

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



Grado en ...

TRABAJO FIN DE GRADO

**TÍTULO DEL TFG**

Autor: Nombre Apellido1 Apellido2

Tutor: Nombre Apellido1 Apellido2

Ponente: Nombre Apellido1 Apellido2

MES 20xx



# TÍTULO DEL TFG

Autor: Nombre Apellido1 Apellido2  
Tutor: Nombre Apellido1 Apellido2  
Ponente: Nombre Apellido1 Apellido2

Grupo de la EPS (opcional)  
Dpto. de XXXXX  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
MES 20xx



## 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).

Este trabajo pretende explorar el uso de la Realidad Virtual y la robótica como herramienta para ayudar a que personas con discapacidad motora sean más independientes.

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.

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

## Palabras Clave

**Abstract**

**Key words**

## Agradecimientos





# Índice general

<b>Índice de Figuras</b>	<b>IX</b>
<b>Índice de Tablas</b>	<b>X</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación del proyecto . . . . .	1
1.2. Objetivos y enfoque . . . . .	2
1.2.1. Objetivos . . . . .	2
1.3. Metodología y plan de trabajo . . . . .	3
<b>2. Realidad Virtual y robótica. Estado del arte</b>	<b>5</b>
2.1. Introducción . . . . .	5
2.2. Realidad virtual y salud . . . . .	5
2.3. Robótica y realidad virtual. . . . .	5
2.4. BCIs, EEG y realidad virtual . . . . .	6
<b>3. Análisis, diseño y desarrollo</b>	<b>7</b>
3.1. Análisis . . . . .	7
3.1.1. Construcción del Robot . . . . .	7
3.1.2. Streaming . . . . .	8
3.1.3. Control del Robot . . . . .	8
3.2. Diseño . . . . .	8
3.2.1. Diseño del robot . . . . .	9
3.2.2. Router . . . . .	11
3.2.3. Oculus Rift . . . . .	11
3.2.4. Diseño de conexiones . . . . .	14
3.3. Desarrollo e implementación . . . . .	15
3.3.1. Streaming . . . . .	15
3.3.2. Control del Robot . . . . .	16
3.4. Segmentación . . . . .	18
3.5. Normalización . . . . .	18

3.6. Codificación . . . . .	18
3.7. Matching . . . . .	18
<b>4. Experimentos Realizados y Resultados</b>	<b>19</b>
4.1. Bases de datos y protocolo . . . . .	19
4.2. Sistemas de referencia . . . . .	19
4.3. Escenarios de pruebas . . . . .	19
4.4. Experimentos del sistema completo . . . . .	19
<b>5. Conclusiones y trabajo futuro</b>	<b>21</b>
5.1. Conclusiones . . . . .	21
5.2. Trabajo futuro . . . . .	21
<b>Glosario de acrónimos</b>	<b>23</b>
<b>Bibliografía</b>	<b>24</b>
<b>A. Manual de utilización</b>	<b>27</b>
<b>B. Manual del programador</b>	<b>29</b>

# Índice de Figuras

1.1. Resumen de los objetivos del proyecto . . . . .	3
1.2. Ejemplo pie de figura 2 . . . . .	3
3.1. Raspberry Pi 3 model B . . . . .	9
3.2. Esquema conexiones servos , cámara, Raspberry Pi 3 . . . . .	11
3.3. Ejes de giro . . . . .	12
3.4. Corrección del tilt error . . . . .	13
3.5. Esquema conexiones . . . . .	14
3.6. Uso de CPU del programa Mjpeg-Streamer por el que se hae el streaming de video	16
3.7. Virtual Desktop . . . . .	16
3.8. Virtual Desktop . . . . .	17



## Índice de Tablas



# 1

## Introducción

### 1.1. Motivación del proyecto

---

La motivación de este proyecto es explorar las posibilidades que nos ofrece combinar la robótica con la realidad virtual.

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 o los drones.

La realidad virtual lleva mucho tiempo estudiándose .En 1968 Ivan Sutherland creó el primer casco de realidad virtual.A pesar de ello, hasta hace bien poco, para la gran mayoría de gente era solo ciencia ficción.

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

Este trabajo pretende aplicar estas dos tecnologías para hacer más fácil el día a día de las personas, en concreto nos hemos centrado en las personas con discapacidad motora.

Hemos construido un robot que lleva incorporada una cámara y que se comunica con las Oculus rift.

El usuario puede controlar el movimiento del robot y de la cámara solo con el movimiento de la cabeza, aunque no sería difícil adaptar este control para personas que les sea más fácil controlarlo a través de mandos.

El robot transmite a las Oculus en tiempo real un video del entorno. De esta forma el robot actúa como extensión de los ojos del usuario.

¿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, como modificar el entorno según las necesidades del usuario , meter realidad aumentada para controlar una casa domotizada, etc.

De esta forma daríamos al usuario más independencia y le abrimos un mundo al que hasta ahora, tenía difícil acceso.

Ejemplo de referencia a la bibliografía [1].

## 1.2. Objetivos y enfoque

---

En esta sección se presentarán los objetivos que se pretenden alcanzar en este proyecto y cómo vamos a enfocar el trabajo para conseguirlos.

### 1.2.1. Objetivos

El objetivo final consiste en ser capaces de mover una cámara , controlando el movimiento con la cabeza y recibir en las Oculus Rift a tiempo real todo lo que vea la cámara.

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

#### Streaming

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

Una buena candidata para la transmisión inalámbrica de vídeo a tiempo real al ordenador es a través de una **red local**.

Por lo tanto uno de los primeros objetivos es crear una red local a través de la cual se puedan conectar la cámara y el PC. Además se deberá implementar un software que capture todo lo que vea la cámara y transmita el vídeo a tiempo real por la red local. Por último se necesita un software que recoja el video en el PC.

#### 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.



## Visualización del entorno

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

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.

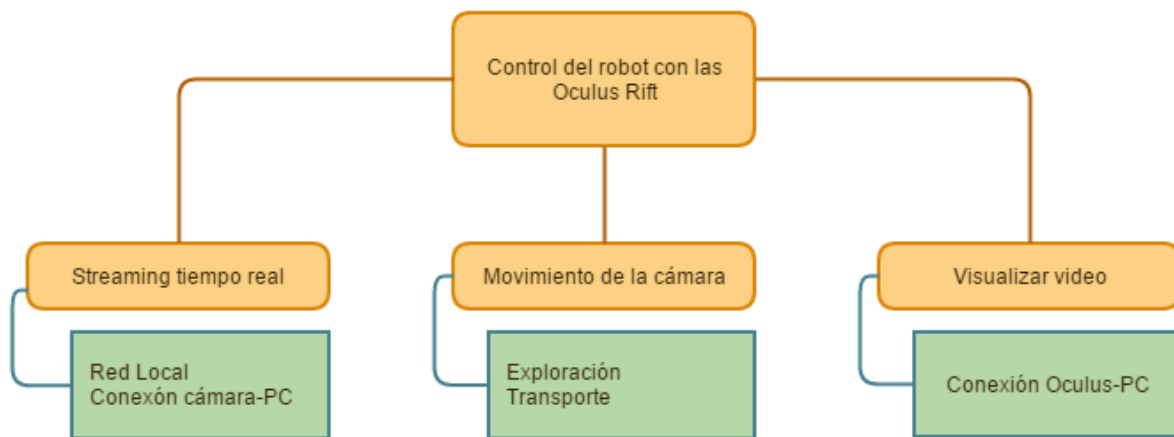


Figura 1.1: Resumen de los objetivos del proyecto

### **1.3. Metodología y plan de trabajo**

---

# 2

## Realidad Virtual y robótica. Estado del arte

### 2.1. Introducción

---

### 2.2. Realidad virtual y salud

---

El uso de realidad virtual está siendo muy útil en medicina, ya que permite estudiar comportamientos en diferentes situaciones de forma bastante realista.

Un ejemplo de esto es la evaluación del deterioro cognitivo en personas que sufren esclerosis múltiple. Para dicha evaluación se requiere 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 se pueden crear entornos virtuales y estudiar el comportamiento de los usuarios en diversas situaciones de forma que el estudio es más realista y eficiente. [ref]

También se está estudiando el uso de programas de rehabilitación para que los pacientes no necesiten trasladarse hasta el hospital.

Uno de los estudios se hizo a personas con hemiparesia, con un software que mostraba por pantalla el movimiento de los pies y les mandaba ejercicios para mejorar su capacidad motora. Los resultados de este estudio mostraron que la herramienta de realidad virtual era efectiva para la rehabilitación y muy útil para los pacientes con dificultades para trasladarse hasta el hospital[ref]

### 2.3. Robótica y realidad virtual.

---

La unión de robótica y realidad virtual es algo que se lleva haciendo desde hace bastante tiempo.

En el campo de la salud una de las aplicaciones son entrenamientos de cirugía. La realidad virtual permite crear un entorno semejante al quirófano. El estudiante hace uso de herramientas que simulan los utensilios quirúrgicos y puede hacer una "operación virtual" sin riesgo para ningún paciente y sin necesidad de gastar recursos o un equipo quirúrgico de apoyo. [ref]

En educación hay varios programas de realidad virtual que hacen el aprendizaje más sencillo y completo a los estudiantes. Un ejemplo de esto es la visualización de brazos robóticos para diseñar correctamente el movimiento. Es relativamente complejo traducir cada movimiento de las articulaciones del brazo robótico en la posición final de éste. Si estas pruebas se hicieran directamente sobre el robot, sería muy sencillo romperlo. Con los programas de realidad virtual se simulan todos los movimientos sin peligro de malgastar recursos. [ref]

Otra aplicación es el control remoto de robots, tanto con software que crea una interfaz virtual para que el usuario obtenga de forma clara toda la información del robot (batería, almacenamiento de datos, posición...) [ref NASA xD], como interfaces BCIs.

## **2.4. BCIs, EEG y realidad virtual**

---

Las interfaces BCIs (Brain Computer Interfaces) aspiran a que el usuario controle el robot a través de pensamientos, al igual que controlamos nuestro cuerpo. Esta aplicación sigue investigándose y se van consiguiendo grandes avances. [ref]??

# 3

## Análisis, diseño y desarrollo

En este capítulo se explicará cómo se ha enfocado el desarrollo del proyecto.

La primera parte trata del análisis del problema. Se explican las diferentes partes en las que he dividido el trabajo para su posterior implementación. También se presenta de forma muy general qué requisitos básicos debe cumplir cada parte.

La sección de diseño explica las herramientas que se han utilizado para implementar cada parte y cómo funcionan.

Por último se explica de forma más detallada cómo se ha desarrollado cada parte del problema, qué dificultades han ido apareciendo y cómo se han solucionado. También explica cómo se ha realizado la integración para conseguir el proyecto final.

### 3.1. Análisis

---

El objetivo de este trabajo es conseguir un robot con una cámara que transmita video a tiempo real. Este video le llegará a unas gafas de realidad virtual. El robot se controlará con el movimiento de cabeza del usuario.

#### DIBUJO CHULI DE TODO JUNTO

El proyecto por lo tanto se divide en 3 problemas:

1. Construcción del robot
2. Streaming
3. Control del robot

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

#### 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 o virtual.

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

- Permitir grabar video y transmitirlo a tiempo real.
- Ser capaz de desplazarse en todas las direcciones
- Tener la capacidad de recibir comandos de control a través de una red inalámbrica
- Ser capaz de que la cámara apunte en la misma dirección que la cabeza del usuario

## DIBUJO AMPLIADO ROBOT

### 3.1.2. Streaming

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

Para ello uno de los requisitos principales es que la transmisión de video sea a tiempo real, con la mínima latencia posible.

Por otra parte 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).

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

Dado que la placa que utilizamos en el robot es la Raspberry Pi 3 tenemos dos opciones para la transmisión inalámbrica: Bluetooth o red Wifi.

### 3.1.3. Control del Robot

Como hemos dicho anteriormente, el robot pretende ser una extensión de los ojos de 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 → Debemos diseñar un sistema intuitivo para mover el robot en todas las direcciones con movimientos de la cabeza.
- Movimiento de la cámara → la cámara debe estar apuntando siempre en la misma dirección que la cabeza.

Si el usuario mueve la cabeza hacia la derecha o hacia la izquierda, el robot se moverá en esa dirección, por lo que no hay que preocuparse por la cámara.

Para controlar los movimientos verticales vamos a incorporar un servo unido a la cámara, de forma que ésta se mueva en función de la posición de la Oculus.

---

## 3.2. Diseño

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

Los principales dispositivos que componen el proyecto son

- Robot
- Router
- Oculus Rift

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

### 3.2.1. Diseño del robot

El robot que se utiliza en este proyecto está basado en un diseño anterior hecho por (Nombre del alumno que diseñó en 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 conectado a la cámara

FOTO ROBOT

### 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.

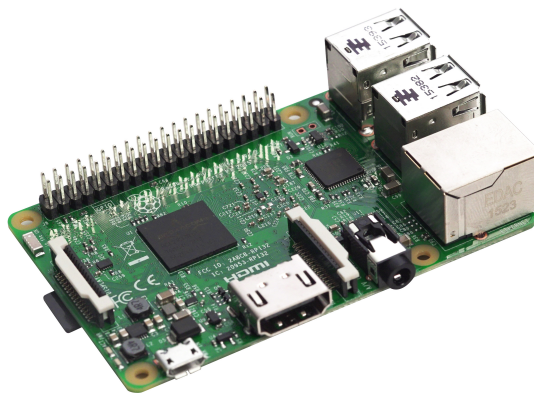


Figura 3.1: Raspberry Pi 3 model B

## Servo motores

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

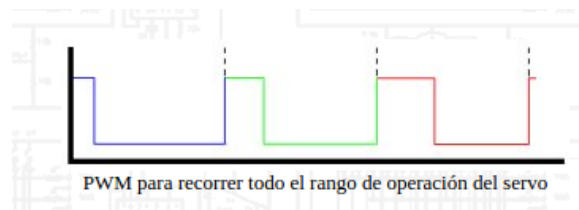
Un servo motor es un motor eléctrico que se puede controlar su velocidad y su posición (dentro del rango de posición permitido).

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 desee.



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 hemos dicho que era 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, variamos la potencia que le entregamos al motor, con lo que controlamos la velocidad de giro con mucha precisión.

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.

## 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.



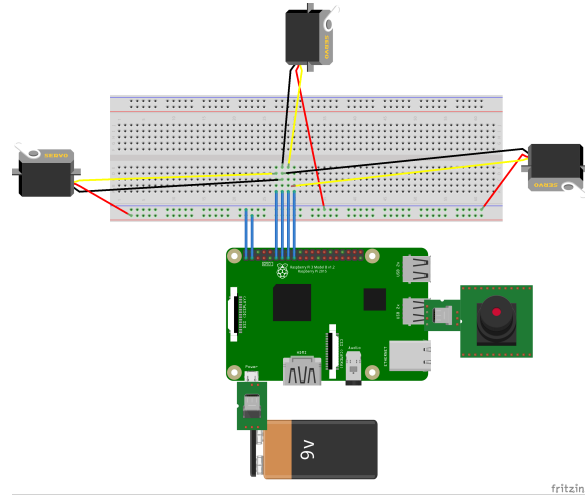


Figura 3.2: Esquema conexiones servos , cámara, Raspberry Pi 3

### 3.2.2. Router

### 3.2.3. Oculus Rift

Las Oculus Rift son unas gafas de Realidad Virtual [parrafo definiendo las Oculus]

De las gafas nos interesan especialmente:

- Detección de la orientación de las Oculus
- Recepción de video

#### Detección de la orientación

Las Oculus Rift tienen integradas un giroscopio, un acelerómetro y un magnetómetro que manda constantemente información al ordenador, de forma que éste sabe en todo momento la orientación de las gafas.

La técnica para interpretar la señal de estos sensores se llama **fusión de sensores**

A continuación se explica en qué consiste esta técnica.

**Fusión de Sensores** Como ya hemos mencionado, en las Oculus tenemos un giroscopio, un acelerómetro y un magnetómetro.

Los dos primeros dan información acerca de la orientación respecto a los ejes X y Z, mientras que el magnetómetro mide la orientación respect al eje Y.

Vamos a ver cómo funciona cada uno.

El **giroscopio** de las Oculus mide el cambio de orientación de la cabeza a una velocidad de 1KHz.

Una forma simplificada de calcular la orientación actual es:

Orientación actual = Orientación anterior + Diferencia horaria · Velocidad observada (giroscopio).

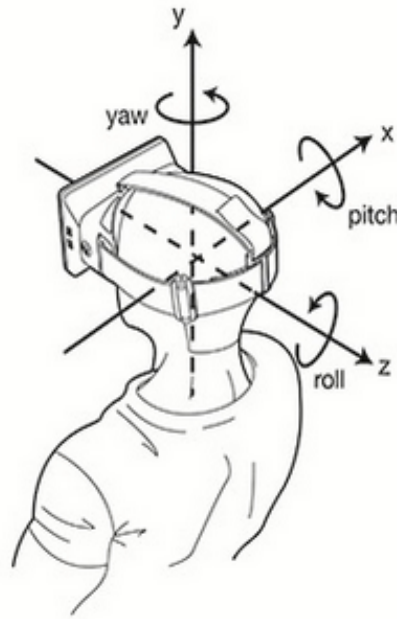


Figura 3.3: Ejes de giro

El problema está en que la cabeza puede rotar alrededor de tres ejes.

Vamos a suponer que solo rotase alrededor del eje Y y que el sensor captase una velocidad angular  $w$  por segundo.

Si tuviéramos 1000 sensores la fórmula de la orientación sería:

$$\text{Orientación actual} = \text{Orientación anterior} + 0,001 \cdot w$$

Pero en el caso de las Oculus, al estar midiendo un objeto que se mueve en 3 ejes, el giroscopio da la velocidad angular respecto de los tres ejes, devolviendo un vector 3D  $(w_1, w_2, w_3)$

Al basar nuestro cálculo en el cálculo anterior, el error irá creciendo con el tiempo.

Vamos a llamar *tilt error* al error en la medición de los ángulos sobre los ejes XZ. El error sobre el eje Y se llamará *yaw error*.

El *tilt error* se corresponde con nuestra sensación de lo que "está arriba", percepción que se basa en la gravedad.

La gravedad se expresa con un vector de aceleración, por lo que utilizamos un **acelerómetro** para medirla.

Nos encontramos con el inconveniente de que el acelerómetro no solo mide la gravedad. Para asegurarnos de que el momento en el que tomamos los datos de referencia solo estamos midiendo la gravedad, esperamos a que se den 2 condiciones:

1. El acelerómetro nos devuelve una medida cercana a 9,8
2. El giroscopio nos devuelve una velocidad angular muy lenta (es decir, no estamos girando).

En este momento sabemos que nos encontramos en una posición vertical y podemos corregir el error.

**Cómo corregimos el error:** Supongamos que se dan las dos condiciones previas, y la posición que nuestro cálculo de orientación nos devuelve un vector  $\vec{a}$ , tal y como vemos en el dibujo.

Tenemos el ángulo  $\theta$  entre el eje Y y el vector  $\vec{a}$  pero ¿cómo calculamos el eje de rotación para rotar  $\vec{a}$  y corregir el error?

Dicho vector debe ser perpendicular a  $\vec{a}$  y al eje Y, y apoyarse en el plano XZ.

Para encontrar el vector basta con hacer la proyección de  $\vec{a}$  en el plano XZ, obteniendo así  $(a_x, 0, a_z)$ . Haciendo un vector perpendicular a éste obtenemos  $(-a_z, 0, a_x)$ . Y ya tenemos el eje de rotación que necesitábamos para corregir el error.

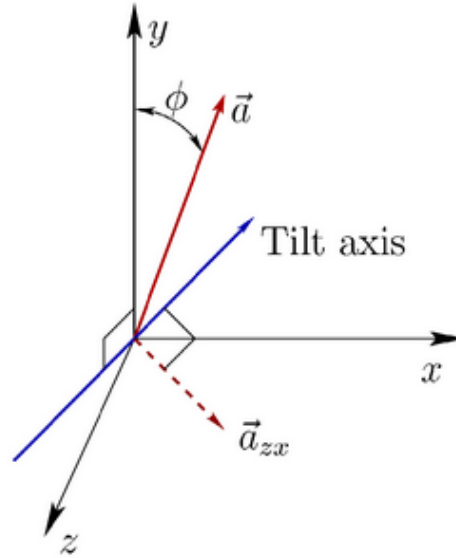


Figura 3.4: Corrección del tilt error

### Error sobre el eje Y (yaw error)

Este error se basa en nuestra percepción de dónde está el norte.

Para esto utilizamos el magnetómetro. COMPLETAR

#### 3.2.4. Diseño de conexiones

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

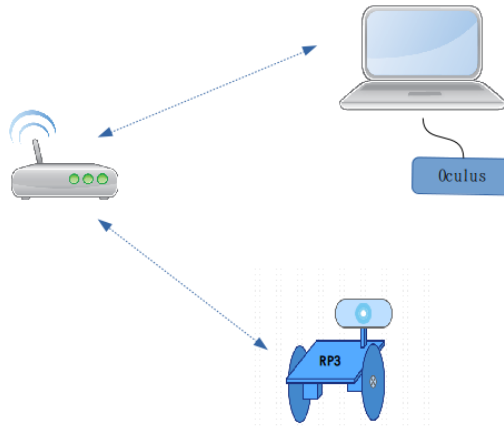


Figura 3.5: Esquema conexiones

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.

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 python que recoge esa información y la traduce en comandos sencillos para mandárselos 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 la cámara.

### 3.3. Desarrollo e implementación

---

En esta sección se explicará cómo hemos desarrollado las distintas funcionalidades del proyecto. 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.

#### 3.3.1. Streaming

##### 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 mjpeg-streamer, disponible en github.

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, Unidad de Procesamiento Gráfico) de la cámara, en nuestro caso H.264 Advanced Video Coding (AVC) Standard, que es el compreso de la Webcam Logitech.

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

El puerto que emplea es el 8080

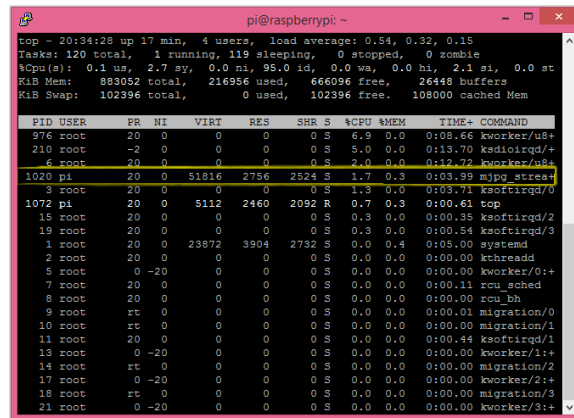


Figura 3.6: Uso de CPU del programa Mjpeg-Streamer por el que se hace el streaming de video

## Ordenador

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

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

El ordenador tiene instalado un programa, Virtual Desktop.

Virtual Desktop es una aplicación desarrollada por Oculus Rift y HTC Vive para utilizar las Oculus Rift en Windows.

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...)



Figura 3.7: Virtual Desktop

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.

IMAGEN DEL VIDEO SBS DE VLC

### 3.3.2. Control del Robot

#### 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` y se lo manda a la raspberry a través de sockets.

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

A continuación se muestra una gráfica que compara el movimiento (en grados) de las Oculus y de un servo motor respecto al tiempo, medido en microsegundos.

Vemos que la latencia entre la medición en las Oculus y la recepción en la Raspberry es muy pequeña.

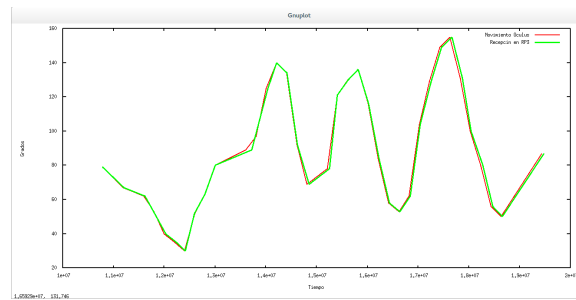


Figura 3.8: Virtual Desktop

## Raspberry Pi3

La Raspberry tiene a su vez 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.

**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 With 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** : es el porcentaje de tiempo que el pulso está arriba. (recordemos que la señal es una señal cuadrada)

Veamos un ejemplo para clarificar esto:

**PWM- Frecuencia:50Hz , DutyCycle 50 %** Si fijamos una frecuencia de 50Hz estaremos mandando una señal de 50 pulsos por segundo, es decir, un pulso cada 0,02 segundos.

Al poner un DutyCycle al 50 % estamos estableciendo que durante esos 0,02 segundos el pulso estará 0,01 segundos en posición "High" el resto del tiempo en posición "Low".

**PWM- Frecuencia:50Hz , DutyCycle 80 %** Si fijamos una frecuencia de 50Hz, como la del ejercicio anterior pero por el contrario ponemos un DutyCycle del 80 % tendremos:

- 1 pulso cada 0,02 segundos
- El pulso estará en posición "high" el 80 % del tiempo (0,016 segundos) y en posición "low" los otros 0,004 segundos restantes.

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

Para transformar este ángulo el pulso(PWM) correspondiente debemos tener en cuenta que un servo requiere una señal PWM con un periodo de 20 ms y un ancho de pulso entre 0,9 y 2,1 ms.

Ya que el servo tiene un rango de ángulos de 0 a 180 grados, es fácil ver que el ancho de pulso de 0,9 ms se corresponde con 0° y 2,1 ms con 180°

---

### 3.4. Segmentación

---

### 3.5. Normalización

---

### 3.6. Codificación

---

### 3.7. Matching

---



# 4

## Experimentos Realizados y Resultados

**4.1. Bases de datos y protocolo**

---

**4.2. Sistemas de referencia**

---

**4.3. Escenarios de pruebas**

---

**4.4. Experimentos del sistema completo**

---



# 5

## Conclusiones y trabajo futuro

### 5.1. Conclusiones

---

Tras realizar el trabajo y probarlo con numerosas personas, concluyo que es un proyecto con una muy buena acogida, 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.

### 5.2. 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 permite 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.



## Glosario de acrónimos

- **IS**: Iris Subject
- **DCT**: Discrete Cosine Transform
- **WED**: Weighted Euclidean Distance



## Bibliografía

- [1] Autor Apellidos. Titulo del artículo. *Revista de publicación*, pages 65–73, 2008.







# Manual de utilización





# Manual del programador