), fue introducido al mercado en la Muestra de Consumidores en Electrónica (CES), enfocado inicialmente para la industria de los videojuegos, lo convierte en el más barato y versátil dispositivo de retroalimentación háptica disponible.

El Novint Falcon requiere de una alimentación adicional para sus motores, consiste en un puerto USB y tiene tres manijas las cuales se unen a un mando removible que reemplaza al mouse convencional. El usuario puede mover el mando en tres dimensiones lo cual sigue la regla de la mano derecha, moverlo de derecha a izquierda genera un cambio en el eje X , adelante y atrás para el eje Z , y arriba y abajo para generar un cambio en el eje Y . Tres motores dentro del Falcon producen el efecto háptico sobre el dispositivo, mientras que tres encoders capturan la posición del mando con una precisión por debajo de milímetros y tasas de muestreo de mil veces por segundo. Cabe mencionar que la posición del mando está representada por los ejes (X, Y, Z) , las cuales son la combinación de los datos adquiridos por los tres encoders. El mando por default es una esfera removible con cuatro botones, pero también cuenta con una pistola la cual contiene tres botes y el gatillo se implementa como el cuarto botón (Novint, 2013).

2.3. La matriz de rotación

Antes de abordar el estudio de los manipuladores seriales se dará una breve introducción sobre la teoría detrás de la cinemática de los manipuladores. El espacio físico de tres

dimensiones es una particularidad de un espacio vectorial. Un espacio vectorial es un conjunto de objetos, llamados vectores, que siguen ciertas reglas algebraicas.

Una transformación lineal, representada con un operador L , de un espacio vectorial U dentro un espacio vectorial V , es una regla que asigna a cada vector \mathbf{u} de U a un vector \mathbf{v} de V , representado como $\mathbf{v} = L\mathbf{u}$, con L dotada con dos propiedades:

- i. Homogeneidad: $L(a\mathbf{u}) = a\mathbf{v}$;
- ii. Aditividad: $L(\mathbf{u}_1 + \mathbf{u}_2) = \mathbf{v}_1 + \mathbf{v}_2$

Algunos tipos particulares de transformaciones lineales de espacio tridimensional Euclidiano son las proyecciones, reflexiones y rotaciones. Particularmente, cuando L hace mapeo de U en sí misma, L es suficiente para la representación en forma de una matriz. Una matriz ortogonal es aquella que cumple con

$$\mathbf{R}^{-1} = \mathbf{R}^T \quad (1)$$

La ecuación (1) conduce a la siguiente interpretación

$$\mathbf{R}\mathbf{R}^T = \mathbf{1} \quad (2)$$

Sea \mathbf{R} , una transformación lineal que representa a una rotación, todas las rotaciones son matrices ortogonales (Ángeles, 2007).

2.4. Manipuladores Seriales

La forma más simple de modelar geoméricamente un manipulador robótico es por medio del concepto de cadena cinemática. Una cadena cinemática es un conjunto de cuerpos rígidos llamados eslabones, acoplados por pares cinemáticos, también conocidos como juntas cinemáticas. Un par cinemático es el acoplamiento de dos cuerpos rígidos para restringir su movimiento relativo. Se distinguen dos tipos básicos de pares cinemáticos, los superiores e inferiores. Los superiores son aquellos en que el contacto ocurre en un punto o una línea. Este tipo de acoplamiento ocurre en mecanismos de leva seguidor, trenes de engranes,

rodamientos, por dar algunos ejemplos. Un par cinemático inferior es aquel en el que el contacto ocurre a lo largo de una superficie común a los dos ejes. Existen seis tipos de pares cinemáticos inferiores diferentes, pero todos pueden producirse de dos tipos básicos, el par rotacional, denotado por **R** y también llamado revoluta, y el par deslizable, representado por **P** y también llamado prismático. La superficie común de contacto del par revoluta es un cilindro circular. Así dos cuerpos rígidos acoplados por una revoluta pueden girar relativamente uno sobre el otro y sobre el eje del cilindro común, al cual se le conoce como eje de la revoluta.

Un manipulador, en general, es un sistema mecánico enfocado a la manipulación de objetos. El análisis de desplazamiento de manipuladores robóticos de tipo serial se conoce como geometría de robots seriales. Un manipulador se distingue por la capacidad de ser programado y de realizar tareas de forma repetitiva. Se han aprovechado las capacidades de los robots industriales debido a la posibilidad de programar trayectorias complejas (Ángeles, 2007).

2.5. La notación Denavit-Hartenberg

Para realizar el control del robot es necesario encontrar el modelo geométrico del manipulador robótico, a fin de proporcionar instrucciones de manipulación sin considerar la dinámica de la carga manipulada y del robot mismo.

La nomenclatura de Denavit-Hartenberg (D-H) se emplea para describir de manera única la arquitectura de la cadena cinemática; es decir, definir la ubicación relativa y orientación de los ejes de pares vecinos.

A continuación se resumen los pasos necesarios para obtener dicha nomenclatura.

Paso 1: Localizar y etiquetar los ejes de unión como z_0, \dots, z_{n-1} .

Paso 2: Establecer el marco base, fijando el origen en cualquier parte del eje z_0 . Los ejes x_0 y y_0 se escogen convenientemente de acuerdo a la regla de la mano derecha.

Para fijar los orígenes $i = 1, \dots, n - 1$, se deben realizar los pasos del 3 al 5.

Paso 3: Localizar el origen o_i donde la recta normal a z_i y z_{i-1} interseca a z_i . Si z_i interseca z_{i-1} marcar esta intersección como o_i . Si z_i y z_{i-1} son paralelos, se debe localizar a o_i en cualquier posición conveniente a lo largo de z_i .

Paso 4: Establecer x_i a lo largo de la recta normal formada entre z_{i-1} y z_i sobre o_i , o en la dirección normal al plano formado por z_{i-1} a z_i si z_{i-1} y z_i intersectan.

Paso 5: Establecer y_i basándose en la regla de la mano derecha.

Paso 6: Establecer el marco del órgano terminal $o_n x_n y_n z_n$. Asumiendo que la unión n – $esima$ es revoluta, fijar $z_n = a$ a lo largo de la dirección z_{n-1} . Establecer el origen o_n de forma convenientemente a lo largo de z_n , preferentemente en el centro de la pinza o en la punta de la herramienta. Fijar $y_n = b$ en dirección de la pinza cerrada y fijar $x_n = n$ como $b \times a$. Si la herramienta no tiene la forma de una pinza común fijar x_n y y_n convenientemente para formar un maraco en base a la regla de la mano derecha.

Paso 7: Crear una tabla para los parámetros a_i , b_i , α_i y θ_i de cada unión.

a_i es la distancia a lo largo de x_i de o_i a la intersección de los ejes x_i y z_i .

b_i es la distancia a lo largo de z_{i-1} de o_{i-1} a la intersección de los ejes x_i y z_{i-1} . b_i es variable si la unión i es un par prismático.

α_i es el ángulo entre z_{i-1} y z_i medido sobre x_i como se observa en la Figura 2.2a.

θ_i es el ángulo entre x_{i-1} y x_i medido sobre z_{i-1} , mostrado en la Figura 2.2b. θ_i es variable si la unión i es revoluta (Spong, 2004).

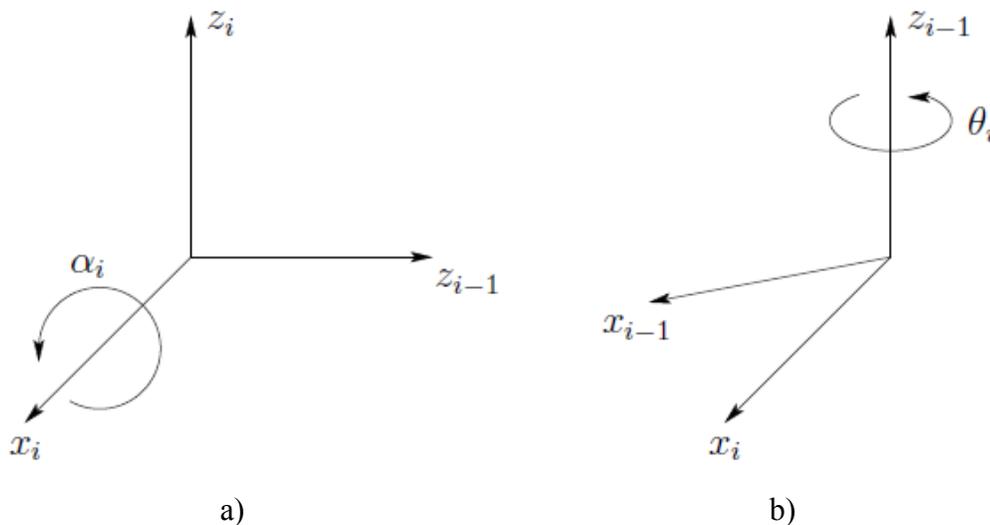


FIGURA 2.2. Signo de sentido de giro a) α_i y b) θ_i .

Una vez obtenidos los parámetros de (D-H) se procede a obtener las matrices de rotación para cada eje, primero se generaliza la representación de la matriz de rotación \mathbf{R}_{i+1}^i , que acarrea el marco \mathcal{F}_i a \mathcal{F}_{i+1} a partir de la ecuación (3) (Ángeles, 2007).

$$\mathbf{R}_{i+1}^i = \mathbf{C}_{i+1}^i \mathbf{\Lambda}_{i+1}^i \quad (3)$$

donde \mathbf{C}_i y $\mathbf{\Lambda}_i$ se representan por las matrices

$$\mathbf{C}_i = \begin{bmatrix} \cos\theta_i & -\text{sen}\theta_i & 0 \\ \text{sen}\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{\Lambda}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \lambda_i & -\mu_i \\ 0 & \mu_i & \lambda_i \end{bmatrix} \quad (4)$$

y donde λ y μ son

$$\lambda_i = \cos \alpha_i, \quad \mu_i = \text{sen}\alpha_i \quad (5)$$

Por lo tanto una rotación se define como

$$\mathbf{R}_i^{i-1} = \begin{pmatrix} \cos\theta_i & -\lambda_i \text{sen}\theta_i & \mu_i \text{sen}\theta_i \\ \text{sen}\theta_i & \lambda_i \cos\theta_i & \mu_i \cos\theta_i \\ 0 & \mu_i & \lambda_i \end{pmatrix} \quad (6)$$

2.6. Cinemática directa

El problema de la cinemática directa, se reduce al calcular la posición P y orientación \mathbf{R} que adopta el manipulador dado los valores de sus variables de unión, es decir

$$\mathbf{R} = \mathbf{R}_1^0 \mathbf{R}_2^1 \mathbf{R}_3^2 \mathbf{R}_4^3 \mathbf{R}_5^4 \mathbf{R}_6^5 \quad (7)$$

$$P = \mathbf{a}_1 \mathbf{R}_1^0 + \mathbf{a}_2 \mathbf{R}_2^0 + \mathbf{a}_3 \mathbf{R}_3^0 + \mathbf{a}_4 \mathbf{R}_4^0 + \mathbf{a}_5 \mathbf{R}_5^0 + \mathbf{a}_6 \mathbf{R}_6^0$$

Donde el vector \mathbf{a}_1 se define como el vector que va de un origen o_0 a o_1 dentro del manipulador

$$\mathbf{a}_i = \begin{bmatrix} a_i \cos \theta_i \\ a_i \sin \theta_i \\ b_i \end{bmatrix} \quad (8)$$

El problema de la cinemática directa no es muy común en el manejo de los manipuladores. Generalmente, el usuario fija las coordenadas del punto de operación del manipulador y no los valores de sus variables de unión (Ángeles, 2007).

2.7. Cinemática inversa

Para el seguimiento de una trayectoria es necesario conocer los ángulos de rotación de cada eslabón del manipulador para que el órgano terminal se encuentre en la posición deseada. Una manera de obtener esas variables es mediante la cinemática inversa. Los cálculos para la cinemática inversa se vuelven la parte más compleja en manipuladores de múltiples grados de libertad. Sin embargo, cuando se intersecan los últimos tres ejes de rotación en un punto es posible desacoplar la cinemática en dos problemas simples, cinemática inversa de posición y cinemática inversa de orientación. El punto de intersección de los tres ejes de rotación recibe el nombre de centro de la muñeca (*wrist center*), C (Ángeles, 2007). Lo importante de este punto es que el movimiento de estos últimos tres ejes no afecta la posición de C ; de este modo, la posición del centro de la muñeca esférica sólo depende de los tres primeros ejes. El origen del marco de coordenadas propuesto es obtenido simplemente por la traslación b_6 a lo largo de z_5 , es decir:

$$C = P - \mathbf{R}b_6 \quad (9)$$

donde \mathbf{R} es una matriz de rotación que define la orientación con la que el órgano terminal llega al punto propuesto P . Como sólo se tienen tres ejes para manipular, \mathbf{R} se fija por el usuario y es previamente calculada por los ángulos de Euler mediante:

$$\mathbf{R} = \begin{pmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi s_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta c_\psi + c_\phi s_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{pmatrix} \quad (10)$$

donde s_θ se usa para denotar $\sin \theta$.

2.8. El problema de la Posición

Para el cálculo del problema de la posición, se debe identificar la estructura del mismo, esto se hace mediante los parámetros de D-H. Éstos se pueden obtener a partir del esquema mostrado en la Figura 2.3. En la Tabla 1 se muestran los parámetros obtenidos para el manipulador ROMAT 56.

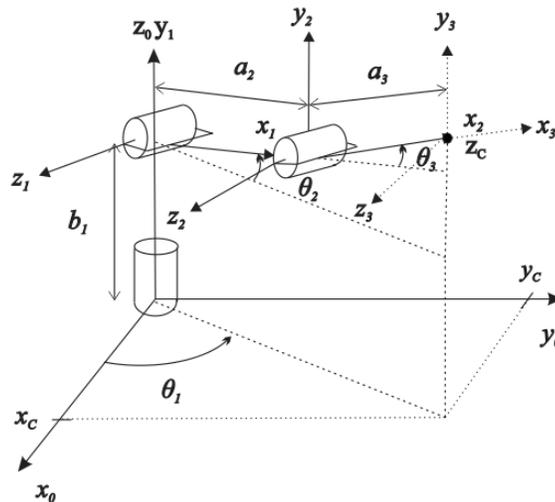


FIGURA 2.3. Diagrama esquemático de los primeros 3 ejes.

TABLA 1. Parámetros D-H del robot ROMAT de los ejes 1, 2 y 3.

Unión	a_i	b_i	α_i	θ_i	Cuentas eje
1	0	0.9	90	θ_1^*	20000
2	0.45	0	0	θ_2^*	385000
3	0.54	0	0	θ_3^*	295000

Para simplificar el cálculo de los tres primeros ángulos que definen la posición, se puede utilizar un análisis geométrico, debido a que la mayoría de los manipuladores son estructuralmente simples; además de que existen pocas técnicas que pueden manejar el problema de la cinemática inversa para configuraciones arbitrarias.

En base a un análisis para la Figura 2.3 se observa que el eslabón dos y tres proyectan una sombra sobre el plano XY, lo que permite inferir que éstos son dependientes de θ_1 , que queda en función de x_c e y_c , es decir:

$$\theta_1 = \text{atan}\left(\frac{y_c}{x_c}\right) \quad (11)$$

En cuanto a los otros dos ángulos, se utiliza la ley de cosenos y una identidad de ángulos suplementarios para determinar:

$$\cos\theta_3 = \frac{x_c^2 + y_c^2 + (z_c - b_1)^2 - a_2^2 - a_3^2}{2a_2a_3} := D \quad (12)$$

por lo tanto, θ_3 se obtiene a partir de (12):

$$\theta_3 = \text{atan}\left(\frac{\pm\sqrt{1 - D^2}}{D}\right) \quad (13)$$

θ_2 se obtiene a partir de la resta del ángulo formado entre C y la proyección de C , usando la intersección de z_0 y z_1 como pivote, menos el ángulo interno (Spong, 2004):

$$\theta_2 = \text{atan}\left(\frac{z_c + b_1}{\sqrt{x^2 + y^2}}\right) - \text{atan}\left(\frac{a_3 \sin(\theta_3)}{a_2 + a_3 \cos(\theta_3)}\right) \quad (14)$$

2.9. El problema de la Orientación

Una vez que se tienen los primeros tres ángulos, es posible determinar la orientación del marco de referencia en el punto C . Sabiendo que la rotación \mathbf{R} se puede obtener como el producto de las rotaciones del marco \mathcal{F}_0 al \mathcal{F}_3 y del marco \mathcal{F}_3 al \mathcal{F}_6 de cada eje, mediante:

$$\mathbf{R} = \mathbf{R}_3^0 \mathbf{R}_6^3 \quad (15)$$

De este modo es posible obtener los tres ángulos, de acuerdo a:

$$\mathbf{R}_6^3 = (\mathbf{R}_3^0)^{-1} \mathbf{R} = (\mathbf{R}_3^0)^T \mathbf{R} \quad (16)$$

$$\mathbf{R}_6^3 = \begin{pmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - s_4 c_6 & c_4 s_5 \\ s_4 c_5 c_6 + c_4 s_6 & -s_4 c_5 s_6 + c_4 c_6 & s_4 s_5 \\ -s_5 c_6 & s_5 s_6 & c_5 \end{pmatrix} \quad (18)$$

Al observar cuidadosamente (18), es posible obtener los ángulos referentes de la muñeca esférica separando algunos elementos. Comenzando por la columna 3, se tiene que:

Para θ_4

$$\frac{\text{sen}\theta_4 \text{sen}\theta_5}{\text{cos}\theta_4 \text{sen}\theta_5} = \frac{r_{23}}{r_{13}} \therefore \theta_4 = \text{Atan}\left(\frac{r_{23}}{r_{13}}\right) \quad (19)$$

Para θ_5

$$\theta_5 = \text{Atan}\left(\frac{\pm\sqrt{1 - r_{33}^2}}{r_{33}^2}\right) \quad (20)$$

mientras que para θ_6 (Spong, 2004)

$$\frac{-\text{sen}\theta_6 \text{sen}\theta_5}{\text{cos}\theta_6 \text{sen}\theta_5} = \frac{r_{32}}{r_{31}} \therefore \theta_6 = \text{Atan}\left(-\frac{r_{32}}{r_{31}}\right) \quad (21)$$

2.10. Controlador PID

El controlador PID ha sido extensamente usado desde su invención en 1910. Cabe mencionar que la mayoría de los controladores industriales que se usan hoy día utilizan esquemas PID o PID modificado. La utilidad de los controles PID se observa en el hecho de que se aplican casi en forma general a la mayoría de los sistemas de control (Ogata, 2010). Existen muchas implementaciones de control pero la más común es la que se muestra en la Figura 2.5b, la cual cuenta con una entrada y salida simple, y un lazo de retroalimentación.

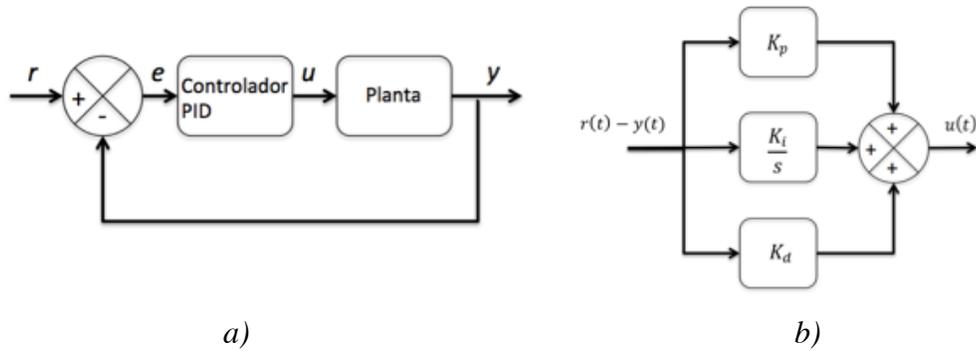


FIGURA 2.5. a) Diagrama simplificado de un sistema en lazo cerrado y b) Diagrama a bloques de la estructura típica de un controlador PID.

La estructura típica del control PID se muestra en la Figura 2.5a, donde la señal de error $e(t)$, entra al bloque de control y el resultado de la señal de excitación es la suma de la señal de error por la acción proporcional, integral y derivativa.

La salida de la señal de excitación sobre el controlador PID, donde $e(t) = r(t) - y(t)$, es

$$u(t) = Pe(t) + I \int e(t)dt + D \frac{d}{dt} e(t) \quad (22)$$

También, la ecuación (22) puede escribirse en su forma de Laplace como

$$u(s) = e(s) \left(P + \frac{I}{s} + Ds \right) \quad (23)$$

La función de transferencia puede ser reescrita de la siguiente manera.

$$C(s) = \frac{u(s)}{e(s)} = P + \frac{I}{s} + Ds \quad (24)$$

La ecuación (24) puede ser acomodada en su forma recursiva como una ecuación de diferencias para un filtro IIR (Infinite Impulse Response) de segundo orden

$$y(k) = a_0 u(k) + a_1 u(k-1) + a_2 u(k-2) - b_1 y(k-1) - b_2 y(k-2) \quad (25)$$

Su correspondiente función de transferencia discreta es

$$C(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2}} \quad (26)$$

Para obtener los valores constantes de la ecuación (26) se requiere discretizar sus constantes, fijando el periodo de muestreo T_s y utilizando la Tabla 3 que define las relaciones entre las constantes discretas.

TABLA 3. Relación para discretizar los valores P, I y D.

$C(z)$	T_i	T_d	K_p	K_i	K_d
$C(s)$	$\frac{P}{I}$	$\frac{D}{P}$	$P - \frac{P T_s}{2 T_i}$	$\frac{P T_s}{T_i}$	$\frac{P T_d}{T_s}$

Una vez que han sido discretizadas las constantes, se obtienen sus equivalencias para ser implementadas dentro del controlador digital utilizando la Tabla 4 como guía.

TABLA 4. Equivalencia de los parámetros para el controlador PID discreto.

a_0	a_1	a_2	b_1	b_2
$K_p + K_i + K_d$	$-(K_p + 2K_d)$	K_d	1	0

CAPÍTULO III

METODOLOGÍA

Con el fin de lograr el objetivo propuesto se desarrolló una serie de pasos a seguir tal y como se muestra en la Figura 3.1, el trabajo consiste de tres etapas principales: en la primera etapa se busca la documentación del tema referente a la obtención de la cinemática inversa del manipulador a utilizar, visto en el capítulo anterior y la revisión sobre la conexión del dispositivo háptico para Simulink y C++.

Para la segunda etapa o etapa de simulación, se plantea la implementación del sistema general en Simulink; posteriormente, se planteó el diseño de un algoritmo de seguridad y se diseñó el manipulador virtual utilizando la herramienta VRBuild. Este conjunto se implementó en Simulink para finalmente mostrar la simulación completa. La simulación ayudará a visualizar el sistema para pruebas preliminares, además en esta etapa se implementa el sistema háptico. En la tercera etapa se realiza la implementación y comienza con obtener las ecuaciones de cinemática inversa en su forma más simple para que sean implementadas en C++, después se añaden módulos de: lectura y almacenaje de datos, envío y recepción por el protocolo USB, conversión de valores angulares en cuentas. La búsqueda del modelo matemático de la planta mediante la implementación de mínimos cuadrados recursivo (RLS) diseñada por Morales-Velazquez et al., (2010), se utiliza para realizar la sintonización de un controlador PID por el método de margen de fase. Además, se realiza la implementación del sistema digital al cual se añaden los módulos de comunicación USB, el driver de la SRAM, el driver de encoder, el driver del DAC, la máquina para el control del sistema y los controladores PID. Finalmente, una vez que todo funciona correctamente se procede a implementar el sistema para realizar pruebas con algunas trayectorias en el robot y utilizar el mando háptico para fijar las referencias de forma continua.

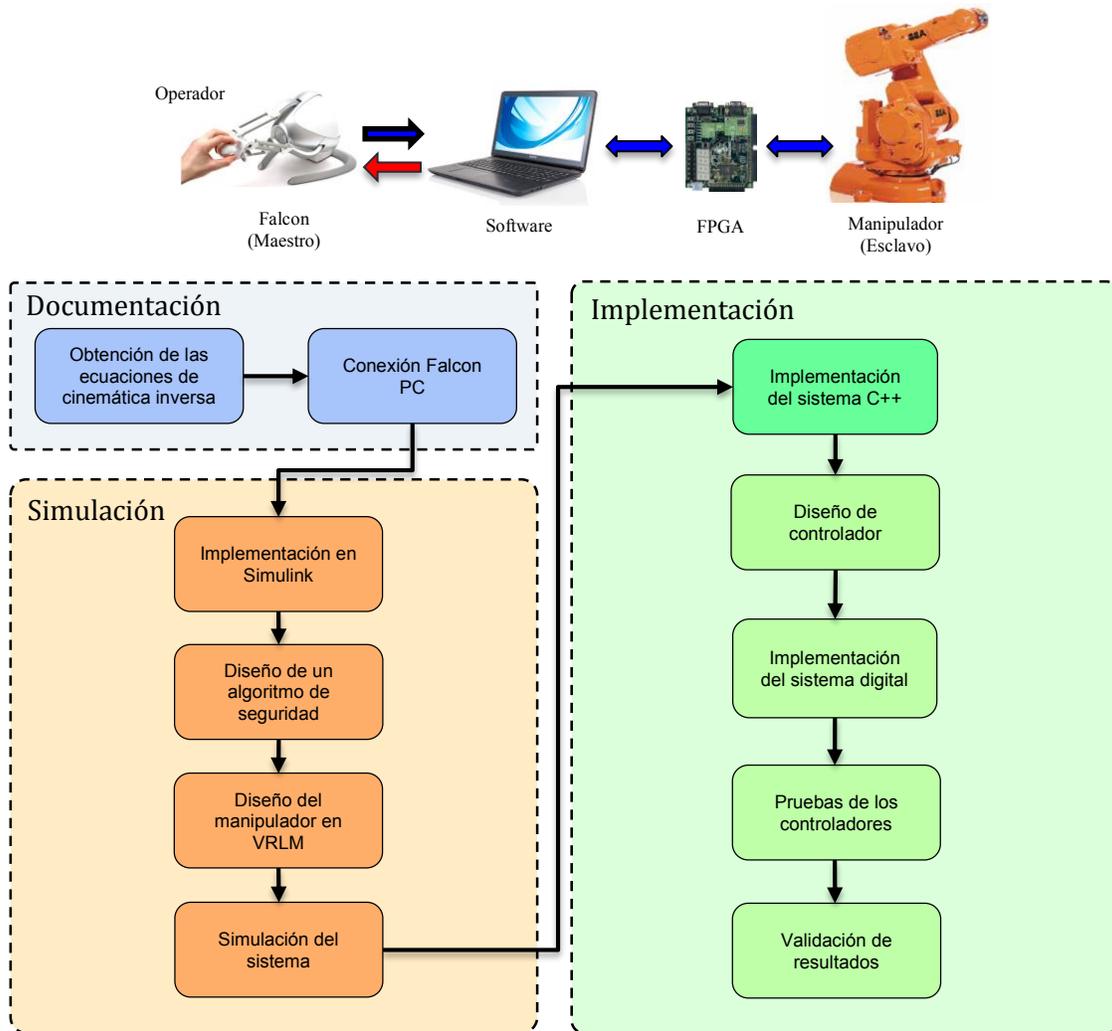


FIGURA 3.1. Metodología propuesta.

3.1. Documentación

Esta parte es el punto de partida de cualquier trabajo, consiste en la búsqueda de información referente al tema a tratar, en este apartado se hace referencia a la obtención de las ecuaciones de cinemática inversa visto en los apartados del 2.5 al 2.9, también se hace la investigación para la conexión del Falcon.

3.1.1. Conexión entre Falcon y PC

Se hace hincapié en establecer la comunicación del Falcon con la computadora, ya que a pesar que el fabricante provee las herramientas de conexión para desarrolladores, éstas se

crearon para versiones viejas de software comercial. Debido a esto, se buscaron herramientas alternas que cumplieran con la misma función, la organización SIRLab diseñó las librerías para Matlab *haptik library* las cuales brindan dll's, bloques de Simulink y ejemplos para su implementación, mientras que para una implementación en C++, se utilizan librerías libnifalcon que son de acceso libre.

3.2. Simulación

En esta parte, se propone la metodología para diseñar el modelo virtual que ayudará en la realización de pruebas previas a la implementación con el fin de dar una idea general de los movimientos del manipulador real. En esta etapa se tiene la implementación general de Simulink donde se hace la conexión del Falcon, un tratamiento a las referencias, la obtención de la cinemática inversa, posteriormente el diseño de un algoritmo de seguridad y el diseño del manipulador virtual.

3.2.1. Implementación en Simulink

En este módulo se agrupa toda la implementación de la simulación como se observa en la Figura 3.2.

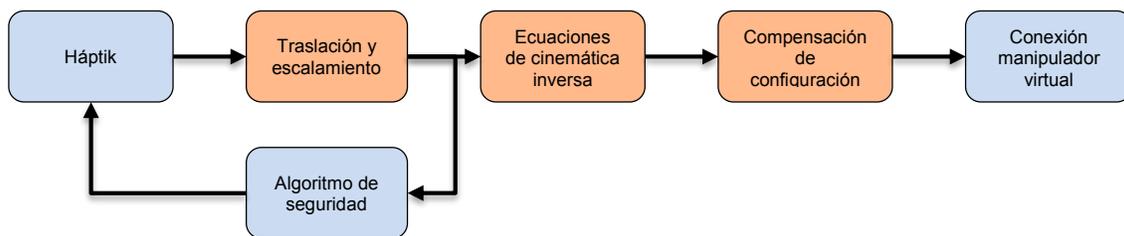


FIGURA 3.2. Implementación en Simulink.

En la implementación se integra el módulo *haptik*, que fija las referencias en coordenadas rectangulares, una traslación y escalamiento son requeridos para posicionar y dar alcance a los movimientos del háptico, se retroalimenta fuerza por medio de un algoritmo de seguridad, se obtienen las referencias angulares a partir de las ecuaciones de cinemática inversa, después se hace una compensación a los ejes del manipulador y finalmente se manda llamar el diseño del manipulador.

3.2.1.1. Traslación y Escalamiento

Debido a que se tiene un dispositivo háptico cuyo espacio de trabajo es muy pequeño con respecto al del manipulador, se requieren ciertas consideraciones. Es necesario tener en cuenta que el manipulador comenzará en un punto denominado *Home*, de ahí debe seguir una trayectoria suave hasta una región donde se hará el trabajo con el robot, teniendo en cuenta no pasar por zonas de singularidad como son el eje Z_0 o salir del espacio de trabajo del manipulador. Ya que el movimiento del Falcon es casi inapreciable se debe escoger un factor de escala que magnifique el movimiento, sin embargo esto causa que se pierda resolución en los movimientos, así que se debe buscar un equilibrio entre alcance y resolución que satisfaga la tarea que se desea realizar. La implementación de la traslación y el escalamiento se muestran en la Figura 3.3 a) y Figura 3.3 b) respectivamente.

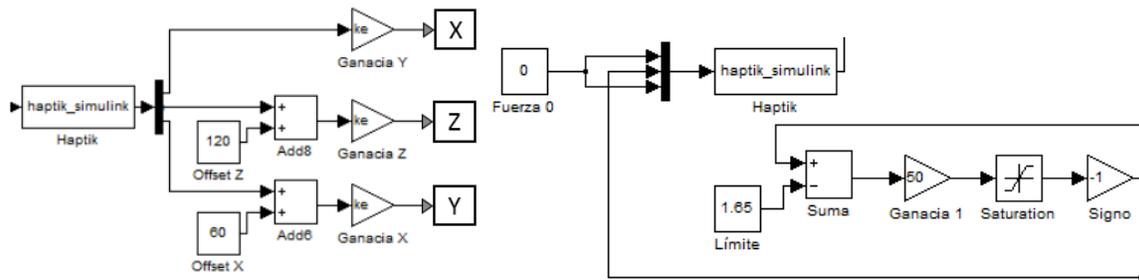


FIGURA 3.3. a) Escalamiento de valores b) Implementación de la detección de objetos.

3.2.1.2. Inclusión de las ecuaciones de cinemática inversa

Simulink permite la inclusión de ecuaciones como si se trabajará directamente con Matlab, mediante los bloques de funciones embebidas, simplificando al usuario el cálculo de las matrices, de este modo es posible implementar las ecuaciones de cinemática inversa en forma de un módulo, al cual se fijan las coordenadas y la matriz de rotación, para obtener los ángulos de rotación, a este módulo se añaden seis constantes definidas como *ban1*, *ban2*... *ban6*, cuya función es determinar el signo en algunas de las raíces cuadradas que arrojan las ecuaciones obtenidas en los apartados 2.8 y 2.9, el ajuste de estas banderas cambia la configuración con la que el manipulador llega a un punto si es; izquierda con el brazo arriba, izquierda con el brazo abajo, derecha con el brazo arriba o derecha con el brazo abajo.

3.2.1.3. Compensación debido a la configuración

Este punto es importante debido a que se debe realizar una compensación de acuerdo a como se planteó la configuración inicial del manipulador con los parámetros de Denavit - Hartenberg, esto se logra añadiendo una constante al resultado de las ecuaciones de cinemática.

3.2.2. Diseño de un algoritmo de seguridad

Una de las principales ventajas de utilizar un dispositivo háptico es la posibilidad de implementar un algoritmo de seguridad. Se hace uso del concepto de pared virtual el cual consiste en un límite que el usuario fija, y que se utiliza para restringir el movimiento del manipulador dentro de su espacio de trabajo. Se implementa haciendo uso de la posición actual (P) y el límite fijado (lim), mientras la posición actual no rebase el límite, la fuerza (F) de respuesta al movimiento es cero, sin embargo, si el límite es excedido, se genera una fuerza en oposición en función del error de posición magnificado con una ganancia (K). Una manera clara de observar el funcionamiento se muestra en la Figura 3.4.

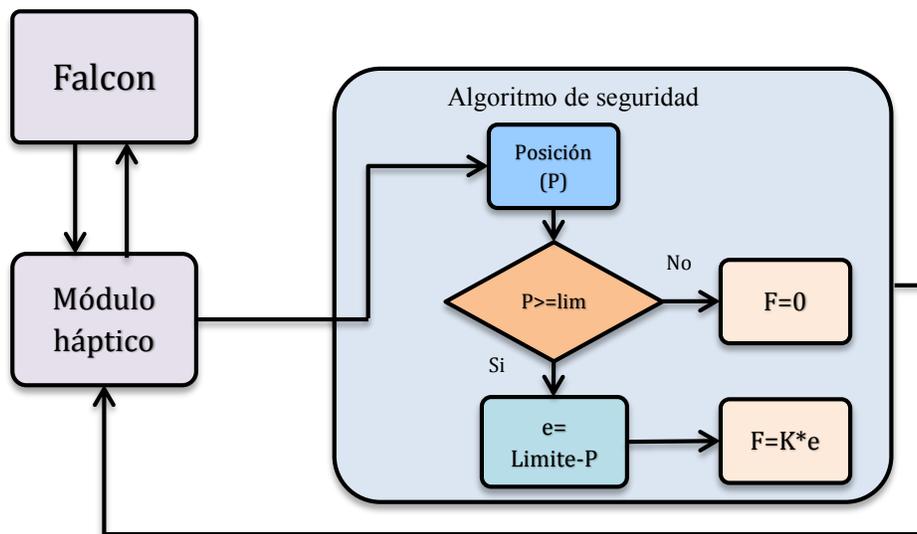


FIGURA 3.4. Diseño de medida de seguridad.

Se observa como la ecuación (27) está relacionada con la ecuación del resorte, a su vez se observa que entre más se aleja el mando, la fuerza opositora sigue incrementándose.

$$F = K * (Limite - P) \quad (27)$$

Al igual que se puede implementar el efecto de un resorte en el háptico, es posible simular la acción que ejercería un amortiguador, cuya principal ventaja es el impedimento de movimientos abruptos hechos por el usuario. Por razones de practicidad solo se simuló la ecuación (27).

3.2.3. Diseño del modelo virtual

Las simulaciones son de gran ayuda si se desea observar un sistema complejo, de esta manera se pueden evitar accidentes y dar una mejor perspectiva del funcionamiento del sistema.

Utilizando la herramienta V-Realm Builder de Matlab, se puede diseñar elemento por elemento un manipulador virtual, creando un diseño estilizado que lo asemejan al manipulador real gracias a sus múltiples herramientas. Como se muestra en la Figura 3.5.

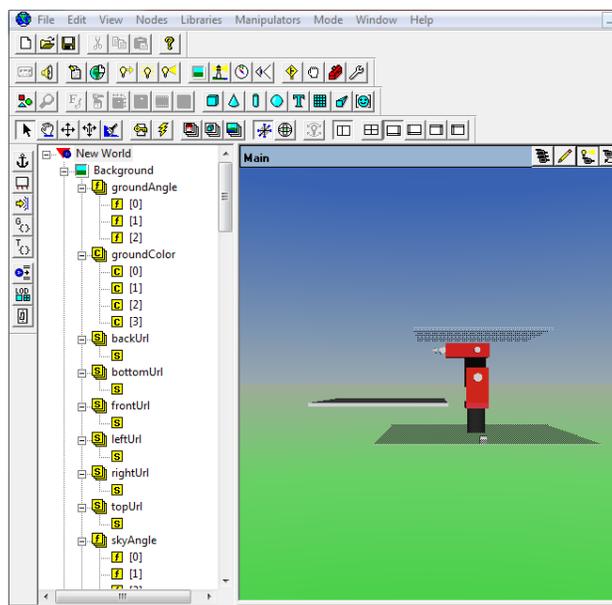


FIGURA 3.5. Diseño del manipulador virtual en V-Realm Builder.

Cabe mencionar que cuando se crea un eslabón E1 sobre un eslabón E2, E2 hereda las coordenadas de origen de E1, de este modo se considera automáticamente el movimiento E1 que ejerce sobre E2, como un sistema acoplado.

3.3. Implementación

Una vez que se han realizado las pruebas de simulación se procede a plantear la implementación del sistema, en este apartado se hablará del desarrollo de un software en C++, que establece la conexión con el Falcon y ayuda en el tratamiento de las referencias en bruto, se establece el diseño del controlador, se realiza la implementación del sistema digital, posteriormente se prueban con los controladores y se validan los resultados, estos dos últimos se tratarán en el capítulo IV.

3.3.1. Implementación en C++

Dado que este sistema se planteó inicialmente para ser implementado con un robot industrial, se diseñó un software en C++, que permitiera la conexión simultánea del dispositivo háptico y el FPGA para brindar mayor velocidad de respuesta a las lecturas, que utilizando Matlab. Para esta implementación se incluyó: la conexión entre Falcon-PC, la función para resolver la cinemática inversa en su forma escalar, la conexión entre FPGA-PC por medio del puerto USB y un fichero para almacenar los datos. El funcionamiento del sistema en C++ puede verse a grandes rasgos en el diagrama a bloques Figura 3.6.

Al igual que en la simulación, el robot comienza en un *Home* debido a eso se implementa una lectura de datos almacenados, los cuales son enviados como referencia al FPGA aproximadamente cada milisegundo, una vez que el manipulador se encuentra en el lugar de trabajo, se procede a realizar la lectura de los datos provenientes del Falcon cuyo valor se actualiza aproximadamente cada milisegundo, como se verá en el apartado 4.2 no siempre trabaja de la misma forma. Los datos de las coordenadas del Falcon pasan a la función de cinemática inversa que en conjunto con los ángulos de Euler que son fijados como constantes, ayudan a obtener los ángulos referencia para los seis ejes. Las conversiones de radianes a cuentas (basadas en las tablas 1 y 2) se almacenan y posteriormente son enviadas a través del USB hacia la tarjeta FPGA, el proceso se repite hasta que la memoria SRAM de la tarjeta se encuentre llena. Finalmente se implementó en C++ la lectura del USB y el almacenaje de los datos en un archivo .txt, que posteriormente será leído en Matlab para verificar el funcionamiento del controlador.

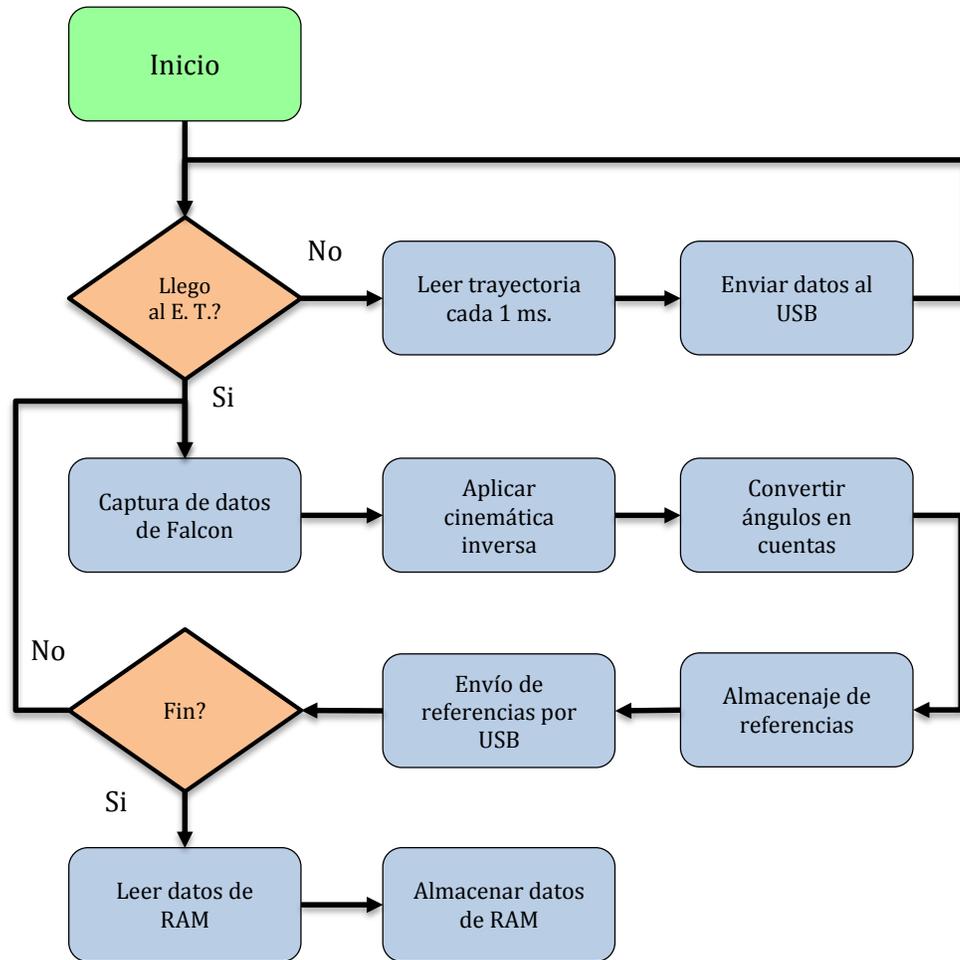


FIGURA 3.6. Implementación del sistema en C++.

3.3.2. Diseño del controlador

Se plantea el diseño del controlador, empezando con el modelado del sistema el cual ayuda a dar una idea de las propiedades de la planta, posteriormente utilizando esa misma planta se obtiene un controlador PID sintonizado.

3.3.2.1. Modelado del sistema

Para implementar un buen controlador es necesario conocer las propiedades del sistema que se desea controlar. Se puede utilizar un método analítico en el caso que se conocieran todas las propiedades del sistema, en este caso como no se conocen, se usa un método experimental. Se llegó a la conclusión de utilizar el algoritmo de mínimos cuadrados debido a que es un

método que se aplica a cualquier sistema lineal y que arroja buenos resultados, este algoritmo trabaja con un vector de entrada y un vector de salida, para obtener un resultado óptimo se deben proponer valores de entrada aleatorios generalmente ruido gaussiano, a partir de este procesamiento se obtiene la función de transferencia discreta. La implementación del algoritmo se observa en la Figura 3.7. tomado de Morales-Velázquez (2010), en la implementación, el modulo físico USB recibe las referencias de la (PC), después se insertan al controlador (PID) a través del bloque (USB) y se realiza el control a esas referencias, mientras el procesador (RLS) trabaja como un observador para almacenar y procesar los datos (Aguado, 2000).

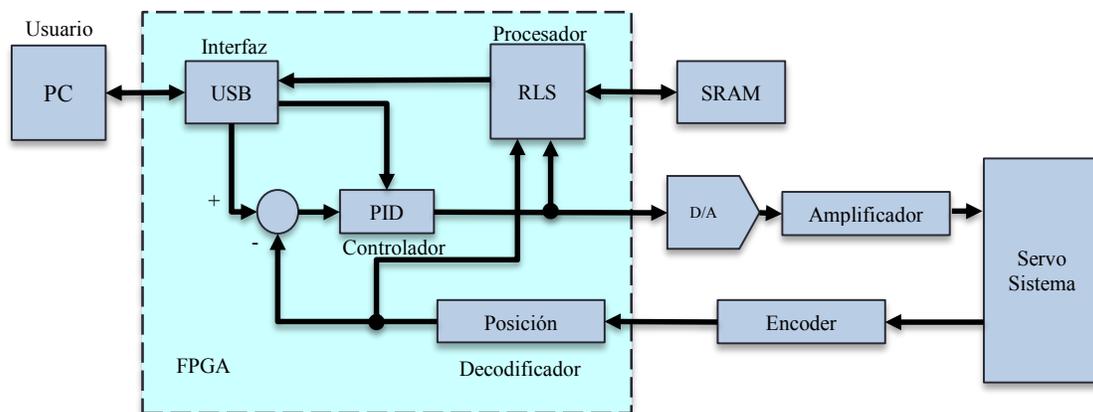


FIGURA 3.7. Implementación del algoritmo de mínimos cuadrados recursivos.

Para las pruebas se utilizó la tarjeta de diseño propio MCUAQ4X y el software de la tarjeta MCUAQ4X, en cada prueba se hacía la identificación eje por eje.

3.3.2.2. Sintonización del controlador

Utilizando la planta que se obtuvo con el algoritmo de mínimos cuadrados se planteó el tipo de control que se debe utilizar, por la facilidad de su implementación y su efectividad se seleccionó el PID, el cual puede ser sintonizado por margen de fase, Ziegler-Nichols o lugar de las raíces.

Al igual que para la identificación de la planta, para la sintonía del controlador, se utilizó la tarjeta MCUAQ4X junto con el software de esta tarjeta, en este software queda almacenada

la planta obtenida en el paso anterior, debido a que utiliza margen de fase para la sintonía, se requiere fijar la ganancia del controlador normalmente 1, la fase del controlador normalmente $\frac{\pi}{4}$ y la frecuencia de respuesta del sistema, para este caso se utilizó 250 Hz. Al fijar estas variables, el software arroja las constantes del controlador P, I y D. Un controlador PID se expresa en su forma de ecuación de diferencias (25), o su forma discreta aplicando la transformada bilineal, a su vez es necesario discretizar las constantes utilizando la Tabla 3. Los coeficientes discretizados se ajustan utilizando la Tabla 4 para que sean implementados en la estructura del controlador mostrado en la Figura 3.8, (Morales-Velázquez 2010).

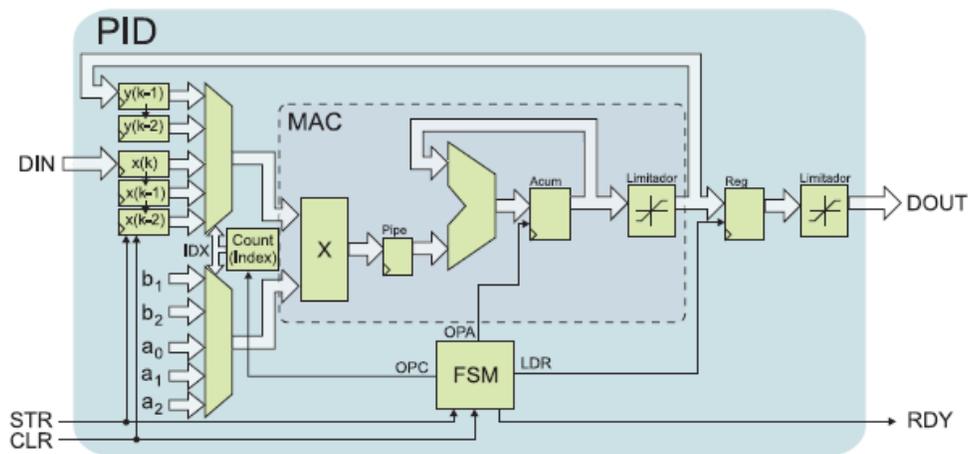


FIGURA 3.8. Implementación del controlador

Cabe mencionar que no siempre se obtiene un buen modelo de la planta debido a no linealidades del sistema o a que la señal de entrada no tiene una amplitud óptima, para estos casos se busca una sintonización de manera heurística tomando como punto de partida el controlador que arroja el software (apartado 4.3).

3.3.3. Implementación del sistema digital

El sistema digital propuesto está conformado por un temporizador de un milisegundo que gobierna la mayoría del sistema, una etapa de recepción y transmisión de datos, una máquina de estados para controlar el sistema, una etapa para el controlador y un módulo para la escritura y lectura de la memoria (Figura 3.9).

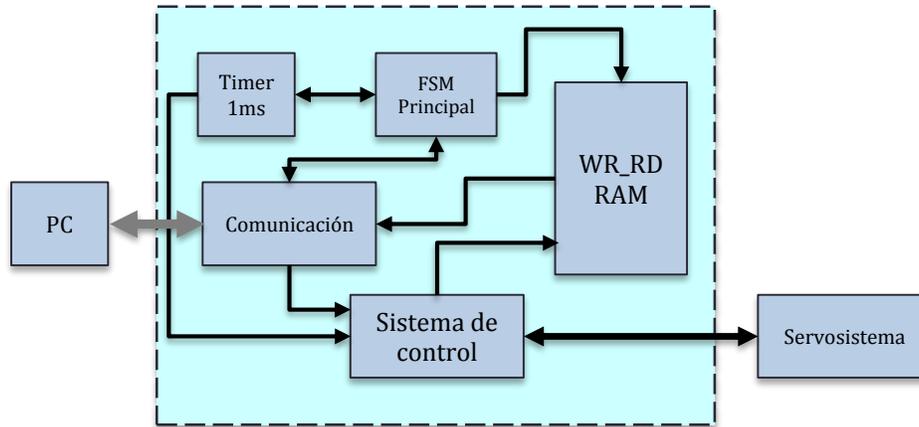


FIGURA 3.9. Implementación del sistema digital.

3.3.3.1. Máquina principal

La máquina principal (FSM_Principal) gobierna la mayoría del sistema, tiene la función de accionar el temporizador (Timer_1ms), liberar el puerto de comunicación (Comunicación), escribir y leer de la memoria (WR_RDRAM). Su funcionamiento empieza cuando recibe un dato del puerto, el cual es verificado para determinar si es una de las opciones predeterminadas denominadas Password *PWD*, de no ser así libera el puerto de comunicación para esperar otro dato. Cuando llega el *PWD* que indica la opción de escritura, se acciona el temporizador de un milisegundo, habilita la SRAM en modo de escritura $WR='1'$, inicia el funcionamiento del sistema de control. A partir de este punto lo único que hace la máquina es mantener los datos entrantes en un registro y liberar el puerto de comunicación, hasta que la memoria haya sido llenada. Cuando termina la escritura $EWR='1'$, se reinicia la máquina en espera de otro *PWD*, el cual puede ser seleccionado en modo de escritura nuevamente o lectura, si se selecciona lectura, se habilita la SRAM en modo de lectura, no se habilita el Timer_1ms y se acomodan los datos para ser enviados utilizando un contador y un multiplexor que ajustan los datos en tramas de 128 bits Figura 3.10 y otro contador (CNTRD) para recorrer la memoria como se muestra en la Figura 3.13, el ciclo continua hasta que se habilite la señal *ERD*.

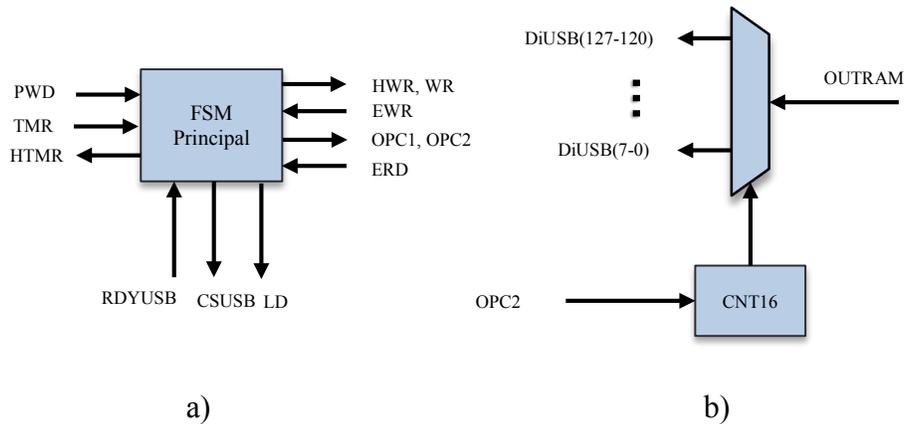


FIGURA 3.10. Implementación de a) la máquina de estados principal y b) agrupamiento de datos.

3.3.3.2. Módulo de comunicación

Este módulo está conformado por el driver del usb 5.0 que se utiliza para recibir y enviar datos, y un registro (REG) para mantener el dato proveniente de la computadora, la máquina de control gobierna ambos módulos Figura 3.11.

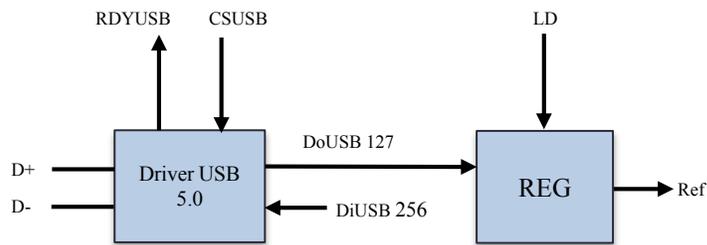


FIGURA 3.11. Módulo de comunicación USB.

3.3.3.3. Sistema de control

El sistema de control está constituido por el controlador PID (PID), driver de los encoders (Decodificador) y del driver del DAC (D/A) vistos en la Figura 3.12.

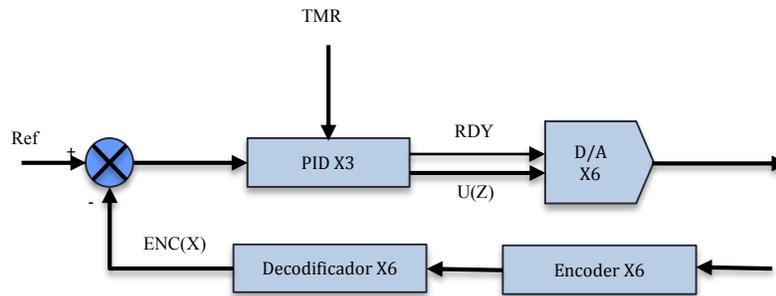


FIGURA 3.12. Esquema del sistema de control PID.

El sistema de control es un módulo que trabaja de manera indirecta a la máquina de control principal, ya que solo se acciona cuando (Timer_1ms) está activo, en otras palabras se acciona en la etapa de escritura y realiza sus funciones de control sin importar que pase en los otros módulos. Lo primero que realiza el controlador es el cálculo del error, una vez que termina de realizar el procesamiento, se acciona la señal de *RDY* la cual habilita el driver del DAC para mandar la señal de referencia hacia los servoamplificadores.

3.3.3.4. Módulo de escritura y lectura de memoria

En este módulo se implementa una máquina para controlar dos contadores, uno de ellos para acomodar los datos entrantes con un multiplexor mientras que el otro para direccionar la dirección de la SRAM, como se observa en la Figura 3.13.

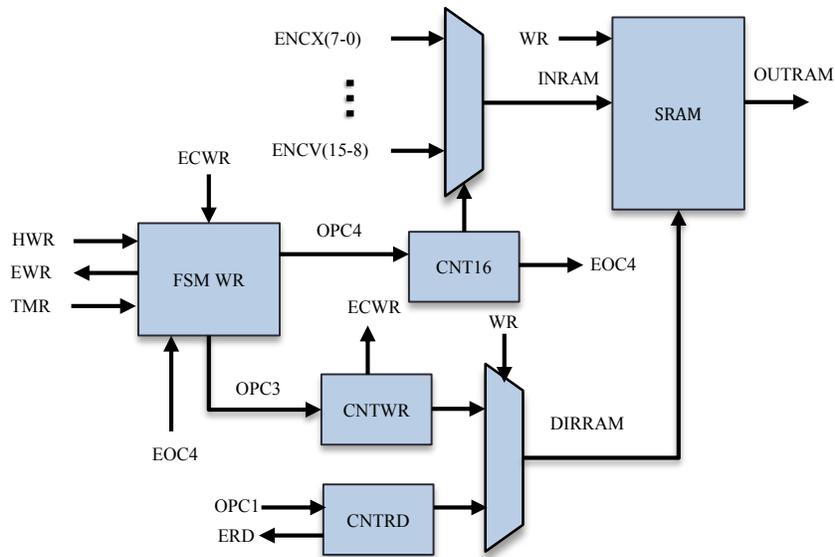


FIGURA 3.13. Módulo de escritura y lectura de RAM.

Este módulo se activa si la opción de escritura ha sido seleccionada, la maquina se habilita con la señal $HWR='1'$ y queda en espera de que se cumpla un periodo de muestreo $TMR='1'$, cuando ocurre se desencadenan dos contadores que incrementan al mismo tiempo para permitir el flujo de las lecturas de los seis encoders en tramas de 8 bits hasta que $EOC4='1'$, es decir 16 datos se han escrito y vuelve a su estado de espera hasta otro periodo de muestreo. Como se puede observar uno de los principales problemas en la memoria RAM es el bus de datos que es de tan sólo 8 bits, el multiplexor ayuda a que los datos sean acomodados de 8 en 8 bits, mientras el contador (CNTWR) va recorriendo la memoria, en este caso para que no exista conflicto en la lectura y en la escritura se utiliza un multiplexor que permite el paso del contador de escritura o del contador de lectura.

CAPÍTULO IV

PRUEBAS Y RESULTADOS

En este capítulo se presentan cuatro casos de estudio: el primero consiste en verificar el funcionamiento tanto de la conexión pc – dispositivo háptico como de las ecuaciones de cinemática inversa mediante una simulación, en esta parte también se implementa el efecto háptico; el segundo consiste en verificar el tiempo de trabajo del sistema para dar una idea de a que frecuencia se debe hacer la sintonización PID. Posteriormente en el tercer caso, se utiliza el controlador descrito en el capítulo de metodología, se realizan pruebas para el seguimiento de una trayectoria suave (un perfil). Finalmente en el cuarto caso, se implementa el sistema háptico completo para seguir trayectorias fijadas por el movimiento del FALCON.

4.1. Simulación

La primera prueba consiste en verificar tanto la correcta conexión entre el Falcon y la PC, verificar que las ecuaciones hayan sido resueltas correctamente y que el modelo virtual del manipulador se mueva acorde a lo esperado con una orientación fija y movimientos suaves (Figura 4.1). Del mismo modo, se propone hacer uso del efecto háptico al usuario, es decir, sí el manipulador encuentra un obstáculo se debe restringir el movimiento al operador.

Para esta prueba se utilizó las librerías *haptik*, Simulink bajo la versión de Matlab 7.1, una computadora con Windows 7, el dispositivo Falcon y un modelo creado en V-Realm builder.

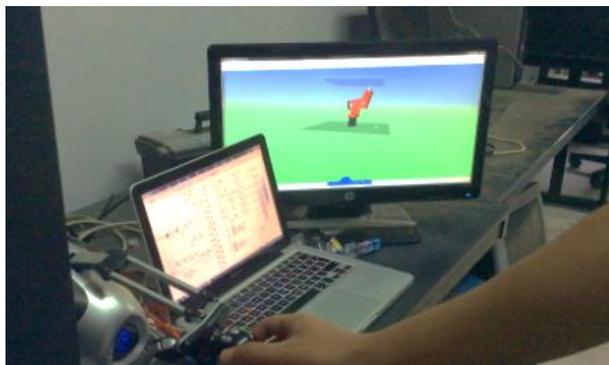


FIGURA 4.1. Sistema simulado.

Para esta prueba se hicieron movimientos en el espacio, siguiendo trayectorias lineales realizando movimientos de derecha a izquierda, de arriba hacia abajo y de adentro hacia fuera, como se muestra en la Figura 4.2 a).

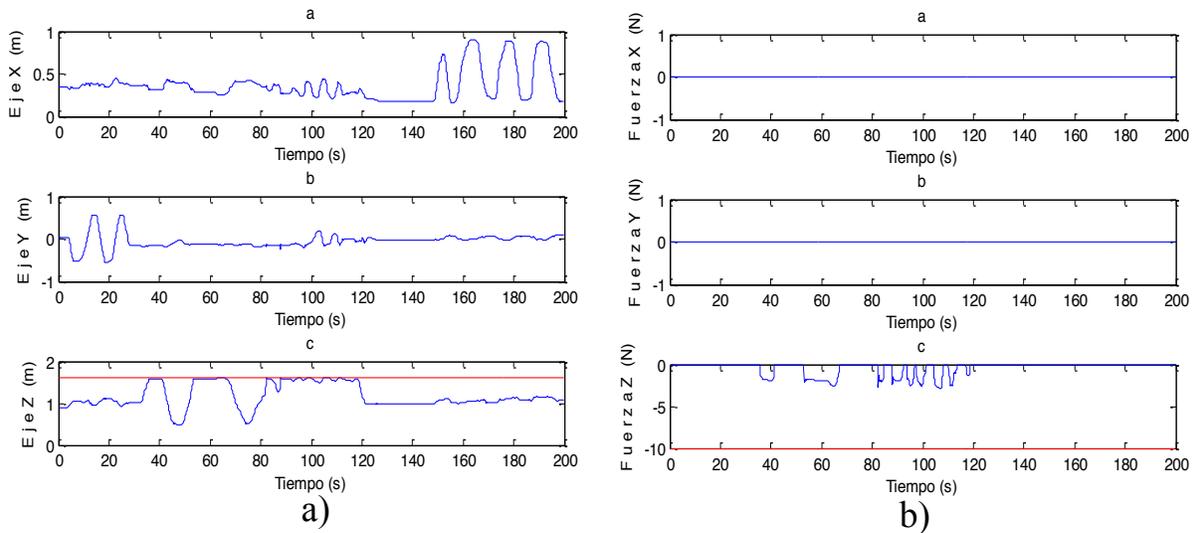


FIGURA 4.2. a) Lecturas de los ejes simulados, b) Fuerza ejercida por el Falcon.

En la prueba se implementó un muro virtual en la posición $z=1.6$ indicado con una línea roja en la figura 4.2a, c) y que es representado por un plano en la Figura 4.1., el efecto háptico comienza al intentar sobrepasar el límite propuesto manifestándose con una fuerza opositora, ilustrado en la Figura 4.2b.

Las gráficas en la Figura 4.2a, muestra la facilidad de moverse libremente a través del espacio, esto se complica cuando se lleva el mando al límite propuesto denotado por una línea en el eje Z. Comparando las Figuras 4.2a y 4.2b se observa como entre más se intenta sobrepasar el límite, la fuerza incrementa de manera proporcional. Debido a que el objetivo de esta prueba fue simular el efecto de un techo sólo se implementó un objeto para el eje Z. Con esta prueba se demostró que el Falcon es un mando fácil de operar pues los movimientos para crear líneas fueron suaves; además, se ajustaron a los movimientos naturales del operador y que el dispositivo háptico puede ser implementado como medida de seguridad al restringir la movilidad al espacio donde se trabaja, tal y como se demostró en la Figura 4.2a el órgano terminal no sobrepaso este límite.

4.2. Cálculo del periodo de muestreo

El objetivo de esta prueba consiste en dar una idea general del tiempo que toma el háptico en proporcionar un dato válido a la tarjeta ya que a pesar de que el fabricante asegure que el háptico provee un dato valido cada un milisegundo no siempre es así. Como se sabe el tiempo puede variar en función del uso de la computadora. Los tiempos que se consideran en esta prueba son: el tiempo que toma el háptico en tener un dato valido, el cálculo de la cinemática inversa, la conversión del valor angular a cuentas y el tiempo que tarda en ser enviado un dato a través del USB. La prueba se ilustra en la Figura 4.3.



FIGURA 4.3. Implementación para obtener el retardo.

Para lograr lo anterior, se implementó un contador programable con una resolución de 10 microsegundos, que se inicia cuando se tiene un dato valido en el puerto USB y se resetea cuando llega el siguiente valor, antes de limpiar la cuenta se almacena en la memoria SRAM de 16 bits, cuando se llena la memoria se inicia el envío de los datos a la PC.

Para esta prueba se utilizó una computadora con Windows 7 con puertos USB 2.0, el software de desarrolladores CodeBlocks, el Falcon y una tarjeta FPGA Spartan 3. El resultado de la prueba se observa a continuación Figura 4.4.

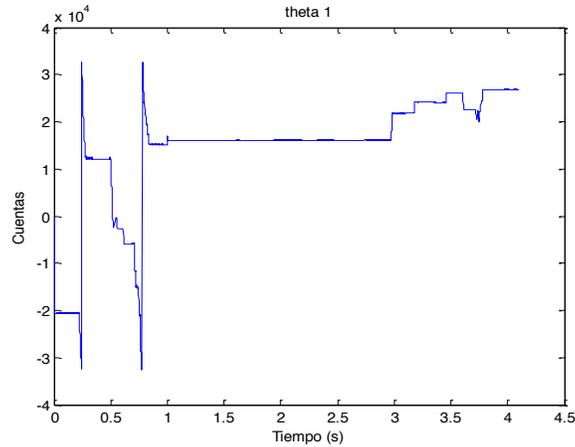


FIGURA 4.4. Cuentas calculadas y enviadas hacia el FPGA.

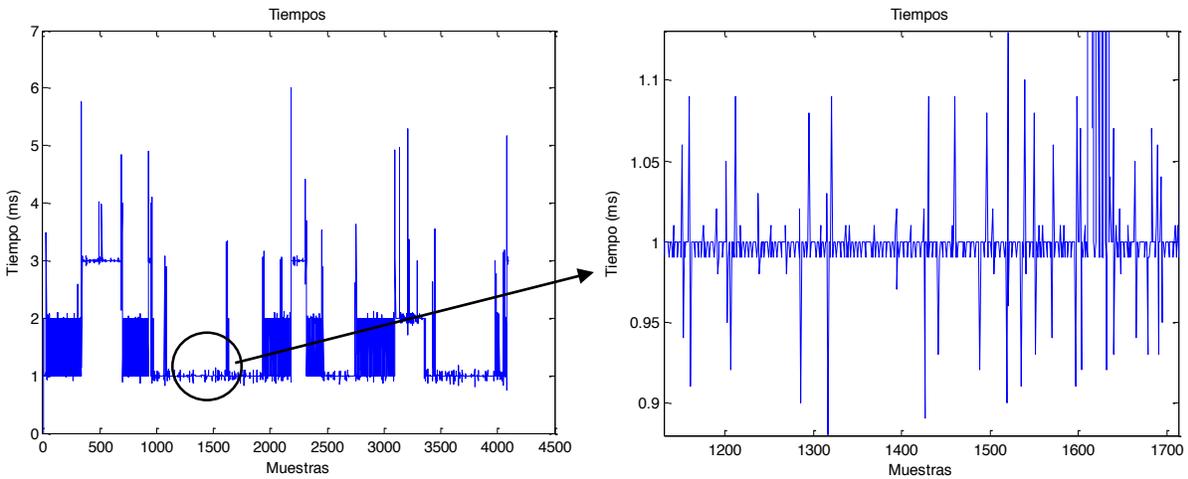


FIGURA 4.5. Retardos del sistema.

Las pruebas en la Figura 4.5., muestran claramente que existen muchas variaciones en cuanto a los tiempos en los que un valor válido es enviado, en algunos casos alcanza hasta de 6 ms de retraso, pero también es evidente que el periodo que más predomina es el de 1 milisegundo, es por ésta razón que el controlador se fijó para trabajar a una frecuencia de 1 KHz. Debe añadirse que, si la respuesta fuera mucho más lenta sería requerido un controlador que trabaje a una frecuencia menor, pues cuando la referencia se mantiene mucho tiempo, la respuesta del error en el controlador digital incrementa y al momento de cambiar provoca jaloneos en el robot.

4.3. Pruebas de seguimiento

En este punto se planteó utilizar pruebas con perfiles definidos, con el fin de demostrar el seguimiento del controlador. En este caso se utilizó el sistema representado en la Figura 4.6 que consiste de una computadora con Windows 7, el programa de CodeBlocks con las librerías libnifalcon, la tarjeta FPGA MCUAQ4X y el robot ROMAT 56. Para la prueba se mandará un perfil con 2500 cuentas que el controlador deberá de seguir. Esto se hace de manera asíncrona ya que las referencias son mandadas desde la computadora. En la tarjeta FPGA se implementó el diseño del apartado 3.9, el cual recibe los datos de forma asíncrona y los sincroniza para ser direccionados al PID. La prueba se hace de forma asíncrona debido a que el dispositivo háptico las enviará de esa manera, posteriormente se vacían los datos de la memoria hacia la computadora para discutir los resultados. Una de las consecuencias de usar un manipulador de 3 DoF, es que en el manipulador solo se mueven 5 ejes de los 6 disponibles de esta forma solo se presentan los ejes que si se mueven, los cuales son el eje 1, 2, 3, 5 y 6. Finalmente se calculará el error cuadrático medio (ECM) para cuantificar el funcionamiento del controlador.

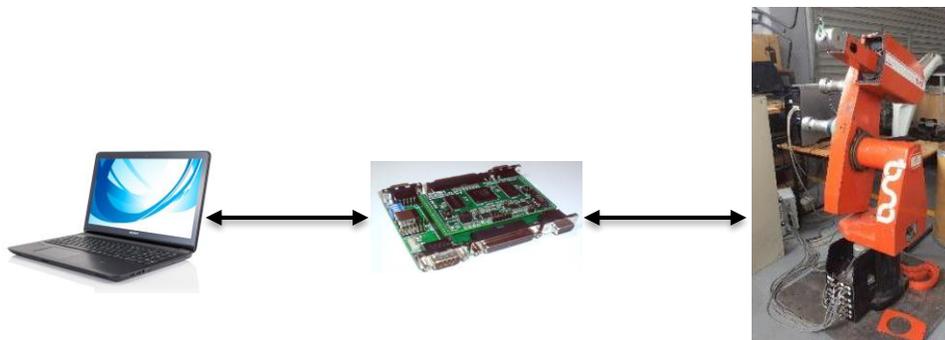


FIGURA 4.6. Sistema dedicado a la sintonía y prueba de controladores.

A continuación, se presentan los controladores utilizados para las pruebas (Tabla 5), que se obtuvieron después de haber realizado la identificación de la planta, la sintonización por margen de fase (vistos en los apartados 3.7 y 3.8.) y el ajuste fino mediante prueba y error para dar un mejor resultado.

TABLA 5. Controladores analógicos

	P	I	D
Eje 1	95	125.5	0.54
Eje 2	700	8100	1.014
Eje 3	641.891	14039.592	0.312
Eje 4	180	1100	0.914
Eje 5	1671	3700	0.0514
Eje 6	220	910	0.871

Los controladores fueron discretizados y ajustados para ser implementados dentro de una arquitectura tipo filtro IIR (Infinite Impulse Response, Respuesta Infinita al Impulso), como se trató en el apartado 2.9. El formato utilizado para la conversión a binario es de 12 bits para enteros y 20 bits para las fracciones, aplicado a las constantes a_0 , a_1 y a_2 .

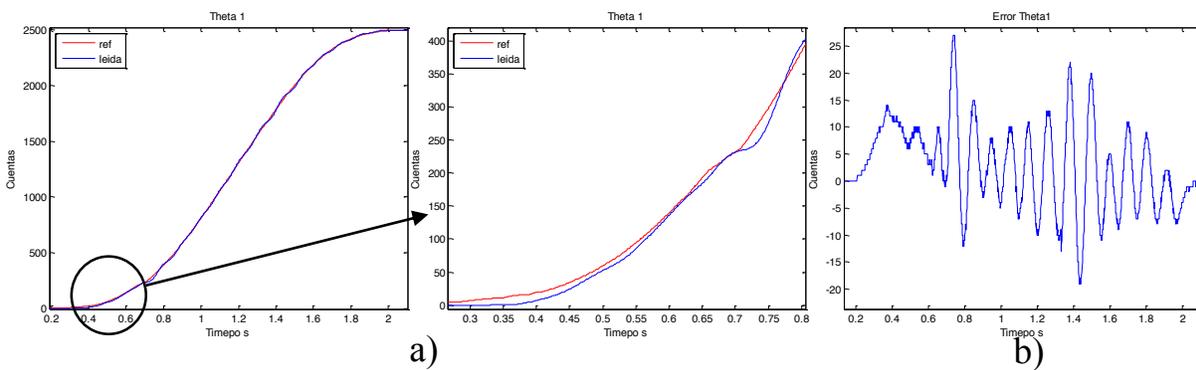


FIGURA 4.7. Prueba del controlador eje 1 a) Seguimiento de perfil, b) Error.

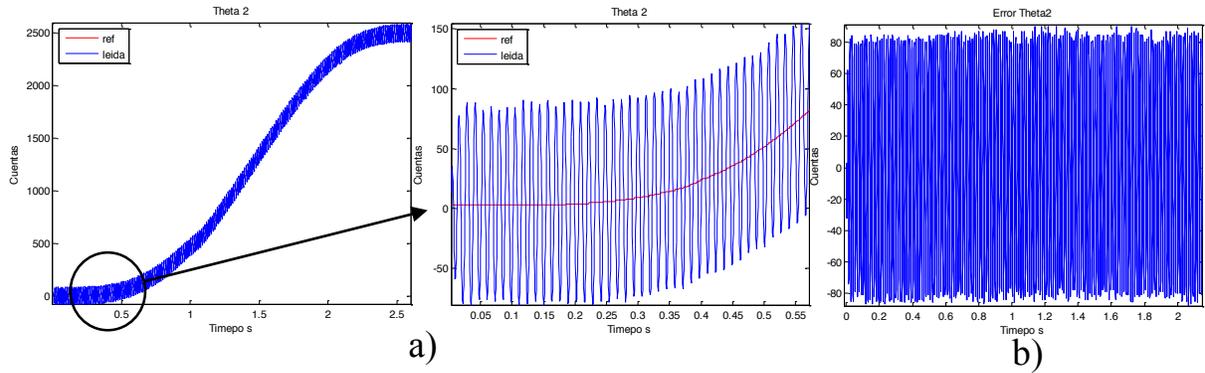


FIGURA 4.8. Prueba del controlador eje 2 a) Seguimiento de perfil, b) Error.

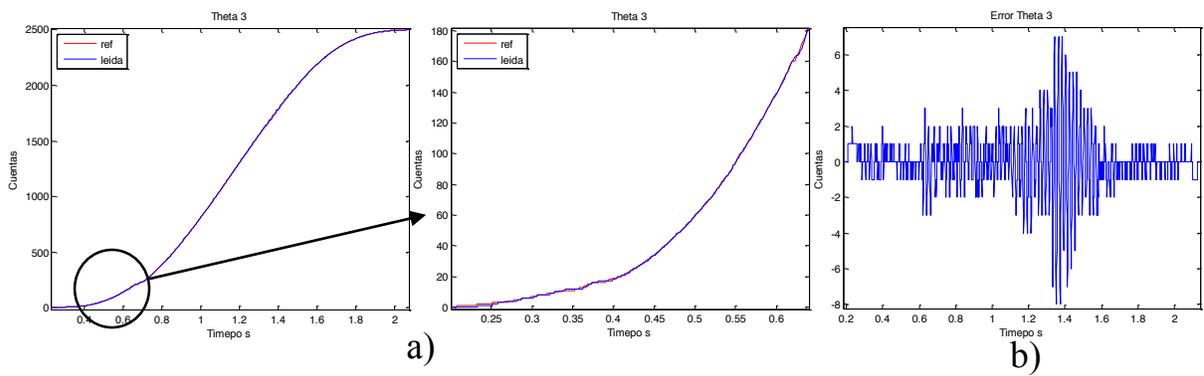


FIGURA 4.9. Prueba del controlador eje 3 a) Seguimiento de perfil, b) Error.

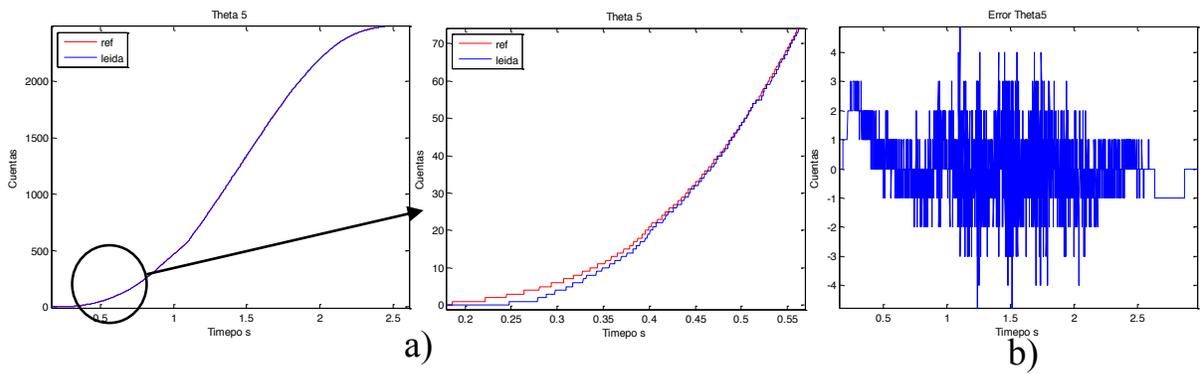


FIGURA 4.10. Prueba del controlador eje 5 a) Seguimiento de perfil, b) Error.

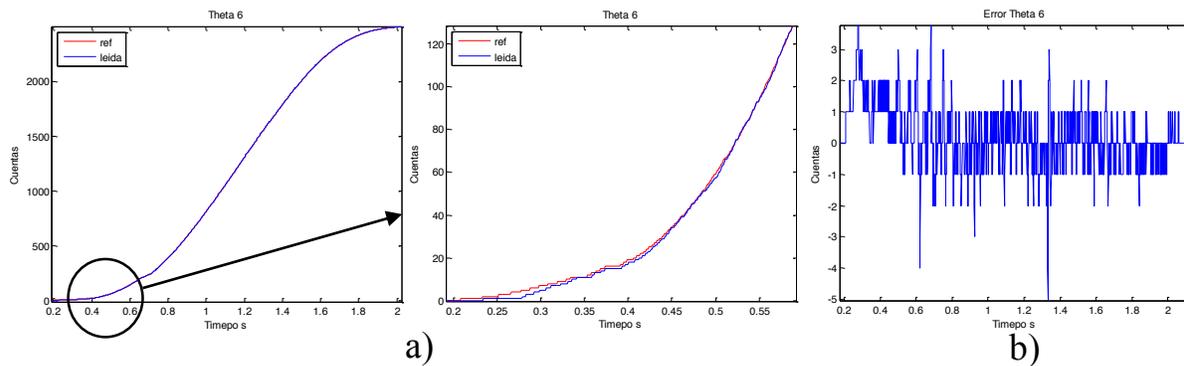


FIGURA 4.11. Prueba del controlador eje 6 a) Seguimiento de perfil, b) Error.

TABLA 6. Error cuadrático medio para perfiles

Ejes	θ_1	θ_2	θ_3	θ_5	θ_6
ECM	1.124×10^{-5}	5.608×10^{-4}	7.661×10^{-7}	2.433×10^{-7}	2.346×10^{-7}

De los resultados obtenidos para el eje 1 de la Figura 4.7a, se observa un ligero retraso al inicio, un poco debido al controlador y a la carga del eslabón, sin embargo, sigue la referencia como lo demuestra el ECM visto en la Tabla 6, el error a través del tiempo se muestra en la Figura 4.7b, donde se hace evidente el retraso. El eje 2 es el que presenta mayor problema debido al juego mecánico entre sus engranes como se muestra en la Figura 4.8a. A pesar de tener errores de ± 80 cuentas como se observa en la Figura 4.8b, éstas son insignificantes debido a la amplia movilidad de este eje como se verá en pruebas posteriores. En la implementación del eje 3 visto en la Figura 4.9a, se observa que genera menos retraso y se mueve de manera estable en comparación con el eje 2, esto se debe a que este eje no tiene juego mecánico, el error de este eje se muestra en la Figura 4.9b en el cual se aprecia poco error. El eje 5 que se muestra en la Figura 4.10a, presenta una gran mejoría en el seguimiento ya que maneja menor carga que los anteriores, lo cual se observa claramente al comparar las gráficas de errores y en el error cuadrático medio. El eje 6 (Figura 11a) es quien mejor seguimiento tiene al perfil, lo cual se hace evidente en la Figura 4.11b, cuyo error es aún menor que en los anteriores, en parte debido que es el que menor peso carga.

4.4. Pruebas finales

Esta prueba ayuda a evaluar el seguimiento de los controladores hacia una referencia generada por el háptico en uso continuo. Cabe destacar que el manipulador debería seguir la trayectoria que el operador genera en ese instante y que esta trayectoria puede tomar cualquier dirección en cualquier instante a necesidades del operario y de la situación del ambiente de trabajo. En este caso se utilizó el Falcon como mando, una computadora con Windows 7, el software Codeblocks con las librerías `lfnifalcon`, la tarjeta MCUAQ4X y el robot ROMAT 56, que se muestran en la Figura 4.12.

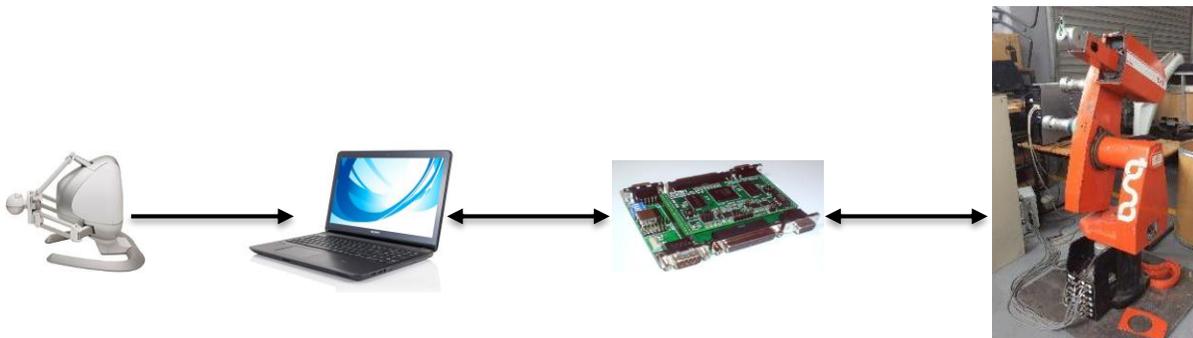


FIGURA 4.12. Implementación del sistema completo.

La prueba consiste en mover el dispositivo háptico desde un punto establecido en este caso el centro del espacio de trabajo, en esta prueba se implementa vía software el algoritmo de seguridad que se representa con un resorte cuya función para obtener la fuerza se representa por la ecuación (28).

$$F = 150(x_{Lim} - x) \quad (28)$$

Donde x_{Lim} es la restricción a la posición

x es la posición actual del Falcon

Los movimientos en esta prueba se observan en la Figura 4.13a, se realizan en zigzag primero de izquierda a derecha, arriba a abajo y de adentro hacia afuera, donde la línea roja representa el límite fijado por el usuario en este caso en $x_{Lim} = -0.045$ m, mientras que la fuerza de respuesta se muestra en la Figura 4.13b, en el cual se observa como experimenta el usuario la oposición al movimiento.

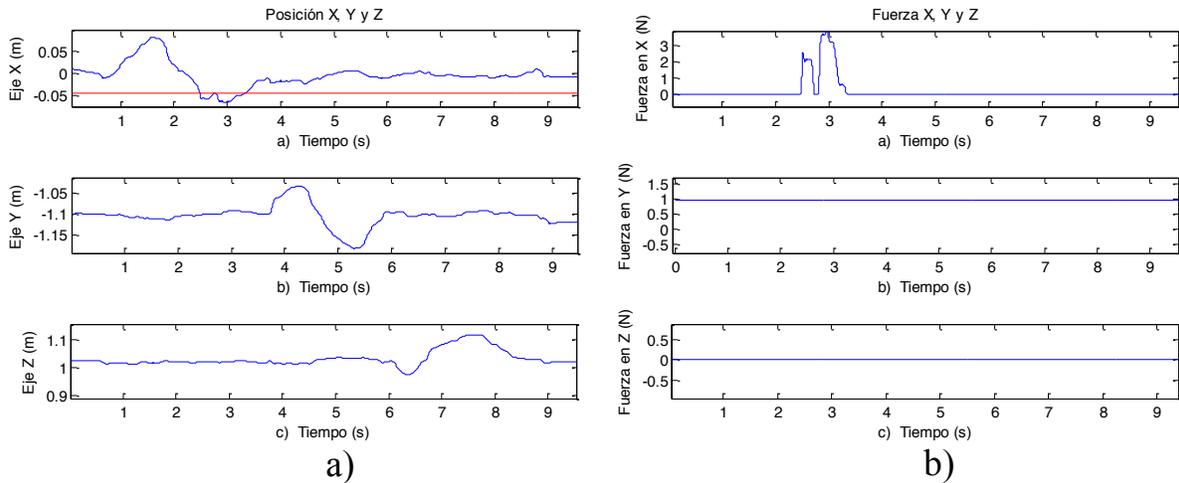


FIGURA 4.13. a) Trayectoria generada por el Falcon, b) Respuesta del sistema háptico.

Las referencias convertidas en cuentas se muestran a continuación para los ejes 1, 2, 3, 5 y 6.

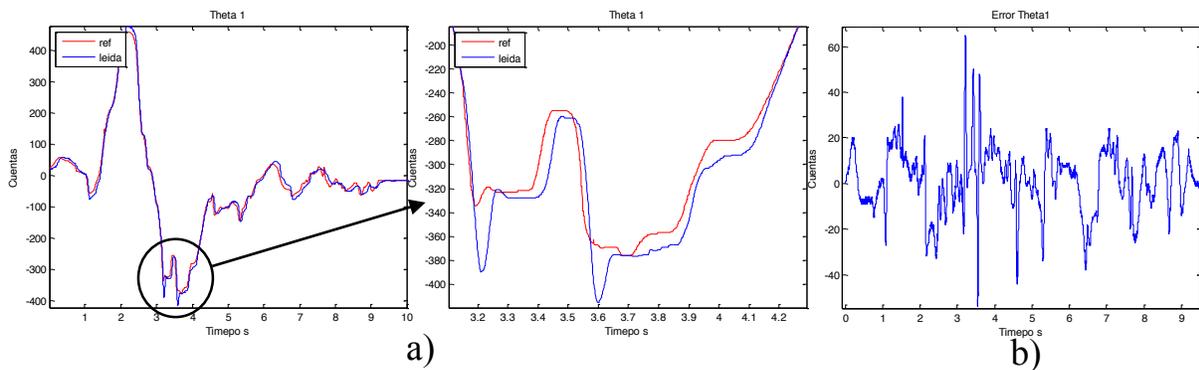


FIGURA 4.14. Prueba de trayectoria eje 1, a) Seguimiento, b) Error.

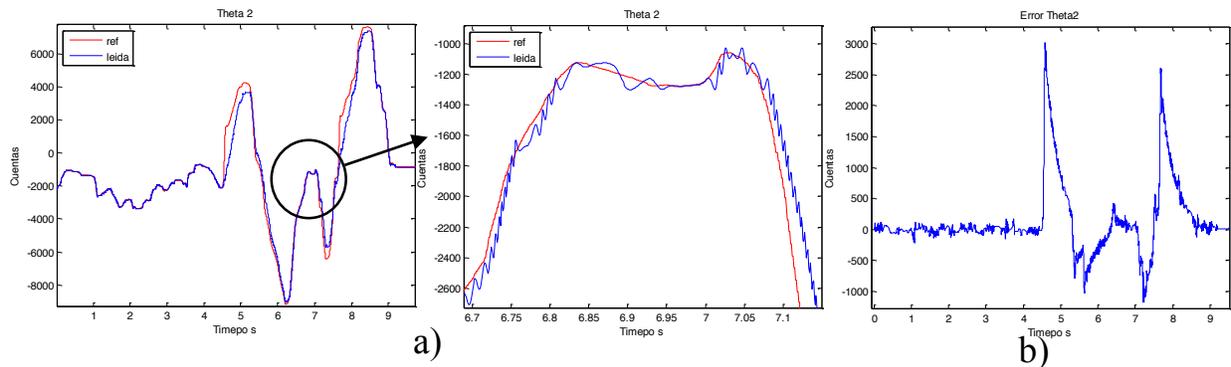


FIGURA 4.15. Prueba de trayectoria eje 2, a) Seguimiento, b) Error.

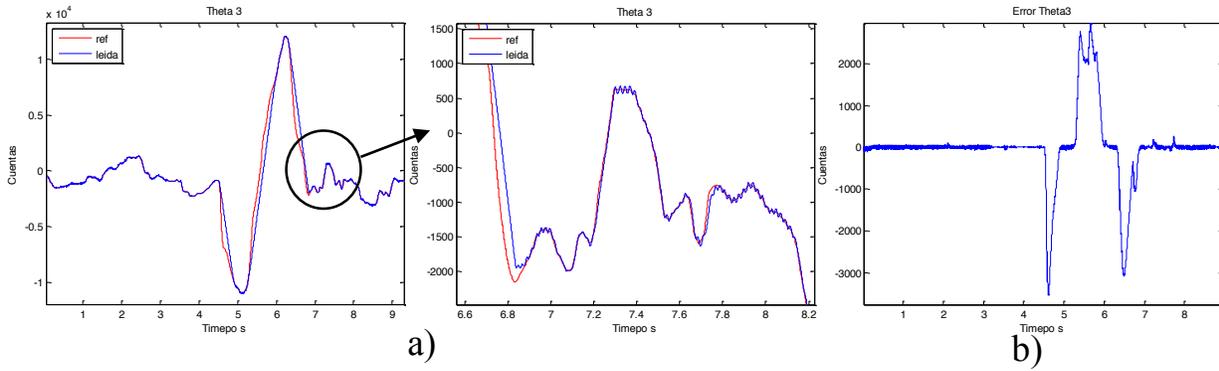


FIGURA 4.16. Prueba de trayectoria eje 3, a) Seguimiento, b) Error.

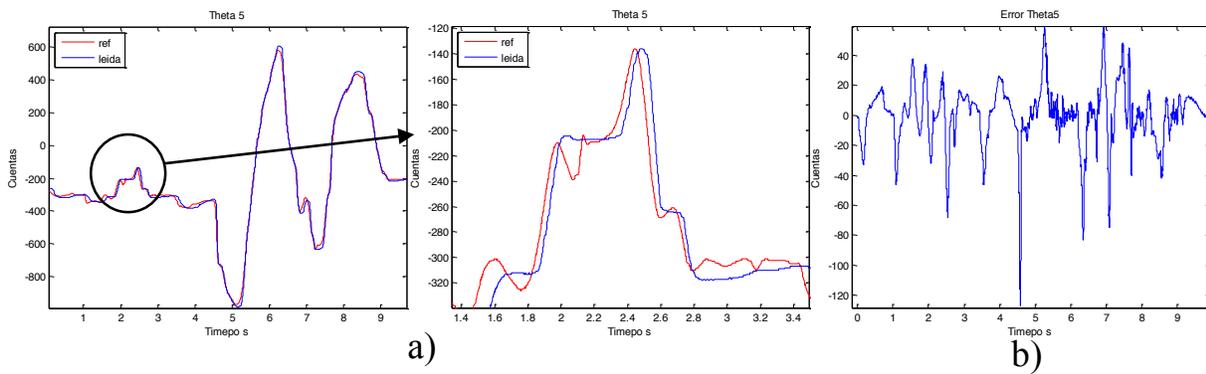


FIGURA 4.17. Prueba de trayectoria eje 5, a) Seguimiento, b) Error.

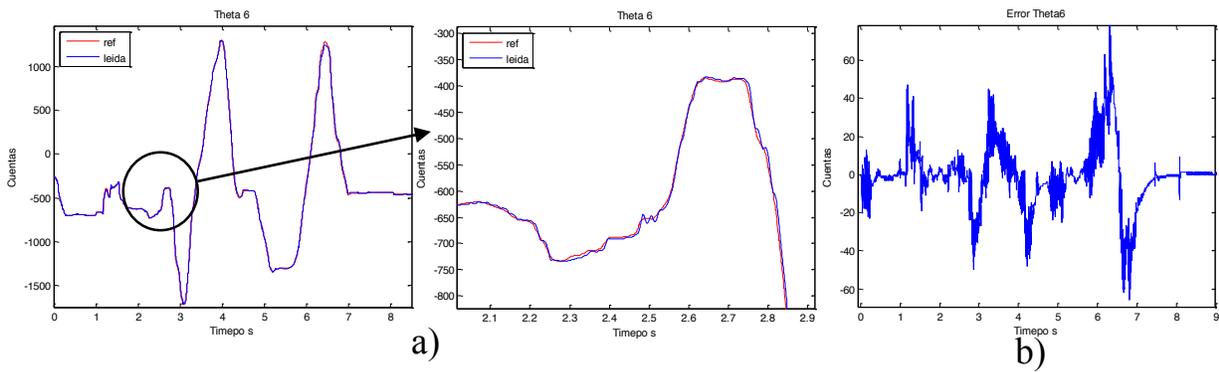


FIGURA 4.18. Prueba de trayectoria eje 6, a) Seguimiento, b) Error.

TABLA 7. Error cuadrático medio prueba

Ejes	θ_1	θ_2	θ_3	θ_5	θ_6
ECM	0.000253	0.0011	0.0009	0.000159	0.000163

En la prueba del controlador para el eje 1 que se observa en la Figura 4.14a, se mide un buen seguimiento, a pesar de que se observen algunos retrasos, también se empieza a observar en el error que se muestra en la Figura 4.14b como lo afecta una trayectoria oscilante, el error cuadrático medio se muestra en la Tabla 7 y define un seguimiento aceptable, ya que este eje está estrechamente relacionado con la coordenada X , se alcanza a preciar el efecto háptico en la ampliación de la Figura 4.14a. El eje 2 sigue mostrando vibraciones sin embargo como los movimientos son muy amplios casi no se observa su efecto, la Figura 15b da una mejor idea de la evolución del seguimiento. El eje 3 tiende a seguir la trayectoria sin embargo la lentitud de este, genera error en el seguimiento como se puede observar en las Figuras 16a y 16b, además este eje acarrea el efecto de las vibraciones generadas por el eje 2. El eje 5 presenta menor error que los ejes anteriores en parte debido a que sus movimientos son de menor magnitud por esta razón el seguimiento es mucho mejor como se observa en la Figura 17a y en la Tabla 7. Por último el eje 6 mostrado en la Figura 18a, al igual que el eje 5 demuestra un seguimiento con pocas perturbaciones y retrasos. Cabe mencionar que para mostrar más detalle en las gráficas solo se muestra el diferencial de cuentas entre la posición inicial y el movimiento generado por el dispositivo Falcon.

En general se observó buen seguimiento a las referencias del sistema, comenzando por el eje 1 quien se mueve de forma suave a pesar del frenado causado por la pared virtual, en gran medida auxiliado por el controlador PID. Las vibraciones mencionadas en los ejes 2 y 3 no tuvieron un efecto apreciable en el sistema maestro, mientras que los ejes 5 y 6 su efecto fue nulo, debido en parte a que esos ejes no cargan una masa grande que sea perturbada por la vibración.

Es importante considerar que para el sistema propuesto no se considera un modelo dinámico del robot y la retroalimentación háptica hacia el dispositivo FALCON, sin embargo el tipo de controlador implementado permitió el seguimiento de las referencias variables.

CAPÍTULO V

CONCLUSIONES

Los manipuladores son comúnmente utilizados para aplicaciones pre-programadas, sin embargo, cuando la tarea es compleja se requiere de la intervención humana. La finalidad de este trabajo fue la implementación de un sistema maestro-esclavo que permita capturar los movimientos del Falcon sin mayor complejidad para el usuario y reproducirlos de una manera rápida en un manipulador industrial.

Se puede decir que la integración del dispositivo háptico como mando para el robot y la inclusión de las ecuaciones de cinemática inversa mejoraron la operatividad del sistema, debido a que es más cómodo para el usuario simplemente realizar un movimiento en el espacio del control háptico, que utilizando los métodos convencionales.

La creación de un modelo virtual del manipulador fue de mucha ayuda durante la etapa de pruebas, ya que permitió verificar las ecuaciones de la cinemática inversa y la visualización del movimiento de los eslabones en conjunto. La adición de la librería *haptik* al sistema permitió mandar referencias continuas, de este modo los movimientos del Falcon son reproducidos en línea y la implementación de una medida de seguridad demostró la importancia de usar un dispositivo háptico ya que fue efectivo al momento de restringir el movimiento. Además, sin utilizar el dispositivo háptico, este modelo virtual se le pueden añadir trayectorias en el espacio euclidiano y con orientación variable para observar la evolución del movimiento de los eslabones durante una tarea.

La búsqueda del mejor controlador PID, incrementó mucho el seguimiento en cada uno de los ejes, ya que a pesar de que había trayectorias irregulares el controlador mostró bajo sobrepaso en cuanto a los tres primeros ejes a pesar de la fuerza de inercia y en los últimos dos exhibió un buen seguimiento. Ajustado en su forma recursiva, el PID es fácil de implementar en FPGA, además de que ocupa pocos recursos en hardware.

Al utilizar un controlador propio implementado en hardware y de arquitectura abierta permitió obtener la sincronía necesaria que las implementaciones en software no pueden ofrecer y que es requerida para que el movimiento sea lo más fluido posible, además de que

libera de carga de procesamiento a la computadora para que esta se enfoque en tareas que no son fáciles de implementar en hardware, tales como una simulación virtual, operaciones con puntos flotantes entre otras tareas.

Como prospectivas se podría plantear el mejorar el modelo del manipulador considerándolo esta vez como un sistema acoplado en el cual el movimiento de todos los otros ejes contribuye al modelo de los ejes individuales. Mejorar el controlador en base a un nuevo modelo y tal vez probar otro método de modelado como; redes neuronales o espacio de estados. Ver la posibilidad de añadir sensores de proximidad en el órgano terminal del robot y utilizar la corriente de los servoamplificadores para tener el sistema háptico completo. Buscar la opción de completarlo integrando un sistema de video para poder manejar el robot vía internet utilizando un dispositivo háptico.

BIBLIOGRAFÍA

ABB, 2013. IRB 140 Industrial Robot User's Guide, Zurich, Switzerland.

Aguado Behar A., 2000. *Temas de identificación y control adaptable*. 1ra edición PALCIEN. ISBN: 959-7056-11-9

Angeles J., 2007. *Fundamentals of Robotic Mechanical System*, 3rd Edition Springer Science Business Media.

Arteaga R., 2001. *Estudio y simulación en 3d de la cinemática de un robot manipulador con seis grados de libertad*. Tesis de licenciatura, Universidad Autónoma de Querétaro, México.

Chotiprayanakul P. y Liu D. K., 2009. *Workspace Mapping and Force Control for Small Haptic Device based Robot Teleoperation*. Memorias de 2009 IEEE International Conference on Information and Automation. PP 1613 – 1618.

Geomagic, 2013. haptic application gallery, Woburn, MA, USA.

Glamnik A. and Šafarič R., 2012. *Control of KUKA KR 5 robot with a háptico device*. Memorias de 2012 9th International Conference on. PP 1 – 7.

Hyuk Kim Y., Phong L. D., Park W. M., Kim K. y Rha K. H., 2009. *Laboratory-level Telesurgery with Industrial Robots and Haptic Devices Communicating via the Internet*. International Journal Of Precision Engineering And Manufacturing Vol. 10, No. 2, PP 25-29.

Jaen-Cuellar, Arturo Y., De Santiago-Perez, Jesus J., Romero-Troncoso, Rene J., Morales-Velazquez, Luis, Benitez-Rangel, Juan P., Osornio-Rios, Roque A.. 2012. *A hardware-software system for coordinated multi-axis control based on a non uniform rational B-splines interpolator applied to industrial computer numerically controlled machines*. Proceedings Of The Institution Of Mechanical Engineers Part I-Journal Of Systems And Control Engineering, Vol. 226, PP 831 - 840.

Jaen-Cuellar Arturo Y., Romero-Troncoso Rene de J., Morales-Velazquez Luis and Osornio-Rios Roque A., 2009. *PID-Controller Tuning Optimization with Genetic Algorithms in Servo Systems*, International Journal of Advanced Robotic Systems, Vol. 10, DOI: 10.5772/56697

López J. S. M., 2007. *Aportaciones al diseño mecánico de los entrenadores basados en realidad virtual*. Tesis de doctorado, Universidad Rey Juan Carlos de Madrid, España 2007.

Martinez-Prado M., Franco-Gasca A., Herrera-Ruiz G., Soto-Dorantes O., 2012. *Multi-axis motion controller for robotic applications implemented on an FPGA*. Springer-Verlag London 2012, PP 2367–2376.

Mitsou N. C., Velanas S. V., Tzafestas C. S., 2006. *Visuo-Haptic Interface for Teleoperation of Mobile Robot Exploration Tasks*. Memorias de The 15th IEEE International Symposium on. PP 157 -163.

Mizoguchi T., Nozaki T., and Ohnishi K., 2013. *Stiffness Transmission of Scaling Bilateral Control System by Gyration Element Integration*. IEEE Transactions On Industrial Electronics, Vol. 61, NO. 2, PP 1033 – 1043.

Morales-Velázquez L., 2010. *Diseño de plataforma hardware-software para el desarrollo de aplicaciones industriales basadas en FPGA*. Tesis de doctorado, Universidad Autónoma de Querétaro Facultad de Ingeniería, México 2010.

Morales-Velazquez L., Romero-Troncoso R. D. J., Osornio-Rios R. A., Herrera-Ruiz G., Santiago-Perez J. J., 2010. *Special purpose processor for parameter identification of CNC second order servo systems on a low-cost FPGA platform*. ELSEVIER Mechatronics, pp. 265-272.

Morgan R., 2011. *Haptic control whit a robotic gripper*, Tesis de Maestría, University Öbreo, Suecia.

Munoz-Barron B., Rivera-Guillen J. R., Osornio-Rios R. A., Romero-Troncoso R. J., 2014. *Sensor Fusión for Join Kinematic Estimation in Serial Robots Using Encoder, Accelerometer and Gyroscope*. Springer Journal of Intelligent & Robotic Systems, DOI 10.1007/s10846-014-0125-7.

Novint, 2013. Falcon specifications User's Guide, Albuquerque, NM, USA.

Ogata, K. 2002. *Sistemas de Control en Tiempo Discreto*, 2da edición. Prentice Hall.

Ogata, K. 2010. *Ingeniería de Control Moderna*, 5ta edición. Prentice Hall.

Radi M. y Reinhart G., 2009. *Idustrial Haptic Robot Guidane System for Assembly Process*. Memorias de Haptic Audio Visual Environments and Games, 2009. HAVE 2009. IEEE International Workshop on. PP 69 - 74.

Rodriguez-Donate C., Osornio-Rios R. A., Rivera-Guillen J. R., Romero-Troncoso R. J., 2011. *Fused Smart Sensor Network for Multi-Axis Forward Kinematics Estimation in Industrial Robots*. Sensors 2011, 11, 4335-4357.

Rosheim Mark E., 1994. *Robot evolution: the develompment of anthrobotics*, 1st Edition, John Wiley & Sons.

Spong M. W., Hutchinsong S. y Vidyasagar S., 2004. *Robot Dynamics and Control*, 2da edición, 2004, 79-89.

Salisbury C., Gillespie R.B., Hong Tan, Barbagli, F. y Salisbury, J. K., 2009. *Effects of Haptic Device Attributes on Vibration Detection Thresholds*, Memorias de Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. PP 115 – 120.

Tanaka H., Ohnishi K., Nishi H., Kawai T., Morikawa Y., Ozawa S., and Furukawa T., 2009. *Implementation of Bilateral Control System Based on Acceleration Control Using FPGA for Multi-DOF Haptic Endoscopic Surgery Robot*. IEEE Transactions On Industrial Electronics, Vol. 56, NO. 3, PP 618 – 627.

TABLA 8. Abreviaturas utilizadas en este trabajo

ADC	Analog to digital converter, Convertidor analógico a digital.
DAC	Digital to analog converter, Convertidor digital a analógico.
D-H	Denavit-Hartenberg
DoF	Degrees of Freedom, Grados de libertad.
ECM	Error Quadratico Medio.
FPGA	Field Programmable Gate Array, Arreglo de compuertas de campo programable.
MCUAQ4X	Controlador de Movimientos 4 ejes, UAQ.
PUMA	Programmable Universal Manipulation Arm, Brazo programable de manipulación universal.
VHDL	Very High-Speed Integrated Circuit Hardware Description Language, Lenguaje Descriptivo de Hardware para circuitos integrados de muy alta velocidad.

Apéndice A

Artículo publicado en congreso

Sistema háptico aplicado a un robot tipo PUMA simulado dentro de un entorno virtual

F. J. Mendoza-Galindo¹, B. Muñoz-Barrón¹, D. A. Elvira-Ortiz¹, C. S. López Cajún², R. A. Osornio-Ríos¹

¹ HSPdigital-CA Mecatrónica, Facultad de Ingeniería, Universidad Autónoma de Querétaro, Campus San Juan del Río, Río Moctezuma 249, Col. San Cayetano, 76807 San Juan del Río, Qro., México;

² División de Estudios de Posgrado, Facultad de Ingeniería, Universidad Autónoma de Querétaro, Cerro de las Campanas s/n, Col. Niños Héroe, 76010, Querétaro, Qro., México.

fmendoza@hspdigital.org (F.J.M.-G.); bmunoz@hspdigital.org (B.M.-B.); delvira@hspdigital.org (D.E.E.-O.); cajun@uaq.mx (C.S.L.-C.); raosornio@hspdigital.org (R.A.O.-R)

Resumen— La necesidad de manejar brazos robóticos de manera rápida y sencilla ha cobrado gran importancia en el ámbito industrial; además, para determinados procesos, añadir el sentido del tacto al manejo, ayuda a prevenir accidentes. El avance en las nuevas tecnologías ha permitido la creación de dispositivos que pueden fijar posición y realimentar fuerza, los cuales se conocen como hápticos. En este trabajo se plantea utilizar un dispositivo háptico para el manejo de un robot tipo PUMA dentro de una interfaz virtual. Al obtener las ecuaciones de cinemática inversa del robot, es posible que los movimientos dentro del entorno virtual se realicen a la par que los del operador. Adicionalmente, se implementó un algoritmo de seguridad el cual restringe los movimientos del robot dentro de su espacio de trabajo, esto se ve reflejado en el háptico como una fuerza opositora en dirección contraria al movimiento. Diversas pruebas son realizadas para probar la validez del algoritmo propuesto. El entorno desarrollado puede ser usado como banco de pruebas y centro de entrenamiento para los operadores del robot.

Palabras Clave— retroalimentación de fuerza, manipulador, cinemática inversa.

Abstract— Industrial applications require that robotic arms can be manipulated in a simple and fast way. Further, in some process, if the touch sense is used, then accidents can be prevented. The haptic is a device that can get position and force feedback; these features allow its use for robotic arms operation. In this work, a PUMA robot is handled using a haptic device into a virtual environment. The inverse kinematic equations are obtained so the operator movements can be replicated at the same time into the virtual environment. Additionally, a safety algorithm is proposed by sending a limiting force to the haptic in order to restring the robot movements into its workspace. Several experiments are carried out in order to demonstrate the safety algorithm effectiveness. The virtual environment created can be also used for testing different trajectories so the operator can be trained.

I. INTRODUCCIÓN

El uso de entornos virtuales ha cobrado gran importancia en la simulación de procesos, lo que ha generado aplicaciones en

ingeniería, medicina, manufactura, entre otros, debido a que permiten obtener una buena aproximación sobre el entorno real de trabajo. La conjunción de las condiciones mencionadas previamente, junto con las mejoras en dispositivos con realimentación de fuerza ha permitido el desarrollo de aplicaciones hápticas lo que convierte a estos dispositivos en una herramienta poderosa. Para muchas aplicaciones, la interacción con un entorno virtual, junto con el sentido del tacto, ayuda al usuario a entender el entorno de trabajo sin la necesidad de estar físicamente en el lugar; además, se incrementa la experiencia del operador durante la simulación de un proceso. Por ejemplo, la retroalimentación háptica se ha vuelto muy útil en simuladores de entrenamiento para cirujanos, donde el maestro, un dispositivo háptico, controla el movimiento de un esclavo en este caso un robot virtual equipado con herramientas quirúrgicas [1].

A pesar de la existencia de diversos trabajos que involucran simulaciones con dispositivos hápticos, la mayoría continúan en la etapa de investigación [2], son prototipos diseñados para actuadores individuales [3], o se enfocan a tareas muy específicas [4]. En los trabajos que emplean dispositivos hápticos con una interfaz para simular manipuladores, suelen utilizarse manipuladores que son poco usados en la industria y tienen menos de seis grados de libertad, lo que simplifica el cálculo de su cinemática inversa [5]; otros utilizan dispositivos hápticos de seis grados de libertad para mover eslabón por eslabón y así evitar el cálculo de la cinemática inversa, pero esto incrementa el costo de la aplicación [6]. Las implementaciones de entornos virtuales suelen involucrar el empleo de más de un software diferente para sus módulos, lo que hace más complejo su desarrollo; además, al no emplear software comercial, la información relacionada a la aplicación es escasa [7, 8].

Este trabajo se propone el desarrollo de un sistema háptico para mejorar la operación de un brazo robótico. El esquema general del sistema propuesto se observa en la Fig. 1. En éste, se integra el dispositivo háptico Falcon de la Fig. 2a, originalmente enfocado al área de videojuegos, convirtiéndolo en un dispositivo de muy bajo costo en comparación con otros dispositivos hápticos, cuyo precio es diez veces mayor, que se orientan al área de investigación. Este dispositivo de tres grados de libertad fue utilizado para manejar un brazo robótico de seis grados de libertad, se utilizó el lenguaje VRML (Virtual Reality Modeling Language) para diseñar tanto el manipulador así como

su entorno virtual, todo se desarrolló sobre MATLAB el cual es un software completo y de uso muy común. Una de las principales fortalezas de esta propuesta es el hecho de que se realizará la simulación de un robot tipo PUMA (Programable Universal Machine for Assembly) de seis grados de libertad, cuyas dimensiones serán muy parecidas a las de un robot ROMAT 45 que se muestra en la Fig. 2b, el cual es un brazo robótico muy utilizado en la industria por su facilidad de movilidad dentro de su entorno de trabajo.

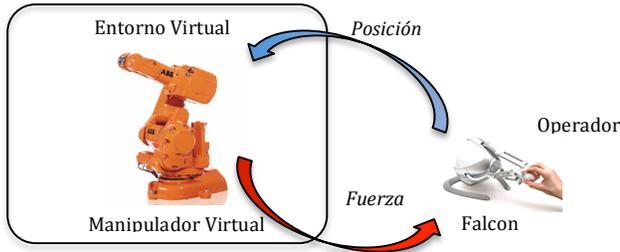


Fig. 1. Esquema general de un sistema háptico.

El artículo se organiza en las siguientes secciones: Introducción donde se expone el tema del trabajo, los antecedentes a este trabajo y una breve explicación del mismo, en la sección II se presenta el modelo matemático utilizado, además en la sección III se muestra la metodología empleada, mientras que en la sección IV se muestran los resultados que validan el trabajo, finalmente en la sección V se exponen las conclusiones obtenidas.

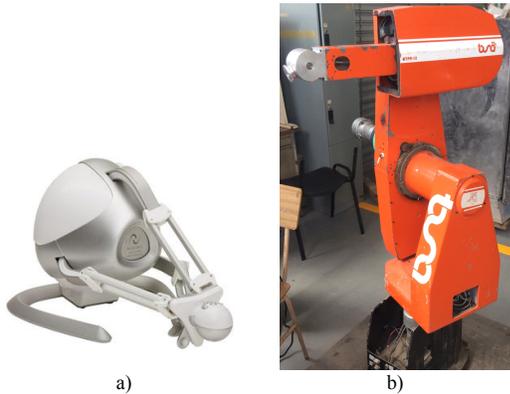


Fig. 2. a) Dispositivo Falcon, b) Robot utilizado para la simulación.

II. CINEMÁTICA INVERSA

Debido a que el dispositivo háptico entrega las referencias en coordenadas rectangulares es importante que el robot también se mueva en coordenadas rectangulares. Para eso, es necesario obtener la cinemática inversa del robot; de este modo el usuario puede manejar los tres ejes coordenados del robot sin la necesidad de intercambiar los ejes. El robot empleado es un manipulador desacoplado, cuyo diseño permite dividir el problema de la cinemática inversa en dos partes: el problema de la posición y el problema de la orientación. El ser un

manipulador desacoplado implica que los últimos tres ejes z_3, z_4 y z_5 intersectan en el punto C , conocido como el centro de la muñeca esférica, tal y como se muestra en la Fig. 4. Lo importante de este punto es que el movimiento de estos últimos tres ejes no afecta la posición de C ; de este modo, la posición del centro de la muñeca esférica sólo depende de los tres primeros ejes. El origen del marco de coordenadas propuesto es obtenido simplemente por la traslación b_6 a lo largo de z_5 , es decir:

$$C = P - Rb_6 \quad (1)$$

donde R es una matriz de rotación que define la orientación con la que el órgano terminal llega al punto propuesto P . Como solo se tienen tres ejes para manipular, R se fija por el usuario y es previamente calculada por los ángulos de Euler mediante:

$$R = \begin{pmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{pmatrix} \quad (2)$$

donde s_θ se usa para abreviar $\text{sen}\theta$.

A. Configuración articulada (RRR)

Para encontrar las relaciones en cada junta, es necesario encontrar los parámetros de D-H (Denavit-Hartenberg), los cuales están en función de la estructura física del robot que se muestra en la Fig. 3. Para definir éstos, se emplea el método propuesto en [9]. En la Tabla I se muestran los parámetros obtenidos.

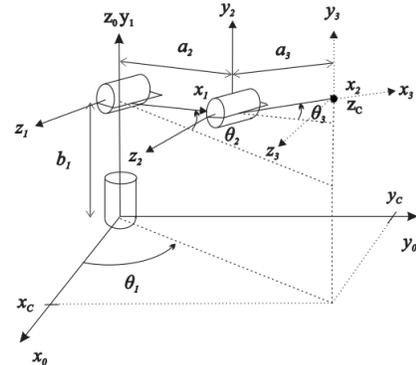


Fig. 3. Diagrama esquemático de los primeros 3 ejes.

TABLA I
PARÁMETROS DE LOS PRIMEROS TRES EJES

Unión	a_i	b_i	α_i	θ_i
1	0	0.9	90	θ_1^*
2	0.45	0	0	θ_2^*
3	0.54	0	0	θ_3^*

Para simplificar el cálculo de los tres primeros ángulos que definen la posición, se puede utilizar un análisis geométrico, debido a que la mayoría de los manipuladores son estructuralmente simples; además, existen pocas técnicas que

pueden manejar el problema de la cinemática inversa para configuraciones arbitrarias.

Al analizar la estructura de la Fig. 3, se puede observar la sombra proyectada de los últimos dos ejes.

Es evidente que θ_1 queda en función de x_c e y_c , es decir:

$$\theta_1 = \text{atan}\left(\frac{y_c}{x_c}\right) \quad (3)$$

En cuanto a los otros dos ángulos, se utiliza la ley de cosenos y una identidad de ángulos suplementarios:

$$\cos\theta_3 = \frac{x_c^2 + y_c^2 + (z_c - b_1)^2 - a_2^2 - a_3^2}{2a_2a_3} := D \quad (4)$$

por lo tanto, θ_3 se obtiene a partir de (4):

$$\theta_3 = \text{atan}\left(\frac{\pm\sqrt{1-D^2}}{D}\right) \quad (5)$$

θ_2 se obtiene a partir de la resta del ángulo formado entre C y la proyección de C , usando la intersección de z_0 y z_1 como pivote, menos el ángulo interno:

$$\theta_2 = \text{atan}\left(\frac{z_c + b_1}{\sqrt{x^2 + y^2}}\right) - \text{atan}\left(\frac{a_3 \sin(\theta_3)}{a_2 + a_3 \cos(\theta_3)}\right) \quad (6)$$

B. Configuración de la muñeca esférica

Una vez que se tienen los primeros tres ángulos, es posible determinar la orientación del marco de referencia en el punto C . Sabiendo que la rotación \mathbf{R} se puede obtener como el producto de las rotaciones del marco \mathcal{F}_0 al \mathcal{F}_3 y del marco \mathcal{F}_3 al \mathcal{F}_6 de cada eje, mediante:

$$\mathbf{R} = \mathbf{R}_3^0 \mathbf{R}_6^3 \quad (7)$$

De este modo es posible obtener los tres ángulos, de acuerdo a:

$$\mathbf{R}_6^3 = (\mathbf{R}_3^0)^{-1} \mathbf{R} = (\mathbf{R}_3^0)^T \mathbf{R} \quad (8)$$

donde una rotación que va del marco \mathcal{F}_{i-1} al marco \mathcal{F}_i , está definida como:

$$\mathbf{R}_i^{i-1} = \begin{pmatrix} \cos\theta_i & -\lambda_i \sin\theta_i & \mu_i \sin\theta_i \\ \sin\theta_i & \lambda_i \cos\theta_i & \mu_i \cos\theta_i \\ 0 & \mu_i & \lambda_i \end{pmatrix} \quad (9)$$

Los valores de λ_i y μ_i se definen en función de los parámetros de D-H:

$$\mu_i = \text{sen}\alpha_i \quad (10a)$$

$$\lambda_i = \text{cos}\alpha_i \quad (10b)$$

Teniendo esto en cuenta, se pasa a analizar la estructura de la muñeca esférica de la Fig. 4. En base a la forma de la muñeca

esférica, se determinan los parámetros de D-H, resumidos en la Tabla II.

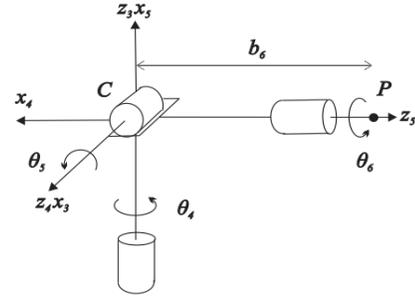


Fig. 4. Diagrama esquemático de los últimos tres ejes

TABLA II
PARÁMETROS DE LOS ÚLTIMOS TRES EJES

Unión	a_i	b_i	α_i	θ_i
4	0	0	-90	θ_4^*
5	0	0	90	θ_5^*
6	0	0.15	0	θ_6^*

Una vez obtenidos los parámetros, es necesario utilizar (8) y (9) para obtener tanto la matriz resultante de $(\mathbf{R}_3^0)^T \mathbf{R}$, como la matriz \mathbf{R}_6^3 . Primero se define a (11) como:

$$(\mathbf{R}_3^0)^T \mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (11)$$

mientras que la matriz de orientación es definida en (12), donde se usa c_4 para abreviar $\cos\theta_4$.

$$\mathbf{R}_6^3 = \begin{pmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - s_4 c_6 & c_4 s_5 \\ s_4 c_5 c_6 + c_4 s_6 & -s_4 c_5 s_6 + c_4 c_6 & s_4 s_5 \\ -s_5 c_6 & s_5 s_6 & c_5 \end{pmatrix} \quad (12)$$

Al observar cuidadosamente (12), es posible obtener los ángulos referentes a la muñeca esférica separando algunos elementos. Comenzando por la columna 3, se tiene que:

Para θ_4

$$\frac{\text{sen}\theta_4 \text{sen}\theta_5}{\text{cos}\theta_4 \text{sen}\theta_5} = \frac{r_{23}}{r_{13}} \therefore \theta_4 = \text{Atan}\left(\frac{r_{23}}{r_{13}}\right) \quad (13)$$

Para θ_5

$$\theta_5 = \text{Atan}\left(\frac{\pm\sqrt{1-r_{33}^2}}{r_{33}}\right) \quad (14)$$

mientras que para θ_6

$$\frac{-\text{sen}\theta_6 \text{sen}\theta_5}{\text{cos}\theta_6 \text{sen}\theta_5} = \frac{r_{32}}{r_{31}} \therefore \theta_6 = \text{Atan}\left(-\frac{r_{32}}{r_{31}}\right) \quad (15)$$

III. METODOLOGÍA

El método propuesto se resume en el diagrama de la Fig. 5. Comienza con el dispositivo háptico Falcon, el cual fija las

referencias de la posición; posteriormente, éstas son convertidas por la librería Haptik en datos numéricos para Simulink. Éstos entran a un bloque de escalamiento que convierte dichos datos en referencias de posición. De aquí surgen dos ramificaciones: la primera entra al módulo donde se implementan las ecuaciones de la cinemática inversa, para calcular los ángulos necesarios para llegar al punto propuesto, y alimentar al bloque VRSink que contiene el manipulador virtual. La segunda ramificación contiene al módulo de detección de objetos, que determina si ha ocurrido alguna colisión; de ser el caso, regresa un dato con la fuerza contraria al eje de detección en función de la cercanía con el mismo (entre más cerca del objeto, más fuerza opositora). Debido a que el Falcon está limitado a ± 2 libras o ± 8.9 N, se colocó un limitador para evitar un sobrepaso. Finalmente, el dato es enviado de regreso por la librería Haptik para posteriormente ejercer una fuerza al usuario a través del Falcon.

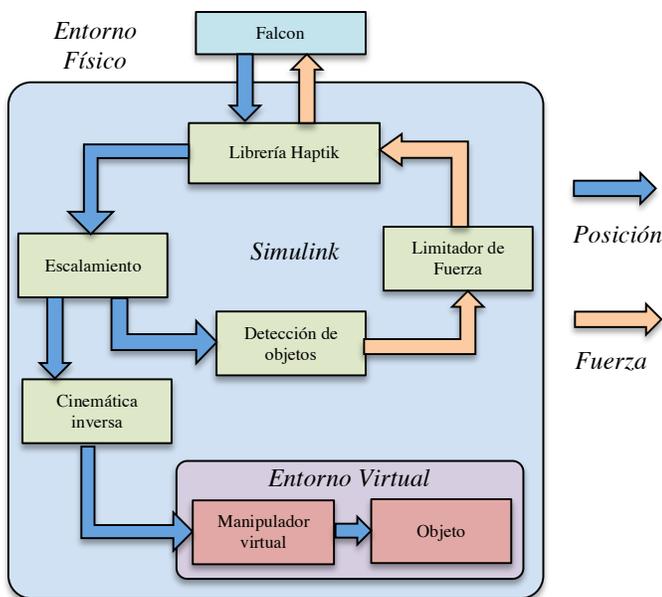


Fig. 5 Diagrama esquemático de los módulos del sistema.

A. Conexión entre háptico y módulo haptik

Para hacer una conexión entre el Falcon y Simulink son requeridas algunas librerías. SIRSLAB provee un conjunto de complementos que pueden ser utilizados para diferentes tipos de dispositivos hápticos tales como: Falcon, Phantom, Phantom Omni entre otros. Haptik es un código fuente de acceso libre y contiene librerías para C++, dll's necesarias para la conexión con la PC, ejemplos en Simulink y Matlab. Para Simulink, se provee un bloque llamado Haptik, el cual se utiliza para una comunicación bidireccional entre Falcon y PC. De este modo es posible adquirir la posición del Falcon e imprimir una fuerza hacia el mando, todo en coordenadas rectangulares [10].

B. Módulo de Escalamiento

Debido a que la librería haptik entrega los datos en un rango de valores entre +50 y -50 unidades en cada eje, es necesario convertir dichas unidades a metros (unidades requeridas por el módulo de cinemática inversa). Esto se logró mediante un factor de escalamiento implementado como una ganancia en Simulink. Una observación importante es que no es necesario que el robot se mueva por todo el espacio de trabajo, por lo que para aprovechar aún más la precisión del Falcon, se define una pequeña región para trabajar. Para conseguir este espacio, se traslada la coordenada del origen del háptico a la coordenada del centro del nuevo espacio de trabajo, además con este traslado se evita la singularidad generada al cruzar por el eje Z de (3) y (6).

C. Algoritmo para detección de objetos

Para saber cuándo se ha invadido un objeto, es necesario conocer los límites de éste. Para definir un muro sólo se requiere la coordenada de intersección a su eje normal, ya que únicamente es un plano. Teniendo este límite en cuenta, se realiza una diferencia entre la posición del manipulador y el límite; si el valor resultante es mayor a cero, no es requerida ninguna fuerza. Si la fuerza es menor a cero, la diferencia es amplificada y dirigida a su eje correspondiente, ya que el objeto ha sido invadido y es necesario repeler el órgano terminal.

D. Implementación en Simulink

Una vez obtenidos los elementos importantes, se crea un archivo Simulink donde se coloca el bloque Haptik y se implementan las ecuaciones de cinemática inversa obtenidas en la sección II. Para este propósito, se utiliza una subfunción M-File llamada *cinematica* para escribirlas como código de Matlab que contienen como entrada: la matriz R , como una constante de acuerdo a (2), y los parámetros de D-H de (10a) y (10b), en forma de un arreglo de vectores. Estas ecuaciones se calculan previamente en el espacio de trabajo ya que son variables fijas y no es necesario recalcularlas. A continuación, se fijan como entradas las coordenadas rectangulares que provienen del módulo Haptik. Como salidas del bloque *cinematica*, se obtienen los ángulos de cada eslabón y las coordenadas rectangulares del órgano terminal. Los ángulos se conectan al módulo de VRSink para poder observar sus movimientos en conjunto. Adicionalmente se creó un limitador de fuerza colocado a la salida del algoritmo de seguridad.

E. Entorno virtual

En cuanto al diseño del robot y el entorno virtual, se utilizó VRBuild de Matlab. VRBuild es un software intuitivo de diseño gráfico que permite crear y enlazar figuras en 3D. Esta herramienta tiene conexión directa con Simulink mediante el módulo de VRSink lo que permite entrada de señales para ejercer tanto traslaciones como rotaciones a cualquiera de los eslabones. En base a las dimensiones reales del robot ROMAT 45 de la Fig. 2b, se crean las uniones según los parámetros de D-H, indicando con una notación especial los elementos

rotacionales, para posteriormente crear las piezas que dan estética al robot.

IV. RESULTADOS

A. Manipulador virtual

Como se hizo mención, uno de los principales resultados fue el diseño del manipulador virtual bajo VRML, mostrado en la Fig. 6a. El manipulador virtual coincide con las medidas del robot de la Fig. 2b, lo que permite observar los movimientos de una mejor manera. Además, se logra determinar algunos puntos donde al manipulador le es imposible llegar, o incluso se puede visualizar si hay choque entre los eslabones. Para visualizar mejor la prueba experimental, se implementó un muro en la coordenada $z = 1.6$, el cual puede simular un elemento físico como un techo u otro objeto.

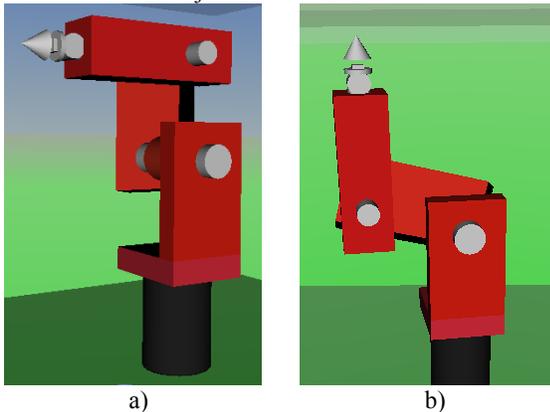


Fig. 6 Robot ROMAT 45 diseñado en VRML, a) posición de inicio, b) en operación

B. Análisis y discusión

A continuación se muestran algunas de las pruebas que demuestran el correcto funcionamiento del háptico como un dispositivo de seguridad. El experimento consiste en mover el Falcon en zigzag, tal y como se observa en la Fig. 6b. Primero, se mueve sobre el *eje Y*, luego sobre el *eje Z* y finalmente sobre el *eje X*, lo cual se ve reflejado en las posición del órgano terminal, ilustrado en la Fig. 7, donde la línea roja en la Fig. 7c indica el límite establecido que no debe ser sobrepasado por el órgano terminal del robot.

Cuando se alcanza $z = 1.6$, se crea una fuerza opositora en el háptico con las magnitudes de la gráfica en la Fig. 8c, las cuales están en función de la cercanía, lo que impide al usuario continuar con el movimiento. Esta fuerza repercute en gran parte sobre el *eje Z* del háptico, aunque también afecta a los otros dos ejes de la Fig. 7, que se observan en el intervalo de tiempo entre 40 s y 120 s.

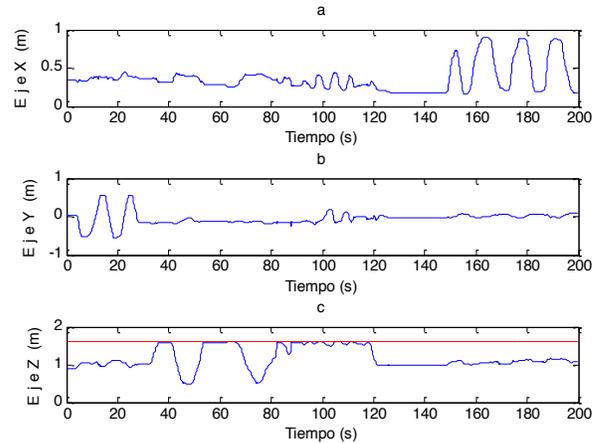


Fig. 7 Posición del robot fijada por el háptico

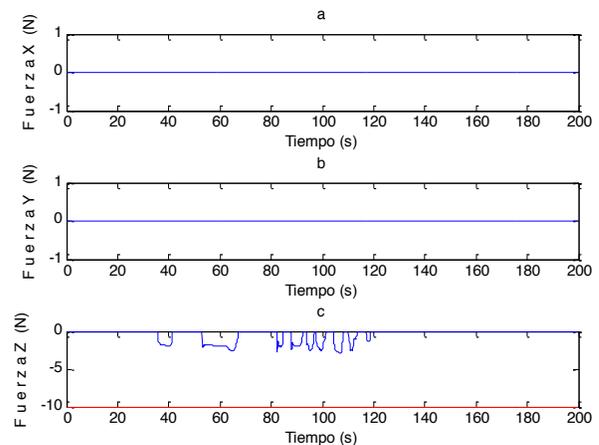


Fig. 8 Respuesta en fuerza del háptico al usuario

Un análisis más detallado de los ejes muestra algunos efectos interesantes sobre el muro virtual. En las Figs. 9b y 9c se observa que entre los 80 y 120 s, el efecto es directo sobre los ángulos θ_2 y θ_3 , ya que tienden a formarse planicies. También en la Fig. 9b se observa cómo cambia abruptamente alrededor de los 60 s, esto repercute en un gran esfuerzo al robot que también puede dañarse, lo que demuestra la importancia de realizar simulaciones previas a la implementación.

Del mismo modo, en la Fig. 10 se pueden observar los efectos sobre los últimos tres ángulos. A pesar de ser movimientos más cortos, su desplazamiento angular es discontinuo, como se aprecia en las Figs. 10a y 10b, que corresponden a las uniones de θ_4 y θ_6 , con los cuales debe tenerse cuidado durante su implementación física.

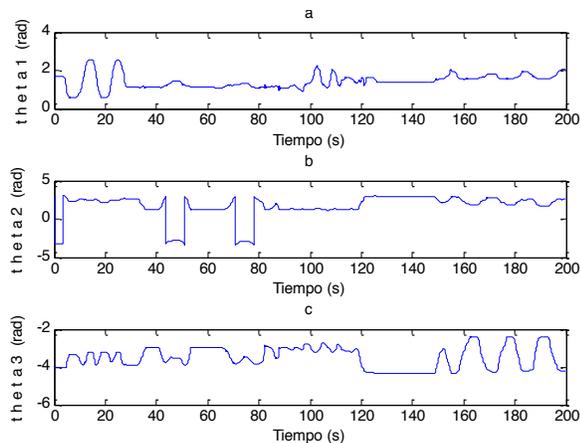


Fig. 9 Ángulos referentes a la posición

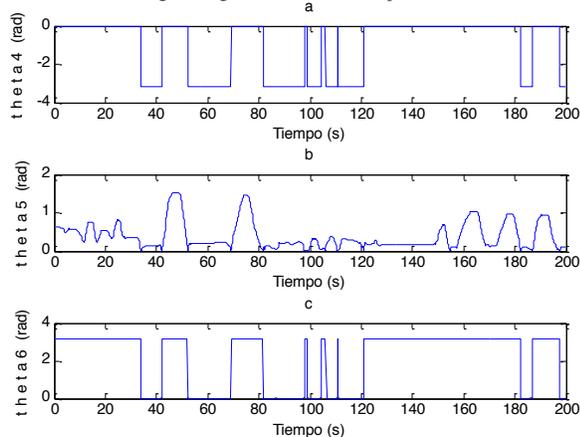


Fig. 10 Ángulos referentes a la orientación

V. CONCLUSIONES

Se desarrolló un sistema háptico completo utilizando un dispositivo de bajo costo. También se demostró que utilizar este tipo de dispositivos permite la implementación de algoritmos de seguridad para mantener la integridad del robot. Obtener las ecuaciones de cinemática inversa ayudó a que los movimientos del manipulador sean fluidos. Se desarrolló una interfaz virtual que permite realizar un gran número de pruebas como la planificación de trayectorias preprogramadas o el seguimiento de referencias a diferentes velocidades; además, en esta interfaz se pueden añadir un gran número de objetos como paredes, cajas y algunas otras herramientas virtuales que cumplan la función de objetos físicos. El desarrollo de simulaciones dentro de una interfaz virtual ayuda a visualizar situaciones que quizá no habían sido previstas. De lo anterior se concluye que la plataforma que se presenta es un entorno práctico para el entrenamiento de un operador, adicionalmente, permite la implementación de algunos algoritmos de seguridad aplicables a procesos reales.

Esta investigación abre la puerta a un gran número de trabajos a futuro, uno de ellos es el desarrollo de una interfaz gráfica

mucho más compleja que tenga la función de un sistema para el entrenamiento en operaciones *pick and place* (tomar y colocar), o un laboratorio para el entrenamiento a estudiantes en cirugía, que contenga un conjunto de herramientas de corte, sujeción o agujas, el cual permita sentir la suavidad del tejido de un paciente virtual, de tal forma que sea lo más real posible. Encontrar e implementar la dinámica del manipulador es otro trabajo a futuro, ya que ésta contiene efectos indeseados como inercias, que afectan la estabilidad del sistema. Para esto se puede crear un control bidireccional que permita fijar la posición del háptico en función de la posición del robot, lo que ayudaría aún más en cómo experimenta el usuario el movimiento del manipulador.

RECONOCIMIENTOS

Este proyecto fue financiado parcialmente por el Consejo Nacional de Ciencia y Tecnología (CONACYT) con la beca número...

REFERENCIAS

- [1] K. Y. Hyuk, L. D. Phong, W. M. Park, K. Kim y K. H. Rha. *Laboratory-level Telesurgery with Industrial Robots and Haptic Devices Communicating via the Internet*. International Journal of Precision Engineering And Manufacturing Vol. 10, No. 2, 2009, pp. 25-29.
- [2] T. Hilliard and Y. Pan, *Stabilization of Asymmetric Bilateral Teleoperation Systems for Haptic Devices with Time-varying Delays*, American Control Conference, 2013, pp. 4538-4543.
- [3] S. Sakaino, T. Sato, y K. Ohnishi, *Precise Position/Force Hybrid Control With ModalMass Decoupling and Bilateral Communication Between Different Structures*, IEEE Transactions on industrial informatics, Vol. 7, No. 2, 2011, pp. 266-276.
- [4] S. Mulatto, A. Fromaglio, M. Malvezzi y D. Prattichizzo. *Using Postural Synergies to Animate a Low-Dimensional Hand Avatar in Haptic Simulation*, IEEE Transactions on haptics, VOL. 6, NO. 1, 2013, pp. 106-116.
- [5] K. Chang, K. Seong, L. Hyo, L. Jong y K. Ki. *A Haptic Device Interface for the Operation of the Virtual Arm*, International Conference on Control Automation and Systems, 2010. 27-30, 2010 in KINTEX, Gyeonggi-do, Korea. pp. 1775-1758. -7
- [6] G. R. Luecke, *Haptic Interactions Using Virtual Manipulator Coupling With Applications to Underactuated Systems*, IEEE Transactions on robotics, Vol. 27, No. 4, 2011, pp. 730-740.
- [7] J. S. M. López. *Aportaciones al diseño mecánico de los entrenadores basados en realidad virtual*. Tesis de doctorado, Universidad Rey Juan Carlos de Madrid, España 2007.
- [8] G. R. Luecke y J. A. Beckman, *Haptic Interactions With Under-Actuated Robots Using Virtual Mechanisms*, IEEE International Conference on Robotics and Automation, 2008, pp. 2878-2883.
- [9] M. W. Spong, S. Hutchinson y M. Vidyasagar, *Robot Dynamics and Control*, 2da edición, 2004, 79-89.
- [10] Haptiklibrary, <http://www.haptiklibrary.org>, Marzo de 2014.