

UNIVERSIDAD AUTÓNOMA DE MADRID

Escuela Politécnica Superior



Doble Grado en Ingeniería Informática y Matemáticas

## TRABAJO FIN DE GRADO

REALIDAD VIRTUAL PARA PERSONAS QUE TIENEN  
DISCAPACIDAD MOTORA

Cristina Kasner Tourné

Tutor: Francisco de Borja Rodriguez Ortiz

Enero 2017



# REALIDAD VIRTUAL PARA PERSONAS QUE TIENEN DISCAPACIDAD MOTORA

Autor: Cristina Kasner Tourné  
Tutor: Francisco de Borja Rodriguez Ortiz

Escuela Politécnica Superior  
Universidad Autónoma de Madrid

Enero 2017



# RESUMEN

**Resumen** La fusión de la Realidad Virtual con la robótica supone una apertura a infinitas posibilidades.

Ambas tecnologías están en continuo desarrollo, de hecho la realidad virtual empezó a darse a conocer muy recientemente, a pesar de que su nacimiento data sobre (año).

En los últimos años estamos viendo cómo la Realidad Virtual va haciéndose un hueco en las tecnologías que usamos habitualmente. Quizá el uso más comercial que se le está dando es en lo referente al mundo de los videojuegos. Sin embargo se puede aplicar a muchos otros campos , en concreto en el campo de la salud , donde se están obteniendo muy buenos resultados (en cirugía, rehabilitación, etc).

Este trabajo pretende explorar el uso de la Realidad Virtual y la robótica como herramienta enfocada a personas con movilidad reducida , ayudarles a ser más independientes y dándoles la posibilidad de transportarse de forma virtual por entornos reales.

Para conseguir este objetivo se ha construido un sistema que consta de: un robot y las Oculus Rift (gafas de realidad virtual). El robot se controla desde las Oculus Rift, de forma que si el usuario lleva puestas las gafas, podrá controlar el robot con movimientos de cabeza. El usuario verá a través de las gafas todo lo que vea el robot, actuando éste último como extensión de la vista del usuario.

Al utilizar las Oculus Rift dejamos abierta la posibilidad de ver, no solo el entorno en el que se mueve el robot, sino un entorno virtual creado por el propio usuario. Esta parte se deja como tema de estudio para trabajos futuros.

Además hemos diseñado el proyecto de forma que la conexión entre el robot y las gafas sea a través de red inalámbrica, lo que le da al robot una libertad de movimiento fundamental para el objetivo que se persigue.

A lo largo del desarrollo del trabajo han surgido varias complicaciones, la gran mayoría referentes a la utilización de los sensores de las Oculus y de la librería

de las mismas, que aún no está perfectamente adaptada a la versión DK2 de las gafas.

Esto se ha ido solucionando calibrando el movimiento de los servo motores mediante pruebas experimentales.

A pesar de esto el proyecto se ha terminado con éxito, dejando abierta una línea de investigación y mejora sobre la que trabajar.

# ABSTRACT

## **Abstract**

# ÍNDICE GENERAL

<b>Índice general</b>	<b>IV</b>
<b>Índice de tablas</b>	<b>VI</b>
<b>Índice de figuras</b>	<b>VII</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación del proyecto . . . . .	1
1.2 Objetivos . . . . .	2
1.2.1 Streaming . . . . .	2
1.2.2 Movimiento de la cámara . . . . .	2
1.2.3 Visualización del entorno . . . . .	3
1.3 Estructura del documento . . . . .	4
<b>2 Estado del arte</b>	<b>5</b>
2.1 Evolución histórica de la realidad virtual . . . . .	5
2.2 Realidad virtual y salud . . . . .	7
2.3 Robótica y realidad virtual. . . . .	8
<b>3 Análisis</b>	<b>11</b>
3.1 Requisitos Funcionales . . . . .	12
3.1.1 Construcción del Robot . . . . .	12
3.1.2 Streaming . . . . .	12
3.1.3 Control del Robot . . . . .	12
3.2 Requisitos no funcionales . . . . .	13
<b>4 Diseño</b>	<b>15</b>
4.1 Diseño del robot . . . . .	15
4.1.1 Raspberry Pi 3 model B . . . . .	16
4.1.2 Servo motores . . . . .	17
4.1.3 Cámara . . . . .	18
4.2 Router . . . . .	19
4.3 Oculus Rift . . . . .	19
4.3.1 Detección de la orientación . . . . .	20
4.3.2 Sensación de inmersión _ 3D . . . . .	21

4.4	Diseño de conexiones . . . . .	21
<b>5</b>	<b>Desarrollo e implementación</b>	<b>23</b>
5.1	Construcción del Robot . . . . .	23
5.1.1	Soporte de la cámara . . . . .	24
5.2	Streaming . . . . .	25
5.2.1	Raspberry Pi3 . . . . .	25
5.2.2	Ordenador . . . . .	27
5.3	Control del Robot . . . . .	29
5.3.1	Recogida y procesamiento de la información que mandan las Oculus al ordenador . . . . .	29
5.3.2	Movimiento de los servo motores . . . . .	32
	<b>Control de los motores de las ruedas del robot</b> . . . . .	33
	<b>Control del motor de la cámara del robot</b> . . . . .	33
	<b>Instrucciones de mando</b> . . . . .	34
<b>6</b>	<b>Pruebas y Resultados</b>	<b>37</b>
6.1	Fases del ciclo cerrado . . . . .	38
6.2	Desarrollo de las pruebas . . . . .	38
6.2.1	Primer ciclo . . . . .	38
6.2.2	Segundo ciclo . . . . .	39
6.2.3	Tercer ciclo . . . . .	40
<b>7</b>	<b>Conclusiones y trabajo futuro</b>	<b>43</b>
7.1	Conclusiones . . . . .	43
7.1.1	Tecnologías aprendidas . . . . .	44
7.2	Trabajo futuro . . . . .	44
	<b>Bibliografía</b>	<b>45</b>

# ÍNDICE DE TABLAS

# ÍNDICE DE FIGURAS

1.1	Resumen de los objetivos del proyecto . . . . .	3
2.1	Sword of Damocles. Imagen obtenida de <i>A head-mounted three dimensional display</i> [7] . . . . .	6
2.2	Rehabilitación usando herramientas de realidad virtual. Imagen obtenida de <i>Water-Friendly Virtual Reality Pain Control During Wound Care</i> [1] . . . . .	8
2.3	Da Vinci Surgical System . . . . .	9
4.1	Raspberry Pi 3 model B . . . . .	17
4.2	Esquema conexiones servos , cámara, Raspberry Pi 3 . . . . .	18
4.3	Tabla DHCP del router con las parejas MAC-IP fijas . . . . .	19
4.4	Archivo de configuración de red Raspberry Pi 3 . . . . .	19
4.5	Ejes de giro . . . . .	20
4.6	Esquema conexiones . . . . .	21
5.1	Diseño 3D de la pinza de agarre en Blender . . . . .	24
5.2	Imagen de la configuración del Cura para enviar la Pieza de agarre a la impresora 3D . . . . .	25
5.3	Uso de CPU del programa Mjpeg-Streamer por el que se hace el streaming de video . . . . .	26
5.4	Ejecución del comando para iniciar el streaming . . . . .	26
5.5	Página html donde se recibe el streaming a tiempo real desde la Raspberry . . . . .	26
5.6	Configuración y resultado de la reproducción de video en VLC . . . . .	27
5.7	Posición de las Oculus respecto al eje X una vez procesados los datos de HeadPose.ThePose.Orientation . . . . .	30
5.8	Relación entre el PW y la posición de los servo motores . . . . .	32
5.9	Posibles movimientos del robot según el movimiento de las Oculus Rift . . . . .	34
6.1	Fases del ciclo cerrado de pruebas que se han realizado para valorar la funcionalidad del dispositivo. . . . .	37
6.2	Promedio del tiempo que tardan los usuarios en realizar el circuito. . . . .	41

6.3 Promedio del tiempo que tardan los usuarios en realizar el circuito. . . 42

# INTRODUCCIÓN

## 1.1 Motivación del proyecto

Este proyecto busca explorar las posibilidades que nos ofrece combinar la robótica con la realidad virtual con el objetivo de incrementar la autonomía e independencia de personas con discapacidad motora.

Existen muchos proyectos anteriores a este en los que se combinan estas dos técnicas con éxito, como ya se explicará en el estado del arte.

El campo de la **robótica** lleva muchos años en desarrollo y mejora exponencialmente a lo largo del tiempo. Está inmerso en nuestro día a día y no paramos de sorprendernos con nuevos avances como las impresoras 3D, los drones o los brazos mecánicos.

Por otra parte puede parecer que la **realidad virtual** es una tecnología muy reciente pero lo cierto es que lleva mucho tiempo estudiándose. En 1968 Ivan Sutherlan creó el primer casco de realidad virtual pero, como se explicará más adelante, aún no tenían las herramientas necesarias para desarrollar proyectos de realidad virtual de calidad.

Ahora nos encontramos en un momento en el que ya podemos empezar a disfrutar de la realidad virtual. Tenemos las CardBoard, las Oculus Rift, etc, a nuestro alcance. El uso más generalizado que se está dando a estas herramientas es en el mundo de los videojuegos.

Viendo el potencial que tiene unir ambas tecnologías, resulta muy interesante explorar nuevas aplicaciones, en esta ocasión orientadas a mejorar la calidad de vida de un amplio sector de la sociedad.

La realidad virtual nos ofrece la posibilidad de **crear un entorno** en el cual, con el software adecuado, el usuario puede **interactuar** con otros usuarios, participar en actividades de **ocio y entretenimiento**, etc, sin necesidad de moverse de casa. Además puede resultar muy interesante utilizar esta tecnología para **reproducir virtualmente entornos reales** a los que el usuario tenga difícil acceso.

## 1. INTRODUCCIÓN

---

El primer paso para conseguir esto es introducir imágenes del entorno en tiempo real dentro del mundo virtual. Para ello debemos disponer de un mecanismo móvil(p.ejemplo: un robot) que sea capaz de capturar fotogramas y mandárselas al usuario. Para que el usuario pueda ver dichas imágenes se utilizarán las Oculus Rift, unas potentes gafas de realidad virtual que son capaces de crear una **sensación de inmersión** total en cualquier entorno.

Ya que este trabajo está orientado a personas con movilidad reducida sería ideal que se pudiera controlar el movimiento del robot y de la cámara solo con el movimiento de la cabeza. En un futuro, no sería difícil adaptar este control a través de mandos.

¿Porqué utilizar las Oculus Rift? Al ser las Oculus gafas de realidad virtual, si consiguiéramos una correcta conexión entre las gafas y el robot, se nos abrirían muchísimas posibilidades. Se podría modificar el entorno según las necesidades del usuario, integrar realidad aumentada para controlar una casa domotizada, incluir en el mundo virtual más entornos, a parte del real, donde el usuario pueda interactuar, realizar actividades de ocio, etc. De esta forma daríamos al usuario más independencia y le abriríamos un mundo al que hasta ahora, tiene difícil acceso.

### 1.2 Objetivos

El objetivo de este proyecto consiste en diseñar un dispositivo que actúe como extensión de la vista del usuario a través de una webcam implantada en el robot. El usuario deberá ser capaz de controlar la cámara con tan solo el movimiento de la cabeza, ya que el proyecto está orientado a personas con movilidad reducida.

Además, las imágenes se reproducirán a través de unas gafas de realidad virtual: las Oculus Rift. Esto nos permitiría, en un proyecto futuro, añadir realidad virtual o aumentada a las imágenes que captura la cámara.

Para ello definimos pequeños objetivos que debemos ir implementando para conseguir la meta final.

#### 1.2.1 Streaming

Como se ha explicado anteriormente, debemos ser capaces de ver en las Oculus Rift a tiempo real todo lo que capta la cámara.

Dado que la comunicación con las Oculus Rift debe ser a través del PC al que estén conectadas, el streaming deberá **transmitirse desde la cámara hasta el PC**.

Por lo tanto uno de los primeros objetivos es implementar un método de comunicación a través del cual se puedan conectar la cámara y el PC. Además se deberá implementar un software que capture la salida de la webcam y transmita el vídeo a tiempo real. Por último se necesita un software que recoja el video en el PC.

#### 1.2.2 Movimiento de la cámara

Otro de los objetivos básicos es que el usuario sea capaz de controlar la dirección y posición de la cámara con el movimiento de la cabeza.

Con este objetivo de van a definir dos fases del movimiento:

- **Fase de exploración:**

Durante la exploración la cámara no se mueve de sitio, tan solo explora el entorno.

Los movimientos que se deberán implementar para esta fase serán:

- Movimiento vertical : El usuario podrá mirar hacia arriba y hacia abajo.
- Movimiento horizontal : La cámara deberá ser capaz de rotar sobre si misma permitiendo al usuario mirar hacia ambos lados.

- **Fase de transporte:**

El usuario podrá llevar la cámara a cualquier punto de la habitación moviendo la cabeza.

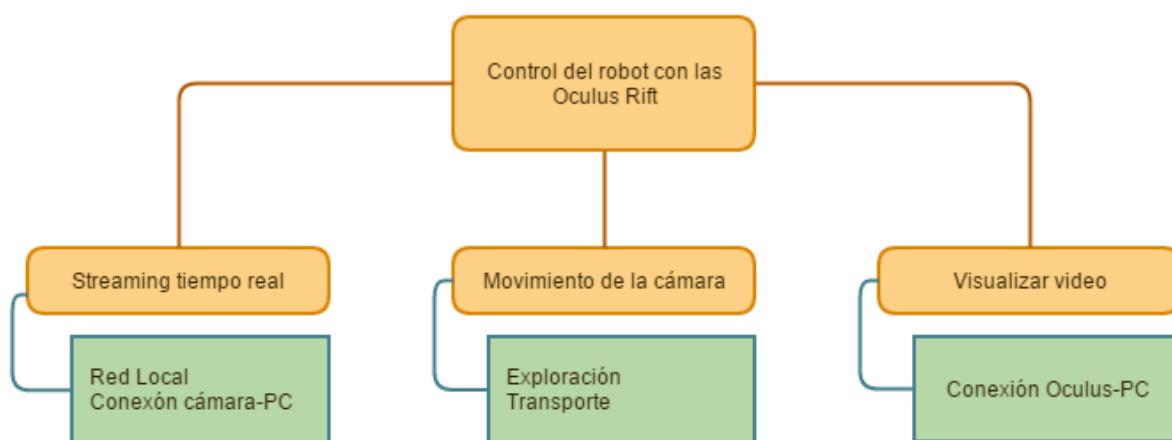
Para este objetivo, el diseño tendrá un papel esencial ya que el control del robot debe ser sencillo e intuitivo. Se intentará crear un sistema de control con la cabeza que no sea complicado para el usuario.

### 1.2.3 Visualización del entorno

El último de los objetivos que es necesario desarrollar para el proyecto es conseguir visualizar en las Oculus Rift, el entorno en el que se encuentra la cámara.

Para ello necesitaremos un software que, en tiempo real, recoja el vídeo del PC y lo reproduzca en las gafas.

A continuación se añade un esquema a modo de resumen de los objetivos ya explicados.



**Figura 1.1:** Resumen de los objetivos del proyecto

### 1.3 Estructura del documento

El objetivo de este documento es explicar el trabajo realizado y los resultados obtenidos.

En el siguiente capítulo se explica el estado del arte, hablando brevemente de la realidad virtual y de algunos proyectos previos que utilizaban esta tecnología. Como se verá más adelante el capítulo se centra especialmente en proyectos relativos a salud y robótica.

Luego se verá el análisis, diseño y desarrollo del proyecto.

En el análisis se presentan los requisitos que debe cumplir nuestro sistema. Posteriormente se hace un estudio de diseño de los componentes del trabajo (robot, conexiones entre dispositivos...) y por último se explicará como se ha desarrollado el proyecto.

Los resultados y las pruebas realizadas para ver si el proyecto cumplía con el objetivo se verán en el capítulo "Pruebas y resultados" de este documento.

Para finalizar expondremos brevemente las conclusiones y el trabajo futuro que se puede realizar en base a este proyecto.

# ESTADO DEL ARTE

En este capítulo se contará brevemente la historia de la realidad virtual. También se presentarán diferentes proyectos que ha utilizado realidad virtual y robótica y cuales fueron sus resultados.

A pesar cara más visible de esta tecnología es el uso que se le da en los videojuegos, no es su única aplicación.

Los proyectos que se van a explicar utilizan la realidad virtual en el campo de la salud, sobre todo aplicada a rehabilitaciones. Esto es muy interesante para este trabajo ya que muestra cómo usuarios con alguna dificultad motora responden bien a entornos virtuales.

También se hablará de cómo se están integrando actualmente las dos tecnologías que utilizaremos en este trabajo: la robótica y la realidad virtual.

De esta forma se dará una fotografía general del estado y las aplicaciones actuales de ambas tecnologías, dando especial importancia a aquellos proyectos con un enfoque similar al de nuestro trabajo.

## 2.1 Evolución histórica de la realidad virtual

La realidad virtual comenzó siendo ciencia ficción y poco a poco está ganando terreno en la vida cotidiana, ya sea en el mundo de los videojuegos, la medicina o la aviación, entre otros campos.

A pesar de ser una tecnología de la que se ha empezado a hablar hace relativamente poco, los primeros proyectos que utilizaban realidad virtual datan de los años 60.

Uno de los acontecimientos más significativos del inicio de la realidad virtual es la creación de **“The Sword of Damocles”**, un casco de realidad virtual (HMD-head-mounted display) creado por Ivan Sutherland en 1968 [7].

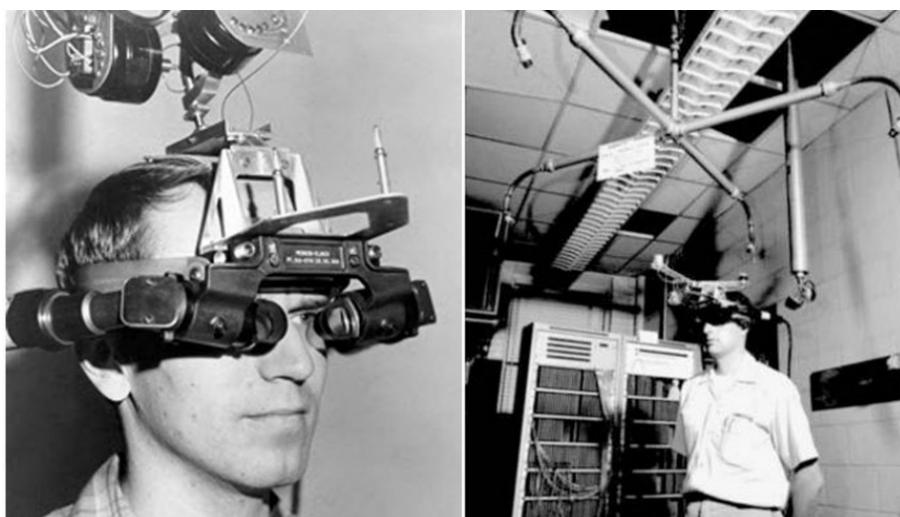
The Sword of Damocles permitía ver las imágenes tridimensionales generadas por el ordenador superpuestas ópticamente, a través de un prisma, al entorno real que rodeaba al usuario.

El sistema estaba montado sobre un brazo mecánico suspendido del techo que también servía para detectar, mediante el uso de sensores, la posición y la orientación de la cabeza del usuario. De este modo, cuando el usuario se movía, los objetos creados por el ordenador daban la impresión de encontrarse en una posición estable dentro del entorno real.

Es lógico preguntarse cómo ha tardado tanto en llegar al mercado la realidad virtual si ya en 1968 consiguieron sacar adelante este proyecto.

La respuesta es que **la tecnología de los años 60 aún no estaba preparada** para desarrollar herramientas de realidad virtual.

Para conseguir capturar la posición y el movimiento de la cabeza del usuario, se requería un brazo de **grandes dimensiones**. Además la realidad virtual necesita una **capacidad de cómputo muy potente**. Problemas tales como "hidden line" solo podía ser resueltos con un ordenador que tenía la NASA en Houston, como especifica Sutherland en su trabajo *A head-mounted three dimensional display* [7].



**Figura 2.1:** Sword of Damocles. Imagen obtenida de *A head-mounted three dimensional display* [7]

Al mismo tiempo Thomas A. Furness III empezó a introducir herramientas de **realidad virtual en entrenamientos de vuelo** [3], con el objetivo de que estos fueran mucho más seguros para los pilotos. A pesar de que fue una gran aportación en este campo, la escasez de recursos y capacidad de cómputo también retrasaron mucho el desarrollo de estas herramientas.

Hoy en día tenemos la capacidad necesaria para desarrollar estas herramientas por lo que se siguen utilizando y mejorando ya que, no solo es más segura, sino que además es más barata.

Durante las últimas décadas del s.XX, se hicieron algunos proyectos de realidad virtual , pero no tuvieron mucho éxito debido a su alto coste.

Con el desarrollo exponencial que ha vivido la tecnología las últimas décadas, la realidad virtual ha ido haciéndose un hueco en la industria tecnológica. El empujón definitivo vino con la creación de las **Oculus Rift**, en el año 2010.

**Palmer Luckey** es el fundador de Oculus VR (una compañía de realidad virtual) y el diseñador de la primera versión de las Oculus Rift.

Oculus Rift es un casco de realidad virtual (también nos podemos referir a ellas como gafas de realidad virtual) mucho más manejable que "The Sword of Damocles", con mucha más calidad y con un coste asequible. Uno de los grandes logros de Luckey con las Oculus Rift es que consiguió un ángulo de visión de 90°, algo nunca visto hasta el momento.

Tras el lanzamiento de las Oculus Rift, el mercado comenzó a interesarse por proyectos relacionados con la realidad virtual y a financiarlos. Gracias a esto la tecnología ha podido desarrollarse y hoy podemos disfrutar de una realidad virtual con mucha más calidad y **sensación casi total de inmersión** en entornos virtuales.

## 2.2 Realidad virtual y salud

La realidad virtual ofrece grandes posibilidades en el campo de la medicina. Permite estudiar cómo responde el paciente a distintos estímulos o cómo interactúa con diferentes entornos de forma bastante realista.

Podemos encontrar un ejemplo de esto en un proyecto que se realizó para evaluar el deterioro cognitivo en personas que sufren esclerosis múltiple. Para que dicha evaluación sea precisa se requiere de una gran cantidad de datos sobre la velocidad de procesamiento de la información, atención, etc. Esto se conseguía haciendo múltiples test que no siempre eran fieles a la realidad. Ahora con la realidad virtual se consiguen crear entornos en los que el usuario tiene tal sensación de inmersión, que su **comportamiento es muy similar al que tendría en el entorno real**. Esto permite hacer un estudio mucho más amplio y realista de los estímulos del paciente y obtener resultados más completos.<sup>[4]</sup>

Otra aplicación interesante es el uso de realidad virtual en programas de rehabilitación. Se realizó un estudio a personas con hemiparesia, es decir, pacientes que han perdido parcial o totalmente la capacidad motora de las extremidades de uno de los lados de su cuerpo. El proyecto consistía en un software que mostraba por pantalla el movimiento de los pies del usuario y le mandaba ejercicios para mejorar su locomoción. Los resultados de este estudio mostraron que la herramienta de realidad virtual era **efectiva para la rehabilitación de las capacidades motoras** de los pacientes y además es especialmente útil para aquellos que tienen dificultades para trasladarse hasta el hospital.<sup>[5]</sup>

Los dos proyectos que se han presentado en este capítulo son ejemplos de cómo la realidad virtual ha contribuido a mejorar técnicas que ya se usaban anteriormente.

El siguiente proyecto también está enfocado a mejorar el tratamiento de pacientes pero, en esta ocasión, utiliza una característica propia de la realidad virtual, que es la inmersión en un entorno distinto al que se encuentra el paciente.

Esta capacidad que tiene la realidad virtual es muy útil para personas que se encuentran en un proceso de rehabilitación doloroso.

El experimento que se presenta a continuación se realizó con un paciente de 40 años con un 19 % de su cuerpo quemado. El trabajo consistía en utilizar un entorno virtual como paliativo durante los ejercicios de rehabilitación. Se le proporcionó al paciente un casco de realidad virtual y se creó un entorno en el que el paciente pudiera "resguardarse" del dolor ocasionado por la rehabilitación.



**Figura 2.2:** Rehabilitación usando herramientas de realidad virtual. Imagen obtenida de *Water-Friendly Virtual Reality Pain Control During Wound Care* [1]

Los resultados fueron muy positivos. El usuario puntuó la sensación de dolor en un 7 sobre 10 cuando no usaba realidad virtual y en un 2 sobre 10 utilizando realidad virtual.[1]

Estos experimentos suponen una prueba de la increíble **capacidad de inmersión** que nos proporciona la realidad virtual, aspecto muy importante para nuestro trabajo.

### 2.3 Robótica y realidad virtual.

A continuación se presentan algunos proyectos de robótica que utilizan técnicas de realidad virtual. Es interesante ver, en los resultados de estos trabajos, cómo ambas tecnologías se complementan de forma muy beneficiosa.

Puesto que al final de este trabajo lo que se quiere conseguir es controlar un robot a través de una interfaz que utiliza realidad virtual, es interesante estudiar cómo ha resultado esto en proyectos parecidos.

En 1997 , el IMG (Intelligent Mechanisms Group) de la NASA y el instituto de robótica de la Carnegie Mellon University desarrollaron las **interfaces de control** necesarias para teledirigir a Nomad, un robot diseñado para la exploración de planetas. [6]

Hasta el momento este control era altamente complicado, ya que requería del uso de muchos comandos poco intuitivos. Para mejorar y facilitar esto, crearon dos interfaces: *Virtual Dashboard Interface* y *Telepresence Interface*.

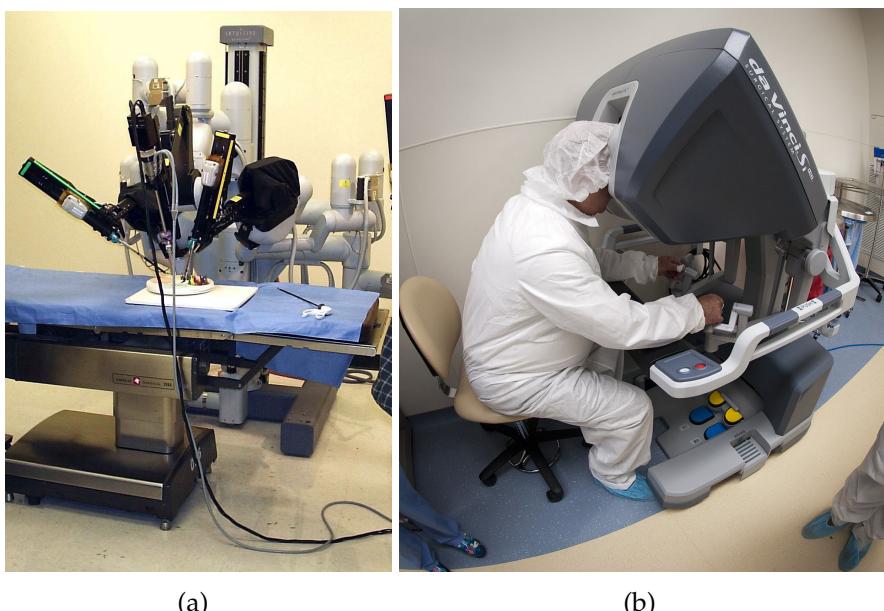
*Virtual Dashboard interface* consiste en un panel de control donde el operador podía ver de forma simple el estado actual de Nomad y controlarlo de forma intuitiva.

*Telepresence Interface* es una interfaz que hace uso de la cámara panorámica de Nomad. A través de dicha interfaz el usuario podía ver las imágenes capturadas por Nomad e incluso podía girar el punto de vista o ampliar la imagen para conocer mejor el entorno.

Estas imágenes se reproducían en una pantalla ancha, parecida a un parabrisas de un coche, frente al operador, para dar más sensación de telepresencia.

Uno de los inconvenientes de este trabajo fue el delay a la hora de recibir las imágenes (5 segundos), que hacía más difícil el control. A pesar de esto **los resultados fueron positivos**, concluyendo que los operadores manejaban mejor el robot haciendo uso de la Telepresence Interface que sin ella.

Otra de las aplicaciones de la combinación de robótica con realidad virtual es la mejora en **intervenciones quirúrgicas**.



**Figura 2.3: Da Vinci Surgical System**

Un ejemplo de esto es **Da Vinci Surgical System**, un sistema en el cual el cirujano controla, a través de unos mandos, las herramientas necesarias para operar al paciente. Además tiene una pantalla 3D en la que se ve a tiempo real la zona que está siendo intervenida. Este sistema hace posible que el cirujano realice **movimientos mucho más precisos**, además de facilitarle una postura más cómoda durante la operación. [2]

## **2. ESTADO DEL ARTE**

---

Lo que nos muestran los resultados de estos proyectos es que los usuarios son capaces de manejar dispositivos que ven a través de realidad virtual de forma exitosa. Esto también es imprescindible en nuestro trabajo ya que el usuario debe controlar el robot tan solo con las gafas de realidad virtual.

# ANÁLISIS

Ahora se explicarán las fases que se han llevado a cabo para desarrollar el proyecto.

La primera fase es la de análisis. Se estudia el objetivo y los requisitos que debe cumplir. De esta forma el trabajo queda dividido en pequeñas tareas y facilita su implementación.

En el capítulo de *Objetivos* se explica que la meta final del proyecto es conseguir que el usuario pueda controlar una cámara con el movimiento de cabeza y ver todo lo que la cámara vea.

La forma más eficiente de hacer que la cámara sea móvil es construir un robot con ruedas que lleve la cámara integrada.

De esta forma cuando el usuario mueva la cabeza, se le enviarán las instrucciones necesarias al robot para que se mueva. A su vez el robot le estará mandando video a tiempo real al usuario, que lo verá a través de las Oculus Rift.

El proyecto por lo tanto se divide en 3 problemas:

1. **Construcción del robot** → para ello necesitaremos una placa base, una cámara y la estructura que le permita desplazarse.
2. **Streaming** → Debemos conseguir transmitir a tiempo real todo lo que ve el robot a la Oculus
3. **Control del robot** → Queremos que el robot sea una extensión de los sentidos del usuario, por lo que debemos idear movimientos intuitivos para que el usuario mande comandos de movimiento al robot.

Vamos a desarrollar cada uno de ellos, identificando qué funcionalidades básicas debe cumplir.

## 3.1 Requisitos Funcionales

### 3.1.1 Construcción del Robot

El robot debe ser un dispositivo fácil de controlar, que a su vez actúe como una extensión de los ojos del usuario, dándole a éste la sensación de inmersión en un espacio real.

Por tanto los requisitos que debe cumplir el robot son los siguientes:

- **RF-CR1**

Permitir grabar video y transmitirlo a tiempo real.

- **RF-CR2**

Ser capaz de desplazarse en todas las direcciones del plano.

- **RF-CR3**

Recibir comandos de control a través de una red inalámbrica.

- **RF-CR4**

La cámara debe apuntar en la misma dirección que la cabeza del usuario.

### 3.1.2 Streaming

Se quiere conseguir la máxima sensación de inmersión para la persona que controla el robot.

Para ello los requisitos principales que debe cumplir son:

- **RF-S1**

La transmisión de video debe ser a tiempo real, con la mínima latencia posible.

- **RF-S2**

El usuario verá el video con las Oculus Rift, por lo que la imagen debe estar desdoblada (formato SBS, que se explicará en la sección de desarrollo).

- **RF-S3**

La transmisión debe ser inalámbrica ya que queremos que el robot tenga total libertad de movimiento, siendo éste independiente del ordenador.

### 3.1.3 Control del Robot

Como hemos dicho anteriormente, el robot pretende ser una extensión de los ojos del usuario, por lo que es lógico que se controle con el movimiento de cabeza de la persona que esté recibiendo el video.

Dentro del control del robot vamos a diferenciar entre control del movimiento del robot y control de la dirección de la cámara:

- Movimiento del robot :

**RF-CTRL1**

El robot debe moverse en todas las direcciones con movimientos de la cabeza.

- Movimiento de la cámara:

**RF-CTRL2**

La cámara debe estar apuntando siempre en la misma dirección que la cabeza.

## 3.2 Requisitos no funcionales

- **RNF1 - Usabilidad**

El sistema para controlar el movimiento del robot y de la cámara debe ser intuitivo, sin requerir movimientos de cabeza inusuales por parte del usuario.

- **RNF2 - Accesibilidad**

Dado que el trabajo está orientado a personas con discapacidad motora, el control del robot debe ser accesible para las mismas, de forma que no requiera de movimientos bruscos o exagerados.



# DISEÑO

Tras analizar las diferentes funcionalidades del trabajo vamos a ver cómo está diseñado cada componente.

Recordemos que el objetivo es construir un dispositivo que se comunique de forma inalámbrica con el usuario. El robot debe transmitir imágenes a tiempo real del entorno y recibir comandos de movimiento para que el usuario pueda explorar el espacio.

Por lo tanto los principales dispositivos que componen el proyecto son

- **Robot**

Debe ser un dispositivo móvil, con una webcam integrada para poder transmitir imágenes.

Además de la cámara deberá tener integrado algún mecanismo de envío y recepción de datos.

- **Router**

Se utilizará un router para crear un red local a través de la cual se envíen tanto las imágenes como los comandos de control del robot.

- **Oculus Rift y PC**

Las gafas de realidad virtual reproducirán las imágenes para el usuario. Deben estar conectadas a un PC que reciba el video y envíe los comandos al robot.

También se tratará el **diseño de conexiones** entre los 3 componentes.

## 4.1 Diseño del robot

El robot que se utiliza en este proyecto está basado en un diseño anterior hecho por (Nombre del alumno que diseñó el robot). De este diseño hemos reutilizado (estructura???)

En un principio también se pretendía utilizar la misma placa: **Arduino Yun**.

Esta placa tiene la capacidad de conectarse a **internet a través de Wifi**, además tiene dos procesadores, el Atmega y el Atheros AR9331, lo que le permite de **combinar Arduino con Linux**. Esto era muy útil ya que el control de servomotores es muy sencillo con Arduino y tener un sistema operativo Linux en la placa nos permitía instalar programas de manejo de vídeo.

Tras muchas pruebas se decidió no utilizar esta placa ya que el módulo Wifi daba muchos problemas.

Finalmente el robot consta de:

- Una placa base → Raspberry Pi 3
- Dos ruedas conectadas con dos servos que permiten al robot desplazarse.
- Una cámara Logitech, conectada a la placa por USB
- Un servo unido a la cámara

### 4.1.1 Raspberry Pi 3 model B

La Raspberry Pi 3 model B es un ordenador de placa reducida que lleva un sistema operativo Linux.

Su procesador es un ARM Cortex A53 de cuatro núcleos a 1.2GHz de 64 bits.

Tiene 1 GB de memoria RAM , 4 puerto USB, 40 pins GPIO, puerto HDMI , Ethernet y entrada para MicroSD.

Además tiene Wifi 802.11n integrado y bluetooth 4.1.

Tras rechazar el Arduino Yun como placa para el proyecto, la Raspberry Pi 3 quedó como mejor candidata ya que cumplía todos los requisitos necesarios. A saber:

- **Módulo Wifi:** Imprescindible para la transmisión.
- **S.O Linux:** Nos permite instalar programas de manejo y transmisión de video.
- **Puerto USB:** Para conectar la cámara y la fuente de alimentación.
- **GPIO:** Necesario para conectar los servo motores.

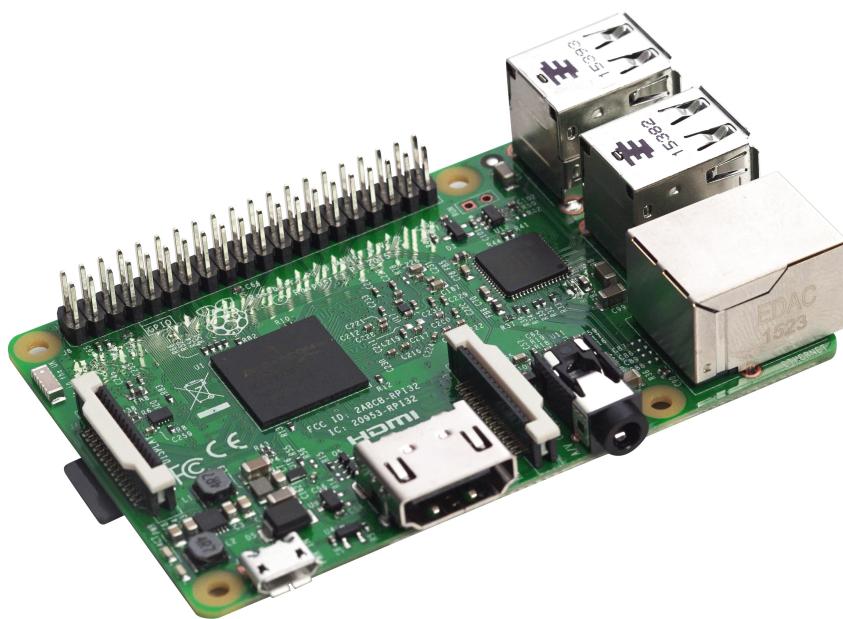


Figura 4.1: Raspberry Pi 3 model B

#### 4.1.2 Servo motores

Para el movimiento del robot y de la cámara utilizamos servo motores

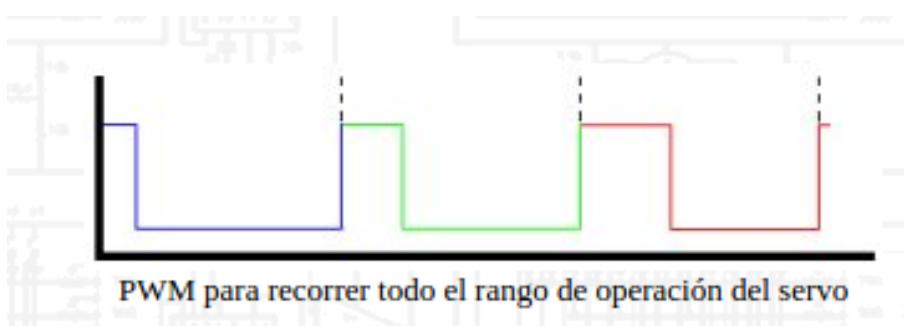
Un servo motor es un motor eléctrico que se puede controlar su velocidad y su posición (dentro del rango de posición permitido).

Los servos constan de:

- Un motor de corriente continua
- Una caja reductora
- Un circuito de control

El sistema que utilizamos para controlar la velocidad y la posición de los servos es la modulación por anchura de pulso (PWM).

**PWM** Este sistema consiste en generar una onda cuadrada en la que se varía el tiempo que el pulso está a nivel alto, manteniendo el mismo período (normalmente), con el objetivo de modificar la posición del servo según se deseé.



## 4. DISEÑO

---

Para que un servo se mantenga en la misma posición durante un cierto tiempo, es necesario enviarle continuamente el pulso correspondiente.

La otra función del PWM es el control de velocidad.

Esto lo hace alimentando el motor con una señal de pulsos con la frecuencia suficiente para que el motor no note las variaciones y haga un giro constante, ya que variando el porcentaje de tiempo de la señal rectangular en alta, y en baja, variaremos la potencia que le entregamos al motor, con lo que podemos controlar la velocidad de giro.

En el robot utilizamos 3 servomotores. Dos de ellos se utilizan para el movimiento de las ruedas y el tercero moverá la cámara permitiendo al usuario mirar hacia arriba y hacia abajo.

Los tres servos están controlados por el movimiento de la cabeza del usuario, que obtenemos gracias a las Oculus Rift.

### 4.1.3 Cámara

La cámara utilizada es una webcam USB. El proyecto también se podría haber hecho con una RaspiCam.

A continuación se muestra un esquema de la conexión entre los componentes que acabamos de explicar.

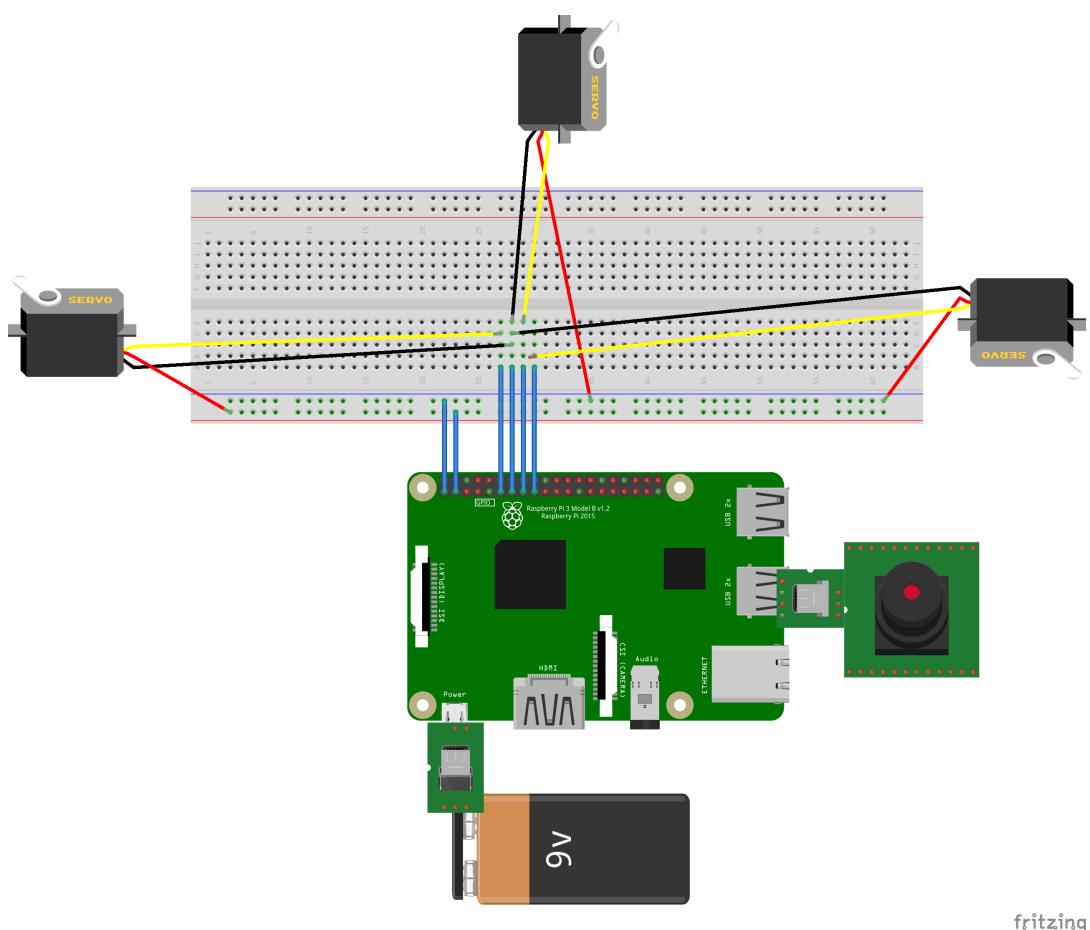


Figura 4.2: Esquema conexiones servos , cámara, Raspberry Pi 3

## 4.2 Router

Para la comunicación entre el robot y el ordenador se necesita una red local, para este trabajo la hemos creado haciendo uso de un router inalámbrico Xavi7968.

El router crea una red Wifi con el nombre de WifiRaspi3 y sin acceso a internet.

Para una mayor facilidad de conexión se ha configurado la tabla DHCP del router para asignar siempre la misma IP a la Raspberry y al equipo que utilizamos para conectarnos.

The screenshot shows the 'DHCP Server Configuration' page of a web-based router management interface. On the left, a sidebar menu includes 'Overview', 'Troubleshooting', 'Configuration' (selected), 'Quick Setup', 'Wireless Network', 'Internet Connection', 'Local Network (LAN)', 'DHCP Server' (selected), 'Vlan Config', 'Port-PVC', 'Security', 'Services', 'Port Statistics', and 'Admin'. The main content area has two sections: 'Existing DHCP server subnets' and 'Existing DHCP fixed IP/MAC mappings'. The 'Existing DHCP server subnets' section shows one entry: Subnet Value 192.168.1.0, Subnet Mask 255.255.255.0, Use local host address as DNS server false, Use local host address as default gateway true, Assign Auto Domain Name true, and Edit, Delete, and Add Subnet buttons. The 'Existing DHCP fixed IP/MAC mappings' section shows three entries:

IP Address	Mac Address	Max Lease Time	Default Lease Time
192.168.1.35	74:da:38:2e:03:38	86400	43200
192.168.1.33	b8:27:eb:70:7b:14	86400	43200
192.168.1.34	f8:16:54:c7:f1:ff	86400	43200

Buttons for Edit, Delete, and Add Host are available for each row. A 'DHCP Server (Server Settings)' sidebar on the right provides information about the DHCP protocol and its configuration, including options for upgrading the firmware and managing fixed IP mappings.

Figura 4.3: Tabla DHCP del router con las parejas MAC-IP fijas

La Raspberry Pi 3 está configurada para conectarse automáticamente al router. Para ello se ha modificado el archivo /etc/wpa\_supplicant/wpa\_supplicant.conf donde se especifica el nombre y la clave de la red local.

The screenshot shows a terminal window titled 'pi@raspberrypi: /etc/wpa\_supplicant'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', 'Ayuda'. Below the menu is the text 'GNU nano 2.2.6' and 'File: wpa\_supplicant.conf'. The main area contains the following configuration text:

```

ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=GB

network={
    ssid="Wifi_Raspi3_2"
    key_mgmt=NONE
}

```

Figura 4.4: Archivo de configuración de red Raspberry Pi 3

## 4.3 Oculus Rift

Las Oculus Rift son unas gafas de realidad virtual creadas por Palmer Luckey.

La primera versión de las Oculus fue la DK1 (Development Kit 1), enfocada para programadores, que salió en abril de 2013. Esta versión se retiró del mercado poco antes de la presentación del segundo kit de desarrollo (DK2) en marzo de 2014.

Finalmente, en 2016 salió la versión para consumidor.

Para este proyecto se utilizan las Oculus Rift DK2.

De las gafas nos interesan especialmente:

- Detección de la orientación de las Oculus
- Sensación de inmersión (3D)

### 4.3.1 Detección de la orientación

Las Oculus Rift son capaces de detectar la **posición y orientación de la cabeza** del usuario.

Para la **posición**, las gafas llevan un conjunto de emisores infrarrojos en su superficie. La información enviada por dichos emisores es recogida por un sensor de infrarrojos que se asemeja a una cámara.

Para nuestro trabajo la posición no es del todo relevante ya que, al estar orientado para personas con discapacidad motora, se espera que el usuario se encuentre más o menos en la misma posición durante el uso del dispositivo.

Por el contrario, conocer la **orientación** de la cabeza es esencial en este proyecto ya que el usuario enviará comandos al robot rotando la cabeza en el sentido que convenga en cada momento.

Cualquier objeto en un entorno tridimensional puede rotar respecto a tres ejes: x, y, z.

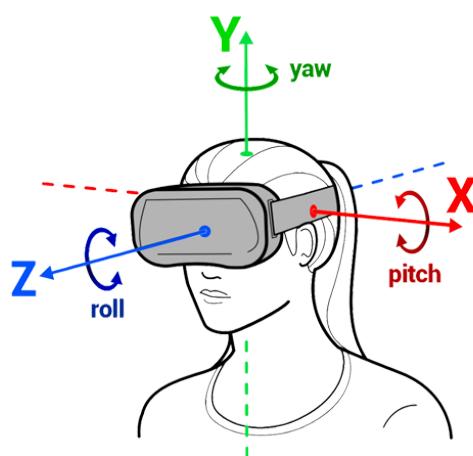


Figura 4.5: Ejes de giro

Para detectar la orientación, las Oculus tienen integrados tres sensores: un giroscopio, un acelerómetro y un magnetómetro.

Con el giroscopio y el acelerómetro se calcula la orientación respecto a los ejes X y Z, mientras que el magnetómetro manda información del eje Y.

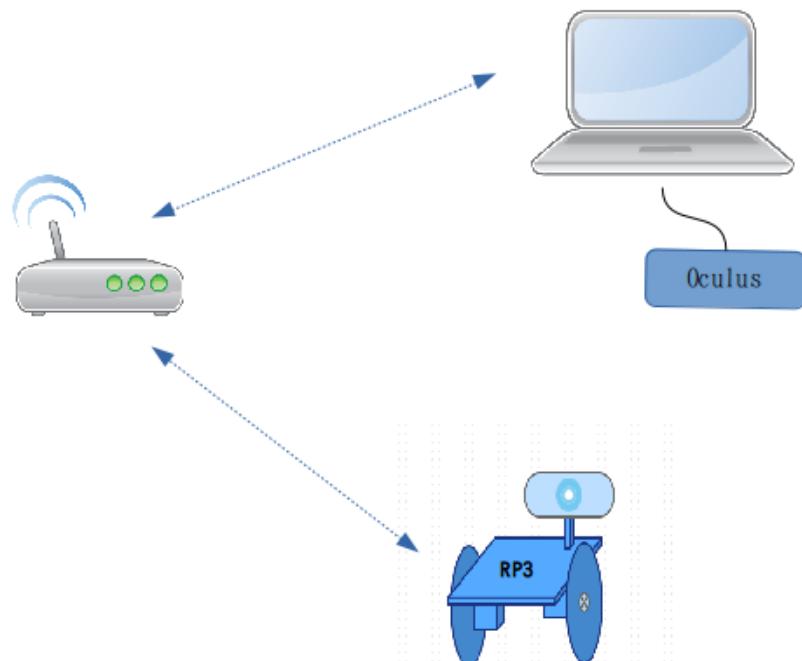
#### 4.3.2 Sensación de inmersión \_ 3D

Una de las características más importantes de las Oculus Rift es su capacidad de reproducir un entorno virtual de tal forma que el usuario se sienta totalmente parte de él.

Esto lo hace usando un par de pantallas (una para cada ojo) que dividen la imagen en dos (Side by side - SBS). En cada pantalla hay un conjunto de lentes que modifican la imagen para crear el fenómeno de la estereopsis por el cual, a partir de dos imágenes ligeramente diferentes proyectadas en la retina de cada ojo, el cerebro es capaz de recomponer una tridimensional.

## 4.4 Diseño de conexiones

La arquitectura del proyecto es muy sencilla, como podemos observar en la siguiente imagen:



**Figura 4.6:** Esquema conexiones

Como podemos ver en el esquema, las Oculus Rift están conectadas al ordenador, el ordenador y la Raspberry se mandan información a través de un router que crea una red local.

La raspberry recibe en streaming de la cámara y lo va mandando a una IP disponible dentro de la red local, la (192.168.1.35). A su vez el ordenador accede a esa IP, recoge el streaming utiliza VLC para dividir la imagen (SBS) y reproducir el video y con la herramienta Virtual Desktop podemos ver el video desde las Oculus.

#### 4. DISEÑO

---

Por otra parte las Oculus tienen integrado un giroscopio , un acelerómetro y un magnetómetro.

Combinando la información de estos sensores a través de un proceso conocido como fusión de sensores se determina la orientación de la cabeza del usuario en el mundo real , y se sincroniza con la perspectiva virtual del usuario en tiempo real.

Se ha implementado un programa en phyton que recoge esa información y la traduce en comandos sencillos para mandarselos a la Raspberry. Para obtener esa información hacemos uso de la librería de las Oculus (ovr).

La Raspberry lee los comandos del ordenador y mueve los servos según lo recibido, de esta forma controlamos el movimiento del robot y de a cámara.

# DESARROLLO E IMPLEMENTACIÓN

En esta sección se explicará cómo hemos desarrollado e integrado las distintas funcionalidades para conseguir el proyecto final. Para ello vamos a diferenciar entre:

- Construcción del Robot
- Streaming
- Control del Robot

Para conseguir **transmitir a las Oculus el video en tiempo real** es necesario que:

1. La Raspberry recoja el video de la cámara y lo transmita al ordenador.
2. El ordenador recoja el video, lo transforme a formato SBS y lo envíe a las Oculus Rift.

Para el **control del robot** se deberán implementar las siguientes tareas:

1. Leer la señal de los sensores de las Oculus Rift (giroscopio, acelerómetro y magnetómetro) para saber la orientación de la cabeza del usuario.
2. Transmitir la información obtenida al robot.
3. Traducir dicha información en comandos para mover los servo motores.

## 5.1 Construcción del Robot

Como se ha dicho en la sección de diseño, el robot está construido basándose en el diseño de un robot de un trabajo anterior.

Para este proyecto se han añadido algunos componentes:

- Soporte de la cámara
- Batería portátil

### 5.1.1 Soporte de la cámara

Para integrar la cámara en el robot han sido necesarias dos piezas:

- Soporte vertical → En esta pieza se coloca el servo motor que controla el movimiento de la cámara.
- Pinza de agarre → Se utiliza para unir la cámara con el servo motor que la mueve.

Ambas piezas han sido impresas con la impresora 3D del Club de Robótica de la Universidad Autónoma de Madrid.

Para realizar el diseño 3D de la pinza de agarre se ha utilizado el programa de diseño 3D Blender.

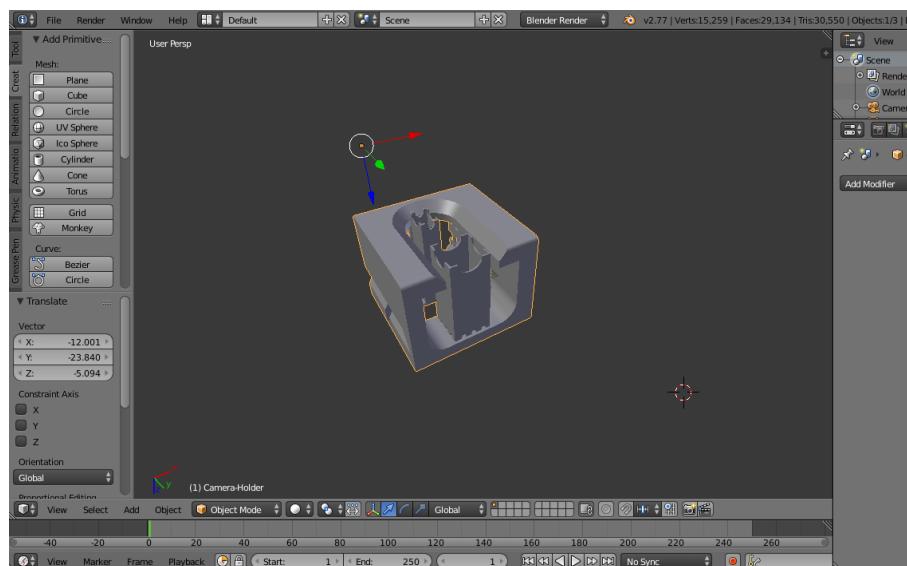
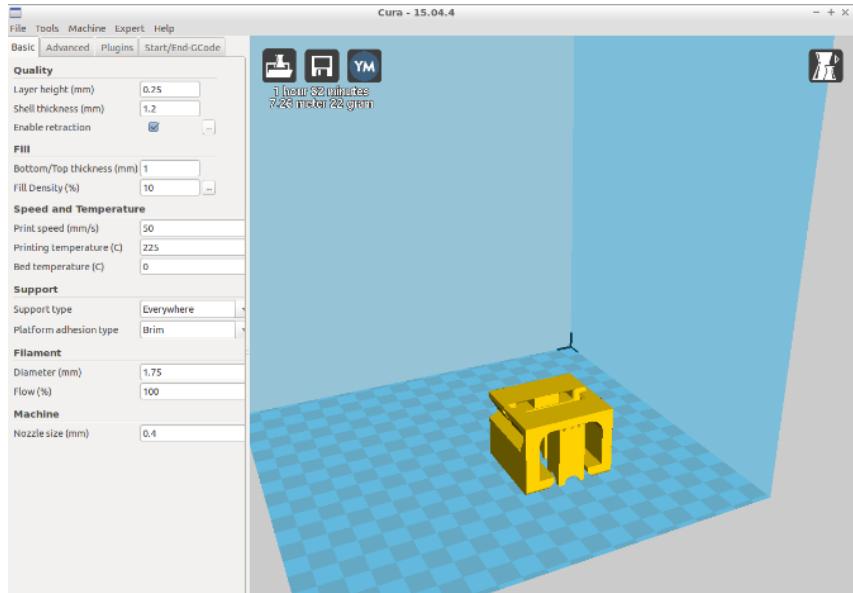


Figura 5.1: Diseño 3D de la pinza de agarre en Blender

Una vez diseñadas las piezas se ha utilizado Cura, un programa que genera archivos .gcode a partir de diseños 3D. Este tipo de archivos son los que se le envían a la impresora 3D.



**Figura 5.2:** Imagen de la configuración del Cura para enviar la Pieza de agarre a la impresora 3D

## 5.2 Streaming

A continuación se explicará cómo se ha implementado la captura, transmisión y reproducción de video del entorno en tiempo real.

Para esto son necesarios dos elementos:

- Raspberry Pi3 → Se encarga de la captura y transmisión de video.
- PC → Se encarga de la recepción de video y de reproducirlo en las Oculus Rift.

### 5.2.1 Raspberry Pi3

El primer elemento en el streaming de video es la Raspberry Pi, que tiene conectada por usb una cámara.

Para recoger el video y mandarlo en tiempo real utilizamos la librería de software libre `mjpeg-streamer`, disponible en [github](https://github.com/jacksonliam/mjpg-streamer). <https://github.com/jacksonliam/mjpg-streamer>

Mjpeg-streamer es una aplicación en linea de comandos que permite crear un servidor, para retransmitir imágenes JPG sobre una red basada en IP, desde la cámara hasta un navegador convencional.

Soporta la compresión por hardware (GPU) de la cámara, en nuestro caso H.264 Advanced Video Coding (AVC) Standard, que es la compresión utilizada por la Webcam Logitech.

Esto permite reducir drásticamente el uso de la CPU de este servidor, haciendo esta aplicación un servicio ligero.

## 5. DESARROLLO E IMPLEMENTACIÓN

```
pi@raspberrypi: ~
top - 20:34:28 up 17 min, 4 users, load average: 0.54, 0.32, 0.15
Tasks: 120 total, 1 running, 119 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.1 us, 2.7 sy, 0.0 ni, 95.0 id, 0.0 wa, 0.0 hi, 2.1 si, 0.0 st
KiB Mem: 883052 total, 216956 used, 666096 free, 26448 buffers
KiB Swap: 102396 total, 0 used, 102396 free. 108000 cached Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
    976 root      20   0        0      0      0 S 6.9  0.0  0:08:66 kworker/u8+
   210 root      -2   0        0      0      0 S 5.0  0.0  0:13:70 ksdiocirqd/+
    6 root      20   0        0      0      0 S 2.0  0.0  0:12:72 kworker/u8+
  1020 pi      20   0  51816  2756  2524 S 1.7  0.3  0:03:99 mjpg_strea+
    3 root      20   0        0      0      0 S 1.3  0.0  0:03:71 ksoftirqd/0
  1072 pi      20   0  5112  2460  2092 R 0.7  0.3  0:00:61 top
   15 root      20   0        0      0      0 S 0.3  0.0  0:00:35 ksoftirqd/2
   19 root      20   0        0      0      0 S 0.3  0.0  0:00:54 ksoftirqd/3
    1 root      20   0  23872  3904  2732 S 0.0  0.4  0:05:00 systemd
    2 root      20   0        0      0      0 S 0.0  0.0  0:00:00 kthreadd
    5 root      0 -20       0      0      0 S 0.0  0.0  0:00:00 kworker/0:+
    7 root      20   0        0      0      0 S 0.0  0.0  0:00:11 rcu_sched
    8 root      20   0        0      0      0 S 0.0  0.0  0:00:00 rcu_bh
    9 root      rt   0        0      0      0 S 0.0  0.0  0:00:01 migration/0
   10 root      rt   0        0      0      0 S 0.0  0.0  0:00:00 migration/1
   11 root      20   0        0      0      0 S 0.0  0.0  0:00:44 ksoftirqd/1
   13 root      0 -20       0      0      0 S 0.0  0.0  0:00:00 kworker/1:+
   14 root      rt   0        0      0      0 S 0.0  0.0  0:00:00 migration/2
   17 root      0 -20       0      0      0 S 0.0  0.0  0:00:00 kworker/2:+
   18 root      rt   0        0      0      0 S 0.0  0.0  0:00:00 migration/3
   21 root      0 -20       0      0      0 S 0.0  0.0  0:00:00 kworker/3:+
```

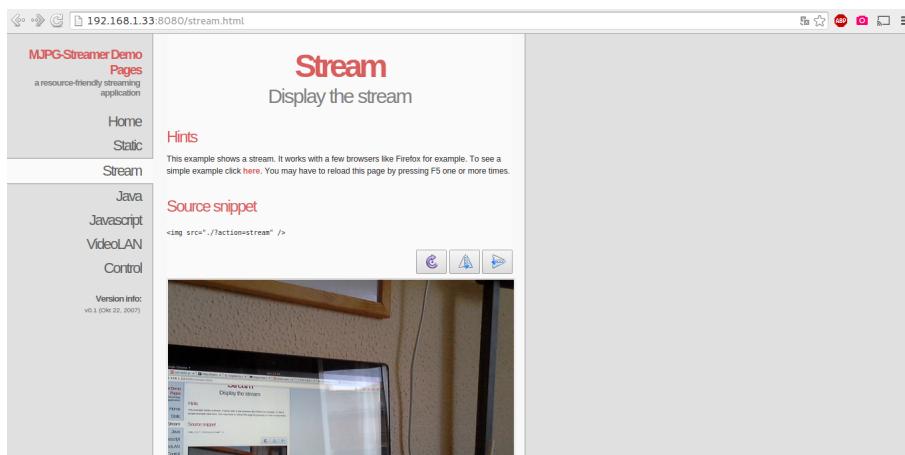
**Figura 5.3:** Uso de CPU del programa Mjpeg-Streamer por el que se hace el streaming de video

Para iniciar el streaming debemos ejecutar un script Streaming que manda el siguiente comando: `./mjpg_streamer -i "./input_uvc.so" -d "/dev/video0" -o "./output_http.so" -w "./www"`.

```
pi@raspberrypi:~/mjpg-streamer/mjpg-streamer-experimental$ ./mjpg_streamer -i "./input_uvc.so" -d "/dev/video0" -o "./output_http.so" -w "./www"
MJPG Streamer Version: 2.0
i: Using V4L2 device: /dev/video0
i: Desired Resolution: 640 x 480
i: Frames Per Second.: 1
i: Format.....: JPEG
i: TV-Norm.....: DEFAULT
o: www-Folder-path: ./www/
o: HTTP TCP port...: 8080
o: username:password: disabled
o: commands.....: enabled
```

**Figura 5.4:** Ejecución del comando para iniciar el streaming

En este momento la Raspberry empezará a coger el video de la cámara y transmitirlo en tiempo real a la siguiente url: `http://192.168.1.33:8080/stream.html`



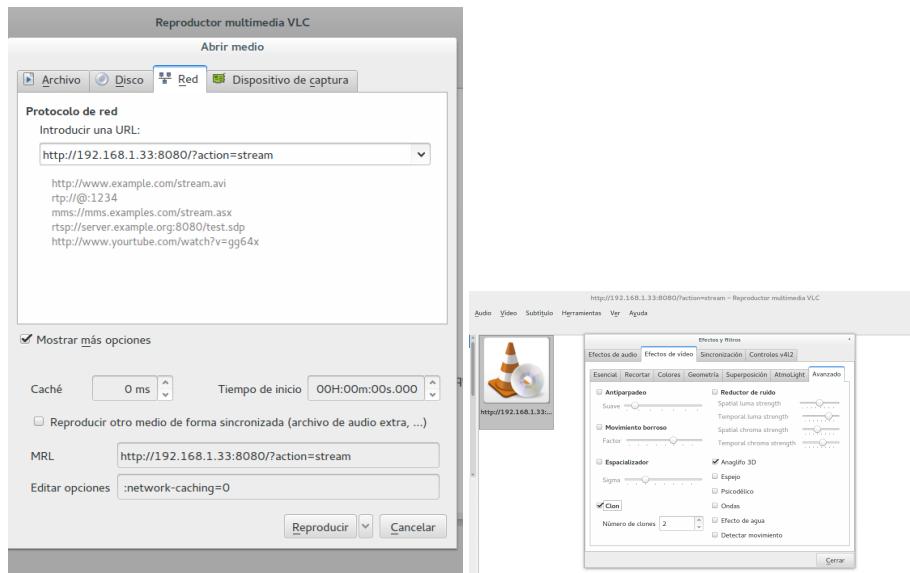
**Figura 5.5:** Página html donde se recibe el streaming a tiempo real desde la Raspberry

### 5.2.2 Ordenador

El ordenador, conectado a la red local a través del router, accede a la IP donde mjpeg-streamer está retransmitiendo el video y lo recoge a través del reproductor de video VLC.

Para ello especificamos en VLC que el medio de reproducción será una ubicación de red.

Posteriormente dividimos el video en dos pantallas exactamente iguales para poder verlo desde las Oculus en modo SBS (Side by Side).



(a) Medio de reproducción-Ubicación de red

(b) División de imágenes



(c) Reproducción SBS

**Figura 5.6:** Configuración y resultado de la reproducción de video en VLC

Para visualizar el video en la Oculus Rift hemos instalado en el PC el Virtual Desktop. Virtual Desktop es una aplicación desarrollada por Oculus Rift y HTC Vive para utilizar las Oculus Rift en Windows.

## 5. DESARROLLO E IMPLEMENTACIÓN

---

Permite ver el escritorio desde las Oculus, pudiendo configurar efectos como ver las imágenes SBS o cambiar el entorno en el que te encuentras (el espacio, una sala de cine...)

También te permite ver videos descargados en el ordenador y reproducir videos 360.

Nosotros utilizamos la funcionalidad de ver la pantalla del ordenador desde las Oculus con SBS, para visualizar el video que se está reproduciendo en VLC a tiempo real.

## 5.3 Control del Robot

El control del robot es el control del movimiento de los tres servos que lo componen.

Este movimiento se basará en el movimiento de las Oculus, que llevará puestas el usuario.

Por lo tanto el control del robot se divide en tres acciones:

- Recogida y procesamiento de la información que mandan las Oculus al ordenador → Esto se hará mediante la utilización de la librería OVR.
- Envío de la información procesada a la Raspberry Pi 3 → Para ello el ordenador abrirá dos sockets a través de los cuales mandará los datos al robot
- Movimiento de los servo motores

### 5.3.1 Recogida y procesamiento de la información que mandan las Oculus al ordenador

**Librería OVR** Para recoger desde el ordenador la información que mandan las Oculus utilizamos la librería de Oculus (OVR), escrita en python.

La función `ovr.getTrackingState` nos devuelve una estructura (*TrackingState*), que contiene el campo:

**Pose Statef HeadPose**

- **Posef ThePose**  $\begin{cases} \text{Quatf Orientation} \\ \text{Vector3f Position} \end{cases}$

Para obtener la orientación de la cabeza se lee el valor de *HeadPose.ThePose.Orientation.x/y* lo procesa para obtener los grados de orientación de la cabeza y se lo manda a la raspberry a través de sockets.

La información obtenida sobre la orientación respecto al eje *y* servirá para mover los servos que hay en las ruedas del robot, mientras que la información de *x* servirá para mover la cámara.

El ordenador crea un servidor y abre dos sockets distintos, uno para la información de *x* y otro para la de *y*.

De esta forma manejamos los dos movimientos de forma independiente.

**Procesamiento de HeadPose.ThePose.Orientation** Como se ha dicho en el apartado anterior, la información proveniente de las Oculus nos llega a través de la estructura *HeadPose*, donde podemos leer el campo *Orientation* que nos manda la orientación actual de las Oculus Rift en forma de cuaternión.

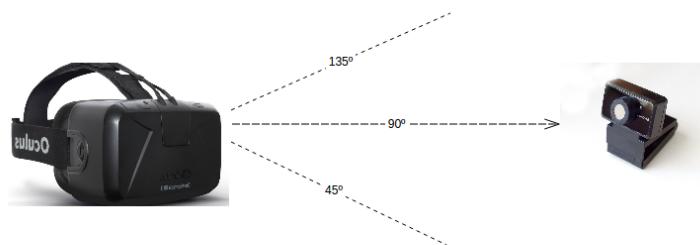
Para averiguar en qué dirección apuntan las Oculus respecto al eje *X* guardamos los datos relativos a la posición y la velocidad angular.

## 5. DESARROLLO E IMPLEMENTACIÓN

---

Estos datos son procesados y se le aplican unos parámetros empíricos para obtener una correcta calibración.

De este modo se ha podido realizar una conversión de los datos obtenidos por las Oculus a unidades de grados que nos facilitan el control posterior de la posición de los servomotores.



**Figura 5.7:** Posición de las Oculus respecto al eje X una vez procesados los datos de HeadPose.ThePose.Orientation

**Envío de la información procesada a la Raspberry Pi 3** Para el control del movimiento del robot es necesario que el ordenador le mande los datos referentes a la posición de las Oculus de forma automática, inmediatamente después de procesarlos.

Con este fin, desde el mismo programa donde se recibe y procesan los datos de las Oculus, el ordenador abre dos sockets, y se queda a la espera de que el robot se conecte.

Recordemos que el ordenador y la Raspberry Pi 3 están dentro de la misma red local (WifiRaspi3), y que ambos tenían IP fija.

La apertura de sockets por parte del ordenador se hace de la siguiente forma:

```
host = "192.168.1.35"      #IP asignada al ordenador
port = 4446                #movimiento de la camara
port2 = 4447                #movimiento de las ruedas

s = socket(AF_INET, SOCK_STREAM)
s2 = socket(AF_INET, SOCK_STREAM)

s.bind((host, port))
s2.bind((host, port2))

s.listen(5)
s2.listen(5)

q,addr = s.accept()
q2,addr2 = s2.accept()
```

De esta forma, cada vez que se recibe y procesan datos referentes a la posición de las Oculus, se dividen en información para la cámara e información para las ruedas y se ejecutan las instrucciones `q2.send(data)` y `q.send(data)` para mandar los datos.

*Recepción de los datos en el robot.* La Raspberry tiene dos programas (uno para las ruedas y otro para la cámara) que están escuchando continuamente lo que manda el ordenador y en cuanto recibe el comando lo manda a los servos.

Para ello, al iniciarse cada programa, se conectan al socket correspondiente

```
host = "192.168.1.35"
port=4447
s=socket(AF_INET, SOCK_STREAM)
s.connect((host, port))
```

Y se ponen a escuchar hasta que les llegan los datos.

Una vez que tiene los datos los transforma en comandos válidos para los servomotores, tal y como se explica en el siguiente apartado, les manda la instrucción y vuelve a escuchar.

### 5.3.2 Movimiento de los servo motores

Como ya se indicó en la sección (sección de los servos, PWM, etc..) el movimiento de los servos se controla con una señal PWM (Pulse Width Modulation).

Por ello utilizamos la librería RPi.GPIO, que ofrece funciones para PWM.

Hay dos parámetros principales para determinar el pulso que se envía al motor:

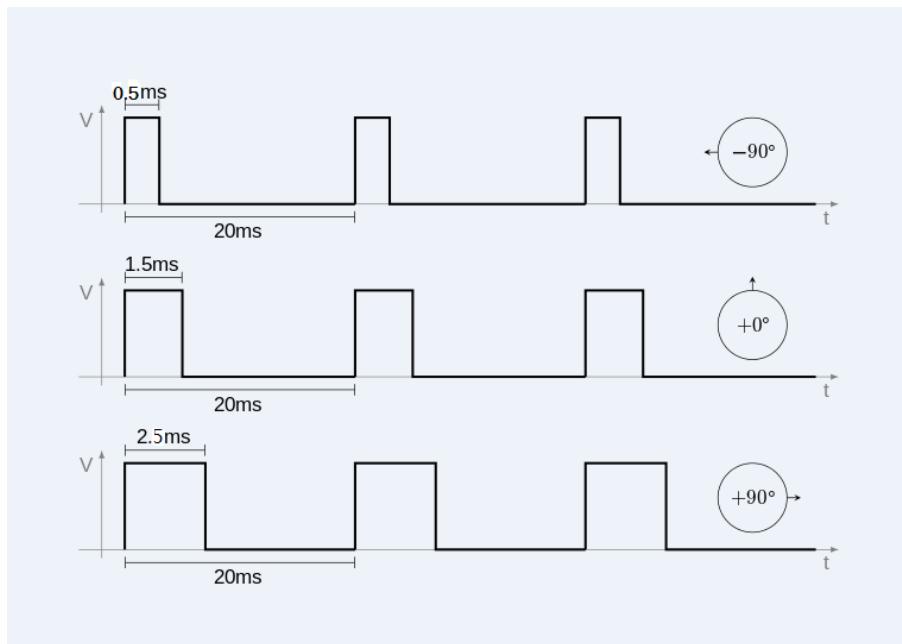
- **Frecuencia**: veces por segundo que se genera el pulso.
- **Duty cycle**: indica el porcentaje de tiempo que dura la señal cuadrada en relación con el periodo.

$$DutyCycle = \frac{PW}{T} \cdot 100$$

Donde  $PW$  indica el ancho de pulso (pulse width) y  $T$  es el periodo.

Para controlar los servo motores del robot se fija una frecuencia de 50 Hz, es decir,  $T = \frac{1}{50} = 20ms$ . En las especificaciones de los servos se indica que el rango de  $PW$  es  $0.5\text{ ms} \sim 2.5\text{ ms}$  y con esto ya podemos calcular el Duty Cycle adecuado para cada acción.

En la siguiente imagen se muestra cómo varía la posición de los motores según el pulso que se le envíe.



**Figura 5.8:** Relación entre el PW y la posición de los servo motores

Recordemos que estos valores son orientativos, pudiendo variar ligeramente según el servo motor.

La Raspberry Pi recibe a través de los sockets información sobre la orientación de la cabeza sobre el eje y o x.

Esta información se procesa de forma distinta según si queremos mover el servo de la cámara o las ruedas del robot.

Esto es porque los servos de las ruedas son de rotación continua, a diferencia que el de la cámara.

#### ***Control de los motores de las ruedas del robot***

Los motores que mueven las ruedas del robot son dos servo motores de rotación continua. El funcionamiento de los servo motores de rotación continua ya se ha explicado en la sección de análisis.

Recordamos muy brevemente que en los servo motores de rotación continua, la pulsación que le mandas al motor no indica la posición en la que debe colocarse si no la dirección y velocidad a la que debe rotar.

Por lo tanto si se envía un pulso cercano a 1.5 ms, el motor girará muy despacio. Por el contrario, si el pulso es cercano a 0.5 o 2.5 ms, el motor se moverá a mayor velocidad.

Para controlar nuestro robot se ha fijado una velocidad constante de giro en base a pruebas experimentales que se explicarán en la sección de Pruebas y resultados.

De esta forma para rotar nuestro robot hasta una posición concreta la variable no es la velocidad si no el tiempo durante el que rotan los motores.

Para girar el robot hacia la derecha el usuario enviará la señal de "rotar hacia la derecha" y el robot empezará a rotar en esa dirección hasta recibir la señal de "parar".

#### ***Control del motor de la cámara del robot***

La cámara se mueve gracias a un servo motor cuyo rango de movimiento va de 0 a 180 grados.

Este motor se mueve por pulsos, tal y como se explica en la sección de Análisis.

La Raspberry recibe un ángulo que indica hacia donde están orientadas las Oculus en ese momento.

Para transformar este ángulo el pulso(PWM) correspondiente debemos tener en cuenta el rango de pulsos del motor que se está utilizando.

Como vemos en la figura 5.8, un pulso de 1.5 ms equivale a la posición del motor que señala el ángulo 0, de la misma manera para que el motor esté orientado hacia el ángulo 180 deberemos mandar un pulso de 0.5 ms.

Con estos datos es sencillo encontrar una fórmula de conversión de grados a pulsos:

Así podemos convertir los ángulos en pulsos PWM y mover la cámara hacia la dirección deseada.

**Control del Robot por parte del usuario** En las anteriores secciones se ha explicado el funcionamiento interno del control de movimientos del robot.

En este nuevo apartado se presenta cómo debe utilizar el usuario este producto, qué rango de movimiento tiene y cómo puede hacer avanzar el robot.

### Instrucciones de mando

Como se especifica al inicio de este trabajo, el robot estará controlado exclusivamente por el movimiento de cabeza del usuario.

Para ello se debía idear una forma cómoda e intuitiva de mandar instrucciones al robot.

Diferenciamos dos estados de movimiento del robot:

- Exploración → En este estado el robot no avanza, solo explora lo que tiene a su alrededor. Esto lo hace moviendo la cámara o rotando sobre sí mismo.
- Transporte → El robot avanza hacia delante. Al poder girar sobre sí mismo en la fase de **exploración**, es suficiente con este movimiento.

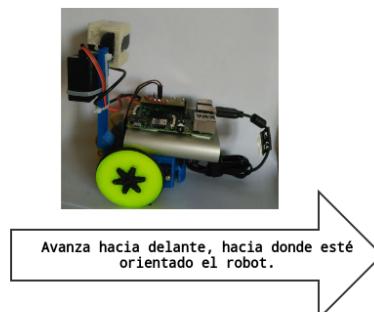
Finalmente se ha implementado de la siguiente forma:



(a) Rotación sobre sí mismo



(b) La cámara gira respecto del eje X



(c) Movimiento hacia delante si el usuario baja la cabeza

**Figura 5.9:** Posibles movimientos del robot según el movimiento de las Oculus Rift

Como viene explicado en la imagen, si el usuario mira hacia los lados el robot rotará sobre sí mismo sin moverse del sitio, hacia la dirección que señalen las Oculus Rift.(**exploración**)

Si el usuario mueve la cabeza hacia arriba o hacia abajo sin llegar a menos de 50°, la cámara se moverá como si fuera la cabeza del usuario.(**exploración**)

En el caso de que el usuario oriente la cabeza hacia abajo, con un ángulo menor de 50° respecto al eje Y, la cámara se orientará horizontalmente (90° respecto al eje Y) y el robot avanzará hasta que el usuario cambie la posición de la cabeza.(**transporte**)

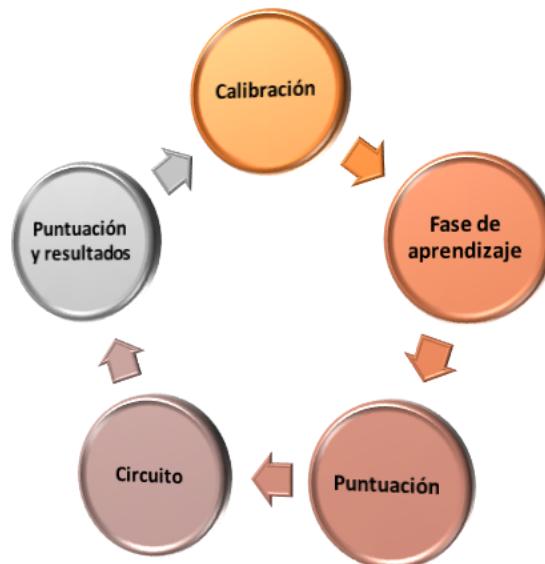


# PRUEBAS Y RESULTADOS

A continuación se explican las pruebas y los resultados obtenidos tras desarrollar el proyecto.

Puesto que el objetivo es crear un dispositivo sencillo de controlar por el usuario, las pruebas constarán de varias fases que se repetirán dentro de un ciclo cerrado. De esta forma el dispositivo podrá ir mejorando conforme a la demanda de los usuarios.

En la siguiente imagen se representan las diferentes fases del ciclo de pruebas.



**Figura 6.1:** Fases del ciclo cerrado de pruebas que se han realizado para valorar la funcionalidad del dispositivo.

### 6.1 Fases del ciclo cerrado

- **Calibración:** Ajustamos los parámetros de velocidad de los motores y sensibilidad de movimientos.
- **Fase de aprendizaje:** Se explica al usuario la interfaz y se deja que la pruebe durante un par de minutos.
- **Puntuación:** El usuario puntuá la interfaz y la facilidad de uso del dispositivo.
- **Círculo:** Se propone al usuario un circuito que debe seguir con el robot.
- **Puntuación y resultados:** Tras realizar el circuito e interaccionar con el dispositivo durante 5/10 minutos el usuario vuelve a puntuar y propone mejoras para la nueva fase de calibración.

### 6.2 Desarrollo de las pruebas

El primer ciclo se realizó con un único usuario ya que la interfaz aún no estaba claramente definida.

Tras este primer ciclo se realizaron dos ciclos más hasta que se consiguieron resultados satisfactorios.

A continuación se explicarán los resultados y mejoras realizadas en cada ciclo.

#### 6.2.1 Primer ciclo

Como ya se ha explicado anteriormente, el primer ciclo de pruebas lo realizó un solo usuario ya que en base a sus valoraciones se modificó la interfaz prácticamente por completo.

**Antes de las pruebas** el robot se controlaba de la siguiente forma:

- **Giros:** El robot giraba sobre sí mismo tantos grados como girara el usuario la cabeza.
- **Avanzar:** El robot avanzaba si el usuario mantenía la cabeza en posición natural, es decir, mirando hacia el frente.
- **Parar:** el robot se detenía cuando el usuario giraba la cabeza, es decir, en fase de exploración.
- **Movimiento de la cámara:** la cámara se movía tanto grados hacia arriba o hacia abajo como grados se moviera el usuario.

**Valoración del usuario** La primera valoración no fue muy positiva. A continuación se muestran las dificultades con las que se encontró el usuario:

- Giros: El usuario puso de manifiesto la dificultad que supone dar la vuelta al robot con la interfaz diseñada para este ciclo ya que el usuario tendría que girar sobre sí mismo.

**Solución:** Modificar el giro de forma que el robot gire con velocidad continua hacia la derecha o izquierda según la orientación de la cabeza del usuario.

- Condición de parada: Otra dificultad con la que se encontró el usuario fue que la forma de parar el robot(girando la cabeza) era muy poco intuitiva. A esto se le añade la dificultad de que el robot esté avanzando por defecto.

**Solución:** Modificar el estado por defecto del dispositivo, en la nueva interfaz el robot está parado por defecto y en caso que esté en movimiento basta con mirar al frente para detenerlo.

Tras realizar las modificaciones da comienzo el segundo ciclo de pruebas.

### 6.2.2 Segundo ciclo

Durante el segundo ciclo de pruebas participaron 7 usuarios. La fase de aprendizaje durante este ciclo consistió en:

- Girar a la derecha
- Girar a la izquierda
- Mirar hacia arriba
- Estar 30 segundos aproximadamente girando el robot sobre sí mismo hasta tener controlado el giro.
- Andar hacia delante
- Parar
- Estar cerca de un minuto paseando para familiarizarse con los movimientos.
- Fijar un objetivo y alcanzarlo.

La siguiente tabla muestra la valoración de los usuarios tras realizar las pruebas.

	Valoración de los usuarios en el segundo ciclo						
	Usuario1	Usuario2	Usuario3	Usuario4	Usuario5	Usuario6	Usuario7
Giro derecha	9	6	6	4	5	5	6
Giro izquierda	7.5	4	6	4	5	5	7
Mirar hacia arriba	9	10	9	10	9	9	9
Andar	10	8	10	8	8	9	9
Parar	9	10	9	7	7	7	8
Alcanzar un objetivo	8.5	6	8	7	5	5	8
Sensación de Control	7.5	6	9	4	5	4	7
Facilidad de aprendizaje	9	8	10	10	9	8	10

**Comentarios tras las pruebas** Analizando las puntuaciones y los comentarios de los usuarios se sacaron las siguientes conclusiones:

Se debe reducir la velocidad de giro ya que es complicado controlarlo a la velocidad actual. La nueva interfaz de control del robot resulta mucho más intuitiva para los usuarios y su sensación de control incrementa en muy pocos minutos.

## 6. PRUEBAS Y RESULTADOS

---

### 6.2.3 Tercer ciclo

A continuación se explicará el desarrollo del tercer y último ciclo de pruebas:

- **Fase de calibración:** Durante esta fase se realizaron los siguientes cambios en base a la opinión de los usuarios que participaron en el segundo ciclo de pruebas:
  - Se disminuyó la velocidad de giro de los servomotores de las ruedas del robot.
  - Se disminuyó la velocidad de avance del robot.
  - Se disminuyó la sensibilidad de detección de movimiento vertical , reduciendo las posibles posiciones que puede tomar la cámara a 4.
    - \* Abajo: 30° aproximadamente respecto a la posición frontal.
    - \* Posición frontal.
    - \* Arriba: 40° aproximadamente respecto a la posición frontal.
    - \* Arriba: 90° aproximadamente respecto a la posición frontal.
- **Fase de aprendizaje:** Se le explicó a cada usuario la forma de controlar el robot y se les dejó durante dos minutos moverse libremente.
- **Puntuación:** A continuación se muestra la primera puntuación de los usuarios.

	Valoración tras la fase de aprendizaje					
	Usuario1	Usuario2	Usuario3	Usuario4	Usuario5	Usuario6
Avanzar	8	7	8	9	6	8
Giro derecha	6	5	6	7	7	6
Giro izquierda	5	5	5	5	7	5
Telepresencia	9	8	8	7	5	6
Sensación de control	7	4	5	6	6	5
Interfaz	9	7	7	9	6	9

- **Círculo:** Los usuarios realizan el mismo circuito 3 veces. En el circuito se marcan 4 puntos críticos y se cronometra el tiempo que tarda cada usuario en alcanzar dichos puntos. También se cronometra el tiempo total.

A continuación se muestra un gráfico donde se muestran los promedios de tiempos entre todos los usuarios de cada intento de hacer el circuito.

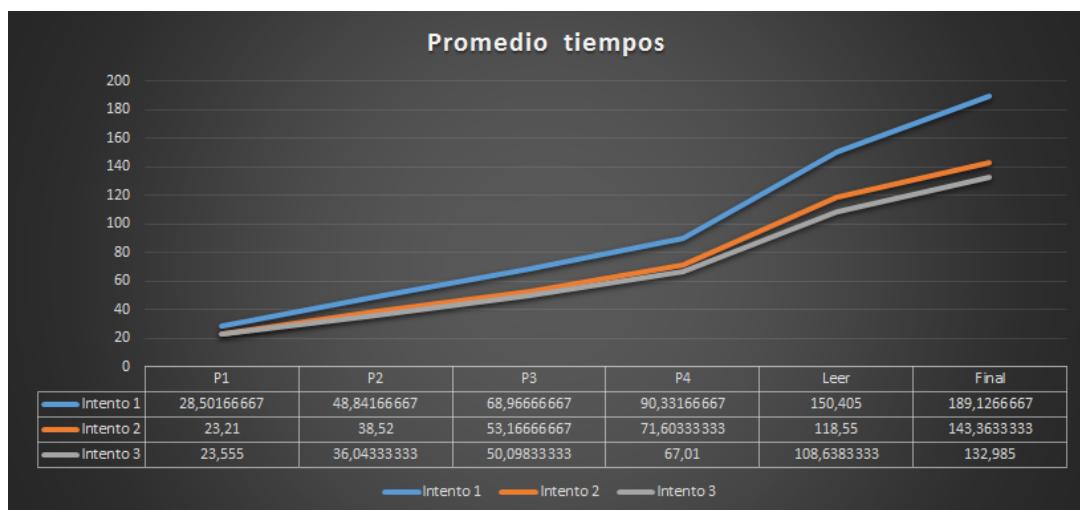


Figura 6.2: Promedio del tiempo que tardan los usuarios en realizar el circuito.

- **Puntuación:** Por último se les vuelve a pedir a los usuarios que valoren el control del robot.

	Valoración tras la fase de aprendizaje					
	Usuario1	Usuario2	Usuario3	Usuario4	Usuario5	Usuario6
Avanzar	8	9	8	6	6	8
Giro derecha	7	7	7	8	6	6
Giro izquierda	7	7	6	8	6	5
Telepresencia	9	8	9	7	6	6
Sensación de control	8	7	7	6	7	5
Interfaz	9	8	7	9	6	9

## 6. PRUEBAS Y RESULTADOS

---

Respecto a la valoración de los usuarios es muy interesante ver cómo varía la sensación de control que tienen los usuarios cuando empiezan a utilizar el dispositivo y cuando ya han realizado 3 veces el circuito. Dicha variación se ve reflejada en el siguiente gráfico.

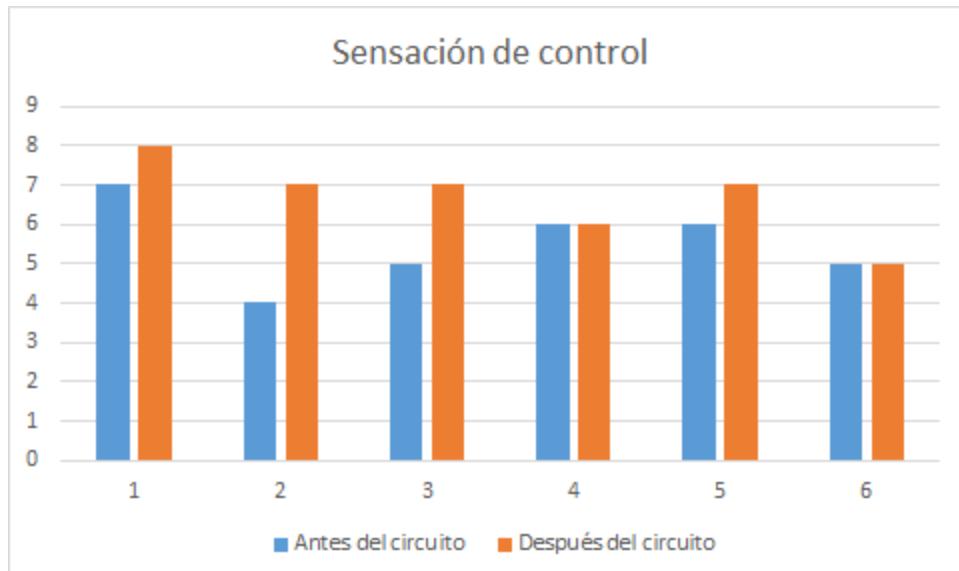


Figura 6.3: Promedio del tiempo que tardan los usuarios en realizar el circuito.

# CONCLUSIONES Y TRABAJO FUTURO

## 7.1 Conclusiones

En este apartado analizaremos los resultados obtenidos, hablaremos de posibles mejores y se expondrán las conclusiones del proyecto.

**Puntos positivos y a mejorar** Tras las pruebas se ve que el mayor defecto del proyecto es la **velocidad de giro** de los motores de las ruedas.

La calibración de servo motores es una tarea que se resuelve de forma experimental. La dificultad que nos encontramos es que la velocidad de giro cambia según en la superficie en la que se encuentre el robot por lo que habría que realizar las pruebas en diferentes entornos y hacer una calibración media.

Los usuarios han estado cada uno menos de 7 minutos con el robot y **todos menos uno lograron alcanzar su objetivo** en menos de un minuto, por lo que es una **aplicación fácil de manejar y de aprender**, como se ha visto en las valoraciones.

Otro problema es el **alcance de red de la Raspberry Pi 3**. Si el robot se alejaba mucho del router los servos seguían recibiendo correctamente los comandos pero el streaming tenía algo más de latencia.

Además la **Raspberry Pi 3 necesita de más amperaje que la Raspberry Pi 2** por lo que se concluye que por ahora, y hasta que se mejore la Raspberry Pi 3, es mejor utilizar Raspberry Pi 2 con un pincho Wifi para poder conectarse a la red local.

De esta forma se consigue mayor alcance de red y se necesita menos Amperios para alientar la placa.

A pesar de todos estos puntos a mejorar el resultado es muy bueno, teniendo en cuenta que es una primera aproximación a lo que pretende ser un proyecto futuro.

Además es un proyecto con una **muy buena acogida**, a todos los usuarios les ha resultado ameno hacer las pruebas y ver cómo iban mejorando.

## 7. CONCLUSIONES Y TRABAJO FUTURO

---

También es fácil de desarrollar con las herramientas de las que disponemos y con **mucho potencial**, ya que incluye dos tecnologías que están en pleno desarrollo y que aún tienen mucho que avanzar.

Otra ventaja es que hace uso de dispositivos de **fácil acceso** por lo que no resultaría demasiado caro de fabricar.

### 7.1.1 Tecnologías aprendidas

Durante la realización de este proyecto he utilizado muchas tecnologías, alguna de ellas completamente nuevas para mí.

El ejemplo más claro de esto son las gafas de realidad virtual Oculus Rift. Ha sido una suerte tener acceso a ellas y poder aprender a utilizar algunas de las librerías que hay disponibles para los desarrolladores.

También ha sido una sorpresa descubrir la gran variedad de posibilidades a desarrollar que permiten las Oculus, como es este proyecto.

He afianzado los conocimientos de robótica que ya tenía. Todo lo que sabía de robótica lo había aprendido en el Club de Robótica de esta universidad y gracias a este trabajo he podido aplicarlos y aumentarlos.

Todo el proyecto está desarrollado en python, lenguaje de programación que no he utilizado en ninguna asignatura de la carrera y que solo había visto muy por encima para proyectos sencillos. Durante este proyecto he aprendido a utilizar las librerías de python disponibles para las Oculus Rift y para el control de servomotores.

La placa base del robot es una Raspberry Pi 3, que tampoco había tenido la oportunidad de utilizarla nunca. De todas formas esto no ha sido novedoso para mí porque sí había participado en algún proyecto sencillo con Raspberry.

En conclusión este trabajo ha sido muy instructivo y ha ampliado de forma significativa mis conocimientos sobre robótica y realidad virtual.

## 7.2 Trabajo futuro

Este trabajo es solo el inicio de lo que podría ser un proyecto muy interesante.

El robot que hemos construido es una extensión del sentido de la vista, pero se podría construir un robot controlado por una persona y que fuera la extensión de todos sus sentidos, permitiendo así conocer y sentir el mundo sin necesidad de moverse de su casa.

Además la realidad virtual es una herramienta muy potente, que no solo te permite modificar tu mundo a placer sino que te premita crear entornos totalmente distintos. Se podría estudiar cómo interactuar con ese mundo virtual no solo a través de la vista, sino con todos los sentidos.

Otro camino por el que se puede investigar es la unión de este proyecto con la domótica. Creo que en un futuro no muy lejano, las casas serán casas domotizadas, que igual que se podrán controlar desde el móvil podremos controlarlas con las gafas de realidad virtual.

# BIBLIOGRAFÍA

- [1] H. G. Hoffman, D. R. Patterson, J. Magula, G. J. Carrougher, K. Zeltzer, S. Dagadakis, and S. R. Sharar. Water-friendly virtual reality pain control during wound care. *Journal of Clinical Psychology*, 60(2):189–195, 2004.
- [2] G. Hubens, H. Coveliers, L. Balliu, M. Ruppert, and W. Vaneerdeweg. A performance study comparing manual and robotically assisted laparoscopic surgery using the da vinci system. *Surgical Endoscopy And Other Interventional Techniques*, 17(10):1595–1599, 2003.
- [3] D. Kocian, A. M. R. L. W.-P. A. OHIO., and A. F. A. M. R. L. (U.S.). *A Visually-Coupled Airborne Systems Simulator (VCASS): An Approach to Visual Simulation*. Aerospace Medical Research Laboratories, 1977.
- [4] D. Lamargue-Hamel, M. Deloire, A. Saubusse, A. Ruet, J. Taillard, P. Philip, and B. Brochet. Cognitive evaluation by tasks in a virtual reality environment in multiple sclerosis. *Journal of the Neurological Sciences*, 359(1–2):94 – 99, 2015.
- [5] R. Lloréns, E. Noé, C. Colomer, and M. Alcañiz. Effectiveness, usability, and cost-benefit of a virtual reality-based telerehabilitation program for balance recovery after stroke: A randomized controlled trial. *Archives of Physical Medicine and Rehabilitation*, 96(3):418 – 425.e2, 2015.
- [6] L. A. Nguyen, M. Bualat, L. J. Edwards, L. Flueckiger, C. Neveu, K. Schwehr, M. D. Wagner, and E. Zbinden. Virtual reality interfaces for visualization and control of remote vehicles. *Autonomous Robots*, 11(1):59–68, 2001.
- [7] I. E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I), pages 757–764, New York, NY, USA, 1968. ACM.

