



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Aeroespacial
y Diseño Industrial

Control de trayectoria y velocidad de un vehículo simulado
mediante reconocimiento de la posición de los dedos de la
mano por visión artificial

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y Automática

AUTOR/A: Valls Martos, Beatriz

Tutor/a: Casanova Calvo, Vicente Fermín

CURSO ACADÉMICO: 2022/2023

AGRADECIMIENTOS

Primero, quiero agradecer a mi tutor Vicente por su confianza durante todos estos años.

También, agradezco a mi madre, Joaquín y mi abuelo por nunca dudar de mí y darme el espacio para aprender de mis errores.

A Brandon y Razvan, gracias por su apoyo incondicional y por ser parte esencial en mi vida. Nunca creí en casualidades hasta que os conocí.

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

Documentos del Proyecto

MEMORIA

PLIEGO DE CONDICIONES

PRESUPUESTO

ANEXOS

UNIVERSIDAD POLITECNICA DE VALENCIA

ETSID

**GRADO EN INGENIERIA ELECTRONICA INDUSTRIAL Y
AUTOMATICA**

CONTROL DE TRAYECTORIA Y VELOCIDAD DE UN
VEHÍCULO SIMULADO MEDIANTE RECONOCIMIENTO
DE LA POSICIÓN DE LOS DEDOS DE LA MANO POR
VISIÓN ARTIFICIAL

DOCUMENTO:

MEMORIA

ALUMNO: BEATRIZ VALLS MARTOS

TUTOR: VICENTE FERMÍN CASANOVA CALVO

DOCUMENTO 1. MEMORIA

Índice

INTRODUCCIÓN	8
RESUMEN.....	8
MOTIVACIÓN	8
OBJETIVO	9
ESTADO DEL ARTE.....	10
EVOLUCIÓN DE LA DETECCIÓN DE MOVIMIENTOS HUMANOS	10
PLAYSTATION EYETOY	11
NINTENDO WII	12
PLAYSTATION MOVE.....	13
XBOX KINECT	14
MANUS	14
ELECCIÓN DE HERRAMIENTAS.....	16
LIBRERÍA MEDIAPIPE	16
PYTHON.....	18
MATLAB.....	18
DISEÑO DE LA SOLUCIÓN.....	19
DETECCIÓN DE LOS GESTOS DE LA MANO	20
DETECCIÓN DE LA POSICIÓN DE LOS DEDOS DE LA MANO DERECHA PARA LA DIRECCIÓN DEL VEHÍCULO	20
DETECCIÓN DE LA POSICIÓN DE LOS DEDOS DE LA MANO IZQUIERDA PARA EL CONTROL DE VELOCIDAD	21
ELECCIÓN DE LA CONFIGURACIÓN	21

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

ENVÍO DE LA INFORMACIÓN VÍA UDP	23
SIMULACIÓN DEL VEHÍCULO.....	24
RESULTADOS	26
EVALUACIÓN DEL FUNCIONAMIENTO.....	26
GIRO A LA DERECHA	26
GIRO A LA IZQUIERDA	29
RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030.....	31
CONCLUSIONES	33
TRABAJOS FUTUROS.....	33

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

Índice figuras	
<i>Figura 1 Cámara EYETOY</i>	12
<i>Figura 2 Consola y mando de la Nintendo Wii</i>	13
<i>Figura 3 Controladores de PlayStation Move</i>	13
<i>Figura 4 Cámara de la Xbox Kinect</i>	14
<i>Figura 5 Sensores de detección de movimiento de MANUS</i>	15
<i>Figura 6 Puntos de detección de MediaPipe para una mano</i>	18
<i>Figura 7 Esquema de la solución adoptada</i>	19
<i>Figura 8 Puntos clave de la mano detectados por MediaPipe</i>	20
<i>Figura 9 Detección de la distancia entre los dedos pulgar e índice</i>	21
<i>Figura 10 Diagrama de bloques de la simulación</i>	24
<i>Figura 11 Bloque de la simulación del vehículo</i>	25
<i>Figura 12 Diagrama de bloques del vehículo con Simscape Multibody</i>	25
<i>Figura 13 Velocidad de la rueda izquierda (Gráfica superior) y velocidad de la rueda derecha (Gráfica inferior) en el giro hacia la derecha</i>	26
<i>14 Valores obtenidos en el script de python con las velocidades de cada rueda en un giro a derechas. Primer valor rueda izquierda y segundo valor rueda derecha</i>	27
<i>Figura 15 Velocidad de la rueda izquierda (Gráfica superior) y velocidad de la rueda derecha (Gráfica inferior) en el giro hacia la derecha con mayor ángulo</i>	28
<i>Figura 16 Velocidad de la rueda izquierda (Gráfica superior) y velocidad de la rueda derecha (Gráfica inferior) en el giro hacia la izquierda</i>	29
<i>Figura 17 Valores obtenidos en el script de python con las velocidades de cada rueda en un giro a izquierdas. Primer valor rueda izquierda y segundo valor rueda derecha</i>	30
<i>Figura 18 Velocidad de la rueda izquierda (Gráfica superior) y velocidad de la rueda derecha (Gráfica inferior) en el giro hacia la izquierda con mayor ángulo</i>	31

Índice tablas

Tabla 1 – Objetivos de Desarrollo Sostenibles 32

1. INTRODUCCIÓN

1.1. RESUMEN

En este informe se presenta un proyecto dedicado a la detección de diversos gestos de la mano con el propósito de controlar una simulación en Matlab. El objetivo es lograr el control de la velocidad y el giro de la simulación de un vehículo de dos ruedas, mediante una adecuada programación de las velocidades específicas para este tipo de vehículo.

Los próximos capítulos del trabajo abordarán de manera detallada las características del proyecto y las etapas fundamentales del diseño. Se profundizará en la elección de los gestos específicos para el control, así como la configuración necesaria para realizar giros a ambos lados y ajustar la velocidad tanto hacia arriba como hacia abajo.

Asimismo, se examinará el procesamiento y tratamiento de los datos obtenidos a partir del detector de gestos, así como el envío eficiente de esta información hacia la simulación. Se describirá en detalle la implementación llevada a cabo para estudiar la viabilidad y los resultados de la solución planteada.

Finalmente, se presentarán las conclusiones derivadas de los resultados obtenidos durante el desarrollo del proyecto, evaluando la efectividad y funcionalidad de la solución propuesta en el control del vehículo de dos ruedas a través de la detección de gestos de la mano.

1.2. MOTIVACIÓN

La realización de este proyecto surge de una fuerte motivación por explorar la detección de gestos como medio para controlar diversos artefactos.

Los gestos, como expresiones corporales, tienen una relevancia significativa en diferentes culturas, ya que comunican información importante, incluso sobre el estado de ánimo de las personas. A lo largo de los años, ha habido un notable avance en la investigación de la detección de gestos, especialmente en el ámbito de los videojuegos, donde se han logrado avances significativos.

La intención de este trabajo es abordar un estudio más profundo, centrándose en la simulación de control de un vehículo. La idea subyacente es que este enfoque pueda sentar las bases para futuras aplicaciones prácticas, donde la detección de gestos pueda aplicarse en prototipos de vehículos reales.

La oportunidad de explorar esta tecnología abre un amplio abanico de

TRABAJO FINAL DE GRADO INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

posibilidades, como mejorar la interacción entre humanos y máquinas, facilitar la accesibilidad en diversos contextos, e incluso enriquecer la experiencia de usuario en diferentes dispositivos y sistemas.

Con esta motivación, se busca contribuir al avance del campo de detección de gestos y explorar cómo su implementación en la simulación de control de vehículos puede representar un paso adelante hacia la materialización de aplicaciones prácticas en un futuro próximo.

1.3. OBJETIVO

El presente trabajo tiene como objetivo principal el diseño e implementación de un sistema de detección de gestos, el cual será evaluado y probado en una simulación de un vehículo de dos ruedas.

El proyecto se enfoca en cumplir con los siguientes requisitos fundamentales:

- Desarrollar un sistema de detección de gestos capaz de reconocer y distinguir entre los dedos levantados y agachados, con el propósito de controlar la dirección del vehículo. Para lograr esto, se buscará implementar algoritmos y técnicas de visión por computadora que permitan interpretar y procesar los movimientos de los dedos de manera precisa y eficiente.
- Diseñar una funcionalidad en el sistema que permita detectar la distancia entre el dedo pulgar e índice, lo cual posibilitará el control de la velocidad del vehículo. Esta característica requerirá un análisis cuidadoso de los datos obtenidos para garantizar una medición fiable y adecuada para el control de la simulación.
- Integrar un mecanismo de comunicación por medio de sockets, que permita la transmisión de los datos capturados por el detector de gestos hacia la simulación del vehículo. Esta comunicación en tiempo real es esencial para que los gestos detectados sean interpretados y respondidos adecuadamente por el sistema simulado.
- Desarrollar una simulación del vehículo de dos ruedas que esté debidamente preparada para recibir y obedecer las órdenes de los gestos detectados por el sistema. La simulación debe ser capaz de interpretar y actuar en consecuencia a los comandos recibidos, proporcionando una experiencia realista y precisa en función de los gestos detectados.

2. ESTADO DEL ARTE

2.1. EVOLUCIÓN DE LA DETECCIÓN DE MOVIMIENTOS HUMANOS

En la era actual, caracterizada por un constante avance tecnológico y una creciente interacción entre humanos y máquinas, el reconocimiento de gestos ha surgido como una prometedora área de investigación y desarrollo. El fundamento clave de esta tecnología radica en la habilidad de interpretar y comprender el lenguaje corporal humano, permitiendo una comunicación más fluida y natural entre las personas y los dispositivos electrónicos.

Imaginemos por un momento el poder de un simple gesto: cerrar el puño y extender el dedo pulgar. De manera instantánea, se establece una comunicación clara y directa, transmitiendo un mensaje inequívoco de aprobación o satisfacción. Durante años, los expertos se han dedicado a desentrañar los secretos detrás de estos gestos y a desarrollar algoritmos y sistemas informáticos capaces de reconocer y comprender su significado.

En las últimas dos décadas, se han producido notables avances en el campo del reconocimiento de gestos, lo que ha llevado a la creación de dispositivos interactivos que operan exclusivamente a través de estos movimientos corporales. Estos avances han generado un enorme interés debido a su amplio espectro de aplicaciones potenciales, tanto en el ámbito comercial como en el científico.

Entre los dispositivos más fascinantes se encuentran aquellos que se utilizan en entornos de realidad virtual y aumentada. Mediante la detección y seguimiento preciso de los gestos del usuario, estos dispositivos permiten una inmersión total en mundos virtuales, donde los movimientos del cuerpo se traducen en acciones y respuestas dentro del entorno simulado. Desde jugar videojuegos envolventes hasta simular experiencias de entrenamiento o terapia, la tecnología de reconocimiento de gestos ha abierto un amplio abanico de posibilidades en estos campos.

Otra área de aplicación interesante es la interacción hombre-máquina en entornos cotidianos, como la domótica y los dispositivos inteligentes del hogar. Los gestos pueden convertirse en una forma intuitiva y natural de controlar el encendido y apagado de luces, regular la temperatura ambiente, ajustar los sistemas de sonido o incluso interactuar con electrodomésticos inteligentes. Esta tecnología ofrece una experiencia más ergonómica y accesible, eliminando la necesidad de teclados o mandos a distancia.

Además, el reconocimiento de gestos tiene el potencial de revolucionar el campo de la medicina, especialmente en la rehabilitación física. Al capturar y

analizar los movimientos corporales de los pacientes, los profesionales de la salud pueden evaluar y monitorear con precisión el progreso de la terapia, brindando una atención personalizada y eficiente. Esto abre las puertas a nuevas formas de tratamiento y recuperación, optimizando los resultados y mejorando la calidad de vida de los pacientes.

En la actualidad, el mercado ofrece una variada selección de productos que permiten la interacción a través de gestos, y su diversidad radica en los métodos utilizados para capturar y reconocer dichos movimientos. Algunos dispositivos periféricos se encargan de registrar los gestos realizados por el usuario, mientras que otros emplean cámaras especializadas para capturarlos, tal como lo hemos implementado en nuestro proyecto. A continuación, describiremos los productos más populares de esta categoría.

2.1.1. PLAYSTATION EYETOY

El PlayStation EyeToy fue un revolucionario dispositivo desarrollado por Sony Computer Entertainment para su consola PlayStation 2. Lanzado en el año 2003, el EyeToy consistía en una cámara especializada que permitía a los jugadores interactuar con los videojuegos mediante gestos y movimientos corporales, sin necesidad de utilizar controladores convencionales.

La aparición de esta herramienta supuso un hito importante en la historia de los videojuegos y la tecnología de reconocimiento de gestos. Por primera vez, los jugadores podían abandonar los tradicionales mandos y controlar el juego utilizando su propio cuerpo como interfaz. Esto brindaba una experiencia de juego más inmersiva y activa, rompiendo las barreras físicas entre el jugador y la pantalla.

El EyeToy revolucionó la forma en que los jugadores interactuaban con los videojuegos, al proporcionar una experiencia de juego más intuitiva y natural. Los jugadores podían realizar movimientos físicos reales, como golpear, esquivar, saltar o bailar, y el dispositivo los traducía en acciones dentro del juego. Esto abrió nuevas posibilidades en términos de jugabilidad y entretenimiento, creando una experiencia más activa y participativa para los jugadores.

Además de su impacto en el mundo de los videojuegos, el PlayStation EyeToy también tuvo repercusiones en otros campos. Fue utilizado en aplicaciones de fitness y ejercicio, permitiendo a los usuarios realizar rutinas de entrenamiento frente a la pantalla, con el dispositivo registrando y evaluando sus movimientos. También se exploraron usos educativos, como programas interactivos que promovían el aprendizaje y la coordinación motriz en niños. En la Figura 1 ver la cámara EYETOY.



Figura 2 Cámara EYETOY

2.1.2. NINTENDO WII

El Nintendo Wii fue una consola de videojuegos que destacó por su enfoque innovador y único en 2006. Su lanzamiento significó un cambio significativo en la forma en que las personas interactuaban con los videojuegos al introducir un enfoque novedoso basado en el reconocimiento de gestos.

El impacto del Wii en la industria de los videojuegos fue extraordinario. La consola se convirtió en un éxito de ventas a nivel mundial, atrayendo tanto a jugadores habituales como a nuevos públicos, incluyendo a personas que previamente no mostraban interés por los videojuegos. La facilidad de uso y la simplicidad de los controles basados en gestos del Wii capturaron la atención de una amplia variedad de jugadores, desde personas mayores hasta familias y grupos sociales más amplios.

El Wii se destacó por su enfoque en la jugabilidad basada en gestos, lo cual abrió un nuevo mundo de posibilidades en la forma en que disfrutábamos de los juegos. Desde deportes simulados como tenis, golf o boxeo, hasta juegos de aventuras y fiestas que fomentaban la interacción entre los jugadores, la consola ofrecía experiencias de juego innovadoras y emocionantes. En la Figura 2 observamos la consola junto con el mando que detecta los movimientos.

La aparición del Wii también influyó a otros fabricantes de la industria de los videojuegos, quienes comenzaron a adoptar tecnologías similares de reconocimiento de gestos en sus consolas y dispositivos. Esto demostró el impacto significativo que tuvo el enfoque del Wii en la forma en que las personas interactuaban con los videojuegos. Su legado se ve reflejado en las consolas y plataformas actuales, que continúan utilizando controles basados en movimientos.



Figura 4 Consola y mando de la Nintendo Wii

2.1.3. PLAYSTATION MOVE

La Nintendo Wii tuvo tanto éxito que las empresas de consolas empezaron a investigar sobre mejoras para la detección de gestos. Tras todas estas investigaciones, en 2010 Sony lanzó PlayStation Move.

A diferencia de los mandos tradicionales de la marca, el PlayStation Move consistía en un controlador de movimiento con forma de varita, equipado con sensores de movimiento y una esfera luminosa en el extremo como se puede ver en la Figura 3. Estos componentes permitían al sistema detectar y rastrear los movimientos precisos de los jugadores en el espacio tridimensional.



Figura 6 Controladores de PlayStation Move

El sistema PlayStation Move atrajo a una amplia base de jugadores, desde aficionados casuales hasta jugadores más experimentados, y abrió nuevas posibilidades para el juego en grupo y la diversión familiar. Además, su compatibilidad con otros periféricos de PlayStation, como la cámara PlayStation Eye, mejoraba aún más la precisión y la inmersión en los juegos.

2.1.4. XBOX KINECT

Microsoft también hizo lo propio e investigo acerca del reconocimiento de gestos, y de ahí apareció Kinect que se convirtió en uno de los controladores de juegos más vendidos del momento. A diferencia de Nintendo Wii y PlayStation move este no requería de ningún mando para el reconocimiento de gestos.

A través de una combinación de cámaras y sensores de profundidad, como se observa en la Figura 4, el Kinect era capaz de rastrear y reconocer los movimientos del cuerpo del jugador en tiempo real. Esto permitía una experiencia de juego completamente manos libres, donde los jugadores podían controlar y participar en los juegos simplemente moviéndose y realizando gestos frente al sensor.



Figura 8 Cámara de la Xbox Kinect

El impacto del Xbox Kinect en la industria de los videojuegos fue significativo. Ofrecía una forma más natural e inmersiva de jugar, eliminando la necesidad de aprender comandos o pulsar botones, y permitiendo a los jugadores sumergirse por completo en la experiencia del juego. Desde juegos deportivos y de baile hasta aventuras y juegos familiares, el Kinect ofrecía una amplia variedad de títulos que aprovechaban al máximo su tecnología de reconocimiento de gestos.

2.1.5. MANUS

El Manus es un sistema de control avanzado que ha revolucionado la interacción en entornos virtuales. Se trata de un conjunto de guantes de alta precisión diseñados para capturar y transmitir los movimientos de las manos y

TRABAJO FINAL DE GRADO INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

los dedos del usuario a través de sensores y tecnología de seguimiento.

Este dispositivo permite a los usuarios controlar de manera intuitiva y realista sus acciones en entornos virtuales, ofreciendo una experiencia inmersiva y precisa. El Manus utiliza sensores de alta calidad para capturar los movimientos de las manos en tiempo real, lo que permite un seguimiento preciso de la posición, la orientación y los gestos de los dedos. Estos sensores se observan en la Figura 5.



*Figura 10 Sensores de detección de movimiento de
MANUS*

La aparición del Manus ha supuesto un avance significativo en la interacción en realidad virtual y aumentada. Gracias a su tecnología de seguimiento de alta fidelidad, los usuarios pueden manipular objetos virtuales con gran precisión, realizar gestos y expresiones naturales, e incluso interactuar con otros usuarios en entornos virtuales compartidos.

3. ELECCIÓN DE HERRAMIENTAS

Existen diversas herramientas disponibles para abordar la detección de gestos, cada una con sus propias características y enfoques. Desde la implementación de algoritmos personalizados de aprendizaje automático y aprendizaje profundo, que implican el desarrollo desde cero de modelos utilizando conjuntos de datos seleccionados, hasta el aprovechamiento de bibliotecas previamente entrenadas que ofrecen una mayor eficiencia en términos de tiempo y recursos. Entre las opciones de bibliotecas, destacan OpenCV y MediaPipe, las cuales brindan soluciones sólidas y confiables.

En esta ocasión, nuestra elección se inclina hacia la utilización de la biblioteca MediaPipe, por diversas razones fundamentales que se expondrán en este apartado.

Además de seleccionar la herramienta de detección de gestos adecuada, se debía considerar una plataforma que permitiera simular el comportamiento de un automóvil y fuera capaz de recibir la información proporcionada por MediaPipe para seguir las instrucciones adecuadas. En este caso, la elección se inclina hacia Matlab, junto con su poderosa biblioteca Simulink. Matlab ofrece una amplia gama de herramientas y funcionalidades para la simulación de sistemas complejos, y su integración con Simulink permite una representación y visualización precisa de la dinámica del automóvil. Esta combinación de MediaPipe y Matlab con Simulink proporciona un entorno sólido y eficiente para la implementación y prueba de soluciones de detección de gestos en un contexto de simulación de automóviles.

Las herramientas utilizadas van a ser las siguientes:

3.1. LIBRERÍA MEDIPIPE

MediaPipe utiliza una combinación de técnicas de aprendizaje automático y visión por computadora para detectar y rastrear manos en tiempo real. El proceso de detección de manos en MediaPipe consta de varios pasos:

1. Detección de puntos clave: MediaPipe utiliza una red neuronal convolucional (CNN) entrenada para detectar y localizar puntos clave en la mano, como las puntas de los dedos, la base de los dedos y el centro de la palma. Esta etapa inicial identifica las regiones de la imagen que contienen la mano.
2. Seguimiento de puntos clave: Una vez que se detectan los puntos clave iniciales, se utiliza un algoritmo de seguimiento para seguir los puntos clave a medida que la mano se mueve en el video. Esto garantiza una detección precisa y continua de la mano, incluso cuando está en movimiento.

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

3. Estimación de pose de la mano: Con los puntos clave detectados y rastreados, MediaPipe puede estimar la pose de la mano, es decir, la configuración y orientación de los dedos y la palma. Esto permite una comprensión más detallada de la posición y gestos de la mano.

Es importante destacar que MediaPipe se basa en redes neuronales pre-entrenadas y modelos de aprendizaje automático para realizar estas tareas de detección y seguimiento de manos. Estos modelos se han entrenado previamente en grandes conjuntos de datos que contienen imágenes y videos etiquetados con información sobre la ubicación y la pose de las manos. Al inferir en tiempo real sobre los datos de entrada, MediaPipe utiliza estos modelos para identificar y rastrear las manos en movimiento en el video capturado por la cámara.

En la implementación predeterminada de MediaPipe, se detectan y rastrean 21 puntos clave en una mano. Estos puntos clave representan las partes principales de la mano, incluyendo las puntas de los dedos, las articulaciones de los dedos, la base de los dedos y el centro de la palma.

Los 21 puntos clave detectados por MediaPipe (Figura 6) en una mano son los siguientes:

1. Muñeca
2. Pulgar (articulación 1)
3. Pulgar (articulación 2)
4. Pulgar (articulación 3)
5. Pulgar (punta)
6. Índice (articulación 1)
7. Índice (articulación 2)
8. Índice (articulación 3)
9. Índice (punta)
10. Medio (articulación 1)
11. Medio (articulación 2)
12. Medio (articulación 3)
13. Medio (punta)
14. Anular (articulación 1)
15. Anular (articulación 2)
16. Anular (articulación 3)
17. Anular (punta)
18. Meñique (articulación 1)
19. Meñique (articulación 2)
20. Meñique (articulación 3)
21. Meñique (punta)

Estos puntos clave se utilizan para representar la posición y la pose de la mano, lo que permite realizar diversas tareas de seguimiento, reconocimiento de gestos y manipulación de objetos en aplicaciones interactivas y de realidad

aumentada.

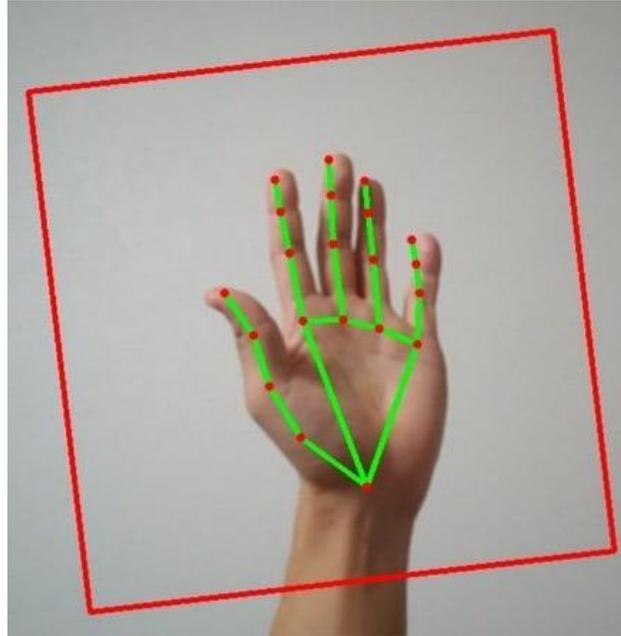


Figura 12 Puntos de detección de MediaPipe para una mano

3.2. PYTHON

Se ha tomado la decisión de emplear el lenguaje de programación Python como base fundamental para nuestro proyecto, debido a su destacada eficiencia y su excelente compatibilidad con la biblioteca MediaPipe.

3.3. MATLAB

En nuestro proyecto de simulación del automóvil, hemos tomado la decisión de emplear MATLAB junto con la biblioteca Simulink como herramientas fundamentales. Esta elección se fundamenta en la poderosa funcionalidad que ofrece MATLAB, así como en la versatilidad y capacidades de diseño que proporciona Simulink. Ambas herramientas combinadas nos permiten desarrollar una amplia gama de prototipos, siendo en este caso un vehículo el enfoque principal.

Además, la integración de MATLAB y Simulink con UDP (User Datagram Protocol) nos permite recibir información de manera eficiente. Este método de comunicación se utilizará para transmitir las órdenes desde Python hasta MATLAB. UDP es un protocolo de red eficiente y rápido que nos brinda una comunicación de datos confiable y de bajo retardo, lo cual es crucial para lograr una interacción fluida y en tiempo real entre los dos entornos.

4. DISEÑO DE LA SOLUCIÓN

La solución adoptada se presenta visualmente en la Figura 7, donde se ilustra de manera gráfica la arquitectura general del sistema implementado. En este apartado, cada bloque del esquema será abordado en el mismo orden en el que aparecen, a fin de proporcionar una comprensión exhaustiva de cada componente clave involucrado en el proceso. Dichos componentes incluyen el Script de detección de las manos, el Script para el envío de la información a través del protocolo UDP y, por último, la simulación del vehículo mediante el uso de la plataforma Matlab.

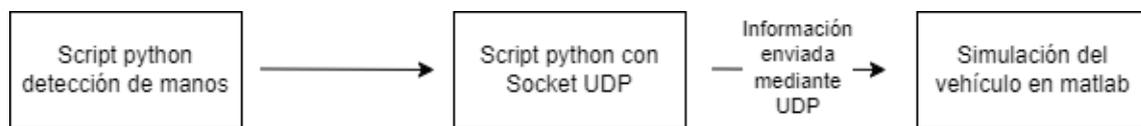


Figura 14 Esquema de la solución adoptada

El resumen es el siguiente:

El primer bloque, el Script de detección de las manos, despliega un conjunto de algoritmos y técnicas basadas en la biblioteca MediaPipe para detectar y rastrear los movimientos de las manos en tiempo real. Esta parte del sistema se encarga de procesar las imágenes o secuencias de video de entrada, utilizando algoritmos de aprendizaje profundo y técnicas de visión por computadora para identificar con precisión la posición y los gestos de las manos.

El segundo bloque consiste en el Script para el envío de la información a través del protocolo UDP. Una vez que se ha realizado la detección de las manos, este script se encarga de enviar los datos relevantes a través de la red utilizando el protocolo UDP (User Datagram Protocol). Este enfoque permite una transmisión rápida y eficiente de la información hacia el destino deseado, asegurando una comunicación fluida y en tiempo real.

Por último, el bloque de simulación del vehículo vía Matlab permite recrear y visualizar de manera realista el comportamiento del vehículo en función de los gestos detectados. Utilizando la plataforma Matlab, se construye un entorno virtual donde se simulan las interacciones del vehículo con base en los datos de detección de las manos recibidos previamente. Esta simulación permite evaluar y perfeccionar el desempeño del sistema, así como explorar diferentes escenarios y situaciones de conducción.

4.1. DETECCIÓN DE LOS GESTOS DE LA MANO

En el contexto de la detección de gestos de las manos, se ha desarrollado un script en Python que permite detectar la posición de los dedos de ambas manos, tal como se ha mencionado en puntos anteriores. Dado que cada mano tiene un papel específico en el control del vehículo, la implementación del script se ha dividido en dos partes, una para la mano izquierda, encargada de controlar la dirección del vehículo, y otra para la mano derecha, que controla la velocidad de desplazamiento.

4.1.1. DETECCIÓN DE LA POSICIÓN DE LOS DEDOS DE LA MANO DERECHA PARA LA DIRECCIÓN DEL VEHÍCULO

En la primera parte del script, se aborda la detección de la posición de los dedos índice, corazón, anular y meñique de la mano izquierda. Para lograr esto, se obtienen las coordenadas de los puntos 8, 12, 16 y 20, los cuales corresponden a las puntas de los dedos mencionados. Estas coordenadas se comparan con las coordenadas de los puntos 5, 9, 13 y 17, que representan las últimas articulaciones de cada dedo, como se observa en la Figura 8. Para determinar si los dedos están levantados o agachados, se realiza una resta de coordenadas. Si el resultado de la resta es positivo, indica que los dedos están levantados, mientras que un resultado negativo señala que los dedos están agachados.

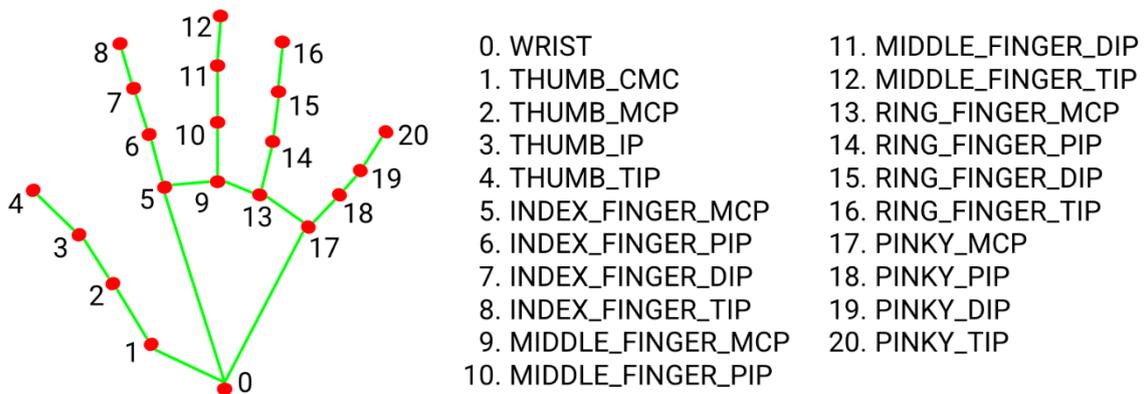


Figura 16 Puntos clave de la mano detectados por MediaPipe

4.1.2. DETECCIÓN DE LA POSICIÓN DE LOS DEDOS DE LA MANO IZQUIERDA PARA EL CONTROL DE VELOCIDAD

En la segunda parte del script, se aborda el control de la velocidad del vehículo. Para ello, se calcula la distancia entre los puntos 8 y 4 de la mano derecha. Estos puntos corresponden a las posiciones superiores de los dedos índice y pulgar respectivamente. Cuanto más cercanos estén estos puntos, menor será la velocidad del vehículo, y a medida que se alejen, la velocidad aumentará. En la Figura 9 tenemos un ejemplo de esta detección.

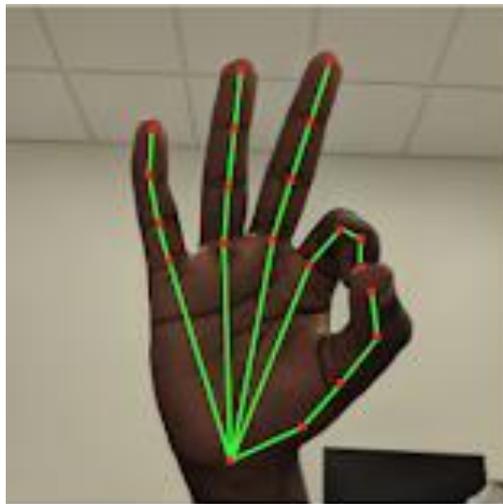


Figura 18 Detección de la distancia entre los dedos pulgar e índice

4.1.3. ELECCIÓN DE LA CONFIGURACIÓN

En el proceso de selección de los gestos a ser monitorizados para controlar la velocidad y dirección del vehículo, se ha tenido en cuenta la comodidad y facilidad de ejecución de dichos gestos. Se reconoce que realizar movimientos complejos, como levantar simultáneamente el dedo meñique y anular, puede resultar complicado para muchas personas. Por lo tanto, se ha establecido una relación específica entre los gestos y la dirección del vehículo, considerando la simplicidad y la facilidad de ejecución de cada gesto. A continuación, se detalla la relación establecida:

1. Para desplazar el vehículo hacia la izquierda con un ángulo pequeño, se ha asignado el gesto de levantar el dedo meñique. Este gesto resulta intuitivo y cómodo de realizar para la mayoría de las personas.

TRABAJO FINAL DE GRADO INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

2. En el caso de desplazar el vehículo hacia la izquierda con un ángulo mayor, se ha establecido el gesto de levantar los dedos meñique e índice simultáneamente. Este gesto, aunque requiere una ligera complejidad adicional, sigue siendo manejable para la mayoría de los usuarios.
3. Para mover el vehículo hacia la derecha con un ángulo pequeño, se ha asignado el gesto de levantar el dedo índice. Este gesto es sencillo y puede ser ejecutado sin dificultad.
4. Finalmente, para desplazar el vehículo hacia la derecha con un ángulo mayor, se ha establecido el gesto de levantar los dedos índice y corazón al mismo tiempo. Aunque este gesto requiere un poco más de destreza, sigue siendo manejable para la mayoría de los usuarios.

Como se verá más adelante, el vehículo implementado consiste en un modelo de dos ruedas, por esta razón, a fin de lograr el desplazamiento del vehículo en una dirección específica, es fundamental que las ruedas del mismo giren a velocidades diferentes. Dicha diferencia de velocidad permite controlar la trayectoria del vehículo, disminuyendo la velocidad de la rueda correspondiente al sentido hacia el cual se pretende dirigir y aumentando, en igual medida, la velocidad de la otra rueda. Por ejemplo, si se desea mover el vehículo hacia la izquierda, la rueda izquierda debe girar a una velocidad menor que la rueda derecha, y viceversa si se desea dirigir hacia la derecha.

Consecuentemente, en el script de control, se lleva a cabo un proceso de escalado del valor de la velocidad, ajustándolo en un rango de 0 a 1. A continuación, este valor escalado se multiplica por factores específicos, dependiendo de la dirección hacia la cual se pretende desplazar el vehículo. En caso de que se desee mover el vehículo hacia adelante, se aplica el mismo factor de multiplicación a ambas ruedas, garantizando así que ambas ruedas giren a la misma velocidad y el vehículo se desplace en línea recta.

Por el contrario, si se desea dirigir el vehículo hacia la derecha con un ángulo menor, se multiplica la velocidad de la rueda derecha por un factor de 0.75, mientras que la velocidad de la rueda izquierda se multiplica por un factor de 1.25. Si se busca una dirección hacia la derecha con un ángulo mayor, se multiplicaría la velocidad de la rueda derecha por un factor de 0.5 y la velocidad de la rueda izquierda por un factor de 1.5. El mismo principio se aplica si se pretende dirigir el vehículo hacia la izquierda, utilizando los correspondientes factores de multiplicación para cada rueda.

El resultado de la implementación de los algoritmos descritos previamente es la generación de un array de 2 dimensiones que contiene los valores correspondientes a cada una de las ruedas del vehículo. Estos valores, que representan las velocidades ajustadas de las ruedas, son transmitidos mediante

el protocolo UDP hacia el entorno de Matlab, donde se llevará a cabo la simulación del vehículo.

En el Anexo 1 adjunto a este informe, se proporciona el código utilizado para la detección de la posición de los dedos y la implementación del socket encargado de enviar la información a través del protocolo UDP.

4.2. ENVÍO DE LA INFORMACIÓN VÍA UDP

En el proceso de transferencia de datos desde el script anterior hacia el entorno de Matlab, se utiliza el protocolo UDP (User Datagram Protocol) a través de un socket creado en Python. Antes de profundizar en el uso de UDP, es importante comprender qué es un socket.

Un socket es una abstracción de software que proporciona una interfaz de programación para la comunicación entre procesos, tanto en la misma máquina como a través de una red. Actúa como un punto final de una conexión, permitiendo el intercambio de datos entre diferentes aplicaciones o dispositivos. Los sockets se utilizan comúnmente en aplicaciones de red para establecer comunicación entre un cliente y un servidor.

En el caso específico de este proyecto, se crea un socket en Python para enviar los datos recolectados hacia el entorno de Matlab mediante el protocolo UDP. UDP es un protocolo de transporte orientado a datagramas, lo que significa que los datos se dividen en paquetes individuales llamados datagramas y se envían de manera independiente. A diferencia del protocolo TCP (Transmission Control Protocol), UDP no establece una conexión persistente ni realiza un control de errores exhaustivo.

La elección de UDP en lugar de TCP para la transmisión de los datos se debe a una consideración particular del proyecto. En este caso, no se requiere una confirmación de entrega o una respuesta para cada paquete enviado, ya que la pérdida ocasional de paquetes no afecta significativamente el funcionamiento general del sistema. Ejemplos similares en los que se utiliza el protocolo UDP son los videos en streaming, donde la prioridad es mantener la reproducción fluida, aunque se pierdan algunos paquetes en el camino.

El código implementado se encuentra en el Anexo 1 junto con el código de la detección de los gestos de la mano.

4.3. SIMULACIÓN DEL VEHÍCULO

Para la realización de la simulación, como se mencionó previamente, se empleará el entorno de programación MATLAB junto con Simulink. La elección del vehículo de 2 ruedas se basa en su idoneidad para el propósito de esta simulación, las configuraciones de las velocidades de cada rueda, citadas en secciones anteriores, están específicamente adaptadas para este tipo de vehículos, lo cual permite su adecuado funcionamiento y maniobrabilidad. El diagrama de bloques que representa la estructura de la simulación se presenta en la Figura 10.

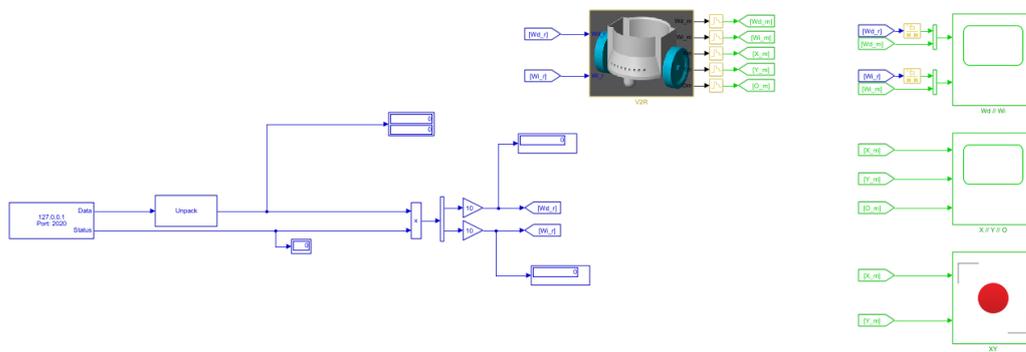


Figura 20 Diagrama de bloques de la simulación

En el inicio del proceso de simulación, se puede observar el bloque receptor UDP, que es responsable de recibir los datos enviados desde el entorno de programación Python. Una vez que estos datos son recibidos, se procede a desempaquetarlos para obtener las velocidades correspondientes. Cabe destacar que, debido a consideraciones específicas del vehículo a simular, cada valor de velocidad es multiplicado por un factor de 10, con el propósito de adecuar dichas velocidades al vehículo en cuestión.

Las velocidades resultantes después del proceso de ajuste son las que se utilizan en el bloque donde se simula cada componente del vehículo. En las Figuras 11 y 12 se pueden observar los bloques que conforman la estructura del vehículo en la simulación desarrollada con la librería Simscape Multibody, detallando cada una de sus partes y cómo interactúan entre sí.

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

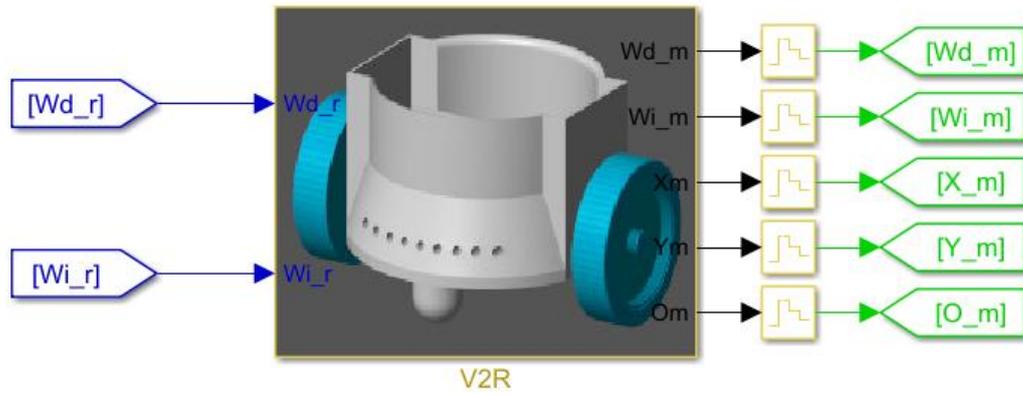


Figura 22 Bloque de la simulación del vehículo

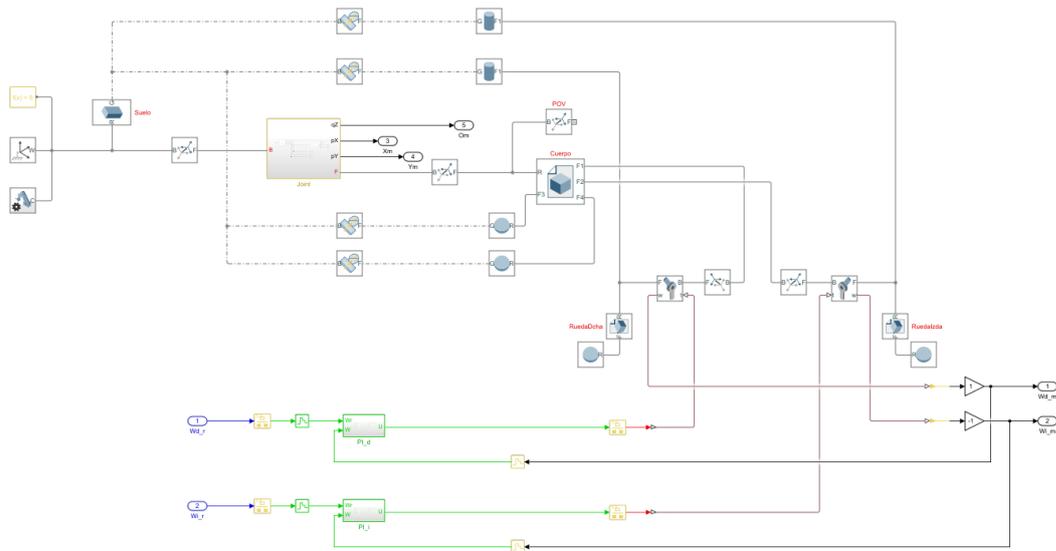


Figura 24 Diagrama de bloques del vehículo con Simscape Multibody

5. RESULTADOS

En el presente apartado, se procederá al análisis de los resultados obtenidos durante varias simulaciones realizadas. En primer lugar, se mostrarán los resultados obtenidos en una simulación en la que se indicó al vehículo que gire únicamente hacia la derecha, mediante la modificación de la velocidad. A continuación, se presentará otra simulación idéntica, pero esta vez se indicó al vehículo que mantuviera una dirección constante hacia la izquierda.

5.1. EVALUACIÓN DEL FUNCIONAMIENTO

5.1.1. GIRO A LA DERECHA

Durante la primera prueba, se llevó a cabo el levantamiento del dedo índice, ajustando la velocidad en función de dicho gesto, lo que provocó que el vehículo se moviera exclusivamente hacia la derecha. En la Figura 13 se observan dos gráficas que representan las velocidades de ambas ruedas. Se evidencia que, a pesar de las variaciones en la velocidad debido a la distancia entre el dedo índice y el pulgar, la velocidad de la rueda izquierda (gráfica superior) siempre es mayor que la de la rueda derecha (gráfica inferior), lo que resulta en el giro hacia la derecha del vehículo.

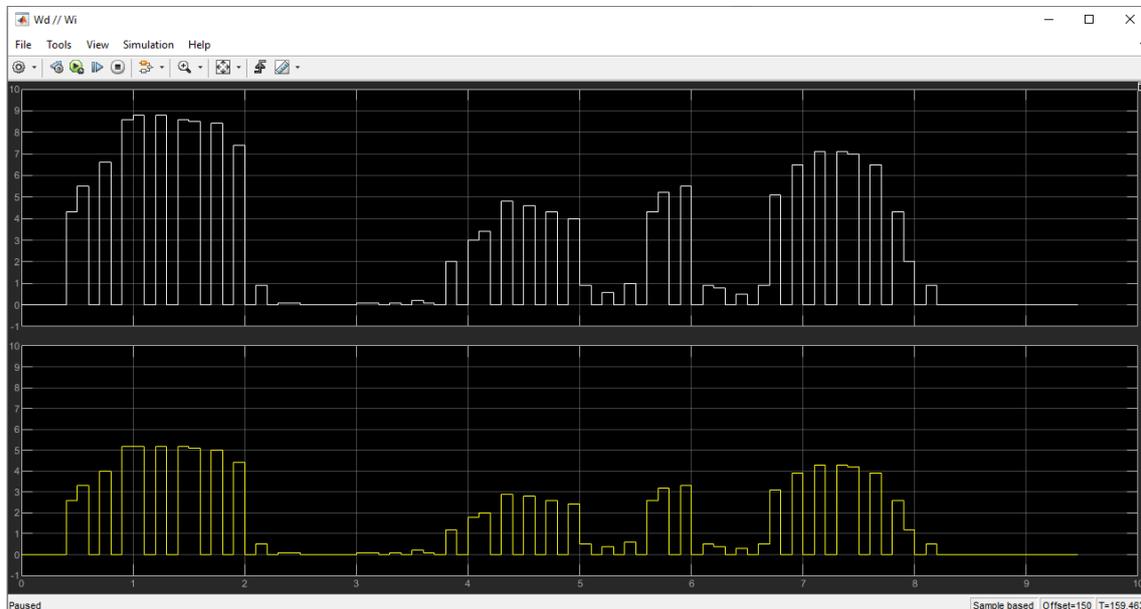


Figura 26 Velocidad de la rueda izquierda (Gráfica superior) y velocidad de la rueda derecha (Gráfica inferior) en el giro hacia la derecha

La Figura 14 muestra la salida generada por los scripts de Python, donde el primer valor del arreglo corresponde a la velocidad de la rueda izquierda y el

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

segundo valor a la rueda derecha.

```
[0.85 0.51]
[0.84 0.5 ]
[0.74 0.44]
[0.09 0.05]
[0.01 0.01]
[0.01 0.01]
[0. 0.]
[0. 0.]
[0.01 0.01]
[0.01 0.01]
[0.01 0.01]
[0.02 0.02]
[0.01 0.01]
[0.2 0.12]
[0.3 0.18]
[0.34 0.2 ]
[0.48 0.29]
[0.46 0.28]
[0.43 0.26]
[0.4 0.24]
[0.09 0.05]
[0.06 0.04]
[0.1 0.06]
[0.43 0.26]
[0.52 0.32]
[0.55 0.33]
[0.09 0.05]
[0.08 0.04]
[0.05 0.03]
[0.09 0.05]
[0.51 0.31]
[0.65 0.39]
[0.71 0.43]
[0.71 0.43]
[0.7 0.42]
[0.65 0.39]
[0.43 0.26]
[0.2 0.12]
[0.09 0.05]
[0. 0.]
```

Figura 28 Valores obtenidos en el script de python con las velocidades de cada rueda en un giro a derechas. Primer valor rueda izquierda y segundo valor rueda derecha

A continuación, procederemos a examinar las mismas pruebas anteriores, pero esta vez levantando el dedo índice y corazón, lo cual implica, como ya se mencionó en puntos previos, que la velocidad de la rueda derecha se

TRABAJO FINAL DE GRADO INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

multiplicará por un factor de 0.5, mientras que la velocidad de la rueda izquierda se multiplicará por 1.5. Esta ajustada asimetría en las velocidades de las ruedas influirá en el comportamiento del vehículo durante la simulación.

Al analizar la Figura 15, se puede observar claramente que la velocidad de la rueda izquierda (gráfica superior) es tres veces mayor que la velocidad de la rueda derecha (gráfica inferior). Esta discrepancia provoca que el vehículo gire hacia la derecha con un ángulo más pronunciado en comparación con las pruebas anteriores.

Este efecto se produce debido a que las velocidades desiguales de las ruedas provocan una asimetría en las fuerzas aplicadas a cada lado del vehículo, lo que resulta en un mayor momento de giro hacia la derecha. La rueda izquierda, al experimentar una mayor velocidad, genera una fuerza de rotación más significativa, inclinando la dirección del vehículo hacia ese lado en mayor grado que la prueba anterior.

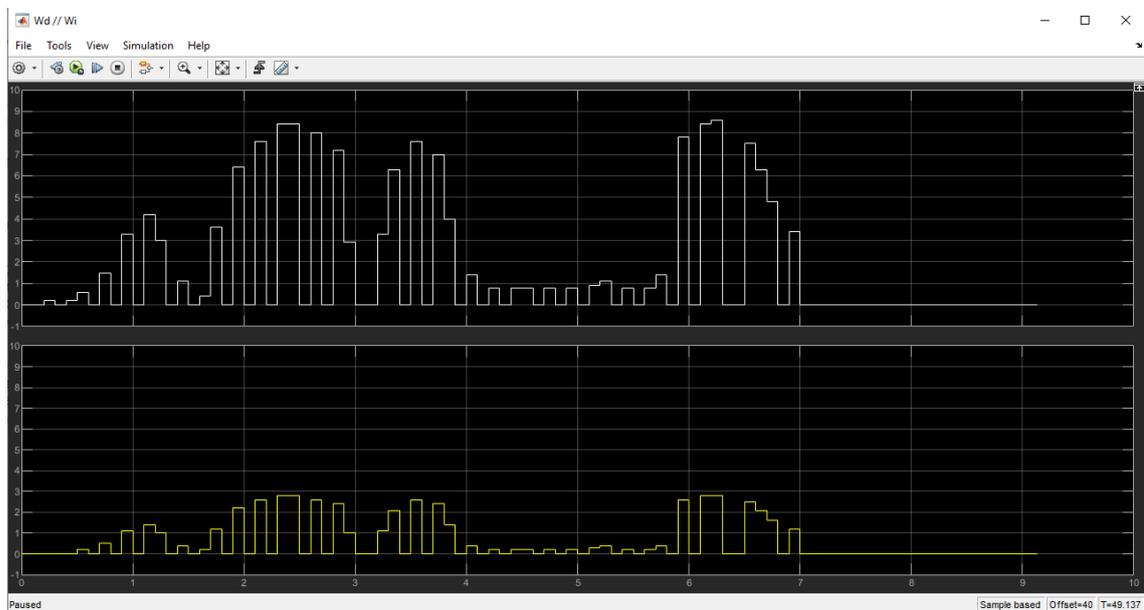


Figura 30 Velocidad de la rueda izquierda (Gráfica superior) y velocidad de la rueda derecha (Gráfica inferior) en el giro hacia la derecha con mayor ángulo

5.1.2. GIRO A LA IZQUIERDA

En la siguiente prueba, se realizó el levantamiento del dedo pulgar, al igual que en la prueba anterior, y se procedió a modificar la velocidad. En esta ocasión, el vehículo giró hacia la izquierda, tal como se muestra en la Figura 16, donde la velocidad de la rueda izquierda (gráfica superior) se mantuvo siempre por debajo de la velocidad de la rueda derecha (gráfica inferior).

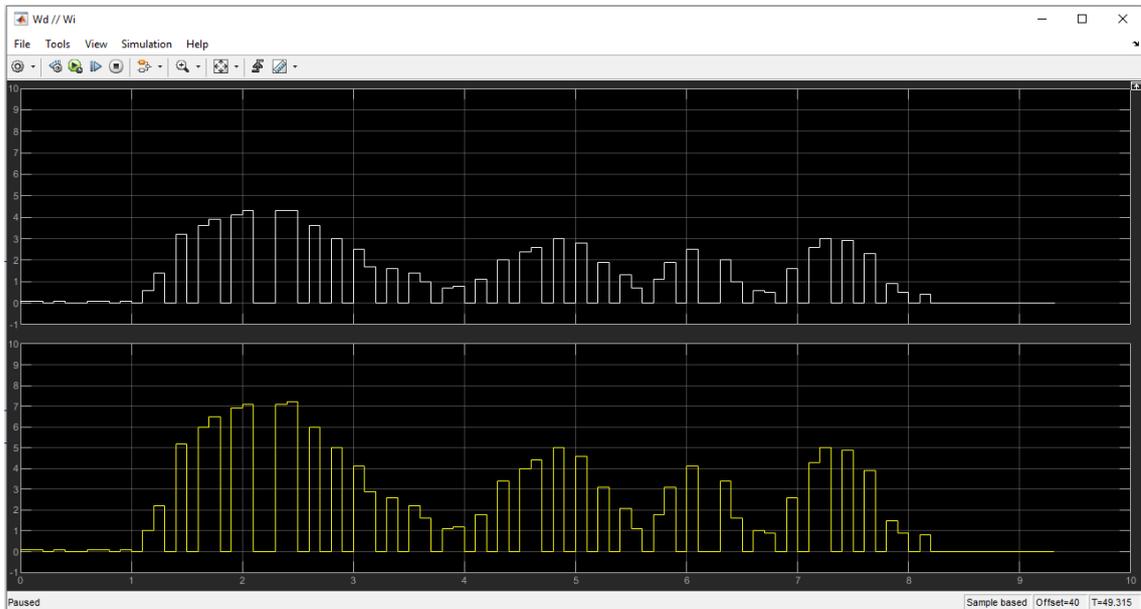


Figura 32 Velocidad de la rueda izquierda (Gráfica superior) y velocidad de la rueda derecha (Gráfica inferior) en el giro hacia la izquierda

Nuevamente, la Figura 17 presenta la salida generada por los scripts de Python, manteniendo la misma convención que en el caso anterior, donde el primer valor del array corresponde a la velocidad de la rueda izquierda y el segundo valor a la rueda derecha.

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

```
[0.32 0.52]  
[0.36 0.6 ]  
[0.39 0.65]  
[0.41 0.69]  
[0.43 0.71]  
[0.43 0.71]  
[0.43 0.72]  
[0.36 0.6 ]  
[0.3 0.5]  
[0.25 0.41]  
[0.17 0.29]  
[0.16 0.26]  
[0.14 0.22]  
[0.1 0.16]  
[0.07 0.11]  
[0.08 0.12]  
[0.11 0.18]  
[0.2 0.34]  
[0.24 0.4 ]  
[0.26 0.44]  
[0.3 0.5]  
[0.28 0.46]  
[0.19 0.31]  
[0.13 0.21]  
[0.07 0.11]  
[0.11 0.18]  
[0.19 0.31]  
[0.25 0.41]  
[0.2 0.34]  
[0.1 0.16]  
[0.06 0.1 ]  
[0.05 0.09]  
[0.16 0.26]  
[0.26 0.43]  
[0.3 0.5]  
[0.29 0.49]  
[0.23 0.39]  
[0.09 0.15]  
[0.05 0.09]  
[0.04 0.08]
```

Figura 34 Valores obtenidos en el script de python con las velocidades de cada rueda en un giro a izquierdas. Primer valor rueda izquierda y segundo valor rueda derecha

Las siguientes pruebas han sido llevadas a cabo con el gesto de levantar el dedo índice y meñique, lo cual tiene un impacto significativo en el ángulo de giro hacia la izquierda del vehículo. Como se mencionó previamente, este gesto provoca una modificación en las velocidades de las ruedas, multiplicando la

TRABAJO FINAL DE GRADO

INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

velocidad de la rueda izquierda por 0.5 y la velocidad de la rueda derecha por 1.5.

El resultado de estas modificaciones se aprecia claramente en la gráfica X, donde se observan las velocidades de ambas ruedas. La velocidad de la rueda izquierda (gráfica superior) se ve reducida a la mitad de su valor original, mientras que la velocidad de la rueda derecha (gráfica inferior) experimenta un aumento del 50%.

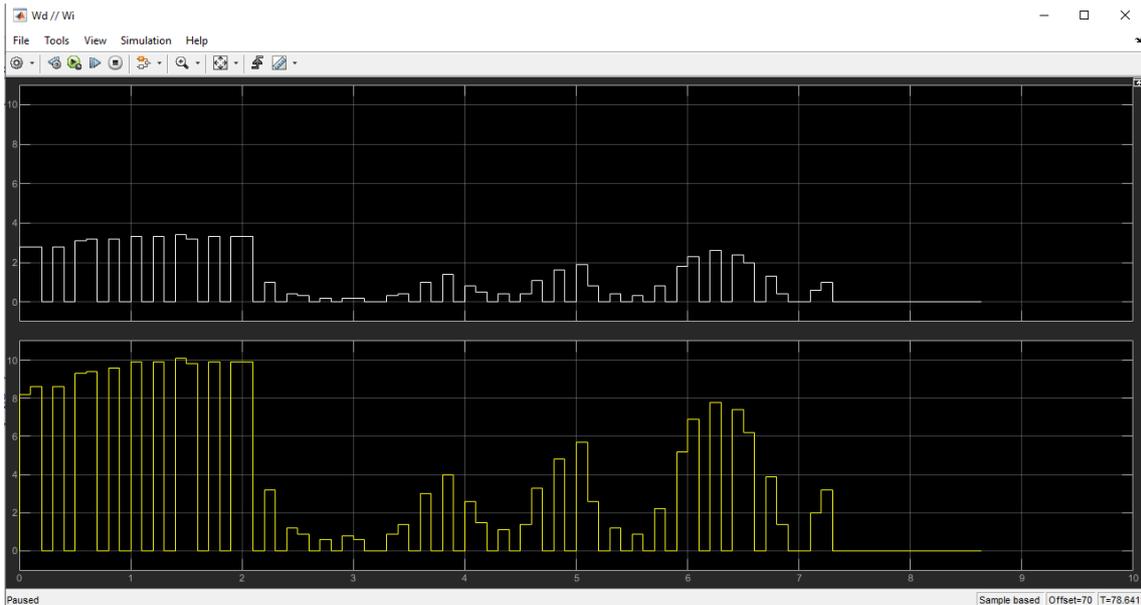


Figura 36 Velocidad de la rueda izquierda (Gráfica superior) y velocidad de la rueda derecha (Gráfica inferior) en el giro hacia la izquierda con mayor ángulo

5.2. RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030

En la siguiente tabla se detalla la relación de este trabajo con los Objetivos de Desarrollo Sostenible (ODS) de la agenda 2030.

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

Tabla 1 – Objetivos de Desarrollo Sostenibles de personal

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No procede
ODS1. Fin de la pobreza				X
ODS2. Hambre cero				X
ODS3. Salud y bienestar				X
ODS4. Educación de calidad				X
ODS 5. Igualdad de género				X
ODS 6. Agua limpia y saneamiento				X
ODS 7. Energía asequible y no contaminante				X
ODS 8. Trabajo decente y crecimiento económico				X
ODS 9. Industria, innovación e infraestructuras	X			
ODS 10. Reducción de las desigualdades				X
ODS 11. Ciudades y comunidades sostenibles				X
ODS 12. Producción y consumo responsables				X
ODS 13. Acción por el clima				X
ODS 14. Vida submarina				X
ODS 15. Vida de ecosistemas terrestres				X
ODS 16. Paz, justicia e instituciones sólidas				X
ODS 17. Alianzas para lograr objetivos				X

El contenido de este proyecto corresponde con un sistema de detección de gestos para el control de la simulación de un vehículo, lo cual se relaciona con el ODS 9, Industria, innovación e infraestructuras.

6. CONCLUSIONES

Basándonos en los resultados presentados en el apartado anterior, podemos afirmar que el diseño de la aplicación para la detección de gestos y la simulación del automóvil son funcionalmente correctos. Las herramientas desarrolladas han demostrado su capacidad para detectar los gestos realizados con la mano y enviarlos de manera satisfactoria a través de UDP hacia MATLAB. Asimismo, MATLAB, al recibir esta información mediante el bloque específico de receptor UDP en Simulink, ha logrado transmitirla de manera efectiva a la simulación del vehículo, que ha respondido adecuadamente a todos los movimientos generados por los gestos.

Estos resultados son un indicador alentador que sugiere que el análisis realizado en el contexto de la simulación ha sido efectivo. En consecuencia, se puede concluir que es factible progresar en este tipo de proyectos, incluso considerando el uso de maquetas reales de automóviles. La posibilidad de experimentar en un entorno más real y tangible resulta prometedora, ya que permitiría abordar escenarios más cercanos a la realidad y evaluar el comportamiento del vehículo en condiciones similares a las que se encontraría en situaciones prácticas.

6.1. TRABAJOS FUTUROS

A partir de las conclusiones extraídas, se vislumbran diversas mejoras y ampliaciones que podrían enriquecer el proyecto en cuestión:

Implementación en un entorno real: Resulta sumamente interesante llevar a cabo la etapa de prototipado y experimentar las simulaciones previamente realizadas en un contexto más realista. Este enfoque permitiría evaluar el comportamiento del sistema bajo condiciones más cercanas a situaciones prácticas de uso. Al abordar la implementación real, se podrán identificar y resolver desafíos específicos asociados con el funcionamiento en el mundo físico, lo que potencialmente dará lugar a mejoras y optimizaciones significativas.

Ampliación de gestos: En el presente trabajo, se ha tenido en cuenta un conjunto de cuatro gestos para el control del vehículo. Sin embargo, sería sumamente enriquecedor experimentar con otros tipos de gestos o movimientos de la mano con el objetivo de estudiar su eficiencia en el control del vehículo. Esta ampliación permitiría evaluar la viabilidad de diferentes gestos y determinar cuáles son más intuitivos y prácticos para el usuario.

Implementación en otros sistemas: Siguiendo la línea de aumentar los gestos, se propone extender este trabajo para su implementación en otros entornos, como el ámbito de la domótica. La aplicación de este sistema de detección de gestos en la automatización del hogar podría permitir el control de

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

diversos utensilios y dispositivos mediante gestos realizados con una o ambas manos. Esta aplicación tendría un gran potencial para mejorar la interacción y la experiencia del usuario en su entorno doméstico.

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

UNIVERSIDAD POLITECNICA DE VALENCIA

ETSID

**GRADO EN INGENIERIA ELECTRONICA INDUSTRIAL
Y AUTOMATICA**

CONTROL DE TRAYECTORIA Y VELOCIDAD DE UN
VEHÍCULO SIMULADO MEDIANTE RECONOCIMIENTO
DE LA POSICIÓN DE LOS DEDOS DE LA MANO POR
VISIÓN ARTIFICIAL

DOCUMENTO:

PLIEGO DE CONDICIONES

ALUMNO: BEATRIZ VALLS MARTOS

TUTOR: VICENTE FERMÍN CASANOVA CALVO

Documento 2. Pliego de condiciones

Índice

INTRODUCCIÓN	4
CONDICIONES DE ESPECIFICACIONES TÉCNICAS	5
HARDWARE.....	5
SOFTWARE	6
INSTALACIONES	¡ERROR! MARCADOR NO DEFINIDO.

Índice de tablas

Tabla 1 - Especificaciones servidor local..... 6

1. INTRODUCCIÓN

En este documento se presenta el pliego de condiciones del proyecto objeto de estudio. En él se analizarán las especificaciones técnicas y requisitos a cumplimentar, en especial aquello referente al marco legal en el que se engloba el proyecto y a los recursos utilizados durante su realización.

2. CONDICIONES DE ESPECIFICACIONES TÉCNICAS

El proyecto objeto de este documento está regido por la Normativa de Marco de Trabajos De Fin de Grado y Fin de Máster de la Universidad Politécnica de Valencia, aprobada el día 13 de marzo del año 2018.

Esta normativa nace a consecuencia del Real Decreto 1393/2007, por el que se establece la ordenación de las enseñanzas universitarias oficiales, modificado por el Real Decreto 861/2010 que dispone, con carácter general, que todos los títulos oficiales “concluirán con la elaboración y defensa” de un Trabajo Fin de Grado (TFG) o Trabajo Fin de Máster (TFM), según el caso. Adicionalmente las órdenes ministeriales por las que se establecen los requisitos que deben cumplirse para la verificación de los títulos oficiales que habilitan para las profesiones de Arquitecto, Arquitecto Técnico, Ingeniero o Ingeniero Técnico también incluyen prescripciones generales relativas a la naturaleza de los TFG o TFM, condiciones para su presentación y defensa y, en algún caso, composición del tribunal calificador.

2.1. Hardware:

Para la elaboración del presente proyecto no ha sido necesaria la utilización de hardware específico más allá de un ordenador personal con las especificaciones técnicas necesarias para el empleo del software detallado en el siguiente apartado.

Para la utilización del vehículo desarrollado e implementado una vez alcanzada la fase de implantación es necesario disponer de dos ordenadores, uno con la capacidad suficiente para ejercer como servidor local (características expuestas en la tabla 1 – Especificaciones del servidor local), y otro para acceder al software del sistema desde un navegador, desde el que el operario accederá al mando de control del vehículo, este ordenador, no requiere especificaciones técnicas concretas, solamente disponer de un navegador y conexión a la misma red en que se aloje el sistema de gestión que conecta al robot con el servidor local.

Tabla 1 - Especificaciones servidor local

Procesador:	Tecnología Intel o AMD con arquitectura de 32 o 64 bits.
	Mínima frecuencia de reloj: 1.2GHz
	Mínimo 4 núcleos y 8 Hilos de procesamiento
Memoria RAM:	Capacidad mínima de 8 GB
Disco duro:	Capacidad mínima de 50 GB libres
Conectividad:	Puerto Ethernet
Software:	Windows 8 o superior / Windows server 2012 o superior
	Arquitectura de 32 o 64 bits

En este documento no se contemplan las especificaciones técnicas concretas necesarias para la implementación real del vehículo.

2.1. Software:

A nivel de software, para la realización del proyecto, tal y como se explica en el apartado 4. Elección de herramientas del documento I: Memoria, se ha requerido la utilización del siguiente software específico:

- **Librería Mediapipe:** En su versión 0.10.2 con licencia gratuita por ser una librería de código libre. Se ha utilizado para la detección de los gestos de las manos.
- **Python:** En su versión 3.7.3. Se ha utilizado en la programación del detector de gestos y en el servidor para mandar los datos vía UDP.
- **MATLAB:** En su versión R2021b con licencia académica proporcionada por la Universidad Politécnica de Valencia.
- **Paquete Office 365:** Con licencia académica proporcionada por la Universidad Politécnica de Valencia. Utilizado para realizar la redacción del presente documento.

2.2. Instalaciones:

Debido a que el proyecto presentado no consta de un entorno físico implementado no es necesario disponer de instalaciones para su correcto funcionamiento. Sin embargo, en la fase de implementación se deberá disponer de un entorno que cumpla los siguientes requisitos:

- **Espacio:** se debe contar con el espacio máximo aprovechable para el correcto funcionamiento del vehículo. El suelo deberá estar nivelado y liso, no deberá haber obstáculos que obstruyan el movimiento ni

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

objetos que dificulten la visibilidad de las cámaras, tanto para el posicionamiento del vehículo como para la maniobrabilidad.

- **Visibilidad:** La visibilidad debe ser adecuada para permitir una detección precisa de los gestos realizados. Un entorno excesivamente oscuro puede dificultar la capacidad de la cámara para detectar los gestos de manera efectiva.

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA
UNIVERSIDAD POLITECNICA DE VALENCIA

ETSID

**GRADO EN INGENIERIA ELECTRONICA INDUSTRIAL
Y AUTOMATICA**

CONTROL DE TRAYECTORIA Y VELOCIDAD DE UN
VEHÍCULO SIMULADO MEDIANTE RECONOCIMIENTO
DE LA POSICIÓN DE LOS DEDOS DE LA MANO POR
VISIÓN ARTIFICIAL

DOCUMENTO:

PRESUPUESTO

ALUMNO: BEATRIZ VALLS MARTOS

TUTOR: VICENTE FERMÍN CASANOVA CALVO

Documento 3. Presupuesto

Índice

INTRODUCCIÓN	4
ESTADO DE LAS MEDICIONES	5
COSTE DE HERRAMIENTAS UTILIZADAS Y PERSONAL	6
COSTE TOTAL.....	8

Índice tablas

<i>Tabla 1 - Especificaciones servidor local</i>	6
<i>Tabla 1 - Licencias de software</i>	5
<i>Tabla 2 - Personal</i>	5
<i>Tabla 3 – Costes de software</i>	6
<i>Tabla 4 – Costes de personal</i>	7
<i>Tabla 5 – Costes totales</i>	8

1. INTRODUCCIÓN

En este informe se lleva a cabo un análisis del presupuesto necesario para el proyecto actual. Se examinarán los gastos relacionados con las licencias de software y el equipo empleado en el desarrollo del proyecto.

En cuanto a las horas dedicadas a este proyecto, se considerará el tiempo empleado tanto en la simulación en sí misma (que incluye la programación de scripts, su funcionamiento, pruebas y corrección de errores, entre otros) como el tiempo dedicado a la documentación.

2. ESTADO DE LAS MEDICIONES

A continuación, se presentan los recursos utilizados durante la elaboración del proyecto:

Licencias de software: para este apartado se tienen en cuenta aquellos software cuyas licencias son típicamente de pago.

Tabla 1 - Licencias de software

Software	Unidades
MATLAB R2021b	1
Office	

- **Personal:**

Tabla 2 - Personal

Descripción	Unidades
Trabajador en calidad de Ingeniero	1

3. COSTE DE HERRAMIENTAS UTILIZADAS Y PERSONAL

- **Software:** Para calcular los costos de las licencias de software utilizadas, se utiliza el costo anual de estas licencias como punto de referencia. Se estima la parte proporcional de su utilización en un entorno laboral, considerando un total de 1792 horas anuales (según lo establecido en el convenio colectivo del sector de empresas de ingeniería y oficinas de estudios técnicos). De esta manera, se obtiene el costo por hora de uso del software en cuestión y luego se multiplica por las horas implementadas en cada fase del proyecto.

$$\text{Coste por hora } \left(\frac{\text{€}}{\text{h}} \right) = \frac{\text{coste licencia anual (€)}}{\text{horas totales (1792h)}}$$

Tabla 3 – Costes de software

Software	Licencia (€)	Coste horas (€/h)	Horas (h)	Coste total (€)
MATLAB	800	0.45	120	53.57
Office	70	0.04	20	0.78
TOTAL	870	0.49	140	54.35

El coste de software es de:

CINCUENTA Y CUATRO EUROS Y TREINTA Y CINCO CENTIMOS

- **Personal:** Para calcular los costos de mano de obra, se considera el trabajo de un único trabajador desempeñando el rol de becario. Para esta estimación, tomamos como referencia la bolsa económica mínima de ayuda al empleo proporcionada por la Universidad Politécnica de Valencia, que es de 4.3 € por hora. Este valor se utilizará como base para calcular los costos laborales asociados al desarrollo del proyecto. A las horas totales anteriores hay que sumarles las horas que se ha estado trabajando con los scripts de Python. Las horas totales en la programación de Python son 110 horas.

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

Tabla 4 – Costes de personal

Personal	Salario (€/h)	Tiempo empleado (h)	Coste total (€)
Ingeniero	12	250	3000

El coste de personal es de:

TRES MIL EUROS

4. COSTE TOTAL

Finalmente se resumen los costes en la siguiente tabla:

Tabla 5 – Costes totales

Concepto	Coste (€)
Coste de software	54.57
Coste de personal	3000
IVA 21%	641.45
TOTAL	3.696

El presupuesto total del proyecto es de:

TRES MIL SEISCIENTOS NOVENTA Y SEIS

ANEXOS

ANEXOS:

Índice

ANEXO I: PROGRAMACIÓN 3

ANEXO I: PROGRAMACIÓN

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

Programación:

A continuación, se presenta el código de python utilizado para las etapas de detección de gestos y envío de la información vía UDP.

```
import cv2,time
import mediapipe as mp
import math, socket
import numpy as np

mp_drawing = mp.solutions.drawing_utils
mp_hands = mp.solutions.hands

IP = '127.0.0.1'
# IP = "localhost"
PORT = 2020
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Inicialización del modelo de detección de manos
hands = mp_hands.Hands(static_image_mode=False, max_num_hands=2,
min_detection_confidence=0.5, min_tracking_confidence=0.5)

# Inicialización de la cámara
cap = cv2.VideoCapture(0)
array = [1,1]
while cap.isOpened():
    success, image = cap.read()
    if not success:
        break

    # Convertir imagen de BGR a RGB
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # Detección de manos
    results = hands.process(image)

    # Dibuja los puntos de referencia y las conexiones en la
    imagen
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            mp_drawing.draw_landmarks(image, hand_landmarks,
mp_hands.HAND_CONNECTIONS)
```

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

```
# Analiza los puntos de referencia de la mano izquierda
left_hand_landmarks = results.multi_hand_landmarks[0] if
results.multi_hand_landmarks and
len(results.multi_hand_landmarks) > 0 else None
if left_hand_landmarks:
    # Indica qué dedos están levantados en la mano
    izquierda
    finger_tips =
[left_hand_landmarks.landmark[mp_hands.HandLandmark.INDEX_FINGER_
TIP],
                                left_hand_landmarks.landmark[mp_hands.
HandLandmark.MIDDLE_FINGER_TIP],
                                left_hand_landmarks.landmark[mp_hands.
HandLandmark.RING_FINGER_TIP],
                                left_hand_landmarks.landmark[mp_hands.
HandLandmark.PINKY_TIP]]

    # Comprueba si los dedos están levantados o no
    fingers_raised = [1 if finger_tip.y <
left_hand_landmarks.landmark[mp_hands.HandLandmark.INDEX_FINGER_M
CP].y else 0 for finger_tip in finger_tips]

    right_hand_landmarks = results.multi_hand_landmarks[1] if
len(results.multi_hand_landmarks) > 1 else None
    if right_hand_landmarks:
        # Calcula la distancia entre el pulgar y el índice en
        la mano derecha
        thumb_tip =
right_hand_landmarks.landmark[mp_hands.HandLandmark.THUMB_TIP]
        index_tip =
right_hand_landmarks.landmark[mp_hands.HandLandmark.INDEX_FINGER_
TIP]
        distance = math.sqrt((thumb_tip.x - index_tip.x) ** 2
+ (thumb_tip.y - index_tip.y) ** 2)

        x = [0, 0.1, 0.36, 1]
        y = [0, distance, 1, 1]
        valor_convertido = np.interp(distance, x, y)

        if fingers_raised[0] == 1 and fingers_raised[1] == 0
and fingers_raised[2] ==0 and fingers_raised[3] == 0:
            array[0] = round(valor_convertido, 2) * 1.25
            array[1] = round(valor_convertido, 2) * 0.75

        elif fingers_raised[0] == 1 and fingers_raised[1] ==
1 and fingers_raised[2] ==0 and fingers_raised[3] == 0:
            array[0] = round(valor_convertido, 2) * 1.50
```

TRABAJO FINAL DE GRADO
INGENIERIA ELECTRONICA INDUSTRIAL Y AUTOMATICA

```
        array[1] = round(valor_convertido, 2) * 0.50

        elif fingers_raised[0] == 0 and fingers_raised[1] ==
0 and fingers_raised[2] ==0 and fingers_raised[3] == 1:
            array[0] = round(valor_convertido, 2) * 0.75
            array[1] = round(valor_convertido, 2) * 1.25

        elif fingers_raised[0] == 1 and fingers_raised[1] ==
0 and fingers_raised[2] ==0 and fingers_raised[3] == 1:
            array[0] = round(valor_convertido, 2) * 0.5
            array[1] = round(valor_convertido, 2) * 1.5

    else:
        array[0] = round(valor_convertido, 2) * 1
        array[1] = round(valor_convertido, 2) *
1

        data = np.array(array)
        data_round = np.round(data,2)
        print(data_round)
        data = data_round.tobytes()
        # data = data = bytes(str(array), 'utf-8')
        time.sleep(0.1)
        sock.sendto(data_round, (IP, PORT))
        # s.send(str(array).encode())

    # Muestra la imagen
    cv2.imshow('MediaPipe Hands', image)

    if cv2.waitKey(5) & 0xFF == 27:
        break

hands.close()
cap.release()
```