

UNIVERSIDAD DE PAMPLONA
MAESTRÍA EN CONTROLES INDUSTRIALES



**ASISTENCIA HÁPTICA PARA EL GUIADO DE UN ROBOT CUYO
ALGORITMO DE CONTROL ES GENERADO A PARTIR DE LA
INFORMACIÓN OBTENIDA DE UNA INTERFAZ CEREBRO COMPUTADOR**

Ing. ANDRÉS LEONARDO VARGAS GRANADOS
Autor

Ph.D CÉSAR AUGUSTO PEÑA CORTÉS
Director

Ph.D JAVIER ADOLFO CORREDOR CAMARGO
Co - Director

**TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE
MAGÍSTER EN CONTROLES INDUSTRIALES**

PAMPLONA, NORTE DE SANTANDER
Julio de 2021

Agradecimientos

A Dios por darme perseverancia para no desistir en los momentos difíciles y sabiduría para enfrentar los retos que demandaron este proyecto.

A mis padres José Gregorio Vargas Carreño e Isabel Granados, a mis Hermanos Edgar y Freddy, que siempre me han inspirado a seguir adelante sin importar los obstáculos que se presenten y me han apoyado para alcanzar las metas.

A mi pareja Eddy y a mis hijos Itzel e Isaac, quienes con su apoyo, comprensión y amor incondicional, me motivaron y me impulsaron para no desfallecer.

A mis directores Ph.D César Augusto Peña Cortes y Ph.D Javier Adolfo Corredor Camargo, quienes con sus conocimientos, tiempo, motivación, enseñanzas y todo el apoyo en cada una de las etapas para alcanzar los resultados de este proyecto.

Al M.Sc Holger Antonio Cacia Rivera y al ingeniero en formación Manuel Esteban Contreras B, personas que me brindaron su confianza, amistad y apoyo incondicional en los buenos y malos momentos durante el desarrollo de este proyecto.

Ing. Andrés Leonardo Vargas Granados

Índice General

Agradecimientos	II
Resumen	VII
Abstract	VIII
1. Introducción	1
1.1. Planteamiento del problema y justificación	2
1.2. Objetivo general	2
1.2.1. Objetivos específicos	2
1.3. Estructura del documento	3
2. Marco teórico y estado de arte	4
2.1. Robótica móvil	4
2.1.1. Generalidades	4
2.1.1.1. Histórico de los robots móviles	4
2.2. Tipos de robots móviles	6
2.2.1. Robots Terrestres	6
2.2.1.1. Robots con orugas	6
2.2.1.2. Robots con ruedas	7
2.2.1.3. Robots con tracción diferencial	8
2.3. Interfaces hápticas	10
2.4. Interfaz de visualización	14
2.5. Electroencefalografía EEG	14
3. Metodología	19
3.1. Estructura Metodológica	19
3.1.1. Primera Fase	19
3.1.2. Segunda Fase	21
3.2. Creación de la escena	22
3.3. Robot móvil diferencial	24
3.3.1. Modelo cinemático del robot diferencial	24
3.4. Dispositivo Novint Falcon	26
3.4.1. Descripción	26
3.4.2. Configuración del dispositivo Novint Falcon	28
3.5. Dispositivo Emotiv Insight Device	29
3.6. Control de asistencia háptica	34
3.7. Control compartido	41

3.8. Control adaptativo	43
4. Resultados	47
4.1. Algoritmo de Control	47
4.1.1. Implementación del sistema de teleoperación virtual	47
4.1.2. Implementación de los datos provenientes de la interfaz EEG	50
4.1.3. Implementación de las estrategias de asistencia háptica	54
4.1.3.1. Estrategia de control compartido, donde el nivel de autoridad es constante durante la ejecución de la tarea	54
4.1.3.2. Estrategia de control compartido, donde el nivel de autoridad es adaptativo durante la ejecución de la tarea.	55
4.1.3.3. Sistema de almacenamiento de variables emotivas para la definición de la ley de control adaptativa	55
4.1.4. Implementación de mapas promedio de neuroseñales según su orientación . .	58
4.1.5. Implementación del control adaptativo	59
5. Conclusiones, logros y trabajos futuros	65
5.1. Conclusiones	65
5.2. Logros	66
5.3. Trabajos futuros	66
Referencias bibliográficas	67

Índice de figuras

2.1. Robot Móvil Tortoise	5
2.2. Robot Móvil Shakey	5
2.3. Robot Móvil Khepera	5
2.4. Clasificación de los robots móviles	6
2.5. Robot Terrestre OFRO	7
2.6. Tipos de configuración Robots Móviles	8
2.7. Pioneer p3dx Mobile Robot	9
2.8. Dispositivo Grope - I	10
2.9. Exoesqueleto tipo mano	10
2.10. Pantalla de fuerza basada en un bolígrafo	11
2.11. Dispositivo HapticSpace	11
2.12. Dispositivo FELEEX	11
2.13. El robot PR2 manipula una esponja de lavado de coches	12
2.14. Dispositivos hápticos comercialmente	13
2.15. Interfaz del proceso de comunicación	13
2.16. Interfaz de usuario Vrep	14
2.17. Sistema de navegación semiautónomo	15
2.18. Localización del electrodo sobre el cuero cabelludo con el sistema de electrodos 10-20.	17
3.1. Escenas Predeterminadas - Vista Superior	19
3.2. Estructura Metodológica - Primera Fase	20
3.3. Estructura Metodológica - Segunda Fase	21
3.4. Escena básica para la teleoperación de un robot móvil	22
3.5. Creación de la escena	23
3.6. Adquisición de datos Vrep	23
3.7. Robot Móvil Pioneer P3DX	24
3.8. Diagrama del modelo cinemático del robot	25
3.9. Dispositivo Novint Falcon	27
3.10. Especificaciones técnicas del dispositivo Novint Falcon	27
3.11. Conexión del dispositivo Novint Falcon	28
3.12. Interfaz Neuroseñales	29
3.13. Dispositivo Emotiv Insight	30
3.14. Interfaces Principales Neuroseñales	30
3.15. Arquitectura de sistema de teleoperación Virtual	34
3.16. Reglas establecidas	44
4.1. Recreación de los datos exportados por Vrep - Scripts Matlab/Octave	48
4.2. Indicación de la orientación del robot durante la ejecución de la trayectoria	48

4.3. Discretización de la trayectoria por secciones o zonas	49
4.4. Orientación del robot de acuerdo a la trayectoria generada por secciones o zonas . .	49
4.5. Datos obtenidos por la trayectoria original y los puntos discretizados	50
4.6. Captura de los niveles emotivos por medio del panel de control de la interfaz cerebro computador	51
4.7. Nivel de compromiso durante la ejecución de la trayectoria	52
4.8. Nivel de estrés durante la ejecución de la trayectoria	52
4.9. Nivel de entusiasmo durante la ejecución de la trayectoria	53
4.10. Nivel de relajación durante la ejecución de la trayectoria	53
4.11. Nivel de interés durante la ejecución de la trayectoria	54
4.12. Nivel de concentración durante la ejecución de la trayectoria	54
4.13. Nivel de compromiso de cada una de las zonas involucradas en la trayectoria	55
4.14. Nivel de estrés de cada una de las zonas involucradas en la trayectoria	56
4.15. Nivel de entusiasmo de cada una de las zonas involucradas en la trayectoria	56
4.16. Nivel de relajación de cada una de las zonas involucradas en la trayectoria	57
4.17. Nivel de interés de cada una de las zonas involucradas en la trayectoria	57
4.18. Nivel de concentración de cada una de las zonas involucradas en la trayectoria . . .	58
4.19. Mapas de los niveles emotivos para cada una de las zonas del entorno de trabajo . .	59
4.20. Trayectoria global realizada por el usuario	60
4.21. Trayectorias realizadas por el usuario	60
4.22. Compendio de los resultados estadísticos, tomando como base los datos a cada usuario	61
4.23. Error de posición del usuario ejecutando la tarea	62
4.24. Tiempo de ejecución del usuario realizando la tarea	63

Resumen

Hoy día, la teleoperación en la robótica cobra cada vez más importancia, esto, debido a la necesidad del ser humano en interactuar físicamente con objetos y entornos remotos, es allí donde se presenta la base de muchas investigaciones para la mejora en este campo. Por tal motivo se llevó a cabo una investigación enfocada al área de la interacción humano - robot, con la cual se pretendía dar solución a la disminución de la precisión de ejecución de procedimientos, debido a comportamientos adquiridos por el humano cuando se llevan a cabo tareas que requieren interacción con un robot, para desarrollar actividades asignadas que en la mayoría de los casos son repetitivas.

Debido a que en la actualidad uno de los retos en esta área es incorporar interfaces multimodales (utilización de los diferentes sentidos), este proyecto de investigación se enfocó en una inclusión y combinación de señales háptica y neuroseñales, con el fin de contar con la información necesaria para establecer algoritmos adaptativos al comportamiento (intención) del humano.

Con el propósito de caracterizar experimentalmente la interacción humano - robot, se desarrolló un algoritmo de control que se adapte al comportamiento de un operario que busca llevar a cabo una asistencia háptica, para efectuar un control de un robot a distancia, de modo que el algoritmo incorpore información previamente adquirida de una interfaz cerebro computador (BCI), para así lograr establecer la acción de control y/o corrección más acertada para la asistencia háptica y así mejorar la realización de la tarea de acuerdo al comportamiento del operario.

Por último se logró realizar un controlador apropiado, el cual le permitió al usuario teleoperar el robot móvil con asistencia háptica y manteniendo un nivel bajo en las diferentes emociones.

Este proyecto se llevará a cabo en las instalaciones del campus universitario de la Universidad de Pamplona, en Pamplona - Norte de Santander, donde también se desarrolló en conjunto con un proyecto de investigación.

Palabras claves: Control adaptativo, Control Compartido robot móvil, háptica, señales EEG, Interfaz Cerebro Computador (BCI).

Abstract

Nowadays, teleoperation in robotics is becoming more and more important, this, due to the need of the human being to physically interact with objects and remote environments, is where the basis of many research for improvement in this field is presented. For this reason, an investigation focused on the area of human interaction - robot was carried out, with which it was intended to provide a solution to the decrease in the precision of the execution of procedures, due to human acquired behaviors when performing tasks that require interaction with a robot, to develop assigned activities that in most cases are repetitive.

Because currently one of the challenges in this area is to incorporate multimodal interfaces (use of the different senses), this research project focused on an inclusion and combination of haptic signals and neurosignals, in order to have the information needed to establish adaptive algorithms for human behaviour (intention).

In order to experimentally characterize human interaction - robot, a control algorithm was developed that adapts to the behavior of an operator seeking to perform haptic assistance, to perform a remote robot control, so that the algorithm incorporates previously acquired information from a brain computer interface (BCI), in order to establish the most appropriate control and/or correction action for haptic assistance and thus improve the performance of the task according to the operator's behaviour.

Finally an appropriate controller was achieved, which allowed the user to remotely operate the mobile robot with haptic assistance and keeping a low level on different emotions.

This project will be carried out in the facilities of the university campus of the University of Pamplona, in Pamplona - Norte de Santander, where it is also developed in conjunction with a research project.

Keywords: Adaptive Control, Shared Control Mobile Robot, Haptic, EEG Signals, Brain Computer Interface (BCI).

Capítulo 1

Introducción

En el presente, la robótica ha crecido de manera exponencial debido a los requerimientos que se han necesitado a lo largo de la historia, por tanto es conveniente mencionar que uno de los objetivos principales de esta área es ayudar al ser humano en realizar diversas labores domesticas y no domesticas, por lo que se ha creado diversas tecnologías que ayudan con este propósito, como lo es visión artificial, olfato, audio, reconocimiento de voz, tacto, entre otros.

Este último ha irrumpido fuerza en los últimos años, debido a que muchas de las tareas realizadas por los seres humanos requieren de su presencia, exponiendo así su vida, por lo que son útiles los sistemas teleoperados. Uno de los inconvenientes al usar este sistema es que el operador no percibe su entorno, causando así problemas a la hora de ejecutar la tarea. Por tal motivo la háptica es de gran ayuda para la teleoperación.

El sentido del tacto, el cual incluye el sentido cinestésico, permite al operador percibir la configuración de su cuerpo y su entorno. La ciencia que estudia este sentido se denomina háptica.

La interacción humano - robot por medio de señales hápticas sea natural no es fácil, ya que depende de la intensión del operador y el entorno en el que se realiza la tarea, estos elementos tienen dinámicas no - lineales y en algunos casos de parámetros con incertidumbres, lo cual hace el control complejo.

Esta investigación tuvo como objetivo generar un algoritmo de control con asistencia háptica para el guiado de un robot, esto con el propósito de evaluar diferentes emociones durante la ejecución de una tarea de control de un robot móvil, el cual realimenta señales hápticas de la tarea al operador mediante un dispositivo háptico Novint Falcon [47].

Cabe mencionar que el dispositivo para evaluar las diferentes emociones es el Emotiv Insight [12].

Este dispositivo es de bajo costo, nos permite registrar las actividades del cerebro por medio de las señales electroencefalograma EEG para la detección de diversos estados mentales tales como, compromiso, interés, estrés, relajación, concentración y entusiasmo.

Teniendo en cuenta que este dispositivo está disponible para la experimentación y que es un dispositivo comercial de uso frecuente en la academia, las métricas a analizar sobre las emociones se realizarán directamente por medio de la aplicación disponible con este dispositivo.

Para que el usuario logre percibir que el robot es un compañero que coopera en la realización de tareas y no una simple herramienta, el robot es equipado con mecanismos de adaptación, como lo son sensores entre otros.

1.1. Planteamiento del problema y justificación

En los procesos de automatización se pueden presentar sistemas variantes, o que son requeridos con una determinada flexibilidad para poder llevar a cabo diferentes procesos con los mismos dispositivos, en pro de un mayor aprovechamiento de las máquinas, dispositivos, entre otros, sin elevar los costos de operación; contar con procesos variantes o en algunos casos procesos con alta complejidad, limita la implementación de una estrategia de automatización completa, lo cual exige la inclusión de un operario y un mando establecido en secciones del proceso a automatizar.

Ya que una de las soluciones más actuales a la problemática de contar con procesos complejos que exigen la inclusión de operarios para la manipulación de algunos elementos, es la implementación de sistemas teleoperados, se debe tener en cuenta que estos, en algunos casos, cuando no son muy robustos, conllevan a un estrés por parte del operario o bien si estos basan su funcionamiento en la asistencia háptica, pueden ser tan confiables que el operario podría delegar en su totalidad la funcionalidad al sistema y esto podría producir una falta de concentración en el proceso que como consecuencia podría aumentar el margen de error en el desarrollo de la actividad establecida.

Existen técnicas inteligentes de control que aplicadas a la asistencia háptica logran realizar una corrección en el desarrollo de las tareas para las cuales está diseñado el robot, sin embargo, en esta corrección no para todos los casos se tiene en cuenta que la causante de desviación de la óptima tele - manipulación por parte del operario se puede deber a variaciones de carácter cognitiva que repercuten en diferente medida dependiendo del operario, por lo tanto algunos sistemas desarrollados realizan una correcta validación cuando implementa el sistema háptico con el mismo usuario, pero puede presentar un margen de error lógico cuando se pretende la implementación con un usuario diferente.

Conociendo lo anterior, surge la pregunta ¿Es posible diseñar un algoritmo de control para el guiado de un robot teniendo en cuenta diversas emociones?

1.2. Objetivo general

Generar un algoritmo de control de asistencia háptica para el guiado de un robot a partir de la información previamente obtenida por medio de una interfaz cerebro - computador.

1.2.1. Objetivos específicos

- * Establecer un sistema que permita teleoperar un robot virtual o físico por medio de un dispositivo háptico
- * Registrar las neuroseñales obtenidas y procesadas mediante la interfaz cerebro computador de un usuario mientras realiza una tarea de teleoperación preestablecida

- * Diseñar el algoritmo de control de las asistencias hápticas para que se adapte, según el comportamiento previo de un operario al teleoperar un robot
- * Validar el algoritmo de asistencia háptica del robot

1.3. Estructura del documento

Con el fin de mejorar una comprensión de la investigación que se realizó, este proyecto estará estructurado de la siguiente manera:

En el capítulo 1, se presentan los elementos introductorios con el fin de dar a conocer lo que se verá reflejado en el desarrollo del proyecto, para esto se presenta el planteamiento de problema y los objetivos que posteriormente se desarrollarán.

En el capítulo 2, se describe el marco teórico y el estado del arte acerca de los robots móviles, dispositivos hápticos y neuroseñales, la finalidad de esto, es conocer las investigaciones actuales que se han desarrollado en estas áreas, para así tener una visión mas clara durante la ejecución del proyecto.

En el capítulo 3, se presenta la estructura metodológica que se llevó acabo en el proyecto de investigación, donde se presenta el paso a paso que se siguió para una satisfactoria ejecución de esta de investigación. En los pasos a seguir se encuentra el funcionamiento del esquema de control propuesto para realizar la teleoperación, de igual manera el algoritmo de control y uso del dispositivo háptico Novint Falcon como también el dispositivo captador de neuroseñales Emotiv Insight, los cuales fueron usados para la realización de los experimentos.

Los resultados obtenidos de la teleoperación del robot móvil diferencial mediante el dispositivo háptico y las neuroseñales tomadas, son presentadas en el capítulo 4.

Por último, se mencionan las conclusiones, logros y trabajos a futuros por hacer en esta línea de investigación.

Capítulo 2

Marco teórico y estado de arte

2.1. Robótica móvil

2.1.1. Generalidades

Se le conoce como robot móvil a una máquina capaz de transportarse en cualquier tipo de terreno de forma autónoma [6], la creación de esta rama en la robótica surge con la necesidad de expandir el campo de aplicación e incrementar la autonomía de los robots.

En sus inicios los manipuladores estaban restringidos al alcance de una estructura mecánica anclada en uno de sus extremos, de igual manera surgía la necesidad de explorar áreas donde el ser humano no podía tener acceso, de modo que estos a lo largo del tiempo han sido utilizados para la exploración, la investigación y aplicación de nuevas tecnologías, realizar tareas militares como espionaje y detección de objetos, exploración espacial, desarrollo en el ámbito industrial, entre otras.

En la actualidad se siguen trabajando los temas mencionados como es el caso del estudio que se basa en el análisis del tiempo o retardo que existe en la comunicación de información como señales o video entre el robot y los operadores se fijan en aspectos como la distancia que existe entre el robot al centro de control, el ancho de banda de la comunicación del robot para disminuir el efecto de retardo de la señal y mejorar el rendimiento del operador para el proceso en diferentes etapas.

2.1.1.1. Histórico de los robots móviles

Unos de los primeros robots móviles creados fueron los llamados Tortoises como se muestra en la figura 2.1, quien se le atribuye el nombre por su caparazón de tortuga en forma de espalda. Estos fueron creados por el neurólogo o experto en robótica y neurofísica **Grey Walter's**, en la cual su intención era demostrar la interconexión entre una pequeña cantidad de células cerebrales la cual tiene el potencial de crear un comportamiento complejo.

Su funcionamiento se basaba en posicionar un conjunto de válvulas haciendo equivalencia a neuronas y un sensor de luz a un ojo, y estos eran ubicados sobre una rueda en la parte delantera, de tal manera que si el robot no recibía luz se utilizaban el conjunto de válvulas para hacer girar el robot [6].



Figura 2.1: Robot M6vil Tortoise
Fuente: [31]

Por otra parte, fue creado el robot m6vil Shakey como se muestra en la figura 2.2, el cual fue el primer robot m6vil en razonar al tomar decisiones en su entorno, esto lo pod6a hacer debido a que se encontraba equipado con una c6mara de televisi6n, un buscador triangular y sensores de parachoques, su uso fue para programas de percepci6n, modelado del mundo y actuaci6n [19].



Figura 2.2: Robot M6vil Shakey
Fuente: [36]

Sucesivamente en 1994 fue creado el robot m6vil Khepera (ver figura 2.3), desarrollado por laboratorio LAMI encabezado por el profesor **Jean-Daniel Nicoud**, este robot fue usado para actividades de investigaciones y para demostraciones en varias conferencias y talleres.

Este veh6culo ten6a un microcontrolador Motorola **MC68331**, siendo este ideal por su peque6o tama6o y por ser de alta frecuencia, tambi6n pose6a dos motores DC con sensores incrementales magn6ticos y sensores de proximidad infrarrojos [26].



Figura 2.3: Robot M6vil Khepera
Fuente: [50]

En la actualidad se han creado muchos robots móviles para distintas aplicaciones, uno de los más reciente es un método novedoso para estimar la posición de un robot móvil en entornos bajo el piso utilizando nubes de puntos RGB-D, desarrollado por la universidad de Elche, España y la escuela de ingeniería de diseño, Imperial College London [32].

Este robot está equipado con un scanner láser y una cámara RGB, donde los datos capturados por ambos sensores se combinan para así crear una nube de puntos que describen la apariencia del entorno.

2.2. Tipos de robots móviles

Por lo general se tiende a asociar los robots móviles como vehículos que posean llantas, pero la realidad es que existen muchos más como se muestran en la figura 2.4.

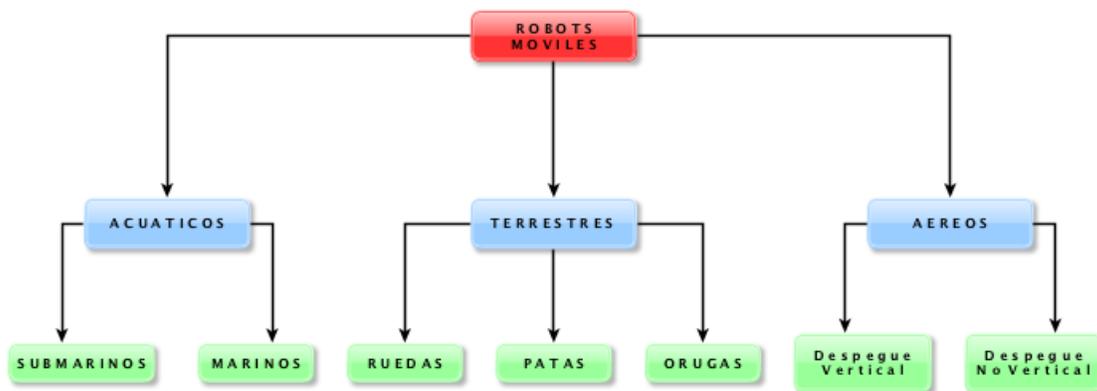


Figura 2.4: Clasificación de los robots móviles

Fuente: Adaptación [32]

2.2.1. Robots Terrestres

2.2.1.1. Robots con orugas

Al manipular un robot en terreno abierto se hace cada vez más difícil su control y estabilidad, por ello era necesario desarrollar vehículos capaces de desenvolverse en cualquier tipo de terreno, para esto se desarrolló la locomoción sobre una banda continua de huellas o placas de oruga impulsadas por dos o más ruedas que se tradujo directamente mayor contacto con el terreno, por lo tanto, más tracción que las ruedas convencionales.



Figura 2.5: Robot Terrestre OFRO
Fuente: [43]

El término de locomoción basado en orugas aparece por primera vez en 1770, donde **Richard L. Edgeworth** publica una patente, estos vehículos en su comienzo no tuvo tanto éxito, esto por que su consumo energético es bastante alto en comparación con las ruedas.

Fue hasta principios del siglo XX, con la aparición de la combustión interna se produjo un salto considerable en los sistemas de locomoción mediante orugas [14].

En la actualidad es muy común encontrar este tipo de robots para tareas de neutralización de explosivos, tal es el caso del robot VALLI, compuesto por una plataforma móvil y un brazo manipulador (robot híbrido), también posee un sistema de orugas de motores independientes, para desplazarse en terrenos no estructurados permitiéndole dar giros sobre su propio eje, subir escaleras y arrastrar cargas pesadas [39].

2.2.1.2. Robots con ruedas

Existen múltiples configuraciones, como se muestra en la figura 2.6:

- ❖ Tracción diferencial
- ❖ Ackerman
- ❖ Tracción omnidireccional
- ❖ skid steer
- ❖ Triciclo clásico
- ❖ Tracción síncrona

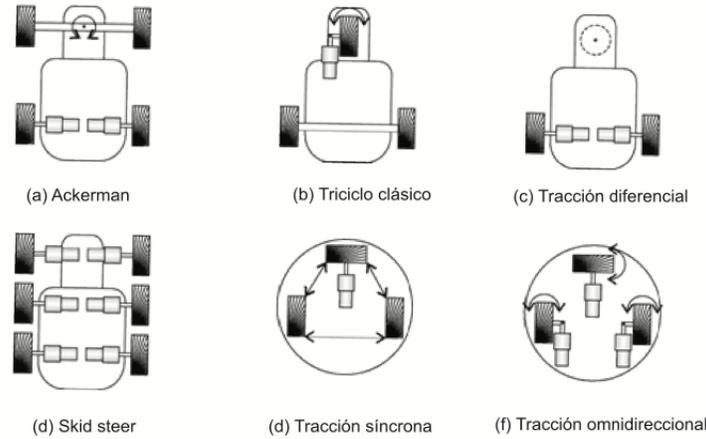


Figura 2.6: Tipos de configuración Robots Móviles
Fuente: [29]

Los robots móviles con ruedas se caracterizan porque no tienen partes flexibles y se les puede aplicar mecanismos de cuerpo rígido, también se distingue por el tipo de ruedas que utilizan: ruedas convencionales, tipo castor, tipo bola y omnidireccionales.

En la configuración Skid Steer se hace imprescindible un sistema de suspensión y capacidad de direccionamiento, y ante la desventaja que tenían desde sus inicios este tipo de robots en terrenos abiertos, los investigadores analizaron el descenso frontal y lateral y ascenso sobre escalones además del desplazamiento sobre zanjas.

Los estudios consistían en variar parámetros de suspensión, como lo son constantes de rigidez en los resortes, constante de amortiguamiento en los amortiguadores y constante de amortiguamiento y rigidez en las ruedas.

Se determinó que para mejorar la capacidad de estabilidad y direccionamiento del robot se debe poseer resortes y ruedas de alta rigidez, amortiguadores con alto coeficiente de amortiguamiento y ruedas con bajo coeficiente de amortiguamiento [13].

2.2.1.3. Robots con tracción diferencial

Una de las características principales de este tipo de robots, es que solo tiene dos ruedas motrices y un rodillo o bola de equilibrio, es decir, no hay ruedas directrices por lo que el cambio de dirección se realiza modificando la velocidad relativa de las ruedas, por lo que, mantener al robot en dirección recta es una tarea compleja, ya que las ruedas tienen que girar a la misma velocidad.

Los robots con tracción diferencial tienen un mecanismo de accionamiento más simple que cualquier otro tipo de configuración, por lo que es el más utilizado en terrenos pavimentados.

Hoy en día en la mayor parte de los casos se requiere que los robots sean autónomos, por lo que es una de las áreas con mayor número de investigaciones.

Los estudios van desde la implementación de controladores PID en conjunto con redes neuronales para conocer el modelo cinemático del robot [39].

Otras investigaciones utilizan el tiempo de vuelo de la señal del RF, para hacer una triangulación proporcional con el fin de estimar el entorno bidimensional y así estimar de manera más exacta la posición y orientación del robot [10].

Por otra parte, también se investiga la tracción y los pares de las ruedas en terrenos rígidos o deformables en este tipo de configuración, esto mediante un filtro extendido de Kalman-Bucy, el cual determina los historiales de tiempo de tracción neta y pares de ruedas resistivos y deslizamiento de la rueda durante maniobras constantes o transitorias [40, 8].

Los investigadores, suelen realizar estudios en robots a pequeña escala, con el fin de ahorrar costos de construcción fácil transporte entre otras.

Uno de los robots móviles más utilizado es el **Pioneer P3DX**, este robot cuenta con dos ruedas, con motores diferenciales y una serie de sensores que se pueden adaptar (ver figura 2.7) siendo así ideal para investigaciones y uno de lo más utilizados en los centros de investigaciones, dado a que es versátil, confiable y duradero.



Figura 2.7: Pioneer p3dx Mobile Robot
Fuente: [42]

Esta plataforma móvil ha sido de gran utilidad en diversos estudios, una de ellas es el filtrado integrado y eficiente de información sensorial de modalidad múltiple, por medio del cual se obtienen los datos sensoriales más relevantes en la situación actual [27].

Por otra parte, este robot es utilizado para el seguimiento de la visión, mapeo del entorno, planificación de rutas y navegación dentro de edificios de servicios.

Dada sus características a este robot se le puede agregar diferentes tipos de sensores que ayudan a una mejor investigación, de igual manera se le puede añadir manipuladores para que su uso sea más extenso [2, 11].

2.3. Interfaces hápticas

La háptica es la ciencia que se encarga del estudio del tacto o sensaciones de movimiento a través de interfaces tecnológicas.

Uno de los primeros dispositivos hápticos fue el Dextrous Hand Master Exoskeleton, desarrollado en el MIT (1990), cuyo objetivo era realizar una simulación de realidad virtual [33].

Para la misma época fue creado el dispositivo GROPE - I como se muestra en la figura 2.8, este estaba equipado con dos potenciómetros que simulaban las posiciones X e Y, luego surgieron nuevas versiones de este como lo son el GROPE - II y el GROPE - III [5, 1].

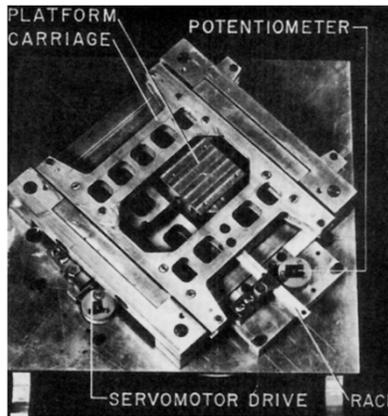


Figura 2.8: Dispositivo Grope - I
Fuente: [5]

Subsecuentemente, fue creado un exoesqueleto tipo mano (ver figura 2.9), el cual contenía 6 grados de libertad (6 DOF). Esto permitía una retroalimentación de fuerza por medio de realidad virtual, de los 6DOF que poseía, tres eran para fuerza dimensional y los otros tres eran torques dimensionales [28].

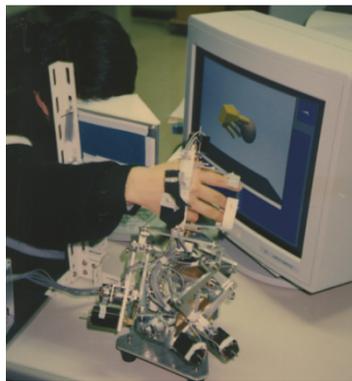


Figura 2.9: Exoesqueleto tipo mano
Fuente: [28]

Siguiendo la cronología, en el 1993 crean una pantalla de fuerza basada en un bolígrafo (ver figura 2.10), este tipo de interfaz háptica tipo manejo de herramientas fue una implementación de visualización de fuerza sin dispositivo tipo guante [15].

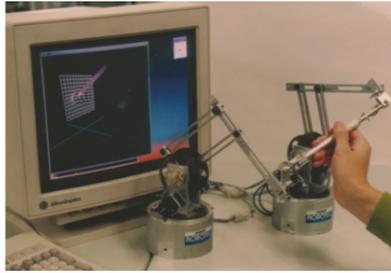


Figura 2.10: Pantalla de fuerza basada en un bolígrafo
Fuente: [15]

En 1994 fue desarrollado el hapticSpace (ver figura 2.11) el cual consiste en un accesorio de aproximación de forma montado en un manipulador [53].



Figura 2.11: Dispositivo HapticSpace
Fuente: [53]

En 1997 fue desarrollado el primer dispositivo FEELEX, este permitía la interacción con las dos manos usando las palmas [16].



Figura 2.12: Dispositivo FELEEX
Fuente: [16]

En los últimos años la háptica ha continuado siendo uno de los temas de estudio más trabajados en el área de la robótica, ya que permite el trabajo en conjunto entre algoritmos de control y la participación humana.

Actualmente se trabaja en aplicaciones de aprendizaje robótico por medio de la interacción humana, permitiendo aprender patrones y comportamientos al robot, necesarios en un entorno de

trabajo predeterminado.

Para que un robot pueda desarrollar tareas complejas y útiles debe reconocer objetos en su entorno.

Tal es el caso del robot Willow Garage PR2 (ver figura 2.13), que incorpora un complejo sistema de sensado táctil SynTouch Bio Tacp para la exploración de objetos, permitiendo conocer su textura, temperatura y fricción, a su vez se desarrollan algoritmos de aprendizaje con la finalidad de descubrir información acerca de los objetos que éste manipula a partir de los sensores instalados, durante la exploración de los objetos, los algoritmos lograron reconocer el objeto tanto entornos dinámicos y estáticos [7].

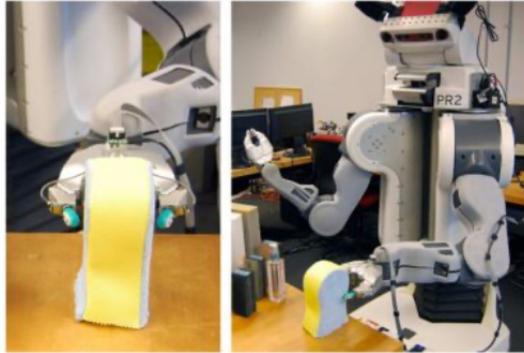


Figura 2.13: El robot PR2 manipula una esponja de lavado de coches
Fuente: [7]

En tareas que involucran tecnología háptica, hay una gran variedad de campos de aplicación, como es el caso de la teleoperación de un robot con retroalimentación háptica para el mapeo del movimiento en la exploración espacial [21].

Mediante un método de mapeo híbrido maestro - esclavo con un modelo de retroalimentación háptica que representa la resistencia operativa y la retroalimentación interactiva del robot esclavo, los dos dispositivos que poseen mecanismos 3R y delta permiten manipular la dirección y posición del robot usando ambas manos.

El mapeo del control de movimiento híbrido combina el control de velocidad y mapeo de posición para lograr el control maestro - esclavo adecuado.

Las técnicas de mapeo propuestas tienen la finalidad de tener una buena ergonomía del sistema y una mejor inmersión en la aplicación de dispositivos hápticos.

En la figura 2.14 se presentan algunos dispositivos hápticos comercialmente disponibles.

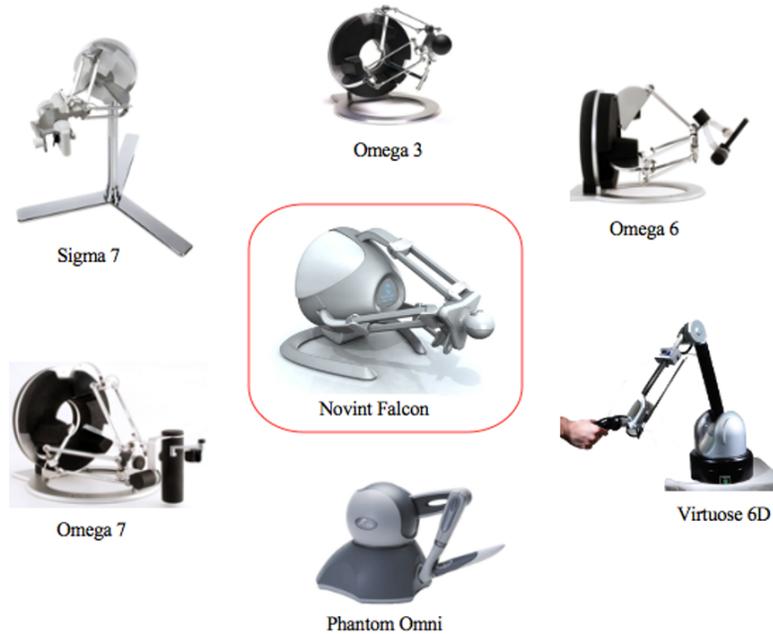


Figura 2.14: Dispositivos hápticos comercialmente
Fuente: Adaptación [37]

Para representar el dispositivo háptico en un entorno virtual, se hace uso de varias herramientas, una de ellas es chai 3D, es un framework de código abierto que admite una variedad de dispositivos hápticos de tres, seis y siete grados de libertad disponibles comercialmente, y hace sencillo admitir nuevos dispositivos personalizados de retroalimentación de fuerza [49].

Otra herramienta es SDK, proveniente de la compañía Force dimensión. Esta permite a los usuarios agregar fácilmente capacidad háptica a su aplicación al ocultar toda la complejidad de la programación de dispositivos hápticos.

El proceso de comunicación entre el dispositivo y el entorno virtual consiste en una interfaz (ver figura 2.15) escrita en C o C++, luego se precisan tareas de control avanzado, haptic SDK y robotics SDK ofrecen funciones de bajo nivel que permiten acceder y controlar todos los aspectos de los dispositivos hápticos Force Dimension.

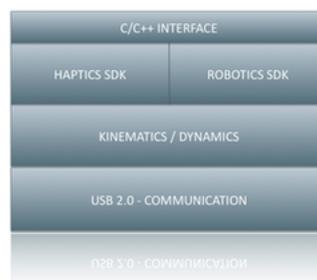


Figura 2.15: Interfaz del proceso de comunicación
Fuente: [52]

Dependiendo del dispositivo, se tienen diferentes tipos de cinemática y dinámica, por lo que la mayoría de dispositivos tiene ajustes preestablecido y la comunicación se hace a través de USB 2.0 [52].

2.4. Interfaz de visualización

Existe una gran variedad de interfaz la cual permiten la visualización de robots, una de ella es el software de Vrep como se muestra en la figura 2.16 que fue creada por CoppeliaSim, es un entorno de desarrollo integrado, la cual se basa en una arquitectura de control distribuido: cada objeto/modelo se puede controlar individualmente a través de un script incrustado [41].

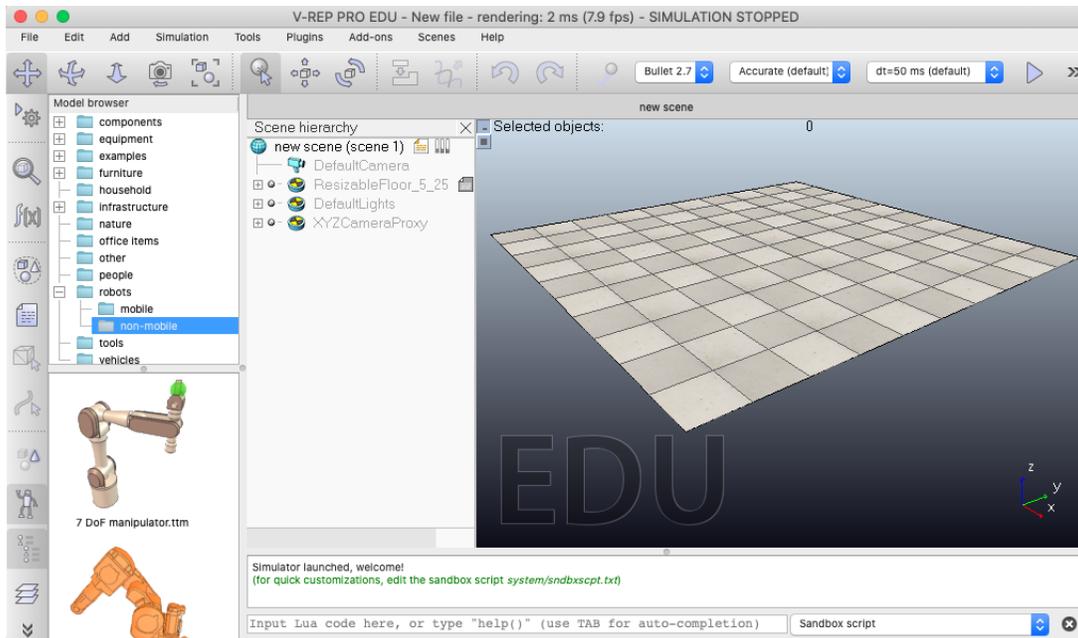


Figura 2.16: Interfaz de usuario Vrep
Fuente: Autor

La plataforma educativa de Vrep nos permite la conexión con múltiples plataformas, entre las cuales están matlab y ROS (Robot Operating System) donde admite programar en múltiples lenguajes de programación como lo son: C/C ++, Python, Java, Lua, Matlab o Octave.

2.5. Electroencefalografía EEG

Son señales que permiten conocer la actividad eléctrica del cerebro, mediante el registro de la actividad bioeléctrica que se da en la corteza cerebral.

Esta técnica se empezó a implementar en conjunto con la robótica para poder hacer sistemas sofisticados de control de robots (móviles, brazos), que permita enviar parámetros a los robots de una forma más fácil y accesible para las personas, como también para personas con afecciones físicas para realizar control del robot mediante las señales dadas por el cerebro [34].

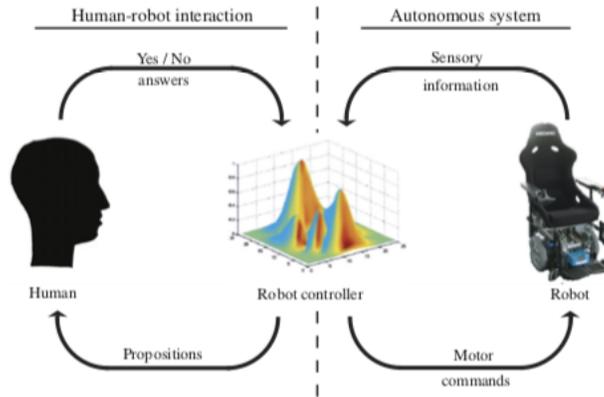


Figura 2.17: Sistema de navegación semiautónomo

Fuente: [34]

En la literatura se encontró que la base para la comunicación entre un robot y un humano usando las señales electroencefalograma EEG son las interfaces BCI (Brain Computer Interfaces), estos son sistemas que se basan en adquirir señales cerebrales en tiempo real para ser procesadas en un ordenador y así hacer uso de ellas en la aplicación que se requiera, sin embargo, se resalta la necesidad de mejorar la adquisición de las señales bioeléctricas como también el rendimiento y procesamiento de dichas señales [23].

Siguiendo este orden de ideas, se han realizado experimentos de las interfaces BCI en conjunto con otros implementos como lo pueden ser auriculares o proyectores, esto con la finalidad de realizar el estudio de las señales electroencefalograma EEG recibidas para su procesamiento y análisis, esto para poder brindar órdenes a los participantes y luego comparar los resultados obtenidos mediante herramientas como lo puede ser el software de Matlab para el control de una mano robótica [17].

Ya entrando a la parte de robótica o robots que trabajan en conjunto con las interfaces BCI o con las señales electroencefalograma EEG, se han realizado aplicaciones con robots de servicio [18], en donde se busca incluir a personas discapacitadas o personas que no están familiarizadas con sistemas robóticos, los cuales puedan controlar un robot mediante una interfaz BCI de forma rápida y confiable.

De esta forma también se introduce el concepto de BCRI (Interfaz de Robot Computador Cerebral) el cual indica la capacidad de la persona de generar el código BCI para atributos cinemáticos y dinámicos aplicados a un robot dinámico [44].

Las aplicaciones de robots con interfaz cerebro máquina con el uso de las señales electroencefalograma EEG buscan conectar tecnologías a los procesos neuronales, e ir realizando cada vez tareas más complejas.

Se plantean trabajos más sencillos que funcionen como base, por ejemplo, se tiene el control de un robot móvil a partir de las señales electroencefalograma EEG a través de procesos simples usando la sincronización y la sincronización de ondas alfa del evento [22].

También se han realizado trabajos relacionado al mapeo usando una interfaz EEG para diversas aplicaciones como lo puede ser el uso de un robot plano asistencial para la selección y desplazamiento de objetos en dos dimensiones para personas con discapacidad mediante una interfaz

máquina cerebral espontánea (BMI o IMC) en relación a los mapeos de un EEG [48].

El desarrollo de aplicaciones de interfaces BCI para el control de robots es una de las aplicaciones principales de los métodos y/o modelos basados en señales electroencefalograma EEG como el trabajo en el cual se busca realizar el movimiento del robot mediante los pensamientos, los cuales pueden ser el movimiento de un brazo izquierdo o derecho o en su defecto el movimiento de un pie del usuario en los cuales estos deberán ser capaces de mover el brazo robótico para una aplicación de asistencia para personas discapacitadas.

Otros estudios se basan en la adquisición de las señales de la actividad neuronal por medio de los auriculares Emotiv EPOC y la extracción de las características obtenidas por medio de las señales mediante el método de análisis de los componentes principales (PCA), con el método de transformada rápida de Fourier (FFT) dentro del rango de la banda de frecuencia de los ritmos (8 Hz – 22 Hz) de los sensores [4].

Los trabajos en simulación de modelos de sistemas mediante las interfaces cerebro computadora (BCI) con retroalimentación háptica, utilizan la interfaz BCI para entrenar un motor con retroalimentación visual y háptica mediante el manejo de un lápiz óptico y un dispositivo háptico.

Los efectos de las pruebas en el motor con y sin retroalimentación háptica quedan guardados en los análisis de los cambios en el EEG [51].

Otros estudios se basan en la rehabilitación a partir de sistemas de interactivos neuronales, utilizando métodos de EEG y empleando dispositivos de retroalimentación háptica. Dando pie a lo que se conoce ahora como Neuro Rehabilitación Mecatrónica (NRM) enfocada en que los paciente sean capaces de recuperar sus funciones motoras causadas por daños neuronales y/o físicos, combinando también los sistemas de neuro - rehabilitación robótica interactiva (HIRN) a partir de dispositivos hápticos.

Los sistemas HIRN aceleran la plasticidad y la recuperación del cerebro a largo del tiempo y aumentan la capacidad de la persona lesionada para realizar actividades de la vida cotidiana [3].

También se emplea la señal biológica del EEG para realizar la evaluación de los conocimientos el manejo de un dispositivo háptico de forma que se muestre si los estudiantes aprendieron el manejo del dispositivo en comparación con el manejo del dispositivo de una persona experta en el ámbito, siendo el aprendizaje de forma visual y práctica [24].

Por otra parte, se tiene el seguimiento de un motor con retroalimentación háptica con el uso del electroencefalograma, donde se estudia la dinámica temporal del cerebro que se da como respuesta a partir de una retroalimentación háptica además, se emplea el análisis de componentes independientes para la descomposición de las señales electroencefalograma EEG.

Los resultados obtenidos son analizados mediante el análisis de frecuencia de tiempo y la coherencia relacionada con eventos (ERCOH), para poder hacer un contraste entre la actividad cerebral con o sin la retroalimentación háptica [20].

Existe otro tipo de señales electroencefalograma EEG de alta calidad, esto se logra con un método de automatizado que descarta las señales de baja calidad o con altos índices de ruido de modo que

solo se tienen en cuenta las señales de alta calidad para el análisis correspondiente, en la mayoría de las aplicaciones de EEG se busca ver la efectividad de las características de las señales dadas por las interfaces BCI, sin embargo, para que estas se puedan obtener de una forma correcta es necesario que las señales posean una buena calidad de medición y descartar así el ruido generado en la medición.

Para realizar una diferenciación entre las señales adquiridas la medición se realiza de forma manual para que los resultados de los trabajos estén lo más correctos posibles y estén lo más cerca posible a la realidad, por eso se propone automatizar esta selección de señales mediante un algoritmo en donde solo se tienen en cuenta las señales de buena o alta calidad para los análisis y estudios [25].

También se viene desarrollando investigaciones con algoritmos capaces de tomar las características necesarias de las señales electroencefalograma EEG para lograr realizar un reconocimiento de emociones, estas nuevas aplicaciones tienen la particularidad de que utilizan técnicas de computación evolutiva que permite aprender patrones y desenvolverse mejor para la tarea asignada.

Gracias a todas las posibles funciones de las señales electroencefalograma EEG se puede llegar a una alta dimensionalidad en las características y una reducción del rendimiento por la redundancia en un sistemas, por esta razón se buscan soluciones para disminuir esta complejidad de alta direccionalidad por medio de métodos que permitan de forma automática buscar el subconjunto óptimo de las características EEG usando algoritmos computacionales evolutivos (EC).

En otras investigaciones se realiza una evaluación por medio de un conjunto de datos públicos (MAHNOB, DEAP), junto a un nuevo conjunto de datos tomados a partir de uno sensores EEG como se muestran en la figura 2.18, en donde las pruebas muestran que el algoritmo propuesto identifica las mejores características de las señales electroencefalograma EEG y mejora el rendimiento para un problema de clasificación de cuatro cuadrantes [30].

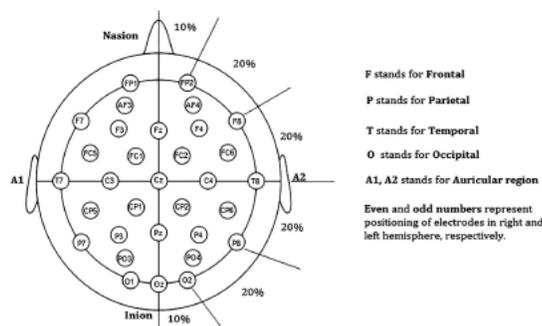


Figura 2.18: Localización del electrodo sobre el cuero cabelludo con el sistema de electrodos 10-20.

Fuente: [30]

También, se adelantan investigaciones en la identificación de las señales electroencefalograma EEG como por ejemplo: las señales alfas que se producen en lectura del cerebro para poder hacer su caracterización y análisis por medio de un dispositivo portátil, un paso importante es poder describir con precisión las señales EEG para cuantificar los ritmos cerebrales, para esto se exploraron dos métodos de técnicas de análisis de datos para las señales electroencefalograma EEG Matching Pursuit (MP) y Fast Fourier Transform (FFT), y realizar una comparación entre las dos etapas del experimento: con los ojos abiertos (EO) y con los ojos cerrados (EC), en ambos casos se utiliza un

dispositivo portátil EEG para la detección de las señales alfa, se tiene que el método MP alcanza un alto porcentaje estadístico en todas las ubicaciones del estudio [45].

cuarenta y ocho (48) para una tarea media, para un total de doce usuarios (12) que realizaron la prueba.

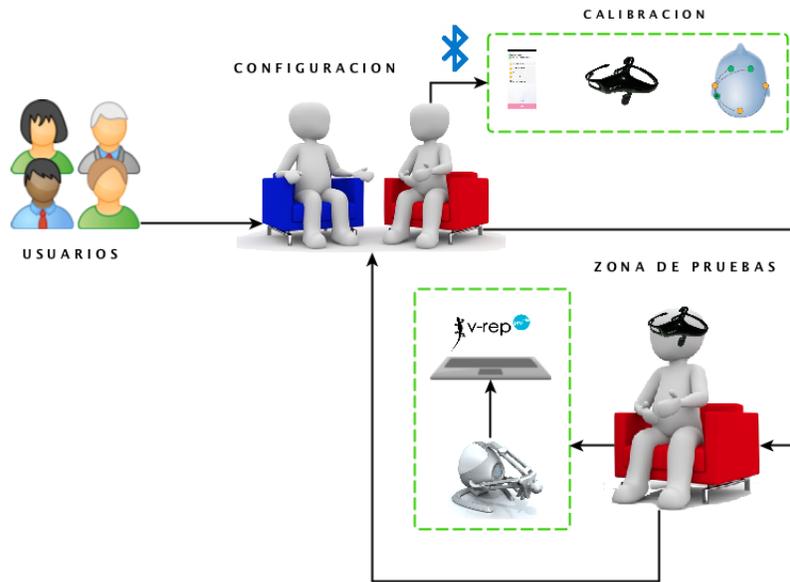


Figura 3.2: Estructura Metodológica - Primera Fase
Fuente: Autor

Se seleccionaron personas mayores entre edades no superiores a cuarenta (40) años, de más o menos del mismo nivel profesional.

El experimento consiste en que el usuario realizará por cada nivel medio y alto, cuatro (4) pruebas diferentes en las cuales iba presentando un grado de complejidad, relacionándola de la siguiente manera:

- ❖ La primera prueba **SAP – SAO**: Sin Asistencia Path (trayectoria) – Sin Asistencia Obstáculos
- ❖ La segunda prueba **SAP – CAO**: Sin Asistencia Path (trayectoria) – Con Asistencia Obstáculos
- ❖ La tercera prueba **CAP – SAO**: Con Asistencia Path (trayectoria) – Sin Asistencia Obstáculos
- ❖ La cuarta prueba **CAP – CAO**: Con Asistencia Path (trayectoria) – Con Asistencia Obstáculos

Para cada uno de los experimentos, se realizaron los siguientes pasos:

- ❖ Se realizó una breve explicación de cual era el objetivo de la prueba en cuanto al procedimiento, la configuración y el manejo de los diferentes dispositivos (Emotiv – háptico) todo lo mencionado con su consentimiento de cada usuario
- ❖ Después, se comenzaba a humedecer el cabello por medio de una solución salina, con el fin de mejorar la conductividad eléctrica de las señales

- ❖ Luego se realiza la sincronización y calibración del dispositivo Emotiv insight utilizando una conexión bluetooth a través de un ordenador o Smartphone, el cual consiste en una serie de preguntas con el fin de analizar el estado anímico del usuario, posteriormente se verifica la conexión de los sensores
- ❖ Por último se realiza la captura e interpretación de los datos obtenidos por cada usuario a través de los dispositivos para su respectivo análisis

Básicamente lo que se busca es corroborar, que tanto cambia con cada uno de las asistencias planteadas, cual es la variación de cada unas de las emociones con respecto a los diferentes eventos, entendemos que los niveles de todas las emociones van variando de usuario en usuario.

3.1.2. Segunda Fase

En la segunda fase se realizaron treinta y seis (36) experimentos a doce (12) usuarios, cada prueba fue realizada en un escenario de alta complejidad como se muestra en la figura 3.1(b), donde se realizaban tres (3) tipos de pruebas, relacionándola de la siguiente manera:

- ❖ **Sin asistencia:** primera prueba
- ❖ **Con asistencia:** segunda prueba
- ❖ **Asistencia adaptativa:** tercera prueba

A continuación, se muestra la estructura metodológica empleada en la segunda fase correspondiente a la figura 3.3.

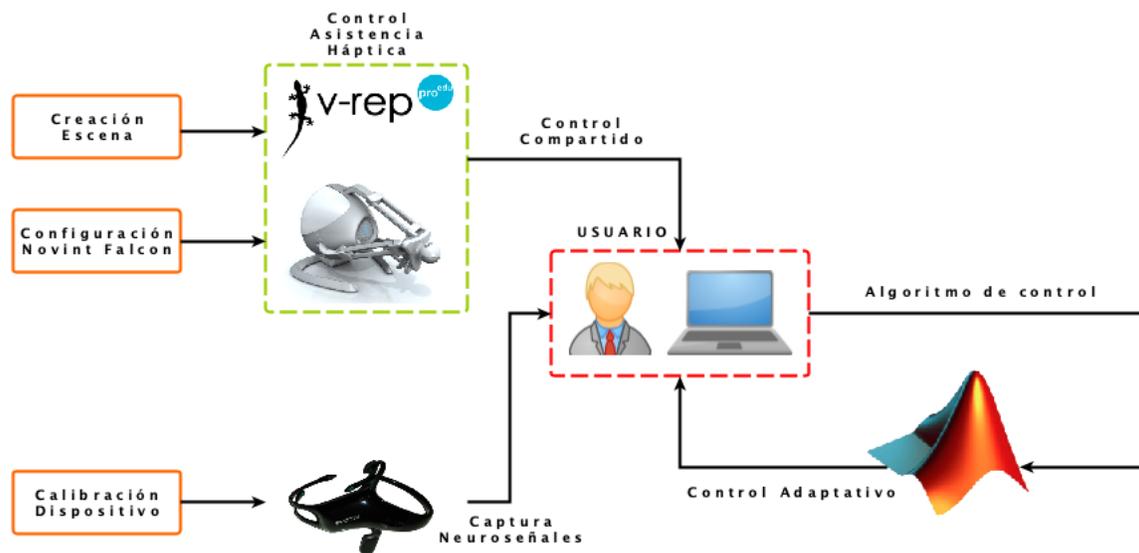


Figura 3.3: Estructura Metodológica - Segunda Fase
Fuente: Autor

Se optó por una serie de pasos como se muestra a continuación:

- ❖ Como primera medida, se desarrolló el control de asistencia háptica, para esto se creó un entorno (escena) en la plataforma de Vrep, donde el usuario realizaría una serie de prueba, de igual manera se configuró el dispositivo Novin Falcon, de tal manera que el operador pueda controlar el robot en la escena antes creada
- ❖ Posteriormente se calibra el dispositivo Emotiv insight, con el fin de hacer mediciones de las diferentes emociones
- ❖ La calibración consiste en posicionarle el casco de manera adecuada al usuario, de tal forma que los sensores queden bien posicionados, seguido, se le indica al usuario que mire hacia al frente un punto fijo durante unos minutos y después que cierre los ojos
- ❖ Luego de tener los dispositivos calibrados y creado la escena se procede a realizar una serie de pruebas, en el cual se captura las neuroseñales mientras el usuario opera el robot mediante un control compartido
- ❖ Estas pruebas son guardadas y analizadas en la plataforma de Matlab, para posteriormente realizar un controlador adaptativo

3.2. Creación de la escena

Para la realización de los primeros experimentos, se creó una escena sencilla, en el cual un usuario teleoperó un robot móvil diferencial desde un punto de salida hacia un punto de llegada evadiendo los obstáculos como se muestra en la figura 3.4.

El fin de esta prueba es corroborar la retroalimentación de fuerza proporcionada por los sensores del robot móvil mediante el dispositivo háptico, de igual manera su teleoperación.

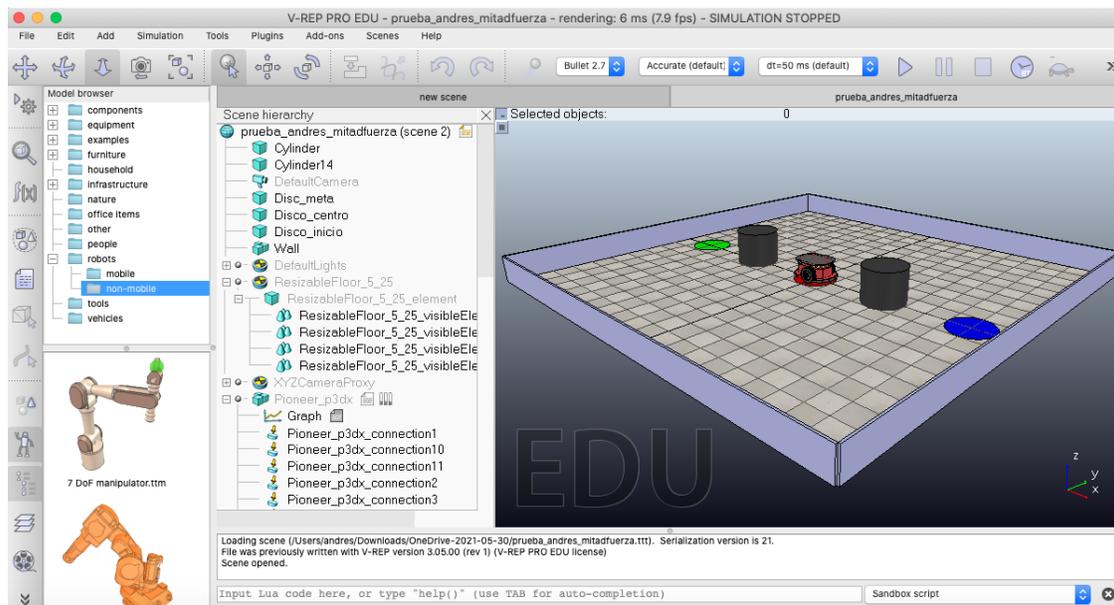


Figura 3.4: Escena básica para la teleoperación de un robot móvil

Fuente: Autor

Posteriormente se creó una escena más compleja como se muestra en la figura 3.5, donde el usuario debe seguir una trayectoria predeterminada, disponiendo de una serie de obstáculos, los cuales son

detectados por el robot móvil diferencial, ya que una de las ventajas es que cuenta con 16 sensores, los cuales son muy provechosos para el caso de estudio.

Vrep permite la inclusión de librerías hápticas como `chai3d`, esta librería es la encargada de re-troalimentar las fuerzas provenientes de los obstáculos detectados por el robot móvil diferencial y transmitirlos al dispositivo Novint Falcon.

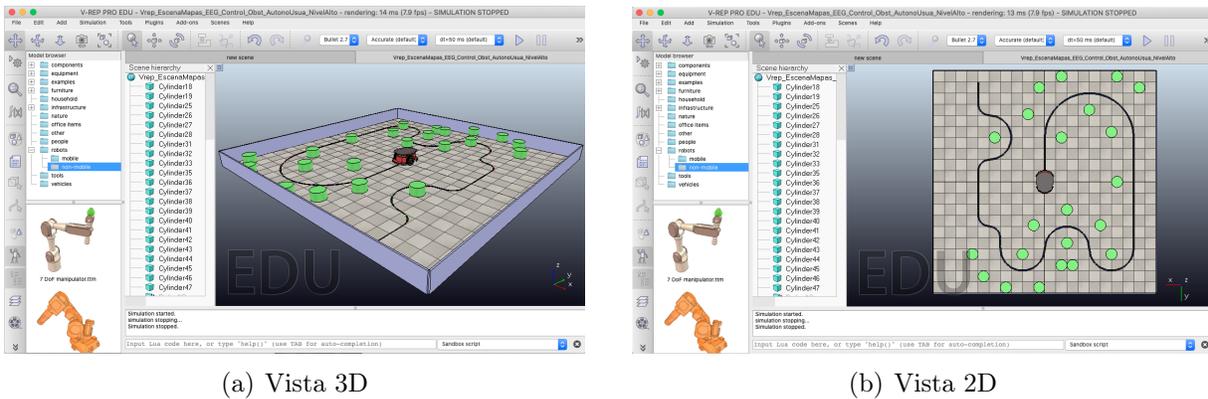


Figura 3.5: Creación de la escena

Fuente: Autor

Por otra parte el programa de vrep tiene la capacidad de incluir modelos cinemáticos y dinámicos del robot.

Esto es de suma importancia dado que con estos modelos es posible conocer la localización (posición - orientación) del robot, parámetro utilizado para el desarrollo del controlador.

En cuanto a la adquisición de datos (ver figura 3.6) se obtienen directamente de la escena en el software de Vrep, para luego ser analizados en un algoritmo de control compartido (ver anexo ??) en la plataforma de Matlab.

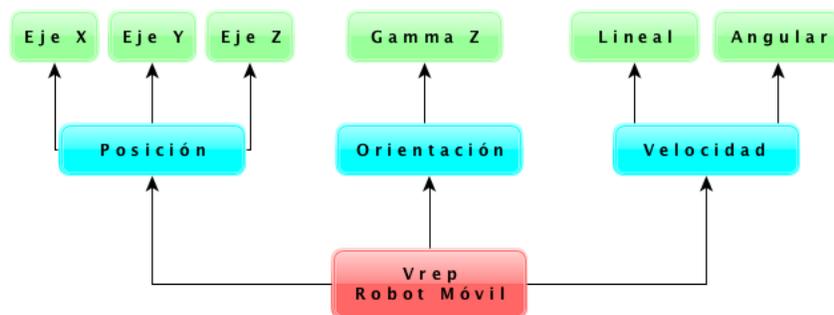


Figura 3.6: Adquisición de datos Vrep

Fuente: Autor

Como se había mencionado anteriormente, Vrep es un programa para simular robots que funciona con Linux, Windows y OS, es gratuito y de código abierto, mientras se use para fines no - comerciales, se utiliza para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones.

En éste proyecto de investigación, se decide iniciar desde este punto, es decir, realizar una plataforma que permitiera simular el comportamiento de un robot móvil tipo diferencial en un entorno con obstáculos que deberá evadir, utilizando el software en su versión de licencia de estudiante gratuita Pro Edu V3.6.2.

Vrep cuenta con un alto grado flexibilidad el cual le permite interactuar con amplia variedad de librerías como Lua y códigos desarrollado en otras plataformas.

La implementación del algoritmo de control de asistencia háptica se visualiza en la sección 3.6, es realizada en el lenguaje de programación Lua, el cual es el predominante en esta interfaz grafica en el entorno del software de vrep.

3.3. Robot móvil diferencial

El robot móvil Pioneer P3dx, es un robot diferencial de uso común en las universidades dado a que es compacto, confiable y preciso, el robot diferencial cuenta con codificadores en sus ruedas y un conjunto de 16 sensores de ultrasonido distribuidos alrededor de su chasis que le permiten detectar obstáculos.

Lo primero que tenemos que conocer del robot móvil diferencial, tiene que ver con la configuración del mismo, esto es, analizar como están distribuidos los principales elementos que lo componen:

- ❖ Ruedas
- ❖ Motores
- ❖ Sensores

En la Figura 3.7 se puede apreciar una imagen del robot móvil diferencial.

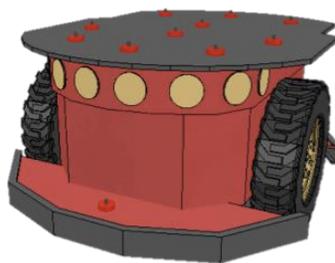


Figura 3.7: Robot Móvil Pionner P3DX
Fuente: [35]

3.3.1. Modelo cinemático del robot diferencial

Para corroborar la relación entre el movimiento de las ruedas del robot y el movimiento de su chasis, se implementan los modelos cinemáticos diferencial directo e inverso del robot.

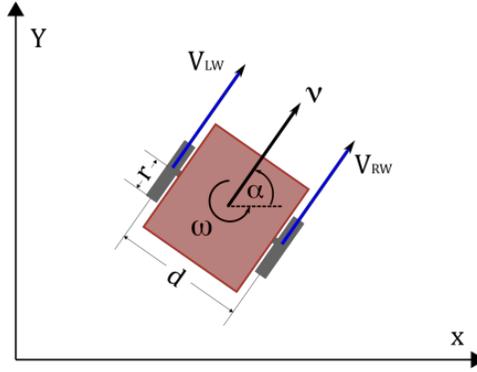


Figura 3.8: Diagrama del modelo cinemático del robot
Fuente: [35]

El modelo cinemático diferencial directo permite calcular la velocidad lineal (v) y angular (ω) del robot a partir de las velocidades angulares de la rueda derecha (ω_{RW}) y la rueda izquierda (ω_{LW}).

En primer lugar se calculan las velocidades lineales de las ruedas a partir de sus velocidades angulares y el radio de las mismas, como se aprecia en las ecuaciones 3.1 y 3.2.

$$V_{RW} = r \cdot \omega_{RW} \quad (3.1)$$

$$V_{LW} = r \cdot \omega_{LW} \quad (3.2)$$

Donde:

- ❖ V_{RW} : Velocidad lineal de la rueda derecha
- ❖ V_{LW} : Velocidad lineal de la rueda izquierda
- ❖ r : radio de la rueda

Conocidas las velocidades lineales de las ruedas se procede a calcular las velocidades del robot (ver Figura 3.8) de la siguiente forma:

- ❖ Velocidad lineal
- ❖ Velocidad angular

$$v = \frac{V_{RW} + V_{LW}}{2} = \frac{r}{2} \cdot (\omega_{RW} + \omega_{LW}) \quad (3.3)$$

$$\omega = \frac{V_{RW} - V_{LW}}{d} = \frac{r}{d} \cdot (\omega_{RW} - \omega_{LW}) \quad (3.4)$$

Donde:

- ❖ d : Distancia entre las ruedas

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = r \begin{bmatrix} 0,5 & 0,5 \\ d^{-1} & -d^{-1} \end{bmatrix} \begin{bmatrix} \omega_{RW} \\ \omega_{LW} \end{bmatrix} \quad (3.5)$$

En caso que se desee calcular las velocidades en un espacio Euclidiano, se puede sustituir la ecuacion 3.5 en la ecuacion 3.6, obteniendo la ecuacion 3.7.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ \sin\alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.6)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = r \begin{bmatrix} 0,5 \cdot \cos\alpha & 0,5 \cdot \cos\alpha \\ 0,5 \cdot \sin\alpha & 0,5 \cdot \sin\alpha \\ d^{-1} & -d^{-1} \end{bmatrix} \begin{bmatrix} \omega_{RW} \\ \omega_{LW} \end{bmatrix} \quad (3.7)$$

Para el caso del cálculo del modelo cinemático diferencial inverso se parte de las velocidades: lineal y angular del robot (v y ω), hasta obtener las velocidades angulares de cada una de las ruedas (ω_{RW} y ω_{LW}). Este modelo se puede obtener fácilmente a partir de las ecuaciones 3.3 y 3.4, obteniendo así:

$$\omega_{RW} = \frac{2v + d\omega}{2r} = \frac{2\sqrt{\dot{x}^2 + \dot{y}^2} + d\dot{\alpha}}{2r} \quad (3.8)$$

$$\omega_{LW} = \frac{2v - d\omega}{2r} = \frac{2\sqrt{\dot{x}^2 + \dot{y}^2} - d\dot{\alpha}}{2r} \quad (3.9)$$

3.4. Dispositivo Novint Falcon

3.4.1. Descripción

El Novint Falcon es un dispositivo fabricado por Novint Technologies Inc. para la interacción en escenarios 3D con realimentación de fuerzas.

Se ha diseñado inicialmente para aplicaciones de entretenimiento, videojuegos o sustitución de periféricos como el mouse o el joystick y por su bajo costo se ha convertido en el pionero en la categoría de productos hápticos para el mercado de consumo.

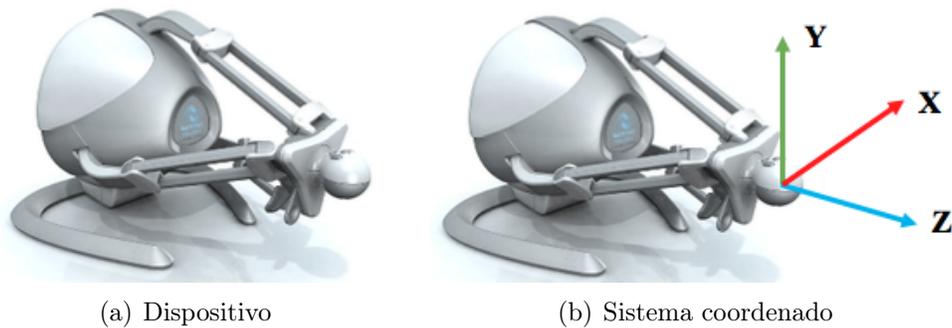


Figura 3.9: Dispositivo Novint Falcon
Fuente: Autor

- ❖ El origen del dispositivo háptico es aproximadamente el centro del espacio de trabajo
- ❖ El sistema coordinado del dispositivo háptico sigue la regla de la mano derecha como se muestra en la figura 3.9
 - X incrementa hacia la derecha
 - Y incrementa hacia arriba
 - Z incrementa hacia el usuario

Este dispositivo háptico mostrado en la figura 3.9, es un esclavo de arquitectura paralela, con un espacio de trabajo de 4" x 4" x 4".

Otras especificaciones técnicas se presentan en la figura 3.10.

Especificaciones de Hardware	
Tamaño	9" x 9" x 9"
Peso	6 lbs
Resolución de posición	> 400 dpi
Interfaz de Comunicación	USB 2.0
Alimentación	30W, 100V- 240V, 50Hz-60Hz
Grado de Libertad	3DOF input, 3DOF output Additional DOF possible through enabled grips
Especificaciones Software	
Plataformas soportadas	Windows XP y Vista Se anticipa migración a PS3 and Xbox
Mínimos requerimientos del sistema	Procesador 1 GHz, tarjeta gráfica de 128MB, DirectX Version: DirectX 9.0c 1.5 GB de espacio libre del disco duro Memoria RAM 512MB Conexión USB 2.0
API	C++ SDK overview Niveles de API: DHDLC (low-level drivers), HDAL (mid-level device communication) Falcon API (high-level programming toolset)
Especificaciones Hapticas	
Espacio de trabajo háptico	4" x 4" x 4"
Capacidad de fuerzas	> 2 lbf
Hilos separados para render háptico y gráfico	
3 grados de libertad con realimentación de fuerzas	

Figura 3.10: Especificaciones técnicas del dispositivo Novint Falcon
Fuente: [38] [46]

3.4.2. Configuración del dispositivo Novint Falcon

Para el enlace del dispositivo Novint Falcon con el software Vrep, primero se instala el programa force dimensión sdk, esto con el fin de instalar todos los drivers del novint y tener una comunicación con el sistema operativo.

Para la comunicación con el software de Vrep, este último tiene predeterminado un ejemplo en donde lo que hace es cargar un Plugin (force dimensión sdk) y hacer uso de el.

El ejemplo consiste realizar una conexión entre el dispositivo y un robot móvil (ver figura 3.7), el cual facilitó mucho a la hora de la implementación.

En la figura 3.11 se puede observar el programa force dimensión sdk, este permite observar el estado de la posición, sensores y botones del dispositivo.

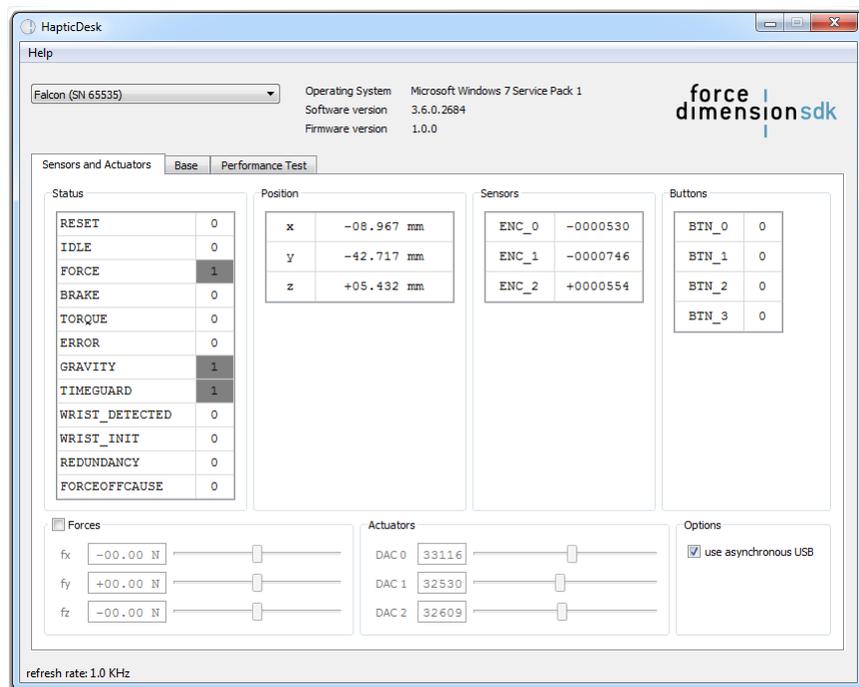


Figura 3.11: Conexión del dispositivo Novint Falcon

Fuente: Autor

3.5. Dispositivo Emotiv Insight Device

Emotiv Insight es un dispositivo diseñado para la interfaz cerebro computador (BCI), este dispositivo cuenta con una electrónica avanzada que está completamente optimizada para producir señales limpias y robustas en cualquier momento y en cualquier lugar.

Este dispositivo (ver figura 3.13) cuenta con cinco sensores, los cuales permiten realizar una detección de las actividades en todo el cerebro, estos sensores son de material de polímero semiseco lo cual le permite una limpieza mas sencilla, además su tiempo de calibración es corto (entre 1 – 5 min).

Así mismo este dispositivo, nos facilita una sincronización con los diferentes dispositivos móviles u ordenadores utilizando un protocolo de comunicación bluetooth.

Para vincular los dispositivos móviles existe una apps llamada MyEmotiv, el cual fue creada por la misma compañía, esta es compatible con sistemas operativos Android y iOS 3.14(a).

Dentro de las características principales de esta aplicación se encuentra la captura de los datos del cerebro, la medida de rendimiento 3.12(a), un visualizador 3D y la opción de ver informes 3.12(b).

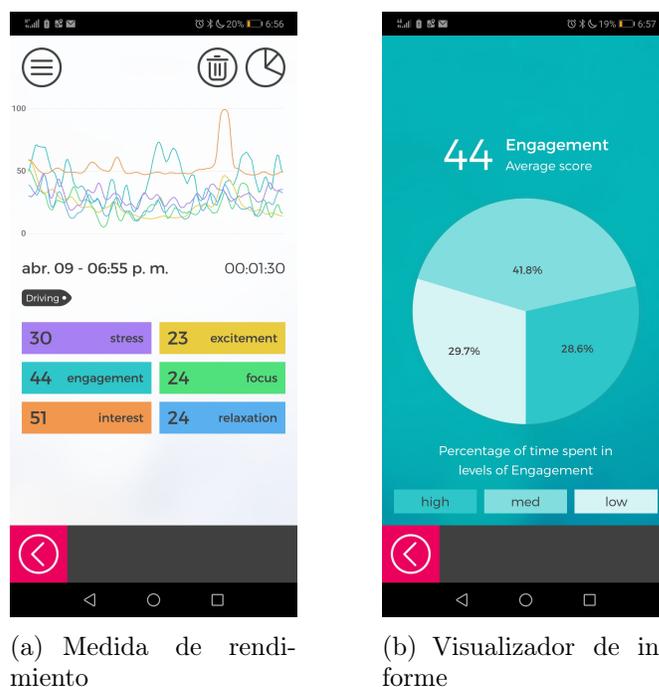


Figura 3.12: Interfaz Neuroseñales
Fuente: Autor

En la figura 3.12(a), se puede observar las mediciones de seis métricas emotivas, donde se han probado y desarrollado haciendo uso de métodos científicos rigurosos, la cual conlleva a muchas pruebas en voluntarios haciendo uso de la diadema emotiv Insight.

Los datos recolectados en las diferentes pruebas, fueron a través del canal de procesamiento de

señales y aprendizaje automático de la empresa, obteniendo como resultado modelos matemáticos detrás de cada medida.

Por otro lado en la figura 3.12(b), se visualiza uno de los informes arrojados por la app, estos informes pueden ser comparados con el tiempo y con otros usuarios.

Por otra parte la empresa realizó un programa para computador el cual lleva por nombre Emotiv Xavier control panel, este programa es compatible con sistemas operativos Windows y MacOS 3.14(b).

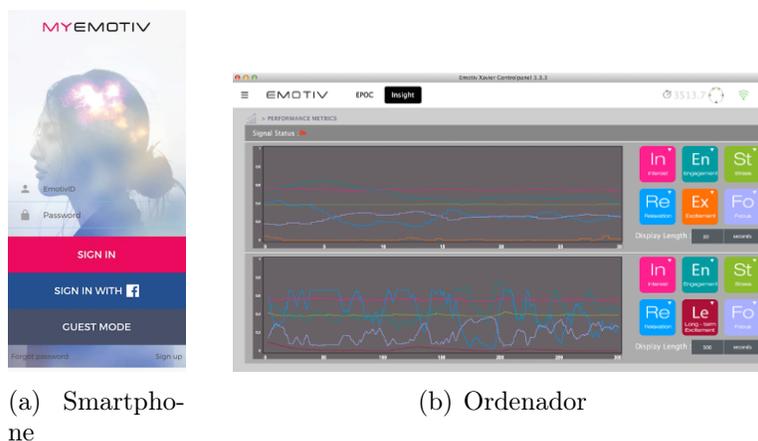
El dispositivo emotiv insight es totalmente compatible con una variedad de flujo de datos que permiten infinitas posibilidades y aplicaciones para BCI, los flujos de datos incluyen señales electroencefalograma EEG sin procesar comandos mentales, métricas de rendimiento, bandas de frecuencia, expresiones faciales y datos de movimiento.



Figura 3.13: Dispositivo Emotiv Insight
Fuente: [12] [9]

El dispositivo Emotiv provee señales cerebrales medidas directamente de los sensores, utilizando un procedimiento no invasivo, específicamente seis medidas básicas de desempeño mental estimadas directamente a partir de la actividad cerebral.

En la figura 3.14 se muestra un ejemplo de la captura de neuroseñales, correspondiente a un smartphone (a) y a una interfaz desde un ordenador (b).



(a) Smartphone

(b) Ordenador

Figura 3.14: Interfaces Principales Neuroseñales
Fuente: Autor

Para registrar las neuroseñales obtenidas y procesadas mediante la interfaz cerebro computador (BCI) del usuario mientras realiza una tarea de teleoperación preestablecida.

Se implementó un algoritmo en la plataforma de programación de matlab (ver anexos ??) que permitió almacenar todos los datos de las señales emotivas por el usuario obtenidas en la ejecución de la tarea como se muestra en la figura 3.12(a) con el fin de transformar dichos datos por medio de un conjunto de operaciones obteniendo como resultados un archivo de texto con extensión “.mat” y una gráfica como se muestra en la figura 4.6(b).

A continuación, se muestra un segmento del código ?? sobre el registro de las neuroseñales en matlab donde se interpreta los datos mencionados anteriormente.

```
clear;, close all;, clc;
tiempo_experimento = 60*2 + 26;
nombre_imagen = 'Sin_Asiistencia'; % Con_Asiistencia; % Adaptativo;
```

En una forma general se mostrarán los fragmentos más importantes del código ?? implementado para realizar el estudio de las neuroseñales obtenidas al realizar las pruebas de cada usuario.

- ❖ **clear** elimina todas las variables del espacio de trabajo actual y las libera de la memoria del sistema
- ❖ **clc** borra todas las entradas y salidas de la pantalla de la ventana de comando, lo que proporciona una pantalla limpia
- ❖ **close all** cierra todas las ventanas de las gráficas abiertas

Inicialmente, se relaciona el tiempo de ejecución realizado por el usuario al cumplir una tarea predeterminada, esto se hace dependiendo la asistencia asociada (sin asistencia, con asistencia, adaptativo), la finalidad de definir esto, es delimitar el tiempo máximo o mínimo total obtenido por segmentos.

```
% Analiza todo:0; segmento:1;
ANALIZAR_SOLO_SEGMENTS_DE_TIEMPO = 0;
tiempo_minimo_analisis = 60*0 + 0;
tiempo_maximo_analisis = 60*10 + 0;
```

Posteriormente se procede a incluir cada una de las neuroseñales obtenidas a graficar según sea la asistencia que se requiera analizar.

```

% Incluir la(s) variables que desea graficar
GRAFICAR_ESTRES           = 1;
GRAFICAR_EXCITACION      = 1;
GRAFICAR_COMPROMISO      = 1;
GRAFICAR_CONCENTRACION  = 1;
GRAFICAR_INTERES         = 1;
GRAFICAR_RELAJACION      = 1;

```

El algoritmo creado, permite configurar el idioma y las características de la gráfica, tales como: ancho de la línea, tamaño de letra y tipo de letra en el que se pretende obtener los resultados.

```

IDIOMA = 0; % Configura el idioma Espanol:0; Ingles:1;

% Caracteristicas de la grafica
ANCHO_DE_LINEA = 2;
TAMANO_LETRA   = 14;
TIPO_DE_LETRA  = 'Times New Roman';

% Inicio del Algoritmo
GRAFICAS_INTERMEDIAS = 0;
extension = '.jpg';
nombre_imagen_ext = [nombre_imagen , extension];
RGB_ORG=imread(nombre_imagen_ext);
figure(1);, imshow(RGB_ORG);

```

Seguido, se importa la imagen proveniente de las pruebas realizadas al usuario (tomada de la aplicación emotiv) y se define los colores de cada una de las neuroseñales, con el fin de obtener una mejor interpretación al momento de su análisis.

```

% Colores de cada neurosenal
color_linea_limite = [224, 226, 225];
color_estres       = [164, 132, 231];
color_excitacion   = [228, 203, 80];
color_compromiso   = [59, 193, 194];
color_concentracion = [96, 215, 131];
color_interes      = [232, 154, 90];
color_relajacion   = [101, 172, 224];

```

A continuación, se realiza el proceso de segmentación de los datos (conjunto de señales), el cual consiste en dividir la imagen digital en varias regiones, para luego binarizar y buscar cada una de las filas y de las columnas mínimas y máximas correspondiente al área de trabajo.

Esto se hace con el fin de recortar la imagen original y tomar los fragmentos más significativos del conjunto de señales.

```

% Proceso de segmentacion de los datos (conjunto de senales)
RGB_fondo = emotiv_conservar_color_margen_de_error(RGB_ORG ,
    color_linea_limite , 0.01);
if GRAFICAS_INTERMEDIAS
    figure;, imshow(RGB_fondo);
end

RGB_fondo_bw = im2bw(RGB_fondo,0.1); % Binariza la imagen
RGB_fondo_bw = bwareaopen(RGB_fondo_bw,100 ); % Elimina las regiones

% Busca las filas y columnas minimas y maximas (area de trabajo:)
[f,c] = size(RGB_fondo_bw);
vec=find(RGB_fondo_bw>0);
fila_ini = fix(min(vec)/f) + 1;
columna_ini = round((min(vec)/f - fix(min(vec)/f))*f);
fila_fin = fix(max(vec)/f) + 1;
columna_fin = round((max(vec)/f - fix(max(vec)/f))*f);

% Recorta la imagen original segmentando el conjunto de senales
RGB_datos = RGB_ORG(columna_ini:columna_fin, fila_ini:fila_fin, :);
if GRAFICAS_INTERMEDIAS
    figure;, imshow(RGB_datos);
end

```

De igual manera, el código nos permite calcular el promedio de cada una de las señales, y visualizarlas por medio de graficas.

```

% Datos estadisticos
Estadisticas.promedio.estres = mean(valor_estres);
Estadisticas.promedio.excitacion = mean(valor_excitacion);
Estadisticas.promedio.compromiso = mean(valor_compromiso);
Estadisticas.promedio.concentracion = mean(valor_concentracion);
Estadisticas.promedio.interes = mean(valor_interes);
Estadisticas.promedio.relajacion = mean(valor_relajacion);

```

Por último, se almacenan todos los datos obtenidos y son exportados a un archivo texto, lo que posteriormente son utilizados en la plataforma de Vrep.

```

% Almacena todos los datos para grabarlos en un archivo de texto
Datos.datos_estres = [tiempo; valor_estres];
Datos.datos_excitacion = [tiempo; valor_excitacion];
Datos.datos_compromiso = [tiempo; valor_compromiso];
Datos.datos_relajacion = [tiempo; valor_relajacion];
Datos.datos_interes = [tiempo; valor_interes];
Datos.datos_concentracion = [tiempo; valor_concentracion];

save([nombre_imagen, '_Datos.mat'], 'Datos')

```

3.6. Control de asistencia háptica

En primer lugar, el usuario se encarga de interactuar con el dispositivo háptico Novint Falcon con el fin de coordinar los movimientos del robot móvil diferencial, el dispositivo además de recibir las indicaciones del usuario se encarga de generar fuerzas que actúan como asistencias hápticas para indicarle al usuario si el robot móvil diferencial colisionó con algún objeto.

Adicionalmente las asistencias hápticas se pueden configurar para ayudarle al usuario a seguir una trayectoria en particular o detectar si existe un obstáculo antes de que se colisione con el mismo.

En la Figura 3.15 se puede apreciar la arquitectura del sistema de teleoperación.

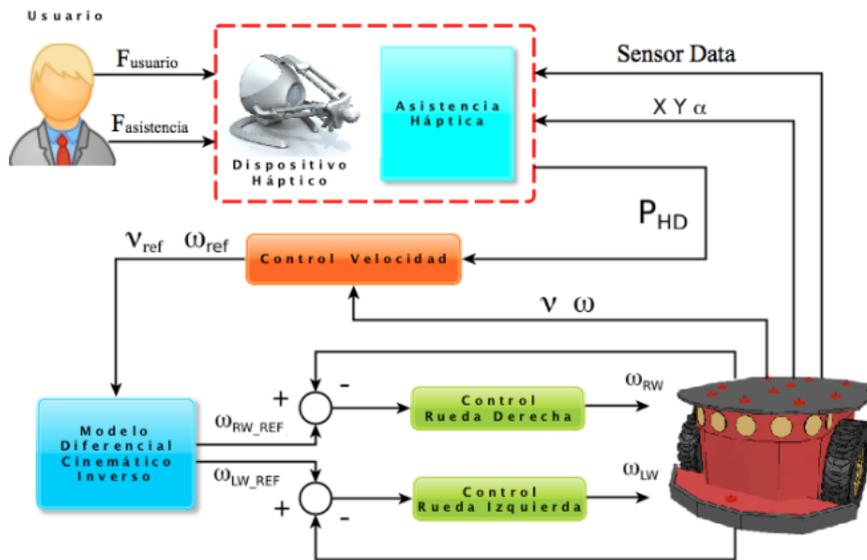


Figura 3.15: Arquitectura de sistema de teleoperación Virtual

Fuente: Adaptación [35]

Como resultado de la interacción de las fuerzas del usuario y las fuerzas de asistencia se obtiene la posición de el efector final del dispositivo háptico (P_{HD}).

Este vector se toma como dato de referencia para generar las velocidades lineal y angular deseadas para el robot, el valor de la magnitud de la velocidad lineal será proporcional a la magnitud de la componente en el eje Z del efector final del dispositivo háptico (P_{zHD}).

Si la componente es positiva el robot móvil ira hacia adelante y si es negativa lo hará en reversa, la velocidad angular se controlara de forma análoga por medio de la componente en el eje X (P_{XHD}).

Por medio del modelo cinemático diferencial inverso, se calculan las velocidades que deben generar cada uno de los actuadores del robot, estas velocidades angulares son las referencias de entrada para los controladores de la velocidades de las dos ruedas, finalmente las señales de los controladores ingresan al modelo que representa al robot móvil Pioneer P3dx en la plataforma de Vrep.

Cabe resaltar que la distancia de los obstáculos con respecto a cada uno de los sensores es retroalimentada al sistema de asistencia háptica para la generación de las fuerzas respectivas.

Como características principales de los experimentos se tienen que las velocidades que puede alcanzar el robot móvil diferencial son relativamente altas y se habilitó el control autónomo, tanto el sistema de asistencia háptica que retroalimenta las fuerzas al usuario correspondientes a campos equipotenciales para detectar obstáculos internos y las paredes externas del entorno de trabajo.

Con respecto al algoritmo de control de asistencia háptica se encuentra en la sección de anexos ?? el cual nos permite establecer las diferentes situaciones, comenzando creando una función, donde se declaran las variables en este caso pertinente a la funcionalidad del robot diferencial, con el fin de obtener información acerca del comportamiento del usuario al teleoperar un robot móvil virtual por medio de un dispositivo háptico.

A continuación, se muestra un segmento del código ?? que se implementó para que el usuario ejecutará los diferentes experimentos mencionados anteriormente, con el fin de recolectar información sobre el control de asistencia háptica.

```
function sysCall_init( )
    usensors = { }
    sensorDetections = { }
    avatarPosition = { }
    proportionalController = { }
```

Con respecto a la variable “proportionalController”, hace referencia al tipo de controlador empleado en el algoritmo de control de asistencia háptica, con el fin de aplicar un control a la escena, buscando obtener mejoras a la hora de ejecutar la tarea asignada al usuario.

El controlador es importado desde la plataforma de matlab, luego de ejecutar un algoritmo de control compartido, obteniendo una matriz de dimensiones [20 * 20] el cual hace referencia al tamaño de la escena.

*Por razones de dimensionamiento, la matriz no es escrita en este código.

En este proyecto se utilizan los scripts embebidos por su facilidad y transigencia gracias al interprete de script que tiene vrep integrado, el tipo de lenguaje que representa vrep es Lua, es un lenguaje de extensión y dado que esta basado en el lenguaje C, fue de gran ayuda a la hora de culminar este proyecto.

Vrep contiene varios tipos de script, pero en este proyecto se van a utilizar los dos más usuales: main script y child script, los primeros controlan el bucle de simulación principal y los otros se encargan de controlar los modelos o robots.

Posteriormente se relaciona el robot móvil pionner p3dx tipo diferencial, que cuenta con 16 sensores, los cuales son llamados mediante un ciclo for con la variable “usensors”.

```
for i = 1,16,1 do
    usensors[i] = sim.getObjectHandle('Pioneer_p3dx_ultrasonicSensor'
    ..i)
    sensorDetections[#sensorDetections+1] = 99
    sensorDetections[#sensorDetections+1] = 99
    sensorDetections[#sensorDetections+1] = 99
end

motorLeft = sim.getObjectHandle('Pioneer_p3dx_leftMotor')
motorRight = sim.getObjectHandle('Pioneer_p3dx_rightMotor')
modelBase = sim.getObjectAssociatedWithScript(sim.handle_self)
pathHandle = sim.getObjectHandle('Path')
sphereHandle = sim.getObjectHandle('Sphere')
targetHandle = sim.getObjectHandle('Target')
```

Para la implementación de la cinemática del robot, es necesario conocer los datos provenientes de los motores, distancia entre las ruedas al centro del robot y de la base, de igual manera del path y el objetivo.

Para esto, se utiliza la función “sim.getObjectHandle('Nombre del objeto’)” con esta función se extraen toda la información requerida del objeto.

En esta función inicialmente se obtiene la posición del avatar con respecto al robot, de igual manera la distancia existente entre el path y el avatar.

Por otra parte se obtiene la posición a la que se desea llegar y la longitud del path, esta última se utiliza para posteriormente calcular la cinemática del robot, por último se utiliza una esfera creada, la cual simula el movimiento que realiza el robot y se define la distancia entre las ruedas al centro del robot.

```

function sysCall_actuation( )
  avatarPosition = sim.getObjectPosition(modelBase , -1)
  distanceOnPath = sim.getClosestPositionOnPath(pathHandle ,
    avatarPosition)
  lengthPath = sim.getPathLength(pathHandle)
  positionOnPath = sim.getPositionOnPath(pathHandle , distanceOnPath)
  spherePosition = sim.getObjectPosition(sphereHandle , -1)
  targetPosition = sim.getObjectPosition(targetHandle , -1)

  spherePosition[1] = positionOnPath[1]
  spherePosition[2] = positionOnPath[2]
  spherePosition[3] = 0.25
  sim.setObjectPosition(sphereHandle , -1, spherePosition)

  LR = 0.3
  LRp = LR / lengthPath
  distanceOnPathT = distanceOnPath + LRp
  if (distanceOnPathT > 1) then
    distanceOnPathT = 1
  end
end

```

Las funciones fundamentales utilizadas en el script son:

- ❖ `sim.getObjectPosition(handle objeto,-1)` devuelve la posición de un determinado objeto, el segundo parámetro indica que sea la posición absoluta
- ❖ `sim.getPositionOnPath('handle Path', 'distancia relativa')` devuelve una posición absoluta interpolada de un punto a lo largo del path, el segundo argumento acepta valores entre 0 y 1. Si es 0 significa que está comenzando el path y si es 1 que ha finalizado
- ❖ `simSetObjectPosition('handle objeto', '-1', 'posición')` fija la posición del tercer parámetro (coordenadas x,y,z) a un objeto

La conexión con el dispositivo háptico, se hace utilizando la librería `chai3d`, esta proporciona un plugin en el cual se puede obtener la retroalimentación de fuerza, posición de los motores, conexión del dispositivo entre otras, por lo tanto el primer paso desarrollado es verificar si la conexión del dispositivo háptico es exitosa o no, esto mediante un `while`, el cual verifica si el dispositivo está conectado o no.

Si el dispositivo no está conectado, mostrara un mensaje diciendo que el dispositivo no es encontrado o el plugin no es inicializado correctamente.

```

moduleName = 0
moduleVersion = 0
index = 0
pluginNotFound = true
while (moduleName) do
  moduleName, moduleVersion = sim.getModuleName(index)
  if (moduleName == 'CHAI3D') then
    pluginNotFound = false
  end
  index = index+1
end

```

Si el dispositivo es conectado correctamente se procede a inicializar el plugin, esto mediante la función “simCHAI3D.start”, de igual manera se crean planos restrictivos, con el fin de limitar el campo de acción del dispositivo háptico, para esto se usa la función “simCHAI3D.addConstraintPlane()” y “simCHAI3D.addConstraintSegment()”.

Luego se procede a obtener los parámetros necesarios para el cálculo de la cinemática del robot.

Se utiliza la función “sim.setJointTargetVelocity(handle objeto, velocidad objetivo)” la cual fija la velocidad del objetivo en una articulación (motor), este comando se usa porque las dos articulaciones de nuestro robot móvil son de tipo par/fuerza, además de estar activado el motor y el control por velocidad y no por posición.

```

if (CHAI3DPluginInitialized) then
  local rotM = sim.buildMatrix({0,0,0},{0,0,math.pi})
  local p = simCHAI3D.readPosition(0)
  p = sim.multiplyVector(rotM, p)

  local forward = p[1] * 8
  local rot = p[2] * 13
  vLeft = forward - rot
  vRight = forward + rot

  sim.setJointTargetVelocity(motorLeft, vLeft)
  sim.setJointTargetVelocity(motorRight, vRight)
end

```

Se condiciona la distancia entre el robot y el path, si la distancia es mayor a 0.99, los motores obtienen una velocidad de 0.

```

if (distanceOnPath > 0.99) then
  vLeft = 0
  vRight = 0
  sim.setJointTargetVelocity(motorLeft, vLeft)
  sim.setJointTargetVelocity(motorRight, vRight)
end

```

La función encargada del sensado del robot móvil diferencial, en esta se definen las respectivas transformaciones.

```
function sysCall_sensing( )
if (CHAI3DPluginInitialized) then

local rot = sim.buildMatrix({0,0,0},{0,0,math.pi})

for i = 1,16,1 do
local res,dist,pt = sim.readProximitySensor(usensors[i])
if res > 0 then
local m = sim.getObjectMatrix(usensors[i],modelBase)
local p = sim.multiplyVector(m, pt)
sensorDetections[3*(i - 1) + 1] = p[1]
sensorDetections[3*(i - 1) + 2] = p[2]
sensorDetections[3*(i - 1) + 3] = p[3]
else
sensorDetections[3*(i-1) + 1] = p[1]
sensorDetections[3*(i-1) + 2] = p[2]
sensorDetections[3*(i-1) + 3] = p[3]
end
end

for i = 1,16,1 do
local x = sensorDetections[3*(i - 1) + 1]
local y = sensorDetections[3*(i - 1) + 2]
local l = math.sqrt(x*x + y*y)

x = x / l
if i > 8 then
y = -y / l
else
y = y / l
end
local f = 0
local maxF = 7
local minFR = 0.5
local maxFR = 0.2
```

Se definen las respectivas asistencias, esto se hace asociando la matriz generada por el control adaptativo en matlab.

```
avatarPosition = sim.getObjectPosition(modelBase,-1)
Xp = avatarPosition[1]
Yp = avatarPosition[2]
Cx = math.floor(Xp / 0.25) + 11
Cy = 10 - math.floor(Yp / 0.25)
k = proportionalController[Cx + (Cy - 1)*20]

    if l < minFR then
        local t = maxF / (maxFR - minFR)
        local s = -minFR * t
        f = math.max(0, l * t+s)
    end

simCHAI3D.updateConstraint(obstacleConstr[i],{x,y,0},{0,0,0},0.25*k,
    0,f*k)

    directionForce = sim.getObjectPosition(targetHandle,modelBase)
    directionForce[1] = directionForce[1] * 8/13

kpControl = 0.4*k
kvControl = 0.2*k
FmaxControl = 5*k

if (distanceOnPath > 0.99) then
    FmaxControl = 0
end

simExtCHAI3D_updateConstraint(controlTarget,{-1*directionForce[1], -1*
    directionForce[2], -1*directionForce[3]},{0,0,0},kpControl, kvControl,
    FmaxControl)
```

3.7. Control compartido

El algoritmo de control compartido se encuentra en la sección de anexos ??, el código nos permite obtener como resultados, gráficas en las que se puede obtener el tiempo de ejecución de la escena, sus niveles de emoción cuando se ejecuta una tarea al teleoperar un robot móvil, entre otros.

A continuación, se presentan fracciones relacionadas al código:

El primer paso de este algoritmo de control, es asociar los datos de entrada, los cuales son procedentes de los resultados obtenidos del análisis del registro de las neuroseñales, como resultado se obtiene un archivo de datos con extensión “.mat” y la obtención de los datos provenientes del entorno de vrep en un archivo de valores separados por comas cuando el usuario ejecuta una tarea preestablecida, los datos referentes contienen las emociones y el comportamiento de las variables (posición orientación y velocidad) de la trayectoria al usuario realizar la tareas de teleoperación.

```
% Datos de entrada
NombreVrep = 'AP_A0_28-01-20_CP';
NombreEmotiv = 'Con_Asistencia_Datos_CP.mat' ;

% Datos de la trayectoria
archivoCSV = [NombreVrep, '.csv'];
path_user = csvread(archivoCSV,2,0);
tiempo_max = inf;
view(180,90);
axis([-2.5, 2.5, -2.5, 2.5, 0, 2.5]);
grid on;

% Datos del Emotiv Insight
load(NombreEmotiv);
```

Posteriormente, se crea la escena realizada en el entorno de vrep en la plataforma de matlab, inicialmente se empieza dibujando el suelo, la escena cuenta con dimensiones de 5x5 metros, luego se incorporan las paredes y se dibujan los obstáculos de acuerdo al nivel de complejidad media o alta, luego se crear la trayectoria original predeterminada.

Ya creada la escena en el entorno de matlab, se procede a vincular los datos de la posición y orientación recorrida por el robot móvil, con el fin de acondicionar los tiempos de ejecución de tal manera que coincidan con los obtenidos por el usuario al realizar la tarea.

Para hacer un análisis del usuario al realizar una prueba de teleoperación de un robot móvil, con respecto al error, se inicia presumiendo los datos estadísticos, estos datos son obtenidos al comparar la distancia existente de la trayectoria original dibujada inicialmente y la realizada por el usuario.

Dado que son muchos los datos estos son reducidos, con la finalidad de lograr un mejor análisis y para obtener una identificación mejor, con respecto a la orientación del robot móvil diferencial es dibujada por medio de flechas, de igual manera para que el análisis sea aún mas fácil, se discretizaron los datos de la trayectoria para obtener una gráfica en forma de puntos. Como salida

se obtienen graficas tanto la posición y la orientación de la trayectoria original como discreta del robot.

```
% Creando Escena_Vrep
DIBUJAR_TRAYECTORIA_ORIGINAL = 1;
REDUCIR_DATOS = 1;
DIBUJAR_FLECHAS_ORIENTACIONES_ORIGINALES = 0;
DIBUJAR_TRAYECTORIA_DISCRETA = 0;
DIBUJAR_FLECHAS_ORIENTACIONES_DISCRETAS = 0;
GRAFICAS_POSICIONES_ORIENTACIONES_ORIGINALES_Y_DISCRETAS = 0;
GENERA_ESCENA_VREP_NIVEL_MEDIO = 0;
GENERA_ESCENA_VREP_NIVEL_ALTO = 1;
DATOS_ESTADISTICOS = 1;
GRABAR_GRAFICAS = 0;
ACOMODAR_TIEMPO_EMOTIV_Y_VREP = 1;
```

Teniendo en cuenta los datos dibujados anteriormente mencionados, se procede a importar los datos provenientes del dispositivo Emotiv Insight, con el fin de que exista un orden y una mejor interpretación, se normalizan estos datos de menor a mayor grado según sea la emoción.

```
% Grafica las Neurosenales 3D
GRAFICAR_COMPROMISO_3D = 0; % Si: 1 No: 0
GRAFICAR_ESTRES_3D = 0; % Si: 1 No: 0
GRAFICAR_ENTUSIASMO_3D = 0; % Si: 1 No: 0
GRAFICAR_RELAJACION_3D = 0; % Si: 1 No: 0
GRAFICAR_INTERES_3D = 0; % Si: 1 No: 0
GRAFICAR_CONCENTRACION_3D = 0; % Si: 1 No: 0

% Grafica las Neurosenales promedio Barras
GRAFICAR_COMPROMISO_PROMEDIO_MAPA_3D = 0; % Si: 1 No: 0
GRAFICAR_ESTRES_PROMEDIO_MAPA_3D = 0; % Si: 1 No: 0
GRAFICAR_ENTUSIASMO_PROMEDIO_MAPA_3D = 0; % Si: 1 No: 0
GRAFICAR_RELAJACION_PROMEDIO_MAPA_3D = 0; % Si: 1 No: 0
GRAFICAR_INTERES_PROMEDIO_MAPA_3D = 0; % Si: 1 No: 0
GRAFICAR_CONCENTRACION_PROMEDIO_MAPA_3D = 0; % Si: 1 No: 0

% Imagenes de mapas promedio de neurosenales segun su orientacion
GRAFICAR_COMPROMISO_PROMEDIO_MAPA_ORIENTACIONES = 0; % Si: 1 No: 0
GRAFICAR_ESTRES_PROMEDIO_MAPA_ORIENTACIONES = 0; % Si: 1 No: 0
GRAFICAR_ENTUSIASMO_PROMEDIO_MAPA_ORIENTACIONES = 0; % Si: 1 No: 0
GRAFICAR_RELAJACION_PROMEDIO_MAPA_ORIENTACIONES = 0; % Si: 1 No: 0
GRAFICAR_INTERES_PROMEDIO_MAPA_ORIENTACIONES = 0; % Si: 1 No: 0
GRAFICAR_CONCENTRACION_PROMEDIO_MAPA_ORIENTACIONES = 0; % Si: 1 No: 0
```

Posteriormente, para que los datos de las señales coincidan con los de la trayectoria, se interpolan los puntos dibujados inicialmente, las neuroseñales, son dibujadas por medio de cilindros según sea la emoción en la trayectoria.

Por otra parte, se visualizan las imágenes de los mapas promedio de las emociones según su orientación, los resultados de este algoritmo de control se pueden apreciar en el capítulo 4.

3.8. Control adaptativo

En esta sección se ilustrará el procedimiento realizado para la generación de tablas con respecto a la ley de control adaptativa.

Inicialmente se importan los tres experimentos realizados al usuario (sin asistencia, con asistencia y control adaptativo). Posteriormente, se generan los mapas de cada uno de los experimentos realizados en el algoritmo de control compartido, la finalidad con la creación de los mapas, es poder analizarlos y así poder crear las reglas que permitirán realizar el control adaptativo.

```

clear, close all, clc
EVALUAR_TODOS_LOS_USUARIOS = 1;
usuario_ini = 1;
usuario_fin = 1;

Arreglo_nombre_experimentos = {...
    'SAP_SAO_28-01-20_CP', 'Sin_asistencia_Datos_CP.mat', ...
    'AP_AO_28-01-20_CP', 'Con_Asiistencia_Datos_CP.mat', ...
    'Adaptativo_28-01-20_CP', 'Adaptativo_Datos_CP.mat'    };

[nusuarios, nada] = size(Arreglo_nombre_experimentos);

for i=1:nusuarios

    Arreglo_Nombres_Emotiv = {Arreglo_nombre_experimentos{i,2},
        Arreglo_nombre_experimentos{i,4}, Arreglo_nombre_experimentos{i,6}};
    [min_normalizado, max_normalizado] =
        encontrar_min_max_entre_varios_experimentos_emotiv(
            Arreglo_Nombres_Emotiv);

    % Analiza experimentos usuario i sin asistencia
    NombreVrep      = Arreglo_nombre_experimentos{i,1};
    NombreEmotiv    = Arreglo_nombre_experimentos{i,2};
    Musuarios{i,1} = generar_mapas_topes_normalizados(NombreVrep,
        NombreEmotiv, min_normalizado, max_normalizado);

    % Analiza experimentos usuario i con asistencia
    NombreVrep      = Arreglo_nombre_experimentos{i,3};
    NombreEmotiv    = Arreglo_nombre_experimentos{i,4};
    Musuarios{i,2} = generar_mapas_topes_normalizados(NombreVrep,
        NombreEmotiv, min_normalizado, max_normalizado);

    % Analiza experimentos usuario i con asistencia adaptativa
    NombreVrep      = Arreglo_nombre_experimentos{i,5};
    NombreEmotiv    = Arreglo_nombre_experimentos{i,6};
    Musuarios{i,3} = generar_mapas_topes_normalizados(NombreVrep,
        NombreEmotiv, min_normalizado, max_normalizado);
end

```

A partir de este análisis se infieren las reglas que me permiten establecer en que momento o lugares de la zona de trabajo se le da mayor o menor prioridad al control autónomo o a la teleoperación como se muestra en la figura 3.16.

NIVELES NEUROSEÑALES		
Nivel_SA	Nivel_CA	Nivel_Neutro
BAJO	BAJO	MEDIO
	MEDIO	BAJO
	ALTO	BAJO
MEDIO	BAJO	ALTO
	MEDIO	MEDIO
	ALTO	BAJO
ALTO	BAJO	ALTO
	MEDIO	ALTO
	ALTO	ALTO

Figura 3.16: Reglas establecidas
Fuente: Autor

- * SA: Sin asistencia
- * CA: Con asistencia

A continuación, se genera las referencias de la variable “MatrizK” con el fin de vincularla al control de asistencia háptica en la plataforma de vrep.

Definimos los niveles de las neuroseñales como se muestra a continuación:

- ❖ Nivel Bajo [0.000 - 0.333]
- ❖ Nivel Medio [0.334 - 0.667]
- ❖ Nivel Alto [0.668 - 1.000]

```
% Definicion de los niveles de las neurosenales
N_bajo = 0.333;
N_medio = 0.667;
N_alto = 1;

% Definicion de los niveles de las constantes
K_bajo = 0.1;
K_medio = 0.5;
K_alto = 1;

N_bajo_L = 1;
N_medio_L = 2;
N_alto_L = 3;
```

Por medio de las reglas anteriores se establece la ley de control:

$$A_T = K * A_{Obs} + K * A_{Path} \quad (3.10)$$

Donde:

- ❖ $A_T \rightarrow$ Asistencia total
- ❖ $K \rightarrow$ Constante proporcional
- ❖ $A_{Obs} \rightarrow$ Asistencia objetos
- ❖ $A_{Path} \rightarrow$ Asistencia path

Inicialmente se establece la matriz en donde será guardada todos los datos provenientes del algoritmo de control adaptativo, posteriormente se procede a exportar los datos a un archivo “txt” en una variable proportionalController, con el fin de incluirla en el código del software de vrep.

Para la definición de la tabla de ganancias de la ley de control adaptativa se requiere que el usuario realice una serie de experimentos con el fin de analizar su comportamiento a la hora de realizar una tarea asignada.

```

MatrizK = zeros(20,20);
for i=1:400
    if MSAN(i) == N_bajo_L & MCAN(i) == N_bajo_L
        MatrizK(i) = K_medio;
    end
    if MSAN(i) == N_bajo_L & MCAN(i) == N_medio_L
        MatrizK(i) = K_bajo;
    end
    if MSAN(i) == N_bajo_L & MCAN(i) == N_alto_L
        MatrizK(i) = K_bajo;
    end

    if MSAN(i) == N_medio_L & MCAN(i) == N_bajo_L
        MatrizK(i) = K_alto;
    end
    if MSAN(i) == N_medio_L & MCAN(i) == N_medio_L
        MatrizK(i) = K_medio;
    end
    if MSAN(i) == N_medio_L & MCAN(i) == N_alto_L
        MatrizK(i) = K_bajo;
    end

    if MSAN(i) == N_alto_L & MCAN(i) == N_bajo_L
        MatrizK(i) = K_alto;
    end
    if MSAN(i) == N_alto_L & MCAN(i) == N_medio_L
        MatrizK(i) = K_alto;
    end
    if MSAN(i) == N_alto_L & MCAN(i) == N_alto_L
        MatrizK(i) = K_medio;
    end
end
end

```

En siguiente fragmento se muestra como es exportada la matriz K (controlador adaptativo) a un archivo de texto, para posteriormente ser usado en la plataforma de Vrep.

```
M = MatrizK;
M = M';
fileID = fopen('MatrizK.txt','w');
fprintf(fileID,'proportionalController={ ');
position = ftell(fileID)
fseek(fileID, position-28, 'bof');
fprintf(fileID, '}');
fclose(fileID);
```

Capítulo 4

Resultados

4.1. Algoritmo de Control

4.1.1. Implementación del sistema de teleoperación virtual

Como se mencionó en la sección 2.4 el software utilizado en este proyecto es Vrep, por sus excelentes características y por la inclusión de algunos dispositivos hápticos como lo es el Novint Falcon, donde el usuario debe seguir una trayectoria (path) como se muestra en la figura 3.1, evadiendo los obstáculos por medio del dispositivo háptico.

Mientras el usuario realiza los movimientos se irán almacenando los estados emotivos del mismo con el fin de utilizar estos datos para generar una estrategia de control que defina el nivel de autoridad para la ejecución de la tarea.

Para el desarrollo de esta trayectoria se mantuvo fijo el ángulo de visualización, esto con el fin de estudiar en un futuro si los estados emotivos del usuario cambian o no de acuerdo a los problemas perdidos de información gráfica, transformaciones de movimiento al mover el robot móvil en diferentes sentidos, entre otros.

Con el fin de realizar un estudio más profundo de los movimientos realizados por el usuario en cada uno de los experimentos, se exportaron los datos capturados en la plataforma de Vrep, de tal forma que puedan ser interpretados por las plataformas Matlab u Octave.

En la figura 4.1 se puede visualizar la trayectoria (color rojo), realizada por el usuario con el propósito de cumplir con una determinada tarea en el entorno de Vrep como se muestra en la figura 3.5, para luego procesar dicha información, obteniendo como resultado la creación de la escena (ver figura 3.2) en el entorno de matlab.

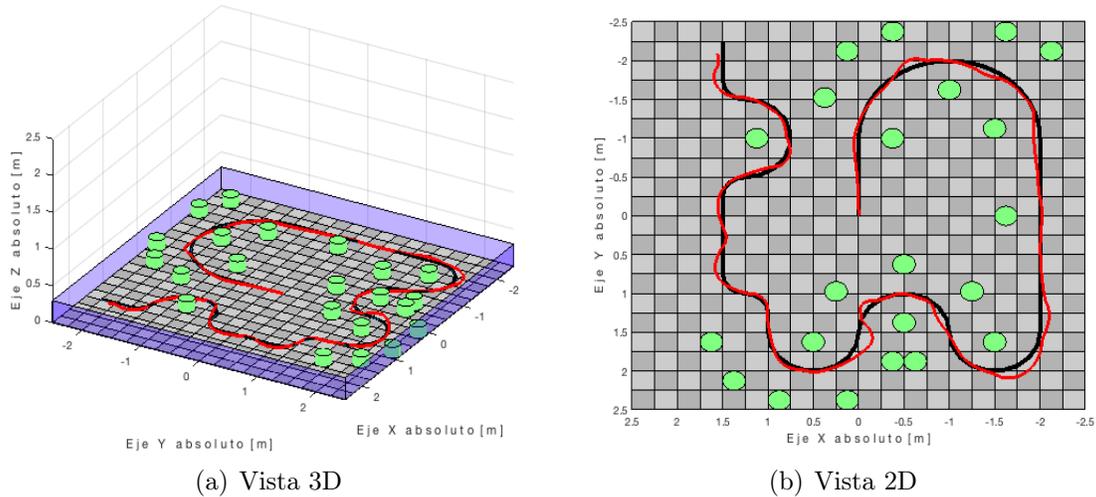


Figura 4.1: Recreación de los datos exportados por Vrep - Scripts Matlab/Octave
Fuente: Autor

Un estudio en particular que se puede efectuar, corresponde con la orientación del robot móvil a lo largo de la trayectoria desarrollada por el usuario

En la figura 4.2, se pueden apreciar una trayectoria en color amarillo que indican esta orientación en algunos puntos (no se visualizaron todos los puntos capturados, debido a que el volumen de información es elevado e impide una apreciación clara).

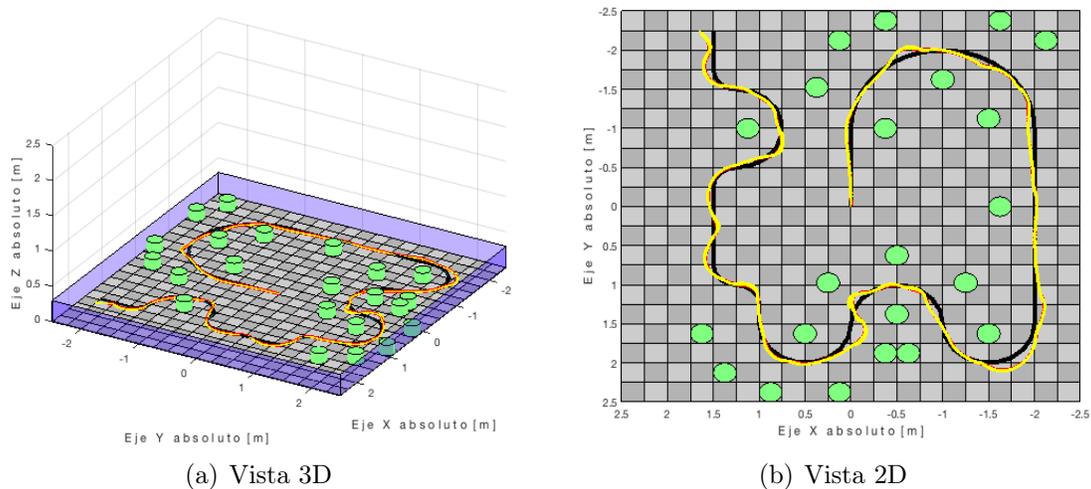


Figura 4.2: Indicación de la orientación del robot durante la ejecución de la trayectoria
Fuente: Autor

La idea general es caracterizar cada una de estas zonas y generar algoritmos que describan el comportamiento emotivo de los usuarios al pasar por cada una de ellas.

En la figura 4.3 se puede ver las zonas involucradas durante la ejecución de la trayectoria generada por el usuario (zonas marcadas en su centro por medio de un punto).

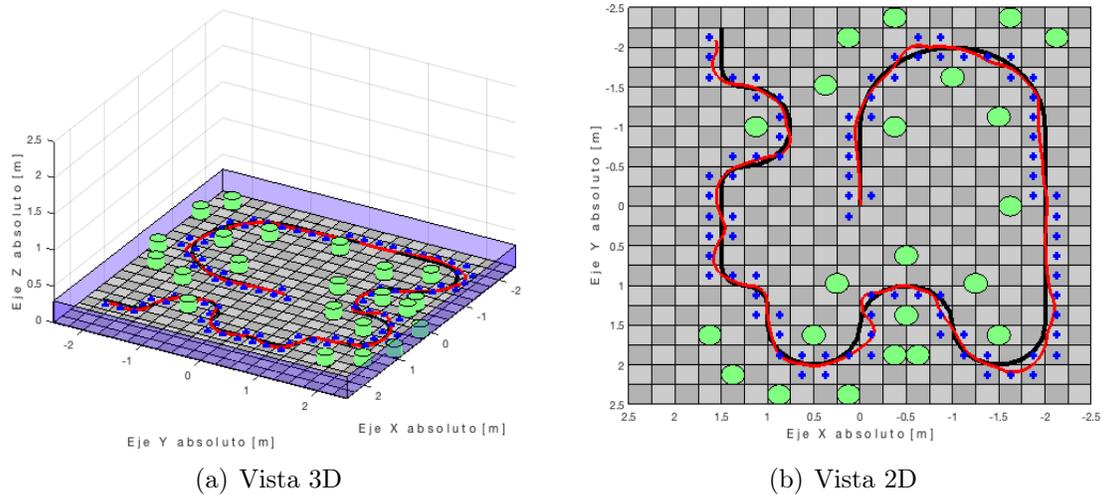


Figura 4.3: Discretización de la trayectoria por secciones o zonas

Fuente: Autor

Adicionalmente se generó un algoritmo que clasifica todos los puntos de la trayectoria dentro de las zonas preestablecidas (color cyan), generan matrices que indican medidas de frecuencia y si se desea, puede generar promedios de las orientaciones, como se aprecia en la figura 4.4.

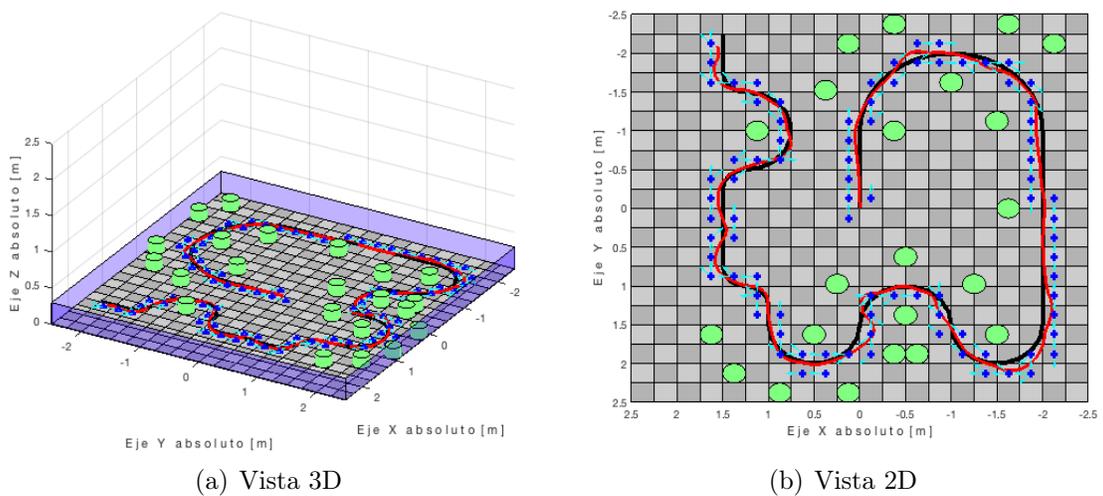


Figura 4.4: Orientación del robot de acuerdo a la trayectoria generada por secciones o zonas

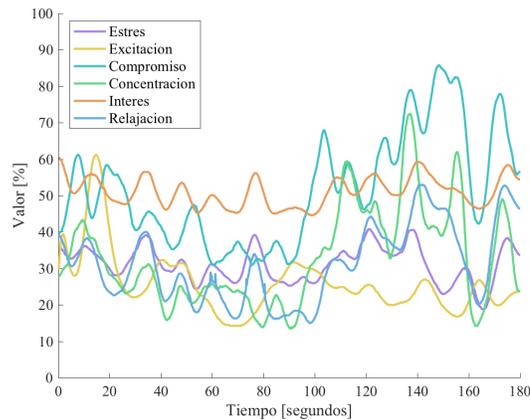
Fuente: Autor

En la figura 4.5 se ilustra la diferencia entre los datos obtenidos por la trayectoria original (color rojo) y los puntos discretizados (centro de las zonas), sobre la posición con respecto al eje X e Y, y la orientación del robot móvil como se muestran a continuación en sus respectivas figuras:

- ❖ Posición del robot móvil sobre el eje X de acuerdo a las trayectorias originales y discretizadas por zonas
- ❖ Posición del robot móvil sobre el eje Y de acuerdo a las trayectorias originales y discretizadas por zonas



(a) Panel principal



(b) Datos exportados a Matlab/Octave

Figura 4.6: Captura de los niveles emotivos por medio del panel de control de la interfaz cerebro computador

Fuente: Autor

Dado que es necesario relacionar los estados emotivos con la trayectoria que realiza el usuario al mover el robot móvil, se hizo necesario exportar estos datos a la misma plataforma como se muestra en la figura 4.6(b), resultado obtenido al implementar el algoritmo ?? que analiza el movimiento del robot.

Con el fin de observar la evolución de los estados emotivos a lo largo de la trayectoria, se generó un algoritmo ?? que ilustra los resultados de neuroseñales sobre cada uno de los puntos o coordenadas desarrolladas por el robot móvil en el escenario, la altura representa el valor del estado emotivo que varía de 0 a 1.

A continuación, se ilustran los resultados de los diferentes niveles emotivos:

En la figura 4.7 se observa que en las zonas donde hay una línea recta los usuarios tienden a tener menos compromiso, caso contrario donde existen curvas y la existencia de más obstáculos, los niveles de compromiso suben al máximo.

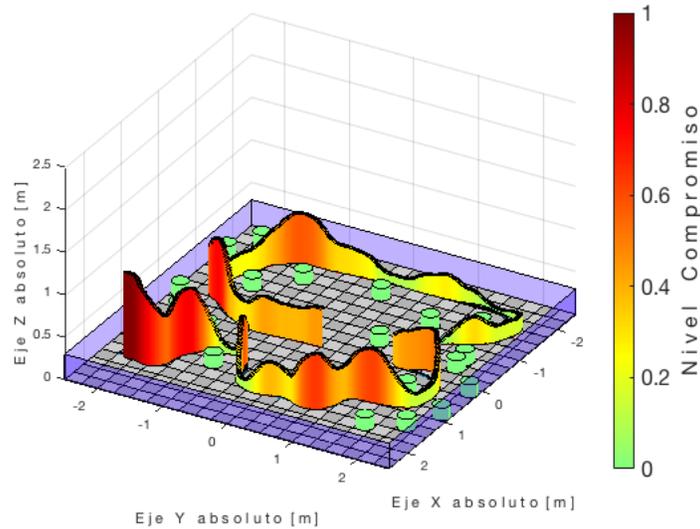


Figura 4.7: Nivel de compromiso durante la ejecución de la trayectoria
Fuente: Autor

Por otra parte los niveles de estrés como se muestra en la figura 4.8, inician en un nivel bastante bajo, pero a medida que el operador avanza estos aumentan, llegando a niveles máximo de estrés.

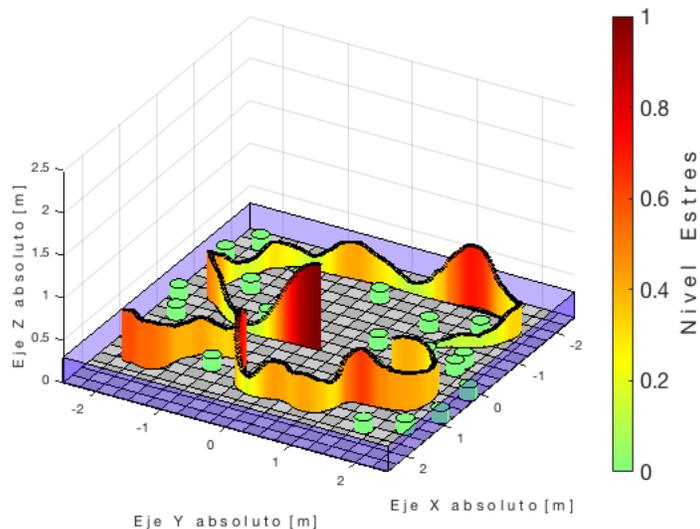


Figura 4.8: Nivel de estrés durante la ejecución de la trayectoria
Fuente: Autor

Así mismo, los niveles de entusiasmo (ver figura 4.9) comienzan con niveles bajos en las zonas en la que hay pocos obstáculos, a medida que el usuario avanza, se encuentra con mas obstáculos, lo que conlleva a manipular mejor el robot, como resultado se observa que su nivel de entusiasmo aumenta.

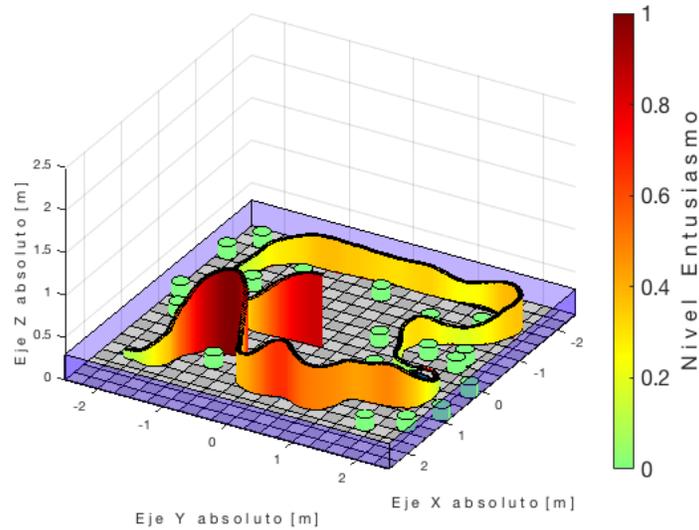


Figura 4.9: Nivel de entusiasmo durante la ejecución de la trayectoria
Fuente: Autor

Los niveles de relajación que se muestra en la figura 4.10 se mantuvieron constantes en la mitad de la prueba, cuando el usuario se encuentra finalizando la prueba, sus niveles de relajación aumenta.

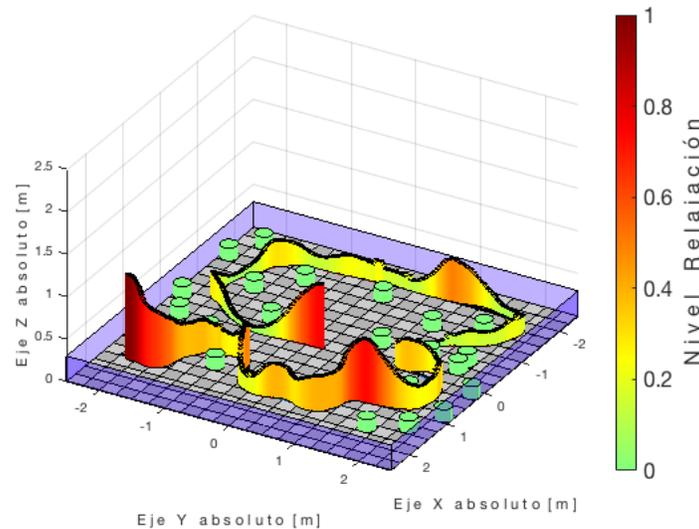


Figura 4.10: Nivel de relajación durante la ejecución de la trayectoria
Fuente: Autor

En la figura 4.11 el operador mantiene un nivel medio de interés constante durante toda la prueba, a excepción de la etapa final, allí su nivel aumenta de manera considerable, lo que da como conclusión que el usuario termina bastante interesado al realizar la prueba.

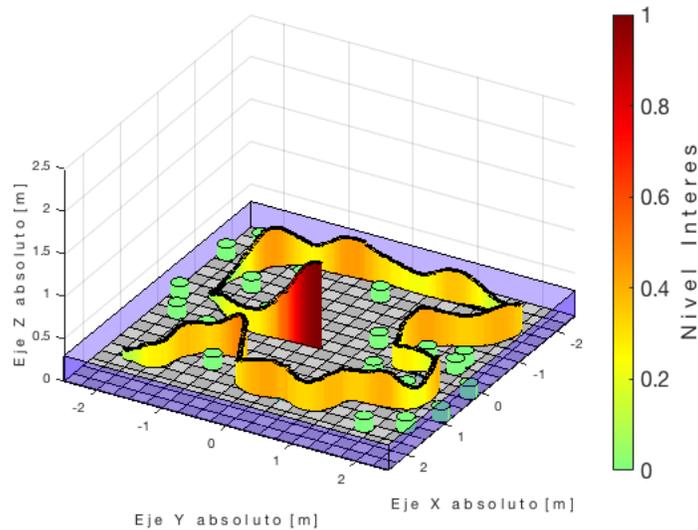


Figura 4.11: Nivel de interés durante la ejecución de la trayectoria
Fuente: Autor

En la figura 4.12 se muestra los niveles de concentración del usuario, donde inicia en un nivel bajo, se mantienen así en la zona en la que hay pocos obstáculos y la línea de operación es recta, a medida que avanza el usuario se concentra más, esto debido a los obstáculos y a las curvas que existen en la escena.

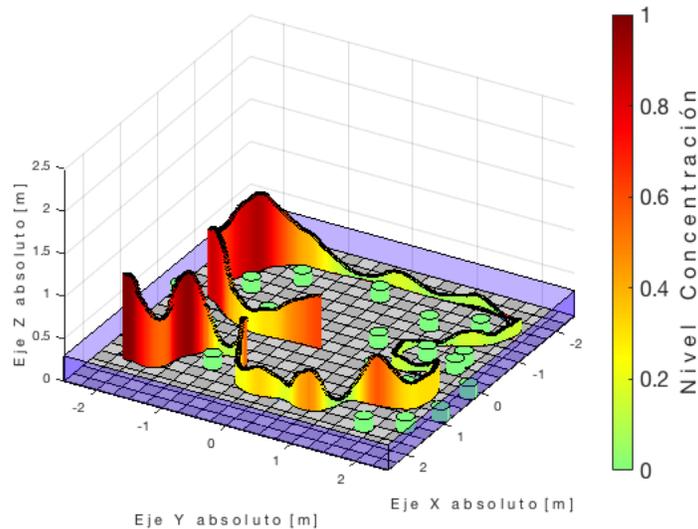


Figura 4.12: Nivel de concentración durante la ejecución de la trayectoria
Fuente: Autor

4.1.3. Implementación de las estrategias de asistencia háptica

4.1.3.1. Estrategia de control compartido, donde el nivel de autoridad es constante durante la ejecución de la tarea

En este caso las fuerzas generadas por el dispositivo háptico se establecen manteniendo un nivel de autoridad constante entre el usuario y el sistema de asistencia. Por lo cual todas las fuerzas generadas dependerán solamente de la posición relativa del robot móvil con respecto a los obstáculos.

4.1.3.2. Estrategia de control compartido, donde el nivel de autoridad es adaptativo durante la ejecución de la tarea.

Para este caso las fuerzas generadas por el dispositivo háptico se generan de acuerdo a una ley de control que cambia de forma adaptativa el nivel de autoridad entre el usuario y el sistema de asistencia. Por lo cual las fuerzas generadas dependerán además de la posición relativa del robot móvil con respecto a los obstáculos, del promedio de un estado emotivo previamente establecido de un usuario en las diferentes zonas de trabajo.

4.1.3.3. Sistema de almacenamiento de variables emotivas para la definición de la ley de control adaptativa

Con el fin de almacenar el nivel emotivo del usuario en cada una de las zonas, se generó un algoritmo de control que registra y clasifica según la posición del robot cada uno de los estados emotivos del operador. Cada zona cuenta con un contador que lleva la frecuencia de los datos registrados y a su vez calcula el promedio de cada uno de los estados emotivos.

A continuación, se ilustran los resultados obtenidos en cada uno de los niveles emotivos de cada una de las zonas involucradas en la trayectoria en sus respectivas vistas en tres dimensiones 3D y dos dimensiones 2D.

En la figura 4.13, se puede observar el nivel de la neuroseñal correspondiente al compromiso, se denota que al empezar a realizar la teleoperación, los niveles son medios, en las zonas donde no hay presencia de tantos obstáculos, estos niveles bajan.

Por otra parte, en las zonas donde hay mas obstáculos, los niveles correspondientes a la emoción suben en ocasiones hasta el máximo. Esto es debido a que el usuario, mientras no realice una actividad donde le requiera tanto esfuerzo, su compromiso va disminuyendo.

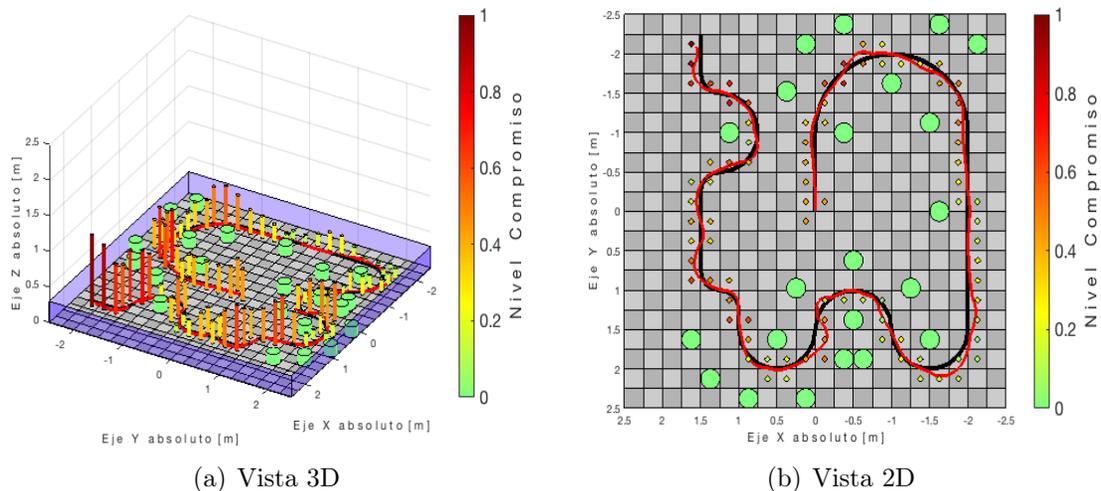


Figura 4.13: Nivel de compromiso de cada una de las zonas involucradas en la trayectoria

Fuente: Autor

Caso similar se presenta en los niveles de estrés (ver figura 4.15) del usuario, al iniciar la actividad el usuario mantiene los niveles al máximo, uno de los motivos puede ser que quiera realizar la tarea

lo mejor posible, generando el estrés al máximo, pero a medida que el usuario se familiariza con la teleoperación y no presencia muchos obstáculos, estos niveles disminuyen, en las zonas donde existen mas obstáculos, los niveles aumentan en algunos casos al máximo.

Estos resultados sirven para poder identificar, donde el usuario requiere más asistencia o mejor control sobre la teleoperacion.

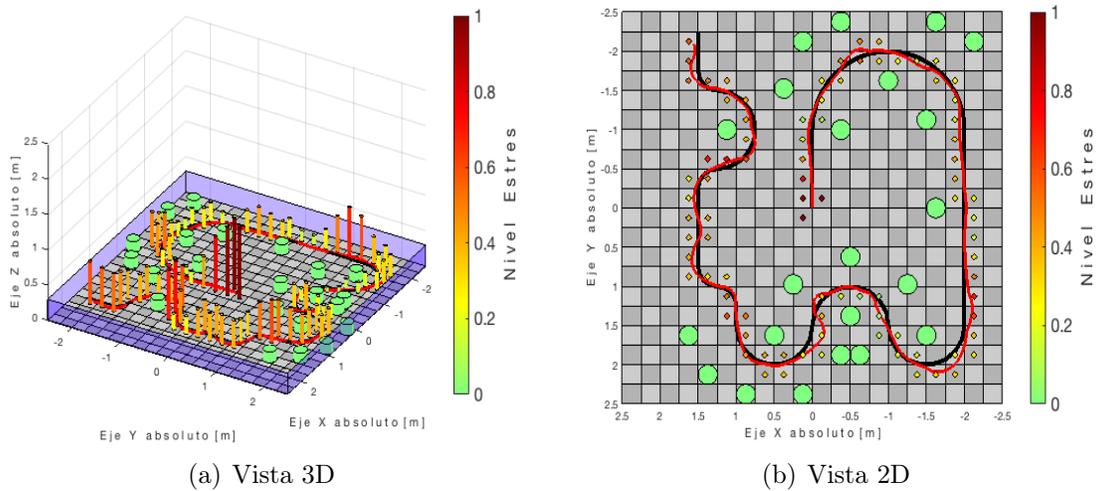


Figura 4.14: Nivel de estrés de cada una de las zonas involucradas en la trayectoria
Fuente: Autor

En los niveles de entusiasmo como se muestra en la figura ??, se inicia con los niveles al máximo, lo que indica que el usuario quiere realizar la actividad de la mejor manera, estos niveles disminuyen cuando el usuario se dirige en línea recta y no presencia muchos obstáculos, ahora, cuando el usuario teleopera el robot móvil sobre las zonas mas complejas, los niveles aumentan considerablemente terminando así con los niveles de entusiasmo al máximo.

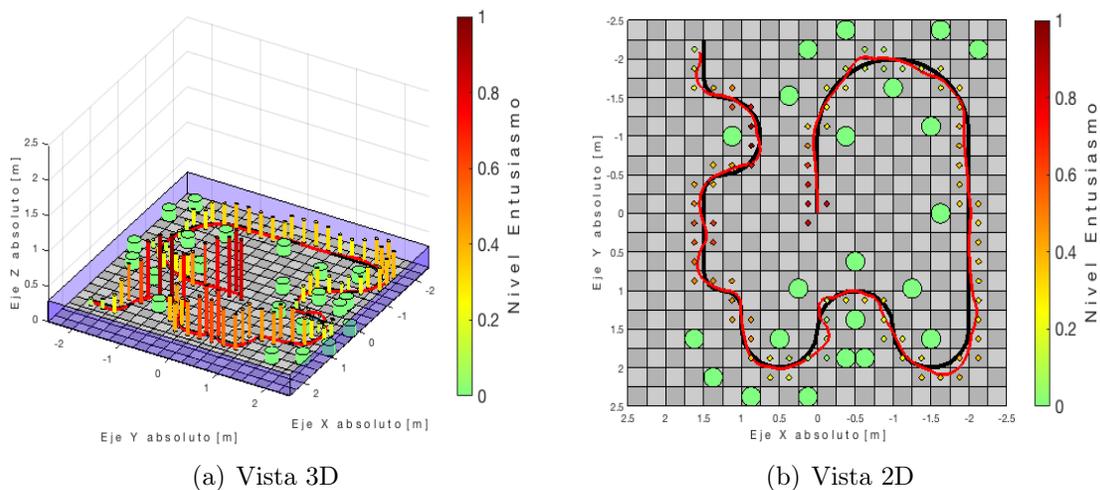


Figura 4.15: Nivel de entusiasmo de cada una de las zonas involucradas en la trayectoria
Fuente: Autor

En los niveles de relajación como se ilustra la figura 4.16, se observa que inicia con los niveles máximos, estos disminuyen a medida que va avanzando, cuando el usuario debe realizar maniobras complejas, estos niveles bajan a tal nivel que el usuario se encuentra relajado, cuando va finalizando la tarea, sus niveles suben, indicando que el usuario termina de realizar la tarea.

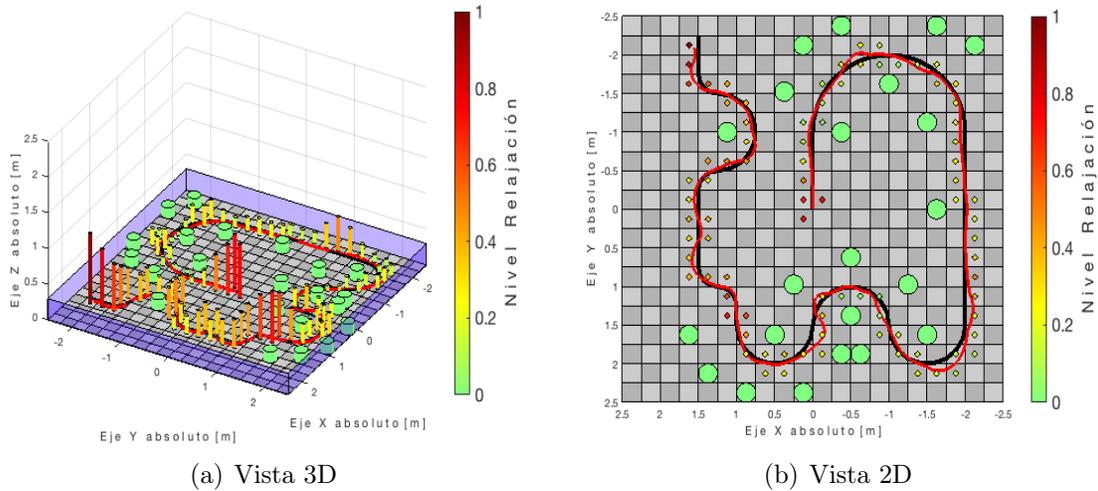


Figura 4.16: Nivel de relajación de cada una de las zonas involucradas en la trayectoria
Fuente: Autor

Para la emoción correspondiente al interés (ver figura 4.17), los niveles inician al máximo, pero van disminuyendo manteniéndose a un nivel medio, esto es debido a que cuando se inicia la teleoperación, muchos de los usuarios se sienten curiosos al hacer este tipo de actividad, lo que les ocasiona que estén interesados, pero a medida que se van familiarizando con el desarrollo de la actividad sus niveles van disminuyendo.

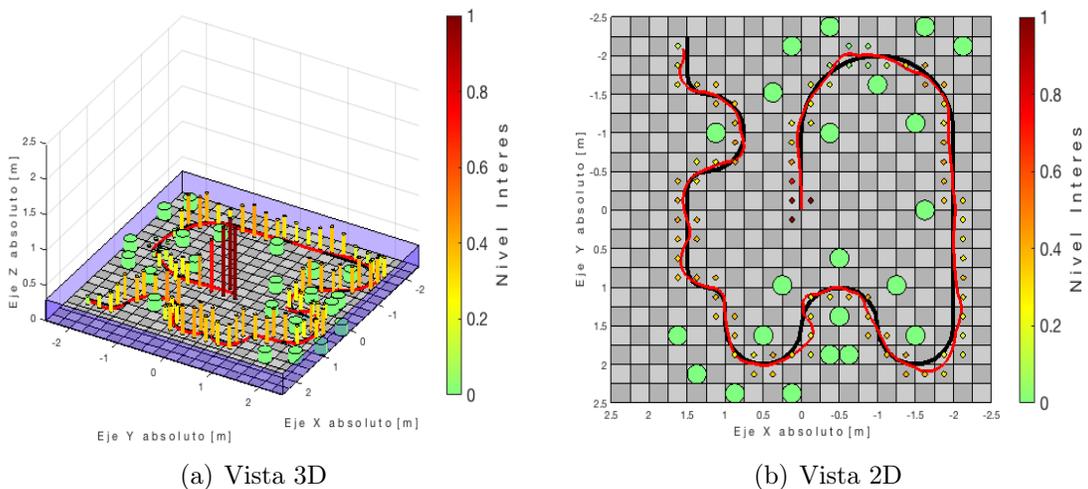


Figura 4.17: Nivel de interés de cada una de las zonas involucradas en la trayectoria
Fuente: Autor

Por último, la figura 4.18 nos presenta los niveles de concentración, los cuales inician con niveles medio, en zonas donde el usuario debe realizar una curva estos niveles aumentan de tal manera que llegan al máximo, caso contrario sucede en la línea recta, los niveles bajan considerablemente, caso contrario aumentan en presencia de los obstáculos.

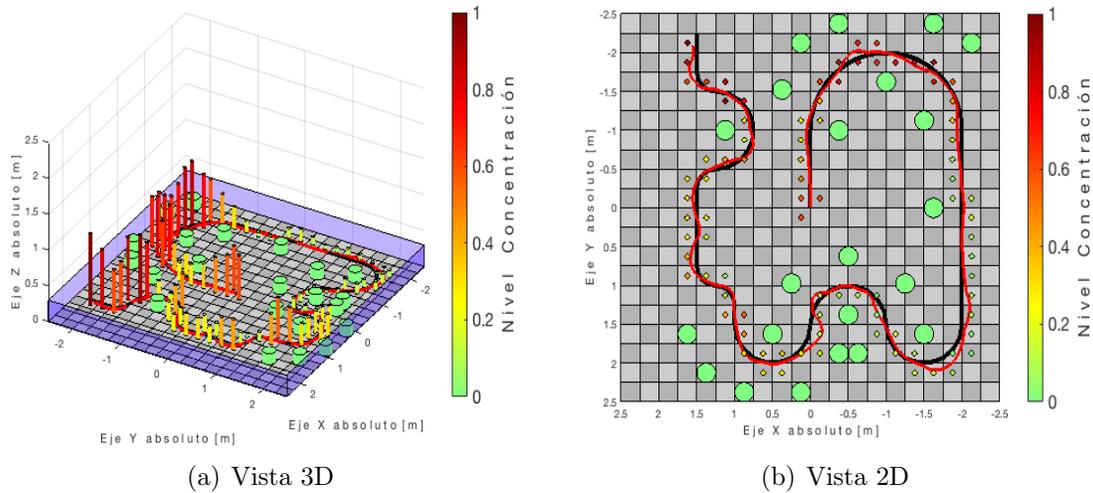


Figura 4.18: Nivel de concentración de cada una de las zonas involucradas en la trayectoria
Fuente: Autor

4.1.4. Implementación de mapas promedio de neuroseñales según su orientación

Para generar la ley de control adaptativa puede ser relevante no solo definir el nivel medio de las neuroseñales en zonas de trabajo, sino separar las mismas de acuerdo a la orientación que lleva el robot móvil.

Es decir que se pueden generar nueve (9) tipos de mapas diferentes para cada una de las neuroseñales. Esto se hace dado que el robot móvil se puede ubicar en una zona determinada, por ejemplo cerca de un obstáculo y obtener diferentes niveles una neuroseñal, esto se debe a que es diferente estar alejándose o acercándose a un obstáculo.

En las siguientes gráficas (ver figura 4.19) se ilustran por medio de imágenes los mapas que representan las neuroseñales para el entorno de trabajo según la orientación que lleva el robot móvil diferencial en cada una de la zonas.

Cabe recalcar que cada pixel corresponde a una zona del entorno de trabajo (cuadros demarcados en el suelo).

La gráfica central de cada figura corresponde a un mapa sin tener en cuenta la orientación, mientras que los que están a su alrededor representa una orientación en particular discretizada cada cuarenta y cinco (45)^o grados.

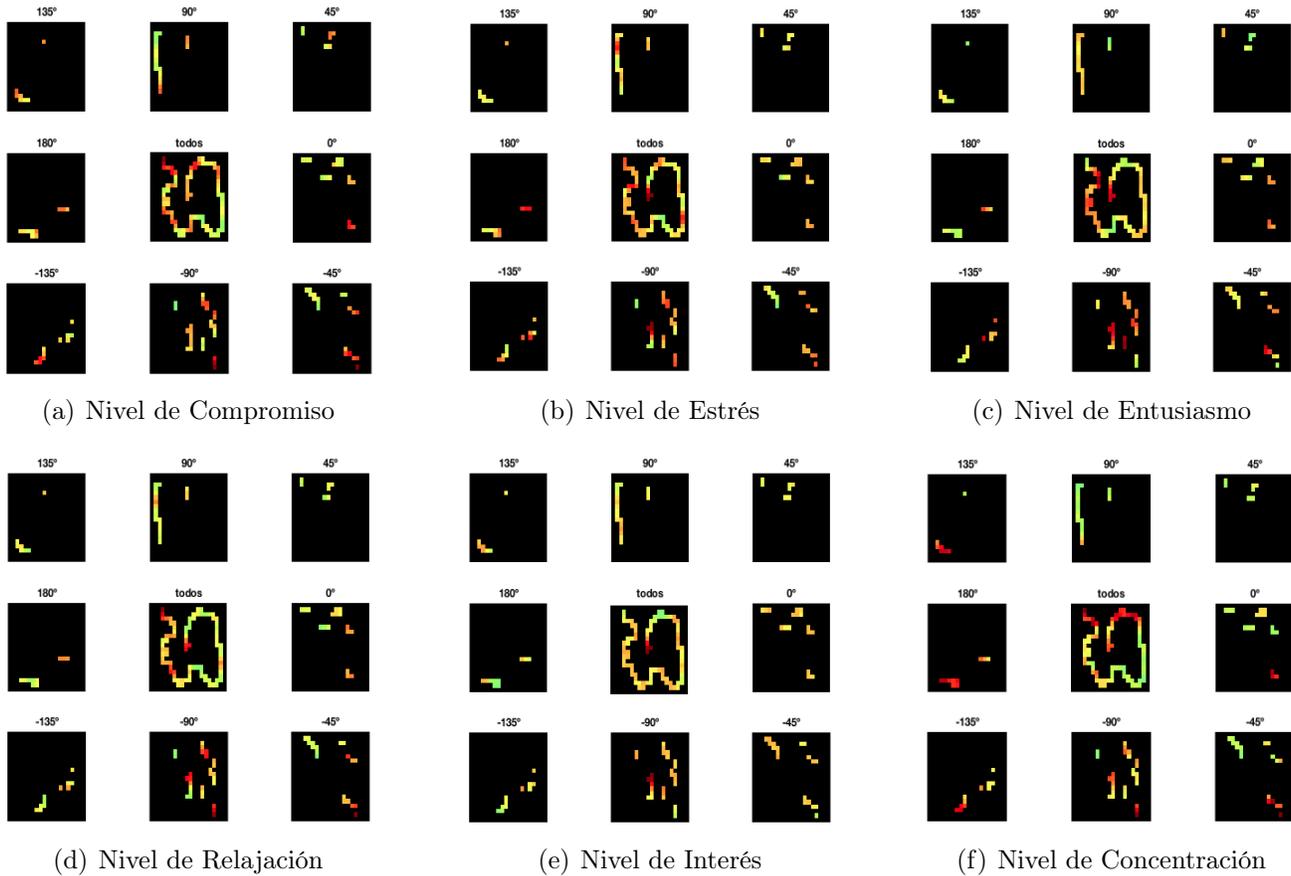


Figura 4.19: Mapas de los niveles emotivos para cada una de las zonas del entorno de trabajo
Fuente: Autor

4.1.5. Implementación del control adaptativo

A continuación, se presentan los resultados obtenidos en el desarrollo de un control humano robot para la teleoperación de un robot móvil a partir de la información obtenida de una interfaz cerebro computador como se ilustra en la figura 3.15.

Para la realización de este control, se optó por el tipo de escena de alta complejidad como se muestra en la figura 3.1(b), con el fin de determinar las zonas en las que el usuario presenta mayor dificultad a la hora de teleoperar el robot móvil.

Observando los resultados del control compartido y haciendo uso de la ecuación 3.7, se logró observar la importancia de agregar este control a la hora de realizar la teleoperación, como se muestran en la figura 4.20.

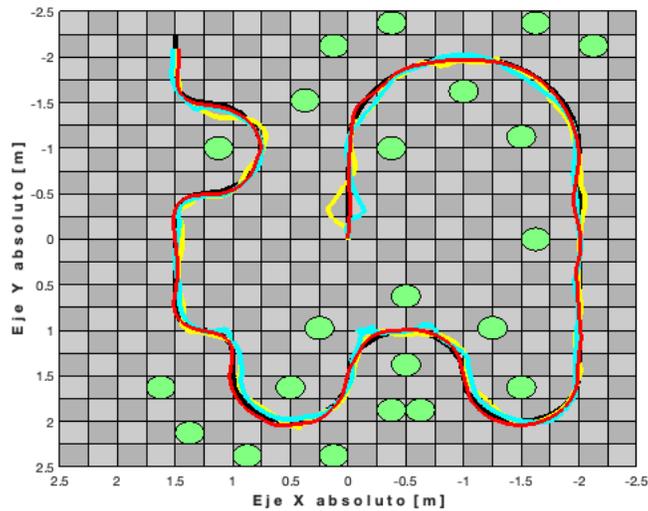


Figura 4.20: Trayectoria global realizada por el usuario
Fuente: Autor

En la figura 4.21 representa las trayectorias realizadas por el usuario en cada una de las asistencias propuestas para determinar la tarea, con el fin de analizar cual seria la asistencia de mayor precisión.

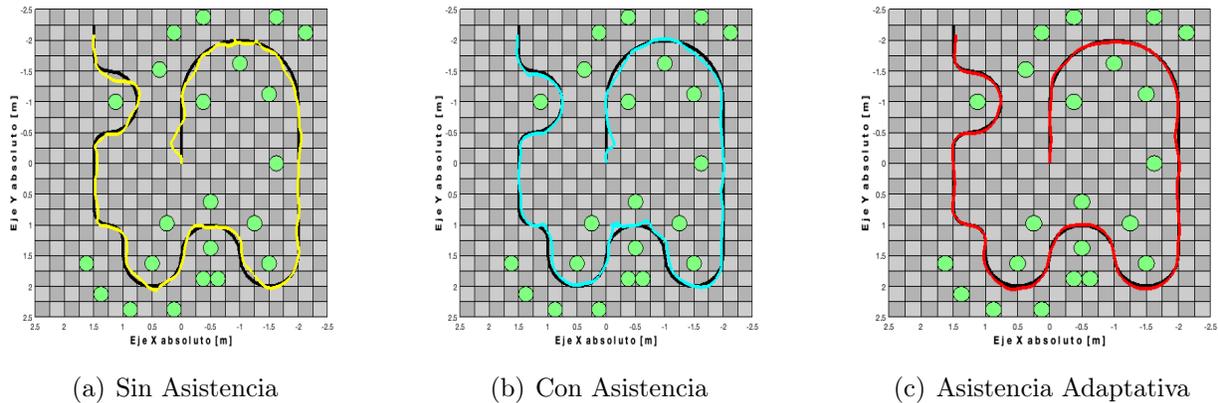


Figura 4.21: Trayectorias realizadas por el usuario
Fuente: Autor

En la primera prueba 4.21(a) los usuarios no obtuvieron ningún tipo de retroalimentación por parte de los obstáculos y de la trayectoria (path). La finalidad de esta prueba es analizar el tiempo y el error de los usuarios al realizar la tarea al teleoperar un robot móvil en el entorno de Vrep.

En la segunda prueba 4.21(b) los usuarios incluían retroalimentación de fuerza por parte de los obstáculos y de la trayectoria (path) relacionando al control autónomo. Para que el usuario percibiera mejor la retroalimentación, fue necesario establecer unos rangos de fuerza enviada hacia el dispositivo novint falcon.

En la tercera prueba 4.21(c) los usuarios obtenían retroalimentación de fuerza según la variable adaptativa, esta se obtuvo mediante un script en matlab procedente del algoritmo de control ?? el cual tenía como entrada los parámetros obtenidos al realizar la tarea predeterminada en el entorno

de Vrep (posición, orientación, velocidades del robot y el tiempo de ejecución) como también la estructura de datos como resultado al implementar el código ?? del registro de las neuroseñales, como resultado se obtiene la asistencia adaptativa.

El usuario realiza el control del robot móvil mediante el dispositivo háptico, así mismo sus niveles de emociones son tomadas mediante el dispositivo Emotiv Insight. Posteriormente, se asocian los datos capturados en la escena y los datos provenientes de las neuroseñales para luego exportarlos a matlab, para luego realizar el proceso de obtener los datos del controlador final.

Este controlador fue examinado nuevamente con los usuarios y se analizó sus emociones, error y tiempo en realizar la tarea predeterminada.

Uno de los objetivos principales es corroborar, que tanto cambia su estado al cumplir una determinada tarea con cada una de las asistencias planteadas, cual es la variación de cada una de las emociones con respecto a los diferentes eventos.

De esta manera se planteó una serie de datos que contienen dicha información obtenida a partir de las tareas realizadas mencionadas anteriormente, con el fin de analizar como era el comportamiento de las emociones de cada usuario.

	ASISTENCIAS	TIEMPO	ERROR Prom
		Segundos	Centímetros
Usuario 1	Sin Asistencia	172	4.387
	Con Asistencia	180	3.864
	Asistencia Adaptativa	201	3.411
Usuario 2	Sin Asistencia	246	3.827
	Con Asistencia	151	3.683
	Asistencia Adaptativa	129	1.741
Usuario 3	Sin Asistencia	461	6.114
	Con Asistencia	254	3.947
	Asistencia Adaptativa	149	5.579
Usuario 4	Sin Asistencia	146	5.874
	Con Asistencia	122	3.732
	Asistencia Adaptativa	121	3.276
Usuario 5	Sin Asistencia	167	5.188
	Con Asistencia	211	3.874
	Asistencia Adaptativa	129	3.805
Usuario 6	Sin Asistencia	141	6.075
	Con Asistencia	140	4.142
	Asistencia Adaptativa	138	2.987

(a) Usuarios 1 - 6

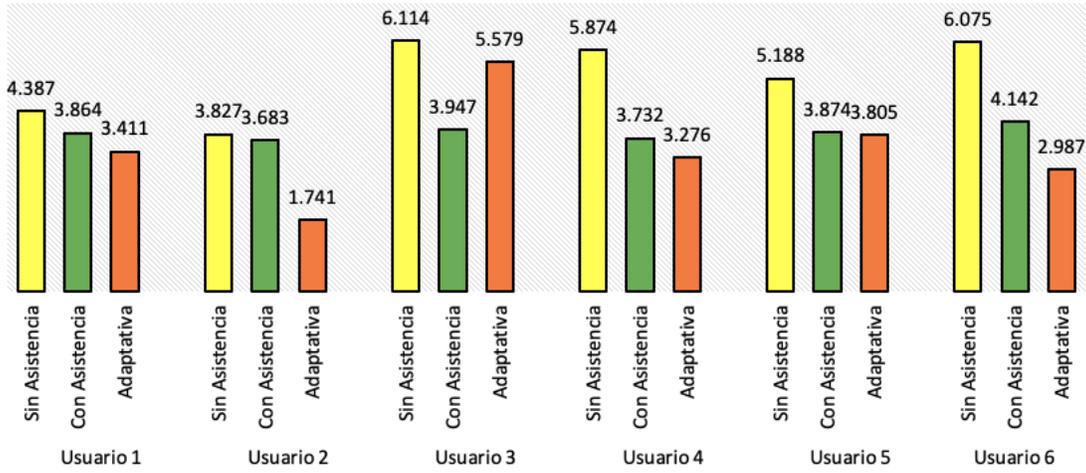
	ASISTENCIAS	TIEMPO	ERROR Prom
		Segundos	Centímetros
Usuario 7	Sin Asistencia	186	4.817
	Con Asistencia	147	2.930
	Asistencia Adaptativa	124	2.832
Usuario 8	Sin Asistencia	131	4.565
	Con Asistencia	126	2.770
	Asistencia Adaptativa	122	2.003
Usuario 9	Sin Asistencia	172	9.714
	Con Asistencia	164	5.408
	Asistencia Adaptativa	151	3.115
Usuario 10	Sin Asistencia	157	3.814
	Con Asistencia	137	3.284
	Asistencia Adaptativa	144	4.604
Usuario 11	Sin Asistencia	138	4.961
	Con Asistencia	129	3.715
	Asistencia Adaptativa	128	2.309
Usuario 12	Sin Asistencia	137	3.869
	Con Asistencia	133	3.205
	Asistencia Adaptativa	121	2.343

(b) Usuarios 7 - 12

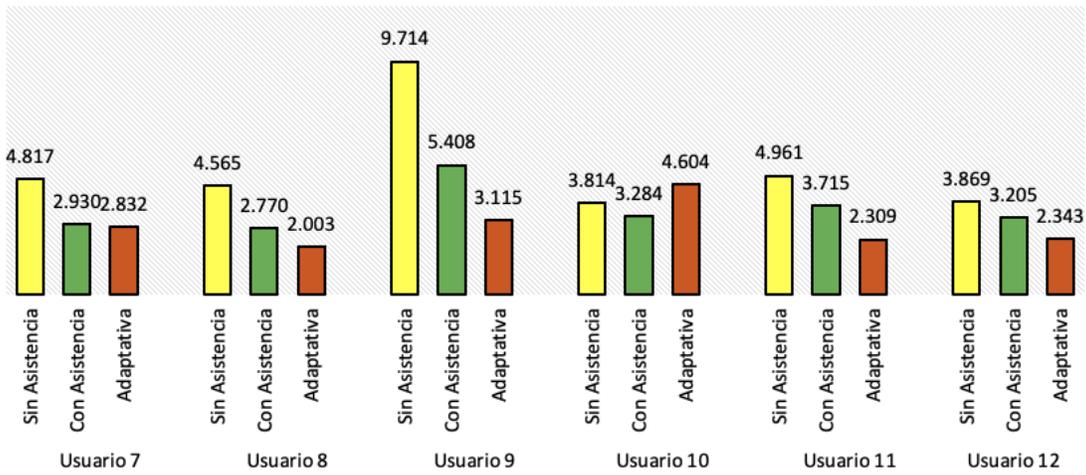
Figura 4.22: Compendio de los resultados estadísticos, tomando como base los datos a cada usuario

Fuente: Autor

En la figura 4.23, se observa los errores de posición de cada uno de los usuarios que ejecutaron la tarea preestablecida, con el fin de analizar y evidenciar los errores cuando el usuario ejecuta una tarea de teleoperación cuando se le aplica cada una de las asistencias definidas.



(a) Error de posición 1 - 6 usuarios

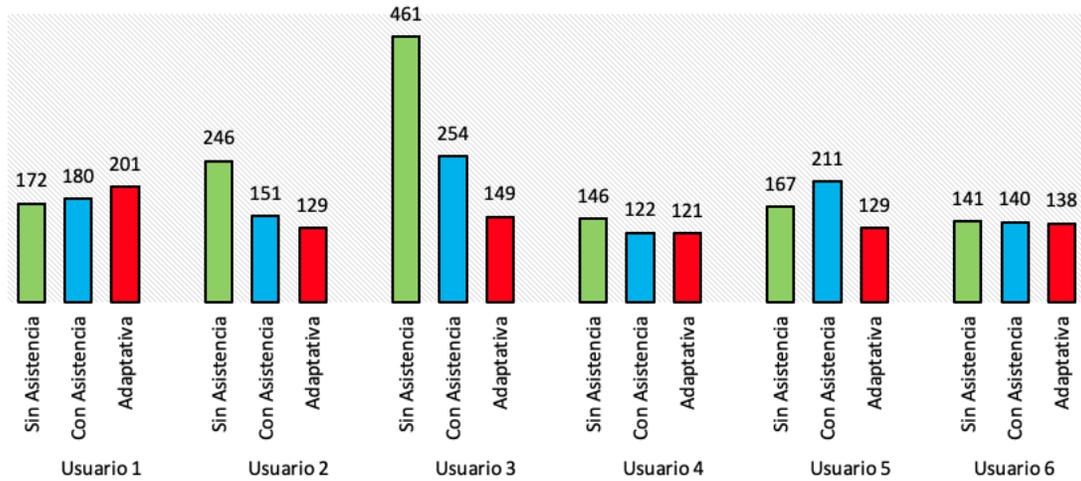


(b) Error de posición 7 - 12 usuarios

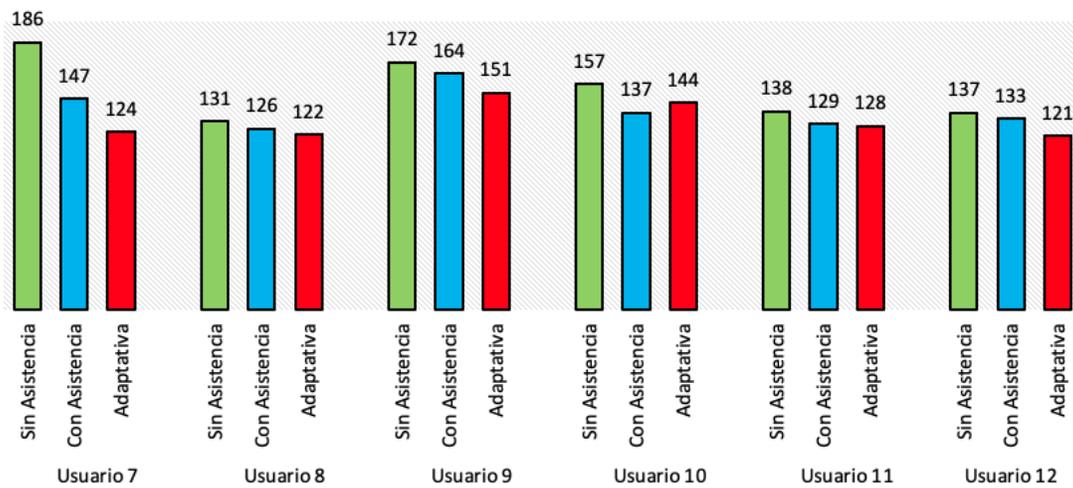
Figura 4.23: Error de posición del usuario ejecutando la tarea

Fuente: Autor

De igual manera, los tiempos de ejecución también fueron medidos, como se observa en la figura 4.24.



(a) Tiempo ejecutado de 1 - 6 usuarios



(b) Tiempo ejecutado de 7 - 12 usuarios

Figura 4.24: Tiempo de ejecución del usuario realizando la tarea
Fuente: Autor

Estableciendo un análisis con respecto a la información suministrada por cada uno de los usuarios, y observando que los tiempos de ejecución al cumplir la tarea asignada y considerando el error, disminuyen cuando el usuario aplica una **asistencia adaptativa**, corroborando nuevamente que el controlador que fue diseñado esta funcionando a la perfección obteniendo como resultado un valor porcentual del 83,33 % de efectividad.

Los resultados permitieron manifestar casos interesantes donde se elevaban los diferentes niveles de los usuarios en zonas donde no existían peligros evidentes o relacionados con colisiones, exponiendo que los mismos requerían ayuda del sistema de asistencia háptica para retomar una ruta u objetivo en particular.

Este tipo de acciones resultaría complejo de interpretar por los algoritmos tradicionales dado a que si no existen anomalías, no se generan acciones.

De igual forma se evidencio que existen muchos casos en los cuales los usuarios desean pasar intencionalmente muy cerca de los obstáculos con el fin de realizar acciones concretas, las cuales el

sistema de navegación autónoma interpretaría como potencialmente peligrosas.

En estos casos es necesario disminuir el nivel de asistencias háptica dándole mayor autoridad al control adaptativo o relevancia a las acciones a la hora de teleoperar un robot móvil diferencial, por lo general esto ocurre en tareas de alta exigencia o complejidad.

Capítulo 5

Conclusiones, logros y trabajos futuros

5.1. Conclusiones

- ❖ La inclusión de un control compartido es de gran ayuda a la hora de realizar tareas de teleoperación, por lo que en este trabajo de investigación se propuso un control compartido con el cual se permitió dar solución al guiado de un robot midiendo sus niveles de emociones y creando un controlador que ayude a mejorar la teleoperación
- ❖ Por medio del análisis de neuroseñales, se puede definir las zonas en un entorno de trabajo donde se requiere mayor o menor nivel de asistencia háptica para teleoperar un robot, con el fin de lograr un equilibrio en el control compartido entre las ordenes emitidas por el usuario y los comandos del sistema de navegación semiautónoma
- ❖ Basándose en las pruebas experimentales realizadas se puede concluir que la medición de las señales electroencefalograma EEG, sirve como medio de verificación para ubicar las zonas del entorno remoto del sistema de teleoperación, lo cual permite tomar las medidas pertinentes para alertar al mismo y evitar accidentes
- ❖ Al analizar las emociones en labores de teleoperación, ayudó a identificar en que zonas el usuario obtendría mayor dificultad para controlar el robot
- ❖ El control compartido evidencio ser una solución sinérgica entre la precisión generada por los sistemas de navegación autónoma y la resolución de problemas de alta complejidad por parte de teleoperadores humanos
- ❖ Al incluir retroalimentación de fuerzas a la hora de realizar la teleoperación en los diferentes experimentos, los niveles de emociones como estrés, concentración, entusiasmo e interés, aumentaron en comparación a cuando no se tenía la ayuda háptica.

Uno de los factores que influye en este resultado, es que los usuarios no estaban familiarizados con este tipo de experiencia, por ende, los motivaba a teleoperar al robot móvil de lo mejor posible

- ❖ Dividir las zonas de trabajo según la orientación del robot, fue un factor clave a la hora de diseñar el controlador, dado que los niveles de las neuroseñales pueden ser cambiantes según sea la ubicación del robot

- ❖ El uso de la plataforma del software de Vrep, fue de gran ayuda, dado que permitió analizar las emociones del usuario cuando realizaba tareas complejas, lo que fue fundamental en la tarea de control compartido propuesto
- ❖ De manera general, se logro evidenciar, que el error de la posición y el tiempo de ejecución disminuyeron de manera exitosa de acuerdo a las tareas ejecutadas por cada uno de los usuarios aplicando el controlador diseñado

5.2. Logros

- ❖ Publicación de autor del capítulo de libro resultado de investigación **“Influencia de las asistencias hápticas en la relajación de un teleoperador de un robot móvil”**. En coautoría con los investigadores y docentes Ph.D César Augusto Peña Cortés y Ph.D Javier Adolfo Corredor Camargo, es un resultado de investigación y forma parte del libro titulado: **“LA PRAXIS EN LA INGENIERÍA Y SU APOORTE AL DESARROLLO REGIONAL”**, de la Facultad de Ciencias Naturales e Ingeniería. El libro está registrado para la versión digital con el ISBN: 978-958-58321-9-0. Departamento de Investigación de la Fundación Universitaria de San Gil – UNISANGIL. Febrero 2020. Anexos ??
- ❖ Publicación del artículo: **“Haptic Assistance Evaluation for the Teleoperation of a Mobile Robot by Means of Stress Analysis”**. International Journal of Advanced Science and Technology. En coautoría con los investigadores y docentes Ph.D César Augusto Peña Cortés y Ph.D Javier Adolfo Corredor Camargo. ISSN: 2005-4238 IJAST. Copyright 2020 SERSC. Anexos ??
- ❖ II Congreso Latinoamericano de Automática y Robótica, **“Avances de un algoritmo de control compartido analizando los niveles de estrés a partir de una interfaz cerebro computador para la teleoperación de un robot móvil con asistencia háptica”**, Pontificia Universidad Javeriana, Santiago de Cali, Colombia, Octubre 30 a Noviembre 1 de 2019. Anexos ??

5.3. Trabajos futuros

- ❖ Como futuro campo de aplicación para los dispositivos adquiridos y analizados en este proyecto de investigación y en base a los resultados obtenidos, se plantean implementar la construcción de un robot móvil tipo diferencial, con el fin de validar todo lo expuesto en este trabajo de investigación
- ❖ Vincular todo el proceso de la investigación al sistema operativo de robótica (ROS), dado que esté posee muchas herramientas que facilitarían la implementación del trabajo, de igual manera, es un framework para el desarrollo de robots abierto y se ha convertido en un estándar en la academia
- ❖ Otro trabajo a futuro, es la integración de un manipulador a la plataforma móvil, esto debido a su gran versatilidad que puede llegar a obtener este tipo de combinación híbrida en tareas de alta complejidad y en espacios que requieran la movilidad de un producto. De igual manera analizar las emociones en este tipo de tareas es importante, ya que se puede observar el tipo de reacción producida en el usuario y así poder mejorar su experiencia al realizar la tarea

Bibliografía

- [1] Human haptic perception: Basics and applications, 2008.
- [2] Paulo Alves, Hugo Costelha, and Carlos Neves. Localization and navigation of a mobile robot in an office-like environment. In *2013 13th International Conference on Autonomous Robot Systems*, pages 1–6. IEEE, 2013.
- [3] Seyed Farokh Atashzar, Mahya Shahbazi, and Rajni V. Patel. Haptics-enabled interactive neurorehabilitation mechatronics: Classification, functionality, challenges and ongoing research. 57:1–19, 2019.
- [4] R. Bousseta, I. El Ouakouak, M. Gharbi, and F. Regragui. Eeg based brain computer interface for controlling a robot arm movement through thought. 39:129–135, 2018.
- [5] Frederick P Brooks Jr, Ming Ouh-Young, James J Batter, and P Jerome Kilpatrick. Project gropohaptic displays for scientific visualization. *ACM SIGGraph computer graphics*, 24(4):177–185, 1990.
- [6] X. Q. Chen, Y. Q. Chen, and J. G. Chase. Mobiles robots - past present and future.
- [7] Vivian Chu, Ian McMahon, Lorenzo Riano, Craig G. McDonald, Qin He, Jorge Martinez Perez-Tejada, Michael Arrigo, Trevor Darrell, and Katherine J. Kuchenbecker. Corrigendum to “robotic learning of haptic adjectives through physical interaction” [robot. auton. syst. 63 (p3) (2015) 279–292]. 83:349, 2016.
- [8] Edison Orlando Cobos Torres. *Traction Modeling and Control of a Differential Drive Mobile Robot to Avoid Wheel Slip*. PhD thesis, Oklahoma State University, 2013.
- [9] Luz Cueva, César Peña Cortés, Marisol Delgado, Surgei Villamizar, and Aldo Pardo. Registro de neuroseñales con una interfaz cerebro-computador para estimar el nivel estrés en un estudiante durante una clase. *INGE CUC*, 13:95–101, 07 2017.
- [10] Leonimer Flávio de Melo, João Mauricio Rosário, and Almiro Franco da Silveira Junior. Mobile robot indoor autonomous navigation with position estimation using rf signal triangulation. 04:20–35, 2013.
- [11] Florin Dragomir, Eugenia Mincă, Otilia Elena Dragomir, and Adrian Filipescu. Modelling and control of mechatronics lines served by complex autonomous systems. *Sensors*, 19(15):3266, 2019.
- [12] EMOTIV. Emotiv insight. <https://www.emotiv.com/insight/>.
- [13] J. M. García, A. Valero, and A. Bohórquez. Efecto de la suspensión en la estabilidad al vuelco y direccionamiento de robots moviéndose sobre discontinuidades de terreno. 17:201.

-
- [14] Ramón González, Francisco Rodríguez, and José Luis Guzmán. Robots móviles con orugas historia, modelado, localización y control. 12:3–12, 2015.
- [15] Alison Gopnik and Andrew Meltzoff. The development of categorization in the second year and its relation to other cognitive and linguistic developments. 58:1523, 1987.
- [16] Hiroo Iwata, Hiroaki Yano, Fumitaka Nakaizumi, and Ryo Kawamura. Project feelex, 2001.
- [17] Adrienne Kline and Jaydip Desai. Simulink based robotic hand control using emotiv eeg-headset, 2014.
- [18] D. Kuhner, L. D. J. Fiederer, J. Aldinger, F. Burget, M. Völker, R. T. Schirrmeister, C. Do, J. Boedecker, B. Nebel, T. Ball, and W. Burgard. A service assistant combining autonomous robotics, flexible goal formulation, and deep-learning-based brain–computer interfacing. 116:98–113, 2019.
- [19] Benjamin Kuipers, Edward A. Feigenbaum, Peter E. Hart, and Nils J. Nilsson. Shakey: From conception to history. 38:88–103.
- [20] Chun-Ling Lin, Fu-Zen Shaw, Kuu-Young Young, Chin-Teng Lin, and Tzyy-Ping Jung. Eeg correlates of haptic feedback in a visuomotor tracking task. 60:2258–2273, 2012.
- [21] Guanyang LIU, Xuda GENG, Lingzhi LIU, and Yan WANG. Haptic based teleoperation with master-slave motion mapping and haptic rendering for space exploration. 32:723–736, 2019.
- [22] Mufti Mahmud, David Hawellek, and Aleksander Valjamae. A brain-machine interface based on eeg: Extracted alpha waves applied to mobile robot, 2009.
- [23] D. J. McFarland and J. R. Wolpaw. Eeg-based brain–computer interfaces. 4:194–200, 2017.
- [24] Hirokazu Miura, Junki Kimura, Noriyuki Matsuda, Masato Soga, and Hirokazu Taki. Classification of haptic tasks based on electroencephalogram frequency analysis. 35:1270–1277, 2014.
- [25] Shady Mohamed, Sherif Haggag, Saeid Nahavandi, and Omar Haggag. Towards automated quality assessment measure for eeg signals. 237:281–290, 2017.
- [26] F. Mondada, E. Franzi, A. Guilgnard, and J. Nicoud. Khepera: The optimal tool for development in mobile robotics.
- [27] Raúl Arrabales Moreno, Agapito Ledezma Espino, and Araceli Sanchis de Miguel. A cognitive approach to multimodal attention. 3:53–63.
- [28] Hermann Munk. *Über die Functionen der Grosshirnrinde: gesammelte Mittheilungen mit Anmerkungen*. Hirschwald, 1890.
- [29] Polibits Instituto Politécnico Nacional. Robots moéviles: Evolucioén y estado del arte. <https://www.redalyc.org/pdf/4026/402640448003.pdf>.
- [30] Bahareh Nakisa, Mohammad Naim Rastgoo, Dian Tjondronegoro, and Vinod Chandran. Evolutionary computation algorithms for feature selection of eeg-based emotion recognition using mobile sensors. 93:143–155, 2018.

-
- [31] National Museum of American History. "tortoise" mobile robot. https://americanhistory.si.edu/collections/search/object/nmah_879329.
- [32] Cristobal Parra, Sergio Cebollada, Luis Paya, Mathew Holloway, and Oscar Reinoso. A novel method to estimate the position of a mobile robot in underfloor environments using rgb-d point clouds. 8:9084–9101, 2020.
- [33] Nicholas JM Patrick. *Design, construction, and testing of a fingertip tactile display for interaction with virtual and remote environments*. PhD thesis, Massachusetts Institute of Technology, 1990.
- [34] Xavier Perrin, Ricardo Chavarriaga, Francis Colas, Roland Siegwart, and José del R. Millán. Brain-coupled interaction for semi-autonomous navigation of an assistive robot. 58:1246–1255, 2010.
- [35] César Peña, Andres Vargas, and Javier Corredor. Haptic assistance evaluation for the teleoperation of a mobile robot by means of stress analysis. *International Journal of Advanced Science and Technology*, 29(04):11254–11264, Dec. 2020.
- [36] Pinterest. Robot shakey. <https://co.pinterest.com/pin/748582769282411440/>.
- [37] Maria Luisa Pinto Salamanca. Análisis e implementación de una interfaz háptica en entornos virtuales/analysis and implementation of a haptic interface in virtual environments. *Departamento de Ingeniería Eléctrica y Electrónica*, 2009.
- [38] Maria Luisa Pinto Salamanca. Análisis e implementación de una interfaz háptica en entornos virtuales/analysis and implementation of a haptic interface in virtual environments. *Departamento de Ingeniería Eléctrica y Electrónica*, 2009.
- [39] HF Ramirez, O Garcia-Bedoya, and O Aviles. Vali: Desarrollo y evolución de un robot para neutralizar explosivos. 2019.
- [40] Laura R. Ray, Devin C. Brande, and James H. Lever. Estimation of net traction for differential-steered wheeled robots. 46:75–87, 2009.
- [41] C Robotics. Coppeliarobotics. com. *Coppelia Robotics, [En línea]*. Available: <http://www.coppeliarobotics.com/>. [Último acceso: 26 mayo 2018].
- [42] Generation Robots. Pioneer p3-dx mobile robot. <https://www.generationrobots.com/en/402395-robot-mobile-pioneer-3-dx.html>.
- [43] RobotWatch. Robot terrestre ofro. <https://yorobot.wordpress.com/2007/06/13/ofro-un-robot-guardian/>.
- [44] U. Sanchez-Fraire, V. Parra-Vega, D. Martinez-Peon, G. Sepúlveda-Cervantes, A. Sanchez-Orta, and A. J. Muñoz-Vázquez. On the brain computer robot interface (bcric) to control robots. 48:154–159, 2015.
- [45] Giordano B. S. Seco, Günther J. L. Gerhardt, Alex A. Biazotti, André L. Molan, Suzana V. Schönwald, and José L. Rybarczyk-Filho. Eeg alpha rhythm detection on a portable device. 52:97–102, 2019.

-
- [46] Inc. Novint Technologies. Novint technologies developer's program. <http://home.novint.com/pdf/2007-03-04GDCDeveloperPresentation.pdf>, 2007.
- [47] Inc. Novint Technologies. Falcon overview. <http://home.novint.com/novint/novint.php>, 2008.
- [48] Andres Ubeda, Jose M. Azorin, Nicolas Garcia, Jose M. Sabater, and Carlos Perez. Brain-machine interface based on eeg mapping to control an assistive robotic arm, 2012.
- [49] Benjamin Vercellone, John Shelestak, Yaser Dhaher, and Robert Clements. Haptic interfaces for individuals with visual impairments. *G/ A/ M/ E Games as Art, Media, Entertainment*, 1(7), 2018.
- [50] Wikipedia. Khepera mobile robot. https://en.wikipedia.org/wiki/Khepera_mobile_robot.
- [51] Hong Wu, Shuang Liang, Wenlong Hang, Xiaolu Liu, Qiong Wang, Kup-Sze Choi, and Jing Qin. Evaluation of motor training performance in 3d virtual environment via combining brain-computer interface and haptic feedback. 107:256–261, 2017.
- [52] Pingjun Xia. Haptics for product design and manufacturing simulation. *IEEE transactions on haptics*, 9(3):358–375, 2016.
- [53] Laurence R Young, Charles M Oman, Daniel Merfeld, Douglas Watt, S Roy, C DeLuca, D Balkwill, J Christie, N Groleau, DK Jackson, et al. Spatial orientation and posture during and following weightlessness: human experiments on spacelab life sciences 1. *J Vestib Res*, 3(3):231–239, 1993.