

UNIVERSIDAD TÉCNICA DE AMBATO



FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

MAESTRÍA EN AUTOMATIZACIÓN Y SISTEMAS DE CONTROL

Tema:

“CONTROL DE UN DISPOSITIVO HÁPTICO PARA
RETROALIMENTACIÓN DE FUERZAS ENFOCADO A TELE-PRESENCIA”

Trabajo de Investigación, previo la obtención del Grado Académico de Magister
en Automatización y Sistemas de Control.

Autor: Ing. Roberto Asdrúbal Segura Flores

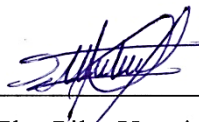
Director: Ing. Santiago Mauricio Altamirano Meléndez, Mg.

Ambato – Ecuador

2018

A la Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

El Tribunal receptor del Trabajo de Investigación presidido por la Ingeniera Elsa Pilar Urrutia Urrutia, Mg., e integrado por los señores Ingeniero Carlos Diego Gordón Gallegos, Dr., Ingeniero Franklin Wilfrido Salazar Logroño, Mg. e Ingeniera Rosa Estefanía Almache Hernández, Mg., designados por la Unidad Académica de Titulación de la Universidad Técnica de Ambato, para receptor el Trabajo de Investigación con el tema: “Control de un dispositivo háptico para retroalimentación de fuerzas enfocado a tele-presencia”, elaborado y presentado por el Ingeniero Roberto Asdrúbal Segura Flores, para optar por el Grado Académico de Magister en Automatización y Sistemas de Control; una vez escuchada la defensa oral del Trabajo de Investigación el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la UTA.



Ing. Elsa Pilar Urrutia Urrutia, Mg.

Presidente de Tribunal



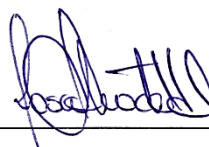
Ing. Carlos Diego Gordón Gallegos, Dr.

Miembro de Tribunal



Ing. Franklin Wilfrido Salazar Logroño, Mg.

Miembro de Tribunal

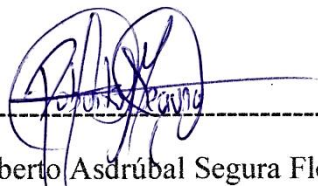


Ing. Rosa Estefanía Almache Hernández, Mg.

Miembro de Tribunal

AUTORÍA DEL TRABAJO DE INVESTIGACIÓN

La responsabilidad de las opiniones, comentarios y críticas emitidas en el Trabajo de Investigación presentado con el tema: “Control de un dispositivo háptico para retroalimentación de fuerzas enfocado a Tele-presencia”, le corresponde exclusivamente al Ingeniero Roberto Asdrúbal Segura Flores, autor bajo la Dirección del Magister Santiago Mauricio Altamirano Meléndez, Director del Trabajo de Investigación; y el patrimonio intelectual a la Universidad Técnica de Ambato.



Ing. Roberto Asdrúbal Segura Flores

C.C.: 1804484697

AUTOR



Ing. Santiago Altamirano, Mg

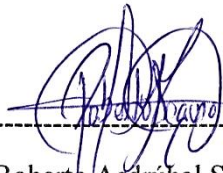
C.C.: 1802338838

DIRECTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que el Trabajo de Investigación, sirva como un documento disponible para su lectura, consulta y procesos de investigación, según las normas de la Institución.

Cedo los Derechos de mi trabajo, con fines de difusión pública, además apruebo la reproducción de este, dentro de las regulaciones de la Universidad.



Ing. Roberto Asdrúbal Segura Flores

C.C.: 1804484697

DEDICATORIA

Principalmente a Dios, Jesús, La Virgen María y el alma noble de mi difunto Papá Asdrúbal, que en ningún momento de mi vida me han dejado solo, han guiado mis pasos por el camino correcto, dándome fortaleza en los momentos difíciles.

Con el más profundo amor del mundo dedico este trabajo a mi Madre: Lida Violeta Flores, por todo su sacrificio a lo largo de nuestras vidas, quien me guía con verdaderos valores; enseñándome a encarar la adversidad sin perder nunca la dignidad, ni desfallecer en el intento, siendo ella la razón de mi existencia y quien me motiva a continuar por el sendero de la superación.

A mis queridos Abuelitos: Julio y Elcira, que Dios los tenga en la Gloria, cuyos sentimientos bondadosos, su amor y cariño, me enseñaron el verdadero significado de la vida. Al cariño, gratitud y apoyo moral de mis familiares y amigos para poder seguir siempre adelante.

Roberto Asdrúbal Segura Flores

AGRADECIMIENTO

A Dios, y a mi madre por su amor y bendiciones.

Un agradecimiento de manera especial al Ing. Santiago Altamirano Meléndez, por brindarme la oportunidad de recurrir a su capacidad y experiencia científica en un marco de confianza, afecto y amistad, fundamentales para la concertación de este proyecto. A la Universidad Técnica de Ambato en especial a la Facultad de Ingeniería en Sistemas, Electrónica e Industrial por contribuir a mi engrandecimiento personal y profesional, y hacerme ver que todo se logra con valentía, coraje, respeto y ardua dedicación para ser mejores día a día. A los distinguidos Maestros del programa de maestría, quienes me guiaron por el camino del bien, proporcionándome todos sus vastos conocimientos, haciendo de mi un Profesional íntegro. Y a todas aquellas personas que de alguna manera colaboraron con la realización de este proyecto.

Roberto Asdrúbal Segura Flores

ÍNDICE GENERAL DE CONTENIDOS

PORTADA.....	i
A LA UNIDAD ACADÉMICA DE TITULACIÓN.....	ii
AUTORÍA DEL TRABAJO DE INVESTIGACIÓN.....	iii
DERECHOS DE AUTOR	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE GENERAL DE CONTENIDOS.....	vii
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE TABLAS	xiv
RESUMEN EJECUTIVO	xv
EXECUTIVE SUMMARY.....	xvii
INTRODUCCIÓN	1
1. CAPÍTULO I.....	3
1.1. Tema de investigación.....	3
1.2. Planteamiento del problema	3
1.2.1. Contextualización.....	3
1.2.2. Árbol del problema	5
1.2.3. Análisis crítico	5
1.2.4. Prognosis	6
1.2.5. Formulación del problema	6
1.2.6. Preguntas directrices	7
1.2.7. Delimitación del problema.....	7
1.3. Justificación.....	7
1.4. Objetivos	8
1.4.1. Objetivo general	8
1.4.2. Objetivos específicos	9
2. CAPÍTULO II.....	10
2.1. Antecedentes investigativos	10
2.2. Fundamentación filosófica	11
2.3. Categorías fundamentales.....	12
2.3.1. Redes de inclusiones conceptuales.....	12

2.3.2.	Constelación de ideas	12
2.3.3.	Automatización	13
2.3.4.	Robótica	14
2.3.5.	Modelación.....	19
2.3.6.	Control en robótica.....	20
2.3.7.	Telerobótica.....	23
2.3.8.	Control bilateral y retroalimentación de fuerzas	28
2.3.9.	Hápticos.....	30
2.4.	Hipótesis	32
2.5.	Señalamiento de variables de la hipótesis	32
CAPITULO III		33
3.1.	Enfoque de la Investigación	33
3.2.	Modalidad de la Investigación	33
3.3.	Operacionalización de variables.....	34
3.4.	Recolección de información	36
3.5.	Procesamiento y análisis	36
CAPITULO IV		37
4.1.	Análisis de Resultados.....	37
4.1.1.	Controladores clásicos para robots terrestres	37
4.1.2.	Inmersión generada por los dispositivos hápticos.....	38
4.1.3.	Control de robots paralelos	40
4.1.4.	Características de un ambiente remoto	41
4.1.5.	Características de un robot maestro	42
4.1.6.	Características de los robots esclavos	45
4.1.7.	Generación de fuerzas en un robot maestro	48
4.1.8.	Control de un robot a través de la retroalimentación de fuerzas.....	51
5.	CAPITULO V	52
5.1.	Conclusiones	52
5.2.	Recomendaciones	52
6.	CAPITULO VI	54
6.1.	Antecedentes de la propuesta	54
6.2.	Justificación.....	54
6.3.	Objetivos de la propuesta	55
6.3.1.	Objetivo general	55

6.3.2.	Objetivos específicos	55
6.4.	Análisis de factibilidad	56
6.4.1.	Factibilidad operativa.....	56
6.4.2.	Factibilidad técnica	56
6.4.3.	Factibilidad económica	56
6.4.4.	Proyección a futuro	57
6.5.	Fundamentación científico – técnica.....	57
6.5.1.	Simulador V-REP	57
6.5.2.	Matlab	57
6.5.3.	Robot unicycle	58
6.5.4.	Manipulador robótico.....	61
6.6.	Metodología, Modelo operativo	64
6.7.	Modelo cinemático del Novint Falcon	65
6.7.1.	Análisis geométrico.....	65
6.7.2.	Método de control propuesto	71
6.7.3.	Análisis de estabilidad del controlador	72
6.7.4.	Sintonización del control PID	74
6.8.	Conexión de Novint Falcon a Matlab.....	74
6.9.	Validación del modelo matemático	76
6.9.1.	Primera prueba del controlador.....	77
6.9.2.	Segunda prueba del controlador.....	82
6.9.3.	Interacción entre MATLAB y V-REP	86
6.9.4.	Desarrollo de escenarios en V-REP	91
6.9.5.	Metodología de control	102
6.9.6.	Pruebas experimentales.....	106
6.9.7.	Experimento 1: Tele-Operación de un robot móvil.	106
6.9.8.	Experimento 2: Tele-Operación de un manipulador	113
6.9.9.	Conclusiones y recomendaciones	119
REFERENCIAS	122

ÍNDICE DE FIGURAS

Fig. 1.1: Árbol del problema	5
Fig. 2.1 Variables dependientes e independientes de las categorías fundamentales	12
Fig. 2.2 Constelación de ideas de la Variable independiente	12
Fig. 2.3 Constelación de ideas de la Variable dependiente.....	13
Fig. 2.4 Generalidades de la robótica.....	14
Fig. 2.5 Desplazamiento autónomo de Robots Móviles.	15
Fig. 2.6 Tipos de robots móviles: a) Robots con ruedas, b) Robots con patas y c) Robots tipo Oruga.	16
Fig. 2.7 Manipuladores Seriales, (a) KUKA con Pinza y(b) KUKA paletizador.	17
Fig. 2.8 Manipuladores paralelos, (a) Robot DELTA de 3 ejes y (b) Robot DELTA de 4 ejes.....	17
Fig. 2.9 Aplicaciones de los robots paralelos.	18
Fig. 2.10 Esquemas de planificación de movimiento.	21
Fig. 2.11 Control remoto de un robot en el entorno.	24
Fig. 2.12 Telepresencia con robots.	26
Fig. 2.13 Arquitecturas de control.	26
Fig. 2.14 Uso de la telerobótica para desactivación de explosivos.....	27
Fig. 2.15 Esquema de control con retroalimentación de fuerzas.	28
Fig. 2.16 Representación de la arquitectura Posición – Posición.	29
Fig. 2.17 Dispositivos Hápticos: (a)Entrada de señales hacia el entorno de trabajo, b) retorno de fuerzas para emular la interacción con el ambiente.....	31
Fig. 4.1: Tipos de dispositivos hápticos	40
Fig. 4.2: Dispositivos hápticos comerciales	43
Fig. 6.1: Cinemática del robot unicycle Pioneer 3-DX.....	58
Fig. 6.2: Análisis matemático para determinar la posición del robot	59
Fig. 6.3: Control en el punto frontal del robot.....	60
Fig. 6.4: Vectores y variables del punto desplazado para control	60
Fig. 6.5: Representación gráfica de un manipulador robótico.....	62
Fig. 6.6: Metodología para resolver la problemática.....	65
Fig. 6.7: Sistemas de referencia del dispositivo Háptico.....	66

Fig. 6.8: Parámetros geométricos del dispositivo Novint Falcon: a) vista lateral, b) vista superior, c) vista frontal, d) vista trasera	67
Fig. 6.9: Lectura/escritura de datos de Novint Falcon.....	76
Fig. 6.10: Simulación del Novint Falcon.....	77
Fig. 6.11: Ejecución de la primera trayectoria.....	78
Fig. 6.12: Errores en cada una de las componentes para la ejecución de la primera trayectoria.....	78
Fig. 6.13: Acciones de control para la ejecución de la primera trayectoria.....	79
Fig. 6.14: Ejecución experimental de la trayectoria definida.	80
Fig. 6.15: Ejecución experimental de la trayectoria definida.	80
Fig. 6.16: Errores de control para x,y,z del experimento realizado.	81
Fig. 6.17: Acciones de control para el Falcon en [cm/s].	81
Fig. 6.18: Ejecución de trayectoria para el segundo experimento	82
Fig. 6.19: Errores en cada una de las componentes para la ejecución de la segunda trayectoria.....	83
Fig. 6.20: Acciones de control para la ejecución de la segunda trayectoria.....	83
Fig. 6.21: Acciones de control para la ejecución de la segunda trayectoria.....	84
Fig. 6.22: Errores de control producidos por el control mediante la implementación del dispositivo háptico.	85
Fig. 6.23: Acciones de control para la ejecución de la segunda trayectoria.....	85
Fig. 6.24: Ubicación de archivos para el direccionamiento en Matlab	86
Fig. 6.25: Redirección de carpetas para la conexión con V-REP y Matlab.....	87
Fig. 6.26: Configuración de V-REP para seleccionar scripts	88
Fig. 6.27: Configuración del script para la simulación.....	88
Fig. 6.28: Selección del child script.....	89
Fig. 6.29: Configuración de la cámara para visualizar el entorno	89
Fig. 6.30: Adición de líneas de programación para la asignación de un puerto de comunicación	90
Fig. 6.31: Conexión satisfactoria entre la plataforma de Matlab y V-REP	91
Fig. 6.32: Configuración de la escena inicial para la primera aplicación.....	92
Fig. 6.33: Menú de objetos de V-REP.....	92
Fig. 6.34: Selección de paredes para la creación de la habitación.....	93

Fig. 6.35: Creación de la habitación simulada.....	93
Fig. 6.36: Selección de objetos para el entorno.	94
Fig. 6.37: Creación del entorno con obstáculos y objetos para la simulación.....	94
Fig. 6.38: Entorno de simulación para la navegación del robot diferencial	95
Fig. 6.39: Robot diferencia ubicado sobre el espacio de trabajo	95
Fig. 6.40: Robot diferencia Pioneer 3-DX integrado en la librería de V-REP	96
Fig. 6.41: Objetos robot para la simulación.....	97
Fig. 6.42: Incorporación del robot Pioneer 3-DX en la simulación	97
Fig. 6.43: Cámara para retroalimentación visual.....	98
Fig. 6.44: Habitación diseñada para el segundo escenario	99
Fig. 6.45: Selección de banda transportadora.	99
Fig. 6.46: Creación de la banda transportadora para las cajas.....	100
Fig. 6.47: Colocación de las cajas a manipular con el brazo robótico.....	100
Fig. 6.48: Manipulador en V-REP	101
Fig. 6.49: Ensamble de la escena con el manipulador robótico en V-REP	101
Fig. 6.50: Flujo de trabajo entre Matlab, V-REP y Novint Falcon.....	103
Fig. 6.51: Modelo de operación para manejar la plataforma móvil en un entorno semiestructurado	104
Fig. 6.52: Modelo de operación para manejar el manipulador móvil sobre un espacio estructurado	105
Fig. 6.53: Robot móvil ubicado en el punto de partida del entorno simulado....	107
Fig. 6.54: Error de control en el eje X dela tarea realizada.	107
Fig. 6.55: Error de control en el eje Y dela tarea realizada.	108
Fig. 6.56: Error de control en el eje Z dela tarea realizada.....	108
Fig. 6.57: Fuerzas de retroalimentación en X como respuesta entre la interacción robot-escenario	109
Fig. 6.58: Fuerzas de retroalimentación en Y como respuesta entre la interacción robot-escenario	109
Fig. 6.59: Fuerzas de retroalimentación en Z como respuesta entre la interacción robot-escenario	110
Fig. 6.60: Usuario ejecutando la prueba de experimentación.....	110

Fig. 6.61: Tendencia del error promedio de cada prueba realizada para la tarea planteada	112
Fig. 6.62: Ejecución de la tarea	113
Fig. 6.63: Errores de posición entre el robot y objetos cercanos en el eje X.....	114
Fig. 6.64: Errores de posición entre el robot y objetos cercanos en el eje Y.....	114
Fig. 6.65: Errores de posición entre el robot y objetos cercanos en el eje Z.....	115
Fig. 6.66: Fuerza de retroalimentación en el eje X.....	115
Fig. 6.67: Fuerza de retroalimentación en el eje Y.....	116
Fig. 6.68: Fuerza de retroalimentación en el eje Z.	116
Fig. 6.69: Usuario ejecutando la tarea para el segundo experimento	117
Fig. 6.70: Tendencia del error promedio de cada prueba realizada.....	118

ÍNDICE DE TABLAS

Tabla 3-1: Variable Independiente: Control de dispositivo háptico.	34
Tabla 3-2: Variable dependiente: Retroalimentación de Fuerzas	35
Tabla 4-1: Diferencias entre robots maestros hápticos y no hápticos.....	42
Tabla 4-2: Características de robots hápticos: 3D Haptic Stylus, Geomagic y Falcon.....	44
Tabla 4-3: Características de los software de simulación	46
Tabla 4-4: Generación de fuerzas de reacción dependientes de la aplicación	49
Tabla 6-1: Parámetros establecidos para encontrar los valores de PID	74
Tabla 6-2: Errores de las pruebas experimentales para ver desempeño del controlador.	111
Tabla 6-3: Errores de las pruebas experimentales para ver desempeño del controlador.	117

UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERIA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL / DIRECCIÓN DE POSGRADO**

MAESTRÍA EN AUTOMATIZACIÓN Y SISTEMAS DE CONTROL

TEMA:

**CONTROL DE UN DISPOSITIVO HÁPTICO PARA
RETROALIMENTACIÓN DE FUERZAS ENFOCADO A TELE-PRESENCIA**

AUTOR: ING. ROBERTO ASDRÚBAL SEGURA FLORES

DIRECTOR: ING. SANTIAGO ALTAMIRANO, MG.

FECHA: Agosto, 2018

RESUMEN EJECUTIVO

La propuesta de uso de un dispositivo háptico para mejorar la tele operación de robots se presenta en este trabajo de investigación. Teniendo como premisa las limitantes en la tele operación de sistemas robóticos, el estudio se centra en optar por dispositivos que suplementen a la retroalimentación visual (comúnmente usada), con el objetivo de tener una perspectiva más cercana del entorno en el cual el robot está trabajando. De esta manera, se analiza el tipo de escenarios en los cuales la robótica estructurada y semi-estructurada trabaja, de forma que se seleccione el sistema robótico maestro apropiado. La selección de un dispositivo dimensionado para este trabajo es llevada a cabo a través del análisis de características de cada uno de los dispositivos hápticos disponibles en el mercado, seleccionando el dimensionado a los objetivos de este proyecto. La selección permite analizar el comportamiento cinemático del dispositivo, el cual puede ser representado por un conjunto de ecuaciones matemáticas que permiten simularlo y ser sujeto de pruebas de diversos métodos de control. Para validar los métodos de control planteados para el dispositivo háptico, un conjunto de elementos ya sean físicos o simulados son requeridos, dentro de los cuales los segundos son seleccionados dada la carencia de financiamiento necesario. De esta manera, se plantea el uso de V-REP, un software que permite simular robots sobre espacios físicos configurables, considerando características físicas y mecánicas de los

sistemas. La conexión entre el software matemático (Matlab) y V-REP adicionalmente se plantea, considerando el análisis de control de cada uno de los robots en ambos escenarios. Finalmente, resultados experimentales se muestran para validar el controlador propuesto y la interacción entre el control y los escenarios virtuales.

Descriptores: Dispositivo Háptico, Tele operación, Tele presencia, Retroalimentación, Matlab, V-REP, Robot Paralelo, Inmersión, Robot móvil, Manipulador, Escenarios simulados.

UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL /DIRECCIÓN DE POSGRADO
MAESTRÍA EN AUTOMATIZACIÓN Y SISTEMAS DE CONTROL

THEME:

CONTROL OF A HAPTIC DEVICE FOR FORCES FEEDBACK FOCUSED
ON TELEPRESENCE

AUTHOR: ING. ROBERTO ASDRÚBAL SEGURA FLORES

DIRECTED BY: ING. SANTIAGO ALTAMIRANO, MG.

DATE: Agosto, 2018

EXECUTIVE SUMMARY

The proposal to use a haptic device to improve the teleoperation of robots is presented in this research work. Having as a premise the limitations in the teleoperation of robotic systems, the study focuses on opting for devices which supplement the visual feedback (commonly used), in order to have a closer perspective of the environment where the robot is working. In this way, the type of scenarios in which structured and semi-structured robotics works are analyzed so that the appropriate robotic master system is selected. The selection of a device sized for this work is carried out through the analysis of characteristics of each of the haptic devices available in the market, selecting the best one to comply with the objectives of this project. The selection allows analyzing the kinematic behavior of the device, which can be represented by a set of mathematical equations that allow simulation and testing of various control methods. To validate the control methods proposed for the haptic device, a set of elements, whether physical or simulated are required, being the second ones the selected given the unnecessary financing. In this way, the use of V-REP is proposed, a software which allows simulating robots on configurable physical spaces, considering the physical and mechanical characteristics of the systems. The connection between the mathematical software (Matlab) and V-REP is also included, considering the control analysis of each of

the robots in both scenarios. Finally, experimental results are shown to validate the proposed controller and the interaction between the controller and the virtual scenarios.

Keywords: Haptic device, Teleoperation, Tele presence, Feedback, Matlab, V-REP, Parallel Robots, Immersion, Mobile Robot, Manipulator, Simulated Scenarios.

INTRODUCCIÓN

Una solución a la problemática de teleoperación de robots móviles y con base fija en ambientes estructurados o semi-estructurados es estudiada en el presente trabajo de investigación. Basándose en trabajos relacionados, se plantea una opción que permite discriminar el entorno de trabajo del robot desde una perspectiva diferente al uso de retroalimentación visual, incluyendo retroalimentación por fuerzas. De esta manera, se determina el robot maestro más apropiado para llevar a cabo tareas tele operadas sobre robots esclavos, los cuales poseen características dinámicas muy similares a robots reales, pero en simulación. Partiendo de este precepto, se estudia el comportamiento cinemático del robot maestro para determinar un modelo matemático equivalente, el cual sirve para el planteamiento de controladores autónomos. Para conseguir la solución de los objetivos planteados, este trabajo se divide en subsecciones descritas a lo largo de todo este informe escrito, el cual incluye adicionalmente un resumen ejecutivo y bases bibliográficas.

El Capítulo I justifica el trabajo de investigación, partiendo de una perspectiva crítica de la problemática encontrada en la tele operación. De esta manera, se plantean preguntas que se buscan contestar a partir del desarrollo de esta investigación, finalizando en objetivos iniciales que permitirán incrementar la viabilidad en la solución del problema, delimitándolo para llegar a un trabajo conciso y justificándolo para incrementar la originalidad del mismo, sin perder de vista el soporte de trabajos futuros.

El Capítulo II analiza la bibliografía actualizada para determinar trabajos similares desarrollados en el ámbito local e internacional, de forma que el trabajo apunte a aportar significativamente en el campo de la investigación y sea usado para resolver problemas actuales y futuros. Para esto, se fragmenta el tema de investigación para determinar qué factores influyen en la solución, de manera que se puedan atacar todas las metodologías que podrían estar inmersas.

Por otra parte, el Capítulo III se centra en la metodología a ser usada en el trabajo desarrollado. De esta manera, se realizan tablas analíticas para formular preguntas puntuales como que tipos de dispositivos hápticos existen o como se puede

controlarlos, con el objetivo de encaminar el tema a una solución adecuada, dimensionada y limitada. Este capítulo finaliza planteando estrategias para la recolección de información bibliográfica, así como métodos para procesar adecuadamente los datos obtenidos.

El Capítulo IV resuelve las preguntas propuestas por el Capítulo III, de modo que la solución del problema se encamine de una manera apropiada. A través de esta sección, se contrastan diversos métodos, dispositivos e información bibliográfica para plantear una solución dimensionada y con recursos escalados, de forma que la solución no contenga elementos innecesarios.

El análisis de todo el estudio se denota en el Capítulo V, donde conclusiones y recomendaciones se muestran como el punto de partida a la solución propuesta. Las conclusiones describen la perspectiva de todo el análisis bibliográfico actualizado, mientras las recomendaciones muestran el inicio en la propuesta de solución.

El análisis de los capítulos anteriores converge en el Capítulo VI, el cual describe el sistema propuesto. Previo al desarrollo de la propuesta, el capítulo presenta una introducción de modelación de los dos sistemas robóticos usados para los experimentos y validación de controladores. El desarrollo de la propuesta incluye un análisis del software matemático, el método de comunicación y el desarrollo de los escenarios virtuales, finalizando con conclusiones del sistema propuesto.

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1. Tema de investigación

CONTROL DE UN DISPOSITIVO HÁPTICO PARA RETROALIMENTACIÓN DE FUERZAS ENFOCADO A TELE-PRESENCIA

1.2. Planteamiento del problema

1.2.1. Contextualización

Los más recientes avances de investigación tecnológica internacional orientan sus estudios a sistemas de control de alta autonomía (Li, 2017), es así, que la literatura presenta algoritmos de control de lazo abierto y lazo cerrado que permite asegurar el cumplimiento de los objetivos propuestos y son seleccionados según los requerimientos especificados (Santina, et al., 2017). Otro estudio tecnológico de gran importancia es el análisis de estabilidad en propuestas que consideran el comportamiento estático y dinámico de un sistema (Vilela, Figueredo, Ishihara, & Borges, 2016).

Por otra parte, la robótica es un área en creciente desarrollo, existiendo una gran variedad de especialidades dentro de este campo, desde la robótica industrial que trabaja en espacios totalmente estructurados, hasta la robótica de servicios, que pretende desenvolverse en espacios variantes en el tiempo. Otra forma de clasificar la robótica es según el tipo de robot, teniendo por un lado la robótica móvil (vehículos terrestres, aéreos y marítimos autónomos) y por el otro la robótica de manipuladores (mecanismos morfológicos que realizan tareas de manipulación).

Las aplicaciones de la robótica de manipuladores han permitido sustituir al ser humano en tareas de alta complejidad y riesgo. Así tenemos dos tipos de manipuladores autónomos: los robots seriales, que son cadenas cinemáticas abiertas con la herramienta de trabajo ubicada al final (Rajan, Yadav, Krishnan, Gupta, & Ghosh, 2017); y los robots paralelos, que son cadenas cinemáticas cerradas cuyo efector final está conectado a la base mediante varias cadenas cinemáticas seriales (Wang, Mao, Lin, Wei, & Zhang, 2017). Así, la robótica otorga a un sistema

mecánico cierta autonomía, permitiendo resolver problemas específicamente diseñados; pero, aún con toda la tecnología moderna, la presencia del ser humano sigue siendo necesaria en tareas muy definidas.

Cuando un robot no puede reemplazar las habilidades de criterio y manipulación de un ser humano, existe un área de la robótica especializada en trasladar la experticia de un operador a ciento o miles de kilómetros. La tele-robótica permite controlar un robot a distancia con gran precisión y brindando las herramientas de apoyo necesarias para el operador, para que esto sea posible, la tele-robótica se clasifica en dos grandes campos; la tele-operación encargada del control a distancia de los sistemas autónomos localizados en el sitio remoto, y la tele-presencia que hace posible replicar las sensaciones del sitio remoto en el sitio local (Islam, Dias, & Seneviratne, 2016).

Los sistemas de tele-presencia tienen niveles de inmersión, es decir, permiten experimentar una o más sensaciones, para ello requieren de dispositivos especializados que generen los comportamientos ocurridos en el sitio remoto. El sentido con mayor grado de inmersión es el tacto, debido a la información que provee al operador, por lo cual es importante el uso de dispositivos hápticos en sistema de tele-robótica y tele-presencia (Surendran & J, 2016).

En la actualidad existe un continuo desarrollo de dispositivos hápticos, que permiten al usuario trasladar el movimiento espacial de sus manos hacia un entorno real remoto o virtual 3D, además, permiten que un computador reproduzca sensaciones de tacto en el operador, estas sensaciones generalmente se relacionan con respuestas que ayudan a la interacción hombre-aplicación. Dichos dispositivos pueden ser utilizados para trasladar la experticia de un médico hacia una sala de cirugía a miles de kilómetros (Therriault, Nagurka, & Johnson, 2014), en control remoto de robots ubicados en lugares de riesgo y en video juegos de entrenamiento moderno.

Sin embargo, las aplicaciones generalmente limitan el funcionamiento del dispositivo háptico a actividades concretas, impidiendo la reutilización en otras tareas. El control de un dispositivo háptico considerado como un robot paralelo es requerido para ampliar y mejorar las aplicaciones tanto en el ámbito industrial como

el de servicio. El controlador del robot háptico elimina la problemática de retroalimentar fuerzas reales a ficticias con baja precisión, con lo cual la fiabilidad del sistema controlado puede incrementarse considerablemente.

1.2.2. Árbol del problema

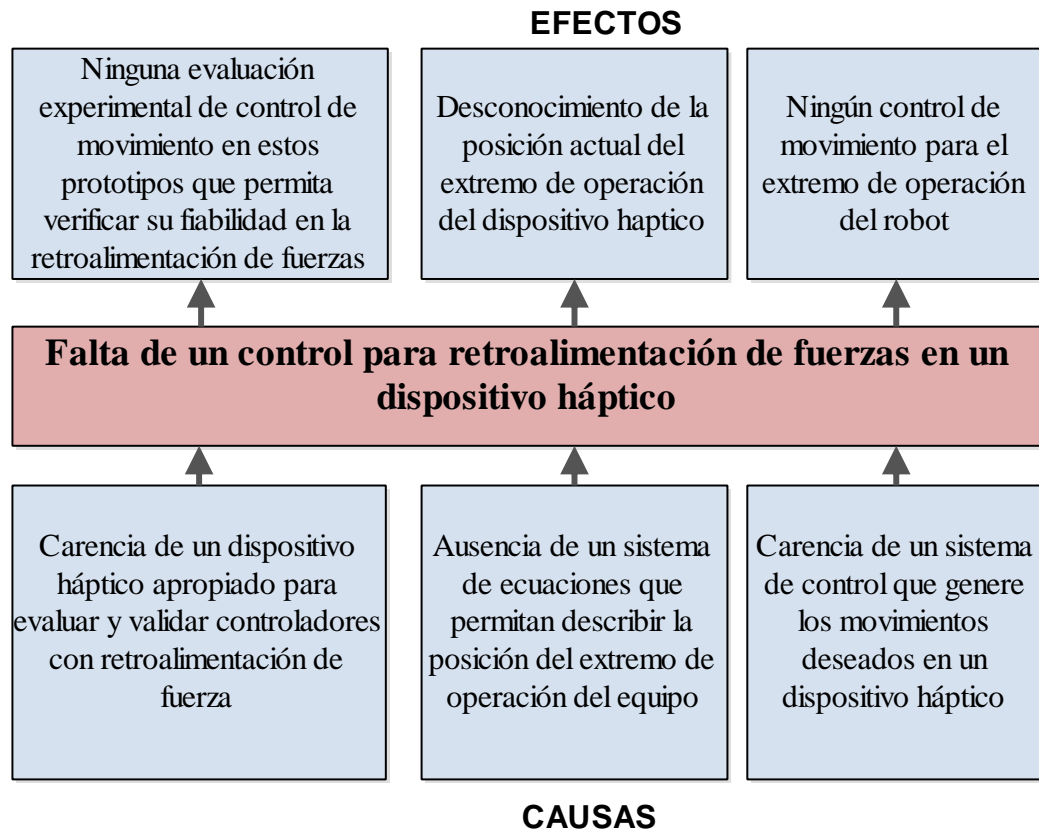


Fig. 1.1: Árbol del problema

Elaborado por: Roberto Segura

1.2.3. Análisis crítico

Los dispositivos hápticos son estudiados en sistemas de control bilateral debido a que además de proveer de entradas, son mecanismos que requieren señales de control para su funcionamiento. Esto implica la necesidad de usar encoders para detectar su posición actual, e instalar motores que permitan el movimiento y generación de fuerzas, proporcionando las características hápticas al equipo.

Un sistema de control a distancia tiene un alto grado de dificultad al momento de comandar el robot ubicado en el sitio remoto, esto por las limitaciones de

percepción presentadas en el sitio local, por lo cual mientras más información se provea al operador, mejor será su desempeño con la tarea a realizar.

Con el propósito de captar más información del entorno en donde un robot desempeña una tarea, dispositivos que retroalimenten fuerzas pueden ser incluidos en un sistema de control. Sin embargo, la selección del dispositivo apropiado es esencial para llevar a cabo esta tarea, ya que no todos permiten incluir algoritmos de control sobre el extremo operativo del robot paralelo. Como consecuencia, una evaluación experimental para retroalimentación de fuerzas se convierte en un trabajo dificultoso.

Dentro del control del robot paralelo, la matemática que describe su comportamiento dado un sistema de referencia fijo es comúnmente compleja. Esto es una limitante al momento de obtener la posición exacta del extremo operativo. Adicional al conocimiento de la posición del extremo operativo, un controlador que permite posicionarlo en puntos deseados es necesaria. Al no contar con un modelo matemático equivalente, ninguna acción de control puede ser implementada de manera adecuada.

1.2.4. Prognosis

Al no llevarse a cabo este trabajo investigativo, se limitará el desarrollo de herramientas de apoyo a sistemas de tele-operación que pueden servir de ayuda en tareas de búsqueda y rescate en incidentes catastróficos que pueden suscitarse en nuestro país.

Otra desventaja de no realizarse este proyecto es que se impedirá el avance de trabajos investigativos que aportan a la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

1.2.5. Formulación del problema

¿Se puede diseñar e implementar el control de un dispositivo háptico para retroalimentación de fuerzas?

1.2.6. Preguntas directrices

- ¿Qué dispositivo háptico se utiliza comúnmente en sistemas de tele-presencia?
- ¿Cómo se puede describir la posición del punto de operación en un dispositivo háptico?
- ¿Qué control se implementa para la retroalimentación de fuerzas en un dispositivo háptico?
- ¿Cómo se evalúa el control de un dispositivo háptico para retroalimentación de fuerzas?

1.2.7. Delimitación del problema

- **Límite del contenido**

Área Académica: Automatización y Sistemas de Control

Línea de investigación: Control y Programación de robots.

- **Límite espacial**

Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato.

- **Límite Temporal**

Este proyecto se realizará en un lapso de 6 meses a partir de la aprobación del Honorable Consejo Académico de la Facultad de ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato.

1.3. Justificación

El campo de la robótica está en constante crecimiento con aplicaciones que se encuentran en múltiples áreas militares y de servicio. En casos especiales, ciertas aplicaciones requieren que el agente controlador se encuentre ubicado en una locación alejada del agente efector. De esta manera, el control de un robot que ejecuta tareas que podrían resultar de peligrosidad o riesgo para el personal humano

se realiza mediante técnicas de teleoperación. La teleoperación implica controlar el robot mediante algún tipo de información de retroalimentación, la cual permite conocer el entorno de trabajo del sistema robótico.

Una de las formas de retroalimentación de información que presenta diversas ventajas es la generación de fuerzas en el mando de control local. A través del empuje de un efector final a la extremidad del usuario, el dispositivo usado para generar velocidades de control puede proveer información del medio donde trabaja el robot remoto. Sin embargo, generar fuerzas empíricamente no resolverá el problema de la manera más adecuada, requiriéndose un control que considere el comportamiento cinemático del dispositivo háptico para mejorar la retroalimentación de fuerzas.

En el campo aplicativo, desarrollar este tipo de herramientas de apoyo servirá como soporte para optimizar tareas tales como búsqueda y rescate en zonas de riesgos, así también, en el progreso de sistemas de operación quirúrgica realizadas por médicos expertos desde sitios lejanos, entre otros.

El control de un dispositivo háptico para retroalimentación de fuerzas enfocado a tele-presencia permitirá a nuestro país continuar su desarrollo tecnológico y científico; un aporte que servirá de apoyo al creciente progreso de la sociedad. Adicionalmente y con los conocimientos adquiridos en base a investigación científica, la generación de un antecedente en este campo de la tele-robótica permitirá influenciar a los centros de investigación nacionales e internacionales.

1.4. Objetivos

1.4.1. Objetivo general

Implementar el control de un dispositivo háptico usando movimientos deseados que permita la retroalimentación de fuerzas en el punto de operación.

1.4.2. Objetivos específicos

- Analizar los dispositivos hápticos comerciales mediante la investigación y comparación de diferentes prototipos para determinar el más adecuado para sistemas de tele-presencia.
- Describir el comportamiento cinemático del punto de interés en el dispositivo háptico usando un análisis matemático para determinar las características del robot paralelo a ser controlado.
- Diseñar el controlador de movimiento de un dispositivo háptico mediante una técnica de control para la retroalimentación de fuerzas.
- Realizar pruebas experimentales del control propuesto usando un prototipo comercial para evaluar el funcionamiento del sistema.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes investigativos

Analizando los repositorio de las universidades internacionales se encuentra el trabajo doctoral titulado “Asistencia háptica basada en planificación de movimientos para la teleoperación cooperativa de sistemas multirobot” desarrollado en la Universidad Politécnica De Catalunya (Ruiz, 2012), en el que se destacan, la realización de tareas de teleoperación con robots tiene mejoras usando un sistema de guiado que apoye al usuario en sus movimientos. La planificación se realiza en base a tareas y entornos de trabajo previamente conocidos, dándole al sistema un enfoque estructurado bastante común en teleoperación. El trabajo realiza un control bilateral, donde el robot industrial es manipulado mediante un dispositivo háptico de escritorio, que dirige al usuario a un camino libre de colisiones mediante generación de fuerzas que lo atraen. Este trabajo tiene un aporte significativo en cuanto al control del dispositivo háptico en retroalimentación de fuerzas para guiado, ya que sugiere movimientos al operador evitando colisiones del robot con elementos externos y con otros robots, además, el sistema posee un comportamiento reactivo, independiente de las fuerzas generadas, que amortigua los desplazamientos en el puesto del operador previniendo situaciones de posible colisión.

Otro trabajo magistral desarrollado a nivel internacional, se realiza en la Universidad de Chile con el tema “Sistema de Teleoperación Basado en Interfaz Háptico para Brazo Robótico” (Urrutia, 2016), donde se propuso nuevo sistema de teleoperación bilateral apoyado en el uso de un dispositivo háptico, el sistema interactúa una Point Cloud y un robot virtual en 3D. De esta tesis concluye que el método implementado permite a los usuarios mejorar su precisión al desarrollar tareas de seguimiento de trayectoria con efector final del robot, y al utilizar la retroalimentación de fuerzas se puede terminar la tarea más rápido, con menos movimientos de la posición del extremo operativo del robot y con un menor número de colisiones del robot con el entorno.

En investigaciones realizadas a nivel nacional, destaca el trabajo titulado “Teleoperación con Retroalimentación de Fuerzas Aplicado a un Robot Móvil con restricción no Holonómica” desarrollado en la Escuela Politécnica Nacional (Salas, 2016), obteniendo como resultados que los algoritmos diseñados permiten teleoperar un robot móvil en ambientes semiestructurados con evasión de obstáculos estáticos. Respecto al control del dispositivo háptico, su uso permite la realimentación de fuerzas como asistencia al operador para evitar obstáculos, y considera una fuerza mayor cuando más cercano está el obstáculo obligando al usuario a cambiar la dirección de movimiento.

El proyecto realizado localmente por la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato se titula “Diseño y Construcción de un Brazo Robótico Industrial Comandado Mediante un Sistema de Control Inalámbrico” (Molina, 2011), concluyendo que el control de las articulaciones del brazo robótico se realiza mediante las pulsaciones del control a distancia, diferenciando el tipo de movimiento por los colores de cada botón. La comunicación inalámbrica se efectuó usando tecnología infrarroja, permitiendo al operador comandar al robot a cierta distancia. Notándose que este proyecto no considera un dispositivo háptico para retroalimentación de fuerzas, siendo ese el propósito de esta investigación.

2.2. Fundamentación filosófica

Se fundamenta en un proyecto de investigación básica aplicada. A través de trabajos desarrollados en la misma línea, se plantea un enfoque orientado a la generación de nueva teoría que sirva como fundamento teórico para trabajos futuros.

2.3. Categorías fundamentales

2.3.1. Redes de inclusiones conceptuales

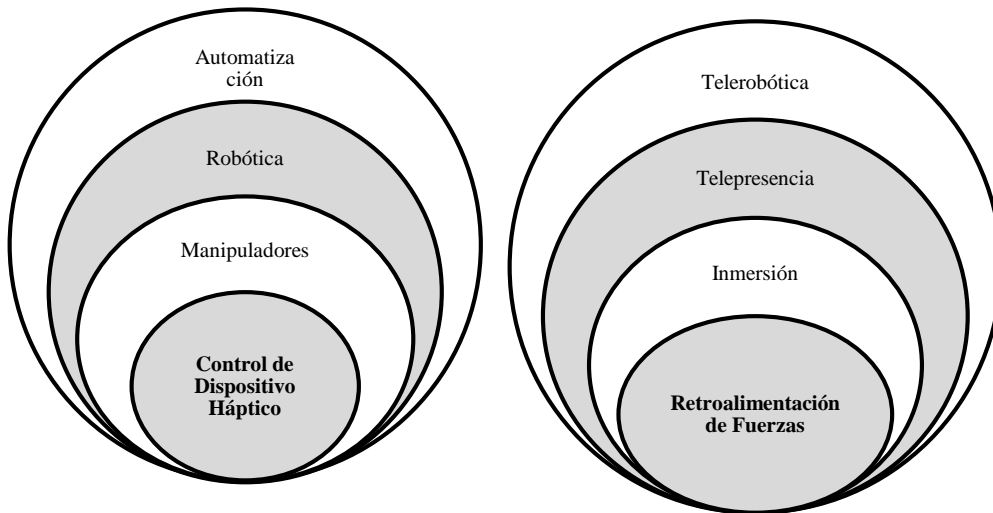


Fig. 2.1 Variables dependientes e independientes de las categorías fundamentales

Elaborado por: Roberto Segura

2.3.2. Constelación de ideas

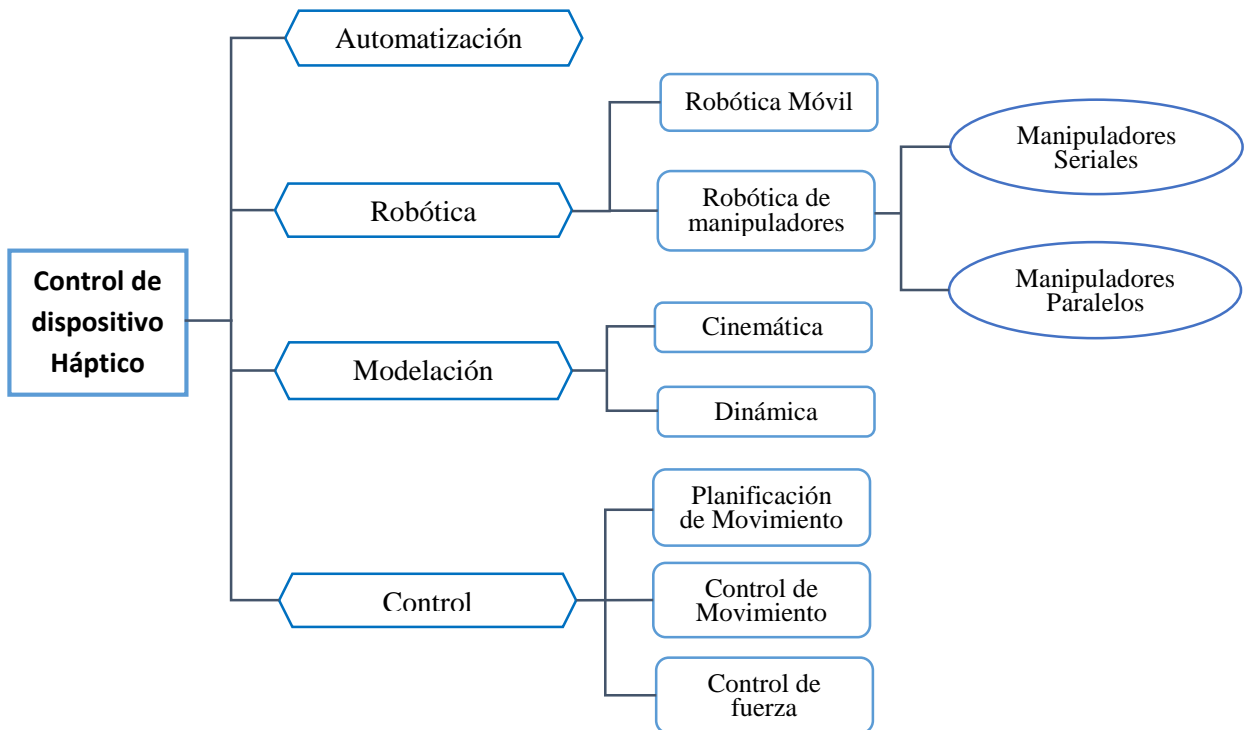


Fig. 2.2 Constelación de ideas de la Variable independiente

Elaborado por: Roberto Segura

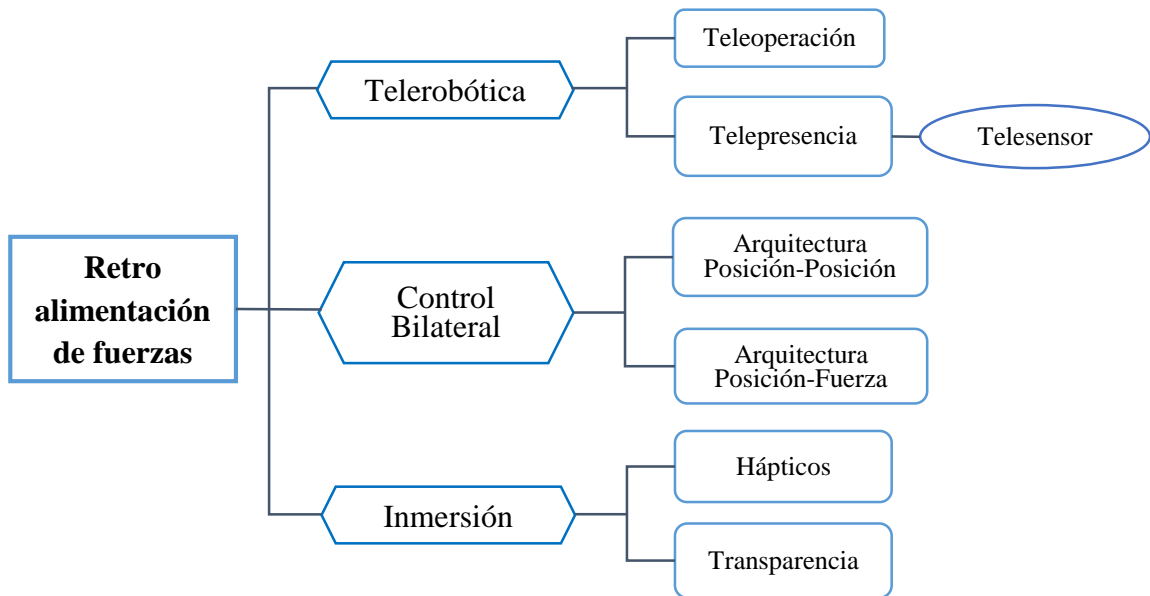


Fig. 2.3 Constelación de ideas de la Variable dependiente

Elaborado por: Roberto Segura

2.3.3. Automatización

La automatización es un sistema creado para ser usado en maquinarias, robots, entre otros; con el fin de ejecutar tareas determinadas que normalmente son efectuadas por el ser humano. Mejora la productividad, estándares de calidad e incluso la seguridad en los procesos industriales.

En la actualidad la automatización está presente en la robótica, incluso se emplea en la industria farmacéutica, petroquímica, mecánica, entre otras. Empleando así soluciones para los operarios que trabajan con sustancias peligrosas, materiales o actividades que afectan a la integridad humana (JOM, 2017). Con la complicidad de diferentes niveles de automatización, se asegura la competitividad y la inclusión de robots proporcionando a nivel industrial ventajas en la producción con beneficios económicos y de muy buena calidad.

La incorporación de la robótica ha producido nuevas oportunidades para la automatización a nivel industrial. Con el uso de robots se generan beneficios en la flexibilidad, seguridad y protección a los operadores en los diferentes ambientes peligrosos de trabajo. Los robots pueden ejecutar tareas de alta complejidad disminuyendo pérdidas por problemas de coordinación, fallos, cansancio o de

planificación, incluso la robótica surge repetitividad, efectividad elevando la producción (CONACYT, 2017).

2.3.4. Robótica

Durante las últimas décadas se han desarrollado grandes avances en el campo de la automatización de procesos, debido a la implantación de control por computadoras. Con la robótica industrial se ha conseguido incrementar el avance productivo, desarrollando tareas con mejor precisión, rápidas y de manera repetitiva, colaborando de esta manera en ambientes hostiles para el personal humano.

La implementación de los robots en ambientes industriales o a nivel de servicio, crean grandes ventajas con la elevada variedad de prototipos autónomos o teleoperados, cada uno de ellos desarrollados para realizar trabajos específicos o robots que son programables para realizar diferentes tareas. En la Fig. 2.4 se indican las generalidades más relevantes de la robótica en sus diferentes campos de aplicación (Funes, 2017).

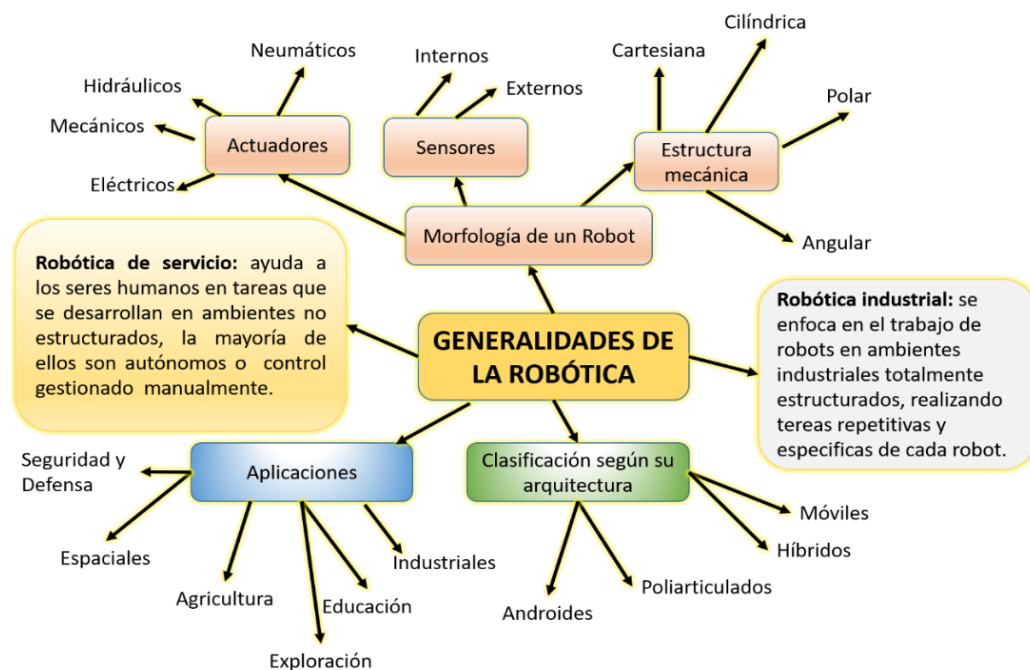


Fig. 2.4 Generalidades de la robótica.

Elaborado por: Roberto Segura

A. Robótica móvil

La robótica móvil emplea el uso de robots móviles con características de desplazarse en cualquier ambiente de trabajo sea este estructurado o no. La característica de estos robots es su autonomía para ejecutar tareas de reconocimiento, transporte o ejecución de funciones de navegación. Con estas características se han desarrollado prototipos robóticos que tienen incorporados sensores para determinar la presencia de objetos y obstáculos durante su tarea de navegación, impidiendo que este colisione y logrando así cumplir con su objetivo final planificado, Fig. 2.5.

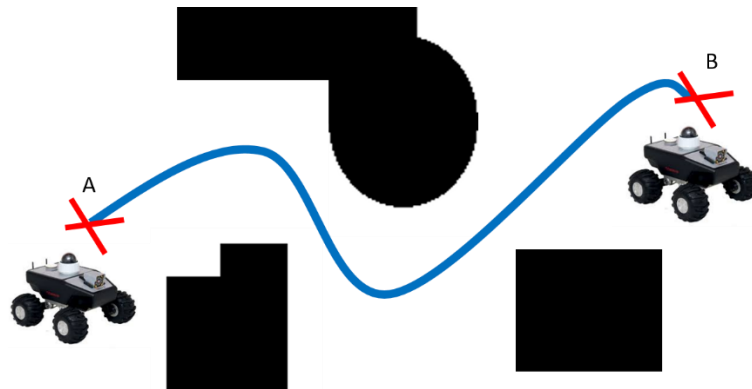


Fig. 2.5 Desplazamiento autónomo de Robots Móviles.

Elaborado por: Roberto Segura

Los robots móviles cuentan con sistemas que los ayudan a desplazarse, como sistemas de automoción, ruedas, patas, orugas o hélices, permitiendo que estos logren moverse de un lugar a otro. Son controlados manualmente a una determinada distancia, o de forma autónoma empleando los datos recopilados por los sensores incorporados (Ollero, 2001).

Dentro de los diferentes robots móviles (Fig.2.6) usados actualmente se cuentan con los siguientes:

Androides: robots idénticos a la estructura humana desplazándose en dos extremidades inferiores.

Zoomorfos: este tipo de robots imitan a los diferentes seres vivos, estos pueden ser por medio de diversas patas o a su vez que no posean ninguna, sino desplazándose de igual forma que lo reptiles.

Existen también robots móviles que empleen otro sistema de locomoción como son las ruedas.

Los robots móviles que cuentan con sistema de locomoción por medio de ruedas tienen diferentes clasificaciones, estos dependen de su configuración y del tipo de ruedas con que estén diseñados.

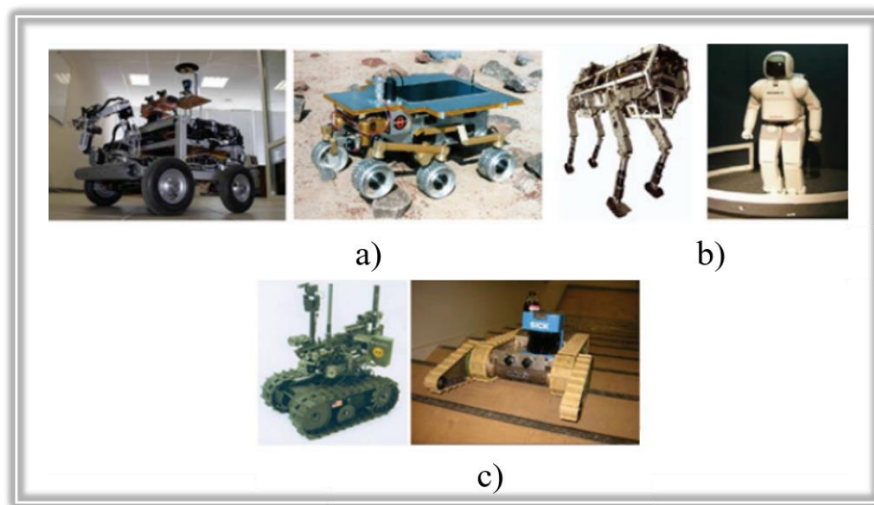


Fig. 2.6 Tipos de robots móviles: a) Robots con ruedas, b) Robots con patas y c) Robots tipo Oruga.

Fuente: Robots Móviles: Evolución y Estado del Arte (Barrientos Sotelo, 2007).

B. Robótica de manipuladores

• Manipuladores seriales

Los manipuladores (Fig. 2.7) son usados para realizar trabajos y operaciones de manejo de objetos que resulten peligrosos, incómodos o incluso muy pesados para las personas. Los manipuladores seriales son aquellos que están formados por barras rígidas que están unidas entre sí por medio de articulaciones de un grado de libertad (GDL), ya sea de tipo rotacional o prismático, similares al brazo humano. Se caracterizan por contar con una gran rigidez estructural, por su peso propio y su repetibilidad.

Los manipuladores seriales son empleados para ejecutar tareas eficaces repetitivas y de muy buena precisión, utilizando siempre su cinemática para el posicionamiento y la orientación del punto de interés del robot en este caso del efector final (González, Caro, & González, 2015).

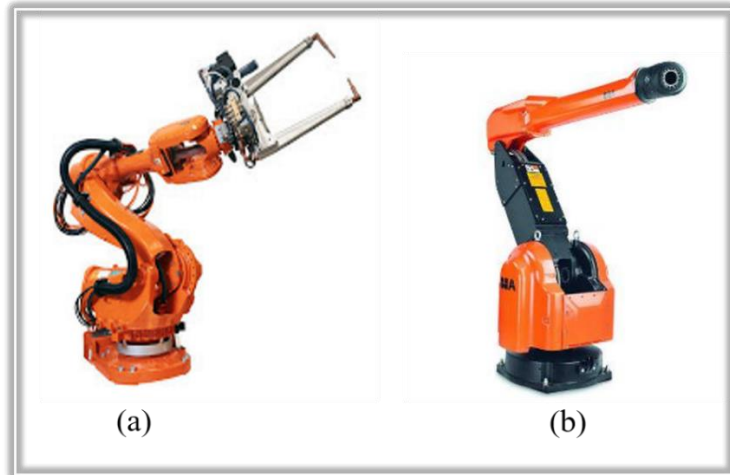


Fig. 2.7 Manipuladores Seriales, (a) KUKA con Pinza y(b) KUKA paletizador.

Fuente: Robots Industriales (Ariel, 2017).

- **Manipuladores paralelos**

Los manipuladores paralelos (Fig. 2.8) son robots con mecanismo de cinemática cerrada, en otras palabras, están conformados por una plataforma móvil unida por varias cadenas cinemáticas que son independientes.

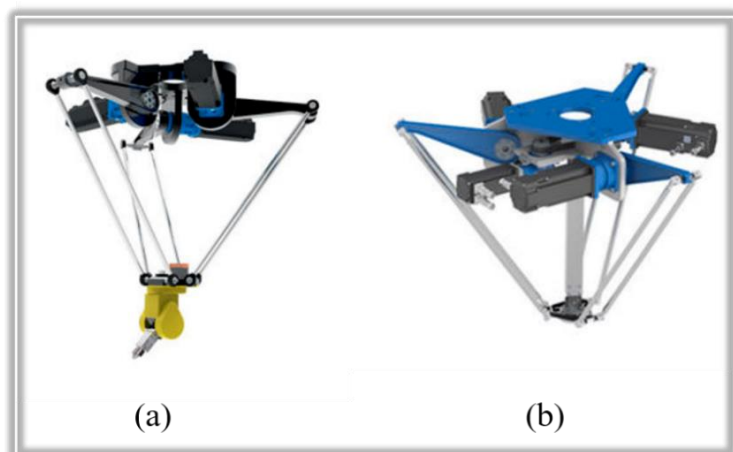


Fig. 2.8 Manipuladores paralelos, (a) Robot DELTA de 3 ejes y (b) Robot DELTA de 4 ejes.

Fuente: Robots Paralelos (directindustry, 2017).

Con la integración de una plataforma móvil y un manipulador; se tiene ventajas como es la rigidez, precisión, velocidad, mejor estabilidad, peso ligero, capacidad de manipular cargas relativamente grandes e inercia en el movimiento, Fig. 2.9. En el estudio de los robots paralelos no solo se puede utilizar un análisis cinemático, también se incorpora el análisis del comportamiento del movimiento. Adicionalmente se debe tomar en cuenta las singularidades presentes en el robot (configuraciones en la que el robot puede ganar grados de libertad, pero perdiendo el control del mismo) (Milenio.com, 2017).



Fig. 2.9 Aplicaciones de los robots paralelos.

Elaborado por: Roberto Segura

La clasificación de los robots paralelos viene dada por:

- Planares
- Esféricos
- Espaciales
- De acuerdo a su estructura, existen los simétricos y asimétricos.

Para el diseño de un robot paralelo se empieza por dos fases:

Síntesis Estructural: determinar la estructura mecánica general del robot

Síntesis Geométrica: encontrar el valor de los parámetros geométricos involucrados en la estructura.

2.3.5. Modelación

La modelación es el estudio del comportamiento de un robot, considerando el modelo matemático se toma en cuenta las propiedades cinemáticas y dinámicas que cuenta el robot. Según sus propiedades cada robot contara con un modelo matemático y cada uno de ellos tendrá una función diferente de acuerdo a lo que el usuario necesite observar o analizar.

A. Modelación Cinemática

La cinemática estudia los movimientos de un robot sin importar las causas que los producen, todo este estudio se lo hace con respecto a un sistema de referencia.

Para la modelación de un robot se emplea la cinemática directa o inversa de acuerdo al requerimiento del usuario.

- **Cinemática Inversa**

La solución al problema cinemático inverso puede ser sencillo para los robots paralelos. Para ello se puede considerar como ejemplo la plataforma Gough (Hernández-Martínez, Valdez-Peña, & Sánchez-Soto, 2013). La cinemática inversa consiste en determinar en encontrar las longitudes de las articulaciones para una determinada posición de la plataforma dentro de un marco de referencia fijo y a la vez una matriz de rotación que representa la orientación con respecto al mismo marco de referencia fijo.

- **Cinemática Directa**

Consiste en determinar la posición de la plataforma para un conjunto dado de coordenadas articulares. La cinemática directa se emplea para realizar control, calibración y planificación de movimientos (Siciliano & Khatib, 2008).

El problema cinemático directo de un robot paralelo suele ser mucho más complejo que la cinemática inversa, ya que las ecuaciones que cierran el lazo suelen ser expresiones no lineales, es por esto que generalmente se tiene un conjunto de ecuaciones no lineales. Por ejemplo, la plataforma Gough cuenta con cuarenta soluciones.

Se tiene varios métodos para el análisis del problema cinemático directo: por eliminación, continuación, bases de Gröbner y análisis de intervalos. La modelación por eliminación generalmente no es muy estable, es decir puede producir soluciones falsas. La continuación polinómica, es mucho más estable numéricamente.

B. Modelación Dinámica

En tareas que se requiere altas velocidades o transporte de cargas pesadas, se considera el modelo dinámico para usarlo en la ley de control, este permite aprovechar al máximo la potencia del robot sin perder precisión en el objetivo trazado al robot.

Para el modelo dinámico se consideran los efectos producidos por la masa de la estructura en el comportamiento del robot. La modelación dinámica de los robots paralelos es similar a los robots manipuladores. El método clásico para calcular el modelo dinámico de mecanismos de cadena es considerar primero una estructura arbórea equivalente y después aplicar restricciones cinemáticas empleando multiplicadores de LaGrange o aplicando el principio de d'Alembert.

Los inconvenientes de los modelos dinámicos obtenidos de los robots paralelos son demasiado complejos, requieren de parámetros dinámicos que a menudo no se conocen bien y se aplica soluciones de cinemática directa. Por lo cual se emplea computadoras sofisticadas para realizar el cálculo en tiempo real (Siciliano & Khatib, 2008).

2.3.6. Control en robótica

El control en robótica es muy esencial, ya sea cualquiera la estrategia empleada, siempre y cuando cumpla con los objetivos que debe cumplir el robot para ejecutar una tarea propuesta por el usuario.

Existen muchas estrategias de control para robots, en especial para sistemas no lineales; ya que la característica de los robots paralelos es que son no lineales.

La lógica de control es la que determina el flujo de ejecución, decide que tareas han de desarrollarse en cada instante y según la reacción del sistema y del entorno

generar acciones para que el robot las ejecute. Existen varias técnicas de control empleadas en la robótica, entre ellas se encuentra control cinemático, control dinámico y a la vez dentro de estos métodos existen tácticas de control a emplearse (Siciliano & Khatib, 2008).

A. Planificación de movimiento

Con la evolución de la robótica se han desarrollado robots autónomos capaces de decidir su propio movimiento con el objetivo de realizar una determinada tarea. Para ejecutar dicha tarea el robot debe ser capaz de encontrar la forma óptima de cumplirla evitando cualquier inconveniente dentro del espacio de trabajo del robot.

La planificación de movimiento de un robot se basa en el desarrollo de algoritmos geométricos eficientes que doten al robot la capacidad de generar soluciones óptimas en el espacio de trabajo, evitando así colisiones o mal ejecución de la tarea. La planificación de movimiento depende la configuración del robot ya que algunos robots cuentan con restricciones de movimiento.

En la Fig. 2.10 se representa los diferentes esquemas para planificación de movimientos de un robot (Siciliano & Khatib, 2008).

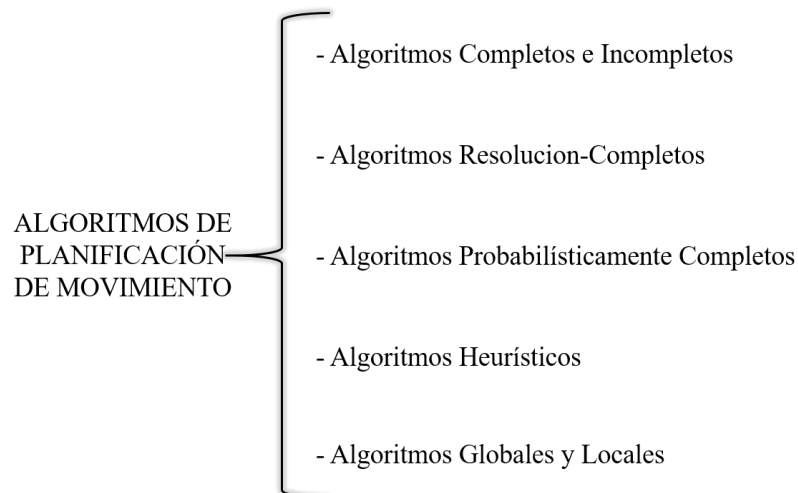


Fig. 2.10 Esquemas de planificación de movimiento.

Elaborado por: Roberto Segura

B. Control de movimiento

Para el control de movimiento, existen varias comparativas en la tarea a ejecutarse por un robot. El seguimiento de trayectoria tiene como objetivo seguir una trayectoria de referencia en un tiempo especificado dentro del espacio de trabajo del manipulador. Se supone que dicha trayectoria deseada cumpla con la capacidad de los actuadores, es decir, las velocidades y la aceleración no deberían sobrepasar el límite máximo de velocidad y aceleración permitido por la configuración y características del manipulador.

El posicionamiento conocido también como control punto a punto especifica una configuración fija en el espacio de trabajo. El objetivo de esta tarea es mantener el efector final del robot fijo en el espacio, pese a las perturbaciones de torque e independientemente de las condiciones iniciales.

Para ejecutar la tarea deseada, por ejemplo, las tareas que únicamente requieran que el manipulador se mueva de una posición a otra sin requerir una precisión relevante durante el movimiento entre la posición inicial y la final pueden ser resueltas por el control punto a punto, mientras tanto tareas como soldadura, pintura, corte y entre otras, requieren controladores de seguimiento (Siciliano & Khatib, 2008).

C. Control de fuerza

En los últimos años, la investigación sobre el control de fuerza ha evolucionado motivando así el deseo general de proporcionar sistemas robóticos con características sensoriales mejoradas. Se espera que los robots que emplean la fuerza, el tacto, la distancia, la retroalimentación visual y de fuerza, trabajen de manera autónoma en entornos no estructurados (A. F. Brumovsky).

El control de fuerza genera robustez y versatilidad en los sistemas robóticos de entornos abiertos, generando una respuesta inteligente en situaciones imprevistas y mejorando la interacción humano-robot. Las estrategias empleadas para resolver el control de fuerza son:

- Control de interacción pasiva.
- Control de interacción activa.
- Mediciones de fuerza.

Control de interacción pasiva: La trayectoria del efector final del robot se ve modificada por las fuerzas de interacción debidas al cumplimiento inherente del robot. El cumplimiento puede deberse a la conformidad estructural de los enlaces, uniones y efector final, o al cumplimiento del servo de posición.

Control de interacción activa: El cumplimiento del sistema robótico se garantiza principalmente mediante un sistema de control diseñado a propósito. Este enfoque generalmente requiere de la medición de la fuerza de contacto y el momento, que se realimentan al controlador y se usan para modificar o incluso generar en línea la trayectoria deseada del efector final del robot.

Mediciones de fuerza: Para una tarea general controlada por fuerza, se requieren seis componentes de fuerza para proporcionar información necesaria sobre la fuerza de contacto: tres componentes de fuerza de traslación y tres pares de torsión. Un sensor de fuerza está ubicado en la muñeca del robot, en la punta de los dedos o de manos robóticas; incluso, las fuerzas y los momentos externos se pueden estimar a través de mediciones de torque.

2.3.7. Telerobótica

La telerobótica es uno de los aspectos iniciales de la robótica. El primer concepto que aparece cuando se habla de tele robótica es relacionar a un robot con un controlador humano cerrando un lazo de control Fig. 2.11. En un nivel explicativo más avanzado, la planeación de movimiento y las decisiones lógicas son tomadas por el operario humano, mientras que el robot ejecuta una tarea física en un área lejana que podría contener un escenario peligroso para el operador. En resumen, el controlador principal o cerebro del proceso está alejado del cuerpo ejecutor (Siciliano & Khatib, 2008).

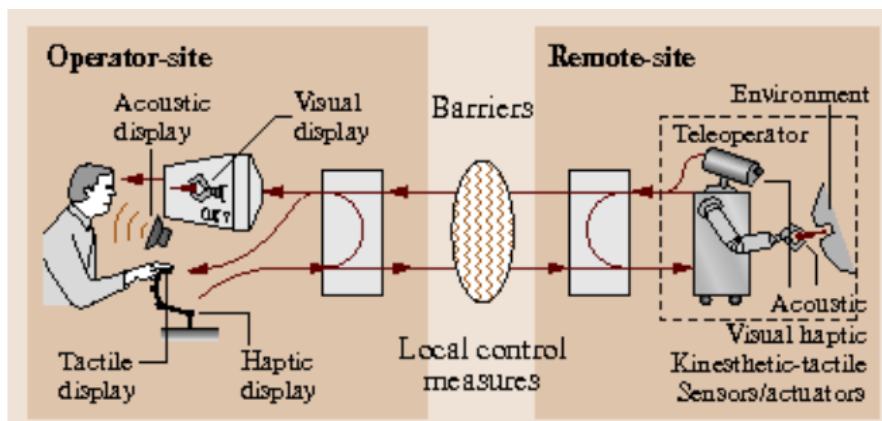


Fig. 2.11 Control remoto de un robot en el entorno.

Fuente: Telerobótica (Siciliano & Khatib, 2008)

El concepto de telerobótica incluye el término que describe la distancia existente entre el /controlador y el sistema controlado. La palabra tele (derivado del griego distante) describe la barrera que separa un operador de un mecanismo ejecutor de tareas cuando el usuario humano no puede desplazarse a un lugar que el robot puede alcanzar sin mayor dificultad.

Sin necesidad de que el operador se encuentre muy lejano al robot, los sistemas tele robóticos se encuentran por lo menos separados en dos sitios: el operador humano y todos los elementos necesarios que ayudan a la comunicación con el robot: joysticks, sensores, monitores, sistemas de comunicaciones y demás dispositivos de entrada/salida.

A. Teleoperación

La inclusión de un operario humano a un sistema robótico ubicado a distancia se muestra como una alternativa muy atractiva a la aplicación de cualquier tipo de robots en ambientes estructurados y no estructurados. En efecto, las aplicaciones para este tipo de configuraciones son diversas, desde enfoques en búsqueda y rescate, sistemas médicos y métodos de rehabilitación.

Los robots en un sistema de tele operación son normalmente denominados maestro y esclavo, donde el maestro es el ubicado en el sitio local y el esclavo se encuentra en el sitio remoto. Con el objetivo de proveer control directo, el robot esclavo es programado para seguir los movimientos generados por el robot maestro. El robot

maestro es a su vez controlado por el usuario, el cual puede ser un joystick, un teclado o algún otro mecanismo. En este aspecto, no es extraño que se tenga en el maestro una réplica del esclavo, generando de alguna manera una interface completamente intuitiva.

Dependiendo de la aplicación, algunos sistemas maestro-esclavo proveen fuerzas de retroalimentación, por tanto, el robot maestro no solo permite inyectar velocidades al robot teleoperado sino también generar fuerzas hacia el usuario. De esta manera, la interface de usuario se convierte en bidireccional y dicho sistema tele robótico es denominado bilateral (Emmanuel Nuño Ortega, 2004).

B. Telepresencia

La telepresencia es comúnmente descrita como el último objetivo de los sistemas maestro-esclavo y la telerobótica en general. La telepresencia provee al humano no solo la habilidad de manipular el entorno remoto, sino además percibir el ambiente de la misma manera en cómo se encuentra (Fig. 2.12). En este caso, el operador humano está provisto con la suficiente cantidad de información de retroalimentación y sensaciones para sentirse presente en el sitio remoto. Esto combina la modalidad háptica con otras modalidades que permiten proveer información a los sentidos de visión, escucha u otros más avanzados para oler y probar. El sistema maestro-esclavo se convierte en el medio por el cual el usuario interactúa con el ambiente remoto e idealmente se tiende a tratar de que el medio sea olvidado completamente. Si esto se logra, se dice que el sistema es totalmente transparente (Sinclair, 2017).

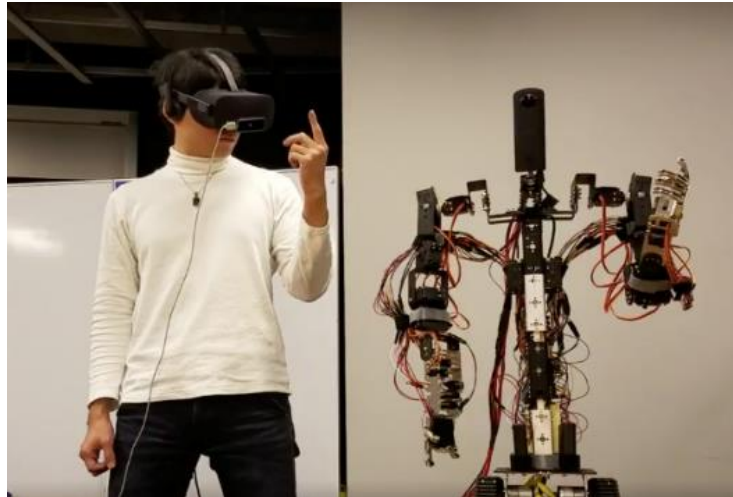


Fig. 2.12 Telepresencia con robots.

Fuente: Telepresence Robot Uses RICOH THETA (Casman, 2017)

C. Arquitecturas de control

A diferencia de los sistemas robóticos habituales, los sistemas telerobóticos proveen información al usuario y al mismo tiempo requieren comandos de control del mismo. Las arquitecturas de control en los sistemas telerobóticos pueden ser descritos por el estilo y el nivel de su conexión, como se muestra en la Fig. 2.13.

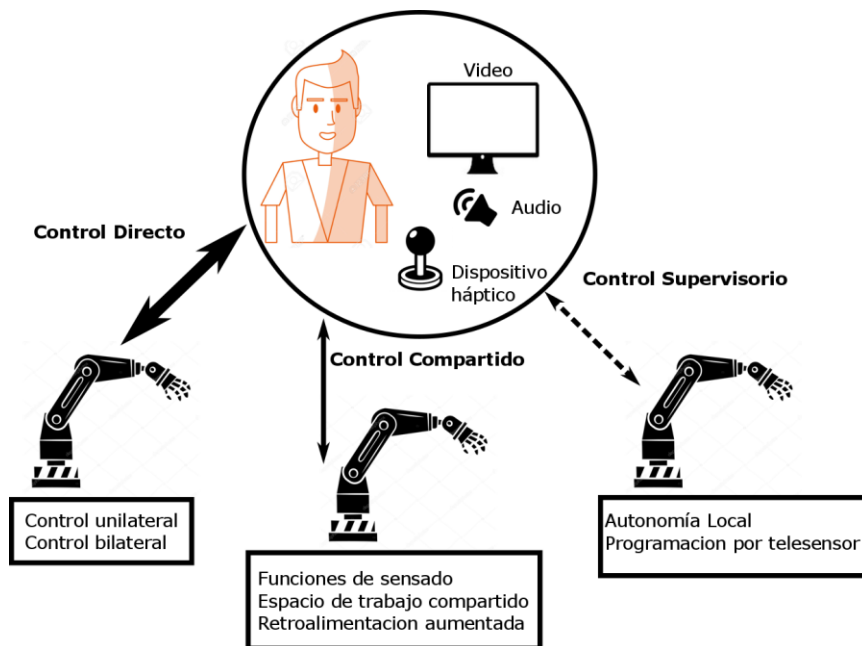


Fig. 2.13 Arquitecturas de control.

Elaborado por: Roberto Segura

En las arquitecturas de control, el control directo implica que no exista inteligencia o autonomía en el sistema dado que los movimientos vistos en el robot esclavo están directamente controlados por el usuario vía la interface de comunicación. A diferencia, si la ejecución de la tarea se basa en control compartido, la tarea usa el control directo y autonomía local a través de información provista por los sensores. Finalmente, en el control supervisorio, el usuario y el esclavo están conectados pero el robot incluye una autonomía local completa, de manera que el operario ejecuta comandos de muy alto nivel.

D. Aplicaciones de la telerobótica

Los sistemas telerobóticos buscan reemplazar al personal humano en la ejecución de tareas peligrosas en ambientes hostiles (locaciones luego de desastres nucleares), ambientes difíciles o costosos de llegar (espacio), ambientes a muy baja escala (cirugía invasiva), entre otros. Proteger al operador humano ha sido uno de los principales objetivos de la telerobótica, donde la industria química y nuclear ha sido las que más han usado este tipo de mecanismos. Asimismo, algunos sistemas han sido desarrollados para el mantenimiento de líneas de alta tensión o para la desactivación de explosivos (Fig. 2.14).



Fig. 2.14 Uso de la telerobótica para desactivación de explosivos.

Fuente: Robot de eliminación de Bombas (Allison, 2016)

2.3.8. Control bilateral y retroalimentación de fuerzas

La retroalimentación de fuerzas nace dada la necesidad de incrementar el rendimiento en la ejecución de la tarea y en búsqueda de mejorar los efectos de la telepresencia (Fig. 2.15). La naturaleza bilateral de esta configuración hace que la arquitectura de control sea particularmente desafiante ya que se forman múltiples lazos de retroalimentación e incluso sin contacto con el entorno o la intervención del usuario, los dos robots forman un circuito cerrado interno. Comúnmente, las comunicaciones entre los dos sitios incluyen retardos, dejando la estabilidad de control como un reto particular a solucionar (Siciliano & Khatib, 2008).

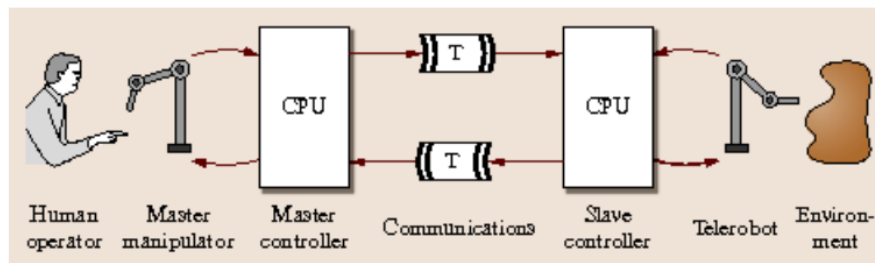


Fig. 2.15 Esquema de control con retroalimentación de fuerzas.

Fuente: Control bilateral y retroalimentación de la fuerza (Siciliano & Khatib, 2008)

Los problemas de estabilidad se solucionan alternando la información transmitida, en estos casos, el audio e información de fuerzas. Sin embargo, la combinación de métodos vibro táctiles con retroalimentación explícita de fuerzas puede incrementar las sensaciones y beneficios sobre el usuario. En esta sección se presentan los tipos de retroalimentación de fuerzas: arquitectura posición - posición y arquitectura posición-fuerza.

A. Arquitectura Posición – Posición

Esta es la manera más simple de arquitectura, donde ambos robots están instruidos en seguir el uno al otro. En este caso, si las ganancias de posición y velocidad son las mismas, las dos fuerzas son las mismas y el sistema retorna fuerzas de retroalimentación adecuadamente. Para propósitos de análisis, esto puede ser interpretado como un sistema resorte-amortiguador entre los dos tipos de robots, como se muestra en la Fig. 2.16.

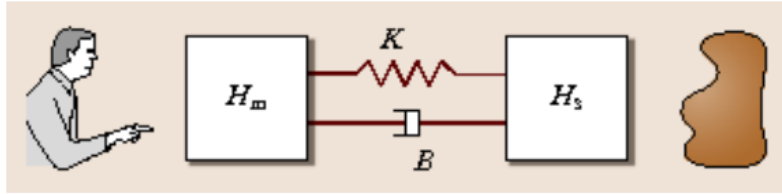


Fig. 2.16 Representación de la arquitectura Posición – Posición.

Fuente: Control de posición / fuerza (Siciliano & Khatib, 2008)

Esta arquitectura hace que el usuario pueda sentir las fuerzas de control del esclavo, que incluyen las fuerzas asociadas con el resorte-amortiguador y la inercia del esclavo robótico con la fuerza real del medio ambiente. En el caso en el que los movimientos se den sin contacto en el ambiente, el usuario sentirá la inercia y otras fuerzas dinámicas propias del esclavo.

B. Arquitectura Posición – Fuerza

En la arquitectura posición – posición, al usuario se le presenta la fuerza de control del esclavo, incluyendo la fricción y la inercia dada por el robot esclavo. Sin embargo, este efecto no es deseado en algunos casos. Con el objetivo de evadir este problema, las arquitecturas de posición – fuerza colocan un sensor de fuerza en la punta del robot esclavo a fin de retroalimentar la fuerza correcta vista desde el extremo operativo.

De esta manera se permite que el usuario solo reciba las fuerzas actuantes entre el esclavo y el ambiente, presentando solo una percepción clara del entorno de trabajo. La principal desventaja de esto es que la arquitectura es menos estable ya que el lazo de control pasa las órdenes del maestro al esclavo, luego las fuerzas del ambiente se retroalimentan hacia el maestro, presentándose un retardo en el control del esclavo. Mientras tanto, la ganancia del bucle puede ser muy alta: un comando de movimiento pequeño puede convertirse en una gran fuerza si el esclavo está presionando contra un entorno rígido. En combinación, la estabilidad puede verse comprometida en contacto rígido y muchos sistemas presentan inestabilidad de contacto en estos casos (Siciliano & Khatib, 2008).

2.3.9. Hápticos

Hápticos es el conjunto de estudios para generar experiencia y sensaciones de presencia en operadores humanos. Una manera de mejorar el rendimiento de sensaciones hápticas en operadores es generar un conjunto de sensaciones como si el operario estuviera en contacto con el entorno de trabajo.

Las interfaces hápticas buscan replicar o mejorar la experiencia al manipular elementos del ambiente de trabajo a través de dispositivos mecatrónicos y control por computadora. A modo general, los pasos para replicar las fuerzas se pueden resumir en tres partes: i) el dispositivo háptico percibe la señal de entrada del operador, la cual puede ser una posición, fuerza, actividad muscular, entre otros. ii) la señal receptada es aplicada a un ambiente virtual o teleoperado. En la teleoperación, un manipulador que está ubicado en una locación remota, intenta seguir los movimientos generados por la entrada provista por el operador. Finalmente, iii) los actuadores del dispositivo háptico son usados para físicamente convertir sensaciones hacia el operador humano.

La Fig.2.17 muestra el modo de trabajo de una interface háptica, donde la entrada que se generan a través de fuerzas hacia el sensor ingresa al ambiente remoto o simulado. Dependiente del ambiente de trabajo, los actuadores retornan fuerzas equivalentes hacia el operador humano. El resultado ideal de este proceso es que el operador humano sienta que en realidad esta interactuando con el ambiente real.

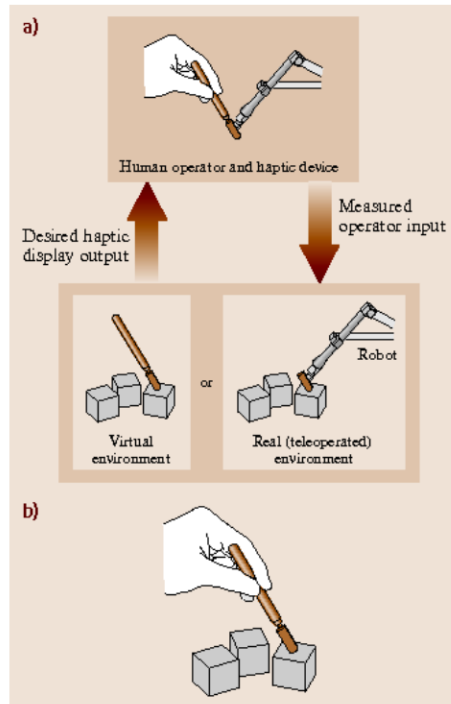


Fig. 2.17 Dispositivos Hápticos: (a)Entrada de señales hacia el entorno de trabajo, b) retorno de fuerzas para emular la interacción con el ambiente.

Fuente: Manipulación de dispositivos Hápticos (Siciliano & Khatib, 2008).

A. Transparencia

La transparencia en un sistema telerobótico es el mayor logro que se desea conseguir (Wortham & Theodorou, 2017). Un sistema telerobótico es transparente si el operador humano siente como si estuviera interactuando directamente con la tarea remota. Formalmente, la transparencia se alcanza solo si las impedancias del entorno de trabajo y las impedancias del control principal compaginan totalmente, es decir, las fuerzas de reacción del entorno de trabajo son exactamente las mismas que las receptadas por el operador humano. En un caso práctico, la transparencia no es alcanzable debido a que la inercia y la fricción no pueden ser completamente canceladas por el control. Adicional a esto, los efectos de la comunicación (especialmente el tiempo de retardo) degradan directamente la transparencia deseada (Ferre, Buss, Aracil, Melchiorri, & Balaguer, 2007).

B. Aplicaciones

El dispositivo háptico más fácil de encontrar es una pantalla de vibración encontrada comúnmente en video juegos. Aunque las interfaces hápticas no están aún dispersadas en el mercado a gran escala con fines de entretenimiento, se han integrado a nivel investigativo un gran número de ellas. Entre otras, un conjunto de aplicaciones novedosas que usan este tipo de dispositivos son: tecnología de asistencia, diseño, simulación médica, prótesis, rehabilitación, etc

2.4. Hipótesis

Es posible desarrollar el control de un dispositivo háptico para la retroalimentación de fuerzas enfocado a la tele-presencia.

2.5. Señalamiento de variables de la hipótesis

Variable independiente: Control de dispositivo háptico

Variable dependiente: Retroalimentación de fuerzas

CAPITULO III METODOLOGÍA

3.1. Enfoque de la Investigación

Se propone usar tres distintos tipos de análisis: propositivo, cualitativo y crítico.

- **Análisis propositivo:** Se busca obtener la tesis y responder la tesis como objetivo principal
- **Análisis cualitativo:** La obtención de la tesis de basa directamente de la calidad de estado del arte y marco teórico investigados.
- **Análisis crítico:** Se determinará el estimado de error en el control del dispositivo háptico.

3.2. Modalidad de la Investigación

El análisis del estado del arte permite conocer los temas más actualizados relacionados con este proyecto, siendo la investigación bibliográfica uno de los pilares fundamentales de esta investigación. Adicional a esto, la investigación es experimental ya que mostrará resultados físicos implementados sobre un dispositivo real.

El proyecto está totalmente dimensionado y se plantea cumplir con cada uno de los objetivos propuestos. De esta manera, los conocimientos obtenidos por la investigación bibliográfica soportarán la implementación del controlador en el robot real. La investigación es exploratoria, ya que parte de una base de conocimientos consultados similares pero que no tienen una relación directa con los objetivos a desarrollar. Para contestar la hipótesis planteada, la exploración es la manera más adecuada de buscar una solución al problema planteado.

En una aplicación real, la investigación propuesta podrá adaptarse a las necesidades de la tarea ya que el control desarrollado tiene ese objetivo, ser completamente escalable y compatible con cualquier modo de uso. Debido al controlador desarrollado, el punto de operación donde se refleja la retroalimentación de fuerzas podrá posicionarse en cualquier punto dentro de su espacio de trabajo o seguir trayectorias definidas, retroalimentando así información del entorno donde el robot remoto ejecuta tareas.

3.3. Operacionalización de variables

Tabla 3-1: Variable Independiente: Control de dispositivo háptico.

Conceptualización	Dimensión	Indicador	Ítems Básicos	Técnica/ Instrumento
<p>Establecimiento de estrategias de control para retornar fuerzas al operador humano, a fin de generar inmersión y proveer información adicional del ambiente remoto. El control se plantea ya que el dispositivo háptico es un robot paralelo, el cual puede funcionar de manera independiente.</p>	-Estrategias de control	Controladores de robots	¿Qué tipos de controladores de robots se pueden emplear?	Técnica: Documental
	-Inmersión	Nivel de inmersión	¿Qué nivel de inmersión se puede conseguir con un dispositivo háptico?	
	-Robot paralelo	Control de un robot paralelo	¿Cómo se puede controlar un robot paralelo?	Instrumento: Ficha Bibliográfica
	-Ambiente remoto	Características del ambiente remoto	¿Cuáles son las características que se pueden extraer de un ambiente remoto?	

Elaborado por: Roberto Segura

Tabla 3-2: Variable dependiente: Retroalimentación de Fuerzas

Conceptualización	Dimensión	Indicador	Ítems Básicos	Técnica/ Instrumento
Método por el cual se generan fuerzas de reacción en el robot maestro que representan elementos encontrados en el entorno de trabajo donde el robot esclavo ejecuta una tarea específica. Las fuerzas generadas por el dispositivo háptico ayudan a la interpretación del ambiente de trabajo, ya que facilitan la ejecución de acciones correctivas en los comandos del robot.	-Generación de fuerzas de reacción -Robot esclavo y robot maestro -Interpretación del ambiente de trabajo	Robot maestro Robot esclavo Fuerzas de reacción Acción correctiva	¿Qué características tiene el robot maestro? ¿Qué características tiene el robot esclavo? ¿Cómo se generan las fuerzas de reacción en el robot maestro? ¿Cómo ayuda la retroalimentación de fuerzas a la acción de control de un robot?	Técnica: Observación Instrumento: Guía de Observación

Elaborado por: Roberto Segura

3.4. Recolección de información

La manera de recolectar la información es crucial para el desarrollo de la investigación. Por ende, se plantea obtener un conjunto de trabajos similares y libros para tener una perspectiva clara del problema a resolver. Esta recolección es llevada a cabo a través del uso de libros físicos, digitales, artículos de revistas y artículos de congresos.

Si la información recogida inicialmente es insuficiente o presenta poca relevancia, no se descarta una segunda recolección para eliminar dudas y reforzar preguntas sin contestar. El medio para la recolección de información es a través de la ficha bibliográfica, la cual permite segmentar la información y ordenarla de acuerdo a la importancia que presenta.

3.5. Procesamiento y análisis

El procesamiento y análisis de la información bibliográfica son el punto de partida de un trabajo original y que sirva de aporte para investigaciones similares futuras. Con esta base de información, se plantean objetivos y una hipótesis a contestar al final del trabajo propuesto. Seguidamente, los resultados que el trabajo presente permitirán analizar si la hipótesis y la formulación del problema han sido contestadas, además de concluir como este trabajo sirve de soporte para investigaciones futuras. Finalmente, se plantea referenciar todos los trabajos que aporten a esta investigación para mostrar el adecuado análisis que tuvo el trabajo previo al arranque de la búsqueda de una solución.

CAPITULO IV

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1. Análisis de Resultados

4.1.1. Controladores clásicos para robots terrestres

Con el propósito de cumplir con lo requerido de un robot, se han desarrollado múltiples controladores para trabajar de manera autónoma o en forma semi-autónoma. El control puede reaccionar de varias formas ante un error y entregar señales de salida para lograr corregir el error producido. De esta manera, esta sección estudia los tipos de controladores clásicos implementados para el control de robots (Sariyildiz, Yu, & Ohnishi, 2015).

A. Control Proporcional

En el control proporcional (P) produce una acción de control proporcional al error, esto quiere decir si el error aumenta la señal de control también y si el error disminuye de la misma forma disminuye la magnitud de la señal de salida del controlador. Cualquiera que se al mecanismo de actuación el controlador proporcional se comporta como un amplificador de ganancia ajustable, este tipo de control puede controlar cualquier tipo de proceso con la desventaja que cuenta con un error en estado estable.

B. Control Integral

El modo de control Integral (I) entrega una señal proporcional al error según transcurra el tiempo, esto quiere decir que siempre que exista error de control la señal cada vez ira incrementando de forma proporcional a la ganancia aplicada, de esta forma se puede incrementar la exactitud de un sistema logrando eliminar el error en estado estacionario.

C. Control Derivativo

El control derivativo (D) genera una acción de control que es proporcional a la derivada del error generando una mayor rapidez en compensación de error. La señal generada por el controlador depende de la cantidad de error que exista, esto se define como si a la existencia de un gran error genera una gran salida de control. La principal característica de este control es anticiparse al error que se puede producir ya que tiende a corregir antes de que el error se incremente en exceso.

Con la combinación de los modos de control mencionados anteriormente se puede tener modos de control mucho más confiables y de mejor rendimiento las combinaciones que se pueden tener de los modos de control son: PI, PD y PID.

D. Control PI

Este modo de control se da ya que casi nunca se utiliza el control Integral solo sino siempre con la compañía de un proporcional. La salida del control PI se da por

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt, \text{ donde } e(t) \text{ representa el error, } K_p \text{ y } K_i \text{ representan las}$$

ganancias proporcionales e integrales respectivamente, por lo tanto la salida del controlador será la suma de P, que será de forma instantánea en la existencia de error y con un pequeño retardo actuará la acción Integral.

E. Control PD

De manera que el control Derivativo no puede ir solo ya que no es capaz de entregar una acción de control cuando existe un error constante. Con la unión de los dos modos de control la acción de control generada por PD se define como;

$$u(t) = K_p e(t) + K_D \frac{de(t)}{dt}$$

donde $e(t)$ representa el error, K_p y K_D son las constantes proporcional y derivativa, respectivamente. Este modo de control tiende a ajustar considerablemente el comportamiento del sistema en cuanto a estabilidad, velocidad y presión.

F. Control PID

El control PID resulta de la combinación de todos los modos de control mencionados

anteriormente, el cual genera una salida dada por
$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_D \frac{de(t)}{dt},$$

por tanto es tipo de control presenta conjuntamente las ventajas de cada modo de control, no tiene desviación en el error y disminuye considerablemente las oscilaciones en estado estable del sistema.

4.1.2. Inmersión generada por los dispositivos hápticos

La inmersión data sobre aquella sensación de que el usuario se traslada hacia otro entorno sin tener que moverse, es decir sentir cosas, oler, manipular objetos de una forma virtual se podría decir. La inmersión de una persona se puede dar por varios métodos ya

sean estos por medio de audio, de forma visual, en el sentido del olfato y de una forma háptica.

La capacidad del ser humano de lograr reconocer objetos con tan solo obtener información táctil es muy avanzada hoy en día. La háptica es la ciencia que estudia el tacto, así como existe la óptica lo que hace con la vista o la acústica con el oído, en otras palabras, busca recibir información por medio de sensaciones provocadas por estímulos en la piel.

A. Interfaces o Dispositivos Hápticas

Los dispositivos Hápticos son robots de pequeña o gran escala que intercambian energía mecánica con el operario, los dispositivos hápticos pueden estar en contacto con alguna parte del cuerpo humano del usuario. Las interfaces Hápticas proporcionan retroalimentación de fuerza, táctil o retroalimentación propio receptiva, permitiendo al usuario conocer información de un determinado lugar. A través de actuadores, los dispositivos hápticos son capaces de proporcionar información como peso, dureza e inercia dinámica de un objeto mediante la retroalimentación de fuerzas, incluyendo en esta retroalimentación la información de un objeto como su peso, inercia y su dureza de un objeto virtual. Las interfaces hápticas que entregan retroalimentación táctil generan datos como geometría de un cuerpo virtual, rugosidad e incluso la temperatura, y entre otras características. En la actualidad los dispositivos manuales son los que más se han destacado dentro del mercado, es por ello que se tiene múltiples dispositivos como: Phantom Omni y el Phantom Desktop de Sensable®), el Falcon de Novint®, y el Virtuose de Haption®.

B. Nivel de Inmersión

Para conseguir un nivel de inmersión de buenas características se requieren propiedades que permitan la capacidad de tocar, manipular y poder sentir un objeto o entorno virtual. La mayor inmersión de un entorno virtual se consigue mediante un dispositivo háptico conjuntamente con una interfaz visual, logrando que la mente recopile toda la información de la forma del objeto. Dentro del objetivo de conseguir un nivel de inmersión suficiente para el usuario se tiene en cuenta el tipo de dispositivo que se emplee. En la siguiente Fig. 4.1 se muestra que tipo de retroalimentación proporcionan diferentes dispositivos, con los

cuales se puede conseguir un nivel de inmersión suficiente, dependiendo de la retroalimentación que ejerce este.

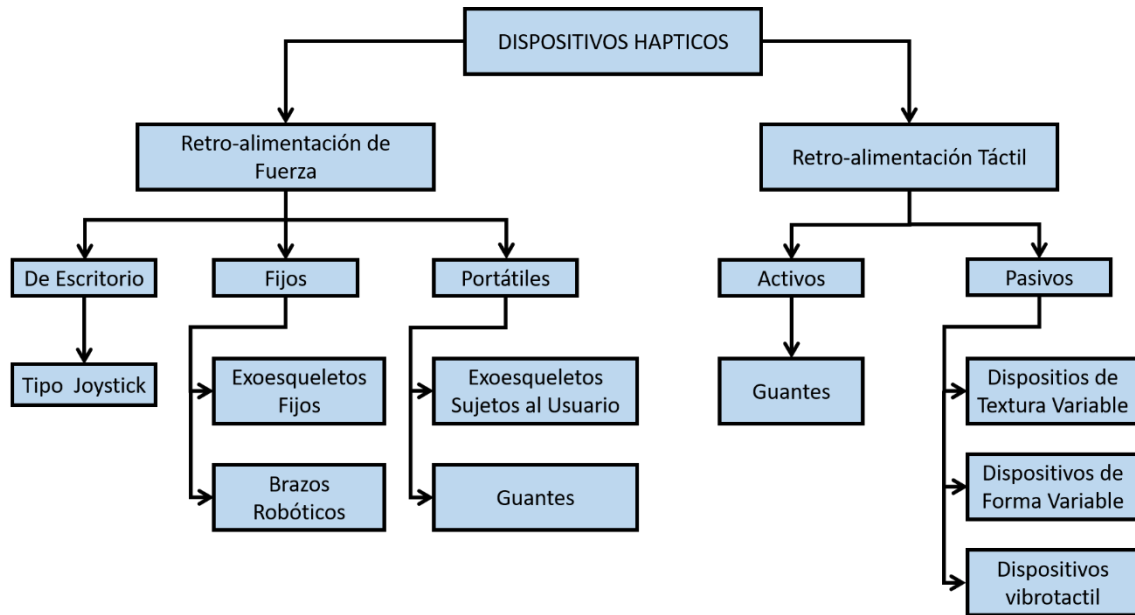


Fig. 4.1: Tipos de dispositivos hápticos

Elaborado por: Roberto Segura

4.1.3. Control de robots paralelos

Los robots paralelos se constituyen por cadenas cinemáticas cerradas, por este motivo son más robustos, fuertes y por ende más precisos que los robots que poseen cadenas abiertas. Un robot paralelo está conformado por un cuerpo rígido con k grados de libertad y una base fija, cuenta con eslabones unidos por lo menos por dos cadenas cinemáticas independientes, su movimiento se produce a través de k actuadores simples (Taghirad, 2017).

Es un poco complejo realizar la clasificación de la robótica paralela, pero según su movimiento se tiene dos tipos de robots paralelos: los planares y los espaciales. En los planares el movimiento se reduce a un plano de dos dimensiones, logrando tener dos o tres grados de libertad, correspondiente a una translación en el plano y a una rotación sobre su eje perpendicular. Por su parte, los robots espaciales pueden posicionarse y desplazarse en los tres ejes, pero con la limitación de no poder rotar en alguno de los ejes (a menos que se incluya un actuador final).

Las aplicaciones de los robots paralelos van desde lo más sencillo en tareas pequeñas de manipulación de objetos como a nivel industrial. El control para este tipo de robots se lo

hace empleando su configuración, es decir obtener un modelo geométrico que gobierne los movimientos del extremo del robot paralelo, logrando aplicar tareas directas e inversas con el robot paralelos, es decir, la cinemática directa y la cinemática inversa. El problema cinemático inverso es complicado para los robots serie, sin embargo, es sencillo para los robots paralelos. Con el modelo geométrico representativo del robot paralelo se puede desarrollar algoritmos que controlen el extremo operativo del robot; la implementación de algoritmos PID, lógica difusa, predicción y entre otros son los más aplicados para cumplir con el objetivo de maniobrar un robot paralelo (Taghirad, 2017).

4.1.4. Características de un ambiente remoto

Un ambiente remoto se define como aquel lugar en el cual se puede desplazar cualquier objeto, en este caso un robot de forma autónoma o de forma tele-operada, localizado a una gran distancia de la estación de mando o del lugar en el cual se está recibiendo datos de navegación del robot (Vertut, 1995).

El ambiente remoto puede presentar varias características en el momento de ser explorado o navegado con un sistema mecánico que logre desplazarse sobre el mismo. Dentro de un entorno remoto se tiene objetos, caminos no deseados, entre otros obstáculos que se puede presentar ante la navegación de un robot. Mediante un robot que es capaz de navegar en el entorno de interés se puede capturar información suficiente, es así como hoy en día se cuenta con dispositivos para reconstruir los ambientes de trabajo. De esta manera, scanners para mapear un entorno en 3D, sensores Lidar para mapear el territorio en un formato 2D y 3D, cámaras para capturar información relevante del entorno, manipuladores con los cuales se puede extraer muestras de cierto lugar, entre otros, han sido desarrollados con el propósito de adquirir información del entorno (Ge, 2006).

La implementación de sensores LIDAR en un robot móvil hace factible la navegación autónoma y de igual forma la recolección de datos del entorno, logrando mapear de forma bidimensional la estructura del entorno remoto. Dependiendo incluso de dispositivos hápticos que permitan interactuar el robot-entorno, se tiene una retroalimentación de fuerza que genera un dispositivo de interacción mediana, la cual se puede deducir objetos con características de textura, rigidez, tamaño, etc., los cuales están presentes en el entorno.

4.1.5. Características de un robot maestro

Las características que definen a un robot maestro pueden variar dependiendo de la aplicación. En este caso, se pueden discriminar a los robots maestros en dos grupos: no hápticos y hápticos, haciendo referencia a los primeros como los robots que solo integran sensores para adquirir información. En este caso, los robots no hápticos están diseñados para poder inyectar velocidades a los robots esclavos, sin el retorno de fuerzas o reacciones físicas como resultado de la interacción del robot remoto con el entorno. Aunque este método no posee un medio para determinar la reacción del robot al ambiente de trabajo, es uno de los más habituales dentro de la robótica de teleoperación, ya que el coste, manejo y alimentación son menores (Ferre, Buss, Aracil, Melchiorri, & Balaguer, 2007).

Una configuración más completa de robot maestro es aquel que permite retornar fuerzas al operario principal. Este tipo de mecanismos se definen como robots hápticos, donde a diferencia del primer caso, no solo están conformados por elementos primarios (sensores), sino también de actuadores y un sistema mecánico completo. La Tabla 4-1 muestra un análisis de las diferencias entre robots maestro hápticos y no hápticos, denotando características entre cada grupo.

Tabla 4-1: Diferencias entre robots maestros hápticos y no hápticos

Característica	Robot no háptico	Robot háptico
Tamaño del robot	Pequeño	Grande
Costo	Moderado/Bajo	Alto
Consumo de corriente	Bajo	Alto
Prestaciones	Limitadas	Altas
Máximo nivel de inmersión	Bajo (Vibración)	Alto (manipulación de texturas)
Sensores	Si	Si
Actuadores	No	Si
Portable	Si	Difícilmente portable
SDK o controlador	Controlador sencillo	Kit para desarrolladores completo

Elaborado por: **Roberto Segura**

Según la Tabla previamente mostrada, las características de un dispositivo háptico superan las prestaciones de un robot maestro sencillo. A pesar de que se tengan dependencias de corriente o un tamaño superior del robot, las prestaciones que presenta un robot háptico permiten elevar el nivel de inmersión de un usuario que controla un robot esclavo en un ambiente remoto no estructurado.

Al denotar a un robot háptico como el apropiado para tareas de teleoperación más precisas y seguras, se debe considerar que existen diferentes tipos de éstos robots con características que los difieren de otros (CHAI3D) (BasTech, 2018). Entre los dispositivos hápticos más conocidos comercialmente se destacan el 3D Touch Haptic Stylus, el Geomagic Touch y el Falcon Novint, todos con características diferentes, pero con propósitos en común (Fig. 4.2).



3D Touch Haptic Stylus



Geomagic Touch



Falcon Novint

Fig. 4.2: Dispositivos hápticos comerciales

Fuente: Haptic Devices (BasTech, 2018)

De manera resumida, la Tabla 4-2 presenta las características más relevantes de los tres dispositivos hápticos mencionados anteriormente.

Tabla 4-2: Características de robots hápticos: 3D Haptic Stylus, Geomagic y Falcon

Característica	3D Haptic Stylus	Geomagic	Falcon Novint
Peso (Kg)	1.5	1.5	6
Dimensiones (cm)	16 x 12 x 7	15 x 12 x 7	10.1 x 10.1 x 10.1
Altura	17.8	15	25
Costo (USD)	650	3900	350
Aplicaciones principales	Games, CAD software, entrenamiento, rehabilitación motricidad fina.	Games, simulación, entrenamiento, armado virtual, mapping.	Games, robótica de control, manipulación de texturas, prueba de habilidades, simulación.
SDK	Windows	Windows	Windows, ROS
Pulsadores	2	2	5
Grados de libertad	6	6	3
Retorno	Fuerza	Fuerza	Fuerza
Ejes de retroalimentación de fuerzas	X, Y, Z	X, Y, Z	X, Y, Z
Interfaz	USB	USB	Serial
Sensor de posición	Potenciómetros	Potenciómetros	Encoders
Fuerza Nominal (N)	3.3	3.3	8.9
Número de brazos	1	1	3

Elaborado por: **Roberto Segura**

El análisis de los dispositivos hápticos permite determinar el más apropiado para la aplicación requerida, donde el robot Falcon Novint presenta características que lo hacen

único para tele presencia. Iniciando con la fuerza nominal que puede generar dada la disposición de los brazos robóticos, el dispositivo seleccionado pudiera ser usado para evasión de obstáculos, generación de trayectorias basadas en fuerza y otros problemas propios de la Robótica. De la misma forma, Falcon Novint muestra características dimensionadas al problema a resolver, donde la cantidad de grados de libertad son suficientes para las posibles tareas de tele operación.

4.1.6. Características de los robots esclavos

Los robots esclavos pueden tener distintas configuraciones dependientes de la aplicación. Desde robots con ruedas (para incluir la motricidad) hasta brazos robóticos pueden ser definidos como robots esclavos, con aplicaciones que involucran tele medicina, desactivación de artefactos explosivos, fotografía aérea, armado de estructuras, etc. Sin embargo, las pruebas sobre robots construidos requieren de una inversión económica importante que únicamente estudios completos pueden costear, obligando a investigadores a buscar opciones cercanas al comportamiento real de los robots (Ruiz, 2012).

A través de la modelación de cinemática y dinámica de robots, diversos simuladores han logrado recrear el comportamiento de diversos mecanismos, sobre los cuales se pueden probar algoritmos de control más sofisticados y complejos, así como ejecutar tareas que involucren espacios confinados o peligrosos para un robot real. Adicional a esto, la simulación permite la interacción entre diversos robots de distintas características, a fin de probar algoritmos relacionados a robótica cooperativa.

Aunque existe una amplia variedad de simuladores de robótica, la mayoría integra codificación propia para poder controlar los robots o están basados puramente en lenguajes de bajo nivel. Sin embargo, dada la necesidad de intercomunicar los objetos simulados con lenguajes matemáticos o entradas de control reales, paquetes de software compatibles con otros programas han sido desarrollados, muchos de los cuales se usan con fines completamente comerciales. Tres de los software más conocidos son V-REP, ARGoS y GAZEBO, definiendo a V-REP como un ambiente de simulación de muy buenas características, las cuales incluyen un editor de escena y de modelos, una librería extensa de modelos, un manipulador de mallas en tiempo real, entre otras características. Por su parte, ARGoS provee una alternativa liviana de procesamiento que es especialmente relevante para procesadores de baja capacidad. Finalmente, Gazebo

representa una opción media entre los mencionados anteriormente, poseyendo el motor de física como V-REP pero con la carencia de alguna de las mejores características para optimizar el procesamiento.

Una descripción detallada de las características de cada uno de los software propuestos se muestran en la Tabla 4-3, detallando mejores características por medio de colores; verde mejores características, rojo peores características (Lenkaspac, 2018).

Tabla 4-3: Características de los software de simulación

Características	V-REP	Gazebo	ARGoS
Plataformas soportadas	MacOS, Linux, Windows	MacOS, Linux, Windows	MacOS, Linux
Motores de física	Bullet 2.78, Bullet 2.83, ODE, Vortex and Newton.	ODE	Motores físicos por defecto
Editor	Editor de código y escenas	Editor de código y escenas	Editor de código
Edición de escenas	Permitido	Permitido	Permitido
Modificación de meshes	Modificación permitida	Modificación no permitida	No existe manipulación de meshes
Calidad de salidas	Salidas de video, ploteo de datos y archivos de texto	Archivos de log, frames de video, fotografías y archivos de texto	Salidas de frames de video y archivos de texto
Tipos de robots incluidos	Gran variedad de robots, incluyendo bípedos, hexápodos, con ruedas, voladores y robots tipo serpiente. Además de actuadores y sensores	Librería menos diversa de robots, que incluyen robots con ruedas y voladores. Robots de terceros pueden ser incluidos siempre y cuando sean compatibles	Una cantidad mínima de robots es incluida

Nivel de detalle en robots simulados	Los modelos de defecto tienen gran detalle y por ende apropiados para simulaciones de alta precisión	Los modelos de defecto son completamente sencillos pero computacionalmente aptos para simulación	Los modelos de defecto son completamente sencillos pero aptos para procesamiento
Tipo de formato	Almacenamiento en un formato especial de V-REP	Escena almacenada en un formato XML.	Escena almacenada en un formato XML.
Modo de control de los robots	Se permite la inclusión de scripts de control	Difícil de reconocer modelos de robots de terceros	Existe cierta documentación del funcionamiento del robot, pero la mayoría requiere ser deducido por código.
Ejemplos	Todos los robots poseen ejemplos de funcionamiento	Ejemplos de internet difíciles de ejecutar dada la gran cantidad de dependencias	Muy pocos ejemplos provistos por los creadores
Documentación API	Buena documentación, basada en tutoriales y ejemplos, además de una comunidad de soporte	Buena documentación basada en tutoriales paso a paso	Buena documentación, pero con una pequeña comunidad de usuarios disponible.
Congelamiento de interface	Congelamiento de interface muy raro	Congelamiento continuo dependiendo del tipo de procesador	Difícil congelamiento de interface

Actualización de librerías	Siempre disponible indiferente a la conexión del internet	Librerías disponibles on-line	Disponible indiferente de la conexión a internet
Organización de las librerías	Organizado en carpetas basado en la categoría del modelo	La larga lista de librerías es encontrada con dificultad	No existe librería de modelos.

Elaborado por: **Roberto Segura**

El análisis de las características de todos los software de simulación permiten determinar el más apropiado dependiendo de la aplicación. Para el caso donde se requieran propiedades dinámicas del robot, el consumo de recursos de computadora debe ser lo suficientemente dimensionado. En la aplicación propuesta en este trabajo investigativo, el software de simulación que mejor se ajusta a los requerimientos propuestos es V-REP, dadas las características previamente expuestas.

4.1.7. Generación de fuerzas en un robot maestro

Las fuerzas de reacción generadas por el robot maestro han sido estudiadas por muchos años. Conociendo que el problema principal de la robótica es el desconocimiento completo o parcial del espacio de trabajo donde un robot ejecuta una tarea específica, ha sido necesaria la inclusión de sensores sobre el robot remoto. De este modo, los robots ubicados remotamente pueden tener la capacidad de sentir los objetos que están manipulado, viendo dichos objetos y reconociendo que obstáculos se encuentran a su alrededor. A su vez, la inclusión de sensores permite tener una perspectiva diferente a un operario remoto, donde la inclusión de fuerzas de reacción permite tele operar al robot esclavo de una mejor manera. Sin embargo, el uso de sistemas de visión supera por mucho al uso de fuerzas de reacción, aun cuando las características de visión presentan aun ciertos problemas para describir apropiadamente un ambiente remoto (Park, Won, Lee, & Sziebig, 2016).

Las fuerzas de reacción se basan directamente del tipo de sensores incluidos sobre el robot esclavo, el cual retorna la información del ambiente a la estación principal. La estación principal es el sitio donde el operador ejecuta acciones de control al robot remoto a través del robot principal. Comúnmente, el robot principal está compuesto únicamente por

sensores que detectan la variación de entradas, sin embargo y para generar fuerzas de reacción, muchos robots incluyen actuadores (motores) para retornar empujes o vibración, redefiniendo a estos robots como hápticos (Ortmaier, y otros, 2007). A continuación (Tabla 4-4), se describen las formas en las que se pueden generar las fuerzas de reacción en los robots hápticos, todas dependientes del tipo de aplicación final.

Tabla 4-4: Generación de fuerzas de reacción dependientes de la aplicación

<p>Fuerza constante</p>	<p>Los sensores de fuerza-torque fueron inicialmente manufacturados para aplicaciones relacionadas a pulimiento, molienda, afilado y algún tipo de aplicaciones para remover material. Dado que estas aplicaciones son difíciles de automatizar con controladores, un operador remoto podría ejecutar estas tareas a través de algún tipo de indicador de fuerza de reacción. De esta manera, el operario remoto puede darse cuenta si está presionando el efector final con sobre fuerza o está aplicando una fuerza no necesaria como para cortar una pieza.</p>
<p>Ubicación de objetos</p>	<p>El método tradicional de ubicación de objetos sobre un espacio de trabajo es a través de sensores de visión. Sin embargo, esta solución involucra aplicaciones donde no se consideran trabajos en el espacio, ya que el reconocimiento a través de una sola cámara se reduce a dos dimensiones. Sin embargo, con el uso de sensores o dispositivos basados en laser se puede determinar la distancia existente entre el robot y objetos cercanos, pudiendo convertir esta distancia en la fuerza necesaria para determinar la ubicación de un elemento o simplemente evadirlo para evitar daños en el mecanismo.</p>
<p>Repetitividad de fuerzas</p>	<p>La robótica muchas de las veces implica repetitividad de actividades, especialmente en tareas de ensamblaje. Sin embargo, tareas que requieren de mayor experticia y toma de decisiones requieren un operario humano encargado de modificar las acciones de control en cualquier instante de tiempo. Para esto, la repetitividad de fuerzas puede jugar un papel importante en el ensamblaje de piezas pequeñas o grandes, donde todas las acciones de control provistas por el operario son trasladadas al robot maestro, el cual se comunica con el remoto a través de alguna plataforma de comunicación. El robot remoto por su</p>

	parte, puede ser de mayor o menor escala, permitiendo ejecutar movimientos más finos o más gruesos, dependiendo de la aplicación requerida.
Levantar y pesar objetos	A través de la retroalimentación de fuerzas, se pueden extender las aplicaciones relacionadas al ensamblaje, selección o apilamiento de objetos. A diferencia de procesos basados en la identificación por color, la retroalimentación de fuerzas permite determinar qué tipo de elemento se ha capturado, con lo cual diversas tareas pueden ejecutarse de manera más óptima. Por ejemplo, la tarea de apilamiento podría no requerir de indicadores de color sobre los distintos tipos de objetos, ya que el propio peso de cada uno de los elementos podría determinar la posición en la cual deberían ser ubicados, adicional a que se puede detectar tanto el agarre como la liberación del objeto.
Orientación y guía en tareas cooperativas	Las tareas colaborativas dentro de la robótica pueden incluir la interacción entre diversos robots y un operario humano, donde los robots ejecutan tareas autónomas mientras los operarios humanos ejecutan tareas tele operadas. En este aspecto, la dinámica generada por el conjunto de robots puede afectar el tele control del operario humano, donde la realimentación de fuerzas puede describir este comportamiento de una manera más sencilla de interpretar por el operador. Por ejemplo, en un conjunto vehículo móvil – brazo robótico, el vehículo móvil puede estar ejecutando el seguimiento de una trayectoria de manera autónoma, mientras el brazo robótico puede ir recolectando objetos del camino a través de la operación de usuario. A través de la respuesta háptica, el operario podría sentir la dinámica generada por el vehículo para poder calcular las acciones y recoger los elementos deseados.

Elaborado por: **Roberto Segura**

A través de las aplicaciones expuestas anteriormente, se puede determinar la forma en las cuales se generan las fuerzas de reacción. A pesar de no tener una relación directa entre ellas, el mismo principio de realimentación de fuerzas puede ser utilizado para mejorar el desempeño de la tarea final.

4.1.8. Control de un robot a través de la retroalimentación de fuerzas

Ya sea a través de un robot real o un robot simulado, la retroalimentación de fuerzas permite crear en el usuario la sensación de estar en contacto con el entorno en donde se ejecutan acciones a través de un robot esclavo. De esta manera, la combinación entre la computación gráfica y la tecnología háptica puede ser usada para entrenar personas para ejecutar tareas que requieren de la coordinación mano-ojo, tales como cirugía, maniobras en espacios peligrosos o tareas espaciales (Ibrahim & Ali, 2016).

A modo de ejemplo, una de las aplicaciones que ha involucrado la retroalimentación de fuerza con mayor enfoque es la de cirugía. Dadas las tareas tan delicadas que se llevan a cabo, la precisión en la ejecución de este tipo de actividades requiere forzosamente la retroalimentación de algún tipo de fuerza que permita determinar cómo se está ejecutando la intervención médica. Por ende, sin la retroalimentación de fuerzas el médico (ubicado en una locación remota) ignoraría el nivel de fuerza que debe ser aplicado para ejecutar una intervención, dejando como resultado dos posibles escenarios: si no se aplica con la fuerza suficiente, la ejecución de la tarea no se llevara a cabo de la manera esperada; mientras que, si se aplica con una sobre fuerza, el trauma y el daño sobre el paciente pudiese resultar más afectivo que el daño inicial. Por otra parte, en tareas de gran escala (tareas espaciales) la retroalimentación de fuerzas puede brindar soporte en la reconstrucción de equipos dañados o en el montaje de nuevos equipos. A través del conocimiento de la fuerza aplicada al colocar objetos en algún espacio, la experticia de un ensamblador puede verse mejorada, donde en el sitio remoto puede estar ubicado un robot de mayores dimensiones y que soporte altas cantidades de radiación o características nocivas del ambiente.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- El análisis de las características de todos los dispositivos hápticos comerciales permite reconocer cuál de todos es el más apropiado y dimensionado para la aplicación propuesta. A través de análisis bibliográfico, se determina que características son requeridas para llevar a cabo el objetivo final, en este caso, la retroalimentación de fuerzas orientada a tele presencia.
- Una vez determinado el dispositivo háptico apropiado para tareas de tele presencia, es posible orientar toda la búsqueda bibliográfica hacia la disposición física de dicho dispositivo con el objetivo de analizar su comportamiento cinemático. A través del análisis matemático del mecanismo háptico, se puede proponer un modelo equivalente.
- El modelo matemático resultante del dispositivo háptico permite simular el equipo, proponer un método de control y probar dicho algoritmo a nivel físico y simulado. De este modo, un modelo matemático equivalente puede permitir probar el comportamiento del robot y verificar los controladores propuestos.
- La manera más apropiada de verificar tanto el modelo matemático equivalente como el controlador propuesto es a través de pruebas experimentales. Las pruebas permiten determinar prácticamente el rendimiento de los controladores, donde el robot físico puede ejecutar las aplicaciones deseadas.
- La bibliografía presentada ayuda a entender la problemática que se busca solucionar, donde se identifican todas las aplicaciones que se podrían llevar a cabo a partir de la solución propuesta a través de esta investigación.

5.2. Recomendaciones

- Aplicar las metodologías establecidas para facilitar la selección de todos los recursos necesarios para llevar al cabo la resolución de la problemática. Asimismo, usar los diagramas y lluvia de ideas para facilitar la búsqueda de los parámetros que permitirán plantear la resolución del tema de investigación planteado.

- Usar la bibliografía investigada para seleccionar tanto el dispositivo más apropiado para resolver la problemática encontrada, así como optar por el software de simulación que mejores características presente.
- Plantear un método de control apropiado para validar el modelo matemático del robot háptico, así como para configurar trayectorias sobre el robot físico.
- Ejecutar diversas pruebas experimentales para demostrar la validez del controlador propuesto. De esta manera, la ejecución de pruebas permitirá conocer las ventajas y limitaciones del robot háptico, el cual está conectado al software de simulación.
- Proponer dos aplicaciones que puedan ser ejecutadas sobre el software de simulación para validar tanto la intercomunicación entre el software controlador del robot háptico y el software de simulación, así como explotar las mejores características del software seleccionado.

CAPITULO VI

PROPUESTA DE INVESTIGACIÓN

6.1. Antecedentes de la propuesta

El control tele operado la mayoría de las veces incluye robots maestros que permiten introducir acciones de control al robot remoto sin una retroalimentación de fuerzas. Dependiendo de la experticia del operador humano, este tipo de técnicas puede ser riesgosa tanto para el robot esclavo como para los elementos que están ubicados en el espacio de trabajo.

El conocimiento del espacio puede definirse entonces en dos partes: espacios estructurados y espacios no estructurados. Los trabajos desarrollados en la actualidad se basan en conocer la mayoría de aspectos espaciales del entorno de trabajo, por ende, se estima que no se tienen obstáculos que deban considerarse al momento del control. Por otra parte, el trabajo sobre espacios no estructurados debe contemplar diversas características que han sido continuamente estudiadas con el propósito de ejecuciones satisfactorias.

Comúnmente, la retroalimentación de información de un espacio de trabajo no estructurado viene dada por una cámara (o un arreglo de cámaras en el mejor de los casos), información que es limitada dependiendo del tipo de aplicación y posee características muy pobres para un control completo en tareas complejas. Por su parte, la retroalimentación de fuerzas extiende diversas ventajas que ayudan físicamente al operador a ejecutar una tarea sobre el entorno, donde la interacción directa con el extremo operativo del robot esclavo puede mostrar una perspectiva más clara del comportamiento del robot sobre el ambiente de trabajo.

6.2. Justificación

Conociendo las limitantes que se tienen al incluir dispositivos no hápticos, este trabajo propone el estudio cinemático del robot maestro Novint Falcon, robot que mejores características presenta para la ejecución del tipo tele operadas de este trabajo. Asimismo, se plantea un método de control dimensionado para el dispositivo, donde se analiza el controlador simuladamente y basado en el modelo matemático que representa al robot maestro. Finalmente, el método de control es aplicado sobre entornos de simulación

robótica, considerando robots móviles y robots estacionarios (brazo robótico) para validar la usabilidad del sistema planteado y las respuestas del dispositivo háptico.

Este tipo de trabajos permitiría incrementar el nivel de destreza en operarios ubicados en estaciones seguras, mientras comandan robots ubicados en locaciones peligrosas o que involucren algún tipo de riesgo físico, tales como el manejo de explosivos, mezcla de químicos peligrosos o la guía de vehículos bélicos. De esta manera, además de tener una percepción visual, el operario tiene como retroalimentación la fuerza exacta a la cual debe comandar un mecanismo robótico, permitiendo aplicarle la fuerza adecuada para tener un accionamiento final esperado.

6.3. Objetivos de la propuesta

6.3.1. Objetivo general

Modelar matemáticamente el dispositivo háptico seleccionado e implementar un control para el mejoramiento de tareas teleoperadas, las cuales son simuladas por un paquete informático e integran diversos sistemas robóticos.

6.3.2. Objetivos específicos

- Utilizar la información bibliográfica como base para el desarrollo de la propuesta, considerando que el software de simulación V-REP posee las mejores características para las pruebas experimentales, el software matemático de Matlab permite la validación de controladores y el dispositivo Novint Falcon se presenta como el dispositivo háptico más apropiado para este trabajo.
- Determinar el modelo matemático que representa al dispositivo, considerando la estructura mecánica y la distribución de actuadores y sensores. Posterior a la determinación de un modelo matemático, se plantea el uso del mismo para la simulación del comportamiento del sistema robótico con el propósito de validar la representación matemática.
- Proponer un método de control basado en el modelo matemático y que sea validado mediante simulación para el control del punto de interés del Novint Falcon, a través de técnicas de ajuste de parámetros.
- Usar el paquete informático de simulación V-REP para la construcción de entornos, donde sistemas robóticos varios puedan interactuar con el medio y

puedan retroalimentar información que sirva para validar el controlador propuesto para el dispositivo háptico.

- Probar experimentalmente la unificación entre el dispositivo háptico, el control provisto por Matlab y la simulación de sistema robóticos permitida por V-REP, con el objetivo de validar todo el trabajo desarrollado.

6.4. Análisis de factibilidad

6.4.1. Factibilidad operativa

El dominio tanto del software matemático como el software de control facilita el desarrollo de la propuesta, donde el entendimiento del análisis matemático del dispositivo permite brindar soluciones de este tipo. Asimismo, el sustento bibliográfico encontrado en trabajos similares viabiliza el desarrollo de la simulación, experimentación y conexión con el dispositivo seleccionado.

6.4.2. Factibilidad técnica

El proyecto es técnicamente factible una vez se adquiera el único dispositivo háptico requerido. Este inconveniente es resuelto mediante la importación del robot maestro, dado que el mercado local no tiene a su disponibilidad este tipo de robots dado que su uso es orientado a aplicaciones puntuales. La selección de equipos y la discriminación de las características de cada uno de ellos permiten seleccionar al más adecuado y dimensionado, evitando de esta manera optar por dispositivos más costosos e innecesarios para el desarrollo de la propuesta.

6.4.3. Factibilidad económica

El análisis de las características de cada uno de los dispositivos hápticos permite seleccionar al más apropiado para el desarrollo de este proyecto. De esta manera, se opta por un robot maestro económico y dimensionado para el trabajo, considerando que las licencias de uso e instaladores vienen incluidos en el paquete de compra. Por otra parte, dado que no se plantea la construcción de robots esclavos para la comprobación del control planteado, únicamente se requiere de un ordenador con características de procesamiento altas, el cual es fácil de adquirir dada la condición económica del país.

6.4.4. Proyección a futuro

El proyecto plantea a largo plazo el uso de entornos de simulación más inmersivos, en los cuales la apreciación de fuerzas y la interacción con medios remotos sea más intuitiva para el usuario. De esta manera, se puede buscar financiamiento externo para el desarrollo de proyectos relacionados a simuladores de vuelo, controladores de robots en superficies peligrosas, manipulación de objetos explosivos, etc. Asimismo, se pueden plantear nuevas técnicas de control a cada lado de los robots (en el sitio local y en el sitio remoto), incluyendo controladores inteligentes o predictivos, con el objetivo de incrementar la transparencia en la teleoperación.

6.5. Fundamentación científico – técnica

6.5.1. Simulador V-REP

Dentro del campo de la simulación de algoritmos y pruebas con robots, muchas de las veces es factible el uso de software que cumpla los requerimientos necesarios para trabajar de forma unificada con un software matemático, donde el software matemático contendrá los algoritmos que permitan manejar a los robots de manera autónoma.

De esta manera, el desarrollo de este trabajo optó por el empleo de V-REP, un software de simulación de robots que permite evaluar el funcionamiento de prototipos y algoritmos sin la necesidad de tener un robot físico real. V-REP cuenta con un IDE que permite el desarrollo de entornos tridimensionales, logrando controlar dentro de éstos a los diferentes robots que se puedan incorporar. Los diferentes algoritmos de programación para los robots pueden ser escritos en C, C++, Java, Octave, Python, Lua, entre otros lenguajes. El simulador puede ser utilizado para varias aplicaciones, como Simulación de procesos de automatización, control de hardware, monitoreo remoto, desarrollo rápido de algoritmos, prototipado rápido y verificación de productos finales. Las características de V-REP incluyen un soporte para trabajar de manera conjunta con varios sistemas operativos y a la vez comunicarse entre varios softwares.

6.5.2. Matlab

En el desarrollo de los algoritmos de control, el dispositivo háptico emplea un software matemático que permita integrar las funcionalidades de cálculos numéricos y programación de tareas para robots. Este software además de simular algoritmos permite

intercomunicarse con otros dispositivos de hardware. A pesar de no ser el único paquete informático matemático que permita el desarrollo de controladores, Matlab posee características que lo hacen excepcional frente a otros paquetes, tales como la inclusión de herramientas para el uso de librerías propietarias o de terceros, cálculo a nivel matricial, herramientas de graficación, etc.

6.5.3. Robot unicycle

El modelo cinemático de un robot entrega las velocidades del punto de operación en función de las velocidades de maniobrabilidad, para encontrar el modelo cinemático se ha empleado el método de matrices rotacionales. En la Fig. 6.1 se indica la cinemática del robot unicycle Pioneer 3-DX.

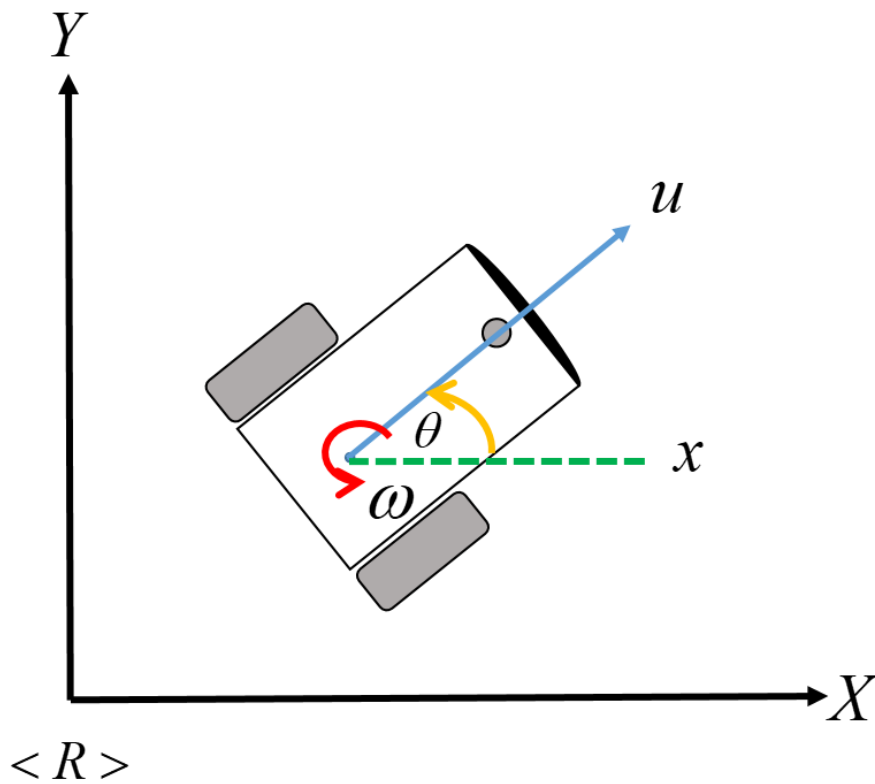


Fig. 6.1: Cinemática del robot unicycle Pioneer 3-DX

Elaborado por: **Roberto Segura**

Un robot diferencial está conformado por dos velocidades un lineal u que desplaza hacia delante al robot y una velocidad angular ω , mediante estas velocidades se pueden lograr trasladar el robot de un lugar hacia otro.

Para encontrar la posición del robot en cualquier instante de tiempo se hace un análisis matemático de la siguiente manera Fig. 6.2.

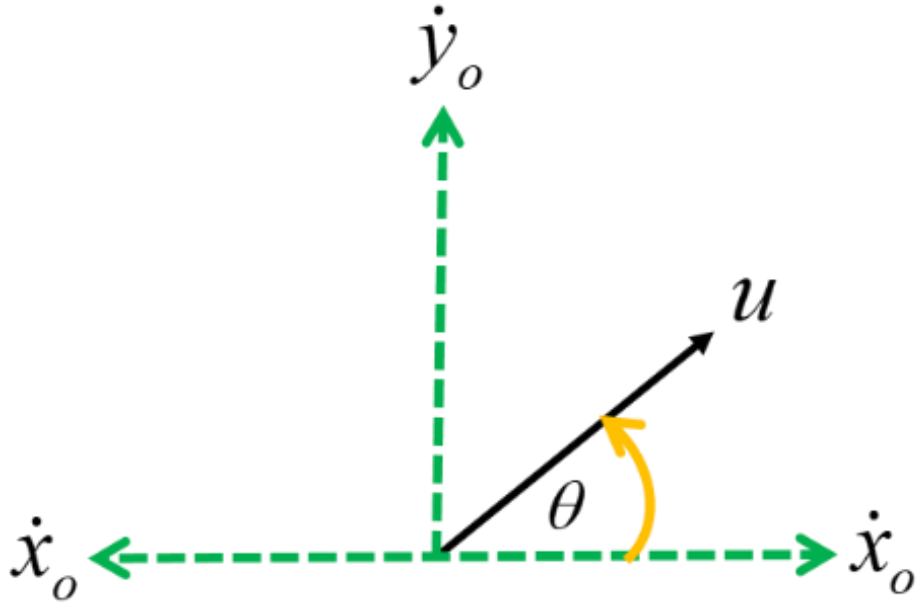


Fig. 6.2: Análisis matemático para determinar la posición del robot

Elaborado por: **Roberto Segura**

Entonces la velocidad en el eje x se define de la siguiente como:

$$\dot{x}_o = u \cos \theta$$

para la velocidad del robot sobre el eje y se describe de la forma

$$\dot{y}_o = u \sin \theta$$

y la velocidad de rotación del robot se define la misma velocidad angular expresada de la siguiente forma:

$$\dot{\theta} = \omega$$

Para realizar el control en el punto frontal del robot donde están ubicados los sensores y no en el centro del robot se realiza un desplazamiento frontal quedando la cinemática como se muestra en la siguiente gráfica Fig. 6.3.

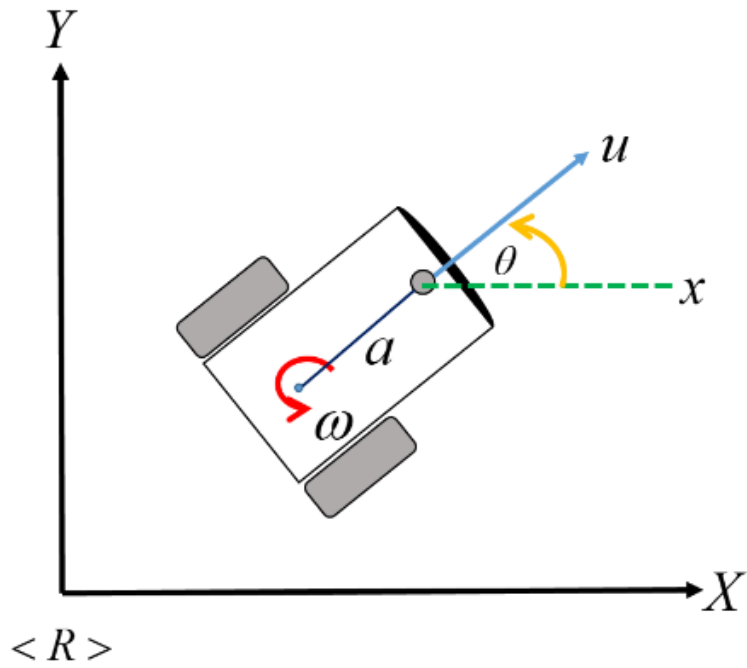


Fig. 6.3: Control en el punto frontal del robot

Elaborado por: **Roberto Segura**

La posición del robot se puede analizar de tal manera que se tenga claro las velocidades del robot y las coordenadas operativas de la plataforma, en la Fig. 6.4 se representa los vectores y variables que influyen en el nuevo punto operativo del robot.

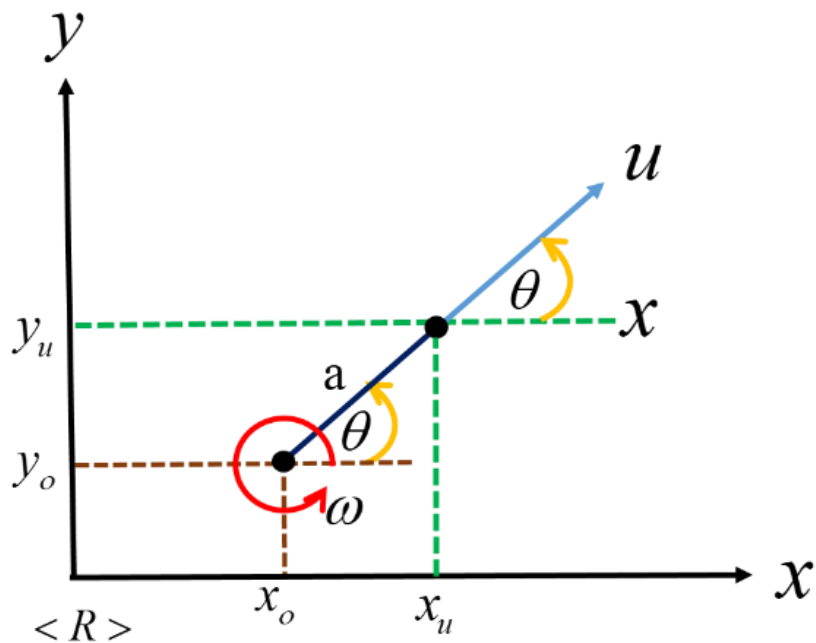


Fig. 6.4: Vectores y variables del punto desplazado para control

Elaborado por: **Roberto Segura**

La posición del robot con respecto al plano referencial $\langle R \rangle$ se define de la siguiente manera:

$$\begin{aligned}x_u &= x_o + a \cos \theta \\y_u &= y_o + a \sin \theta\end{aligned}$$

Si aplicamos la derivada parcial con respecto al tiempo a la ecuación se obtiene la siguiente expresión.

$$\begin{aligned}\dot{x}_u &= \dot{x}_o - a \sin \theta \dot{\theta} \\ \dot{y}_u &= \dot{y}_o + a \cos \theta \dot{\theta}\end{aligned}$$

Si reemplazamos las ecuaciones anteriores se obtiene:

$$\begin{aligned}\dot{x}_u &= u \cos \theta - a \omega \sin \theta \\ \dot{y}_u &= u \sin \theta + a \omega \cos \theta\end{aligned}$$

el modelo cinemático del robot expresado las velocidades del punto operativo en función de las velocidades de control del robot se describen como:

$$\begin{bmatrix} \dot{x}_u \\ \dot{y}_u \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix}$$

El sistema se puede expresar de la siguiente forma:

$$\dot{\mathbf{r}}(\mathbf{t}) = \mathbf{J}(\boldsymbol{\theta}) \mathbf{v}(\mathbf{t})$$

Donde $\dot{\mathbf{r}}(\mathbf{t})$ es el vector que contiene las velocidades en el punto operativo, $\mathbf{J}(\boldsymbol{\theta})$ es la matriz de transformación que relaciona las velocidades en el punto de operación y las velocidades de control del robot $\mathbf{v}(\mathbf{t})$.

6.5.4. Manipulador robótico

La configuración del manipulador robótico a ser empleado se considera para realizar el modelo cinemático el extremo operativo como punto de interés $\mathbf{h} = [hx \quad hy \quad hz]$, donde $\langle R \rangle$ representa el sistema de referencia de coordenadas del brazo robótico, en la Fig. 6.5 se indica como esta expresado cada variable que interfiere en el manipulador.

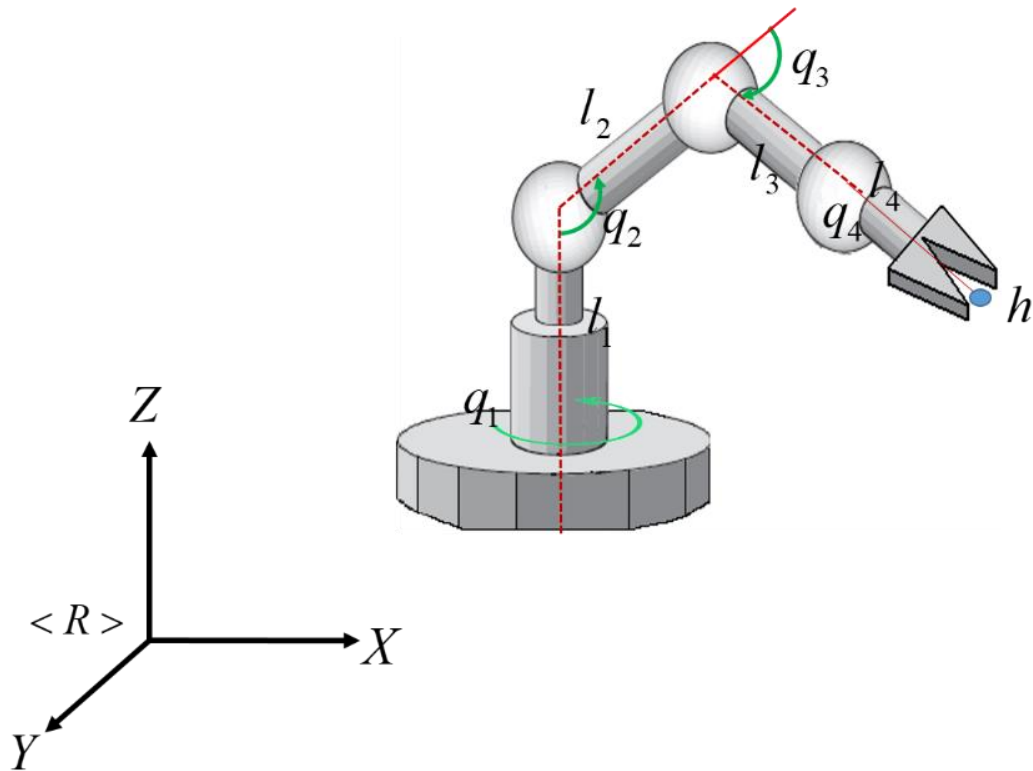


Fig. 6.5: Representación gráfica de un manipulador robótico

Elaborado por: **Roberto Segura**

La posición del extremo operático de define en base a la geometría de los eslabones sobre el plano referencias para caja eje.

Para la posición h_x se tiene:

$$h_x = l_2 \cos q_2 \cos q_1 + l_3 \cos(q_2 + q_3) \cos q_1 + l_4 \cos(q_2 + q_3 + q_4) \cos q_1$$

La posición h_y geoméricamente se detalla como:

$$h_y = l_2 \cos q_2 \sin q_1 + l_3 \cos(q_2 + q_3) \sin q_1 + l_4 \cos(q_2 + q_3 + q_4) \sin q_1$$

Y para la posición de h_z :

$$h_z = l_1 + l_2 \sin q_2 + l_3 \sin(q_2 + q_3) + l_4 \sin(q_2 + q_3 + q_4)$$

Donde l_1, l_2, l_3 y l_4 son las dimensiones respectivas de cada eslabón que contiene el brazo robótico y q_1, q_2, q_3 y q_4 son los ángulo de rotación para cada grado de libertad del manipulador.

Para obtener el modelo cinemático de realiza la derivada parcial de cada posición obteniendo así las siguientes ecuaciones para h_x :

$$\begin{aligned}\dot{h}_x = & -l_2 \left[\sin q_2 \dot{q}_2 \cos q_1 + \cos q_2 \sin q_1 \dot{q}_1 \right] \\ & -l_3 \left[\sin (q_2 + q_3) (\dot{q}_2 + \dot{q}_3) \cos q_1 + \cos (q_2 + q_3) \sin q_1 \dot{q}_1 \right] \\ & -l_4 \left[\sin (q_2 + q_3 + q_4) (\dot{q}_2 + \dot{q}_3 + \dot{q}_4) \cos q_1 + \cos (q_2 + q_3 + q_4) \sin q_1 \dot{q}_1 \right]\end{aligned}$$

De igual manera se deriva h_y :

$$\begin{aligned}\dot{h}_y = & -l_2 \left[\sin q_2 \dot{q}_2 \sin q_1 - \cos q_2 \cos q_1 \dot{q}_1 \right] \\ & -l_3 \left[\sin (q_2 + q_3) (\dot{q}_2 + \dot{q}_3) \sin q_1 - \cos (q_2 + q_3) \cos q_1 \dot{q}_1 \right] \\ & -l_4 \left[\sin (q_2 + q_3 + q_4) (\dot{q}_2 + \dot{q}_3 + \dot{q}_4) \sin q_1 - \cos (q_2 + q_3 + q_4) \cos q_1 \dot{q}_1 \right]\end{aligned}$$

Y para h_z :

$$\dot{h}_z = l_2 \cos q_2 \dot{q}_2 + l_3 \cos (q_2 + q_3) (\dot{q}_2 + \dot{q}_3) + l_4 \cos (q_2 + q_3 + q_4) (\dot{q}_2 + \dot{q}_3 + \dot{q}_4)$$

Con las velocidades obtenidas en el extremo operativo del brazo robótico se obtiene el modelo cinemático definido como:

$$\dot{\mathbf{h}}(t) = \mathbf{J}(q) \dot{\mathbf{q}}(t)$$

Donde, $\mathbf{J}(q)$ es la matriz jacobiana que representa el mapeo lineal en el vector de velocidad del brazo $\dot{\mathbf{q}}(t)$ y el vector de las velocidades del extremo operativo $\dot{\mathbf{h}}(t)$. La matriz jacobina del moldeo cinemático se forma de la siguiente forma:

$$\begin{aligned}
J_{11} &= -\sin q_1[l_2 \cos q_2 + l_3 \cos(q_2 + q_3) + l_4 \cos(q_2 + q_3 + q_4)] \\
J_{12} &= -\cos q_1[l_2 \sin q_2 + l_3 \sin(q_2 + q_3) + l_4 \sin(q_2 + q_3 + q_4)] \\
J_{13} &= -\cos q_1[l_3 \sin(q_2 + q_3) + l_4 \sin(q_2 + q_3 + q_4)] \\
J_{14} &= -\cos q_1[l_4 \sin(q_2 + q_3 + q_4)] \\
J_{21} &= \cos q_1[l_2 \cos q_2 + l_3 \cos(q_2 + q_3) + l_4 \cos(q_2 + q_3 + q_4)] \\
J_{22} &= \sin q_1[l_2 \sin q_2 + l_3 \sin(q_2 + q_3) + l_4 \sin(q_2 + q_3 + q_4)] \\
J_{23} &= \sin q_1[l_3 \sin(q_2 + q_3) + l_4 \sin(q_2 + q_3 + q_4)] \\
J_{24} &= \sin q_1[l_4 \sin(q_2 + q_3 + q_4)] \\
J_{31} &= 0 \\
J_{32} &= l_2 \cos q_2 + l_3 \cos(q_2 + q_3) + l_4 \cos(q_2 + q_3 + q_4) \\
J_{33} &= l_3 \cos(q_2 + q_3) + l_4 \cos(q_2 + q_3 + q_4) \\
J_{34} &= l_4 \cos(q_2 + q_3 + q_4)
\end{aligned}$$

Entonces la matriz jacobina se forma como:

$$\mathbf{J} = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} \\ J_{21} & J_{22} & J_{23} & J_{24} \\ J_{31} & J_{32} & J_{33} & J_{34} \end{bmatrix}$$

6.6. Metodología, Modelo operativo

La metodología usada para resolver la problemática encontrada sigue un conjunto de pasos jerárquicos mostrados en la Fig. 6.6. En ésta, se detallan todas las secciones que contendrá la metodología para resolver la problemática. Inicialmente, se considera un análisis matemático de las ecuaciones que representan al sistema robótico maestro, analizando la cinemática del dispositivo. Seguidamente, se tiene la consideración de controladores y librerías usadas para conectar el dispositivo háptico al ordenador, en el cual se tienen instalados los paquetes informáticos de Matlab y V-REP. La modelación matemática y la implementación de un método de control permiten simular el comportamiento del dispositivo en Matlab, previo a la ejecución de pruebas reales. Asimismo, el método de control permite definir tareas a ambos escenarios, el del robot móvil tipo unicycle así como el del manipulador. Se considera que ambos escenarios contienen elementos primarios o equipos que permiten retroalimentación del entorno de trabajo al software matemático para convertir la información adquirida a fuerzas. La conversión a fuerzas permitió cumplir con el propósito de este trabajo, el cual se centra en presentar una perspectiva más clara de la interacción entre el robot esclavo y el medio con el que interactúa.

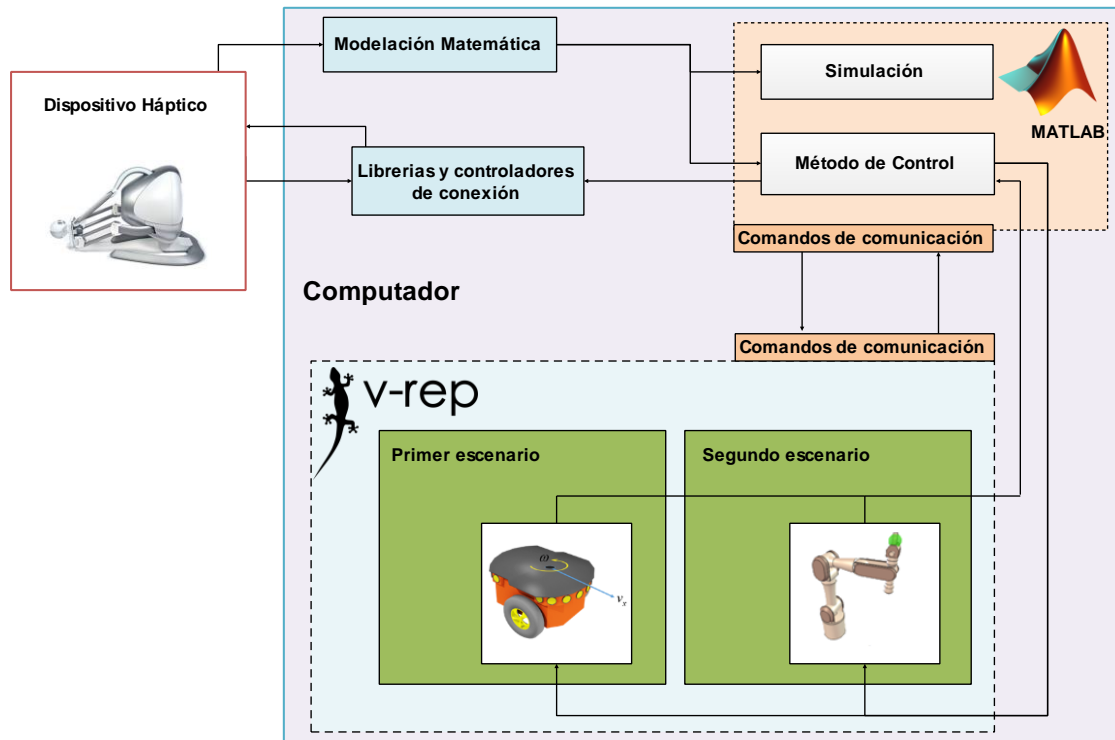


Fig. 6.6: Metodología para resolver la problemática

Elaborado por: **Roberto Segura**

6.7. Modelo cinemático del Novint Falcon

6.7.1. Análisis geométrico

Geoméricamente, el mecanismo del dispositivo háptico puede ser analizado como un robot paralelo con forma triangular en la base y en el extremo de las articulaciones, y se requieren tres sistemas de referencias para sus análisis, donde A es el sistema de referencia de la base de los tres brazos, C es el sistema de referencia del espacio de trabajo; y B_{1-2-3} son los sistemas de referencia relativos a cada brazo del dispositivo háptico, como se muestra en la Fig. 6.7.

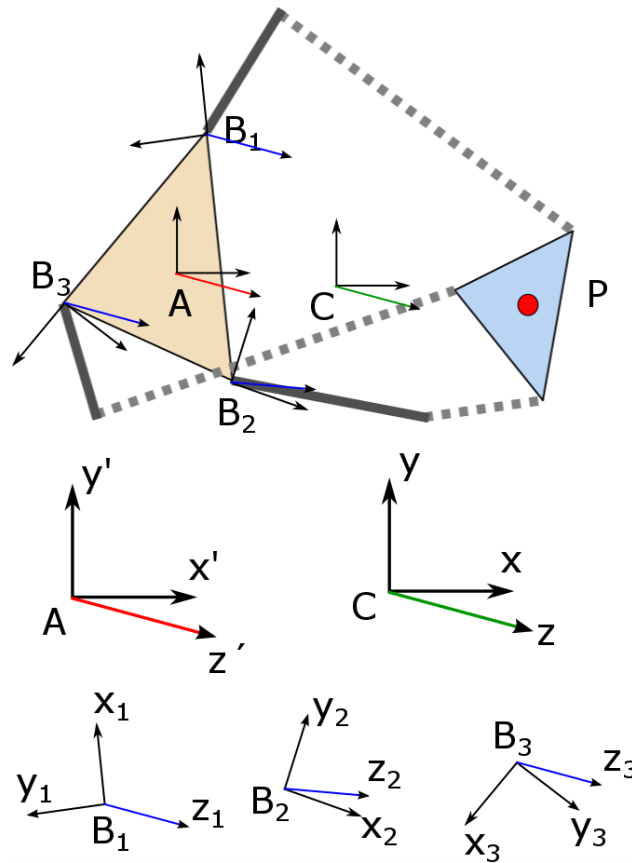


Fig. 6.7: Sistemas de referencia del dispositivo Háptico

Elaborado por: **Roberto Segura**

El punto de interés en el robot paralelo se encuentra en P , que es el extremo para los tres brazos. La ecuación (1) presenta una relación matemática para la posición P respecto a los sistemas de referencia A y C , donde sus elementos son componentes vectoriales como presenta la ecuación 2. Por conocimiento dado por el fabricante, $Z_o = 151\text{ mm}$ y $X_o = Y_o = 0\text{ mm}$.

$$\mathbf{P}_A = \mathbf{P}_C + \mathbf{P}_{C_0} \quad (1)$$

$$\begin{pmatrix} P_{x'} \\ P_{y'} \\ P_{z'} \end{pmatrix} = \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} + \begin{pmatrix} X_o \\ Y_o \\ Z_o \end{pmatrix} \quad (2)$$

Luego, es posible relacionar la posición de P desde los sistemas de referencia A y B_{1-2-3} de manera generalizada usando los desplazamientos r y s de la base triangular, y una matriz de rotación con los ángulos para cada brazo como se tiene en las ecuaciones (3) y

(4), con los ángulos visibles en la Fig. 6.8 literal d, los valores de los ángulos son: $Q_1 = 105$, $Q_2 = 345$ y $Q_3 = 225$.

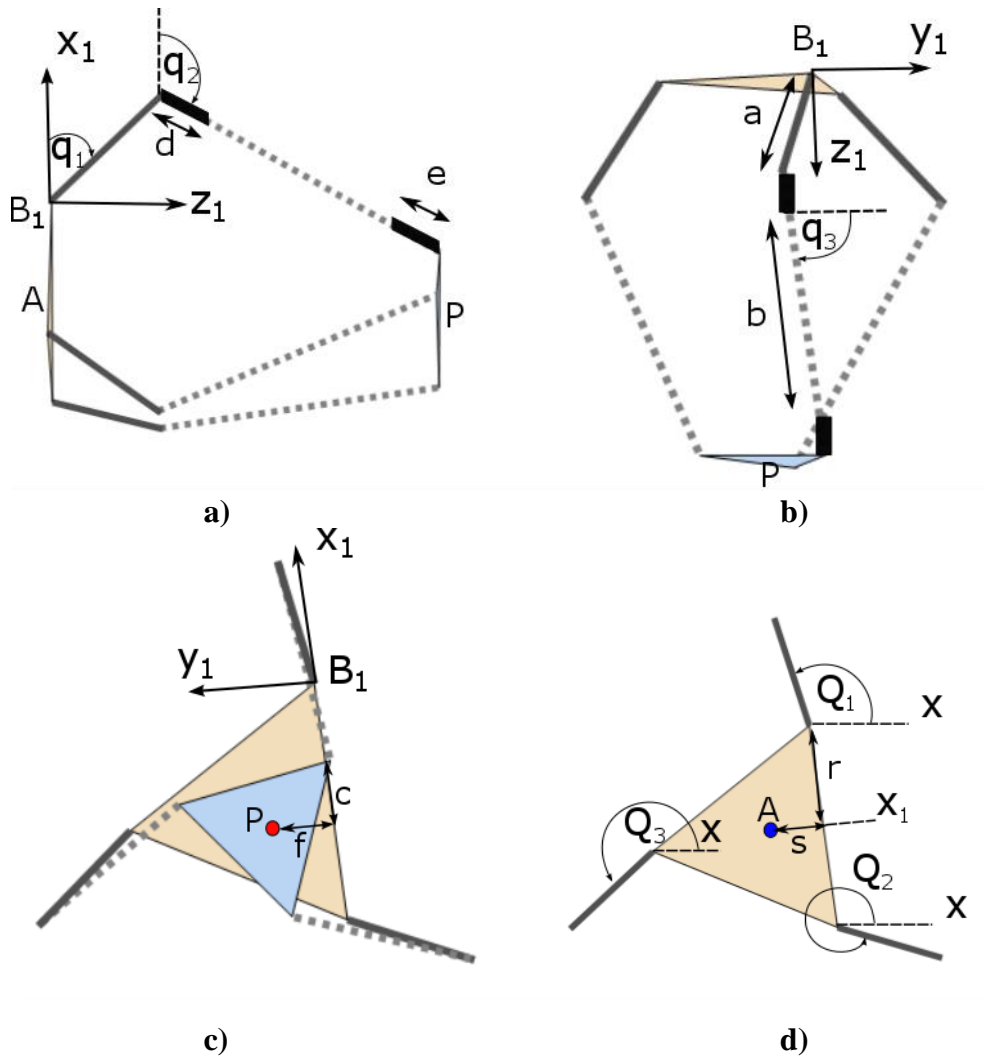


Fig. 6.8: Parámetros geométricos del dispositivo Novint Falcon: a) vista lateral, b) vista superior, c) vista frontal, d) vista trasera

Elaborado por: **Roberto Segura**

$$\mathbf{P}_{B_i} = \mathbf{R}\mathbf{P}_A + \mathbf{d} \quad (3)$$

$$\begin{bmatrix} P_{xi} \\ P_{yi} \\ P_{zi} \end{bmatrix} = \begin{bmatrix} \cos Q_i & \sin Q_i & 0 \\ -\sin Q_i & \cos Q_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{x'} \\ P_{y'} \\ P_{z'} \end{bmatrix} + \begin{bmatrix} -r \\ s \\ 0 \end{bmatrix} \quad (4)$$

A partir de la simplificación del modelo geométrico por cada brazo, es posible definir la localización de extremo operativo P, el cual es el mismo para cada brazo respecto a la referencia B_i mediante sus componentes vectoriales, del análisis de la Fig. 6.8 (a, b, c y d) se obtiene las ecuaciones (5), (6) y (7).

$$P_{xi} = a \cos q_1 - c + (d + e + b \sin q_3) \cos q_2 \quad (5)$$

$$P_{yi} = b \cos q_3 + f \quad (6)$$

$$P_{zi} = a \sin q_1 + (d + e + b \sin q_3) \sin q_2 \quad (7)$$

Combinando las ecuaciones 2, 4-7, se encuentra la expresión de la ecuación (8), de forma resumida según la ecuación (9).

$$\begin{bmatrix} a \cos q_1 - c + (d + e + b \sin q_3) \cos q_2 \\ b \cos q_3 + f \\ a \sin q_1 + (d + e + b \sin q_3) \sin q_2 \end{bmatrix} = \begin{bmatrix} \cos Q_i & \sin Q_i & 0 \\ -\sin Q_i & \cos Q_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} \right) + \begin{bmatrix} -r \\ s \\ 0 \end{bmatrix} \quad (8)$$

$$\mathbf{P}_{B_i} = \mathbf{R}(\mathbf{P}_C + \mathbf{P}_{C_o}) + \mathbf{d} \quad (9)$$

Reordenando esta expresión se puede encontrar la posición de P respecto al sistema de referencia del espacio de trabajo, según la ecuación (10), este modelo es útil para cada brazo y permite posicionar el extremo mediante los ángulos que conforman el sistema geométrico.

$$\mathbf{P}_C = \mathbf{R}^{-1}(\mathbf{P}_{B_i} - \mathbf{d}) - \mathbf{P}_{C_o} \quad (10)$$

A. Geometría inversa

La inversa de modelo geométrico es aplicable cuando se requiere conocer los ángulos para lograr una posición dada para el extremo operativo. Partiendo de la ecuación (6) se logra despejar el ángulo q_3 como presenta la ecuación (11).

$$\theta_3 = \cos^{-1} \left(\frac{P_{yi} - f}{b} \right) \quad (11)$$

Reordenando las ecuaciones (5) y (7), se obtienen la ecuaciones (12) y (13), que resume la expresión $D = (d + e + b \sin q_3)$.

$$a \cos q_1 + D \cos q_2 = c + P_{xi} \quad (12)$$

$$a \sin q_1 + D \sin q_2 = P_{zi} \quad (13)$$

Con el propósito de eliminar q_1 se ordena y se eleva al cuadrado las expresiones de las ecuaciones (12) y (13), como se presentan en las ecuaciones (14) y (15).

$$(a \cos q_1)^2 = (c + P_{xi} + D \cos q_2)^2 \quad (14)$$

$$(a \sin q_1)^2 = (P_{zi} - D \sin q_2)^2 \quad (15)$$

Resolviendo las ecuaciones (14) y (15), se encuentran las ecuaciones (16) y (17), que se proceden a sumar para reducirlas a una sola expresión.

$$a^2 \cos^2 q_1 = (c + P_{xi})^2 - 2(c + P_{xi})D \cos q_2 + D^2 \cos^2 q_2 \quad (16)$$

$$a^2 \sin^2 q_1 = P_{zi}^2 - 2P_{zi}D \sin q_2 + D^2 \sin^2 q_2 \quad (17)$$

Unificando las ecuaciones (16) y (17), y considerando $\cos^2 q + \sin^2 q = 1$, se determina la ecuación (18).

$$a^2 = (c + P_{xi})^2 + P_{zi}^2 - 2(c + P_{xi})D \cos q_2 - 2P_{zi}D \sin q_2 + D^2 \quad (18)$$

Reordenando la ecuación (18) tenemos la expresión de la ecuación 19.

$$E \cos q_2 = F - G \sin q_2 \quad (19)$$

Donde,

$$E = 2(c + P_{xi})D$$

$$G = 2P_{zi}D$$

$$F = (c + P_{xi})^2 + P_{zi}^2 + D^2 - a^2$$

Para simplificar las expresiones trigonométricas, se eleva al cuadrado la expresión de la ecuación (19), hallando la ecuación (20).

$$E^2 \cos^2 q_2 = F^2 - 2FG \sin q_2 + G^2 \sin^2 q_2 \quad (20)$$

Considerando que $\cos^2 q_2 = 1 - \sin^2 q_2$, y aplicando a la ecuación (20), tenemos la expresión de la ecuación (21).

$$E^2 (1 - \sin^2 q_2) = F^2 - 2FG \sin q_2 + G^2 \sin^2 q_2 \quad (21)$$

Reagrupando la ecuación (21), se obtiene la expresión de segundo grado de la ecuación (22).

$$H \sin^2 q_2 + I \sin q_2 + J = 0 \quad (22)$$

Donde,

$$H = G^2 + E^2$$

$$I = -2GF$$

$$J = F^2 - E^2$$

Entonces, se puede aplicar la ecuación de segundo grado de la ecuación (23).

$$\sin q_2 = \frac{-I \pm \sqrt{I^2 - 4HJ}}{2H} \quad (23)$$

Al determinar q_2 , el valor de q_1 puede hallarse mediante alguna de las ecuaciones (12) y (13).

B. Análisis cinemático

El análisis cinemático utiliza las velocidades de la cadena cinemática para describir el movimiento del extremo operativo. Partiendo de la derivada de la ecuación 1, obtenemos la ecuación (23).

$$\dot{\mathbf{P}}_A = \dot{\mathbf{P}}_C \quad (23)$$

Además, la derivada temporal de la ecuación 9, resulta en la expresión de la ecuación 24.

$$\dot{\mathbf{P}}_{Bi} = \mathbf{R} \dot{\mathbf{P}}_A \quad (24)$$

Combinando las ecuaciones 23 y 24 se encuentra la ecuación 25.

$$\dot{\mathbf{P}}_{Bi} = \mathbf{R} \dot{\mathbf{P}}_C \quad (25)$$

Adicional, podemos derivar las ecuaciones 5, 6 y 7, para determinar las velocidades del extremo operativo en función del movimiento de cada articulación, según las ecuaciones 26, 27 y 28.

$$\dot{P}_{xi} = -a \sin(q_1) \dot{q}_1 - (d + e + b \sin q_3) \sin q_2 \dot{q}_2 + b \cos q_3 \cos q_2 \dot{q}_3 \quad (26)$$

$$\dot{P}_{yi} = -b \sin q_3 \dot{q}_3 \quad (27)$$

$$\dot{P}_{zi} = a \cos(q_1) \dot{q}_1 + (d + e + b \sin q_3) \cos q_2 \dot{q}_2 + b \cos q_3 \sin q_2 \dot{q}_3 \quad (28)$$

Las ecuaciones 26, 27 y 28, pueden ser resumidas en la expresión matricial de las ecuaciones 29 y 30.

$$\begin{bmatrix} \dot{P}_{xi} \\ \dot{P}_{yi} \\ \dot{P}_{zi} \end{bmatrix} = \begin{bmatrix} -a \sin(q_1) & -\sin q_2 (d + e + b \sin q_3) & b \cos q_3 \cos q_2 \\ 0 & 0 & -b \sin q_3 \\ a \cos(q_1) & \cos q_2 (d + e + b \sin q_3) & b \cos q_3 \sin q_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \quad (29)$$

$$\mathbf{P}_{Bi} = \bar{\mathbf{J}} \dot{\mathbf{q}} \quad (30)$$

Uniendo las ecuaciones 25 y 30, encontramos el modelo cinemático de la ecuación 31, que permite obtener las velocidades del efector final de cada brazo a partir de las velocidades de las articulaciones.

$$\dot{\mathbf{P}}_c = \mathbf{R}^{-1} \bar{\mathbf{J}} \dot{\mathbf{q}}(t) \quad (31)$$

6.7.2. Método de control propuesto

Para lograr que el usuario ubicado en el sitio local tele-opere adecuadamente y cumpla con los objetivos de control, se propone una ley de control que logre notificar mediante una retroalimentación de fuerza el tipo de excitaciones que el usuario debe hacer con el dispositivo háptico. Para retroalimentar una fuerza adecuada se emplea el modelo cinemático obtenido de la ecuación (31), haciendo uso de la cinemática inversa y obteniendo

$$\dot{\mathbf{q}} = \mathbf{R} \bar{\mathbf{J}}^{-1} \dot{\mathbf{P}}_c$$

De esta forma, se plantea la ley de control denotada por la ecuación (32)

$$\dot{\mathbf{q}} = \mathbf{R} \bar{\mathbf{J}}^{-1} \left(\dot{\mathbf{P}}_{c_d} + PID(\tilde{\mathbf{P}}_c) \right) \quad (32)$$

donde la expresión PID representa el control para compensar el error que se produzca en realizar la tarea de desplazamiento y se define como:

$$PID(\tilde{\mathbf{P}}_c) = \mathbf{kp}(\tilde{\mathbf{P}}_c) + \mathbf{ki} \int_0^t (\tilde{\mathbf{P}}_c) dt + \mathbf{kd} \frac{d}{dt}(\tilde{\mathbf{P}}_c), \quad (33)$$

donde $\tilde{\mathbf{P}}_c$ representa el error de posición deseada versus la posición real del extremo operativo del robot háptico, definida como $\tilde{\mathbf{P}}_c = \mathbf{P}_{cd} - \mathbf{P}_c$; mientras que $\mathbf{kp}, \mathbf{ki}, \mathbf{kd}$ representan las ganancias del controlador PID, las cuales se deben ajustar dependiendo del tipo de aplicación.

6.7.3. Análisis de estabilidad del controlador

Para realizar el análisis de estabilidad del controlador propuesto se ve de manera que fuera un control proporcional representado de la siguiente manera basado en (32)

$$\dot{\mathbf{q}}(ref) = \mathbf{R}\bar{\mathbf{J}}^{-1} \left(\dot{\mathbf{P}}_{cd} + \mathbf{K}(\tilde{\mathbf{P}}_c) \right) \quad (34)$$

Donde \mathbf{K} representa la ganancia del controlador, en este caso se asume el seguimiento perfecto de la velocidad de tal forma que:

$$\dot{\mathbf{q}}(ref) = \dot{\mathbf{q}}(t) \quad (35)$$

Entonces tomando en cuenta (35), se reemplaza (34) en (31) se obtiene la siguiente expresión:

$$\dot{\mathbf{P}}_c = \mathbf{R}^{-1}\bar{\mathbf{J}}\mathbf{R}\bar{\mathbf{J}}^{-1} \left(\dot{\mathbf{P}}_{cd} + \mathbf{K}(\tilde{\mathbf{P}}_c) \right) \quad (36)$$

La multiplicación de las siguientes matrices viene a dar como resultado una matriz identidad $\mathbf{R}^{-1}\bar{\mathbf{J}}\mathbf{R}\bar{\mathbf{J}}^{-1} = \mathbf{I}$, por lo tanto (36) se escribe como:

$$\dot{\mathbf{P}}_c = \dot{\mathbf{P}}_{cd} + \mathbf{K}(\tilde{\mathbf{P}}_c) \quad (37)$$

El error de control $\tilde{\mathbf{P}}_c$ viene dado por:

$$\tilde{\mathbf{P}}_c = \mathbf{P}_{cd} - \mathbf{P}_c \quad (38)$$

ahora derivamos (38):

$$\frac{\partial(\tilde{\mathbf{P}}_c)}{\partial(t)} = \frac{\partial(\mathbf{P}_{Cd} - \mathbf{P}_c)}{\partial(t)} \quad (39)$$

y se obtenemos la siguiente expresión:

$$\dot{\tilde{\mathbf{P}}}_c = \dot{\mathbf{P}}_{Cd} - \dot{\mathbf{P}}_c \quad (40)$$

Si se reemplaza (40) en (37):

$$\begin{aligned} \cancel{\dot{\mathbf{P}}_{Cd}} - \dot{\tilde{\mathbf{P}}}_c &= \cancel{\dot{\mathbf{P}}_{Cd}} + \mathbf{K}(\tilde{\mathbf{P}}_c) \\ \dot{\tilde{\mathbf{P}}}_c &= -\mathbf{K}(\tilde{\mathbf{P}}_c) \end{aligned} \quad (41)$$

La ecuación (41) representa una ecuación en lazo cerrado del sistema.

Mediante el análisis de Liapunov se realiza el siguiente análisis en el cual se determina mediante la propuesta de la ecuación del error cuadrático representado como:

$$\mathbf{q}(\tilde{\mathbf{h}}) = \frac{1}{2} \tilde{\mathbf{P}}_c^T \tilde{\mathbf{P}}_c \quad (42)$$

Derivando la ecuación (42) se obtiene:

$$\dot{\mathbf{q}}(\tilde{\mathbf{h}}) = \frac{1}{2} \left(\dot{\tilde{\mathbf{P}}}_c^T \tilde{\mathbf{P}}_c + \tilde{\mathbf{P}}_c^T \dot{\tilde{\mathbf{P}}}_c \right) = \left(\dot{\tilde{\mathbf{P}}}_c^T \tilde{\mathbf{P}}_c \right) \quad (43)$$

Reemplazando la ecuación de lazo cerrado (41) en (43) se tiene que:

$$\dot{\mathbf{q}}(\tilde{\mathbf{h}}) = -\tilde{\mathbf{P}}_c^T \mathbf{k} \tilde{\mathbf{P}}_c \quad (44)$$

Para que el sistema sea estable se debe cumplir que $\dot{\mathbf{q}}(\tilde{\mathbf{h}}) < 0$, para que se cumpla esto $\mathbf{k} > 0$, es decir que es un valor positivo de constante, entonces en controlador PID propuesto con ganancia \mathbf{k} funcionara de manera correcta, las constantes que se coloque dentro del controlador se debe ir probando y observando el comportamiento del sistema hasta que se defina la ideal, no necesariamente se debe realizar una sintonización mediante fórmulas ya que el controlador PID en función de la cinemática inversa del modelo obtenido asegura de una u otra forma el seguimiento perfecto de trayectoria.

6.7.4. Sintonización del control PID

Para realizar la sintonización del controlador PID se emplea el método de Ziegler y Nichols, el cual permite obtener los valores apropiada para las ganancias proporcional, integral y derivativa del PID. La sintonización se realiza en lazo cerrado, iniciando con la obtención de una ganancia crítica denominada K_c y el tiempo de oscilación t_c que produce en el sistema. Esta prueba se hace únicamente con una ganancia proporcional, logrando obtener una oscilación periódica para poder determinar los parámetros mencionados. La obtención de una oscilación periódica del proceso permite emplear la Tabla 6-1, de manera que se logren determinar los parámetros adecuados de las ganancias del controlador.

Tabla 6-1: Parámetros establecidos para encontrar los valores de PID

	K_p	K_i	K_d
Control P	$0.5 K_c$		
Control PI	$0.45 K_c$	$0.54 \frac{K_p}{t_c}$	
Controlador PID	$0.6 K_c$	$2 \frac{K_p}{t_c}$	$0.125 K_p t_c$

Este método sirve para poder colocar las constantes en el controlador de una manera teórica y no al azar, es decir mediante esta tabla se determina cuáles son los valores apropiados para las constantes del controlador.

6.8. Conexión de Novint Falcon a Matlab

La empresa creadora de Novint Falcon facilita todo el software para la comunicación entre la computadora y el dispositivo háptico. Esto incluye los drives para la adquisición de información de los encoders de cada uno de los tres motores, así como la identificación del botón presionado. Por su parte, el controlador del dispositivo permite modificar las velocidades de los motores para posicionar el punto de interés en una posición dentro del espacio de trabajo resultante. De la misma manera, la empresa fabricante ofrece un demo y un conjunto de videojuegos para la interacción con el dispositivo háptico, intentando demostrar la usabilidad de este dispositivo. A pesar de las facilidades obtenidas por parte

de los creadores, un paquete de software de desarrollo no se facilita al usuario final, impidiendo desarrollar aplicaciones personalizadas sobre el dispositivo.

La carencia de un SDK (Software Development Kit) reduce las opciones de tener el control y la información del sensor en cualquier software de desarrollo, sin embargo, existen opciones para extraer las características a través del uso de librerías de terceros. A pesar de tener librerías compatibles con sistemas operativos de 32 bits, el juego de archivos *.dll provisto por Haptik Library prometen estabilidad al momento de invocar las funciones de escritura/lectura de datos sobre el Novint Falcon. De esta manera, se siguen un conjunto de pasos para poder adaptar la librería al dispositivo en cuestión.

Los pasos para el uso de la librería han sido adecuadamente documentados, presentados en (Monte, 2018). En este, se detalla la importancia del uso de Matlab de 32 bits, debido a que la ejecución en una versión de 64 bits será no satisfactoria (para el caso de computadoras con sistema operativo x64). Asimismo, la invocación de librerías desde Matlab tiene ciertas restricciones para computadoras con sistemas operativos de 64 bits, donde la inicialización de las librerías desde Matlab debe ser llevada a cabo una única vez, ya que la desconexión del dispositivo o la eliminación del objeto en el software matemático llevará a errores de conexión que podrán ser solucionados al momento de reiniciar el ordenador.

El cumplimiento de ciertos requisitos permite que Matlab pueda escribir velocidades a cada uno de los motores con el objetivo de ubicar el extremo operativo en una posición deseada. Por su parte, las librerías permiten tener información de posicionamiento del extremo operativo dentro del software matemático a través de comandos simples, así como obtener la información de los botones presionados. De esta manera, se puede tener control y retroalimentación de información del dispositivo, dos requisitos necesarios para la implementación de controladores autónomos. La Fig. 6.9 muestra cómo se conecta el dispositivo háptico al software matemático, las librerías usadas y las variables que se manejan.

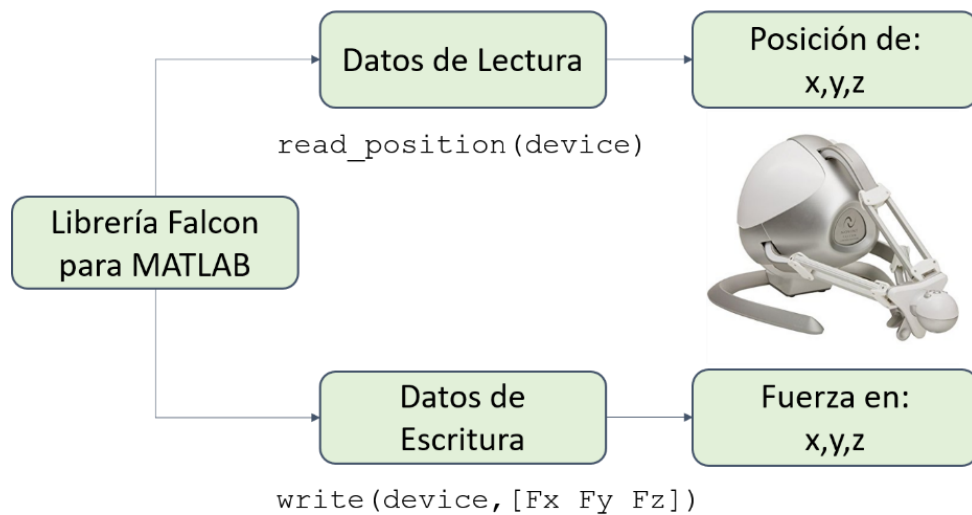


Fig. 6.9: Lectura/escritura de datos de Novint Falcon

Elaborado por: **Roberto Segura**

Para crear la comunicación con el dispositivo háptico se crea un objeto “device” mediante la función “haptikdevice”, el cual va a contener la configuración adecuada para comunicarse con el equipo. La función “read_position(device)” entrega un vector con 3 valores, los cuales representan la posición espacial del extremo operativo del Falcon; mientras que la función “write(device,[Fx Fy Fz]) ” permite escribir una fuerza a cada brazo del dispositivo háptico, logrando así obtener una fuerza de percepción dependiendo el entorno y del control realizado.

6.9. Validación del modelo matemático

Para realizar la validación del modelo matemático se realiza simulaciones y pruebas experimentales con la ley de control propuesta. Tanto el modelo matemático como el método de control propuesto permiten simular el comportamiento del robot en un software matemático. Las características de Matlab previamente establecidas lo muestran como el software matemático más apropiado para comprobar el algoritmo de control establecido, donde el mismo adicionalmente permite simular características cinemáticas del robot.

La Fig. 6.10 muestra la construcción simulada del dispositivo Falcon Novint, todo conforme a la hoja de especificaciones mostradas por el fabricante (Anexos). La simulación se lleva a cabo a través de herramientas de ploteo, funciones de graficación y uso de matrices y vectores.

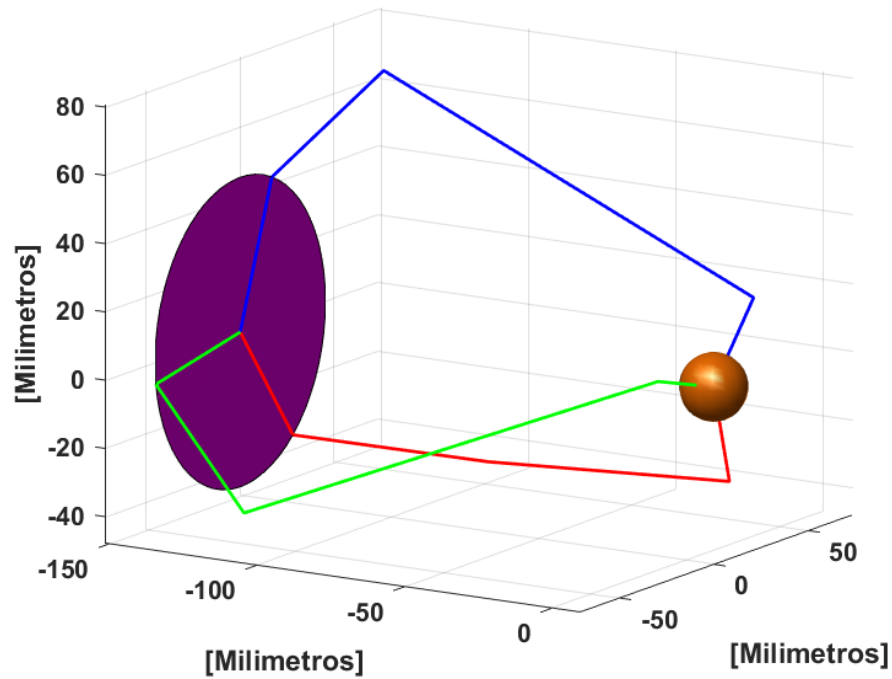


Fig. 6.10: Simulación del Novint Falcon

Elaborado por: **Roberto Segura**

6.9.1. Primera prueba del controlador

El objeto simulado que representa al Novint Falcon permite validar tanto la modelación como el controlador PID propuesto. A continuación, se demuestra el funcionamiento del controlador mediante el seguimiento de dos trayectorias. La primera trayectoria está descrita por $x_d = r * \cos(p)$, $y_d = (2 * \pi) * 4$ y $z_d = \pi * \sin(p)$, componentes que forman una circunferencia la cual debe seguir el punto de interés. La Fig. 6.10 muestra el movimiento estroboscópico del sistema robótico para cumplir la trayectoria deseada, donde se muestra el camino a seguir y la ejecución que el extremo operativo ejecuta. De esta manera, se demuestra la tendencia a cero de los errores de control (Fig. 6.11) dada por las acciones de control presentadas en la Fig. 6.12, en el Anexo 2 se puede apreciar el programa desarrollado en Matlab para realizar el control, las constantes que se establecieron para el control de acuerdo a la Tabla 6-1 con $K_c = 0.833$ y $t_c = 3$ se obtiene $K_p = 0.5$, $K_i = 0.33$ y $K_d = 0.3$.

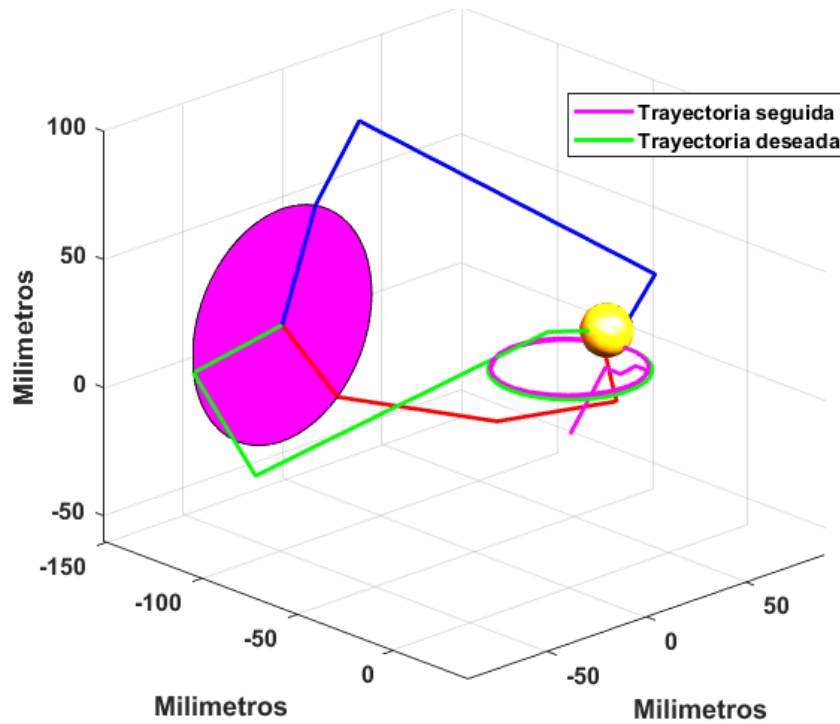


Fig. 6.11: Ejecución de la primera trayectoria

Elaborado por: **Roberto Segura**

A continuación, en la Fig. 6.12 se indica los errores producidos del controlador.

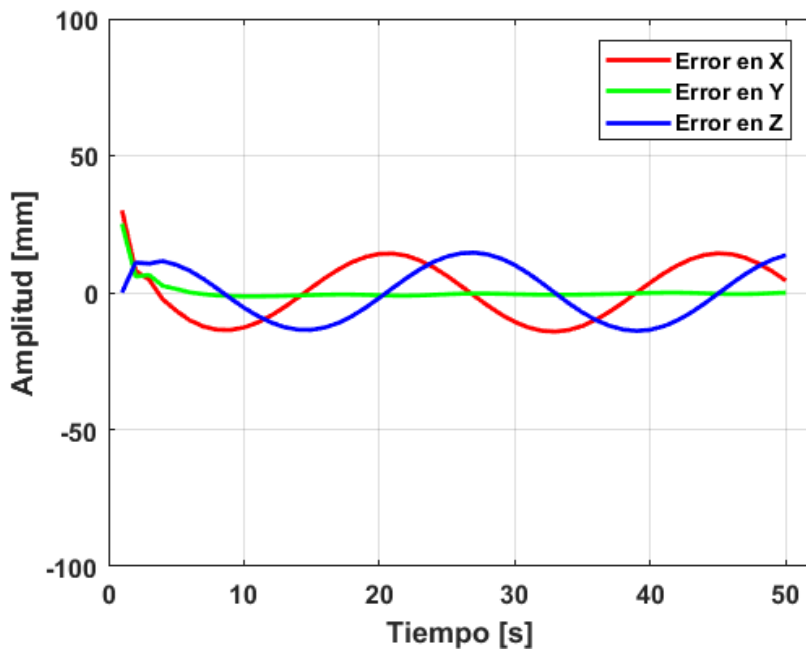


Fig. 6.12: Errores en cada una de las componentes para la ejecución de la primera trayectoria

Elaborado por: **Roberto Segura**

Las acciones de control se indican en la Fig. 6.13, los cuales mediante simulación son muy aproximados a los que se debería aplicar al robot háptico.

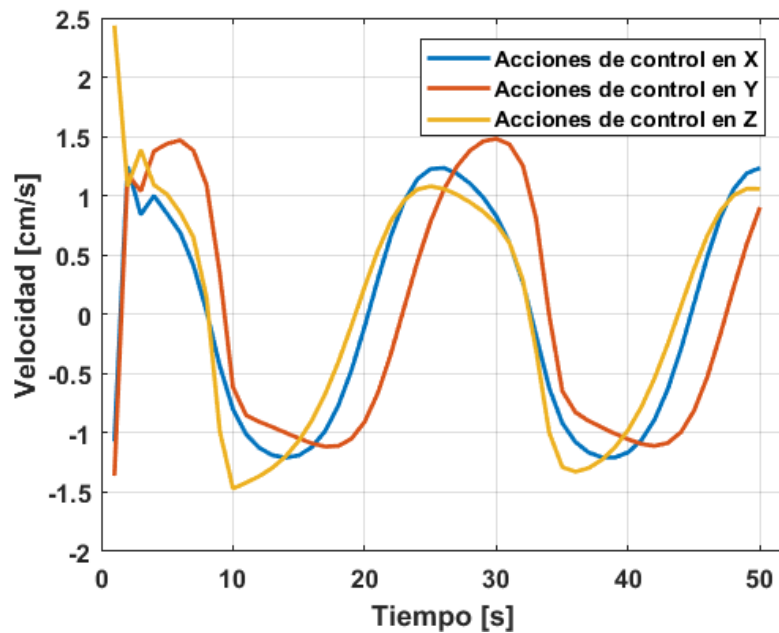


Fig. 6.13: Acciones de control para la ejecución de la primera trayectoria

Elaborado por: **Roberto Segura**

A. Experimentación de la primera prueba del controlador

Para el desarrollo de la experimentación primero se procede a realizar el ajuste de las constantes a través del método de Ziegler Nichols, por medio de la búsqueda de K_c y T_c , donde K_c se define como el valor incremental que se encuentra a través de experimento hasta lograr que la señal de respuesta oscile moderadamente sobre el objetivo (Setpoint); mientras que T_c es el tiempo de oscilación de la señal. Basados en el método de Ziegler Nichols, la oscilación se logra a través de experimentación cuando $K_c = 0.1695$, con un tiempo de Oscilación de 0.6667 segundos, como se muestra en la Figura 6.14. A través de estos valores y la Tabla, los valores de las constantes resultantes son:

$$K_p = 0.6 * K_c = 0.1$$

$$K_i = 2 (K_p / T_c) = 0.301$$

$$K_d = 0.125 * K_p * T_c = 0.0085$$

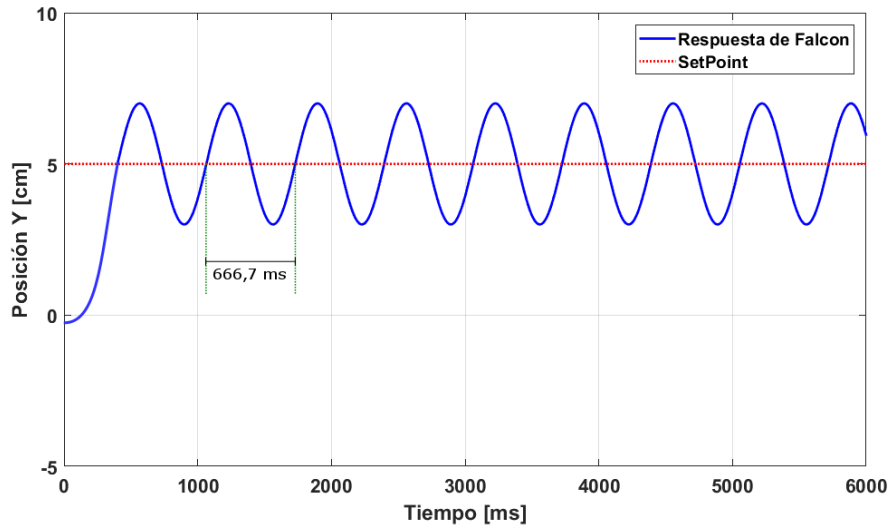


Fig. 6.14: Ejecución experimental de la trayectoria definida.

Elaborado por: **Roberto Segura**

Para realizar la validación ya del control experimental con el dispositivo háptico se realiza el seguimiento de trayectoria anteriormente simulado, en la Fig. 6.15, en el **Anexo 3** se presenta el programa desarrollador para realizar esta prueba las constantes del PID se ajustaron de tal manera que se tienen como $K_p = 0.1$, $K_i = 0.3$ y $K_d = 0.01$, al ser ya experimentación las constantes anteriormente simuladas se pueden variar ya que se presenta variaciones en la dinámica del dispositivo.

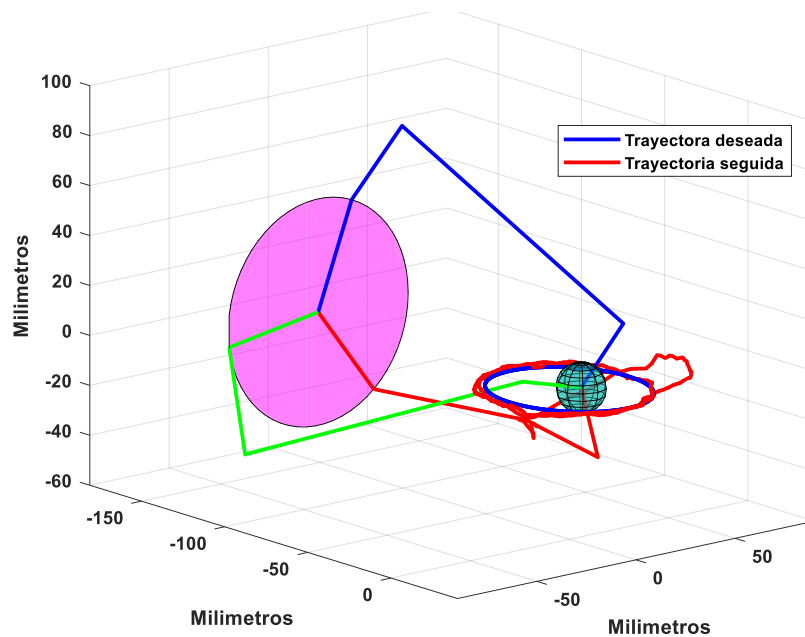


Fig. 6.15: Ejecución experimental de la trayectoria definida.

Elaborado por: **Roberto Segura**

Los errores producidos experimentalmente con el dispositivo háptico se presentan en la Fig. 6.16, observando de tal manera como tienden a ser cero, es decir, el controlador propuesto funciona correctamente logrando que el error tienda ser cero.

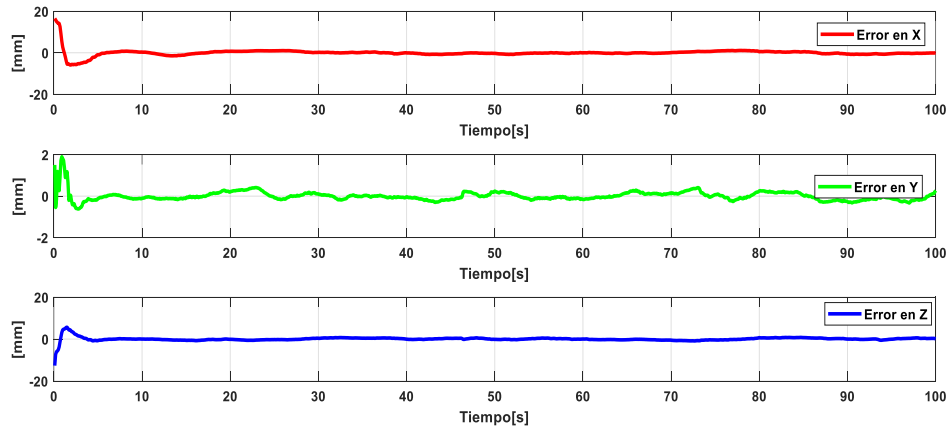


Fig. 6.16: Errores de control para x,y,z del experimento realizado.

Elaborado por: **Roberto Segura**

Las acciones que genera el controlador para realizar la tarea propuesta se presentan en la Fig. 6.17.

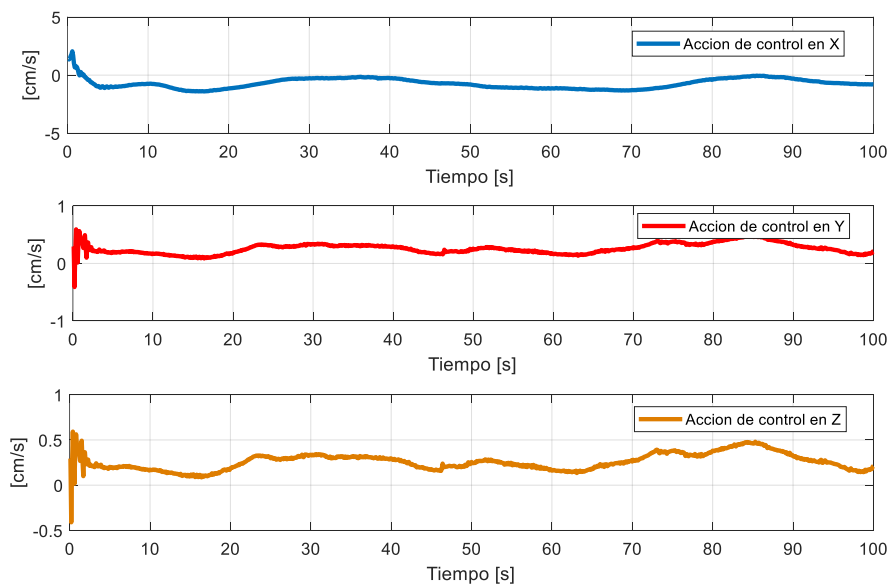


Fig. 6.17: Acciones de control para el Falcon en [cm/s].

Elaborado por: **Roberto Segura**

6.9.2. Segunda prueba del controlador.

B. Simulación

Para el segundo experimento se propone una trayectoria de la forma: $x_d = r * \cos(p)$, $y_d = (p - 2 * \pi) * 4_y$ $z_d = 2 * \pi * \sin(p)$; componentes que forman una circunferencia en los ejes X-Y, mientras incrementa su altura en el eje Z. El comportamiento del controlador se muestra en la Fig. 6.18. donde se presenta el seguimiento de la trayectoria planteada, el programa se presenta en **Anexo 3**.

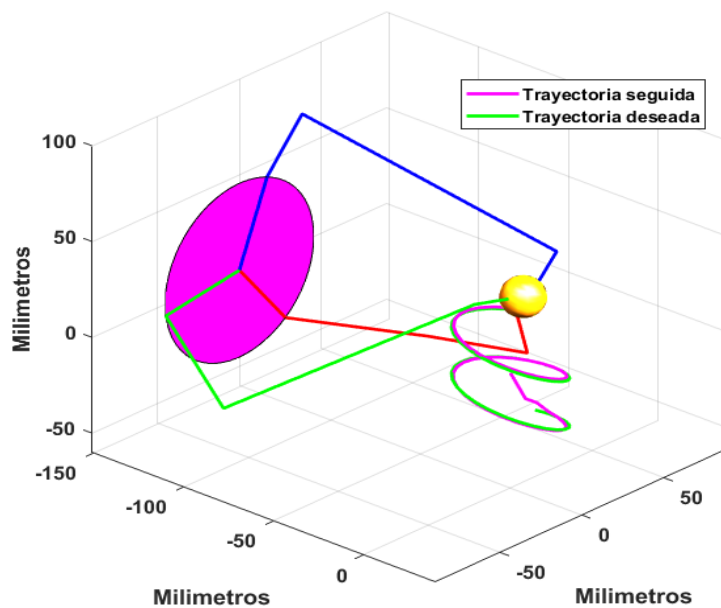


Fig. 6.18: Ejecución de trayectoria para el segundo experimento

Elaborado por: **Roberto Segura**

Los errores producidos por la simulación se pueden ver en la Fig. 6.19, como se observa, funciona adecuadamente mediante la simulación, entonces una vez probada por simulación se realiza el proceso experimental.

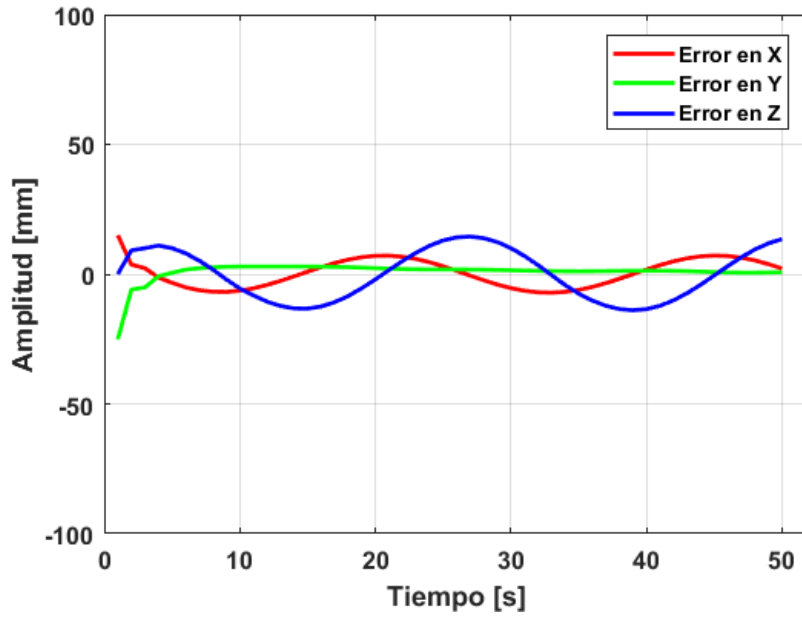


Fig. 6.19: Errores en cada una de las componentes para la ejecución de la segunda trayectoria

Elaborado por: **Roberto Segura**

Las acciones de control, se presentan en Fig.6.20 las cuales son arrojadas del controlador para la simulación de la tarea preparada.

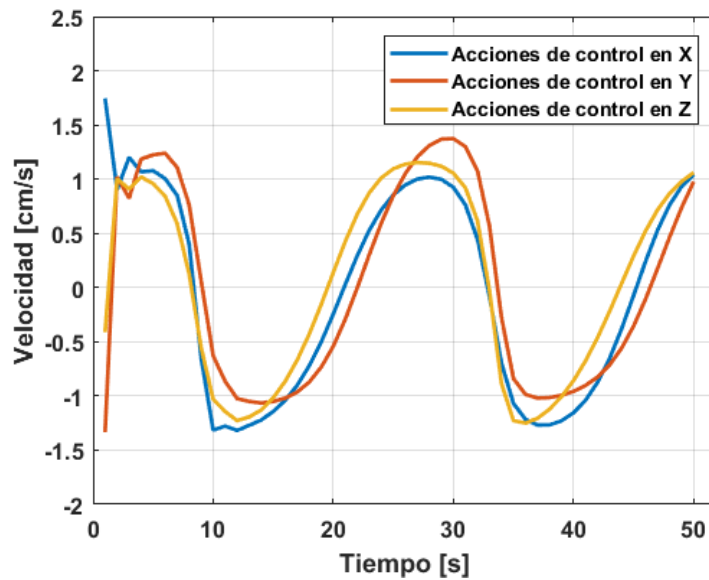


Fig. 6.20: Acciones de control para la ejecución de la segunda trayectoria

Elaborado por: **Roberto Segura**

B. Experimentación

Para validar el funcionamiento del control se realiza otra prueba experimental, trazando la trayectoria simulada, en la Fig. 6.21, se observa la trayectoria realizada por el Falcon, el **Anexo 3** se presenta el programa desarrollador para realizar esta prueba con la variación de la segunda trayectoria deseada, las constantes del PID se ajustaron de tal manera que se tienen como, $K_p = 0.1$, $K_i = 0.3$ y $K_d = 0.01$, al ser ya experimentación las constantes anteriormente simuladas se pueden variar ya que se presenta variaciones en la dinámica del dispositivo.

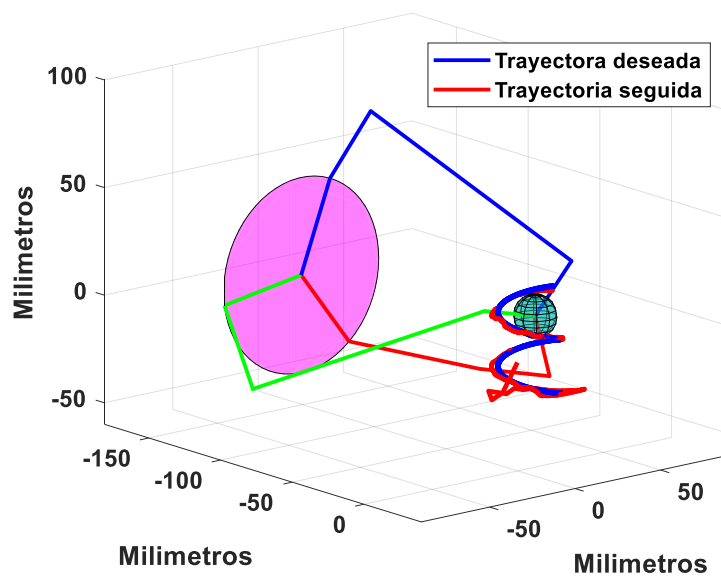


Fig. 6.21: Acciones de control para la ejecución de la segunda trayectoria

Elaborado por: **Roberto Segura**

En la Fig. 6.22 se muestra los errores de control obtenidos de la experimentación, se puede apreciar como tienden a ser cero en todo instante de tiempo, de esta manera se valida que el control funciona correctamente y la ganancia propuesta para este experimento.

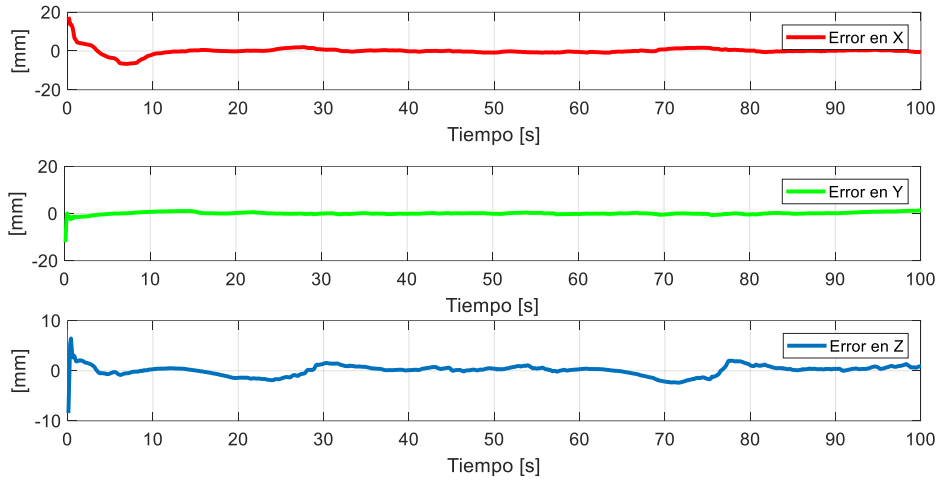


Fig. 6.22: Errores de control producidos por el control mediante la implementación del dispositivo háptico.

Elaborado por: **Roberto Segura**

Las acciones de control producidas por el controlador en la trayectoria planteada se muestran Fig. 6.23, observando como varían de acuerdo a la tarea que está ejecutando.

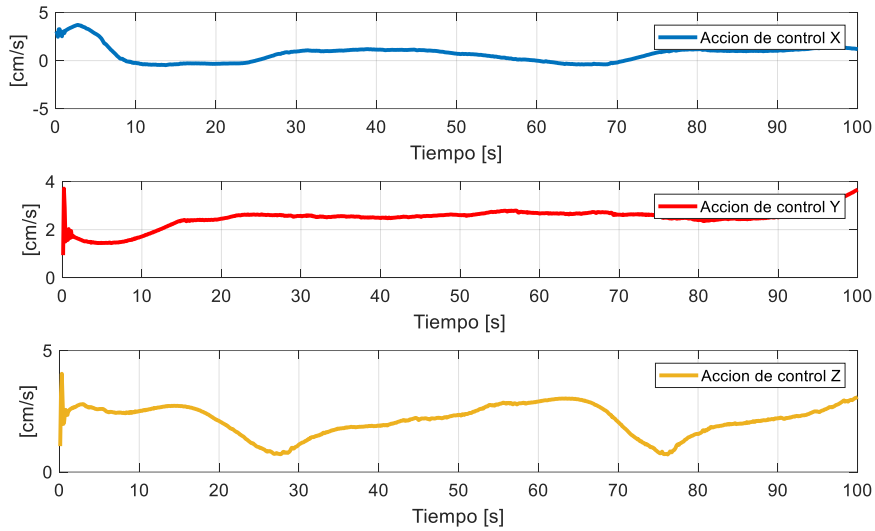


Fig. 6.23: Acciones de control para la ejecución de la segunda trayectoria

Elaborado por: **Roberto Segura**

Con el propósito de realizar y validar de mejor manera el modelo del dispositivo háptico Falcon se plantea desarrollar tareas de tele-operación en las que el operador maneje a robots mediante el dispositivo háptico retroalimentado de fuerzas para guiar el seguimiento perfecto de tarea propuesta, para realizar estos experimentos se lo hace

mediante un entorno de simulación de manejo de robots V-REP, enviando las señales de control desde MATLAB.

6.9.3. Interacción entre MATLAB y V-REP

En el desarrollo de la aplicación, el paso principal es adecuar la configuración del software de simulación y el software matemático. Mediante esta configuración, se logra intercomunicar ambos paquetes para probar el funcionamiento del dispositivo háptico en el proceso de control. Para configurar las dos plataformas, a continuación se detalla el procedimiento que se lleva a cabo (Robologs, 2018).

A. Configurar Matlab

El primer paso es contar con las dos aplicaciones instaladas, una vez que se tenga los softwares se procede a copiar los siguientes archivos que se encuentran en la ruta **C:\ProgramFiles\V-REP3\V-REP_PRO_EDU\programming\remoteApiBindings\matlab\matlab**, copiando los siguientes ficheros *.m que se encuentran en la carpeta de la dirección Fig. 6.24.

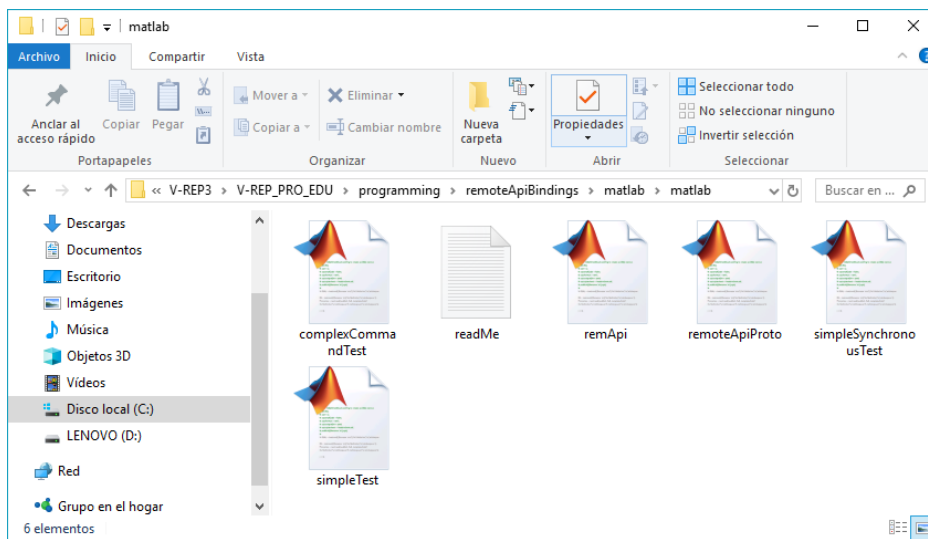


Fig. 6.24: Ubicación de archivos para el direccionamiento en Matlab

Elaborado por: **Roberto Segura**

Una vez copiado los ficheros, se direcciona hacia la carpeta donde se encuentran los archivos de la instalación de MATLAB en la ubicación **C:\Program Files\MATLAB\R2013b\bin** y dentro de esta se adhieren los ficheros. A continuación, es requerido dirigirse nuevamente a la ruta **C:\Program Files\V-REP3\V-REP_PRO_EDU**

programming\remoteApiBindings\lib\lib\Windows\64Bit, copiar el fichero existente dentro de esta Fig. 6.25 y se incluye en la carpeta **\bin** de MATLAB.

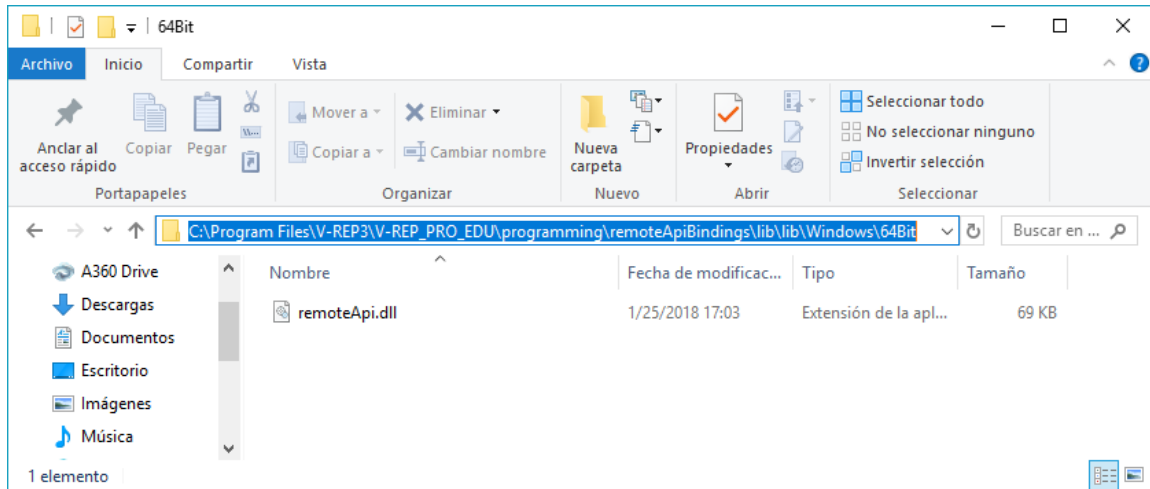


Fig. 6.25: Redirección de carpetas para la conexión con V-REP y Matlab

Elaborado por: **Roberto Segura**

B. Configuración de V-REP

Para el desarrollo de nuevas escenas de simulación dentro del programa y con el objetivo de que interactúe con otro paquete informático, es necesario configurar la escena de la manera indicada en la presente sección. Siempre que se cree un nuevo entorno de simulación, es requerido repetir el inicio de ejecución entre programas para lograr comunicarse con MATLAB. Al iniciar el programa de simulación, en la parte lateral izquierda seleccionar la pestaña de scripts Fig. 6.26.

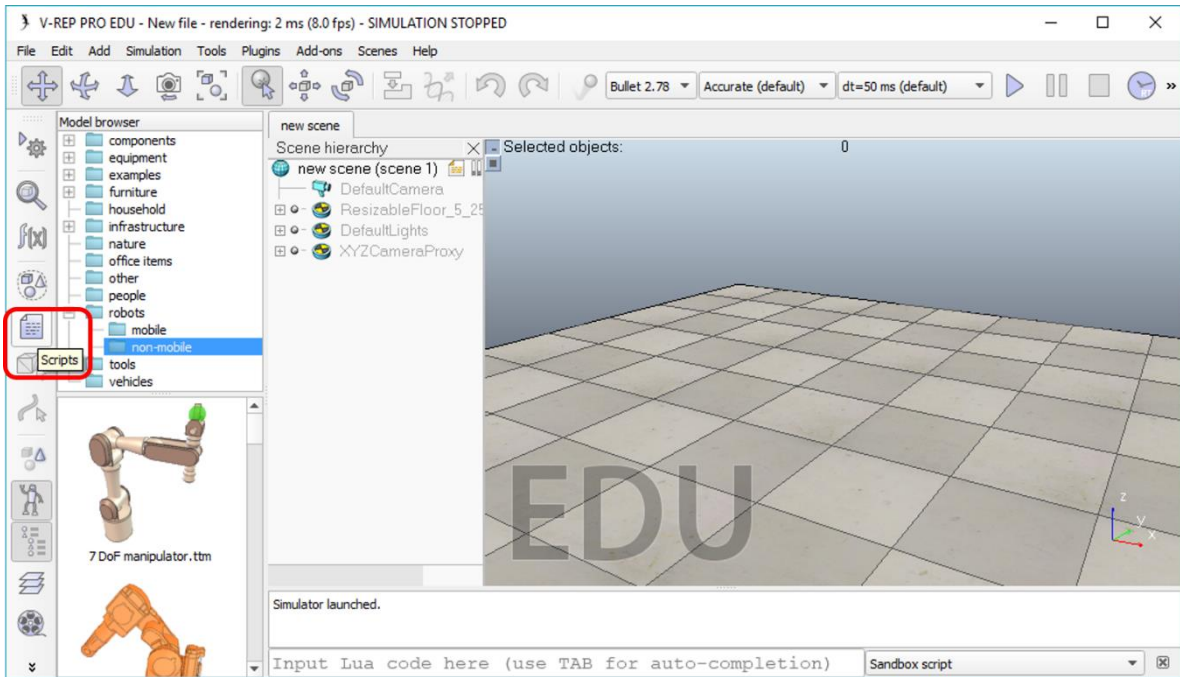


Fig. 6.26: Configuración de V-REP para seleccionar scripts

Elaborado por: **Roberto Segura**

El seguimiento de los pasos anteriores activará una ventana que contiene todos los scripts que hay en la escena de simulación, en el script seleccionado se crea una nueva escena remarcada de color rojo (Fig. 6.27), el cual es el encargado de gestionar los cálculos que tiene que realizar el programa al iniciar la simulación, tener en cuenta que este script no debe ser borrado ni modificado, caso contrario afectará a la simulación.

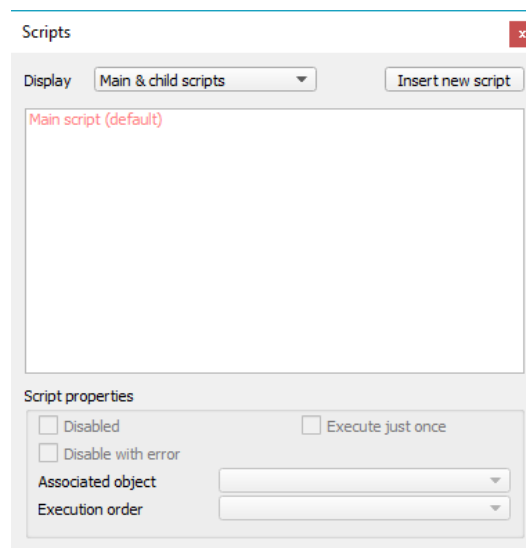


Fig. 6.27: Configuración del script para la simulación

Elaborado por: **Roberto Segura**

El siguiente paso es crear un nuevo script mediante la opción ‘**Insert new script**’, con lo cual aparecerán múltiples opciones de un nuevo script, pudiéndose seleccionar ‘**Child script (threaded)**’ como se muestra en la Fig. 6.28.

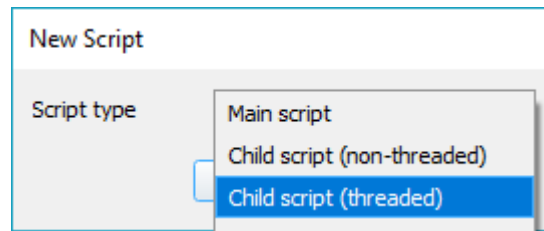


Fig. 6.28: Selección del child script
Elaborado por: **Roberto Segura**

La creación del nuevo script permite seleccionar la opción “**Associated Object**” para escoger un objeto de la escena. Este script habilita la comunicación entre el simulador y MATLAB, se asocia a objetos que no se vaya a destruir durante la ejecución de la simulación, en este caso el objeto **DefaultCamera**, que permite ver y moverse por la escena Fig. 6.29.

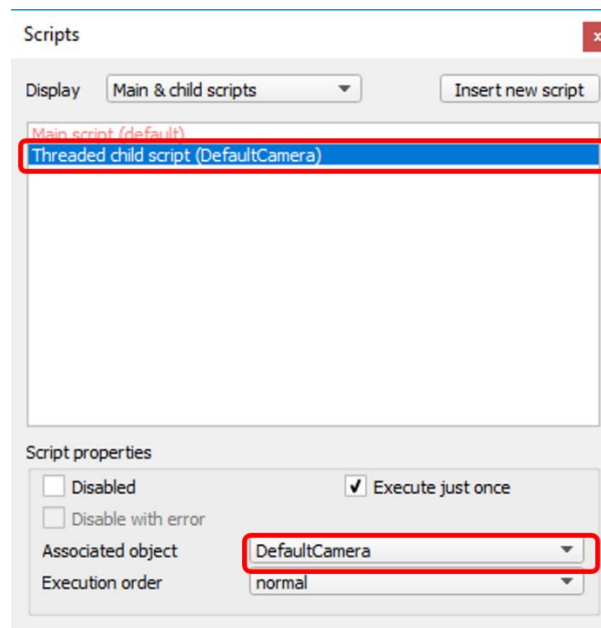
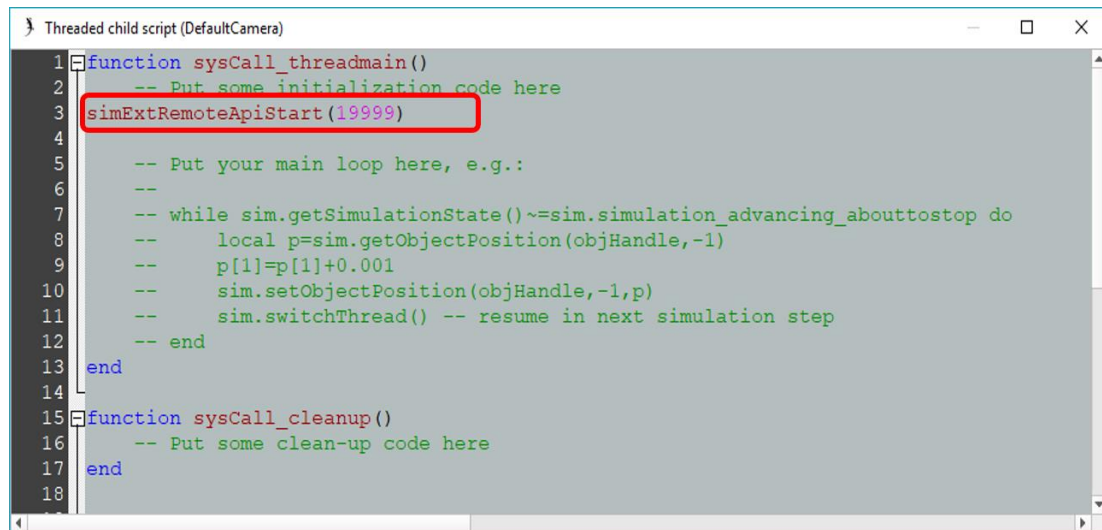


Fig. 6.29: Configuración de la cámara para visualizar el entorno
Elaborado por: **Roberto Segura**

La asociación de un nuevo script a un objeto permite iniciar su configuración. Por defecto, el script viene vacío con algunos comentarios que ayudan a ordenarlo. Para configurarlo para la comunicación con MATLAB se agrega la siguiente instrucción:

simExtRemoteApiStart(19999) al inicio de las líneas de código. Mediante esta instrucción, un puerto es asignado para ser el encargado de la comunicación, como se muestra en la Fig. 6.30.



```
1 function sysCall_threadmain()
2     -- Put some initialization code here
3     simExtRemoteApiStart(19999)
4
5     -- Put your main loop here, e.g.:
6     --
7     -- while sim.getSimulationState()~=sim.simulation_advancing_abouttostop do
8     --     local p=sim.getObjectPosition(objHandle,-1)
9     --     p[1]=p[1]+0.001
10    --     sim.setObjectPosition(objHandle,-1,p)
11    --     sim.switchThread() -- resume in next simulation step
12    -- end
13 end
14
15 function sysCall_cleanup()
16     -- Put some clean-up code here
17 end
18
```

Fig. 6.30: Adición de líneas de programación para la asignación de un puerto de comunicación

Elaborado por: **Roberto Segura**

Una vez efectuados los pasos anteriormente mencionados, se procede a verificar si la conexión es satisfactoria, de tal forma que ejecutamos la escena en V-REP y posteriormente ejecutamos en Matlab las siguientes líneas de comando:

```
vrep=remApi('remoteApi');
clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,5)
vrep.simxFinish(clientID);
```

La primera instrucción crea el objeto V-REP y carga la librería, mientras que la segunda línea de código inicia la conexión con el simulador y la última instrucción termina la conexión. La Fig. 6.31 muestra que la conexión entre las dos plataformas fue la ejecutada satisfactoriamente.

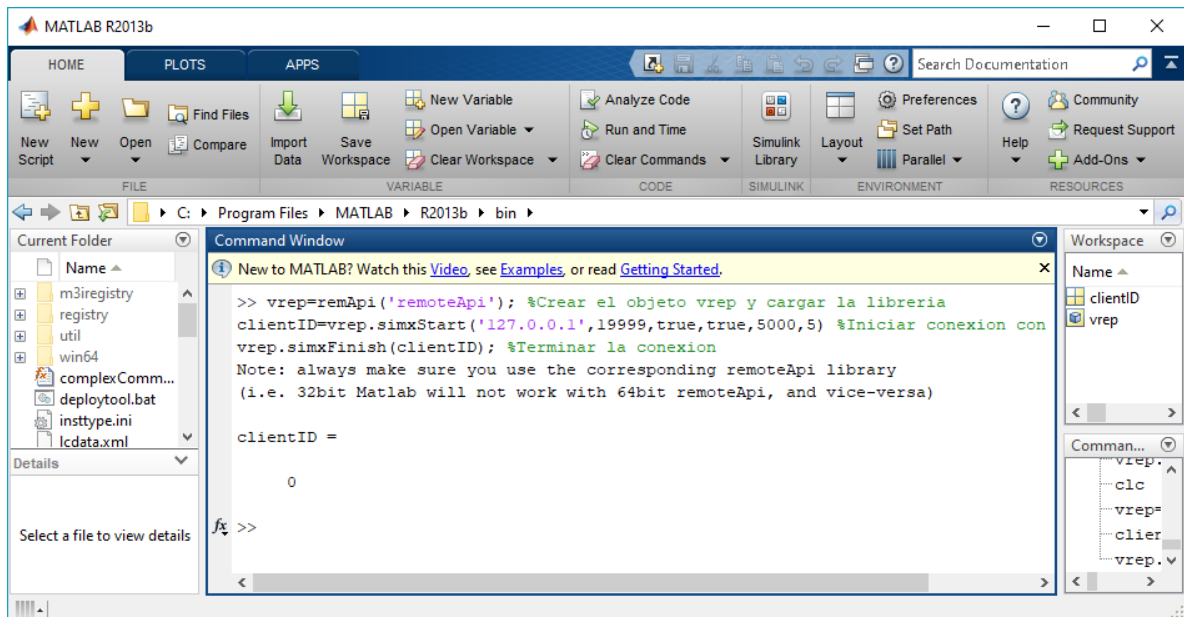


Fig. 6.31: Conexión satisfactoria entre la plataforma de Matlab y V-REP

Elaborado por: **Roberto Segura**

Nota: Tomar en consideración que, para crear un proyecto, los archivos pertenecientes a V-REP copiados en la carpeta `\bin` deben estar colocados dentro de la carpeta donde se almacene el código de Matlab, para que de una u otra forma Matlab lo reconozca y logre cargar la librería.

6.9.4. Desarrollo de escenarios en V-REP

El desarrollo de las escenas simuladas para interactuar entre Matlab y el dispositivo háptico Novint Falcon considera dos aplicaciones, la primera para conducir un robot de tipo diferencial dentro de un entorno en el que se encuentren obstáculos de diferentes magnitudes, y la segunda considera un brazo robótico de 4 grados de libertad para tareas de apilamiento, con el objetivo de obtener una respuesta de retroalimentación del entorno con el dispositivo háptico. El objetivo de la segunda tarea es retroalimentar el peso del objeto enganchado, el cual debe ser colocado sobre una banda transportadora para cumplir con tareas definidas por el programa.

A. Caso 1: Navegación Tele-Operada

La configuración de la escena como se indica en secciones previas, permite dar inicio a la comunicación entre Matlab y V-REP, con lo cual se procede a crear el entorno (Fig. 6.32).

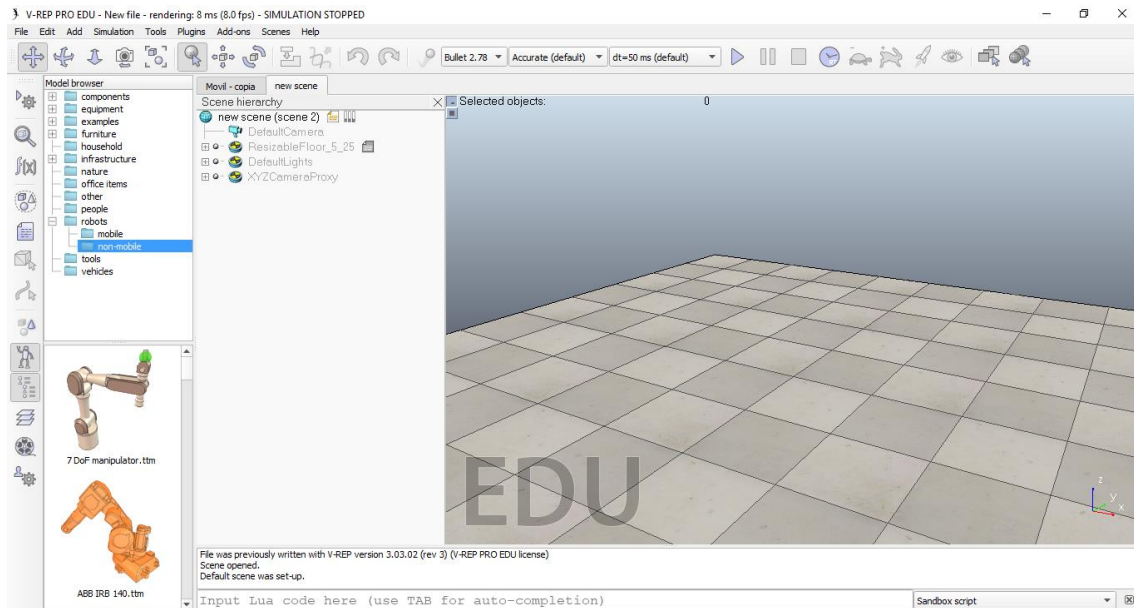


Fig. 6.32: Configuración de la escena inicial para la primera aplicación

Elaborado por: **Roberto Segura**

La interacción del robot con un entorno limitado por paredes puede permitir el incremento en la destreza de un operador ubicado en un sitio remoto. De esta manera, se plantea el uso de propiedades de V-REP para la creación de una habitación, en la cual se podrán ubicar más elementos que actúen como obstáculos los cuales deberán ser evadidos por el usuario. El desarrollo de la habitación se presenta en las siguientes figuras: Fig. 6.33, Fig. 6.34 y Fig. 6.35.

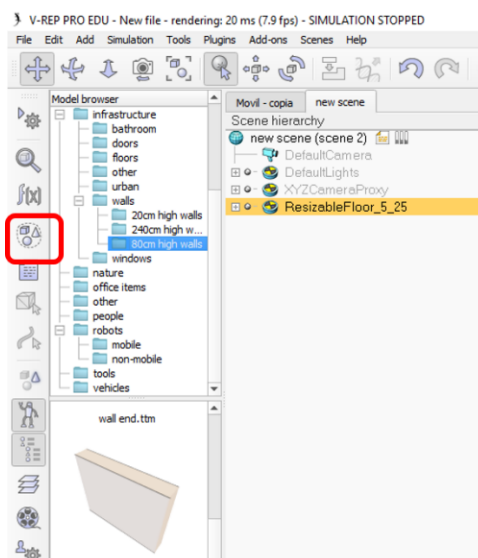


Fig. 6.33: Menú de objetos de V-REP

Elaborado por: **Roberto Segura**

Se escoge el tipo de paredes para recrear el entorno de la habitación como se muestra a continuación, Fig. 6.34.

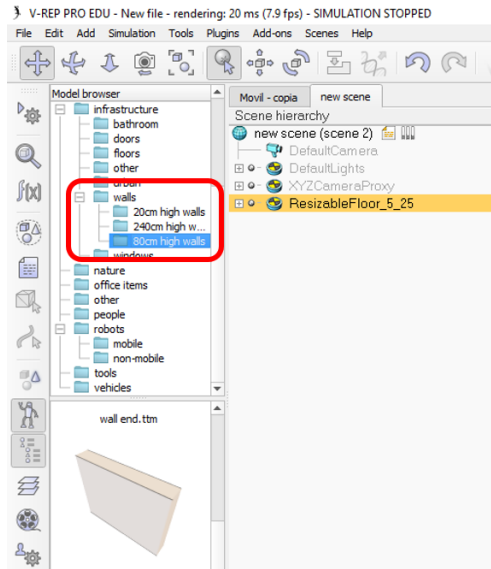


Fig. 6.34: Selección de paredes para la creación de la habitación.

Elaborado por: **Roberto Segura**

En Fig. 6.35 se muestra como se ve de la parte exterior la habitación recreada con los objetos que cuenta el software V-Rep.

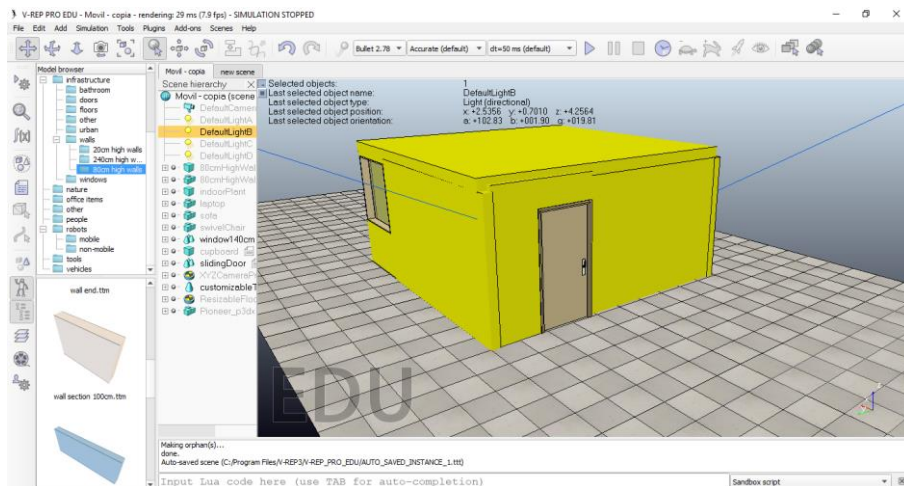


Fig. 6.35: Creación de la habitación simulada.

Elaborado por: **Roberto Segura**

El entorno consta de múltiples objetos ubicados en diversas locaciones dentro de la habitación, donde sillas, mesas, columnas y diferentes muebles que pueden interferir en el camino del robot pueden ser añadidos, los cuales deben ser detectados por un sensor laser integrado en el robot simulado (Fig. 6.36).

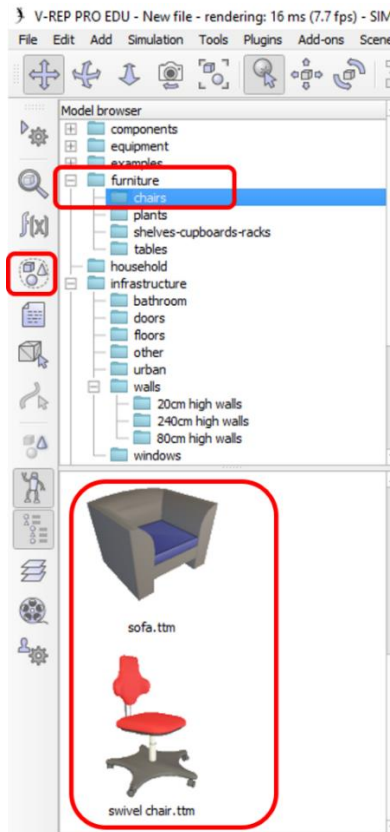


Fig. 6.36: Selección de objetos para el entorno.

Elaborado por: **Roberto Segura**

Dentro de los diferentes objetos que se tiene para seleccionar en V-Rep, se escoge los más apropiados que se asemejen al entorno que vamos a desarrollar Fig. 6.37.

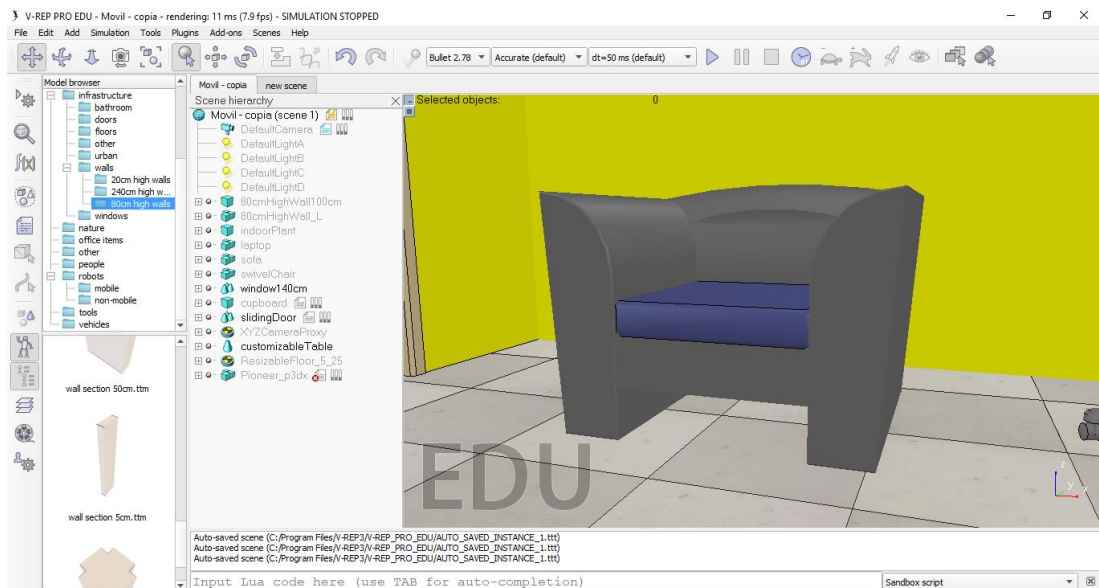


Fig. 6.37: Creación del entorno con obstáculos y objetos para la simulación

Elaborado por: **Roberto Segura**

Los objetos que estarán en el entorno se agregan uno por uno en el sitio que le parezca al programador, logrando obtener un espacio de trabajo completamente no estructurado, como se muestra en la Fig. 6.38.

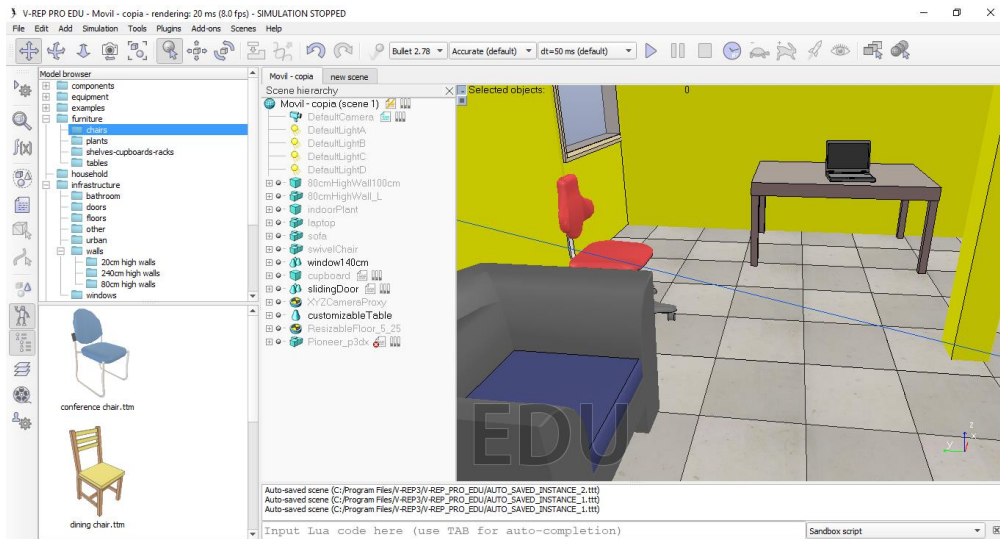


Fig. 6.38: Entorno de simulación para la navegación del robot diferencial

Elaborado por: **Roberto Segura**

Creado el entorno de simulación, se puede colocar un robot diferencial (el cual será comandado mediante el dispositivo háptico), el cual trabajará de manera horizontal sobre el plano $X - Y$. Se considera un robot tipo unicycle el cual posee una cinemática diferencial para desplazarse. En la Fig. 6.39 se indica la configuración de un robot tipo unicycle ubicado sobre el espacio de trabajo, denotando las velocidades que pueden ser inyectadas para su control.

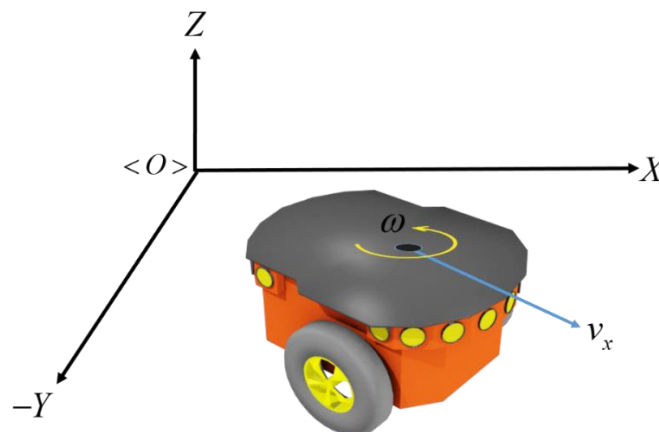


Fig. 6.39: Robot diferencial ubicado sobre el espacio de trabajo

Elaborado por: **Roberto Segura**

Las velocidades que permiten el desplazamiento del robot diferencial son dos: una velocidad lineal v que permite al robot desplazarse frontalmente y una velocidad angular ω , con la cual el robot puede girar. El control de este tipo de robot puede llevarse a cabo a través de la modificación de estas dos velocidades, las cuales permiten comandar al robot de una manera más intuitiva.

El siguiente paso es incorporar el robot al entorno de simulación, tomando en cuenta que este va a ser el que se va a controlar por medio del dispositivo háptico inyectando las velocidades correspondientes. V-REP tiene incorporado en su base de datos un prototipo de robot con las características expuestas anteriormente (Pioneer 3-DX), el cual viene incorporado con sensores que determinan obstáculos presentes en el frente del robot. La Fig. 6.40 muestra el robot integrado en la librería de V-REP y todo lo que se incluye en este prototipo para desarrollar pruebas experimentales.

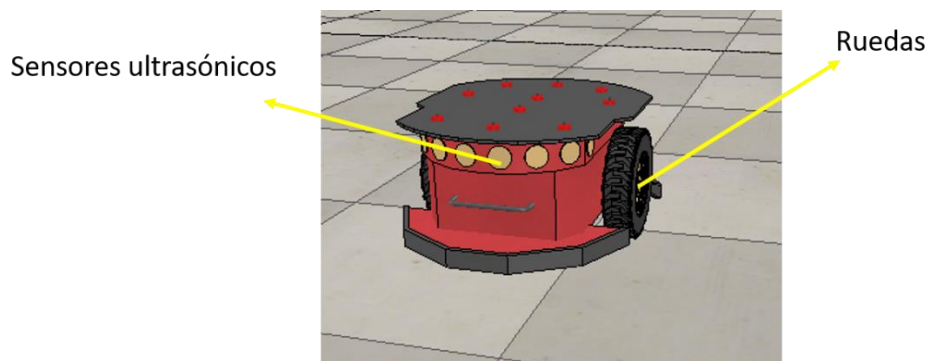


Fig. 6.40: Robot diferencia Pioneer 3-DX integrado en la librería de V-REP

Elaborado por: **Roberto Segura**

Dentro del software de simulación, el robot Pioneer 3-DX está ensamblado de una forma muy similar al real, contando con dos motores (cada motor está asociado a una rueda que le permita desplazarse de acuerdo a su dinámica) y una variedad de sensores ultrasonidos ya establecidos previamente para cualquier uso que el usuario requiera. Para incorporar más dispositivos virtuales al robot como: cámaras, sensores laser o brazos robóticos, se debe seleccionar de entre una lista de librerías desarrolladas de fábrica y colocar los equipos adicionales sobre el robot. Para agregar el robot al entorno creado del cuarto, se lo realiza desde la barra de objetos que posee el simulador, como se muestra en la Fig. 6.41 y Fig. 6.42.

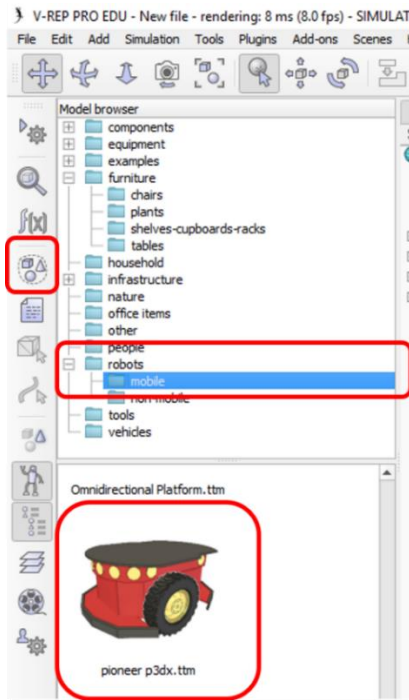


Fig. 6.41: Objetos robot para la simulación
Elaborado por: **Roberto Segura**

Una vez ubicado el robot Pionner en V-Rep se procede a ubicarlo dentro del entorno que se desarrolla Fig 6.42.

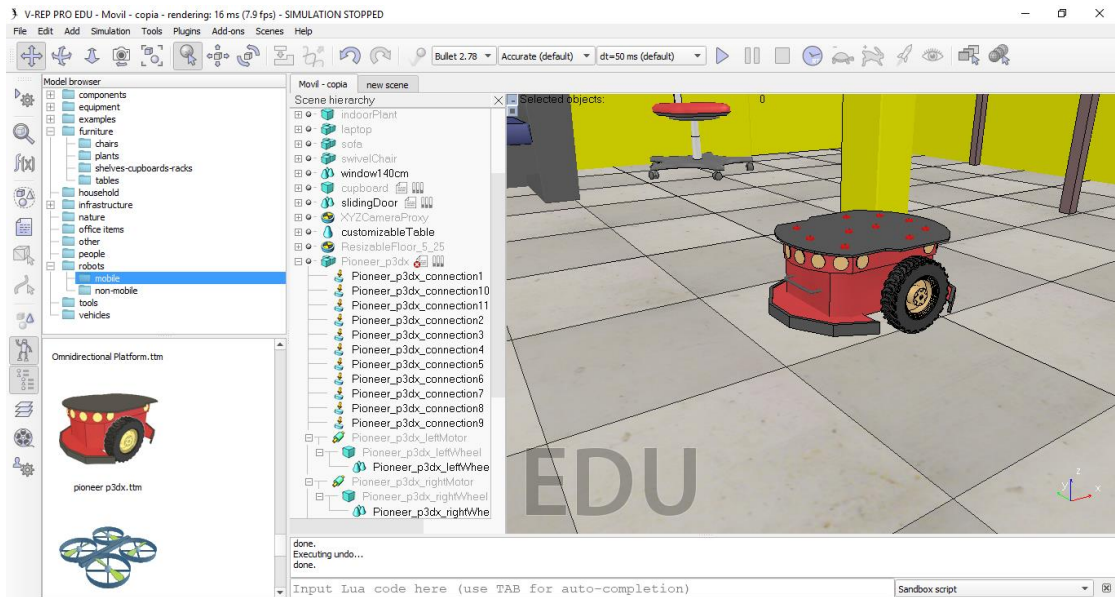


Fig. 6.42: Incorporación del robot Pionner 3-DX en la simulación
Elaborado por: **Roberto Segura**

Para el proceso de experimentación, se agrega un sensor de visión sobre el robot, una cámara en la cual se pueda observar desde la parte frontal cómo el robot está siendo guiando para mostrar una perspectiva in-situ al operario. La Fig. 6.43 muestra la inclusión

de una cámara en la parte delantera del robot, esta cámara permite retroalimentar de manera visual al operador.

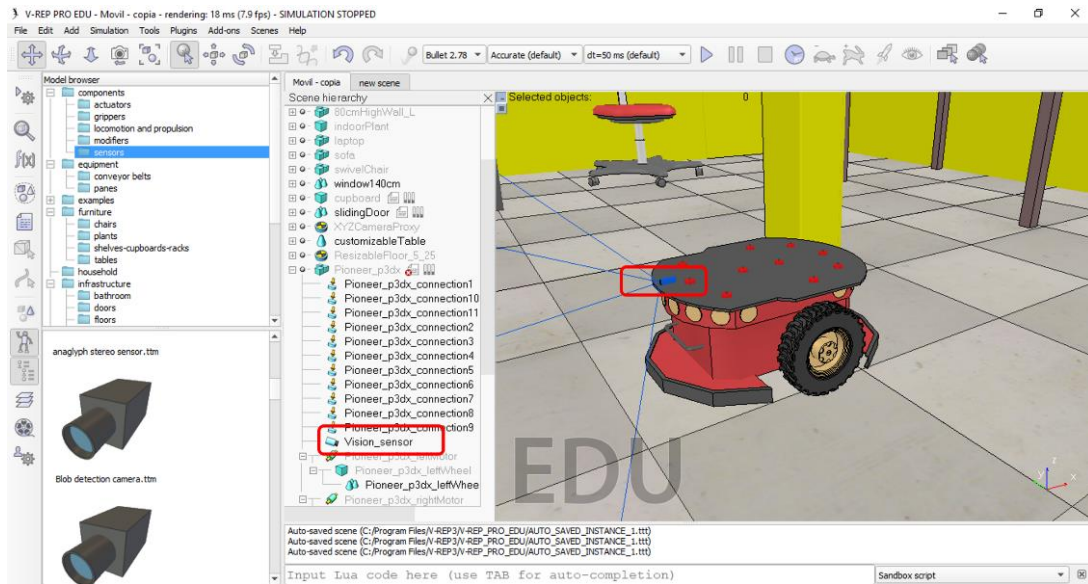


Fig. 6.43: Cámara para retroalimentación visual.

Elaborado por: **Roberto Segura**

B. Caso 2: Manipulación Tele-Operada

De manera similar al desarrollo de la primera escena de simulación, la segunda aplicación requiere de los pasos de configuración para la conexión con Matlab. La segunda aplicación considera un brazo robótico y cajas de distintos pesos con el propósito de colocarlas sobre una banda transportadora. El proceso que se debe ejecutar es mediante el dispositivo háptico, con el cual se puede operar el extremo operativo del manipulador, coordinando de una manera adecuada para atrapar las cajas y ubicarlas sobre la cinta transportadora. Este proceso ayuda a percibir la retroalimentación de fuerza que se genera en el extremo operativo al momento de levantar las diferentes cajas hacia la cinta. De forma parecida al entorno de la habitación construida en el primer entorno, este escenario está conformado de la misma estructura de una habitación, tal y como se muestra en la Fig. 6.44.

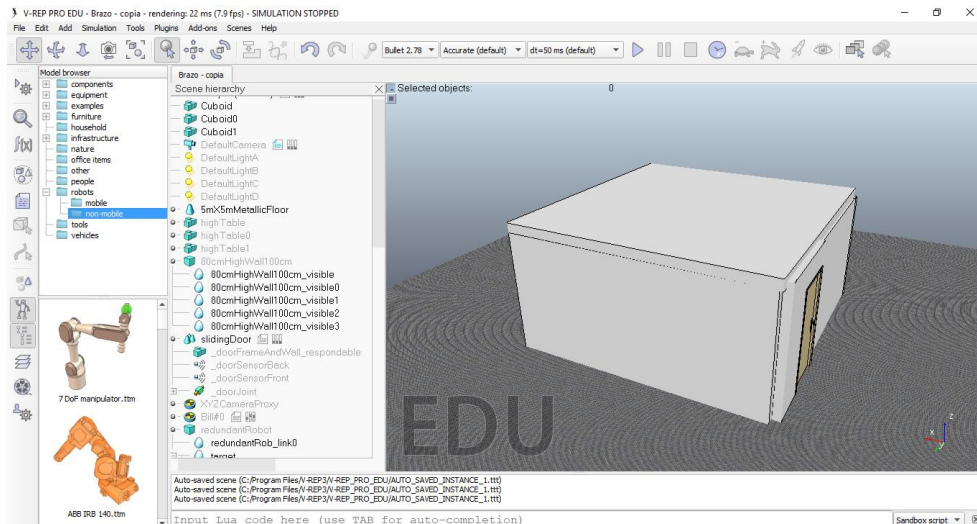


Fig. 6.44: Habitación diseñada para el segundo escenario

Elaborado por: **Roberto Segura**

Posteriormente, se crea el entorno interior de la habitación en la cual se realizará el proceso de manipulación, colocando la cinta transportadora y las cajas con diferente peso

Fig. 6.45

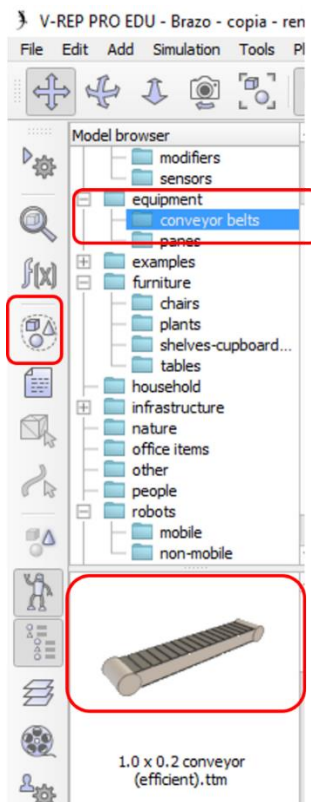


Fig. 6.45: Selección de banda transportadora.

Elaborado por: **Roberto Segura**

De la misma manera se agrega la banda transportadora que desplazará las cajas que el manipulador coloque Fig. 6.46.

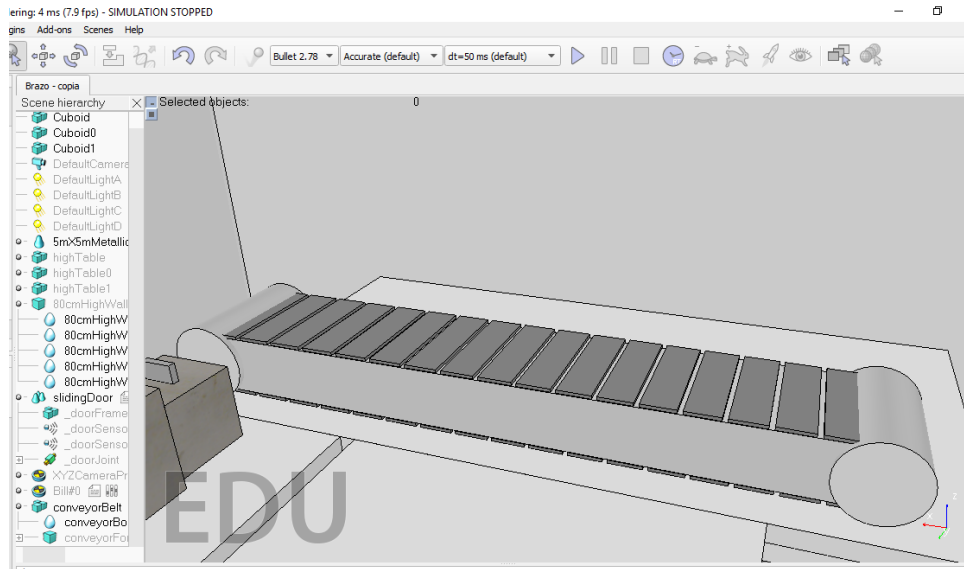


Fig. 6.46: Creación de la banda transportadora para las cajas.

Elaborado por: **Roberto Segura**

Para continuar con la creación del proceso, se empieza a colocar las cajas de diferentes tamaños para que el manipulador al momento de levantar sienta el peso de cada una de ellas y el usuario se retroalimente de esta fuerza que se genera, Fig. 6.47.

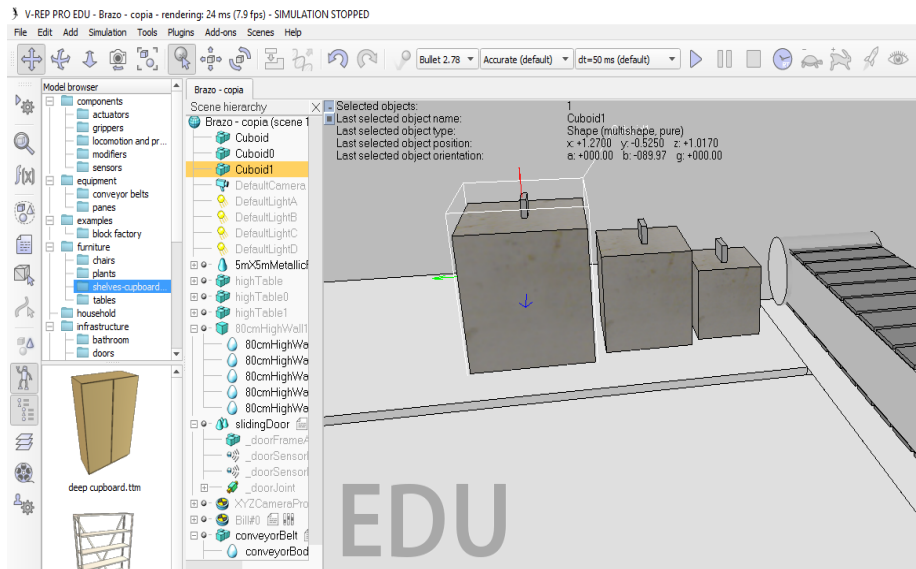


Fig. 6.47: Colocación de las cajas a manipular con el brazo robótico.

Elaborado por: **Roberto Segura**

Con el proceso ya ensamblado, finalmente se escoge el manipulador (brazo robótico) en V-Rep Fig. 6.48.

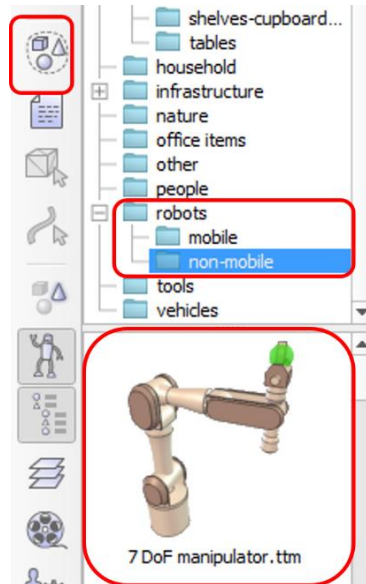


Fig. 6.48: Manipulador en V-REP

Elaborado por: **Roberto Segura**

En la Fig. 6.49 se observa el proceso ya ensamblado por completo, configurado de acuerdo a lo que se empleará para realizar el control tele-operado para levantar las cajas y colocar en la banda transportadora.

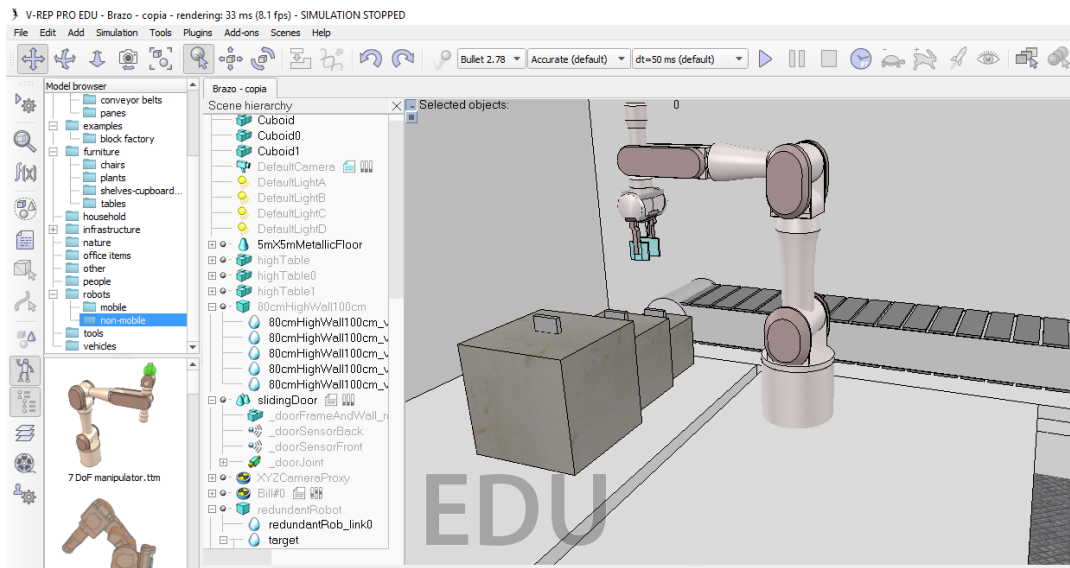


Fig. 6.49: Ensamble de la escena con el manipulador robótico en V-REP

Elaborado por: **Roberto Segura**

La ubicación de la banda transportadora en la mesa a una altura adecuada permite calcular la posición adecuada de las cajas que el manipulador tendrá que levantar. La Fig. 6.46 indica la configuración de las cajas con distinto peso, ubicaciones que muestran una perspectiva de donde debe ser ubicado el brazo robótico.

Culminado de ensamblar las partes principales para transportar las cajas con la banda transportadora, se procede a insertar el brazo robótico tal y como se muestra en la Fig. 6.49.

Los pasos detallados anteriormente permiten la creación de los entornos de simulación en los cuales se va a realizar los experimentos planteados con el dispositivo háptico. Mediante MATLAB ahora se desarrolla la programación adecuada para comunicarse con cada entorno de simulación e interactuar con el dispositivo externo, donde se consideran los algoritmos de control, conexión con el dispositivo háptico y conexión con el software de simulación.

Una vez finalizada la creación del entorno de simulación 3D en V-REP, es requerido desarrollar el control desde el software de MATLAB. Para detallar como está estructurado el control se analiza cada uno de los componentes que se deben tomar en cuenta al momento de interactuar con el simulador. Detalles como el tipo de comandos para enviar y recibir datos, configuración del robot móvil y del manipulador, características de cada robot, etc., son necesarios para plantar una metodología de control.

6.9.5. Metodología de control

Para la ejecución de las dos tareas programadas como se detalló en la sección anterior, se opera a cualquiera de los robots mediante el Falcon Novint enviando y recibiendo datos de retroalimentación para el control de acuerdo a los comandos y estructura detallada en Fig. 6.50.

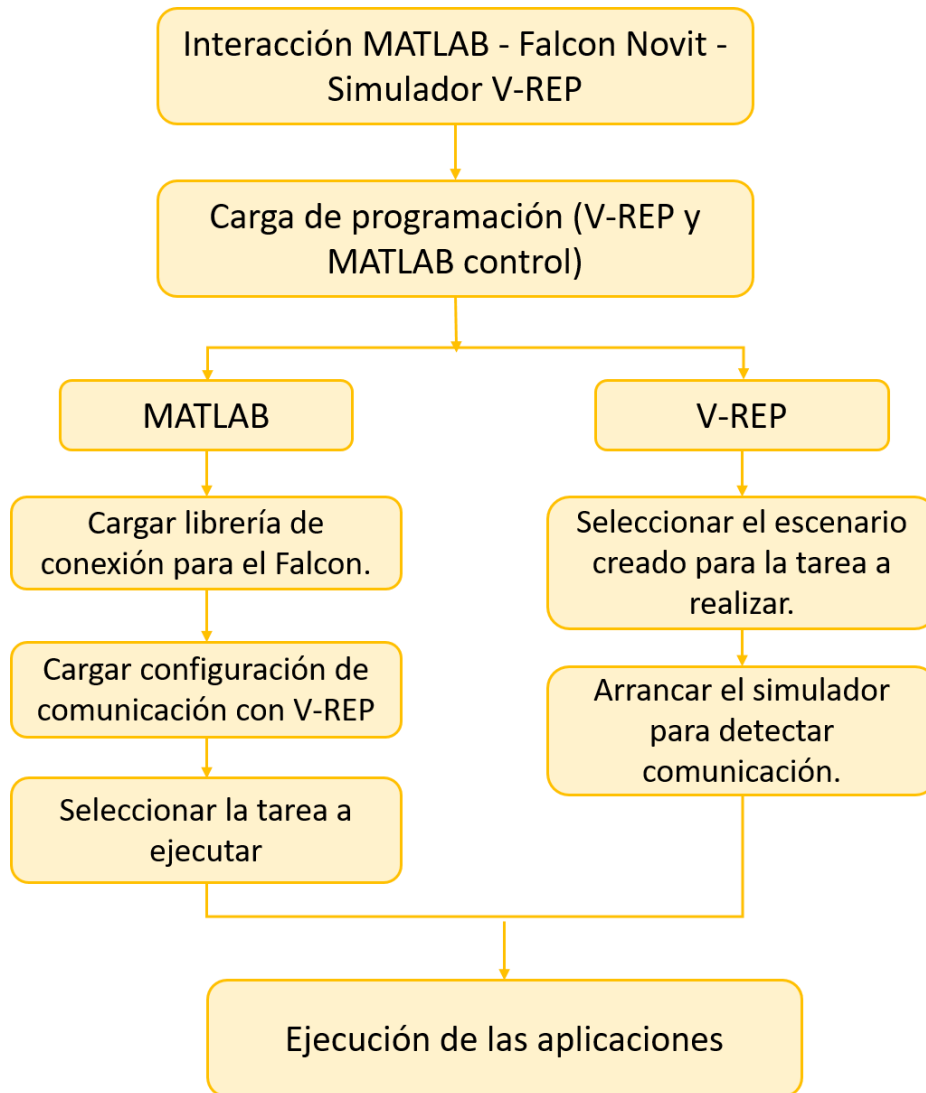


Fig. 6.50: Flujo de trabajo entre Matlab, V-REP y Novint Falcon

Elaborado por: **Roberto Segura**

A. Esquema de control para robot móvil

Con el objetivo de evitar colisiones entre el robot y los elementos del entorno, el primer esquema de control retroalimenta fuerzas en el extremo operativo del robot maestro cuando la proximidad a algún obstáculo supera un umbral. El esquema presentado en la Fig. 6.51 muestra el modelo de operación para manejar una plataforma móvil mediante un dispositivo Háptico, donde se visualiza todos los pasos desarrollados para poder mantener al robot alejado de objetos dispersos en el entorno semiestructurado.

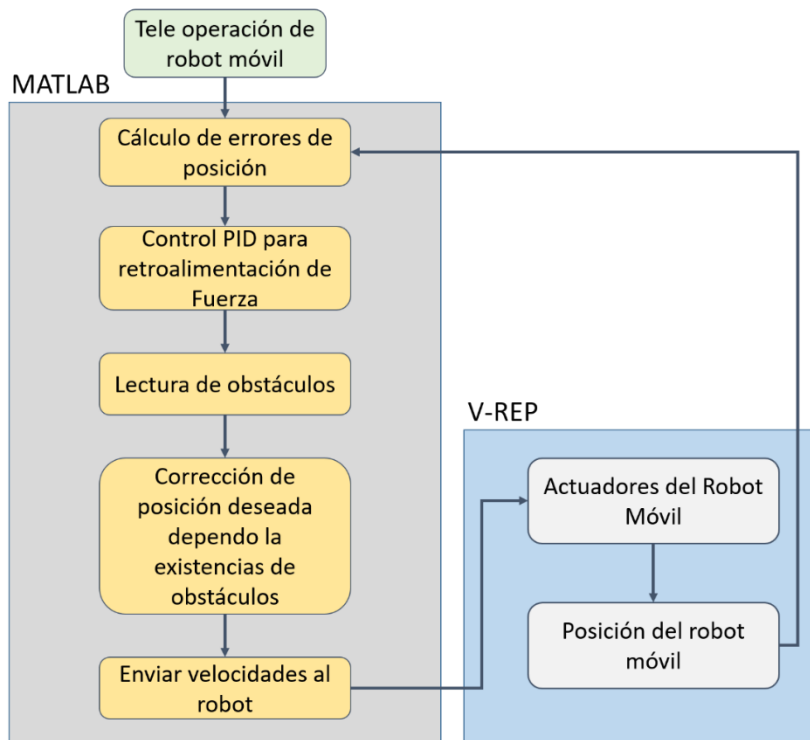


Fig. 6.51: Modelo de operación para manejar la plataforma móvil en un entorno semiestructurado

Elaborado por: **Roberto Segura**

Para poder llevar a cabo la tarea planteada, se calculan las diferencias de posición entre el robot y los elementos vecinos a través de la incorporación de una laser virtual, información que permite tomar acciones para la evasión. El uso de un control PID permite determinar la fuerza de retroalimentación al dispositivo háptico para que el operario se ayude de esta y pueda evitar colisiones que afectarían la integridad del mecanismo robótico, inyectando velocidades proporcionales a cada rueda del robot móvil (dependiendo de la ubicación del obstáculo en el plano). El control del robot móvil es supervisado adicionalmente mediante retroalimentación visual, representado por una cámara. La cámara es ubicada en la parte frontal del robot para visualizar el espacio donde el robot se desplaza, permitiendo validar la retroalimentación obtenida por el dispositivo háptico.

B. Esquema de control para el manipulador

Otra aplicación usual que requiere de retroalimentación por fuerzas es el apilamiento de objetos. De este modo, el segundo esquema de control se centra en retornar fuerzas para que el usuario pueda percibir una sensación de peso que el objeto tenga. Para esto, es

requerido conocer la posición del extremo operativo del dispositivo háptico, posición que sirve para controlar la ubicación del punto de interés del robot esclavo (manipulador robótico) el cual obedece a la cinemática inversa. A pesar que se estudia la cinemática inversa del manipulador en una de las secciones de este trabajo, la cinemática viene incluida por defecto en cada uno de los robots simulados de V-REP. La Fig. 6.52 muestra el modelo de operación del segundo escenario, considerando que el manipulador robótico trabaja sobre un escenario semiestructurado.

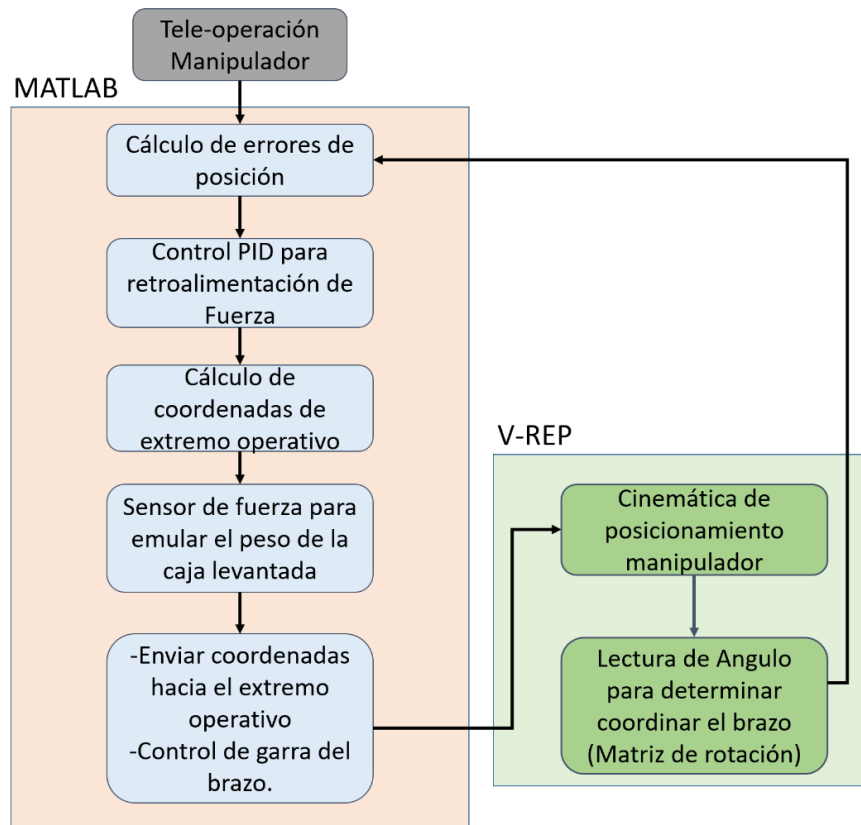


Fig. 6.52: Modelo de operación para manejar el manipulador móvil sobre un espacio estructurado

Elaborado por: **Roberto Segura**

El esquema propuesto para esta tarea de tele-operación y manipulación de objetos es similar la estructura de control del robot móvil. De esta manera, se obtiene una lectura del extremo operativo del robot localizado en el gripper (pinza de sujeción) del manipulador, posición que permite calcular las fuerzas de retorno hacia el dispositivo háptico, donde mientras más cercano se encuentre el gripper al punto de agarre del objeto, más fuerza puede experimentarse en el Falcon Novint. Una vez inyectadas las fuerzas en el dispositivo háptico, se calcula la nueva posición $[hx \ hy \ hz]$ para el extremo operativo

del brazo robótico. Dentro del simulador se cuenta con código pre establecido, el cual contiene la cinemática del manipulador, permitiendo únicamente enviar la posición a la que el punto de interés del manipulador debe ubicarse desde el software matemático. Para determinar el peso del objeto a apilar, sensores de fuerza localizados en la pinza de sujeción del robot se incluyen para crear una fuerza ficticia acorde al peso que este levantando el robot.

6.9.6. Pruebas experimentales

Para realizar la experimentación de tele operación mediante el uso del Falcon se realizó dos tareas propuestas, cada una de ellas con robots diferentes. El ambiente en el cual se va a realizar la tele-operación, es en el software V-REP creando un escenario diferente detallado anteriormente.

6.9.7. Experimento 1: Tele-Operación de un robot móvil.

La primera tarea tiene como objetivo realizar la conducción de un robot móvil tipo unicycle dentro de un ambiente parcialmente estructurado. Considerando que el robot se encuentra en una posición inicial, la tarea principal es dirigirlo hacia un punto deseado, donde el operador puede aplicar velocidades lineales y angulares a través del dispositivo háptico mientras experimenta retroalimentación de fuerzas. De esta manera, si el robot se encuentra cerca de colisionar, una fuerza es generada por el dispositivo Háptico, informando al operador la inminente colisión de una manera directa, para que realice las correcciones requeridas y siga acercándose al punto deseado. La Fig. 6.53 muestra al robot móvil en el punto de inicio del entorno, donde las acciones de control son inicialmente nulas.

Para realizar la simulación y proceso experimental del control, para tele-operar el robot móvil con el Falcon se determina que las constantes del controlador son $\mathbf{K}_p = \mathbf{0.1}$, $\mathbf{K}_i = \mathbf{0.3}$ y $\mathbf{K}_d = \mathbf{0.01}$, con estas constantes se realiza las pruebas para la Tele-Operación, el **Anexo 4** presenta el código del controlador y de toda la aplicación para comunicarse con V-REP y el dispositivo Falcon.

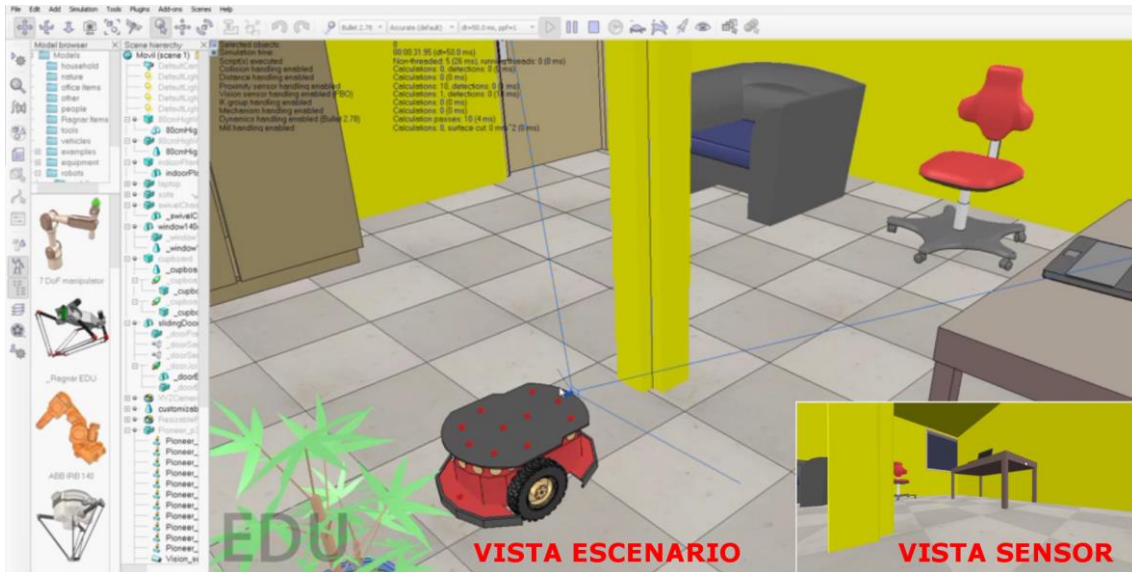


Fig. 6.53: Robot móvil ubicado en el punto de partida del entorno simulado

Elaborado por: **Roberto Segura**

Dependiendo de la distancia entre el robot y los obstáculos cercanos, diversos errores en cada uno de los ejes son calculados (Fig. 6.54), donde las fuerzas de retorno se modificarán para evitar la colisión y se basan directamente en estas diferencias.

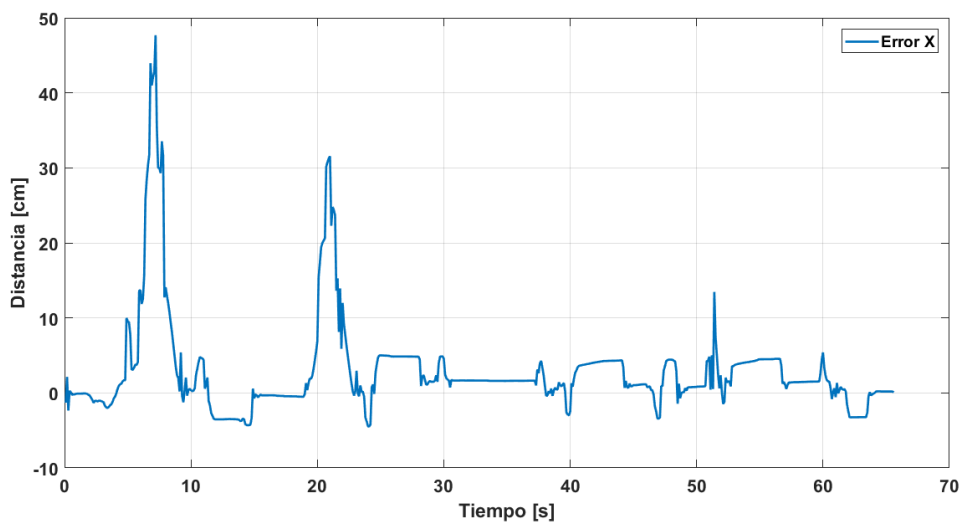


Fig. 6.54: Error de control en el eje X dela tarea realizada.

Elaborado por: **Roberto Segura**

De igual manera en la Fig. 6.55, se indica el error de control en el eje y observando como este de igual manera oscila alrededor de cero.

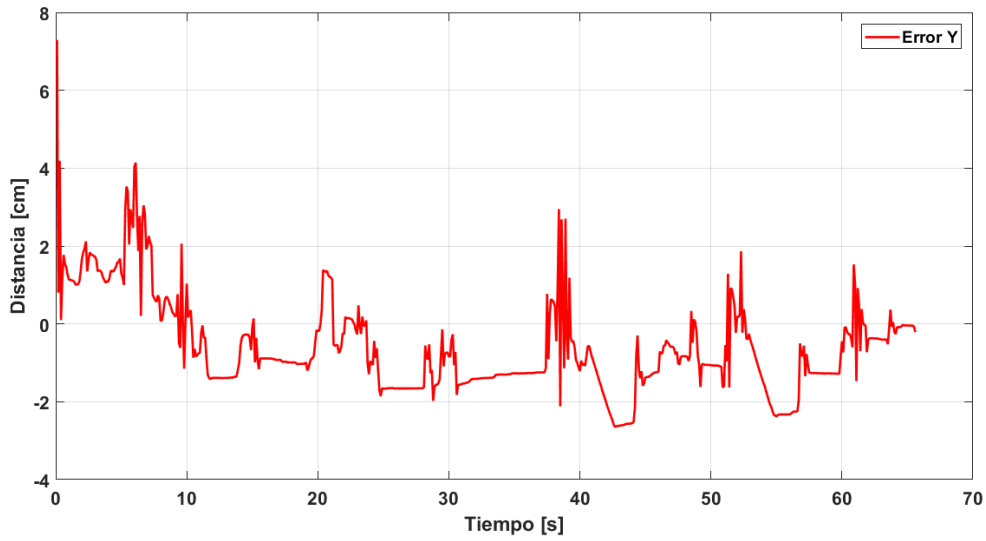


Fig. 6.55: Error de control en el eje Y de la tarea realizada.

Elaborado por: **Roberto Segura**

Y por último en la Fig.6.56, se tiene también el error en el eje z, al igual que los anteriores errores tiene a ser cero u oscila alrededor de cero dependiendo de la tarea realizada.

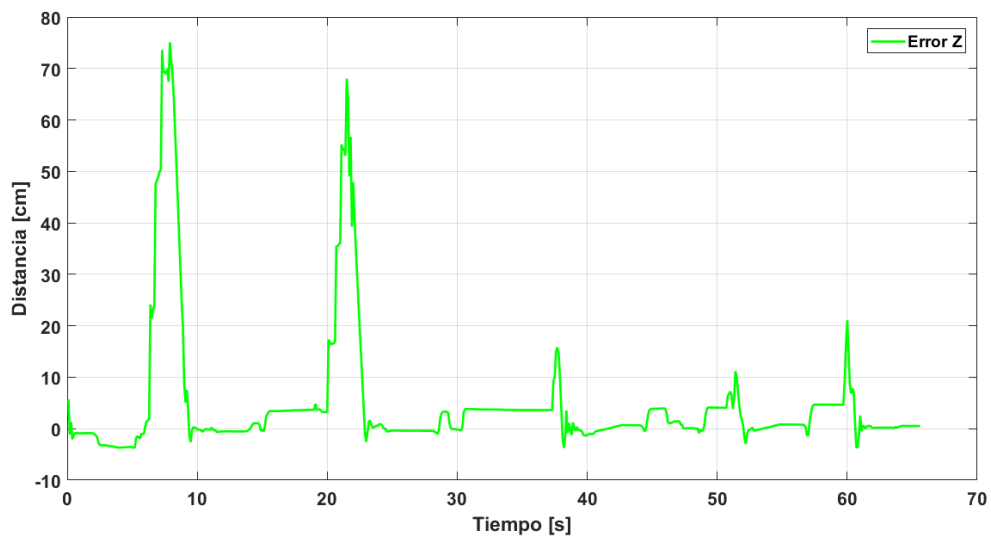


Fig. 6.56: Error de control en el eje Z de la tarea realizada.

Elaborado por: **Roberto Segura**

En la Fig. 6.57 presenta la fuerza en X de corrección inyectadas al dispositivo háptico como respuesta de la interacción entre el robot y el entorno.

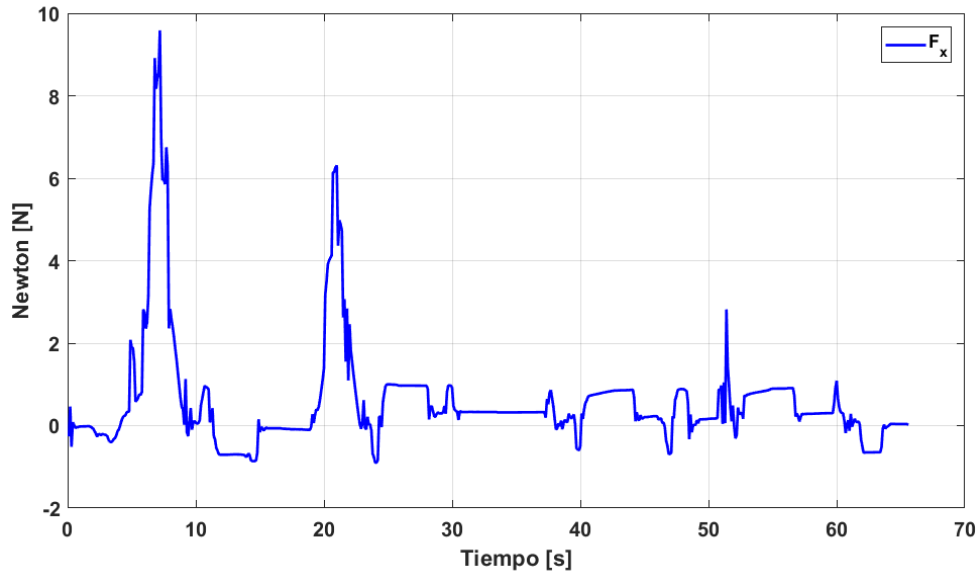


Fig. 6.57: Fuerzas de retroalimentación en X como respuesta entre la interacción robot-escenario

Elaborado por: **Roberto Segura**

En la Fig. 6.58 indica la fuerza en Y inyectada al robot háptico con el propósito de guiar al operador a cumplir la tarea planteada, esta fuerza dependerá del error de control en este mismo eje.

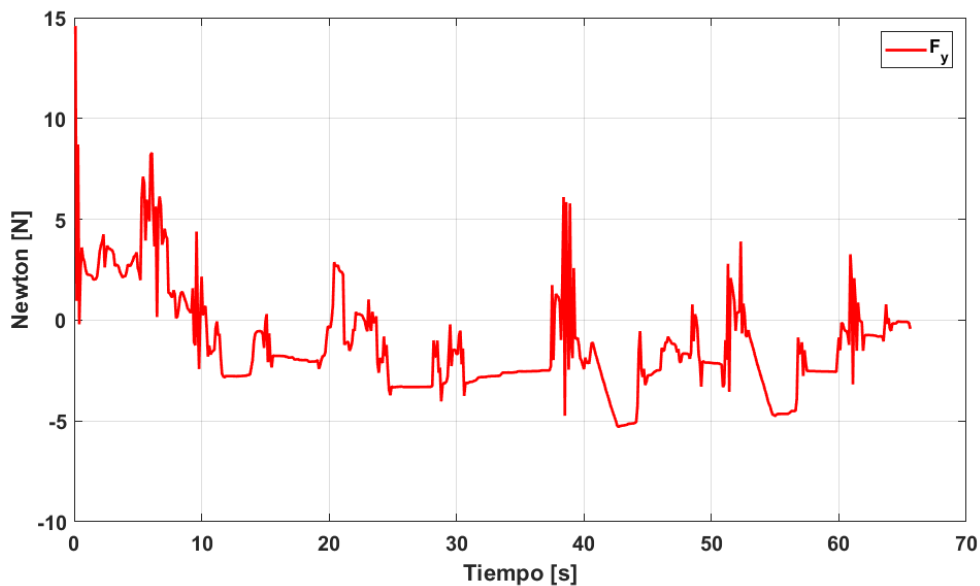


Fig. 6.58: Fuerzas de retroalimentación en Y como respuesta entre la interacción robot-escenario

Elaborado por: **Roberto Segura**

Por último, se indica en la Fig.6.59 la fuerza en Z inyectada al robot háptico producida por la interacción del robot con el escenario.

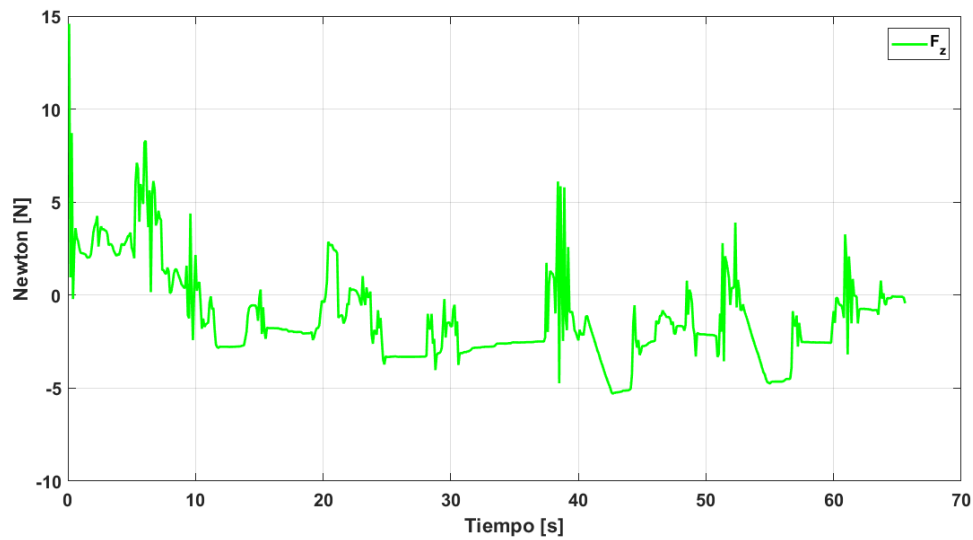


Fig. 6.59: Fuerzas de retroalimentación en Z como respuesta entre la interacción robot-escenario

Elaborado por: **Roberto Segura**

En la siguiente figura (Fig.6.60) se muestra como el operador tele-opera con el dispositivo háptico el robot móvil dentro del entorno desarrollado.

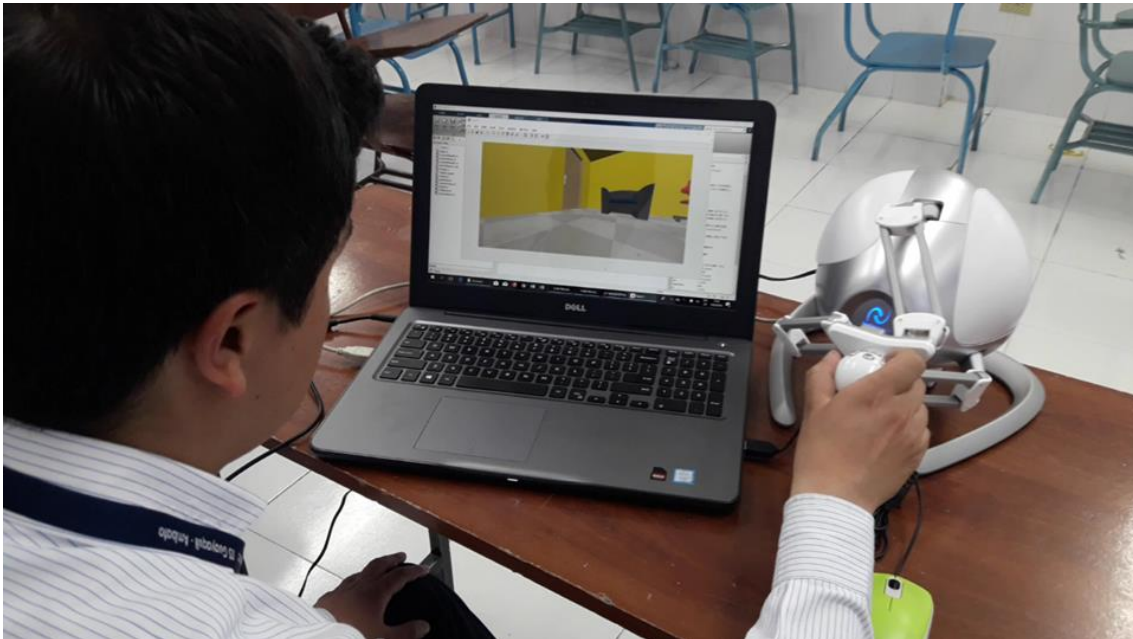


Fig. 6.60: Usuario ejecutando la prueba de experimentación

Elaborado por: **Roberto Segura**

Análisis de la tele-operación del robot móvil.

En el proceso de validar de mejor manera el control implementado se realiza más pruebas experimentales de cada tarea propuesta, para esto se toma en cuenta el error producido cada vez que se ejecuta la tarea, el error calculado para lograr tabular los datos se lo calcula de la forma:

$$\rho = \sqrt{(x_d - x)^2 + (y_d - y)^2 + (z_d - z)^2}$$

La ecuación descrita, representa el error cuadrático medio rho (ρ), representado por la raíz cuadrada de la suma de los cuadrados de los errores en cada eje.

$$\rho = \sqrt{\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2}$$

En la siguiente tabla se representan los errores para cada prueba ejecutada en la tarea de Tele-Operación del robot móvil.

Tabla 6-2: Errores de las pruebas experimentales para ver desempeño del controlador.

Pruebas	Error promedio \tilde{x} [mm]	Error promedio \tilde{y} [mm]	Error promedio \tilde{z} [mm]	$\bar{\rho}$
1	0.10698	0.2394	0.0552	0.2680
2	0.02086	-0.524	0.1592	0.5480
3	-0.03277	-0.9476	-0.0198	0.9484
4	0.00119	0.0888	0.0857	0.1234
5	0.05934	0.7485	-0.3077	0.8115
6	0.01335	0.3024	0.2179	0.3730
7	-0.01330	0.5382	0.5275	0.7537
8	0.02130	1.8339	1.3212	2.2604
9	-0.05169	1.5486	1.7534	2.3399
10	0.01360	-0.1316	0.0009	0.1323
11	-0.09665	1.9576	1.9703	2.7792
12	0.02325	1.737	1.7200	2.4446
13	-0.01061	-0.0675	-0.0662	0.0951
14	0.03229	-0.3814	-0.3786	0.5384
15	0.03842	-1.08	-0.1516	1.0913
16	0.01286	-0.4068	-2.3697	2.4044

17	-0.02129	-0.5773	-0.5837	0.8213
18	-0.01996	0.4362	-0.4085	0.5980
19	-0.01598	0.2078	-0.1121	0.2366
20	0.00796	1.2356	1.2310	1.7442
21	-0.04339	0.9409	1.1528	1.4887
22	-0.01704	-0.2337	-0.5103	0.5615
23	-0.00639	0.048	0.2078	0.2134
24	-0.01216	-1.8427	-0.9641	2.0797
25	0.10698	-1.1604	-1.1032	1.6047
26	0.02086	-1.4473	-0.4528	1.5166
27	-0.03277	-0.7012	-0.2028	0.7307
28	0.00119	-0.0765	0.0242	0.0803
29	0.05934	1.2105	0.6153	1.3592
30	0.01335	0.748	-0.4933	0.8961

Como se puede ver en la tabla anterior los datos experimentales tomados de la tarea de tele operación del vehículo móvil están dentro de un margen de error considerable, en cada prueba se ve que los errores medios tanto para x, y, z oscilan dependiendo del movimiento y posición del móvil.

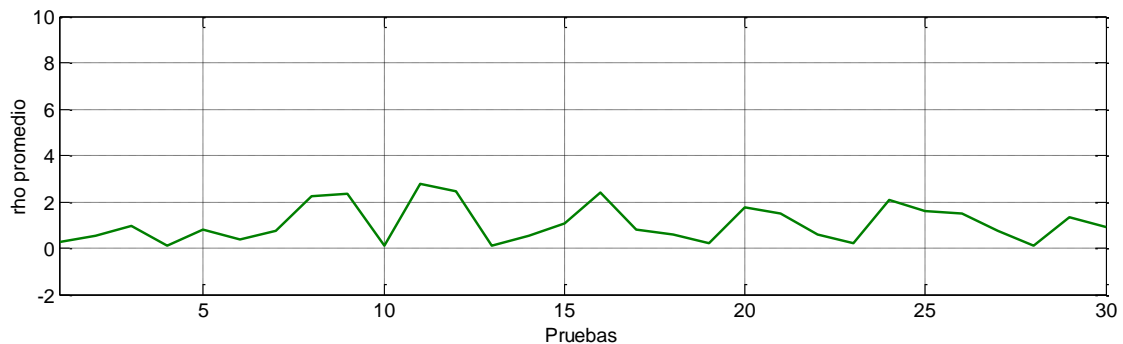


Fig. 6.61: Tendencia del error promedio de cada prueba realizada para la tarea planteada

Elaborado por: **Roberto Segura**

Observado en la tabla los valores promedios están variando moderadamente, concordando casi de igual manera con cada prueba experimental, en la Fig. 6.61 se muestra la tendencia del error promedio de cada prueba ejecutada observando que se mantiene muy estable entre cada una de ellas.

6.9.8. Experimento 2: Tele-Operación de un manipulador

El experimento 2 plantea el control del extremo operativo de un manipulador robótico empleando el dispositivo Háptico para lograr objetivos de apilamiento. De este modo, la tarea del operador es levantar un objeto ubicado dentro de un proceso de transporte por medio de bandas, escenario donde se encuentran cajas de distintos pesos y dimensiones.

La Fig. 6.62 se presenta la ejecución de la tarea por parte de un usuario, logrando cumplir con los objetivos de levantar y colocar las cajas en la banda, de igual manera que la prueba anterior, se determina que las constantes del controlador son $\mathbf{K}_p = \mathbf{0.1}$, $\mathbf{K}_i = \mathbf{0.3}$ y $\mathbf{K}_d = \mathbf{0.01}$, con estas constantes se realiza las pruebas para la Tele-Operación, el Anexo 5 presenta el código del controlador y de toda la aplicación para comunicarse con V-REP y el dispositivo Falcon.

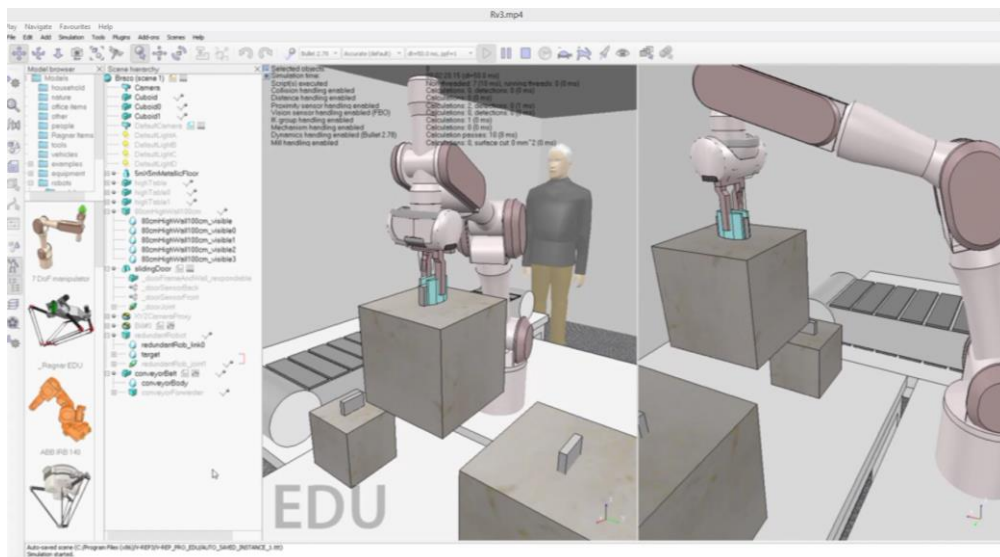


Fig. 6.62: Ejecución de la tarea

Elaborado por: **Roberto Segura**

Los errores para x,y,z en la representación de pesos son presentados a continuación, los cuales por el peso de las cajas tienden a elevarse pero por compensación de control inyectando fuerzas al dispositivo háptico se corrige dichos errores. La Fig. 6.63 proporciona información del error producido en el eje X.

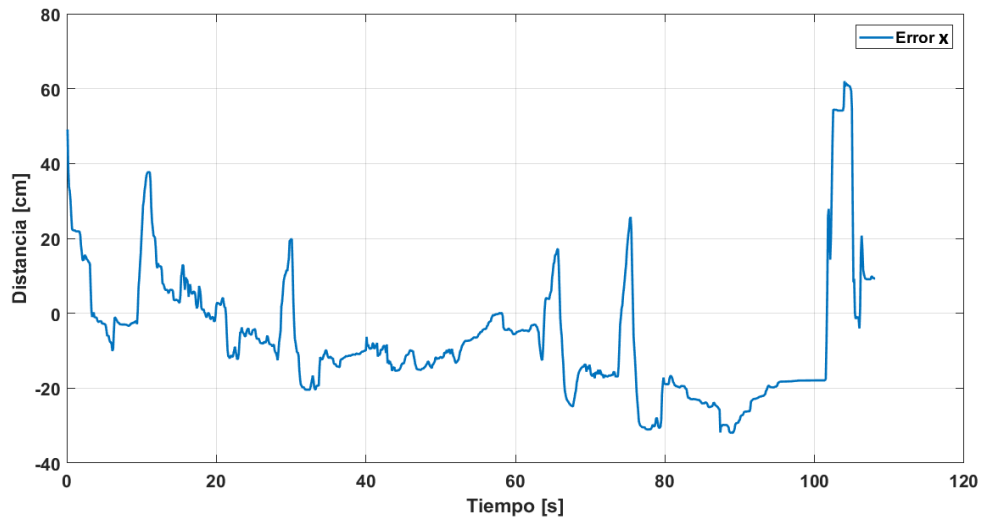


Fig. 6.63: Errores de posición entre el robot y objetos cercanos en el eje X

Elaborado por: **Roberto Segura**

La Fig. 6.64 muestra la evolución del error en Y del control para la tarea planteada, observando como varía dependiendo del peso que se levante de cada caja.

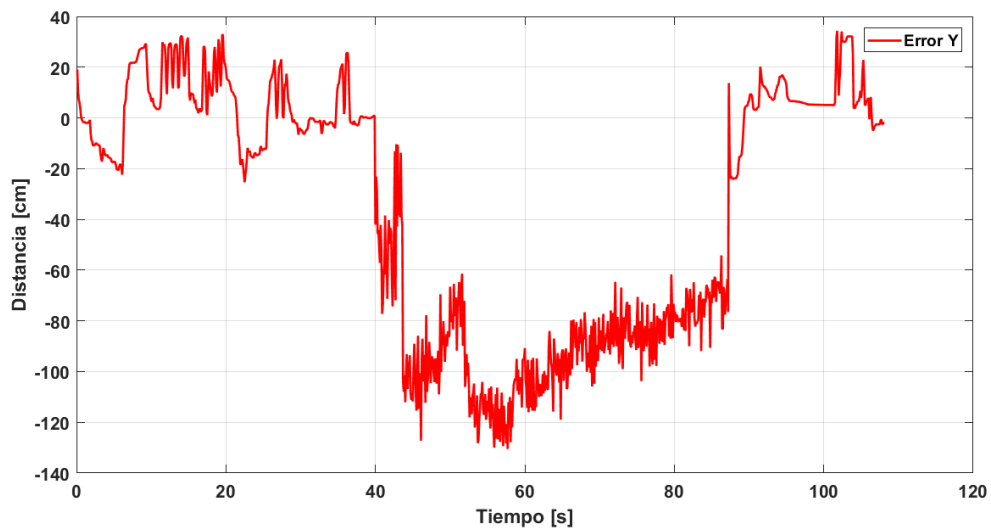


Fig. 6.64: Errores de posición entre el robot y objetos cercanos en el eje Y

Elaborado por: **Roberto Segura**

En la Fig. 6.65 se muestra el error de posición en el eje Z producido entre el objeto y el extremo operativo del robot.

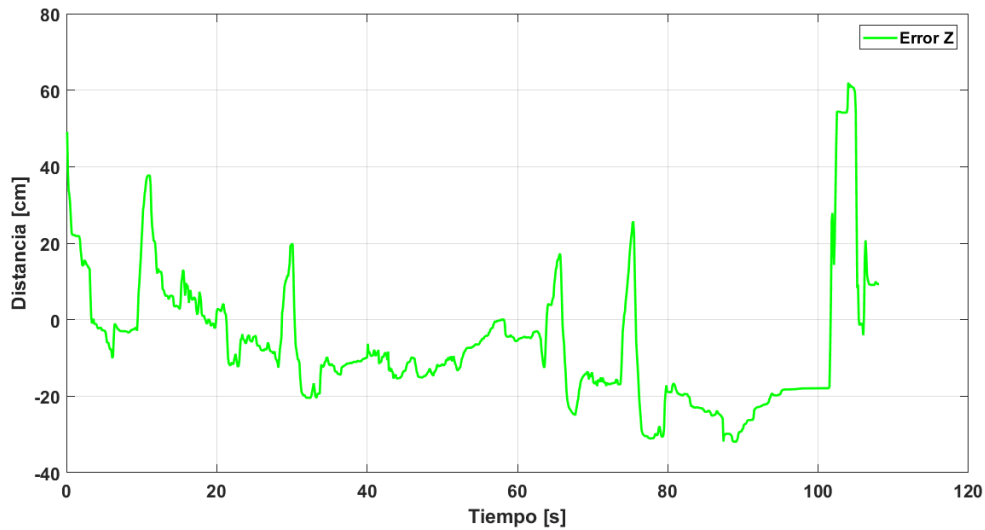


Fig. 6.65: Errores de posición entre el robot y objetos cercanos en el eje Z

Elaborado por: **Roberto Segura**

Las fuerzas que se generan están relacionadas directamente a la respuesta de un sensor de presión ubicado en el gripper del extremo operativo del manipulador móvil. La Fig. 6.66 indica la fuerza producida para el eje X

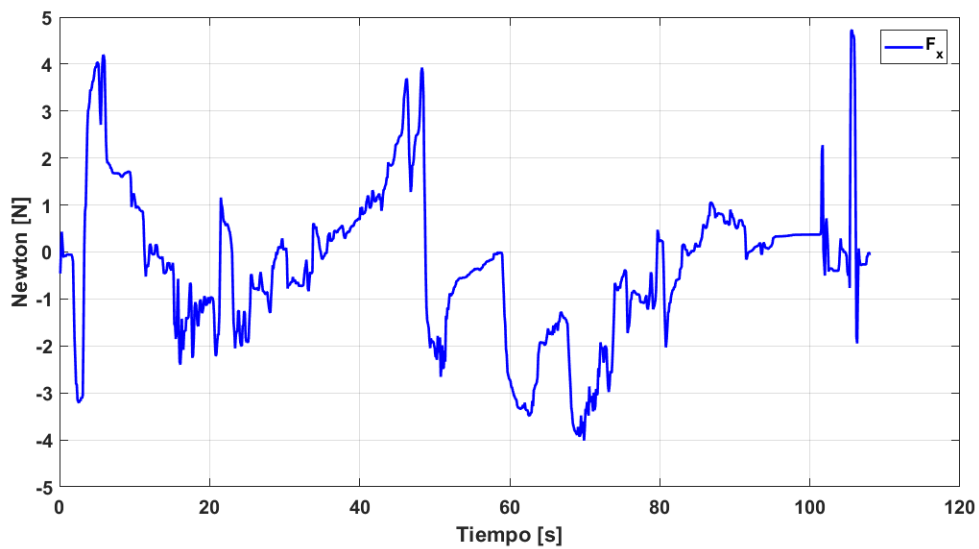


Fig. 6.66: Fuerza de retroalimentación en el eje X.

Elaborado por: **Roberto Segura**

De la misma manera se genera Fuerzas de retroalimentación al operador mediante el dispositivo háptico en el eje Y, Fig. 6.67.

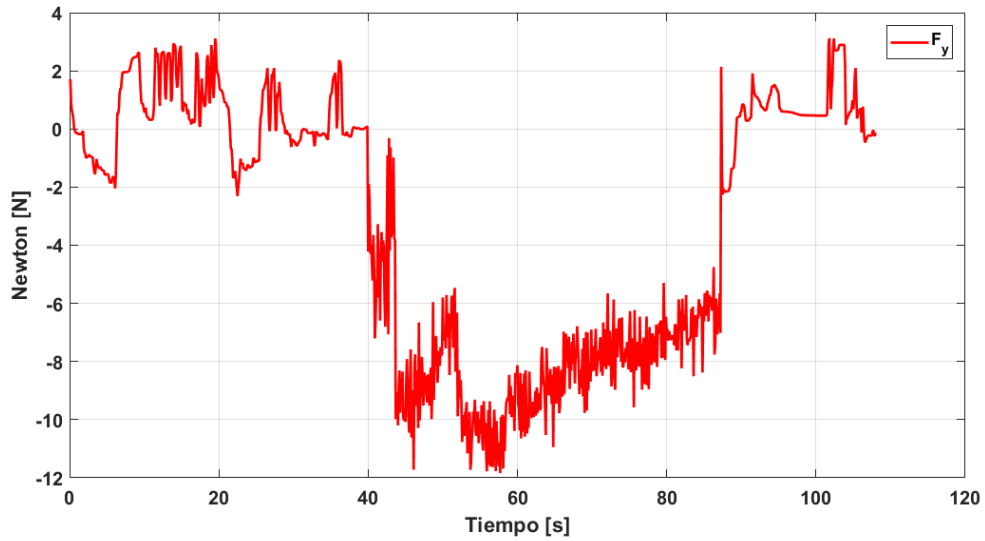


Fig. 6.67: Fuerza de retroalimentación en el eje Y.

Elaborado por: **Roberto Segura**

Y por último en la Fig. 6.68 indica la fuerza producida para el eje Z.

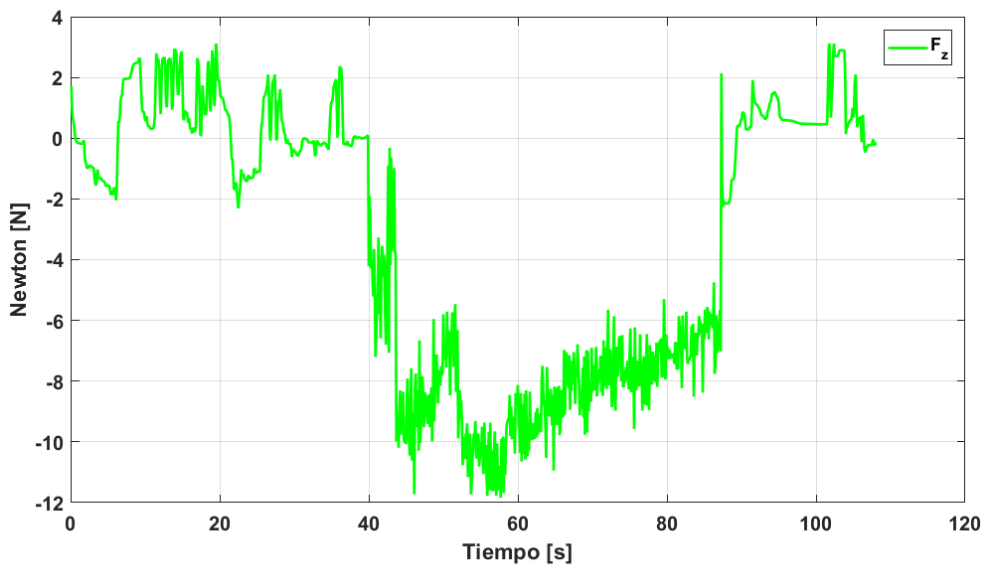


Fig. 6.68: Fuerza de retroalimentación en el eje Z.

Elaborado por: **Roberto Segura**

En la Fig. 6.69 se presenta la ejecución de la tele-operación del manipulador para levantar los pesos y colocarlos en la banda de transporte.

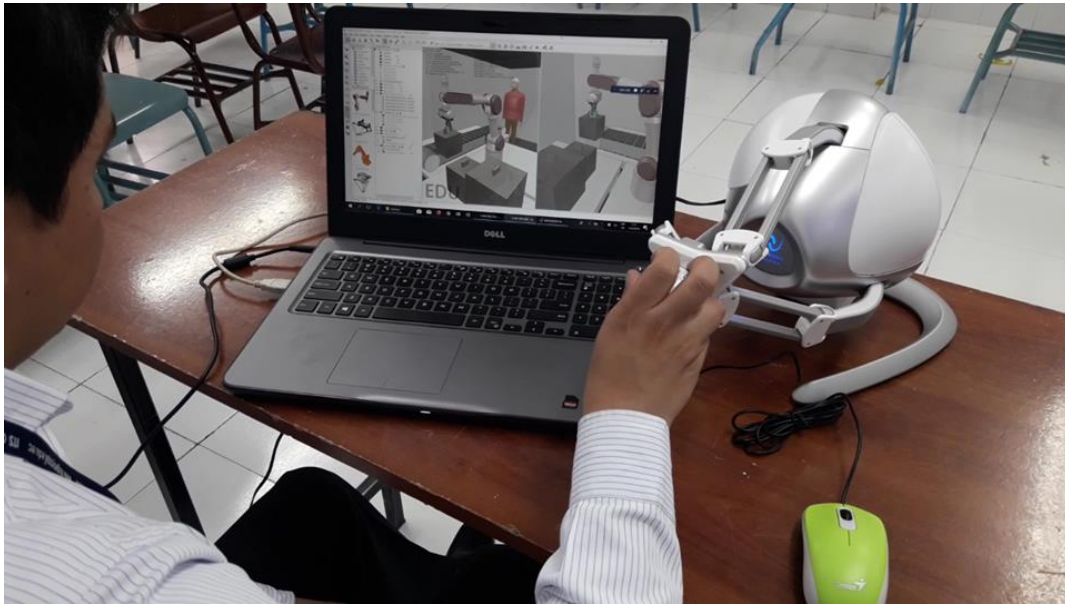


Fig. 6.69: Usuario ejecutando la tarea para el segundo experimento

Elaborado por: **Roberto Segura**

Análisis de la tele-operación del robot manipulador.

De la misma manera que en la tarea uno se propone el cálculo del error de ejecución, en la tabla siguiente se puede ver como evoluciona error promedio para cada prueba ejecutada.

Tabla 6-3: Errores de las pruebas experimentales para ver desempeño del controlador.

Pruebas	Error promedio \bar{x} [mm]	Error promedio \bar{y} [mm]	Error promedio \bar{z} [mm]	$\bar{\rho}$
1	1.3019	-1.6295	0.0458	2.0862
2	-1.0853	-1.5801	-0.4699	1.9737
3	-0.5835	-1.5293	0.2321	1.6532
4	0.0334	-1.4302	-0.1328	1.4367
5	-0.1650	-1.3796	-0.0367	1.3899
6	1.0189	-1.3299	0.0162	1.6754
7	0.0344	-1.2799	0.0198	1.2805
8	-0.2486	-1.2301	-0.0089	1.2550
9	0.0098	-1.1788	-0.0071	1.1789
10	-0.0144	-1.1288	-0.1919	1.1451
11	0.4416	-1.0797	0.0347	1.1670
12	0.0264	-1.03	-0.1904	1.0478

13	-0.2559	-0.9801	-0.0266	1.0133
14	0.5771	-0.9285	0.1117	1.0989
15	-0.5961	-0.8788	-0.3467	1.1171
16	-0.4353	-0.8293	-0.1247	0.9449
17	1.2457	-0.7781	0.1691	1.4785
18	-2.3026	-0.7296	-0.9767	2.6054
19	-1.7910	-0.6794	0.6027	2.0081
20	-0.9610	-0.6303	1.5344	1.9171
21	0.0967	-0.5801	0.6551	0.8803
22	0.5650	-0.5292	-0.2316	0.8080
23	0.1224	-0.4808	0.3347	0.5985
24	-1.0704	-0.4309	-0.8606	1.4395
25	1.3019	-1.6295	0.0458	2.0862
26	-1.0853	-1.5801	-0.4699	1.9737
27	-0.5835	-1.5293	0.2321	1.6532
28	0.0334	-1.4302	-0.1328	1.4367
29	-0.1650	-1.3796	-0.0367	1.3899
30	1.0189	-1.3299	0.0162	1.6754

En la tabla indicada anteriormente se puede ver de igual manera que la tarea uno que los errores estas dentro del margen, sin dispararse de manera exagerada, es decir el control está bien establecido y logra que el operador realice la tarea de tele-operación con el manipulador, si no fuera el caso existiera cada vez errores de más alto valor, mediante esto el operador ya se va familiarizando con la tarea, la Fig. 6.70 muestra el error promedio de todas las pruebas realizadas observando un latencia similar en cada sin tener cambios muy exagerados.

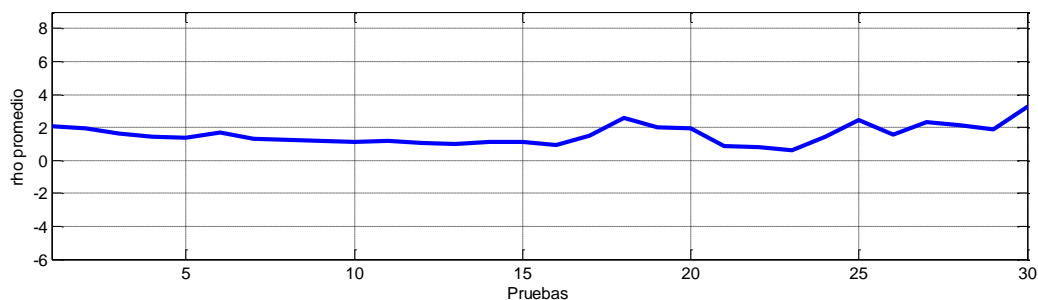


Fig. 6.70: Tendencia del error promedio de cada prueba realizada.

Elaborado por: **Roberto Segura**

6.9.9. Conclusiones y recomendaciones

Conclusiones

- La base bibliográfica sirvió como punto de partida tanto para la selección del dispositivo háptico a utilizar como para la elección del software de simulación. De dicha base, se contrastan las virtudes y desventajas de cada uno de los elementos que podían ser usados para el desarrollo de la propuesta, permitiendo encontrar los que mejor se apegaban a las necesidades del proyecto.
- Con el soporte de las hojas de especificaciones del Novint Falcon, un modelo matemático equivalente es conseguido, el cual representa el comportamiento cinemático del punto de interés del robot maestro. El modelo matemático es usado tanto para la simulación del robot como para la validación de controladores en un software matemático.
- El método de control propuesto es validado tanto en simulación como en experimentación, ajustando cada una de las constantes en base a técnicas de ajuste de parámetros. De esta manera, el controlador permite ubicar el extremo del robot maestro en una posición deseada para resolver las tareas de evasión y simulación de peso.
- La simulación de robots permite la optimización de recursos económicos, donde el software elegido se selecciona en base a un análisis de características entre diversas opciones. V-REP demuestra que las propiedades cinemáticas y dinámicas contenidas en los robots simulados son adecuadas para la simulación de control tele operado, donde el robot maestro puede ingresar velocidades y a la vez usar la información del medio simulado para retornar fuerzas de respuesta.
- Todo el sistema es validado a través de dos pruebas experimentales, en las cuales se integra la modelación matemática, el controlador autónomo, el dispositivo háptico y los ambientes simulados. En ambos experimentos, se demuestra el uso del controlador para generar fuerzas para mantener el robot alejado de obstáculos y para simular el peso de diferentes objetos ubicados en el entorno virtual.
- El modelo matemático cinemático del dispositivo Falcon determinado en el presente caso de estudio, sirve como ayuda para el desarrollo y validación de algoritmos de control basados en la cinemática inversa del mismo, mediante las pruebas simuladas se puede ajustar las constantes de ganancia para el controlador.

- Para este tipo de aplicaciones y desarrollo de controladores se emplea el software Matlab, debido que es un programa que sirve para realizar cálculos numéricos a mayor velocidad de procesamiento, de igual manera contiene múltiples funcionalidades que permitió la lectura/escritura de datos en el dispositivo Háptico Falcon y de igual manera conectarse con el entorno 3D V-Rep.
- El software V- REP ya incorpora dentro de su entorno robots que ejecutan tareas que vienen pre programas para el robot móvil y robot manipulador, excepto el modelo matemático del dispositivo Falcon, la inclusión de un modelo mediante un software de Matemático, en este caso MATLAB y el planteamiento de una ley de control, permiten simular sin tener la necesidad de contar con el robot
- Se puede definir de mejor manera constantes que incorpora la ley de control, en el proyecto se hace uso de V-REP y MATLAB para realizar la tele-operación mediante un dispositivo háptico desde Matlab.

Recomendaciones

- Usar este trabajo como referencia para el desarrollo de tareas de tele operación, ya que se describe tanto el fundamento matemático como los métodos para controlar un robot paralelo.
- Plantear trabajos investigativos donde se implementen físicamente los robots esclavos, de forma que se valide la tele operación en campo y mediante el uso de diversos sensores empotrados al sistema robótico remoto o en espacios estratégicos del ambiente real.
- Saturar las fuerzas mínimas y máximas del dispositivo háptico para evitar daños en los actuadores del mecanismo robótico, así como empujar adecuadamente el robot hacia una superficie sólida para poderlo manipular en las aplicaciones propuestas.
- Trabajar en sistemas operativos basados en 32 bits para mejorar la compatibilidad con las librerías del dispositivo, o a su vez desarrollar librerías que permitan generar archivos *.dll que sean fácilmente ejecutables por paquetes informáticos basados en 64 bits.
- En base al análisis de estabilidad del controlador se puede ajustar las ganancias de acuerdo a la necesidad de la tarea, si se desea una mayor compensación se puede

ajustar de acuerdo a las condiciones propuestas en las respectivas ganancias del controlador.

REFERENCIAS

- A. F. Brumovsky, V. M. (s.f.). IMPLEMENTACION DE CONTROL DE ´. *Laboratorio de Rob´otica-FIUBA*, 2005.
- Allison, P. R. (15 de 07 de 2016). *bbc*. Obtenido de <http://www.bbc.com/future/story/20160714-what-does-a-bomb-disposal-robot-actually-do>
- Ariel, V. (26 de 11 de 2017). *blogspot*. Obtenido de FOTOS DE ROBOTS INDUSTRIALES: <http://ariel-a-viscarra.blogspot.com/2008/12/fotos-de-robots-industriales.html>
- Barrientos Sotelo, V. R. (2007). Robots M´oviles: Evoluci3n y Estado del Arte. *Polibits. Sistema de Informaci3n Cient´ıfica Redalyc* , 12-17.
- BasTech. (16 de 01 de 2018). *Innovative 3D Haptic Devices*. Recuperado el 2018 de 04 de 24, de <http://www.bastech.com/product-software/3d-haptic-devices/>
- BasTech. (28 de 5 de 2018). *INNOVATIVE 3D HAPTIC DEVICES*. Obtenido de <http://www.bastech.com/product-software/3d-haptic-devices/>
- Casman, J. (25 de 11 de 2017). *THETA 360 Developer*. Obtenido de <http://lists.theta360.guide:> <http://lists.theta360.guide/t/telepresence-robot-uses-ricoh-theta/943>
- CHAI3D. (s.f.). *Haptic Devices*. Recuperado el 2018 de 04 de 24, de <http://www.chai3d.org/download/doc/html/chapter6-haptic-devices.html>
- CONACYT. (03 de 11 de 2017). *Integraci3n de la rob3tica y la automatizaci3n al sector industrial*. Obtenido de <https://centrosconacyt.mx/objeto/robotica/>
- directindustry. (26 de 11 de 2017). *Direct Industry*. Obtenido de http://www.directindustry.es/fabricante-industrial/robot-paralelo-85295-_2.html
- Emmanuel Nuño Ortega, L. B. (25 de 11 de 2004). *Universidad Tecnica de Catalalunya*. Obtenido de <http://upcommons.upc.edu/handle/2117/570>
- Ferre, M., Buss, M., Aracil, R., Melchiorri, C., & Balaguer, C. (2007). *Springer Tracts in Advanced Robotics: Advances in Telerobotics*. Berlin: Springer-Verlag Berlin Heidelberg.

- Funes, J. (25 de 11 de 2017). *roboticaeducacion95*. Obtenido de <http://roboticaeducacion95.blogspot.com/2016/08/generalidades-de-la-robotica.html>
- Ge, S. S. (2006). *Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications*. USA: CRC Press.
- González, J. J., Caro, C. A., & González, H. V. (2015). Cinemática inversa de robot serial utilizando algoritmo genético basado en MCDS. *Open Journal Systems*, 1-12.
- Hernández-Martínez, Valdez-Peña, & Sánchez-Soto. (2013). A robust optimization hybrid algorithm for solving the direct kinematics of the general Gough-Stewart platform. *International Journal of numerical methods for Calculus and engineering design*, 241-247.
- Ibrahim, K., & Ali, A. S. (2016). Development a force feedback control of robot manipulator. *International Conference on Control, Automation and Robotics (ICCAR)*, 1-8.
- Islam, S., Dias, J., & Seneviratne, L. D. (2016). Robust coordination control interface for networked based telerobotic system. *Advanced Intelligent Mechatronics (AIM), IEEE International Conference on*.
- JOM. (02 de 11 de 2017). *Automatización industrial*. Obtenido de <http://estampacionesjom.com/automatizacion-industrial-estas-son-sus-aplicaciones/>
- Lenkaspace. (19 de 05 de 2018). *V-REP, Gazebo or ARGoS? A robot simulators comparison*. Obtenido de A DETAILED LOOK AT THE SIMULATOR CHARACTERISTICS: A DETAILED LOOK AT THE SIMULATOR CHARACTERISTICS
- Li, G. (26-27 de May de 2017). High-order iterative learning control for nonlinear systems. *Data Driven Control and Learning Systems (DDCLS)*.
- Milenio.com. (11 de 11 de 2017). *Reseña histórica de los robots paralelos*. Obtenido de http://www.milenio.com/firmas/universidad_politecnica_de_tulancingo/resena-historica-robots-paralelos-milenio_18_937886237.html

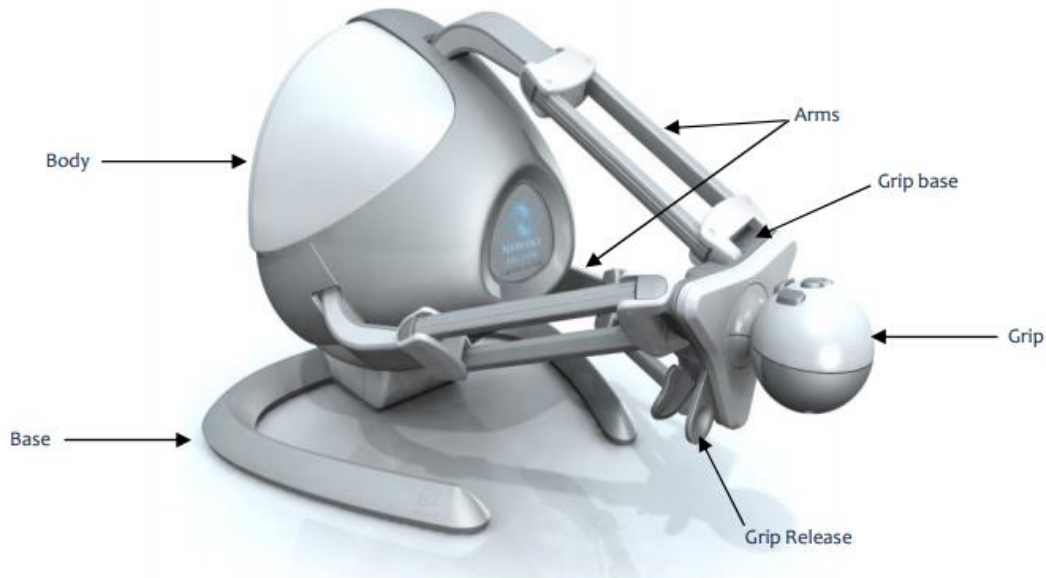
- Molina, V. A. (2011). *DISEÑO Y CONSTRUCCIÓN DE UN BRAZO ROBÓTICO INDUSTRIAL*. Ambato: UNIVERSIDAD TÉCNICA DE AMBATO .
- Monte, F. (10 de 05 de 2018). *Haptik Library*. Obtenido de <http://sirslab.dii.unisi.it/haptiklibrary/index.htm>
- Ollero, A. (2001). *Robótica: Manipuladores y robots móviles*. Barcelona: MARCOMBO.
- Ortmaier, T., Deml, B., Kübler, B., Passig, G., Reintsema, D., & Seibold, U. (2007). Robot Assisted Force Feedback Surgery. *Advances in Telerobotics*, 361-379.
- Park, T. M., Won, S. Y., Lee, S. R., & Sziebig, G. (2016). Force feedback based gripper control on a robotic arm. *Intelligent Engineering Systems (INES)*, 1-7.
- Rajan, N. S., Yadav, V., Krishnan, S. H., Gupta, A., & Ghosh, R. (2017). Straight line generation by a serial two-link manipulator: Design and analysis. *Instrumentation, Control, and Automation (ICA), 5th International Conference on*.
- Robologs. (22 de 05 de 2018). *Interacción entre V-Rep y Matlab*. Obtenido de <https://robologs.net/2017/07/04/interaccion-entre-v-rep-y-matlab/>
- Ruiz, A. P. (2012). *Asistencia háptica basada en planificación de movimientos para la teleoperación cooperativa de sistemas multirobot*. ESPAÑA: UNIVERSITAT POLITÈCNICA DE CATALUNYA.
- Salas, D. D. (2016). *Teleoperación con Retroalimentación de Fuerzas Aplicado a un Robot Móvil con restricción no Holonomica*. Quito: Escuela Politécnica Nacional.
- Santina, C. D., Bianchi, M., Grioli, G., Angelini, F., Catalano, M., Garabini, M., & Bicchi, A. (Septiembre de 2017). Controlling Soft Robots: Balancing Feedback and Feedforward Elements. *IEEE Robotics & Automation Magazine*, 24, 75-83.
- Sariyildiz, E., Yu, H., & Ohnishi, K. (2015). A Practical Tuning Method for the Robust PID Controller with Velocity Feed-Back. *Machines*, 208-222.
- Siciliano, B., & Khatib, O. (2008). *Handbook of Robotics*. Berlin: Springer-Verlag Berlin.
- Sinclair, P. (30 de 03 de 2017). *All Home Robotics*. Obtenido de Best Telepresence Robots for 2017: <https://www.allhomerobotics.com/best-telepresence-robots/>

- Surendran, A., & J, M. S. (2016). Sliding mode controller for robust trajectory tracking using haptic robot. *Power Electronics, Intelligent Control and Energy Systems (ICPEICES), IEEE International Conference on.*
- Taghirad, H. D. (2017). *Parallel Robots: Mechanics and Control*. USA: CRC Press .
- Theriault, A., Nagurka, M., & Johnson, M. J. (2014). Design and Development of an Affordable Haptic Robot with Force-Feedback and Compliant Actuation to Improve Therapy for Patients with Severe Hemiparesis. *IEEE Transactions on Haptics*, 161-174.
- Urrutia, D. R. (2016). *SISTEMA DE TELEOPERACIÓN BASADO EN INTERFÁZ HÁPTICA PARA BRAZO ROBÓTICO*. Chile: UNIVERSIDAD DE CHILE .
- Vertut, J. (1995). *Teleoperation and Robotics*. Netherlands: Springer Netherlands.
- Vilela, J. V., Figueredo, L. F., Ishihara, J. Y., & Borges, G. A. (March de 2016). Robust stability of networked control systems with dynamic controllers in the feedback loop. (IEEE, Ed.) *Control Conference (ICC)*.
- Wang, Y., Mao, Z., Lin, W., Wei, W., & Zhang, J. (2017). Newton-Euler method for dynamic modeling and control of parallel polishing manipulator. *Information and Automation (ICIA), IEEE International Conference on.*
- Wortham, R., & Theodorou, A. (2017). Robot transparency, trust and utility. *Connection Science*, 242-248 .

Anexos

Anexo 1. Instalación del Novint Falcon a Windows

Welcome To The Novint Falcon



Novint™, Novint Falcon™, Feelin' It™ Sports Pack, N VeNT™, Newton's Monkey Business™, and the Novint logo are trademarks of Novint Technologies, Inc. ©2007 Novint Technologies. All Rights Reserved.

NOVINT FALCON INTRODUCTION

Our sense of touch connects us to the world around us and is an integral part of how we experience things, both physically and emotionally. In the past, games have incorporated increasingly sophisticated graphics and sound design, but none have added realistic touch, until now.

The award-winning Novint Falcon makes virtual items and experiences feel real. Feel powerful touch feedback, textures, weight, and shapes when you play our Novint enabled PC games. Hold the Falcon's interchangeable Grip and feel a character's actions instead of controlling a game through mouse-clicks and button-presses. Feel the weight of a basketball as you shoot it towards a hoop -- the momentum and impact as you swing a virtual golf club and strike a ball -- the recoil of a weapon -- or the physical characteristics of virtual objects and environments.

In the same way the introduction of sound changed motion pictures, or graphics made the Internet what it is today, Novint's technology transforms the user experience by adding realistic interactive 3D touch to computing. While the concept of interactive touch in computing is as unimaginable to most people as the idea of television was in the age of radio, we envision a not-too-distant future when it is as commonplace and ubiquitous as a mobile phone or a computer mouse.

Novint is pioneering a new category of 3D touch products and we hope you enjoy your experience with the Novint Falcon by Novint Technologies, Inc. Thank you for taking this leap forward with us.

Health & Safety Information

Important Safety Information - Read the following warnings before setup or use of the Novint Falcon. If this product will be used by young children, this manual should be read and explained to them by an adult. Failing to do so may cause injury.

WARNING—Seizures

- Some people (about 1 in 4000) may have seizures or blackouts triggered by light flashes or patterns, and this may occur while they are watching TV or playing PC games, even if they have never had a seizure before.
- Anyone who has had a seizure, loss of awareness, or other symptom linked to an epileptic condition should consult a doctor before playing a PC game.
- Parents should watch their children play PC games. Stop playing and consult a doctor if you or your child has any of the following symptoms:

Convulsions	Eye or muscle twitching	Altered vision
Loss of awareness	Involuntary movements	Disorientation

- To reduce the likelihood of a seizure when playing PC games:

1. Sit or stand as far from the screen as possible.
2. Do not play if you are tired or need sleep.
3. Play in a well-lit room.
4. Take a 10-15 minute break every hour.

WARNING—Repetitive Motion Injuries & Eyestrain

Playing PC games can make your muscles, joints, skin or eyes hurt. Follow these instructions to avoid problems such as tendonitis, carpal tunnel syndrome, skin or irritation or eyestrain:

- Avoid excessive play. Parents should monitor their children for appropriate play.
- Take a 10-15 minute break every hour, even if you don't think you need it.
- If your hands, wrists, arms or eyes become tired or sore while playing or if you feel symptoms such as tingling, numbness, burning or stiffness, stop and rest for several hours before playing again.
- If you continue to have any of the above symptoms or other discomfort during or after play, stop playing and see a doctor.

Health & Safety Information

WARNING—Motion Sickness

Playing PC games can cause motion sickness in some players. If you or your child feels dizzy or nauseous when playing PC games, stop playing and rest. Do not drive or engage in other demanding activity until you feel better.

WARNING—Software for use only with Novint Falcon and approved accessories and interface devices

The software is for use only with the Novint Falcon™ interface device, and accessories and other interface devices that have been approved by Novint. Use of the software with any other accessory or interface device may result in injury to the user or others, or damage to the interface device or other property. Use of the software with an interface device or accessory that has not been approved by Novint voids all warranties.

WARNING—Device for use only with Novint-approved software and accessories

The Falcon is for use only with software and accessories that have been approved by Novint. Use of the Falcon with any other software or accessories may result in injury to the user or others, or damage to the Falcon or other property. Use of the Falcon with software or accessories that have not been approved by Novint voids all warranties.

WARNING—Electric Shock Warning

To avoid electric shock when you use this system:

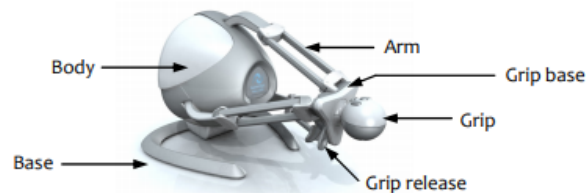
- Do not use the Falcon during a lightning storm. There may be a risk of electric shock from lightning.
- Use only the power supply that comes with your system.
- Do not use the power supply if it has damaged, split or broken cords or wires.
- Make sure that the power supply cord is fully inserted into the wall outlet or extension cord.
- Always carefully disconnect all plugs by pulling on the plug and not on the cord.

WARNING—Radio Frequency Interference

The Falcon can emit radio waves that can affect the operation of nearby electronics, including cardiac pacemakers. Do not operate the Falcon within 9 inches of a pacemaker. If you have a pacemaker or other implanted medical devices, do not use the Falcon without first consulting your doctor or the manufacturer of your medical device.

Setting Up the Falcon

- (1) Open the top of the Falcon Styrofoam box. The top is marked with an indented arrow in the Styrofoam which indicates that it is right-side up.
- (2) Verify the Box Contents:
 - Novint Falcon
 - 2 Falcon Grips
 - Power Supply
 - USB Cable
 - Novint Game DVD
 - Driver Installation CD
 - Novint Falcon User Manual
 - Novint Games Manual
- (3) Remove the Falcon from the box by its body, making sure you support the base. Never move or pick up the Falcon by its arms.



- (4) Attach one of the Falcon Grips by lining up the U-shaped rod on the Grip to the U-shaped hole on the Grip base. Push the Grip forward into the Grip base until it snaps into place. (To remove the Grip, please refer to p. 13.)

Follow instructions for installing the Falcon Driver and the Novint Games.

Installing the Falcon Driver

- (1) Insert the Driver Installation CD into your CD or DVD-ROM drive. This should automatically start up the install wizard.
 - *XP USERS: If CD does not run automatically, select **Start->Run**, type `D:\setup.Falcon.exe` (where `D:` is the letter of your CD-ROM drive) and click on **OK***
 - *VISTA USERS: If CD does not run automatically, select **Start->All Programs->Accessories->Run**, type `D:\setup.Falcon.exe` (where `D:` is the letter of your CD-ROM drive) and click on **OK***
- (2) At the Falcon Welcome screen, click on **Next>**
- (3) Click on "I agree to the terms of this license agreement" and click on **Next>**
- (4) Enter your User Information, click on **Next>**
- (5) At the Installation Folder screen, keep the default location and click on **Next>**
- (6) In the Ready to Install screen, click on **Next>**
- (7) Click on **Finish** when the Installation Successful screen appears.
- (8) When the next screen comes up, you can choose to reboot your PC now or later.
 - ◆ *Review the instructions below to create a restore point before installing 3rd party software.*
 - ◆ *Review the system requirements on page 16 before installing the Novint Games.*

Creating a Restore Point for Third-Party Software Installation

We recommend that prior to installing any third-party software, you create a restore point to which you can return should you encounter any difficulties.

- (1) Click on **Start->Programs->Accessories->System Tools->System Restore**
- (2) Select "Create a restore point" and then click on **Next>**
- (3) You will be prompted to supply a name for this setup.

If you encounter any difficulties during the installation process, you can follow the same path as above to restore your PC to its pre-installed state.

Installing the Novint Games

- (1) Insert the Novint Game DVD into your DVD-ROM drive. This should automatically start up the install wizard.
 - *XP USERS:** If the DVD does not run automatically, select **Start->Run**, type **D:\setup.NVeNT.exe** (where **D:** is the letter of your DVD-ROM drive) and click on **OK**
 - *VISTA USERS:** If DVD does not run automatically, select **Start->All Programs->Accessories->Run**, type **D:\setup.NVeNT.exe** (where **D:** is the letter of your DVD-ROM drive) and click on **OK**
 - (2) At the N VeNT Welcome screen, click on **Next>**
 - (3) Click on "I agree to the terms of this license agreement" and click on **Next>**
 - (4) Enter your User Information, click on **Next>**
 - (5) At the Installation Folder screen, keep the default location and click on **Next>**
 - (6) Select a Shortcut folder. Choose if shortcut is available to current user or to all users
 - (7) In the Ready to Install screen, click on **Next>**. The status of the installation will appear.
 - (8) Click on **Finish** when the Installation Successful screen appears.
 - (9) When the next screen comes up, you can choose to reboot your PC now or later.
- N VeNT player and downloader, Newton's Monkey Business™, and Feelin' It™ Sports Pack are now installed on your computer.
 - The N VeNT icon shortcut is placed on your desktop.



N VeNT

Continue to the Falcon Setup Completion

Complete the Falcon Setup

- (1) Insert the USB Cable into the back of the Falcon and into your computer's USB slot. Your computer will prompt you to complete instructions for installing new hardware.
 - *XP Users:**
 - "Found New Hardware" wizard starts up
 - Click on **No** for "Can Windows connect", click on **Next>**
 - Install from a list, click on **Next>**
 - Search for the best driver. Include location **Program Files/Novint/Falcon/USBDriver**
 - Popup will say the driver hasn't passed Windows Logo Testing. Continue anyway
 - Click on **Finish**
 - *Vista Users:**
 - When popup window appears, select one of three choices: **Locate & Install**; **Ask Later**; **Don't Show Again for this Device**
 - Vista will automatically try to find and install the driver. This will fail.
 - Vista asks for disc for device. Select "**Don't Have Disc, Show Other Options**"
 - Browse the computer and go to **Program Files/Novint/falcon/USBDriver**
 - Click on **Next>** and drivers will be installed.
- (2) Plug in the Power Supply to a standard 120 volt outlet and into the back of the Novint Falcon.



Falcon USB Cable

Falcon Power

Game Control Overview

You are almost ready to play the Novint Games. Let's get comfortable with the Falcon first.

The Novint Falcon is an entirely new type of game controller. It is, essentially, a small robot that replaces your mouse or joystick, to let you experience true virtual touch unlike any controller in history. The Novint Falcon lets you control a game in three dimensions, and feel high-fidelity three-dimensional force feedback.

The Falcon Grip moves right and left, forwards and backwards, like a mouse, but also moves up and down. When you hold the Falcon's detachable Grip and move your cursor to interact with a virtual object, environment, or character, motors in the device turn on and are updated approximately 1000 times a second, letting you feel texture, shape, weight, and dimension.

The Falcon lets you control and interact with games in a more realistic way, allowing you to develop real physical skill and muscle memory, adding a realistic new dimension to gaming.

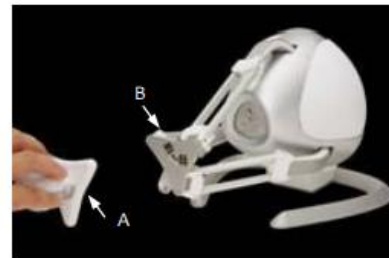
Removing and Attaching the Falcon Grip

Your Limited Edition Novint Falcon Bundle comes with two Grips or handles that are interchangeable. We plan to offer additional Grips at www.novint.com which may provide different shapes and colors and unique game functionality (such as a steering wheel for driving games, or a pistol Grip for first-person shooters).

To remove the Grip, squeeze the Grip Release located on the left lower side of the Grip base. While it is pinched together, pull the Grip away from the Grip base.



Grip Release

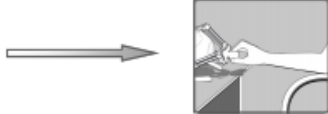


Grip (A) pulled away from Grip base (B)

To attach a new Grip, simply line up the U-shaped rod on the Grip with the U-shaped hole on the Grip base. Then push the Grip forward into the Grip base until it snaps into place.

Ergonomics of the Novint Falcon

Using your Falcon comfortably

1. Novint recommends two workspace setups for using the Falcon to prevent fatigue and injury:
 - 1a. Rest your elbow on your chair's arm rest so that it supports your arm's weight while you are using the Falcon. Ideally you should use a chair that has adjustable arm rests. 
 - 1b. Keep your arm parallel to the ground and not touching the table.
2. While using the Falcon, **maintain your wrist at a neutral position** so that there is no bend from your wrist to your hand.
3. Make sure the Falcon is clear of other objects on your desk (including food and drinks).
4. Never place any part of your body other than your hand near the Falcon's workspace.
5. Take breaks from game play often.

Caution when using your Falcon

- Although the Falcon is meant to be robust, it is still a precision instrument. **DO NOT BE TOO ROUGH IN YOUR MOVEMENTS WITH THE FALCON.** Avoid excessive force at the edges of the movement range to avoid damage to any of its components.
- Be sure that you don't let go of the Falcon Grip while you are using it, especially when there is a big force.
- Don't leave small children unattended while using the Novint Falcon.



DON'T PUSH THE FALCON GRIP
IN ANY DIRECTION WITH
TOO MUCH FORCE

System Requirements for Falcon Bundle

The following specifications are the minimum and recommended system requirements to run the Falcon Bundle which includes Newton's Monkey Business™.

Minimum Requirements	
Processor	1.0 GHz Processor
OS	Windows XP Service Pack 2, Windows Vista
Graphic Card	128Mb 3D hardware accelerated graphics card
DirectX Version	DirectX 9.0c
Hard Drive	1.5 GB free disk space
Memory	512 MB RAM
Other	DVD-ROM (CDs available upon request)
USB	2.0 connection

Recommended Requirements	
Processor	2.4 GHz Processor
OS	Windows XP Service Pack 2, Windows Vista
Graphic Card	256Mb 3D hardware accelerated graphics card
DirectX Version	DirectX 9.0c
Hard Drive	1.5 GB free disk space
Memory	1 GB RAM
Other	Broadband Internet Connection
USB	2.0 connection

Anexo 2. Programación de la simulación del Novint Falcon

En esta simulación se considera ganancias con que mejor tubo rendimiento el controlador durante la simulación, cumpliendo con las condiciones preestablecida en el análisis de estabilidad del controlador.

A. Programa principal

```
clc
clear all
close all
n=50;
p=linspace(0,4*pi,n);
r=30;

xd=r*cos(p);
yd=(2*pi)*4*ones(1,length(xd));
zd=r*sin(p);

th1=[1 1 1]'*0.7862;
th2=[1 1 1]'*2.0967;
th3=[1 1 1]'*1.561;
[x,y,z]=Directa(th1(1),th2(1),th3(1));
fig1=figure;
fig2=fig1;
fig3=fig1;

Kp=0.5;Ki=0.33;Kd=0.3;
ts=0.1;

I_ERR=[0;0;0];
for i=1:length(xd)

    errx(i)=xd(i)-x(i);
    erry(i)=yd(i)-y(i);
    errz(i)=zd(i)-z(i);

    ERR=[errx(i);erry(i);errz(i)];
    ErrorT(i,:)= ERR;
    %derivada del error
    if i>1
        D_ERR=[errx(i)-errx(i-1);erry(i)-erry(i-1);errz(i)-errz(i-1)]/ts;
    else
        D_ERR=ERR/ts;
    end
    %integral del error
    I_ERR=I_ERR+ERR*ts;

    [JJ1,JJ2,JJ3]=MatrizJ(th1,th2,th3);
    %Control PID
    Tp1=inv(JJ1)*(Kp*ERR+Ki*I_ERR+Kd*D_ERR);
    Tp2=inv(JJ2)*(Kp*ERR+Ki*I_ERR+Kd*D_ERR);
    Tp3=inv(JJ3)*(Kp*ERR+Ki*I_ERR+Kd*D_ERR);

    % acciones de control
    tp1=[Tp1(1);Tp2(1);Tp3(1)];
```

```

tp2=[Tp1(2);Tp2(2);Tp3(2)];
tp3=[Tp1(3);Tp2(3);Tp3(3)];
accC1(i,:)= tp1; accC2(i,:)= tp2; accC3(i,:)= tp3;
%integrar
th1=th1+tp1*ts;
th2=th2+tp2*ts;
th3=th3+tp3*ts;
[x(i+1),y(i+1),z(i+1)]=Directa(th1(1),th2(1),th3(1));
delete([fig1 fig2 fig3]);
fig1=Brzo(th1,th2,th3);%plot del brazo
fig2=plot3(z(1:i),x(1:i),y(1:i),'m','linewidth',2);%plot de
extremo
fig3=plot3(zd,xd,yd,'g','linewidth',2);%plot del deseado
view(50,15)
grid on
axis equal
axis([-151 40 -90 90 -60 100])
pause(0.02)
end

set(gca, 'fontweight','bold','fontsize',12)
% legend([fig2 fig3],'Trayectoria seguida','Trayectoria deseada')
xlabel('Milímetros'),ylabel('Milímetros'),zlabel('Milímetros')
%%
width=2;
figure(2)
plot(errx,'r','linewidth',width)
hold on
plot(erry,'g','linewidth',width)
plot(errz,'b','linewidth',width)
grid on
set(gca, 'fontweight','bold','fontsize',12)
legend('Error en X','Error en Y','Error en Z')
axis([0 52 -100 100])
ylabel('Amplitud [mm]'), xlabel('Tiempo [s]')
%%
figure(3)
plot(accC1,'linewidth',width)
grid on
legend('Acciones de control en X','Acciones de control en Y','Acciones
de control en Z')
set(gca, 'fontweight','bold','fontsize',12)
ylabel('Velocidad [cm/s]'), xlabel('Tiempo [s]')
axis([0 52 -2 2.5])

```

B. Función Matriz J

```

function [JJ1,JJ2,JJ3]=MatrizJ(th1,th2,th3)
a=60;
b=102.5;
c=15.7;
d=11.5;
e=11.5;
f=26.2;
r=36.3;
s=27.2;
th11=th1(1); th12=th2(1);th13=th3(1);
th21=th1(2); th22=th2(2);th23=th3(2);
th31=th1(3); th32=th2(3);th33=th3(3);

```



```

phi1=105*pi/180;
R1=[cos(phi1) sin(phi1) 0;
    -sin(phi1) cos(phi1) 0;
    0 0 1];
J1 =[[ -a*sin(th11), -sin(th12)*(d + e + b*sin(th13)),
b*cos(th12)*cos(th13)];
    [ 0, 0, -
b*sin(th13)];
    [ a*cos(th11), cos(th12)*(d + e + b*sin(th13)),
b*cos(th13)*sin(th12)]];
JJ1=inv(R1)*J1;

phi2=345*pi/180;
R2=[cos(phi2) sin(phi2) 0;
    -sin(phi2) cos(phi2) 0;
    0 0 1];
J2 =[[ -a*sin(th21), -sin(th22)*(d + e + b*sin(th23)),
b*cos(th22)*cos(th23)];
    [ 0, 0, -
b*sin(th23)];
    [ a*cos(th21), cos(th22)*(d + e + b*sin(th23)),
b*cos(th23)*sin(th22)]];
JJ2=inv(R2)*J2;

phi3=225*pi/180;
R3=[cos(phi3) sin(phi3) 0;
    -sin(phi3) cos(phi3) 0;
    0 0 1];
J3 =[[ -a*sin(th31), -sin(th32)*(d + e + b*sin(th33)),
b*cos(th32)*cos(th33)];
    [ 0, 0, -
b*sin(th33)];
    [ a*cos(th31), cos(th32)*(d + e + b*sin(th33)),
b*cos(th33)*sin(th32)]];
JJ3=inv(R3)*J3;

```

C. Función Directa

```

function [x,y,z]=Directa(th1,th2,th3)
a=60;
b=102.5;
c=15.7;
d=11.5;
e=11.5;
f=26.2;
r=36.3;
s=27.2;

pul=a*cos(th1)+(d+e+b*sin(th3))*cos(th2)-c;
pv1=b*cos(th3)+f;
pw1=a*sin(th1)+(d+e+b*sin(th3))*sin(th2);
PT=[pul;pv1;pw1];

phi1=105*pi/180;
R=[cos(phi1) sin(phi1) 0;
    -sin(phi1) cos(phi1) 0;
    0 0 1];
PB=inv(R)*(PT-[-r;s;0]);

```

```
PH=PB-[0;0;151];
x=PH(1);    y=PH(2);    z=PH(3);
```

D. Función Brazo

```
function figu=Brazo(th1,th2,th3)
a=60;
b=102.5;
c=15.7;
d=11.5;
e=11.5;
f=26.2;
r=36.3;
s=27.2;

th11=th1(1); th12=th2(1);th13=th3(1);
th21=th1(2); th22=th2(2);th23=th3(2);
th31=th1(3); th32=th2(3);th33=th3(3);

p0=[0;0;0];
p1=[a*cos(th11);0;a*sin(th11)];
p2=[a*cos(th11)+(d+e+b*sin(th13))*cos(th12);b*cos(th13);
    a*sin(th11)+(d+e+b*sin(th13))*sin(th12)];
p3=[a*cos(th11)+(d+e+b*sin(th13))*cos(th12)-c;b*cos(th13)+f;
    a*sin(th11)+(d+e+b*sin(th13))*sin(th12)];

phi1=105*pi/180;
R=[cos(phi1) sin(phi1) 0;
    -sin(phi1) cos(phi1) 0;
    0 0 1];
p00=inv(R)*[-r;s;0];
p11=inv(R)*(p1-[-r;s;0]);
p22=inv(R)*(p2-[-r;s;0]);
p33=inv(R)*(p3-[-r;s;0]);

Pu=[0;p00(1);p11(1);p22(1);p33(1)];
Pv=[0;p00(2);p11(2);p22(2);p33(2)];
Pw=[0;p00(3);p11(3);p22(3);p33(3)];
figu(1)=plot3(Pw-151,Pu,Pv,'b','linewidth',2);

p1=[a*cos(th21);0;a*sin(th21)];
p2=[a*cos(th21)+(d+e+b*sin(th23))*cos(th22);b*cos(th23);
    a*sin(th21)+(d+e+b*sin(th23))*sin(th22)];
p3=[a*cos(th21)+(d+e+b*sin(th23))*cos(th22)-c;b*cos(th23)+f;
    a*sin(th21)+(d+e+b*sin(th23))*sin(th22)];
phi2=345*pi/180;
R2=[cos(phi2) sin(phi2) 0;
    -sin(phi2) cos(phi2) 0;
    0 0 1];
p00=inv(R2)*[-r;s;0];
p11=inv(R2)*(p1-[-r;s;0]);
p22=inv(R2)*(p2-[-r;s;0]);
p33=inv(R2)*(p3-[-r;s;0]);

Pu=[0;p00(1);p11(1);p22(1);p33(1)];
Pv=[0;p00(2);p11(2);p22(2);p33(2)];
Pw=[0;p00(3);p11(3);p22(3);p33(3)];
hold on
```

```

figu(2)=plot3(Pw-151,Pu,Pv,'r','linewidth',2);

p1=[a*cos(th31);0;a*sin(th31)];
p2=[a*cos(th31)+(d+e+b*sin(th33))*cos(th32);b*cos(th33);
    a*sin(th31)+(d+e+b*sin(th33))*sin(th32)];
p3=[a*cos(th31)+(d+e+b*sin(th33))*cos(th32)-c;b*cos(th33)+f;
    a*sin(th31)+(d+e+b*sin(th33))*sin(th32)];
phi3=225*pi/180;
R3=[cos(phi3) sin(phi3) 0;
    -sin(phi3) cos(phi3) 0;
    0 0 1];
p00=inv(R3)*[-r;s;0];
p11=inv(R3)*(p1-[-r;s;0]);
p22=inv(R3)*(p2-[-r;s;0]);
p33=inv(R3)*(p3-[-r;s;0]);

Pu=[0;p00(1);p11(1);p22(1);p33(1)];
Pv=[0;p00(2);p11(2);p22(2);p33(2)];
Pw=[0;p00(3);p11(3);p22(3);p33(3)];
figu(3)=plot3(Pw-151,Pu,Pv,'g','linewidth',2);

rad=norm([r s]);
yb=[rad*cos(0:0.1:3) rad*cos(3:-0.1:0)];
zb=[rad*sin(0:0.1:3) -rad*sin(3:-0.1:0)];
xb=yb*0;
[X,Y,Z]=sphere(15);
tam=10;
figu(4)=surf(tam*X+p33(3)-151,tam*Y+p33(1),tam*Z+p33(2),Z*0);
shading interp
camlight
lighting phong
set(figu(4),'facecolor',[.98 .45 .02]);
figu(5)=patch(xb-151,yb,zb,'m');
% alpha(0.5)

```

Anexo 3. Programación de los algoritmos de control del Novint Falcon

Para esta prueba se realiza colocando ganancias de $K_p=0.1; K_i=0.3; K_d=0.01$;
Según las simulaciones para tener una estabilidad adecuada en el extremo operativo del Falcon.

```
clc
clearvars -except h
warning off
close all
n=1000;
p=linspace(0,4*pi,n);
r=25;
xd=r*cos(p);
yd=(p-2*pi)*3;
zd=r*sin(p)-10;

% Iniciar falcon
if ~exist('h')
    cd 'C:\Program Files (x86)\Siena Robotics and Systems Lab\Haptik
Library\Matlab'
    h = haptikdevice;
    pause(5)
    cd 'D:\Documentos Roberth\MAESTRIA UTA\TESIS\Control
Programas\Programacion\ControlReal'

end
write(h,[0 0 -2]);
pause(0.5)

fig1=figure;
fig2=fig1;
fig3=fig1;

Kp=0.1;Ki=0.3;Kd=0.01;
% Kp=0.48;Ki=6;Kd=0.01;
ts=0.03;
I_ERR=[0;0;0];
for i=1:length(xd)
    tic
    pos = read_position(h);%lee los datos del falcon
    x(i)=pos(1);
    y(i)=pos(2);
    z(i)=pos(3);

    errx(i)=xd(i)-x(i);
    erry(i)=yd(i)-y(i);
    errz(i)=zd(i)-z(i);
    ERR=[errx(i);erry(i);errz(i)];
    if i>1
        D_ERR=[errx(i)-errx(i-1);erry(i)-erry(i-1);errz(i)-errz(i-
1)]/ts;
    else
        D_ERR=0;%ERR/ts;
    end
    I_ERR=I_ERR+ERR*ts;

%Control PID
F=Kp*ERR+Ki*I_ERR+Kd*D_ERR;
lim=5;
```

```

F=F.*double(abs(F)<lim)+lim*(double(abs(F)>lim)).*sign(F);

write(h,[F(1) F(2) F(3)]);% escribe las oredene de los brazos

[th1,th2,th3]=Inversa(x(i),y(i),z(i));
delete([fig1 fig2 fig3]);
fig1=Brazo(th1,th2,th3);
fig2=plot3(z(1:i),x(1:i),y(1:i),'k');
fig3=plot3(zd,xd,yd,'g');
view(50,15)
grid on
axis equal
axis([-181 40 -90 90 -60 100])
pause(0.00001)
ts=toc;
end
write(h,[0 0 0]);
%%
figure,subplot(3,1,1),plot(errx),grid on
subplot(3,1,2),plot(erry),grid on
subplot(3,1,3),plot(errz),grid on

```

Anexo 4. Programación del control real del Novint Falcon para el robot móvil

Para esta prueba se realiza colocando ganancias de $K_p=0.1$; $K_i=0.3$; $K_d=0.01$;
Según las simulaciones para tener una estabilidad adecuada en el extremo operativo del Falcon.

```
clc
clearvars -except h
warning off
close all

% Iniciar falcon
if ~exist('h')
    cd 'C:\Program Files (x86)\Siena Robotics and Systems Lab\Haptik
Library\Matlab'
    h = haptikdevice;
    pause(5)
    cd 'D:\Documentos Roberth\MAESTRIA UTA\TESIS\Control
Programas\Programacion\ControlReal'
end

Kp=0.1;Ki=0.3;Kd=0.01;

%Conectarse con el simulador
vrep=remApi('remoteApi');
clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,10);
%Crear un handle para los motores
[errmi, motor_izquierdo]=vrep.simxGetObjectHandle(clientID,
'Pioneer_p3dx_leftMotor', vrep.simx_opmode_one-shot_wait);
[errmd, motor_derecho]=vrep.simxGetObjectHandle(clientID,
'Pioneer_p3dx_rightMotor', vrep.simx_opmode_one-shot_wait);
%conectarse con sensor de coliccion
for i=9:16 %8 sensores del 9 al 16
    [errs, sensor(i)] = vrep.simxGetObjectHandle(clientID,
strcat('Pioneer_p3dx_ultrasonicSensor',num2str(i)),
vrep.simx_opmode_one-shot_wait);
end
% crear la camara
[errc, camara] = vrep.simxGetObjectHandle(clientID, 'Vision_sensor',
vrep.simx_opmode_one-shot_wait);

%parametros
ts=0.1;% tiempo de muestreo
prof(16)=0.4;% profundidad mini a dtectar

I_ERR=[0;0;0];% inicializar la integral de error

angulos=[0:-25.5:-180 180:-25.5:0];% angulo de los sensor
i=1;
% deseados
xd=0;
yd=0;
zd=0;
while read_button(h)~=1
    % while (vrep.simxGetConnectionId(clientID)~=-1)
    tic% medir el tiempo de ciclo
```

```

pos = read_position(h);% leer el falcon

errx(i)=xd-pos(1);
erry(i)=yd-pos(2);
errz(i)=zd-pos(3);

ERR=[errx(i);erry(i);errz(i)];%errores
% derivar los error
if i>1
    D_ERR=[errx(i)-errx(i-1);erry(i)-erry(i-1);errz(i)-errz(i-
1)]/ts;
else
    D_ERR=0;%no har derivada en i=1;
end
% integral del error
I_ERR=I_ERR+ERR*ts;

%Control PID
F=Kp*ERR+Ki*I_ERR+Kd*D_ERR;
% enviar las fuerzas del PID
write(h,[F(1) F(2) F(3)]);

%Velocidades para el robot

pos=(abs(pos)>15).*(pos-sign(pos)*15);%esfera de 15

% convertir estos datos en velocidades del robot
u=pos(3)*0.05;% vel lineal
w=-pos(1)*0.05;% vel angular
% calculo la vel de cada rueda
rd=u+0.25*w;
ri=u-0.25*w;

%Cambiar la velocidad de los motores en el simulador
vrep.simxSetJointTargetVelocity(clientID,
motor_izquierdo,ri,vrep.simx_opmode_streaming);
vrep.simxSetJointTargetVelocity(clientID,
motor_derecho,rd,vrep.simx_opmode_streaming);
% sensar la collicion
for j=9:16
    [errs, estado, punto] = vrep.simxReadProximitySensor(clientID,
sensor(j), vrep.simx_opmode_streaming);
    prof(j)=punto(3);% punto de coliccion, solo el eje z (3)
    % discrimina de profundidad
    if prof(j)==0;prof(j)=max(prof);end
    % profunidd se descompone en x y
    px(j)=prof(j)*cosd(angulos(j));
    py(j)=prof(j)*sind(angulos(j));
end
%correccion de valores deseados
w_corr=sum(prof(9:12))-sum(prof(13:16));% correccion de w
xd=w_corr*400;% la nueva posicion deseada en xd

u_corr=sum(max(abs(prof))-prof(11:14));%correccion de u
zd=u_corr*800;% la nueva posicion deseada en zd
%plot lidar
% plot(px,py)

%lectura de camera

```

```

[errorCode, resolution, img]=vrep.simxGetVisionSensorImage2(clientID, cam
ara, 0, vrep.simx_opmode_oneshot_wait);
    imshow(img); % mostrar la imagen

    pause(0.001); % pausa la leer los datos del falcon

    i=i+1;
    ts=toc;
end
write(h, [0 0 0]);
%%
rd=0;
ri=0;
%% Finalizar
% detiene lo motores
vrep.simxSetJointTargetVelocity(clientID,
motor_izquierdo, ri, vrep.simx_opmode_streaming);
vrep.simxSetJointTargetVelocity(clientID,
motor_derecho, rd, vrep.simx_opmode_streaming);
% detiene el simulador
vrep.simxStopSimulation(clientID, vrep.simx_opmode_oneshot_wait);
vrep.simxFinish(clientID);

```


Anexo 4. Programación del control real del Novint Falcon para el robot manipulador

Para esta prueba se realiza colocando ganancias de $K_p=0.1$; $K_i=0.3$; $K_d=0.01$;
Según las simulaciones para tener una estabilidad adecuada en el extremo operativo del Falcon.

```
clc
clearvars -except h
warning off
close all

% Iniciar falcon
if ~exist('h')
    cd 'C:\Program Files (x86)\Siena Robotics and Systems Lab\Haptik
Library\Matlab'
    h = haptikdevice;
    pause(5)
    cd 'D:\Documentos Roberth\MAESTRIA UTA\TESIS\Control
Programas\Programacion\ControlReal'
end

Kp=0.1;Ki=0.3;Kd=0.01;

%Conectarse con el simulador
vrep=remApi('remoteApi');
clientID=vrep.simxStart('127.0.0.1',19999,true,true,5000,10);
% %conectar el brazo
[errV, target]=vrep.simxGetObjectHandle(clientID, 'target',
vrep.simx_opmode_oneshot_wait);
% comerctar gripper
[errg, Gripper]=vrep.simxGetObjectHandle(clientID,
'Gripper_closeJoint', vrep.simx_opmode_oneshot_wait);
% sensor de fuerza
[errs, sensor] = vrep.simxGetObjectHandle(clientID, 'sensorForce',
vrep.simx_opmode_oneshot_wait);
%conectar el angulos 1
[erra, angl]=vrep.simxGetObjectHandle(clientID, 'redundantRob_joint1',
vrep.simx_opmode_oneshot_wait);

ts=0.1;% tiempo de muestreo
I_ERR=[0;0;0]; % integral del error
i=1;

xd=0;
yd=0;
zd=0;
% h es el extremo del robot virtual
hx=1.22;
hy=-0.775;
hz=1.21;
while read_button(h)~=1
% while (vrep.simxGetConnectionId(clientID)~=-1)
    tic
    pos = read_position(h);% leer posicion del falcon
    % errores
    errx(i)=xd-pos(1);
    erry(i)=yd-pos(2);
```

```

errz(i)=zd-pos(3);

ERR=[errx(i);erry(i);errz(i)];
% derivada del error
if i>1
    D_ERR=[errx(i)-errx(i-1);erry(i)-erry(i-1);errz(i)-errz(i-
1)]/ts;
else
    D_ERR=0;% no hay derivada
end
% integral error
I_ERR=I_ERR+ERR*ts;

%Control PID
F=Kp*ERR+Ki*I_ERR+Kd*D_ERR;
% evniar al faclon el PID
write(h,[F(1) F(2) F(3)]);

%Velocidades para el robot
% acodicionar con una esfera
pos=(abs(pos)>15).* (pos-sign(pos)*15);%esfera
% leer el angulo 1
[errf th] = vrep.simxGetJointPosition(clientID, angl,
vrep.simx_opmode_streaming);
% matriz de rotacion para coordinar el brazo segun th
fb=0.00015; % factor de velocidad del brazo virtual
hx=hx-(pos(3)*cosd(th)-pos(1)*sind(th))*fb;
hy=hy-(pos(3)*sind(th)+pos(1)*cosd(th))*fb;
% altura del brazo virtual
hz=hz+pos(2)*fb;
%control del gripper
b=read_button(h);

if b==2;
    g=0.01;% cerrar la garra, velocidad positivo m/s
elseif b==8
    g=-0.01;% abrir la garra, velocidad negativa
else
    g=0;% detiene la garra
end
% mueve el extremo del brazo virtual
vrep.simxSetObjectPosition(clientID, target,-1,[hx hy
hz],vrep.simx_opmode_oneshot);
% mueve la garra virtual
vrep.simxSetJointTargetVelocity(clientID,
Gripper,g,vrep.simx_opmode_streaming);
% leer el sensor de fuerza
[errf estado fuerza] = vrep.simxReadForceSensor(clientID, sensor,
vrep.simx_opmode_streaming);
% acodcionamiento de fuerza
fuerza=(abs(fuerza)>20).* (fuerza-sign(fuerza)*20);% 20 N es la
fuerza minima de sensado, su propia masa
% modificar el yd
yd=fuerza(3)*6;
pause(0.1);% permite ller el falcon

i=i+1;
ts=toc;
end
write(h,[0 0 0]);

```

```
%% detiene el simulador
vrep.simxStopSimulation(clientID,vrep.simx_opmode_oneshot_wait);
vrep.simxFinish(clientID);
```