



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

Sistema de control de posicionamiento para realizar la carga inalámbrica y automática de un vehículo terrestre

Autor: D. Carlos Cutillas Silvestre

Director: Dr. Juan Luis Zamora Macho

Codirector: Dr. Aurelio García Cerrada

Madrid
Julio 2022

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
**Sistema de control de posicionamiento para realizar la carga inalámbrica y
automática de un vehículo terrestre**

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico **2021/2022** es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.



Fdo.: Carlos Cutillas Silvestre

Fecha: 26/ 08/ 2022

Autorizada la entrega del proyecto

DIRECTOR Y CODIRECTOR DEL PROYECTO



Fdo.: J. L. Zamora Macho



Fecha: .26../ .08../ 2022

y Aurelio García Cerrada



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

Sistema de control de posicionamiento para realizar la carga inalámbrica y automática de un vehículo terrestre

Autor: D. Carlos Cutillas Silvestre

Director: Dr. Juan Luis Zamora Macho

Codirector: Dr. Aurelio García Cerrada

Madrid
Julio 2022

A mis padres, mi hermano y mi yaya

SISTEMA DE CONTROL DE POSICIONAMIENTO PARA REALIZAR LA CARGA INALÁMBRICA Y AUTOMÁTICA DE UN VEHÍCULO TERRESTRE

Autor: Cutillas Silvestre, Carlos.

Director: Zamora Macho, Juan Luis.

Codirector: García Cerrada, Aurelio.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Este trabajo consiste en el diseño de un sistema de control de posicionamiento para un vehículo terrestre que permite recargar la batería de este de manera inalámbrica mediante una estación de carga y un receptor acoplado al vehículo. El conjunto de cámaras proporciona al sistema de control, en tiempo real, los parámetros de posición y orientación del vehículo, mientras que el detector de corriente monitoriza el nivel de batería.

Palabras clave: Raspberry Pi, OptiTrack, carga inalámbrica, sistema de control de posicionamiento, detección de damero

1. Introducción

El proyecto tiene como objetivo principal el diseño y ejecución de un sistema de detección y control autónomo del posicionamiento de vehículos o robots terrestres para poder realizar un acercamiento a la base de carga inalámbrica cuando el sensor de corriente detecte que la batería se está agotando. El vehículo utilizado en el trabajo es el mismo empleado en las prácticas de laboratorio de asignaturas como Regulación Automática o Control Digital en ICAI.

2. Descripción del proyecto

El vehículo consta de dos ruedas traseras accionadas eléctricamente por motores DC. Mediante un sistema de control digital, implantado por un Raspberry Pi 3B+ [1], que se denomina en este trabajo como Raspberry Pi principal, se gobierna tanto la tensión común,

para hacerlo avanzar, como diferencial, girar sobre sí mismo con un par, aplicada a los motores. La velocidad angular de los motores de corriente continua se mide con un *encoder* magnético. Es a través de las medidas de velocidad angular que el control estima la velocidad de marcha lineal y ángulo de giro del vehículo.

El sistema de cámaras externo para la localización de posición del vehículo está implementado mediante las cámaras de captura de movimiento OptiTrack [2]. Este sistema detecta mediante ondas infrarrojas los marcadores esféricos del vehículo formando un sólido rígido a través del software Motive, desarrollado específicamente para este *hardware*, que establece la posición y rotación del vehículo en un entorno tridimensional limitado. Por otra parte, el sistema de cámara interno se refiere al módulo de cámara Raspberry Pi montado en la parte superior del vehículo en la Raspberry Pi secundaria y proporciona información sobre el posicionamiento y orientación de este respecto a su distancia a un tablero mediante la librería de visión artificial OpenCV (*Open Source Computer Vision*) [3].

Respecto a la base de carga inalámbrica se plantean dos opciones. Una elección posible es un kit de transferencia inalámbrica TDK, que tiene un precio inferior a su contraparte, pero una implementación sencilla; o en el kit de carga RENESAS, con especificaciones más refinadas y potentes, aunque con un coste mayor.

3. Solución

Se ha escrito un programa, en el lenguaje de programación Python utilizando las funcionalidades de la librería OpenCV, que puede determinar la distancia y ángulo de giro del vehículo relativos al damero, mientras que envía esta información a la Raspberry Pi principal para que el sistema de control actúe según convenga en cada situación. Haciendo uso de la información recibida de la cámara del vehículo se ha implementado un sistema de control que corrige el ángulo de giro del vehículo para quedar orientado de forma paralela a la estación de carga inalámbrica, obteniendo la mayor corriente de carga posible.

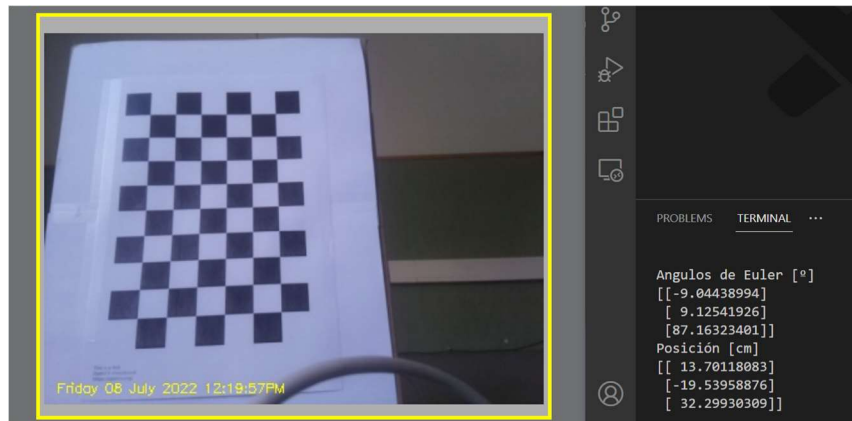


Figura 1 Detección de posición y orientación con OpenCV

Además, se ha implementado el sensor de corriente para la monitorización de la batería y se ha conseguido cargar la batería del vehículo con la estación de carga, no obstante, se debe optimizar la medida de corriente mediante Simulink para dar órdenes al sistema de control.

4. Resultados

Se caracteriza la precisión del sistema de cámaras interno sobre los ejes X y eje Y del plano de trayectorias respecto a las medidas recopiladas por el sistema de cámaras externo OptiTrack para analizar si los resultados registrados por OpenCV son válidos.

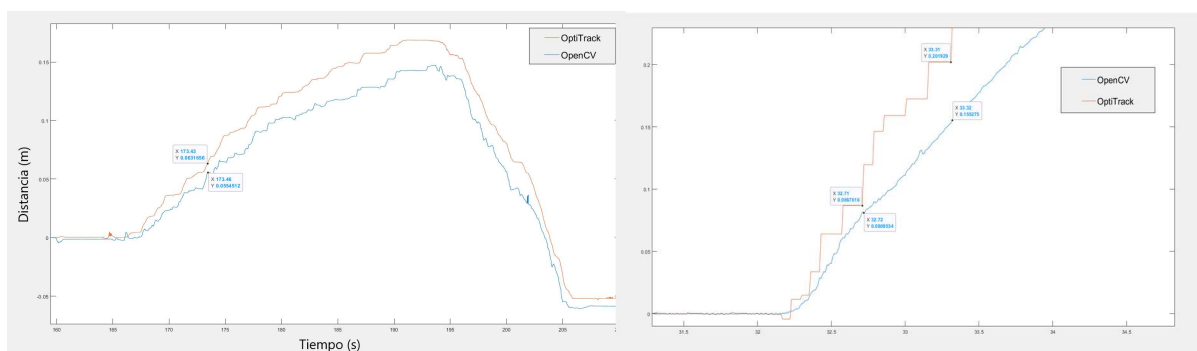


Figura 2 Ensayo de precisión de distancia en eje X y eje Y

Por otro lado, se ensaya un control proporcional de corrección de ángulo para el vehículo para garantizar que no se desvíe de una trayectoria recta y que se encuentre dentro del rango de funcionamiento de la estación de carga.

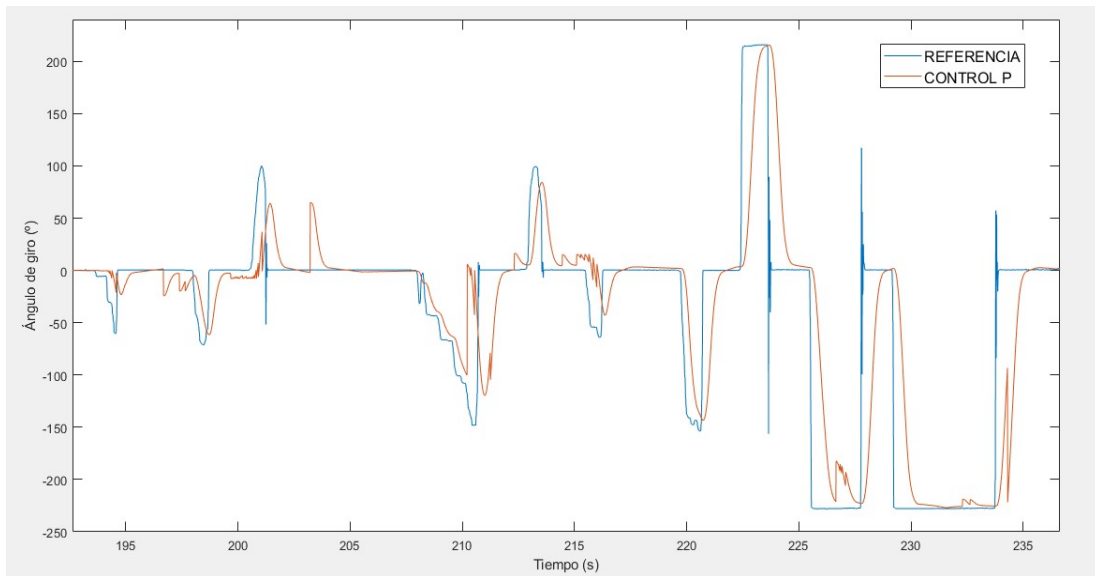


Figura 3 Ensayo de corrección de ángulo

5. Conclusiones

Al comparar las medidas de OpenCV con el sistema de cámaras OptiTrack la medida recopilada de distancias menores a 1m es correcta, aunque es necesaria más precisión en distancias superiores ya que la cámara pierde la detección del damero, debido al desenfoque por movimiento o la calibración de sus parámetros intrínsecos. El control implementado para la corrección de ángulo del vehículo funciona correctamente para orientar la trayectoria de este de manera paralela a la estación de carga.

6. Referencias

- [1] Raspberry Pi. (14 de marzo de 2018). www.raspberrypi.com. Recuperado el 27 de junio de 2022, de www.raspberrypi.com: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>
- [2] OptiTrack. (19 de febrero de 2015). OptiTrack. Recuperado el 2022 de julio de 18, de OptiTrack: <https://optitrack.com/cameras/flex-13/>
- [3] OpenCV. (1 de agosto de 2013). OpenCV.org. Recuperado el 19 de abril de 2022, de docs.opencv.org: https://docs.opencv.org/3.4/d4/d94/tutorial_camera_calibration.html

POSITIONING CONTROL SYSTEM TO PERFORM WIRELESS AND AUTOMATIC CHARGING OF A GROUND VEHICLE

Author: Cutillas Silvestre, Carlos.

Supervisor: Zamora Macho, Juan Luis.

Cosupervisor: García Cerrada, Aurelio.

Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

This project consists of the design of a positioning control system for a ground vehicle to allow the battery recharge wirelessly by means of a charging station and a receiver in the vehicle. The camera array provides the vehicle's position and orientation parameters for a real time control system, while the current detector monitors the battery level.

Keywords: Raspberry Pi, OptiTrack, wireless charging, positioning control system, checkerboard detection

1. Introduction

The project aims to design and implement a system for autonomous detection and control of the positioning of ground vehicles or robots to approach the wireless charging base when the current sensor detects that the battery runs low. The position and orientation of the vehicle is monitored by an external tracking camera system and, independently, a camera located in the vehicle to establish the distance and relative angles with respect to a checkerboard. This vehicle is the used in the laboratory sessions of subjects such as Automatic Regulation or Digital Control at ICAI.

2. Project description

The vehicle consists of two rear wheels electrically driven by DC motors. By means of a digital control system, implemented through a Raspberry Pi 3B+ named in this work as the main Raspberry Pi, the common voltage makes it move forward, and a differential voltage makes it turn with an applied torque. The angular velocity of DC motors is measured with a magnetic encoder. It is through the angular velocity measurements that the control estimates the linear running speed and angle of rotation of the vehicle.

The external camera system for vehicle positioning is implemented by means of OptiTrack motion capture cameras. This system uses infrared waves to detect the vehicle's spherical markers considered as a rigid solid through the Motive software, developed specifically for this hardware, which establishes the vehicle's position and rotation angle in a limited three-dimensional domain. On the other hand, the internal camera system, referred as the Raspberry Pi camera module mounted on top of the vehicle on the secondary Raspberry Pi, provides information about the position and orientation of the vehicle with respect to its distance to a dashboard using the OpenCV machine vision library.

One possible choice for a charging base is the TDK wireless power transfer kit, which has a lower price than its counterpart, but a simple implementation, or the RENESAS charging kit, with more refined and powerful specifications, but with a higher cost.

3. Solution

A Python program with the OpenCV library functionalities, can distinguish the distance and turning angle of the vehicle relative to the checkerboard, while sending this information to the main RPi into the control system. Using this camera information, a steering angle control has been implemented so that it is oriented parallel to the wireless charging station.

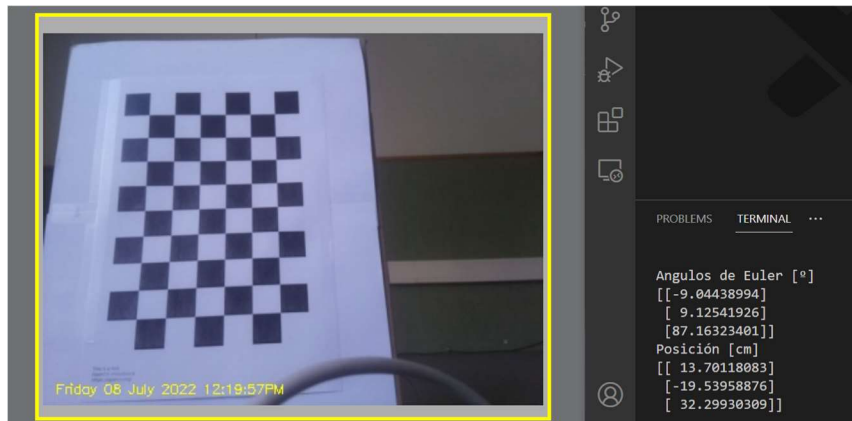


Figure 4 Location and orientation estimation through OpenCV

A current sensor for battery monitoring has been implemented and the vehicle battery has been successfully charged with the charging station, however, the current measurement needs to be optimized using Simulink to send commands to the control system.

4. Results

The accuracy of the internal camera system on the X-axis and Y-axis of the trajectory plane is characterized with respect to the measurements collected by the external OptiTrack camera system to analyze whether the results recorded by OpenCV are valid.

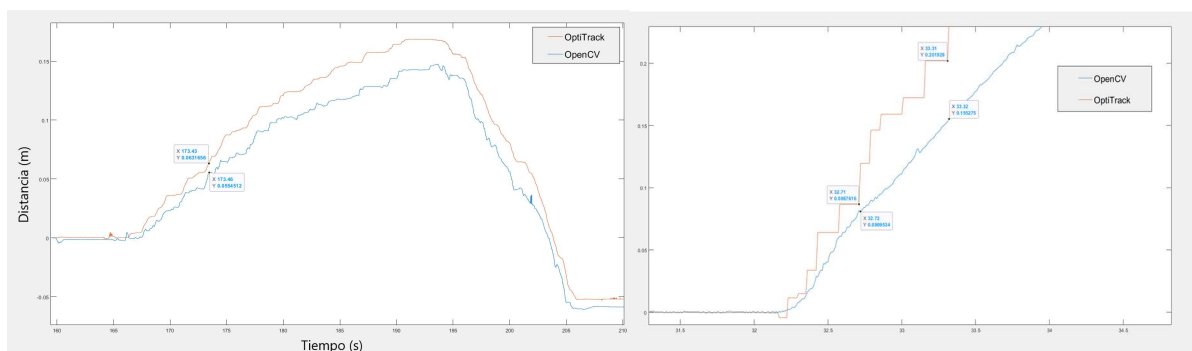


Figure 5 X-axis and Y-axis distance accuracy test

On the other hand, a proportional angle correction control for the vehicle is tested to ensure that it does not deviate from a straight path and is within the operating range of the charging station.

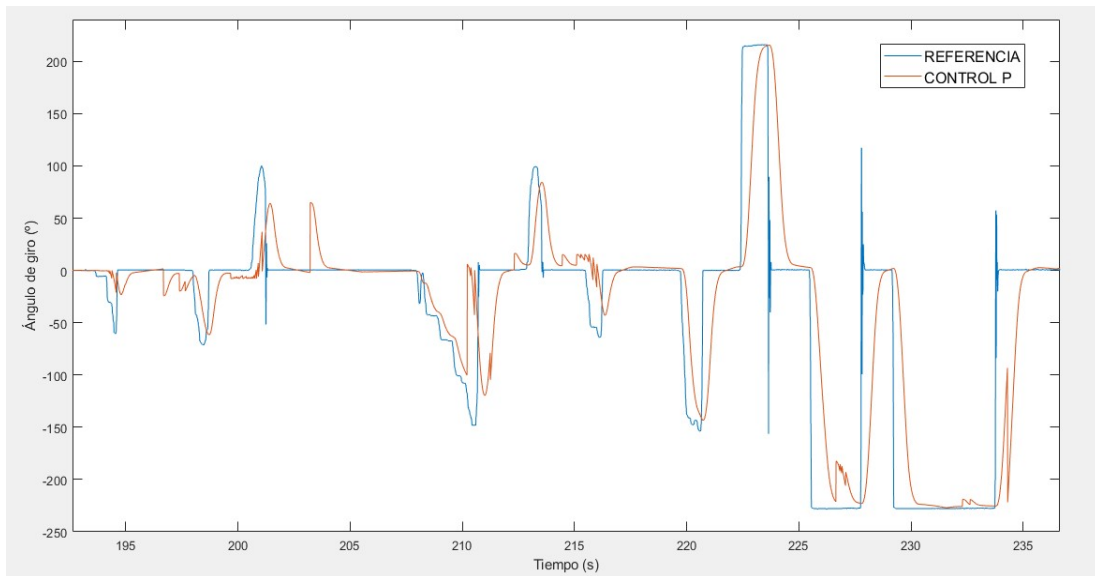


Figure 6 Angle correction test

5. Conclusion

After comparing the OpenCV measurements with the OptiTrack camera system, the measurement collected for distances less than 1m is correct, although more precision is needed at greater distances since the camera loses checkerboard detection due to motion blur or calibration of its intrinsic parameters. The control implemented for the vehicle angle correction works correctly to orient the vehicle trajectory parallel to the charging station.

6. References

- [1] Raspberry Pi. (2016, April 25). www.raspberrypi.com. Retrieved June 27, 2022, from www.raspberrypi.com: <https://www.raspberrypi.com/products/camera-module-v2/>
- [2] OptiTrack. (2015, February 19). OptiTrack. Retrieved July 2022, 18, from OptiTrack: <https://optitrack.com/cameras/flex-13/>
- [3] OpenCV. (2013, August 1). OpenCV.org. Retrieved April 19, 2022, from docs.opencv.org: https://docs.opencv.org/3.4/d4/d94/tutorial_camera_calibration.html

Índice de la memoria

Capítulo 1. Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Recursos	4
1.4 Metodología.....	9
Capítulo 2. Estado del Arte.....	12
2.1 Navegación autónoma	12
2.2 Carga inalámbrica.....	15
2.3 Visión artificial.....	17
Capítulo 3. Hardware	20
3.1 Raspberry Pi 3B+	20
3.2 Módulo de cámara Raspberry Pi v2.1	21
3.3 Vehículo terrestre	23
3.4 OptiTrack	24
3.5 Impresión 3D de soportes.....	25
Capítulo 4. Software y Comunicación	27
4.1 Raspberry Pi secundaria	27
4.1.1 Sistema operativo	27
4.1.2 Protocolo SSH.....	28
4.1.3 OpenCV-Python en Visual Studio Code	29
4.1.4 Calibración de cámara RPi.....	30
4.1.5 Sistema de detección y posicionamiento	37
4.1.6 Interfaz de vídeo	39
4.2 Comunicación entre las dos Raspberry Pi	41
4.3 Motive 47	
Capítulo 5. Carga Inalámbrica	49
5.1 Kit TDK.....	49
5.2 Kit Renesas WP15WBD-RK.....	51
5.3 Elección de kit de carga.....	52
5.4 Medidor de corriente	56

Capítulo 6. Sistema de control y Navegación	59
6.1 Descripción del sistema de control.....	59
6.1.1 Controles de velocidad de avance y guiñada	60
6.1.2 Control de ángulo de giro	60
6.2 Control de vehículo mediante Simulink.....	61
Capítulo 7. Ensayos y resultados	66
7.1 Entorno de trabajo	66
7.2 Ensayo de acoplamiento a base de carga.....	68
7.3 Ensayo de corrección de ángulo de giro.....	71
Capítulo 8. Conclusiones y futuros proyectos	73
8.1 Conclusiones	73
8.2 Futuros proyectos	74
Capítulo 9. Bibliografía.....	76
Anexo A: Instalación de Ubuntu en Raspberry Pi.....	80
Anexo B: WTM905525-12F2-12V-G y WRM483265-10F5-12V-G Datasheet	82
Anexo C: Contribución a los Objetivos para el Desarrollo Sostenible de la ONU.....	86

Índice de figuras

Figura 1.2.1 Detección de movimiento mediante OpenCV (Jalled & Voronkov, 2016)	2
Figura 1.3.1 Módulos TDK transmisor (izquierda) y receptor (derecha) (TDK Corporation, 2019).....	6
Figura 1.3.2 Kit de recarga inalámbrica WP15WBD-RK (Renesas Electronics Corporation, 2020).....	6
Figura 1.3.3 Cámara OptiTrack modelo Flex 13 (OptiTrack, 2015).....	7
Figura 2.2.1 Diagrama de carga inalámbrica capacitiva (Panchal, Stegen , & Lu, 2018)...	16
Figura 3.1.1 Raspberry Pi 3 Modelo B+ Revisión 1.3 (Raspberry Pi, 2018).....	20
Figura 3.1.2 Placa de alimentación X728 conectada a RPi 4 (Geekworm, 2020).....	21
Figura 3.2.1 Módulo de cámara RPi v2.1 y cable plano (Raspberry Pi, 2016).....	22
Figura 3.2.2 Cámara conectada a la Raspberry Pi 3 (Raspberry Pi, 2016)	22
Figura 3.3.1 Vehículo terrestre con tracción diferencial (Fuente: Elaboración propia)	23
Figura 3.4.1 Diagrama de conexión sistema OptiTrack (Fuente: Departamento de Electrónica).....	25
Figura 3.5.1 Soporte para bobina receptora.....	26
Figura 3.5.2 Soporte para bobina receptora.....	26
Figura 3.5.3 Soporte para RPi y cámara.....	26
Figura 4.1.1 Diagrama de funcionamiento del protocolo SSH (Fuente: elaboración propia)	28
Figura 4.1.2 Damero 9x6 (OpenCV, 2013).....	31
Figura 4.1.3 Modelo de cámara sin lente con distorsión radial (Li & Zhang, 2011)	32
Figura 4.1.4 Distorsión radial (izquierda) y corrección (derecha) (Kukelova & Pajdla, 2007)	32
Figura 4.1.5 Parámetros de la matriz de distorsión	33
Figura 4.1.6 Conversión con matriz de cámara	33
Figura 4.1.7 Imágenes muestra de damero para calibrado (Fuente: Elaboración propia)...	34
Figura 4.1.8 Imágenes de muestra con esquinas detectadas (Fuente: Elaboración propia). 35	
Figura 4.1.9 Matriz de cámara y parámetros de distorsión del módulo de cámara RPi (Fuente: Elaboración propia)	36

Figura 4.1.10 Comprobación de sistema de coordenadas de damero (Fuente: elaboración propia).....	37
Figura 4.1.11 Secuencia de Ángulos de Euler obtenida por OpenCV (Diebel, 2006)	39
Figura 4.1.12 Interfaz <i>web</i> de transmisión de video (Fuente: Elaboración propia).....	41
Figura 4.2.1 Funcionamiento de paquetes con JSON (Peng, 2011)	42
Figura 4.2.2 Arquitectura del modelo de protocolo TCP/IP (Parziale, y otros, 2006).....	43
Figura 4.2.3 Estructura de comunicación del proyecto (Fuente: Elaboración propia)	46
Figura 4.2.4 Diagrama de bloques de comunicación de RPi principal (Fuente: Elaboración propia).....	47
Figura 4.3.1 Sólido rígido generado por Motive (Fuente: Elaboración propia)	48
Figura 5.1.1 Microcircuitos P9221-R (Fuente: Renesas)	50
Figura 5.1.2 Microcircuito P9242-R (Fuente: Renesas).....	50
Figura 5.2.1 Diagrama de bloques del kit <i>Renesas</i> (Fuente: <i>Renesas</i>).....	51
Figura 5.3.1 Ensayo de carga kit TDK (Fuente: Elaboración propia).....	53
Figura 5.4.1 Medidor de corriente <i>Current Click</i> (MikroElektronika, 2017).....	57
Figura 5.4.2 Diagrama de bloques del sensor de corriente en Simulink (Fuente: Elaboración propia).....	58
Figura 6.1.1 Diagrama del lazo de control del vehículo (Fuente: Departamento de Electrónica).....	59
Figura 6.2.1 Diagrama de CAR_CONTROL_SYSTEM.slx (Fuente: Elaboración propia).....	62
Figura 6.2.2 Subsistema Control (Fuente: Elaboración propia)	63
Figura 6.2.3 Subsistema Simulation (Fuente: Elaboración propia).....	64
Figura 6.2.4 Subsistema Hardware (Fuente: Elaboración propia)	65
Figura 6.2.5 Subsistema Monitoring (Fuente: Elaboración propia)	65
Figura 7.1.1 Entorno de trabajo con sistema de referencia (Fuente: Elaboración propia) ..	66
Figura 7.1.2 Disposición de botones y LED de marcha del vehículo (Fuente: Elaboración propia).....	67
Figura 7.2.1 Ensayo de acoplamiento en el eje X	68
Figura 7.2.2 Ensayo de acoplamiento cercano en el eje Y	70
Figura 7.2.3 Ensayo de acoplamiento lejano en el eje Y.....	70
Figura 7.3.1 Respuesta del control P para corrección de ángulo de giro	71

Figura 7.3.2 Detalle en respuesta del control P para corrección de ángulo de giro	72
Figura A.1 Selección de OS.....	78
Figura A.2 Configuración de instalación.....	79

Índice de tablas

Tabla 5.3.1 Ensayo de distancia frontal (V_o : voltaje de la batería inicial, V_{carga} : voltaje de la batería cargando, ΔV : $V_{carga} - V_o$, I_{rms} : corriente eficaz en el secundario)	54
Tabla 5.3.2 Ensayo de ángulo relativo (V_o : voltaje de la batería inicial, V_{carga} : voltaje de la batería cargando, ΔV : $V_{carga} - V_o$, I_{rms} : corriente eficaz en el secundario)	55
Tabla 5.3.3 Ensayo de distancia lateral (V_o : voltaje de la batería inicial, V_{carga} : voltaje de la batería cargando, ΔV : $V_{carga} - V_o$, I_{rms} : corriente eficaz en el secundario)	56
Tabla 6.1.1 Parámetros de diseño control PID de velocidades (Fuente: Elaboración propia)	60
Tabla 6.1.2 Parámetros de diseño control PID de ángulo de giro (Fuente: Elaboración propia)	61

Índice de ecuaciones

Ecuación 4.1.4.1 Distorsión radial	33
Ecuación 4.1.4.2 Distorsión tangencial	33

Índice de acrónimos

ADC	Convertidor Analógico Digital
DC	Corriente continua
GPS	<i>Global Positioning System</i>
GNSS	<i>Global Navigation Satellite Systems</i>
PWM	<i>Pulse-Width Modulation</i> (Modulación por ancho de pulso)
IC	Circuito Integrado
IMU	Unidad de Medida Inercial
IP	Protocolo de internet
OpenCV	<i>Open Source Computer Vision</i>
OS	Sistema operativo
PID	Proporcional Integral Derivativo
PWM	<i>Pulse-Width Modulation</i>
RPi	Raspberry Pi
SSH	<i>Secure Shell</i>

Capítulo 1. INTRODUCCIÓN

1.1 MOTIVACIÓN

La motivación principal de este Trabajo de Fin de Grado es apoyar la innovación para un futuro más cómodo para la sociedad mediante la implementación de tecnología cada vez más autónoma tanto en la vida cotidiana como al entorno laboral. Las industrias de la robótica, automática e inteligencia artificial se presentan como la solución para diversos sectores para realizar tareas cotidianas y reducir costes directos debido a su propia naturaleza precisa y de repetitividad con una consistencia impecable.

Por esta razón, también se alienta al desarrollo e inversión en el ámbito de la energía inalámbrica, que está estrechamente ligado al concepto de navegación autónoma, por lo que son necesarias novedades que reflejen un avance en la tecnología del sector de carga sin cables. Sin embargo, el mayor inconveniente para su uso es la menor eficiencia energética comparada con la recarga tradicional que debe mejorarse para poder impulsar su uso al público general.

Junto al sistema de control de posición de un vehículo, se busca una autonomía completa en el proceso de funcionamiento de la máquina, es decir, que sin intervención humana pueda desempeñar sus tareas mientras no haya agentes externos impidiéndolo y su base de carga esté operativa.

En conclusión, este proyecto engloba diversos contenidos y desafíos que serán, sin ninguna duda, beneficiosos para todo el mundo en un futuro y, por tanto, se debe apoyar cualquier tipo de trabajo relacionado con la tecnología para el bienestar humano.

1.2 OBJETIVOS

El trabajo se centra en tres áreas: movimiento autónomo, carga inalámbrica y localización en un dominio cerrado. El objetivo principal del proyecto es combinar estos

aspectos para diseñar un sistema navegación para un vehículo terrestre que pueda ejercer una tarea de forma autónoma mediante cámaras que localicen su posición en el entorno y sea capaz de volver a su estación de carga inalámbrica y, una vez recargada la batería se reincorpore a su puesto. Por tanto, en el trabajo se va a diseñar un sistema de control de posición y ángulo para realizar correctamente la recarga de la batería de un vehículo autónomo cuando este detecte que el nivel de carga está cercano a agotarse. Para la consecución del proyecto planteado se han propuesto una serie de objetivos con el fin de abarcar por partes complementarias la totalidad del trabajo:

1) Detección de posición y orientación mediante OpenCV:

Para poder determinar las posiciones y orientaciones relativas del vehículo autónomo en tiempo real mediante un módulo de cámara Raspberry Pi se ha optado por utilizar la técnica de visualización artificial conocida como en inglés *pose estimation*, o estimación de posición y orientación. Este método se emplea para analizar la postura de cierto objeto o cuerpo con una imagen. A través de la librería OpenCV Python, se utiliza la detección de un tablero de ajedrez o damero para estimar la posición y ángulo del vehículo mediante procedimiento descrito.

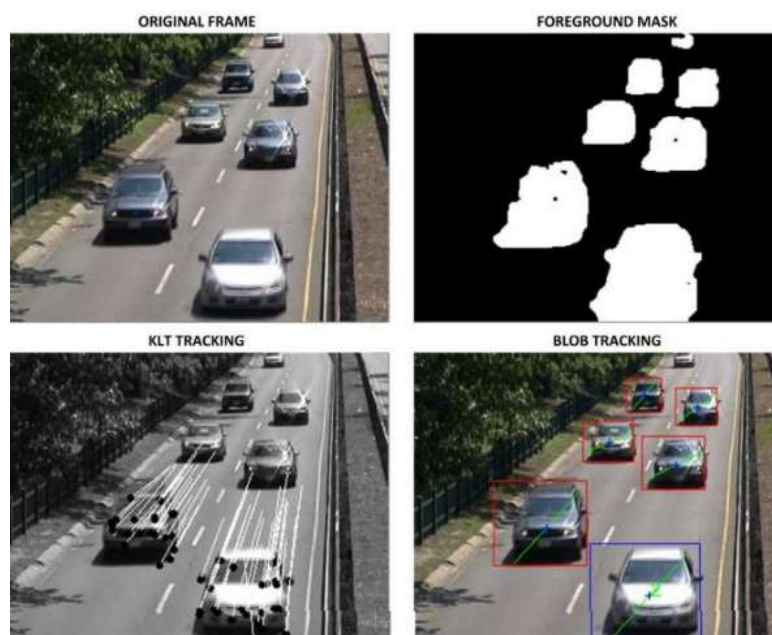


Figura 1.2.1 Detección de movimiento mediante OpenCV (Jalled & Voronkov, 2016)

A la hora de analizar la exactitud del método seleccionado, se comprueba mediante la comparación de las medidas recopiladas por OpenCV y OptiTrack la fidelidad de dichas medidas, tomando siempre como referencia los valores proporcionados por el sistema de cámaras externo OptiTrack.

2) Implementación del control y sistema de navegación autónoma:

La función principal del control empleado es corregir el ángulo del vehículo respecto al tablero que visualiza la cámara ya que la estación de carga se encuentra paralelamente al tablero y cuanto mayor sea el ángulo relativo entre bobina receptora y transmisora de carga, es más probable no encontrar dentro del primario el secundario. Asimismo, deberá corregir cualquier desvío que ocurra en la trayectoria hasta su destino. Es necesario estudiar los controles de regulación más apropiados para esta aplicación y seleccionar el que proporcione unos resultados óptimos en cuanto a tiempo de reacción y precisión se refiere.

El sistema de navegación autónoma diseñado con Matlab Simulink, de la misma manera que los controles, se encarga de gobernar el avance del vehículo según la información sobre posición y orientación recibida de la Raspberry Pi secundaria. Se plantea el caso donde el vehículo necesita volver a recargarse y el caso contrario, abandonar la estación de carga cuando se reciba la instrucción de batería a nivel aceptable o completo.

3) Conexión de dispositivos mediante sistema de comunicaciones:

Este proyecto contiene diversos dispositivos y es necesario utilizar diferentes tipos de comunicación entre ellos según la conveniencia de la aplicación. Se han establecido las vías de comunicación óptimas en cada caso concreto, estudiando las diferentes posibilidades y seleccionando la que proporciona al sistema de comunicación general la mayor velocidad de transmisión y versatilidad en el uso.

Una placa Raspberry Pi, a cargo de recopilar y enviar las medidas a la Raspberry Pi del vehículo, estará situada en la parte superior del mismo vehículo, y por lo tanto puede establecerse una línea de comunicación a través del puerto Ethernet para el envío y recibimiento de los paquetes de información.

Entre cada ordenador, que utiliza un *software* específico para cada pieza del proyecto, y las placas Raspberry Pi se ha escogido el protocolo de comunicación SSH, para poder establecer el acceso remoto y administración del dispositivo de forma segura debido al tipo de cifrado utilizado para la comunicación. A su vez, la comunicación entre las cámaras OptiTrack y el *software* Motive se realiza a través de la tecnología inalámbrica de corto alcance Bluetooth, aunque se propone utilizar la tecnología 5G para realizar el envío de datos, debido a su menor latencia.

4) Medida de corriente de batería:

Debido a que la Raspberry Pi principal no es capaz de distinguir si la batería del vehículo está conectada a la alimentación y cargándose, tampoco puede enviar órdenes al sistema de navegación para que actúe en consecuencia. Es por esto por lo que se implanta al vehículo un sensor de corriente idóneo para la detección de flujo de carga en un circuito. Este tipo de sensores convierte el voltaje actual en una resistencia de carga en la corriente que circula por este. Mediante el diseño de un programa de Simulink se puede conocer el valor que devuelve el sensor y emplearlo en el programa para enviar instrucciones de actuación para el sistema de navegación.

1.3 RECURSOS

Este trabajo de fin de grado requiere una diversa cantidad de medios tanto físicos (*hardware* y herramientas) como digitales (*software*) debido a los diversos ámbitos que se abarcan: electrónica, automática, programación y robótica, entre otros. Los componentes utilizados para la consecución del proyecto son los siguientes:

1) Hardware

En este apartado se describe brevemente el vehículo utilizado, sus especificaciones y complementos; el sistema de cámaras externo OptiTrack, la Raspberry Pi, el módulo de cámara y otros dispositivos empleados.

- Vehículo con tracción diferencial: Vehículo con tracción diferencial: consta de tres modos de navegación diferentes: libre, seguimiento de trayectorias (con OptiTrack) y seguimiento de pared mediante infrarrojos. Para este proyecto se va a trabajar con las dos primeras para conseguir el resultado deseado. Los componentes principales de este vehículo son:
 - 2 motores DC (EMG30) para tracción diferencial.
 - 2 *enconder* magnéticos, uno para cada motor, para medir el ángulo girado del vehículo.
 - 1 Raspberry PI 3B+ (Raspberry Pi principal). Es el miniordenador que adquiere todas las medidas sensoriales, monitoriza los estados del vehículo, da órdenes PWM al controlador MD25 de los motores DC y señala el estado del vehículo con un led RGB. Recibe datos sobre la posición y orientación del vehículo por *Bluetooth* desde un ordenador conectado al sistema de cámaras OptiTrack, recibe los canales de una emisora RC y organiza la comunicación con un programa de monitorización en el ordenador.
 - Batería: proporciona 12V no regulados a los motores y tensión de 5V regulada para alimentar la electrónica del vehículo.
- Base de carga inalámbrica: Se estudia cada sistema de carga inalámbrica, observando sus especificaciones y haciendo ensayos de carga de batería para recopilar valores como amperios entregados, voltaje de la batería, tiempo de recarga o energía pérdida por la naturaleza de la tecnología. Los dos sistemas de carga disponibles que se van a analizar son los siguientes:
 - TDK *Wireless Power Transfer*: este kit de carga diseñado por la empresa TDK consta de una bobina transmisora (WTM905525-12F2-12V-G) y una bobina receptora (WRM483265-10F5-12V-G).

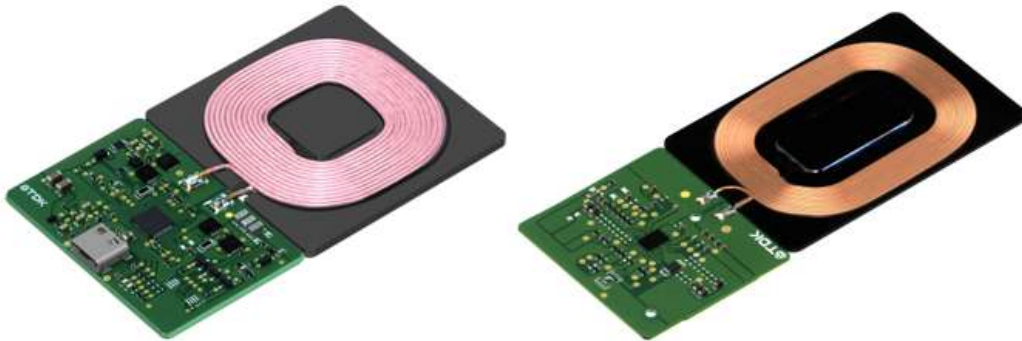


Figura 1.3.1 Módulos TDK transmisor (izquierda) y receptor (derecha) (TDK Corporation, 2019)

- RENESAS WP15WBD-RK *Bi-directional Data Transfer Evaluation Kit*: se divide en parte transmisora (P9242-R3) y la receptora (P9221-R3).



Figura 1.3.2 Kit de recarga inalámbrica WP15WBD-RK (Renesas Electronics Corporation, 2020)

- Cámaras OptiTrack: el modelo Flex 13 disponible en el laboratorio se trata de un dispositivo de captura de movimiento con precisión sobresaliente. Con esta tecnología se puede monitorizar el movimiento del vehículo para obtener en tiempo real parámetros clave como la posición y orientación en el espacio delimitado por el sistema de cámaras.

FLEX 13



Figura 1.3.3 Cámara OptiTrack modelo Flex 13 (OptiTrack, 2015)

- Raspberry Pi 3B+ (Raspberry Pi secundaria): modelo idéntico a la Raspberry del vehículo, pero con distinto propósito en el proyecto. Se encarga liberar espacio de trabajo a la otra placa, teniendo conectado un módulo de cámara Raspberry Pi v2 para realizar la estimación de posición y, posteriormente enviar dichos valores a la Raspberry Pi principal.
- Módulo de cámara Raspberry Pi v2.1: cámara de tamaño reducido, óptima para aplicaciones en movimiento. Está conectada a la Raspberry Pi secundaria mediante un cable de transmisión para realizar las interpretaciones de OpenCV según los fotogramas capturados.
- Ordenador: se utiliza para transferir la información de Matlab/Simulink y como conexión inicial con el servidor de la zona de operación. Además, actúa como cliente al conectarse al vehículo para recopilar datos sobre su funcionamiento y coordinarlo con las cámaras.
- Impresora 3D FDM: la fabricación aditiva ha sido necesaria para obtener la estructura que soporta la Raspberry Pi secundaria y el módulo de cámara en la parte superior del vehículo.

2) Software

En el proyecto se han utilizado diversas aplicaciones para ordenador para convertir, procesar y analizar todos los datos y parámetros recopilados por los distintos dispositivos que conforman la estructura del trabajo.

- **Matlab – Simulink:** los entornos de trabajo de Matlab y Simulink se complementan e integran de forma conjunta. Simulink es una herramienta de la aplicación Matlab, que basa su funcionamiento en la conexión por bloques para la generación de cualquier tipo de modelo físico, eléctrico, dinámico, etc. En este proyecto todos los archivos para el control, navegación y comunicaciones del vehículo con la Raspberry Pi principal están programados para este programa. Los ficheros de Matlab tienen diversas funciones y pueden ser modificados para realizar ensayos distintos en cada situación. Por ejemplo, se puede controlar el vehículo con una emisora de radiofrecuencia o activar el modo de navegación automática para realizar distintas comprobaciones.
- **Motive:** programa diseñado para interpretar la captura de movimiento realizada por las cámaras OptiTrack. Genera un sólido rígido basándose en marcadores dispuestos en el vehículo y de esta manera poder determinar la posición y orientación respecto a un sistema de coordenadas fijo cuando detecta movimiento por parte del vehículo.
- **Visual Studio Code:** se trata de un programa de código abierto de la empresa Microsoft diseñado como editor de código fuente con una interfaz cómoda y soporte para diversos lenguajes de programación, *framework* y librerías. Se ha utilizado para escribir y ejecutar todos los ficheros utilizados en la Raspberry Pi secundaria para captura y transmisión de vídeo, procesado de imágenes y estimación de distancia con detección de damero de la librería OpenCV-Python.

1.4 METODOLOGÍA

Este Trabajo de Fin de Grado parte de avances realizados en otros proyectos de la universidad, mientras que en otras áreas se comienza desde el principio sin ninguna base. Es por esta razón que se ha dividido el proyecto en varias etapas planificadas en función del conocimiento previo, recursos disponibles y dificultad de consecución para poder abordar los objetivos propuestos en el apartado 1.2 de la manera más eficiente y ordenada posible. Sin embargo, si se han encontrado ciertos contratiempos en la realización del proyecto se ha intentado continuar el cronograma con cambios mínimos para no modificar en gran medida la planificación inicial.

Las distintas fases que ha atravesado el proyecto han sido:

Instalación y documentación de sistema operativo en Raspberry Pi

En primer lugar, se debe inicializar desde cero la Raspberry Pi secundaria, ya que se trata de un complemento adicional que no se había utilizado anteriormente para el vehículo terrestre. Además, debido al diseño del sistema operativo, no se puede consultar una interfaz gráfica con un monitor directamente conectado a la placa, por lo que se debe realizar toda la configuración inicial desde un ordenador externo conectado mediante el protocolo de comunicación SSH. Para poder trabajar correctamente de esta manera se debe revisar la documentación tanto del sistema operativo como del lenguaje utilizado para manejar el terminal y sus funcionalidades.

En el Anexo A: Instalación de Ubuntu en Raspberry Pi se detalla la instalación del sistema operativo Ubuntu sobre una placa Raspberry Pi utilizada como servidor.

Estimación de posición con OpenCV-Python y OptiTrack

Una vez instalado y configurado el sistema operativo para utilizar la cámara Raspberry Pi para captura de fotogramas e instalado Python y las librerías de OpenCV, se estudia brevemente el concepto de visión artificial utilizando esta herramienta y su funcionamiento general para detectar objetos, movimiento y patrones, como el damero

utilizado. A continuación, se procede a adaptar varios ficheros sobre estimación de posición y orientación respecto a un damero genérico para OpenCV para poder ser utilizados en el proyecto y, además, proporcionando al usuario, y próximos proyectos, una interfaz de transmisión de vídeo en tiempo real de los fotogramas capturados por el módulo de cámara montado en el vehículo para tener una guía visual para trabajar de manera más eficiente y cómoda.

En cuanto a la estimación de distancia y ángulo, se debe calibrar la cámara para evitar que las distorsiones de las imágenes capturadas interfieran en el algoritmo de OpenCV, de lo contrario podría resultar en fallos evidentes en los valores de distancia mostrados en la terminal. Después de calibrar el módulo de cámara, hay que comprobar que se ha procedido correctamente por lo que se asegura mediante medidas físicas en laboratorio que el damero está a la distancia que afirma el programa de la Raspberry Pi, poniéndolo exactamente a 30 centímetros de distancia en el eje Z de la cámara. Si se detecta, tras varias iteraciones de calibrado y ensayo, que se existe un error consistente en la medida independientemente de la distancia observada, se corrige en el código dicha desviación.

Por otro lado, el denominado sistema de localización externo, es decir, las cámaras OptiTrack tienen delimitado cierta área del laboratorio para realizar los ensayos una vez las comunicaciones y la recopilación de datos esté completada. Se comprueba que las medidas visualizadas a través del *software* Motive se asemeja a las dimensiones de la realidad, de la misma forma que se hizo con el damero y la cámara de la Raspberry Pi secundaria. También se debe comprobar que cuales son los ejes de coordenadas establecidos por el programa y si es consistente con las medidas que se obtienen en los ensayos.

Estructura de comunicación entre dispositivos

Cada dispositivo del proyecto tiene una función concreta dentro de este e independiente del resto. Sin embargo, para poder cumplir los objetivos propuestos deben ser capaces de establecer comunicación entre ellos para enviar información recopilada, como la localización en cada instante de tiempo del vehículo, o realizar cambios en el funcionamiento

de los programas o ficheros. Por lo tanto, es necesaria una red de comunicaciones robusta, rápida y segura.

La primera conexión que se debe establecer es entre los ordenadores y los dispositivos Raspberry Pi, para ello se ha seleccionado el protocolo *Secure Shell* (SSH) ya que proporciona un enlace entre dispositivos totalmente encriptado y eficaz para acceso remoto desde un ordenador. Este protocolo se ha utilizado para los tres ordenadores del proyecto: el ordenador con el editor de código *Visual Studio Code* conectado a la Raspberry Pi secundaria, el ordenador que tiene *Motive* instalado y está conectado a las cámaras OptiTrack y, por último, el ordenador que ejecuta Matlab y Simulink y analiza los resultados que recibe de ambas Raspberry Pi.

La conexión entre las Raspberry Pi puede ser inalámbrica o mediante Ethernet. Como se detalla en el Capítulo 4. Se emplea finalmente un cable Ethernet para asegurar una baja latencia en el envío de los paquetes de información relacionados con la distancia y orientación relativas del vehículo según la interpretación del algoritmo de OpenCV ejecutado en la Raspberry Pi secundaria. Se realizan diferentes comprobaciones de conexión y verificación de la integridad de la información enviando, primeramente, valores individuales y observando el resultado recibido en la interfaz de Simulink. Tras completar las pruebas de comunicación, se pueden enviar los paquetes de posición en un espacio tridimensional y los ángulos de rotación de Euler para ser utilizados y analizados por los controles a través de Matlab.

Capítulo 2. ESTADO DEL ARTE

2.1 NAVEGACIÓN AUTÓNOMA

El concepto de autonomía en la industria aparece como alternativa a la mano de obra convencional que se llevaba utilizando desde la primera revolución industrial. En el caso de la producción en cadena, se sustituyó masivamente a la persona por el robot programado para hacer la misma tarea en menos tiempo, con mayor eficiencia y sin necesidad de descansar. Este es solamente un ejemplo de todos los sectores que han experimentado el avance de la robótica y automática en las últimas décadas.

Uno de los ámbitos más complicados de automatizar es la navegación de vehículos y robots. Esta idea se lleva trabajando desde la década de 1920, aunque no fue hasta la década de 1980 que se empezaron a diseñar los primeros vehículos autosuficientes como *Navlab* (Thorpe, Hebert, Kanade, & Shafer, 1988) de la Universidad de Carnegie Mellon (Pensilvania). No ha sido hasta la década de 2010 que este servicio ha levantado el interés real de inversores y empresas de la industria de la automoción como Ford o Tesla, y empresas tecnológicas que pretenden aprovechar este auge como Waymo (Google). En la década actual ya se han visto los primeros vehículos automatizados o *self-driving* dirigidos a un público general, pero la legislación correspondiente está aún poco desarrollada y es el principal impedimento para seguir implantándolo a la sociedad. Se espera que las grandes corporaciones cooperen en busca de un futuro totalmente autónomo.

El vehículo autónomo debe conocer con precisión tanto su ubicación como su camino a seguir. Los sistemas de localización se pueden dividir en exteriores, como GPS o GNSS (Naciones Unidas, 2013), e interiores (IPS), como Wi-Fi. El cálculo de posición, independientemente del tipo de sistema, puede ser realizado mediante métodos deterministas o por métodos de muestreo. Dentro del vehículo también debe haber componentes, como el acelerómetro y giroscopio, que puedan determinar variables relativas al mismo, como su orientación y aceleración.

Todos los datos proporcionados por el sistema de posicionamiento son interpretados por un control que decide el siguiente movimiento del vehículo. Con el avance en el ámbito del *machine learning*, los vehículos autónomos aprenden mientras trabajan, pudiendo distinguir a un obstáculo de una persona o situaciones que puedan ser seguras o peligrosas para la mercancía o pasajeros que transporte. No obstante, para poder hablar de un vehículo totalmente autónomo debe operar sin ningún tipo de acción externa, excepto en caso de averías, por tanto, también ser capaz de abastecerse de forma independiente de la energía que lo alimenta.

Por tanto, la autonomía en la navegación de un vehículo puede dividirse en cinco niveles claramente diferenciados según el requerimiento de la acción humana tanto en la conducción como en casos excepcionales.

Nivel 0: automatización nula

El nivel inicial de automatización en realidad representa un panorama de navegación que carece de autonomía de cualquier tipo, es decir, debe existir un conductor que decida en todo momento las acciones de navegación sin ningún tipo de ayuda externa. En este nivel puede clasificarse cualquier vehículo tripulado tradicional como coches antiguos o maquinaria.

Nivel 1: conducción asistida

En este nivel se sitúan aquellos vehículos dotados con algún sistema de asistencia en la conducción que proporcione ayuda tanto en la marcha o aceleración como la dirección mantenida por el vehículo, pero no puede controlar ambas de forma simultánea. Por tanto, el sistema actúa como corrección para despistes o fallos del tripulante al cargo de la navegación, resultando en una conducción mucho más cómoda que el método del nivel anterior. Este tipo de asistencia se conoce como ADAS (*Advanced Driver-Assistance Systems*) y establece los cimientos para las tecnologías posteriores basadas en inteligencia artificial y cognitiva (Li, Wen, Zheng, & Shen, 2011).

Nivel 2: autonomía parcial

El segundo nivel de automatización presenta la diferencia respecto al nivel anterior en la posibilidad de asistir a la vez la dirección de navegación y el frenado o acelerado de un vehículo. Sin embargo, se sigue requiriendo la existencia de un conductor puesto que debe atenderse a cualquier tipo de obstáculo al no tener sistema de detección.

Nivel 3: automatización condicionada

A partir del tercer nivel en adelante el vehículo cuenta con el sistema denominado ADS (*Automated Driving Systems*). Este nivel se refiere a cualquier tipo de vehículo que sea capaz de tomar decisión respecto a la navegación, como desviar la ruta o frenar ante obstáculos, ya que cuenta con sensores infrarrojos o láser para detectar cualquier objeto que se encuentre dentro de la trayectoria del vehículo. Incluso puede asistir al conductor y pasajeros para evitar daños mayores en caso de accidente. No obstante, son posibles situaciones en las que el criterio del conductor sea más acertado o necesario incluso, y se requiera su acción para determinar la consecución de cada caso concreto.

Nivel 4: automatización elevada

Para el este nivel el sistema ADS tiene control total sobre todos los elementos de conducción del vehículo y está diseñado para poder navegar correctamente sin necesidad de la intervención humana, incluso en los casos comentados en el tercer nivel. Sin embargo, el conductor puede tomar el control manual en cualquier momento si entiende que es necesario según la situación.

Nivel 5: automatización total

En el último nivel de la clasificación se categoriza a los vehículos que tienen la capacidad de realizar la navegación de forma casi idéntica a como lo haría un conductor humano. En este caso, el vehículo traza y sigue las trayectorias de navegación sin necesidad de un tripulante.

El vehículo terrestre utilizado en este proyecto y el sistema de control y navegación puede catalogarse dentro del quinto nivel, puesto que mediante la cámara como sistema de detección y el programa de Matlab-Simulink como sistema de navegación y control de trayectoria es capaz de realizar cierto recorrido preestablecido de forma totalmente autónoma sin intervención externa.

2.2 CARGA INALÁMBRICA

El concepto de transmisión de energía sin cables se conoce desde hace más de 150 años, con el descubrimiento de la inducción magnética por Michael Faraday. Para que este fenómeno ocurra es necesario una bobina transmisora de campo magnético variable y otra bobina receptora que enviará la corriente a su batería de carga. La sociedad actual se basa de manera fundamental en la tecnología de comunicación instantánea y sencilla movilidad. Para esto ha sido indispensable la transición de los dispositivos con cable al funcionamiento con baterías. En las últimas décadas, se ha ido transformado la manera en la que estas baterías se recargan, pasando de la conexión mediante cable a la red a la tecnología de carga inalámbrica como opción más cómoda y estilizada.

La transferencia inalámbrica de energía eléctrica se define como el envío de energía eléctrica desde una bobina emisora, conectada a la fuente de alimentación principal, mediante inducción electromagnética, hasta una bobina receptora conectada a la batería que se desea recargar. En la Figura 2.2.1 se observa un esquemático general de un circuito de carga inalámbrica en el que se basa el proyecto.

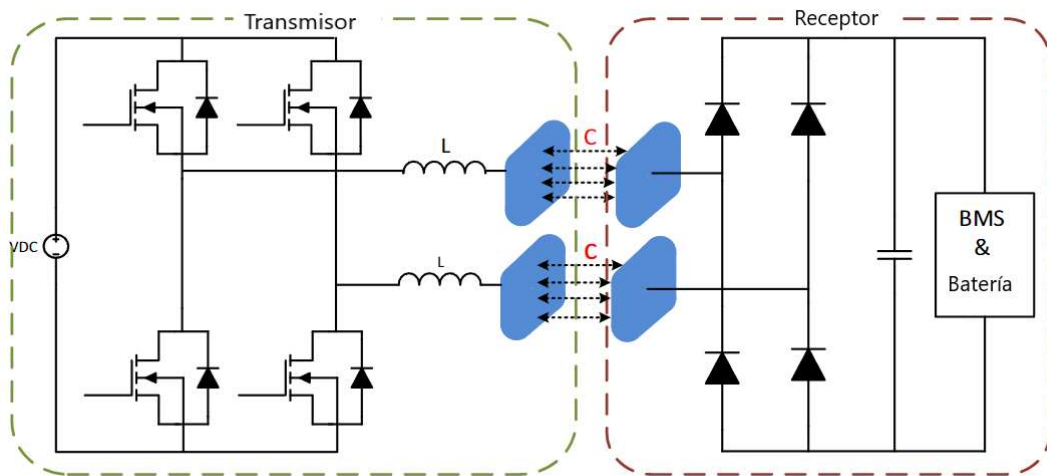


Figura 2.2.1 Diagrama de carga inalámbrica capacitiva (Panchal, Stegen , & Lu, 2018)

La transferencia inalámbrica de energía puede dividirse en distintos tipos según su distancia efectiva y longitud de onda. Generalizando se obtienen dos grupos destacados:

- Rango largo: su distancia efectiva máxima se considera superior a diez veces el diámetro de la antena emisora. Ejemplos de este tipo son las ondas microondas y las ondas de luz, ondas electromagnéticas.
- Rango corto: se denomina corto alcance cuando la energía no puede transferirse a una distancia mayor al diámetro de la bobina. Se considera de rango corto el acoplamiento resonante y la inducción magnética.

En la actualidad, existen diversas empresas que compiten buscando establecer su tecnología de transferencia inalámbrica de rango corto como el referente en la industria. Los tres estándares más expandidos son *Rezence*, *Powermat* y *Qi*. Los dos kits de carga inalámbrica estudiados en este trabajo son compatibles con la tecnología *Qi* por lo que se describe brevemente su funcionamiento y se analizará en mayor profundidad en el Capítulo 5.

El protocolo *Qi* se basa en la inducción magnética y representa la tecnología de carga inalámbrica de corto alcance con mayor expansión en la industria respecto al resto de sus competidores. *Wireless Power Consortium* (WPC) es la empresa encargada de su creación,

fundada por destacadas compañías tecnológicas, y plantea un modelo de negocio e investigación en continua expansión y desarrollo de la tecnología de transferencia de energía inalámbrica.

Por otro lado, es natural que la industria del automóvil inteligente y automático avance hacia tecnologías cómodas y eficientes, junto al compromiso climático de emisiones a la atmósfera, por lo que esta clase de vehículos han coincidido con otro sector puntero en la actualidad, la alimentación eléctrica. Los vehículos eléctricos actuales pueden recargarse mediante enchufe a la red eléctrica, que es el método más extendido actualmente por su montaje y uso sencillo, y la recarga inalámbrica, que aporta un mayor grado de comodidad, pero sigue en desarrollo para competir con la anterior. La industria automovilística está dejando atrás los combustibles fósiles y apostando por un modelo híbrido o totalmente eléctrico y limpio (Litman, 2017). No obstante, es necesaria la actuación humana para enchufar el coche a la estación de carga.

El problema principal de esta tecnología es su baja eficiencia energética respecto al cableado tradicional. En el ámbito de la recarga telefónica inductiva, el *Wireless Power Consortium* (WPC) realizó un estudio en 2015 que demuestra que los factores determinantes en el traspaso inalámbrico de energía son la distancia entre transmisor y receptor siendo necesario estar dentro del rango de acción para funcionar, mostrando un rendimiento del 60% con la tecnología *Qi* respecto a un cable convencional (Perzow, 2016). En el sector del automóvil eléctrico se pretende instalar este tipo de tecnología en las carreteras para poder cargar el vehículo sin necesidad de conexiones o incluso detener la travesía, obteniéndose buenos resultados respecto al rendimiento.

2.3 VISIÓN ARTIFICIAL

El concepto de visión artificial, o visión por ordenador, hace referencia a la aptitud de un ordenador o dispositivo de recoger, procesar y analizar las imágenes que captan y obtener cierto resultados deseados o información útil. Esta tecnología se ha expandido enormemente en las últimas décadas debido al impulso del mundo digital y su implementación en casi la totalidad de la sociedad: rutina diaria, industria, economía,

educación, transporte, etc. Las aplicaciones más habituales de la visión artificial son el reconocimiento facial, identificación y clasificación de objetos, inspección de defectos y navegación autónoma, que combinada con la inteligencia artificial tiene infinidad de utilidades.

Justo antes de la década de 1960 se dan los primeros desarrollos en técnicas de procesado de imágenes, concretamente en el año 1959 se inventa el primer escáner de imágenes capaz de transformar una imagen de muestra a tablas numéricas. Posteriormente, Larry Roberts, considerado por la comunidad de investigadores el precursor de la visión artificial, diseñó el proceso de convertir los parámetros de imágenes bidimensionales en información sobre el objeto tridimensional (GlobalData, 2020). Desde el inicio del siglo XXI el uso de esta tecnología se ha expandido exponencialmente, ya que las empresas tecnológicas más importantes, como Google, comenzaron a invertir en su desarrollo e implementación.

La denominada “industria conectada” necesita la combinación de la visión por ordenador y la inteligencia artificial para ser posible. En el sector de transporte, se plantea el reto de automatizar la totalidad de los vehículos del entorno ya que puede solucionar inconvenientes frecuentes como los atascos de tráfico y accidentes, además de incrementar la seguridad y reducir el consumo de energía actual. Esta infraestructura de intercomunicación y coordinación digital se denomina Sistema de Transporte Inteligente (ITS) y se sigue desarrollando para ser implementado en un futuro (Bertozzi, y otros, 2002).

Respecto a la librería utilizada en este proyecto, OpenCV, es la mayor librería de visión por ordenador del mundo. Se cuenta con más de veinte años de historia y desarrollo de la creación de Intel en 1999. Su objetivo principal inicial era proporcionar una infraestructura de visión artificial avanzada y optimizada, gratuita y abierta, para que los programadores y desarrolladores trabajasen sobre un sistema común, suponiendo una lectura y aporte de código más universalizado. Desde entonces se realizaron una serie de versiones de prueba, o *betas*, hasta 2006 que fue lanzada la primera versión al público. Unos años más tarde fue desarrollado OpenCV 2, que proporcionaba al usuario una interfaz para C++, resultando en un uso más cómodo y nuevas funcionalidades para la librería, y hoy en día

cuenta con interfaces en Python, Java y Matlab con soporte en distintos sistemas operativos como Linux, MacOS, Windows, iOS y Android. A partir de 2012 OpenCV está dirigida por una asociación sin ánimo de lucro y coopera con compañías del sector como Google, Microsoft, Sony e IBM, entre otros.

Capítulo 3. HARDWARE

3.1 RASPBERRY PI 3B+

Se identifica como Raspberry Pi a los pequeños ordenadores monoplaca creados por la organización benéfica *Raspberry Pi Foundation*, con sede en Reino Unido, que busca descubrir a las generaciones más jóvenes las ventajas de la computación y las tecnologías digitales.

En este trabajo se utilizan dos Raspberry Pi 3 Modelo B+ Revisión 1.3, Figura 3.1.1. La revisión final de la tercera generación de Raspberry cuenta con un procesador de cuatro núcleos Bradcom BCM 283480 64-bit, 1.4GHz de CPU y 1Gb de SDRAM; 2.4GHz y 5GHz de conexión *Wireless LAN* y Ethernet, 4 puertos USB 2.0, puerto CSI para módulo de cámara, puerto HDMI y puerto microSD (Raspberry Pi, 2018), como especificaciones generales. Las dimensiones de este modelo (85x56 mm) permiten que una Raspberry pueda ser utilizada cómodamente en proyectos de robótica y automática.



Figura 3.1.1 Raspberry Pi 3 Modelo B+ Revisión 1.3 (Raspberry Pi, 2018)

La primera placa, en adelante "RPi principal", está ubicada en el interior del vehículo y se encarga de dirigir y procesar todo el *software* relacionado con el control de movimiento y comunicaciones del vehículo con elementos externos como el sistema de cámaras OptiTrack o la otra Raspberry. Está alimentada por una batería con capacidad de 11.1V/3Ah con una salida de 5V/2h.

La segunda Raspberry Pi, en adelante "RPi secundaria", se encuentra en un soporte, impreso con tecnología aditiva, en la parte superior del vehículo. El objetivo de esta placa es capturar con la cámara, recopilar y enviar los datos de posición y orientación del vehículo respecto a un damero a la RPi principal. En la Figura 3.1.2 se muestran dos baterías de iones de litio recargables Samsung INR18650-25R conectadas a una placa de alimentación Geekworm X728 proporcionan 5V de salida a través de los pines de la RPi secundaria.



Figura 3.1.2 Placa de alimentación X728 conectada a RPi 4 (Geekworm, 2020)

3.2 MÓDULO DE CÁMARA RASPBERRY PI V2.1

La RPi secundaria tiene un módulo de cámara conectado en el puerto CSI, Figura 3.2.2, mediante un cable plano de 15 cm, Figura 3.2.1. El modelo utilizado es el *Raspberry Pi Camera Module v2.1*, versión mejorada del modelo original de 2016.

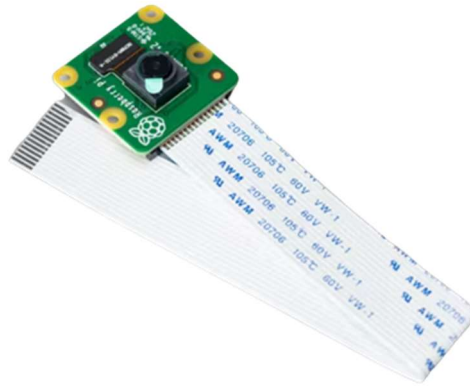


Figura 3.2.1 Módulo de cámara RPi v2.1 y cable plano (Raspberry Pi, 2016)

Las dimensiones físicas son 25x23x9 mm por lo que es ideal para aplicaciones que requieran traslado al tener tamaño reducido. La cámara contiene un sensor óptico SONY IMX219 de 8 megapíxeles con una lente de ojo de pez. La resolución de captura de vídeo utilizada en el proyecto es 640x480p a 90 fotogramas por segundo, aunque permite mayor resolución, como 1080p, en detrimento de la captura de fotogramas que se reduce a 30 fotogramas por segundo. Para la aplicación de este proyecto la resolución de 640x480p es suficiente para reconocer y trabajar con el damero.



Figura 3.2.2 Cámara conectada a la Raspberry Pi 3 (Raspberry Pi, 2016)

3.3 VEHÍCULO TERRESTRE

El trabajo ha desarrollado en base al vehículo utilizado en los laboratorios de las asignaturas de Regulación Automática, Control Digital y Control Avanzado del Grado en Ingeniería en Tecnologías Industriales (GITI) de la Escuela Superior de Ingeniería (ICAI) de la Universidad Pontificia Comillas. Este vehículo ha sido ensamblado en los talleres de la Universidad y su uso es exclusivo para docencia e investigación por parte de alumnos y profesores.

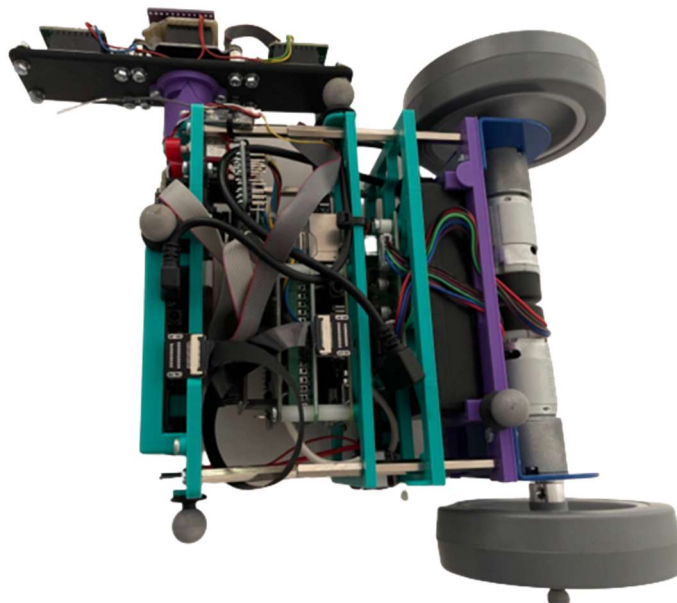


Figura 3.3.1 Vehículo terrestre con tracción diferencial (Fuente: Elaboración propia)

El vehículo dispone de dos motores DC (EMG 30) para tracción diferencial de sus dos ruedas traseras, que cuentan con *encoders* magnéticos para medir el ángulo girado, y un punto de apoyo esférico delantero. La RPi principal recibe y almacena las medidas provenientes de los sensores (IMU y *encoders* magnéticos), gobierna los PWM del *driver* MD25 de los motores, señala el estado del vehículo mediante un LED. Además, en este proyecto, recibe la posición y orientación relativa desde la RPi secundaria para la navegación hasta la base de carga inalámbrica. A su vez, recibe la información de posición y orientación proporcionada por el sistema de cámaras OptiTrack mediante un ordenador conectado por Wi-Fi a la RPi principal.

El vehículo tiene una batería capaz de proporcionar una tensión de 12V no regulados para los motores DC y una tensión de 5V regulados para los elementos electrónicos, entre ellos la RPi principal. Dicha batería es la que se busca recargar inalámbricamente en este proyecto. Además, cuenta con 3 pulsadores, dos de color rojo y uno de color negro, en la parte frontal utilizados para determinar la posición inicial del vehículo como el origen de coordenadas y comenzar el movimiento desde esa referencia creada al ejecutar el fichero de Matlab.

3.4 OPTITRACK

OptiTrack, propiedad de la empresa NaturalPoint, es el mayor distribuidor a nivel mundial de dispositivos relacionados con tecnología de captura de movimiento. Sus cámaras brindan una precisión rigurosa competente en el mercado, manteniendo precios accesibles para cualquier tipo de consumidor. También se especializan en *software* complementario para captura de movimiento como Motive, diseñado específicamente para trabajar junto a las cámaras OptiTrack. Los campos de aplicación de esta tecnología son la cinematografía, animación 3D, videojuegos, investigación del movimiento humano, entrenamiento profesional, entre otros.

En el apartado 1.3 se introducía el modelo utilizado en el laboratorio donde se localiza el proyecto, siendo este el modelo Flex 13, dispuestas en el techo a tres metros de altura delimitando un entorno de trabajo en forma de rectángulo de aproximadamente siete metros de largo y cinco metros de ancho, como puede observarse en la Figura 3.4.1. Las cámaras soportan conexión por USB 2.0 y se dividen por grupos para organizarse en tres multiplexores OptiSync que están conectados al PC, también por USB 2.0, que ejecuta Motive para enviar los datos de posicionamiento del vehículo.

Su lente tiene 1.3 megapíxeles de resolución ofreciendo 4 veces más resolución que su modelo anterior, filtro infrarrojo de paso alto, 56° de campo de visión horizontal, 46° de campo de visión vertical y captura de vídeo ajustable desde 30 hasta 120 fotogramas por segundo con una latencia de 8.3 ms. Cuenta con distintos tipos de procesado de imagen en

escala de grises que permiten, junto a la elevada tasa de refresco del sensor, obtener medidas de alta precisión.

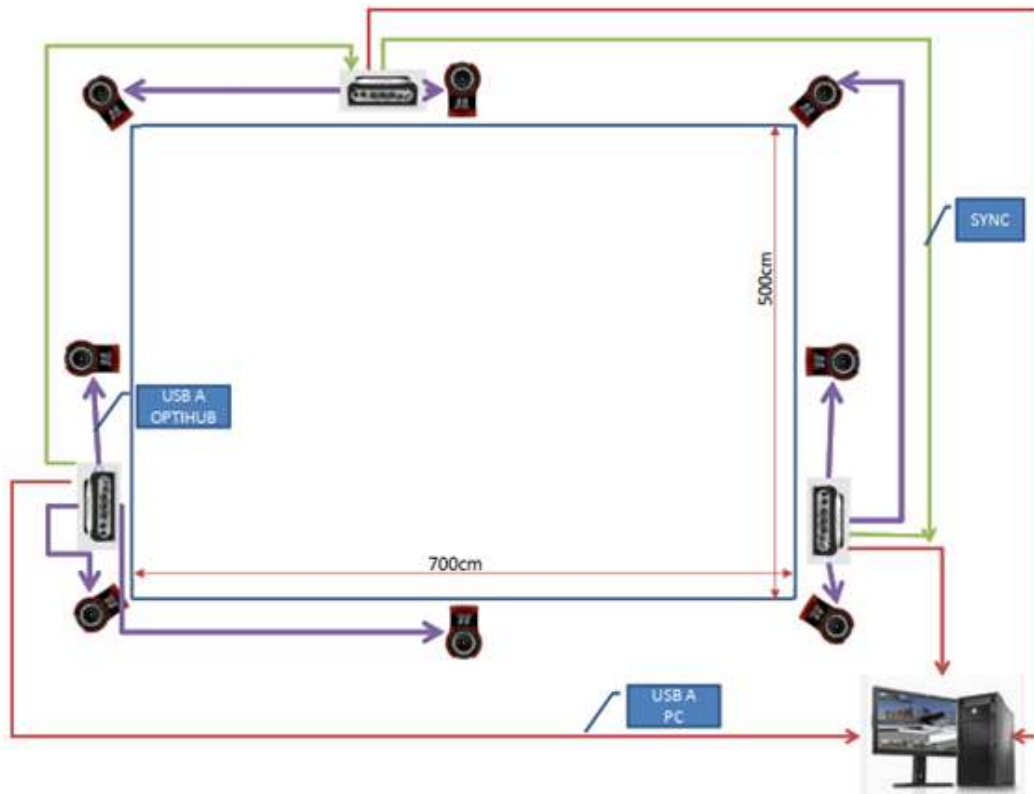


Figura 3.4.1 Diagrama de conexión sistema OptiTrack (Fuente: Departamento de Electrónica)

3.5 IMPRESIÓN 3D DE SOPORTES

Tanto el acoplamiento de la RPi secundaria y la cámara al vehículo por la parte superior (Figura 3.5.3) como los soportes de la bobina receptora en la parte trasera del vehículo (Figura 3.5.1) y la bobina transmisora para la estación de carga (Figura 3.5.2) han sido diseñadas en el software SolidWorks e impresas mediante tecnología de fabricación aditiva con una impresora de tipo FDM (Fused Filament Fabrication).

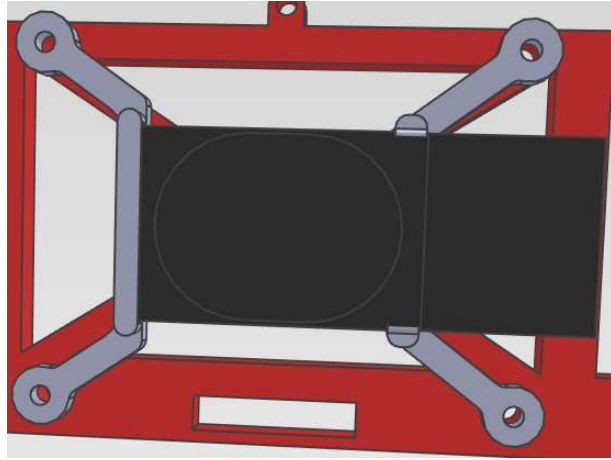


Figura 3.5.1 Soporte para bobina receptora

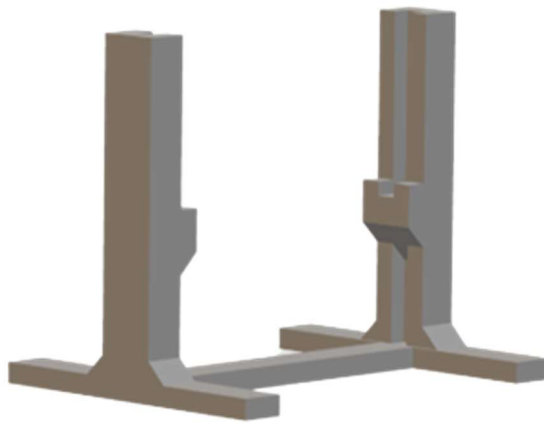


Figura 3.5.2 Soporte para bobina receptora

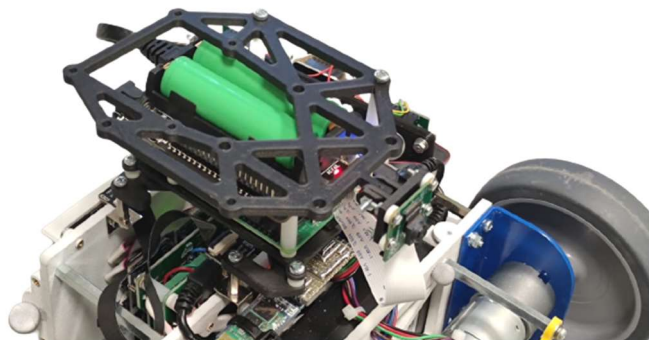


Figura 3.5.3 Soporte para RPi y cámara

Capítulo 4. SOFTWARE Y COMUNICACIÓN

4.1 RASPBERRY PI SECUNDARIA

En este apartado se presentan todos los programas y ficheros relacionados con la comunicación y funcionamiento de la denominada Raspberry Pi secundaria en el proyecto.

4.1.1 SISTEMA OPERATIVO

El sistema operativo escogido para la RPi secundaria es Ubuntu Server 20.04 LTS, desarrollado y publicado por empresa informática Canonical Ltd. Ubuntu Server es un sistema operativo servidor de código abierto diseñado para funcionar en cualquier hardware o máquina virtual. Se utiliza en páginas web e intercambio de archivos con una estabilidad y protocolos de seguridad impecables. En concreto, la versión 20.04 LTS (*Long-term Support*) está basada en el *kernel* Linux 5.0 y tendrá soporte hasta 2025, por lo que también es una herramienta duradera. (Petersen, 2020)

La aplicación de este proyecto busca compartir datos con otro dispositivo, por lo que Ubuntu Server 20.04 LTS es una opción excelente. Aunque el componente de seguridad no es determinante en este caso, por ser una red cerrada, para aplicaciones en empresas y corporaciones es significativa un factor decisivo para escoger el sistema operativo.

Se debe instalar la imagen del sistema operativo en una tarjeta MicroSD. Algunos parámetros importantes en el proceso de instalación son: crear el perfil de usuario, registrar una red de conexión Wi-Fi y activar el protocolo SSH para realizar la comunicación entre RPi secundaria y cualquier PC conociendo la IPv4 de la Raspberry. De esta manera, se asegura el control de la RPi desde un ordenador mediante acceso remoto, Por tanto, una vez introducida la tarjeta MicroSD en la RPi secundaria no se vuelve a interactuar de forma analógica, se realizará todo el proceso mediante el editor de código *Visual Studio Code*.

4.1.2 PROTOCOLO SSH

El protocolo criptográfico de seguridad *Secure Shell* (SSH) permite el acceso remoto y conexión entre redes seguro. SSH consta de tres componentes principales:

- Protocolo de transporte: el cliente solicita acceso al servidor. Se realiza mediante una conexión TCP/IP, típicamente.
- Protocolo de autenticación: el servidor verifica al cliente y le envía la información necesaria para el siguiente protocolo
- Protocolo de conexión: se genera un multiplexado de canales de conexión a partir del túnel encriptado.

El protocolo de conexión funciona por encima del protocolo de autenticación y transporte, y a su vez, el protocolo de autenticación trabaja en una capa superior al protocolo de transporte (Ylonen & Lonvick, 2006). En la Figura 4.1.1 se puede observar un diagrama del funcionamiento de las capas del proceso de comunicación SSH.



Figura 4.1.1 Diagrama de funcionamiento del protocolo SSH (Fuente: elaboración propia)

Para este trabajo, el cliente es un ordenador cualquiera y la RPi secundaria tiene la función de servidor. La conexión a un servidor mediante SSH puede realizarse desde cualquier cliente que tenga acceso a una consola. Para realizar la primera comprobación de la correcta instalación de Ubuntu en la RPi secundaria se utiliza consola de Windows del ordenador. A nivel de acceso del cliente, la única información necesaria es el nombre del cliente que el servidor tiene registrado y la dirección IPv4 del servidor en la red a la que ambos están conectados. Para realizar la conexión SSH entre cliente y servidor se debe escribir en el terminal la siguiente línea:

ssh "usuario"@IPv4"

ejemplo: ssh pi@192.168.27.191

Si la conexión es satisfactoria se verifica la instalación del sistema operativo. Esto se refleja al obtener una respuesta de la RPi, solicitando la contraseña del usuario. Al introducirla se tiene acceso a los archivos de la Raspberry de forma remota y se puede trabajar, con algunas limitaciones de permisos, sobre estos. Esto ocurre por la seguridad del propio sistema operativo para no brindar control total a cualquier usuario, aunque no supone ningún problema para la consecución del proyecto.

4.1.3 OPENCV-PYTHON EN VISUAL STUDIO CODE

El entorno de *Visual Studio Code* permite al usuario editar, crear y manejar los archivos de un servidor mediante la extensión de SSH, por lo que se prefiere antes que el simple uso de un terminal.

El proyecto requiere la instalación de los paquetes de extensión de Python, un lenguaje de programación basado en el uso de indentaciones para ordenar y administrar el código. El motivo principal de utilizar Python es que permite utilizar de forma útil la librería de funciones de visión computacional OpenCV, cuya implementación inicial es en C++, y es fundamental para la captura de parámetros de posición y rotación del vehículo. La versión descargada para trabajar es Python 3.10.4.

OpenCV-Python es una API (*Application Programming Interface*) de Python de código abierto, que combina los mejores aspectos de la C++ API original y el lenguaje de programación Python. OpenCV está diseñado para resolver los retos planteados por la visión artificial por ordenador haciendo uso, principalmente, de la librería Numpy, que contiene diversas operaciones matemáticas altamente optimizadas para soportar y manejar matrices y *arrays* multidimensionales. Todas las estructuras de OpenCV siempre se transforman a través de los *arrays* de Numpy. Por lo que es necesario revisar la documentación de Python y Numpy para comenzar en el ámbito de OpenCV.

4.1.4 CALIBRACIÓN DE CÁMARA RPI

Tras revisar la documentación descrita y realizar una conexión exitosa con SSH a través de *Visual Studio Code*, se procede a efectuar los preparativos para la detección del damero mediante un módulo de cámara procesado por OpenCV.

Previo a la creación de ficheros o código deben instalarse las siguiente librerías y herramientas en el orden indicado introduciendo los siguientes comandos, sin comillas, en el terminal de *Visual Studio Code*:

- Actualizaciones y paquetes de Raspberry Pi: “sudo apt-get update” y “sudo apt-get upgrade”
- Organizador de paquetes Pip: “sudo apt install python3-pip”
- OpenCV-Python: “pip install opencv-python”
- Administración de ficheros: “sudo apt install libgl1-mesa-glx” y “sudo apt-get install ffmpeg libsm6 libxext6 -y”
- Librerías para OpenCV: “pip install numpy”, “pip install glob2”, “pip install imutils”, “pip install tqdm”, “pip install pandas”, “pip install multiprocessing”, “pip install flask”, “pip install argparse”, “pip install datetime”.

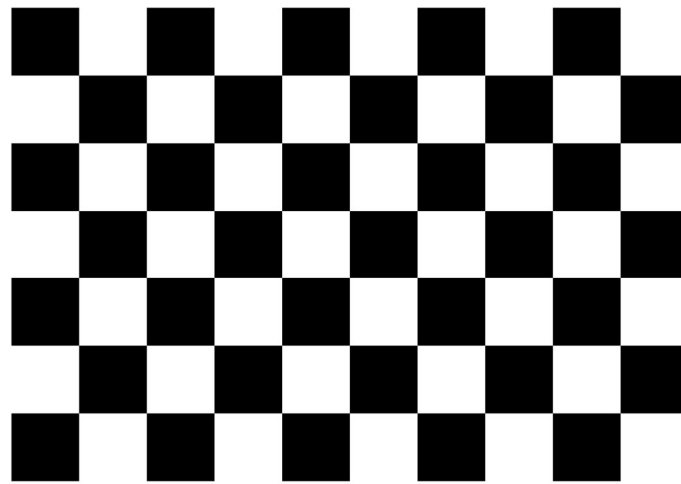
Adicionalmente, se debe editar el fichero *config.txt*, alojado en el directorio *\boot\firmware*, para activar el módulo de cámara ya que al utilizar un sistema operativo distinto a Raspbian no viene activado por defecto. Se deben añadir las siguientes líneas de código al fichero original:

```
gpu_mem=128
disable_camera_led=1
start_file=start4x.elf
fixup_file=fixup4x.dat
```

Tras instalar las librerías y modificar el fichero anterior se debe reiniciar la RPi secundaria para aplicar todos los cambios. Una vez reiniciada se puede comprobar que la cámara está conectada y detectada mediante el comando: *vcgencmd get_camera*. Si se ha

procedido correctamente se debe obtener como respuesta: *supported=1 detected=1* por el terminal.

Se va a estimar la posición del vehículo mediante el tablero de ajedrez o damero de dimensiones 9x6 esquinas interiores de la Figura 4.1.2, es decir, es un damero de 10x7 casillas, pero OpenCV se basa en la detección de esquinas interiores para reconocer el damero.



This is a 9x6
checkerboard
<https://opencv.org/>

Figura 4.1.2 Damero 9x6 (OpenCV, 2013)

El módulo de cámara RPi v2.1 es una cámara pequeña sin un lente objetivo. Esto se debe a que un objetivo puede resultar muy costoso y ocupa bastante espacio, no siendo óptimo para una aplicación como la de este proyecto. Debido a este diseño, pueden producirse ciertas distorsiones en la imagen capturada, resultando en fallos al estimar la posición al producirse una desviación en la proyección rectilínea.

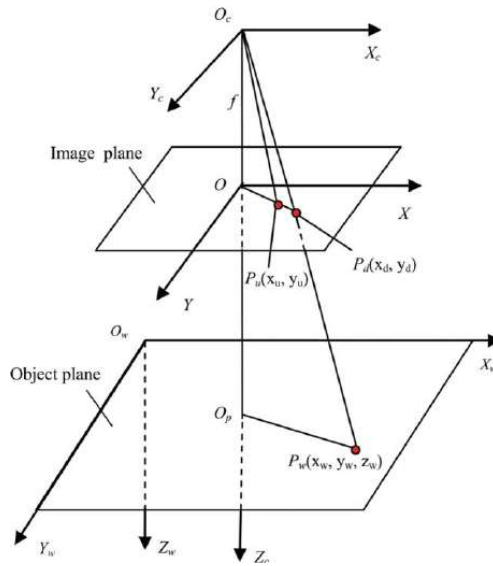


Figura 4.1.3 Modelo de cámara sin lente con distorsión radial (Li & Zhang, 2011)

Existen dos tipos principales de distorsión de imagen: distorsión radial y distorsión tangencial (Bradski & Kaehler, 2000). La distorsión radial afecta a las rectas de una imagen haciendo que aparezcan con cierta curvatura. De manera similar, la distorsión tangencial ocurre porque el plano de imagen no es completamente paralelo al plano de la lente de la cámara, generando la ilusión de ciertas partes de la imagen estar más cerca que otras. En la Figura 4.1.4, se puede observar un ejemplo de distorsión real y su corrección, para demostrar la importancia de corregir estas desviaciones.



Figura 4.1.4 Distorsión radial (izquierda) y corrección (derecha) (Kukelova & Pajdla, 17-22 de junio de 2007)

La expresión de la distorsión radial se representa en la Ecuación 4.1.4.1:

$$x_{distorsión} = x * (1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6)$$

$$y_{distorsión} = y * (1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6)$$

Ecuación 4.1.4.1 Distorsión radial

Análogamente, la expresión de la distorsión tangencial se representa en la Ecuación 4.1.4.2:

$$x_{distorsión} = x + [2p_1 \cdot xy + p_2(r^2 + 2x^2)]$$

$$y_{distorsión} = y + [2p_2 \cdot xy + p_1(r^2 + 2y^2)]$$

Ecuación 4.1.4.2 Distorsión tangencial

Observando las expresiones, se concluye que son necesarios cinco parámetros de distorsión para poder formar la matriz de distorsión de dimensión 1x5:

$$\text{Matriz de distorsión} = [k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3]$$

Figura 4.1.5 Parámetros de la matriz de distorsión

Además, para calibrar la cámara, se necesitan coeficientes adicionales a la distorsión denominados parámetros intrínsecos y extrínsecos de la cámara.

Los parámetros intrínsecos se refieren a las especificaciones propias de la lente del módulo: la distancia focal (f_x, f_y) y centro óptico (c_x, c_y), ambas representadas en milímetros. Estos parámetros son propios de cada cámara y con estos, puede formarse la matriz de la cámara de dimensión 3x3, que se representa con la expresión 4.1.6, y sirve para transformar las coordenadas:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Figura 4.1.6 Conversión con matriz de cámara

Por otra parte, los parámetros extrínsecos se refieren a los vectores de traslación y rotación. Estos vectores transforman las coordenadas de un punto en el espacio a la base de un sistema de coordenadas (OpenCV, 2013).

Para poder corregir tales distorsiones, se debe proporcionar un grupo de muestras del damero fotografiadas con la cámara para que OpenCV calcule los parámetros de distorsión y de cámara. Se necesitan de 80 a 100 muestras, aproximadamente, para obtener una caracterización aceptable con el menor error posible en la aplicación. Estas muestras se realizan con el fichero de Python *take_images.py*, que captura una imagen de 640x480p, para mayor optimización, cada 2 segundos y las almacena en la carpeta *cal_images*. Mediante el código está funcionando, se debe mostrar el damero a la cámara, asegurando que en todas las imágenes el damero se encuentra entero con todas las casillas visibles, sin cortar ninguna esquina, tampoco muy alejado ni demasiado inclinado, puesto que OpenCV no lo detectará correctamente y creará un modelo erróneo de la cámara. No obstante, debe mostrarse en distintas orientaciones y distancias sobre los 3 ejes de coordenadas alrededor de todo el rango de visión para que el modelo sea lo más fino posible.



Figura 4.1.7 Imágenes muestra de damero para calibrado (Fuente: Elaboración propia)

Después de tomar todas las muestras hay que seleccionar las imágenes óptimas para la calibración atendiendo a las pautas comentadas anteriormente, eliminando de la carpeta *cal_images* las muestras no deseadas.

A continuación, se ejecuta el archivo *camera_calibration.py*. Este código se encarga de procesar todas las imágenes de muestra individualmente mediante OpenCV, según el número de esquinas horizontales (6) y verticales (9), y longitud de cuadrado en metros (0.023 m, en este caso) que el usuario indique. Además, se establecen unos criterios generales para OpenCV y una referencia espacial según los parámetros comentados. En primer lugar, convierte la imagen a escala de grises, para poder realizar un mejor procesado, y buscar las esquinas del tablero mediante una función de la librería OpenCV denominada *findChessboardCorners*. Si el algoritmo encuentra los bordes, genera su interpretación del tablero en la imagen original, dibujando las esquinas encontradas, mostrándola al usuario si la detección es adecuada.

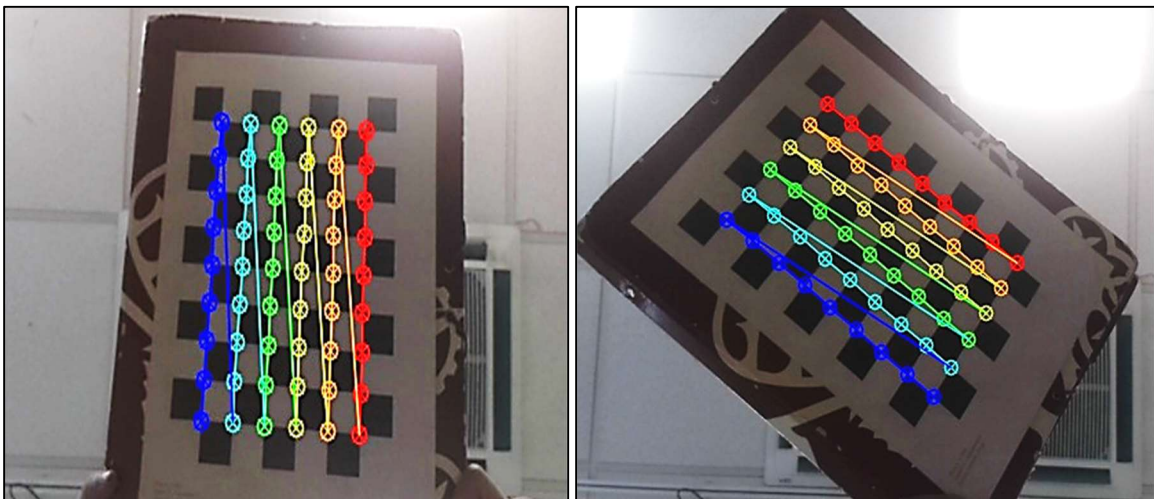


Figura 4.1.8 Imágenes de muestra con esquinas detectadas (Fuente: Elaboración propia)

En la Figura 4.1.8 se observa que el algoritmo ha detectado correctamente las esquinas del damero para sendas muestras, ya que ha dibujado de forma ordenada los puntos 3D en cada esquina desde el primer punto coloreado en rojo hasta el punto final coloreado en azul. La esquina inicial roja es la que el algoritmo ha detectado como sistema de coordenadas del tablero y devolverá los datos de posición y rotación con respecto a este. Si se halla alguna muestra con los puntos mal localizados, ligeramente desviados o con iluminación pobre, se debe eliminar del grupo y volver a realizar el calibrado.

Por último, procede a realizar los cálculos oportunos a partir de las operaciones anteriores mediante la función *calibrateCamera* para obtener tanto las matrices de cámara (*mtx*) y de distorsión (*dist*), como los vectores de rotación (*rvecs*) y traslación (*tvecs*). Esta información se almacena en dos ficheros distintos:

- *calibration_chessboard.yaml*: es un fichero escrito en lenguaje YAML, indicado para almacenar información en una estructura de cómoda lectura y acceso para poder consultarlo en cualquier momento.
- *C.npz*: la extensión *.npz* (*numpy.savez*) pertenece a la librería Numpy. Este formato se utiliza para guardar *arrays* sin comprimir en un mismo archivo ZIP, pero no se puede acceder a la información directamente, es necesario pasar por una función de carga de la librería Numpy. Por tanto, este fichero sirve para cargar las matrices y vectores necesarios en otros ficheros de código conociendo su directorio.

El archivo *camera_calibration.py* también devuelve el error normalizado total de estimación de los parámetros que ha calculado. Es por esto por lo que hay que realizar varias iteraciones de calibración buscando el menor error normalizado posible. En este trabajo, se ha conseguido un error mínimo de 0.03995203, visto en otros trabajos que por debajo de 0.04 es correcto, con la siguiente matriz de cámara y coeficientes de distorsión:

$$\begin{bmatrix} 516.85527912 & 0 & 317.33341116 \\ 0 & 518.79099074 & 237.42665988 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[0.17941819 \quad -0.19701886 \quad 3.6762884e - 4 \quad -2.0228543e - 3 \quad -0.40088372]$$

Figura 4.1.9 Matriz de cámara y parámetros de distorsión del módulo de cámara RPi (Fuente: Elaboración propia)

Si se compara la estructura de la Figura 4.1.5 y Figura 4.1.6 con la Figura 4.1.9, se observa que la dimensión y estructura es idéntica y, además, el orden de los dos parámetros focales es el mismo por lo que se acepta este resultado para continuar el trabajo y se el módulo de cámara está calibrado correctamente. Si por alguna razón al ensayar con estos

parámetros existiese alguna discrepancia espacial, simplemente se deberían seguir otra vez los pasos de calibrado para obtener unos parámetros más apropiados.

4.1.5 SISTEMA DE DETECCIÓN Y POSICIONAMIENTO

En la sección 4.1.4 se menciona la manera en la que OpenCV caracteriza la detección del tablero. Antes de comenzar a recopilar la información angular y posicional de la cámara se debe conocer cómo se han establecido los ejes de coordenadas respecto al tablero para poder realizar cambios si difieren con los del vehículo. Para ello se crea el fichero *ejes.py* que procesa un fotograma de muestra con el damero visible generando un nuevo archivo de imagen *.jpeg* con la interpretación del sistema de coordenadas de OpenCV superpuesta.

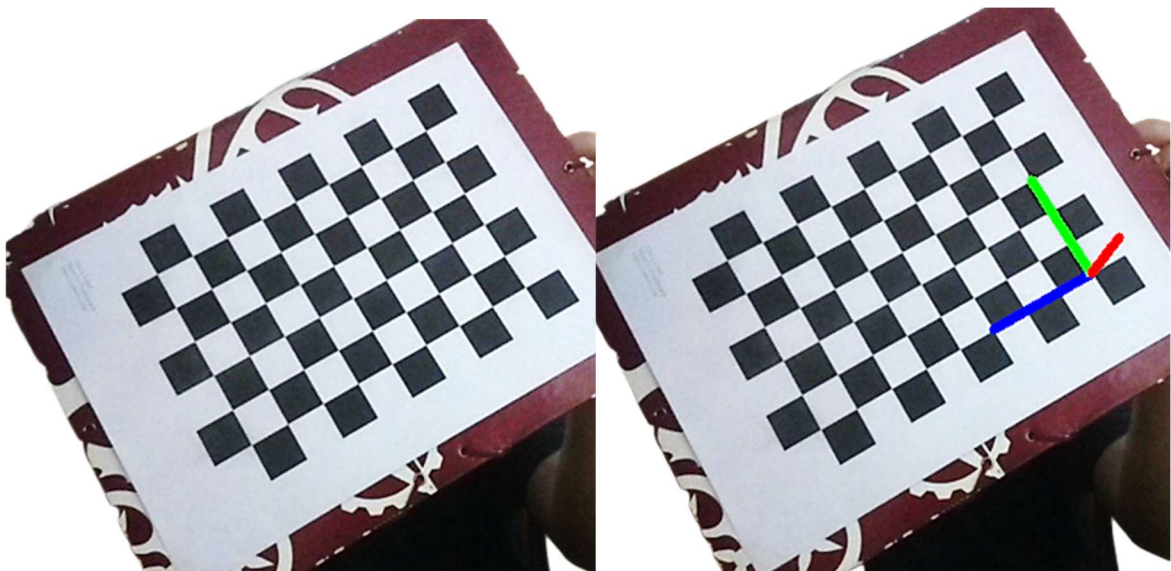


Figura 4.1.10 Comprobación de sistema de coordenadas de damero (Fuente: elaboración propia)

La Figura 4.1.10 confirma lo comentado anteriormente sobre la Figura 4.1.8 respecto a la primera detección de esquina, en este caso alojada en la esquina superior derecha del table cuando se encuentra en posición vertical. Este sistema de ejes es invariable para el damero y representa el cero absoluto para la referencia de la cámara. El eje azul corresponde al eje X, el eje verde es el eje Y y el eje rojo se adopta como el eje Z. Asimismo, los ángulos de giro se asocian a esta base siendo el ángulo ' u ' el giro respecto al eje X, denominado

alabeo; el ángulo ‘ v ’ respecto al eje Y, denominado guiñada; y el ángulo ‘ w ’ respecto al eje Z, denominado cabeceo.

Para este trabajo los ejes más significativos son el eje X y Z para realizar el acercamiento y alejamiento desde distintos lugares a la base de carga inalámbrica. El ángulo de rotación más significativo es la guiñada, *yaw* en inglés, ‘ v ’ alrededor del eje Y puesto que el vehículo permanece en una superficie plana y no experimenta cabeceo ni alabeo. Por tanto, en los primeros ensayos se observará que las seis dimensiones sean correctas, pero se presta especial atención a las dos dimensiones de localización y la dimensión angular señaladas.

Tras realizar la comprobación y caracterización de los ejes se puede comenzar a estimar la posición de la cámara respecto al damero. Para ello se utiliza el archivo escrito en lenguaje Python: *pose_estimation.py* y tiene tres funciones principales.

El primer proceso es utilizar las funciones de OpenCV para conocer la posición y giro de la cámara respecto al tablero en tiempo real. Se importan los parámetros que el archivo *SingleMotionDetector.py* ha recopilado. Dichos parámetros delimitan un rectángulo en cada fotograma para crear una distinción entre cualquier objeto en movimiento en movimiento y un fondo constante. Además, se cargan la matriz de cámara y los coeficientes de distorsión almacenados en el fichero *C.npz* y el usuario debe escribir en el código el número y longitud de casillas verticales y horizontales por si se utiliza un tablero distinto al de la calibración. Mientras recibe los fotogramas en tiempo real comienza a realizar el preprocesado de imagen: reescalado, escala de grises y reducción de ruido gaussiano. La imagen procesada es óptima para aplicar la detección de esquinas de OpenCV. Se realizan dos detecciones. La primera sirve detección para confirmar que se está viendo el tablero y estima sus esquinas. Una vez el algoritmo ha confirmado que se ha detectado el damero se vuelve a estimar con mayor precisión las esquinas con la función *cornerSubPix*, cuya interpretación de esquinas se utiliza para resolver los cálculos con la función *solvePnP*, que determinan la posición y rotación respecto al damero en las variables *tvecs* y *rvecs*, respectivamente. El orden de las variables estimadas es (X, Y, Z) para la posición en metros y (u, v, w) para los ángulos de rotación en grados. Esto resulta conveniente puesto que el

usuario recibe la información directamente en unidades del Sistema Internacional y un sistema de referencia basado en las transformaciones de ángulos de Euler.

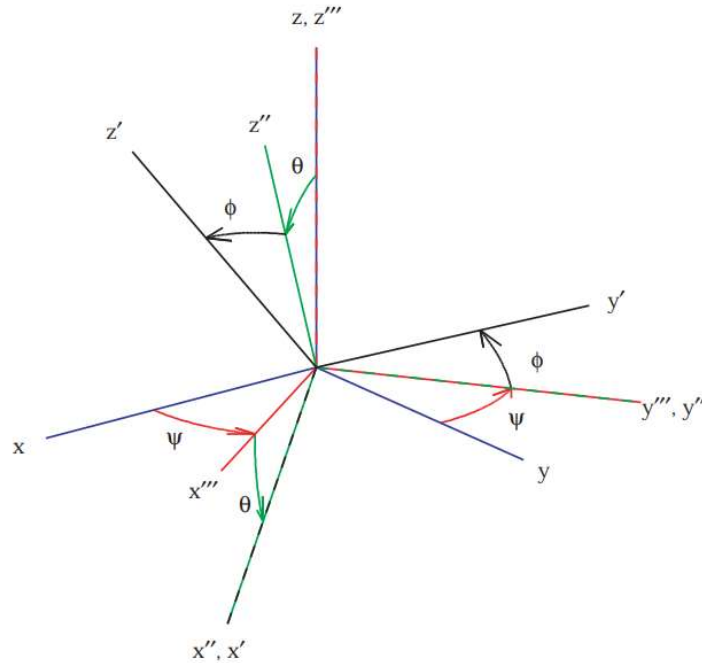


Figura 4.1.11 Secuencia de Ángulos de Euler obtenida por OpenCV (Diebel, 2006)

4.1.6 INTERFAZ DE VÍDEO

La segunda funcionalidad de *pose_estimation.py* no es necesaria para el proceso de detección, pero representa un apoyo importante al usuario pues muestra en tiempo real la visión de la cámara a través de una interfaz programada en HTML. Para obtener el vídeo del módulo de cámara se añade el popular *web framework Flask*, escrito en Python y adaptado para trabajar con este de forma sencilla. Hay un *framework* denominado Django, que también es otro ampliamente utilizado en todo tipo de desarrollos *web*. Ambas son herramientas altamente populares entre desarrolladores. Sin embargo, Flask presenta una estructura óptima para aplicaciones simples, como una interfaz de transmisión de vídeo en vivo o vigilancia, dada su flexibilidad y control de parámetros. Por otra parte, Django está diseñado para trabajos en escalas mayores como, redes sociales o comercio internacional (Ghimire, 2020). Es por esto por lo que sendas opciones son totalmente válidas para esta

aplicación, sin embargo, Flask parece la opción más útil y ligera para desarrollar al programar y utilizar como consumidor.

Para transmitir todos los fotogramas desde la RPi secundaria se necesita combinar la librería *imutils*, mencionada anteriormente en el apartado 4.1.4, y Flask. La librería *imutils* tiene funcionalidades de procesado básico de imagen que habilitan la comunicación entre la interfaz *web* y la cámara. Por tanto, el archivo *pose_estimation.py* almacenará los fotogramas capturados para que el usuario pueda acceder a ellos en tiempo real a través de cualquier navegador *web*, introduciendo en la barra de búsqueda la dirección IPv4 asignada a la RPi secundaria (HOST) y el puerto (PORT) que se haya establecido en el código, con la siguiente estructura:

`https://HOST:PORT/` ejemplo: `https://192.168.27.191:8000/`

Por tanto, cualquier usuario con un ordenador o dispositivo móvil conectado a la misma red que la RPi secundaria, podrá monitorizar en tiempo real las imágenes que se están recopilando mediante la cámara. No obstante, se debe diseñar una interfaz para que se muestre el vídeo, de lo contrario al introducir la dirección mencionada simplemente se encontraría una página en blanco. Por tanto, se debe generar un fichero de formato HTML para dotar de estructura a la captura enviada. Dentro del fichero *index.html* se dispone de un apartado para modificar las dimensiones, filtro, marco, margen y posición del vídeo.

En resumen, se genera un objeto *VideoStream* que se convierte en cada fotograma generado por la cámara y que se envía al cliente la dirección deseada. Estos fotogramas se transforman al formato de imagen JPEG, utilizando las funcionalidades de OpenCV de procesado multimedia, para poder ser mostradas la *web* del fichero *index.html* comentado en el párrafo anterior. Gracias al desarrollo de esta interfaz se pueden realizar los experimentos de detección y navegación de manera óptima.



Figura 4.1.12 Interfaz *web* de transmisión de video (Fuente: Elaboración propia)

4.2 COMUNICACIÓN ENTRE LAS DOS RASPBERRY PI

La tercera función del fichero principal *pose_estimation.py* es enviar los seis parámetros en cada ciclo a través de WLAN o Ethernet a la RPi principal, para que ejecute sus rutinas en base a estos datos.

En primer lugar, previo a enviar cualquier información a la RPi principal, se realiza una comprobación inicial de envío y recibimiento de paquetes de datos sobre la propia RPi secundaria, es decir, se dispone tanto como servidor como cliente al mismo dispositivo antes de probar vías de comunicación reales por cable o inalámbricas.

Como forma de verificación sobre si se codifican y envían correctamente los datos por parte del cliente, se ensambla un valor fijo en un *array* que se denominará *data*, por ejemplo, los valores de distancia y ángulos que OpenCV esté detectando en cada ciclo del

programa. Para poder mandar estos números se utilizan las librerías *socket* y *json*. Un *socket* o enchufe, es un punto final demarcado en una conexión de Internet, por tanto, la librería *socket* implementa de forma básica mediante varios objetos, funciones, constantes y reglas este tipo de enlaces entre dispositivos para poder realizar las comunicaciones, soportando los protocolos de conexión IP4 y IP6, deseadas en un programa. Mediante la función *send* se envía la información al *socket* definido, que debe estar conectado al nodo cliente. Asimismo, dicha información debe ser codificada antes de ser mandada al *socket* y decodificada cuando se alcance su destinatario. De ello se encarga la librería *json*, *JavaScript Object Notation*, un tipo de sintaxis para guardar e intercambiar información. El *array* mencionado anteriormente debe escribirse en la notación JSON para poder manejar de forma independiente el contenido de este y, además podrá ser codificado siguiendo el formato.

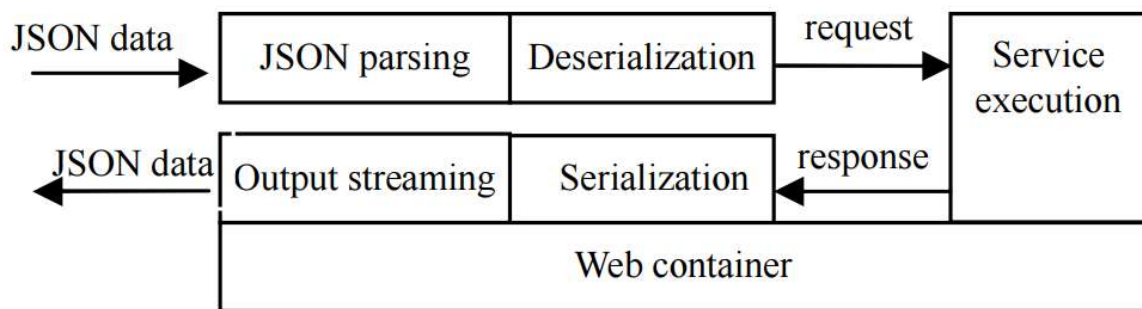


Figura 4.2.1 Funcionamiento de paquetes con JSON (Peng, Cao, & Xu, 2011)

Por tanto, en la primera prueba de comunicación el cliente envía un paquete con seis datos que se pueden emplear y gestionar de manera independiente al ser recibidos. No obstante, debe existir un *socket* o servidor al que lleguen todos los paquetes de información. Para ello se utiliza el fichero denominado *server.py*, que define una dirección y puerto donde se alojará dicho servidor. En esta primera iteración, tanto el servidor como el cliente pertenecen al mismo dispositivo Raspberry Pi. Sin embargo, a efectos de comunicación no representa ningún beneficio o distinción para establecer la conexión y envío de información desde el cliente hasta el servidor. El archivo de código *server.py*, también haciendo uso de las librerías *socket* y *json*, deshace la codificación de los datos del cliente y puede mostrar a través del terminal la información completa del mensaje o solamente el parámetro deseado para comprobar que los datos no sufren ninguna transformación indeseada en el proceso. Si

existe algún problema de conexión en el envío o recepción de los paquetes, o se desconecta la comunicación de forma abrupta, el programa notifica al usuario mediante un mensaje de error en el terminal. Para realizar la comprobación se debe ejecutar el fichero *server.py* en primer lugar, y posteriormente ejecutar el archivo principal *pose_estimation.py*, de lo contrario se produce el error denominado *BrokenPipeError* al intentar establecer una ruta de comunicación con un puerto de una dirección IP que no está abierto en ese instante.

Existen diversos protocolos de comunicación y transmisión de datos en internet, aunque los más importantes son los protocolos denominados *Transmission Control Protocol* (TCP) y el *Internet Protocol* (IP), combinando sus funciones resulta el protocolo TCP/IP *suite*, o Conjunto TCP/IP en español. En este trabajo se utilizará TCP/IP para hacer referencia a la totalidad del protocolo. El objetivo de diseño principal de TCP/IP es construir una red interconectada para brindar medidas de comunicación frente a las conexiones físicas tradicionales para poder enlazar diversos clientes y servidores de cualquier parte del mundo.

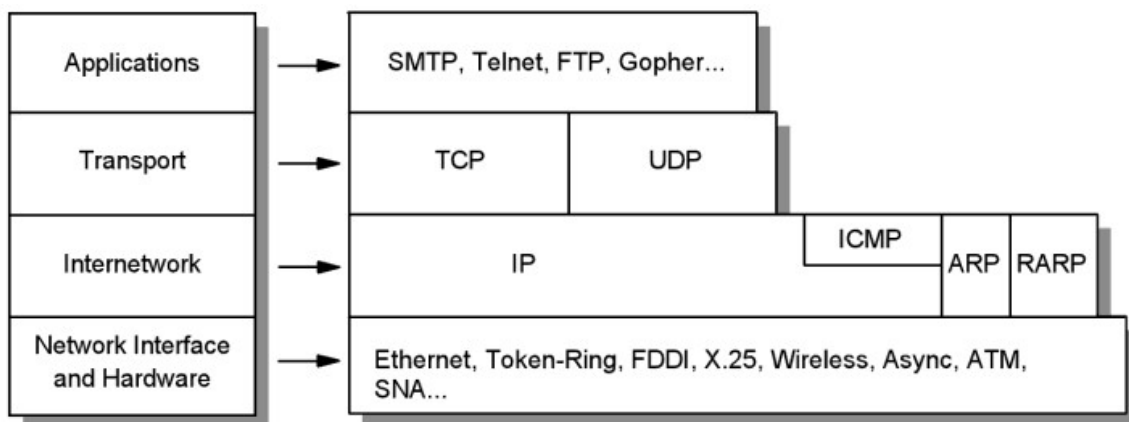


Figura 4.2.2 Arquitectura del modelo de protocolo TCP/IP (Parziale, y otros, 2006)

La comunicación entre RPi principal y RPi secundaria se realiza mediante el protocolo TCP/IP. Esta comunicación puede realizarse tanto por cable Ethernet o de manera inalámbrica. Mediante conexión Ethernet se asegura baja latencia en el envío de paquetes de datos, aunque ambos dispositivos no pueden separarse debido a esta conexión física. En cambio, el método de conexión por Wi-Fi puede acarrear cierto retardo en la comunicación, lo que puede suponer una inestabilidad en el sistema de control, pero elimina las ligaduras

de distancia entre las placas Raspberry Pi. En este trabajo las dos Raspberry Pi están localizadas en el vehículo terrestre, ya que, como se comentaba en el Capítulo 3. la RPi principal controla el propio vehículo y la RPi secundaria posee el módulo de cámara que establece la posición del vehículo. Por tanto, ambas Raspberry Pi están ensambladas a una distancia constante resultando en que la conexión Ethernet se convierta en la opción más apropiada para realizar la comunicación.

Si se desea utilizar una comunicación inalámbrica debido a la naturaleza y arquitectura de otro proyecto, se deben repetir el proceso realizado para comunicar un cliente y servidor dentro de una misma Raspberry Pi. En este caso se debe introducir como servidor la dirección IPv4 que el enrutador haya asignado a la RPi principal y el puerto deseado, por ejemplo, el puerto 8000. No se debe ejecutar el fichero *server.py*, puesto que la RPi principal actúa en como servidor y no es necesario generarlo.

En el caso de la conexión Ethernet, es necesario designar manualmente las IP Ethernet de ambos dispositivos como estáticas, para que no sean modificadas cada vez que son reiniciados, esto se busca cuando se desea utilizar una Raspberry Pi principalmente como servidor. La dirección IP Ethernet de la RPi secundaria se edita, como en el resto del proyecto, mediante la interfaz de *Visual Studio Code*. Se debe añadir la dirección IP estática de la RPi secundaria y una dirección *gateway*, es decir, dirección a través la cual se realiza la conexión deseada al archivo *50-cloud-init.yaml* que se encuentra en el directorio */etc/netplan*. Resulta de suma importancia introducir las líneas de código siguientes sin ningún error o espacio adicional, ya que al tratarse de un archivo principal de conexiones puede acabar corrompido o resultando no ser posible conectarse mediante SSH tras un reinicio. La estructura de dicho fichero es:

```
ethernets:
  eth0:
    dhcp4: no
    addresses: [192.168.0.124/24]
    gateway4: 192.168.0.1
```

```
nameservers:  
  
addresses: [192.168.0.1]
```

De esta manera, se asegura que la dirección IP mediante Ethernet de la RPi secundaria sea 192.168.0.124, en este caso. Además, se debe procurar no seleccionar una dirección IP Ethernet idéntica a la dirección IP WLAN, puesto que puede causar problemas de interpretación por parte de los dispositivos y la conexión entre estos sea denegada cuando se intente enviar o recibir cualquier información.

La RPi principal tiene un sistema operativo distinto a la RPi secundaria, por lo que el proceso de establecer una IP estática no es el mismo. En primer lugar, previo a cualquier modificación, hay que confirmar que el cliente DHCP está funcionando mediante el comando: `sudo service dhcpd status`. En este caso, el archivo a modificar se encuentra en el directorio `/etc` y se denomina `dhcpd.conf`, una implementación de DHCP para el sistema operativo Raspbian. Al final de este fichero de configuración deben introducirse las siguientes líneas:

```
interface eth0  
  
static ip_address=192.168.0.193/24  
  
static routers=192.168.0.1  
  
static domain_name_servers=192.168.0.1
```

Al elegir la segunda dirección IP estática es fundamental que sea distinta a la primera, en este caso se ha seleccionado la dirección 192.168.0.193. Sobremanera, la dirección *gateway*, que hace referencia al router, debe ser idéntica a la introducida en el archivo de configuración de red de la RPi secundaria. Cuando se modifica cualquier archivo fundamental de la configuración o inicio de sistema operativo se debe reiniciar la Raspberry Pi.

Una vez modificados los archivos mencionados, los dispositivos están listos para enviar y recibir información a través de una conexión Ethernet. El esquema final de conexiones y comunicación se muestra en la Figura 4.2.3

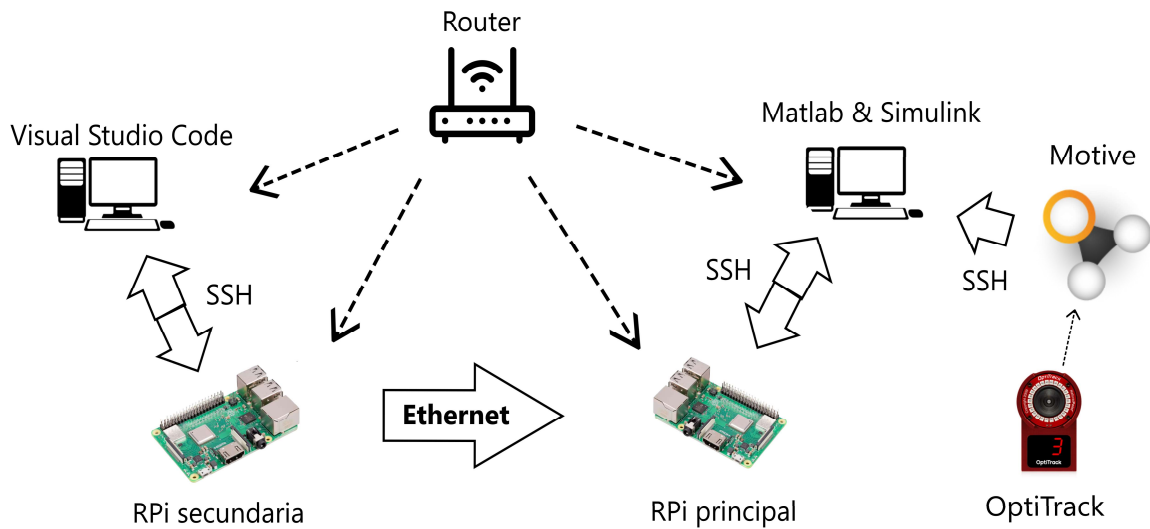


Figura 4.2.3 Estructura de comunicación del proyecto (Fuente: Elaboración propia)

Una vez establecida la red de comunicación entre los dispositivos del proyecto se realiza el envío de información en tiempo real sobre la distancia del vehículo al damero desde la RPi secundaria hacia la RPi principal. Desde el ordenador que está ejecutando Matlab-Simulink puede monitorizarse todos los parámetros que la RPi principal recibe por los distintos puertos de conexión. En este caso, se envían por TCP/IP y el puerto seleccionado es 3000. Matlab deshace la codificación en bytes, representa las seis medidas en Sistema Internacional y almacena los datos para el sistema de navegación, como se muestra en la Figura 4.2.4.

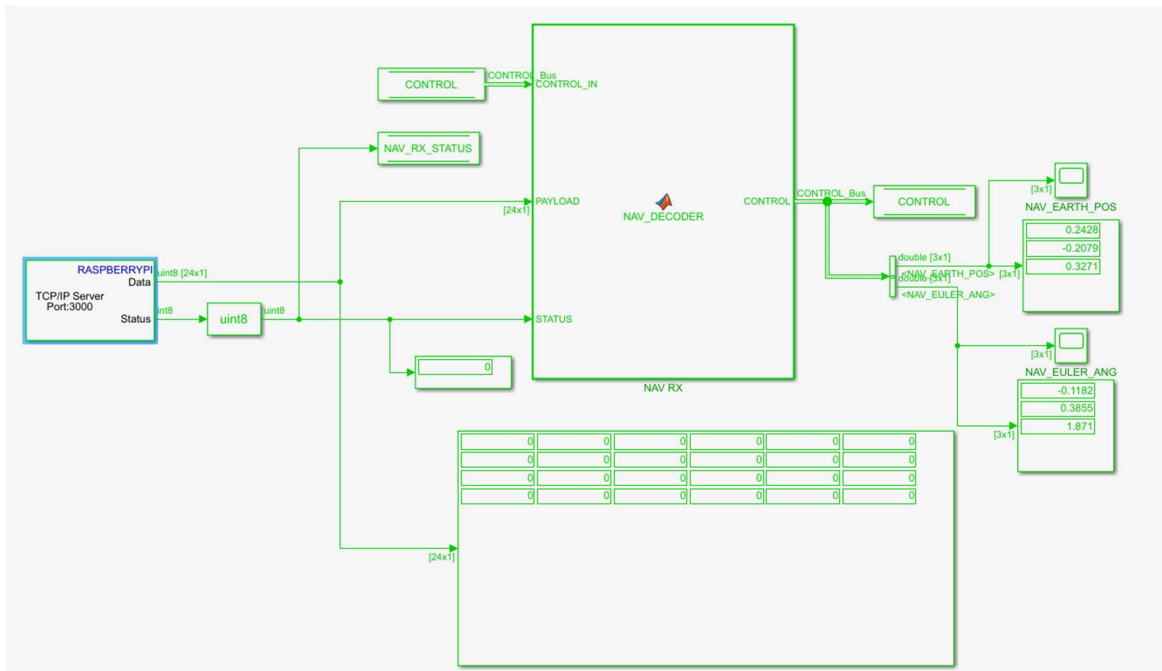


Figura 4.2.4 Diagrama de bloques de comunicación de RPi principal (Fuente: Elaboración propia)

4.3 MOTIVE

Motive es un programa diseñado principalmente para procesar los datos recopilados por el sistema de cámaras OptiTrack proporcionando posiciones en un espacio tridimensional, identificación de marcadores, información sobre rotación y seguimiento de movimiento y morfología humana. El funcionamiento de Motive se basa en la estimación de posición de un objeto o cuerpo, a través de la resolución y conexión espacial de los marcadores esféricos dispuestos en dicho objeto. De esta manera se consigue un resultado preciso con hasta ± 0.2 mm de incertidumbre en medidas posicionales y $\pm 0.1^\circ$ en medidas angulares con un tiempo de obtención menor a 0.1s y latencia de envío de datos menor a 9 ms.

El programa Motive ha producido en su entorno digital una representación a escala del vehículo como un sólido rígido formando una estructura basándose en los marcadores situados en ciertos puntos clave de su chasis (Figura 4.3.1). Los marcadores han sido detectados correctamente por el sistema de cámaras OptiTrack en la zona de ensayos

delimitada en el laboratorio. Por tanto, Motive devuelve unos datos de posicionamiento respecto al centroide del sólido rígido que se ha generado a partir de los marcadores. Una vez obtenido el modelo virtual del vehículo se pueden iniciar los ensayos que resultan en el envío a través de Matlab de las 6 medidas que resultan claves para representar la localización del vehículo: tres magnitudes de posición y tres ángulos de giro, no obstante, como se comenta en la sección 4.1.5, las únicas medidas que interesa analizar en este proyecto son las posiciones del plano horizontal, es decir, el eje X y Z, y el ángulo de giro respecto al eje vertical Y del sistema de coordenadas del vehículo.

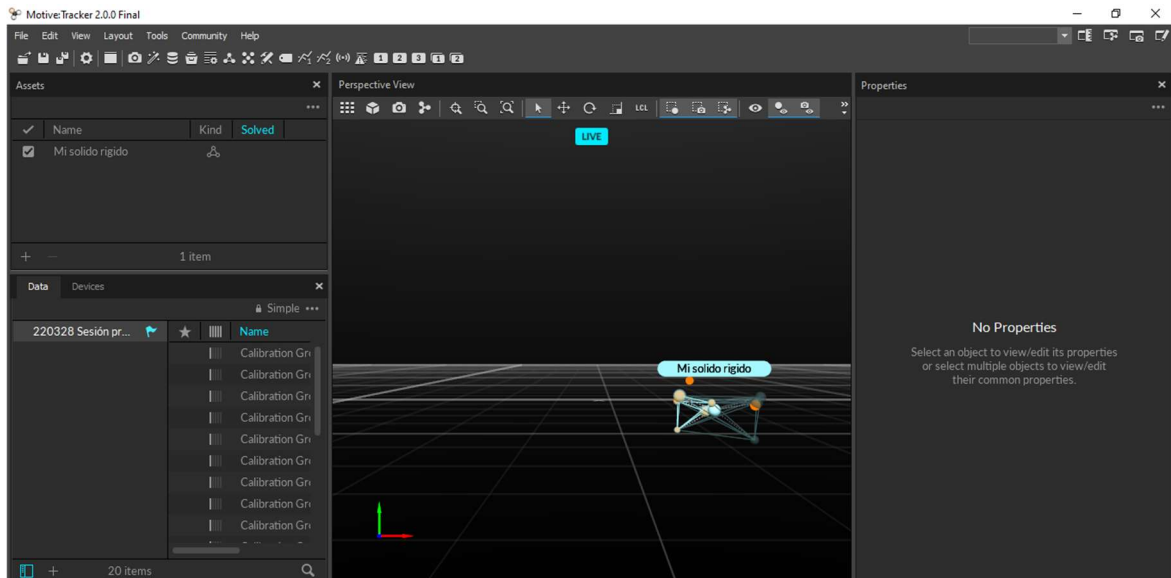


Figura 4.3.1 Sólido rígido generado por Motive (Fuente: Elaboración propia)

Motive cuenta con herramientas nativas para trabajar de forma síncrona con diversos programas como Visual3D, LabVIEW o Matlab, junto con *plugins* complementarios para motores gráficos y entornos de diseño como *Unreal Engine*, *Unity* o *Maya*, entre otros. Para este trabajo es indispensable el uso de la herramienta que comunica Motive con Matlab para procesar directamente los datos de OptiTrack en el entorno de Matlab-Simulink.

Para estudiar en profundidad el funcionamiento de este programa se recomienda revisar la documentación oficial de Motive, debido a la complejidad del calibrado y manejo de las diferentes herramientas disponibles.

Capítulo 5. CARGA INALÁMBRICA

5.1 KIT TDK

El kit de la marca TDK lo conforman dos módulos que se adquieren de forma independiente por lo que no se trata de un kit a lo sumo, pero se denomina de esta manera por conveniencia en la explicación. El módulo de bobina transmisora (WTM905525-12F2-12V-G) y el módulo de bobina receptora (WRM483265-10F5-12V-G) están basados en la tecnología de transferencia de energía inalámbrica Qi . Las aplicaciones recomendadas especialmente para este tipo de bobinas son para productos relacionados con los sectores de la comunicación, industria y medicina, entre otros. Están diseñados para trabajar a un voltaje nominal de 12V y corriente nominal de 1.25 A, resultando en un soporte para transferencia de energía de hasta 15W. La eficiencia de dicha transferencia inalámbrica es del 75% según el fabricante, por lo que se asume que proporcionará buenos resultados en los ensayos de carga.

Estas bobinas traen implementado y programado un microcircuito (IC) manufacturado por *Integrated Device Technology* (IDT), en el caso de la receptora cuenta con un IC P9221-R, mientras que el módulo transmisor contiene el IC P9242-R integrado, ambos diseñado por *Renesas Electronics*¹. Por tanto, estos componentes pueden enchufarse a la red y comenzar a funcionar sin configuración previa, lo que supone un hecho importante en cuanto a comodidad en la aplicación y versatilidad de poder transferir energía entre módulos de modelos y series distintos. Las especificaciones técnicas de estos dispositivos se incluyen en el Anexo B de la presente memoria.

¹ El conjunto bobina e IC es distribuido por TDK, aunque el IC lo manufactura *Renesas Electronics*, una empresa centrada en el diseño y venta de circuitos integrados, como el kit del apartado 5.2.



Figura 5.1.1 Microcircuitos P9221-R (Fuente: Renesas)

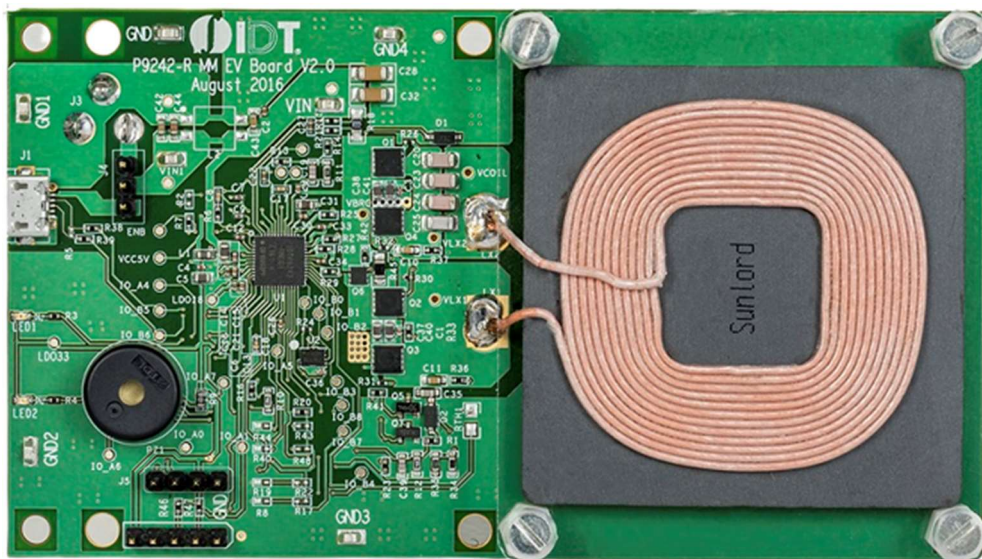


Figura 5.1.2 Microcircuit P9242-R (Fuente: Renesas)

Estos dos componentes se presentan como la propuesta más cómoda y mejor precio en catálogo, aunque sus especificaciones y configuraciones son limitadas y con menos eficientes, respecto al kit de Renesas. No obstante, a priori este kit puede considerarse apto para la aplicación deseada, ya que presenta una potencia de recarga considerable y eficiencia energética aceptable

5.2 KIT RENESAS WP15WBD-RK

El *Renesas WP15WBD-RK Bi-Directional Data Transfer Evaluation Kit* contiene una placa transmisora con el IC P9242-R3 y la placa receptora con el IC P9221-R3. Si se compara con el kit TDK, se observa que son el mismo modelo, pero de una serie superior. Esto se debe a la adición de bidireccionalidad en la comunicación para el estándar de *Renesas*, los dispositivos compatibles con WPC versión 1.2

Presenta hasta 15W de potencia en transferencia de carga junto a un 87% de pico en eficiencia energética, resultando en un producto apropiado para la aplicación del proyecto. La ventaja principal de este kit es el soporte para la transferencia de datos de forma bidireccional, no solamente de transmisor a receptor como es habitual. Además, proporciona soporte para la interfaz de comunicaciones I2C. Es necesario instalar el programa de configuración IDT I2C para actualizar los *drivers* y poder mandar comandos I2C a las placas. Su rango de temperatura de funcionamiento se encuentra entre 0° y 85°, convirtiéndolo en un conjunto de bobinas versátil para distintas aplicaciones como, por ejemplo, equipamiento industrial, circuitos de consumo o equipamiento médico.

En la Figura 5.2.1, se presenta un esquema con la configuración típica de funcionamiento entre placa transmisora y receptora, incluyendo los pines de conexión.

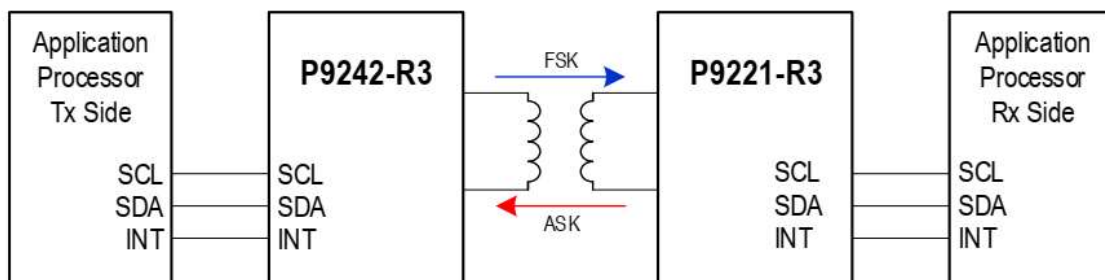


Figura 5.2.1 Diagrama de bloques del kit *Renesas* (Fuente: *Renesas*)

5.3 ELECCIÓN DE KIT DE CARGA

En este apartado se comenta los ensayos realizados con ambos kits de carga mencionados anteriormente. Estos son el kit TDK *Wireless Power Transfer* y el kit *Renesas WP15WBD-RK Bi-directional Data Transfer Evaluation Kit*, en adelante kit TDK y el kit Renesas, respectivamente.

Los parámetros para destacar de la carga inalámbrica son: la corriente en la bobina transmisora, la distancia en el eje horizontal y vertical del receptor y transmisor, y el ángulo relativo entre las bobinas receptora y transmisora. Por tanto, los ensayos que determinen la aptitud de cada kit deben basarse en la medida de estas magnitudes y anotando las restricciones de los grados de libertad (distancia y ángulo relativo). Es importante dimensionar dichos parámetros puesto que el vehículo autónomo debe acercarse a la estación de carga a una distancia concreta para poder realizar el proceso de recarga inalámbrica de la batería y de la precisión de este acercamiento dependen factores como el control seleccionado, el sistema de navegación y la exactitud de medida de los dos sistemas de cámaras, interno y externo. Es decir, es necesario conocer las dimensiones restrictivas de carga de cada kit para poder optimizar los controles y sistema de posicionamiento en base a estos.

En primer lugar, se realizan las comprobaciones pertinentes con el kit TDK. Para ello es necesario el siguiente material, representado en la Figura 5.3.1:

- Pinza amperimétrica, para medir la corriente en la parte del circuito de la bobina receptora. Se utiliza una escala de 100 mV/A.
- Batería auxiliar, para poder trabajar en los ensayos sin necesidad de la batería del vehículo del proyecto.
- Polímetros, para poder establecer el valor del voltaje y corriente de la batería que se va a cargar. Se utiliza un polímetro de 2000 cuentas.
- Cable USB-C y adaptador de corriente, para conectar el módulo de bobina transmisora a la red eléctrica.

Antes de poder realizar estos ensayos se debe soldar al módulo de bobina en el pin 1 de entrada Vchar IN y en el pin 2 de salida GND el cableado con conector compatible para las baterías utilizadas en el proyecto. Además, el cable de salida (negro) contará con una derivación (azul) para conectarse posteriormente al sensor de corriente instalado en el vehículo. El pin número 3 WC_EN queda libre, ya que se trata de la conexión para la unidad de microcontrolador y no ha resultado necesaria para el correcto funcionamiento en la aplicación del proyecto.

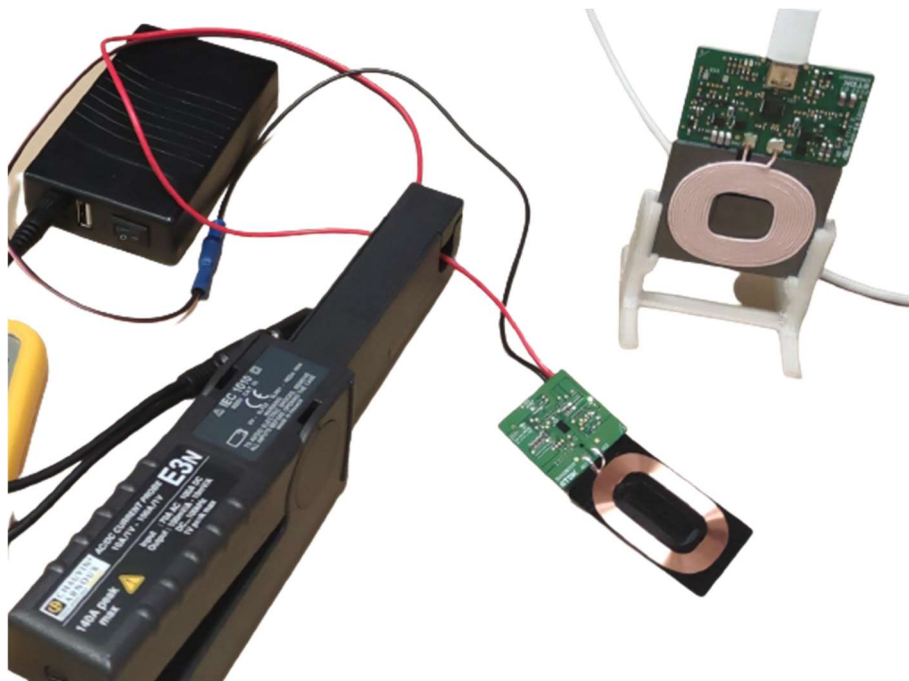


Figura 5.3.1 Ensayo de carga kit TDK (Fuente: Elaboración propia)

Haciendo uso de los soportes para los módulos de bobinas y útiles de medida, como un papel milimetrado, y un medidor de ángulos se proceden a realizar los tres ensayos principales.

1) Distancia frontal

El primer ensayo trata de alejar progresivamente la placa receptora de la bobina transmisora, permaneciendo esta última estacionaria, para emular el alejamiento y acoplamiento del vehículo con el receptor implementado en la parte trasera hacia la estación de carga. Este movimiento se realizará de forma perfectamente paralela entre los dos

módulos de carga, es decir, con un ángulo nulo de giro relativo entre las dos bobinas, y una distancia lateral nula, significando que los centros de ambas bobinas están alineados de forma perpendicular a sus superficies. Esto se debe a la propia naturaleza de la inducción electromagnética, que necesita tener la mayor superficie y menor ángulo relativo para tener mayor efectividad.

Para realizar la carga con este modelo de placas se deben acercar ambos módulos a una distancia dentro del rango de funcionamiento, confirmar que se enciende el LED verde del receptor y esperar a que finalice la alternancia de encendido y apagado entre los LED verde y rojo de la bobina transmisora. En la Tabla 5.3.1 se presentan los resultados de detección en distancia frontal. Como puede observarse en la primera columna, la distancia mínima que puede acercarse el vehículo es 5mm debido al espesor de los soportes. Tras determinar el espacio más cercano de carga, se comienza a alejar en intervalos de 1mm para analizar y registrar los datos sobre corriente. Finalmente, y tras varios intentos fallidos para superarlo, la distancia frontal máxima de carga entre las bobinas es de 10mm. Puede apreciarse el final del rango de carga cuando el LED rojo del módulo transmisor comienza a parpadear, avisando que la carga se ha detenido. No puede volverse a realizar la carga inalámbrica hasta que este LED se apague por completo, por tanto, se debe alejar la bobina receptora completamente del rango de trabajo de ambos dispositivos. Dentro de este rango, se aprecia como la corriente de carga se ha mantenido cercana al valor nominal de 1.25 A, resultando en un aumento del voltaje de la batería según avanza el ensayo.

DISTANCIA FRONTAL (mm)	Vo (V)	Vcarga (V)	ΔV (V)	Irms (A)
5	10,84	11,3	0,46	1,06
6	11,05	11,43	0,38	1,09
7	11,08	11,48	0,4	1,13
8	11,14	11,52	0,38	1,13
9	11,09	11,48	0,39	1,22
10	11,1	11,52	0,42	1,22

Tabla 5.3.1 Ensayo de distancia frontal (Vo: voltaje de la batería inicial, Vcarga: voltaje de la batería cargando, ΔV : Vcarga – Vo, Irms: corriente eficaz en el secundario)

2) Ángulo relativo

Este ensayo analiza el impacto que tiene la variación de ángulo relativa entre las placas receptora y transmisora en la cantidad de corriente proporcionada a la batería. Se procede a variar dicho ángulo girando el módulo receptor alrededor del eje vertical estando posicionado lo más cerca posible de la base transmisora para que la distancia frontal no suponga un parámetro clave ni tenga influencia en este ensayo. Partiendo de la misma posición que el ensayo anterior se anotan las variaciones aplicadas en cada momento. El LED rojo comienza a parpadear cuando la rotación alcanza aproximadamente los 12°, por lo que se considerará dicho resultado como la rotación máxima del vehículo en el momento de realizar la recarga de batería.

Se puede observar en la última columna de la Tabla 5.3.2 que la corriente medida presenta un rango de valores estrecho, por lo que, según este ensayo se concluye que el ángulo no tiene un impacto considerable dentro del margen de funcionamiento como se podía prever dado la naturaleza del fenómeno de la inducción mutua.

ANGULO (º)	Vo (V)	Vcarga (V)	ΔV (V)	Irms (A)
3	11,18	11,58	0,4	1,24
5	11,18	11,58	0,4	1,23
8	11,2	11,6	0,4	1,23
10	11,2	11,61	0,41	1,19
12	11,23	11,64	0,41	1,2

Tabla 5.3.2 Ensayo de ángulo relativo (Vo: voltaje de la batería inicial, Vcarga: voltaje de la batería cargando, ΔV: Vcarga – Vo, Irms: corriente eficaz en el secundario)

3) Distancia lateral

La distancia lateral es el último parámetro a tener en cuenta en los ensayos de este proyecto. La manera óptima de realizar una carga inalámbrica de un dispositivo es que ambas bobinas estén dispuestas con sus centros geométricos alineados maximizando el área común. Es por esto que ser exacto con este parámetro se convierte en un aspecto crucial de la navegación.

El ensayo procede desplazando unos milímetros a la derecha el módulo receptor para comprobar el impacto sobre la carga. No se perciben cambios hasta que se ha incrementado la distancia lateral hasta 4mm, considerando como inicio el centro de las bobinas. Este parámetro es el más restrictivo en cuanto a rango puesto que solo se tienen 7mm desde que se comienza a desplazar. Asimismo, hay una caída de 60mA entre el valor máximo y el mínimo registrado en cada caso, aunque tampoco debería suponer una diferencia importante al cargar la batería. Los datos experimentales de este ensayo se incluyen en la Tabla 5.3.3.

DISTANCIA LATERAL (mm)	Vo (V)	Vcarga (V)	ΔV (V)	Irms (A)
4	11,18	11,59	0,41	1,22
5	11,24	11,64	0,4	1,23
6	11,26	11,65	0,39	1,17
7	11,27	11,67	0,4	1,19

Tabla 5.3.3 Ensayo de distancia lateral (Vo: voltaje de la batería inicial, Vcarga: voltaje de la batería cargando, ΔV : Vcarga – Vo, Irms: corriente eficaz en el secundario)

Por otro lado, se ha intentado realizar estos ensayos de manera idéntica con el kit Renesas, pero ha resultado imposible sacar ninguna conclusión sobre su desempeño puesto que no se ha conseguido establecer el conexionado adecuado para su funcionamiento. No obstante, todo el progreso y modificaciones realizadas en este proyecto a este kit está disponible para futuros trabajos.

En conclusión, debido a que no pudieron llevar a cabo las experiencias con el kit Renesas, y los resultados obtenidos en los ensayos del kit TDK se ha optado por su implementación directa como base de carga para el vehículo.

5.4 MEDIDOR DE CORRIENTE

La placa de medida de corriente eléctrica *Current Click*, diseñado por la empresa serbia MikroElektronika, está conformada por un monitor de corriente *shunt* INA196, un convertidor ADC de 12 bit MCP3201 y MAX6106 como voltaje de referencia. Consta de dos bornes roscados, para la corriente de entrada y salida, y para la conexión de una

resistencia *shunt*. El rango de medida de corriente DC es de 2mA hasta 2048mA, variando según la resistencia *shunt* conectada en el terminal roscado. En este caso se ha optado por una resistencia *shunt* de 100 Ω para poder trabajar en el rango de corriente especificado por el fabricante de las bobinas y el sensor.

Está diseñado para trabajar a 3.3V o 5V y puede comunicarse con un microcontrolador a través de una interfaz SPI utilizando el convertidor ADC o mediante conexión directa al pin AN del microcontrolador. La configuración de trabajo depende del estado del *jumper*, o selector, denominado J1 SMD, en el diseño de la placa. Se encuentra soldado en la posición de tensión 3.3V y conexión ADC por defecto, por lo que se trabaja con este componente de esta manera.

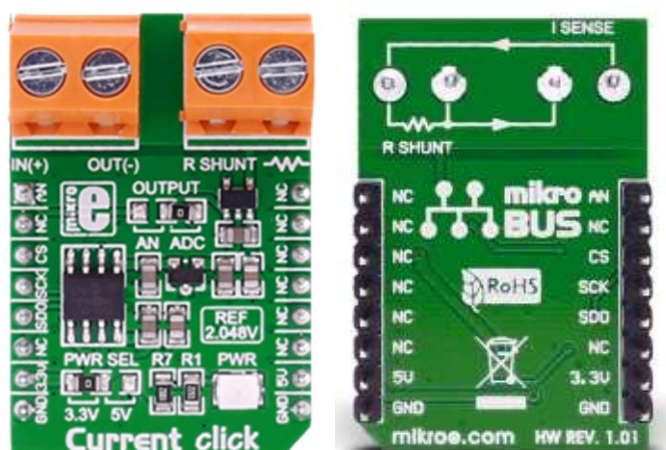


Figura 5.4.1 Medidor de corriente *Current Click* (MikroElectronika, 2017)

Mediante esta placa se tienen medidas de la corriente que circula desde el receptor del sistema de carga inalámbrica hasta la batería del vehículo terrestre y, por tanto, se podrá conocer si se está realizando la recarga de la batería y la cantidad de corriente que se recibe.

Para ello se ha desarrollado un esquema en Simulink (Figura 5.4.2) para obtener la información convertida por el ADC del sensor para ser procesado a través de Matlab. Para conseguirlo es necesario conectar el sensor al adaptador de la placa de extensión Pi 2 Click situado en la parte frontal del vehículo, a su vez conectado al pin GPIO 8 de la RPi principal que tiene la funcionalidad de interfaz serie periférica de comunicación. Este diagrama de

bloques permite enviar a la RPi principal la cantidad de corriente en tiempo real que pasa por la batería al recargar para poder conocer el estado del vehículo y mandar instrucciones al sistema de navegación para retomar la rutina o continuar en la base de carga.

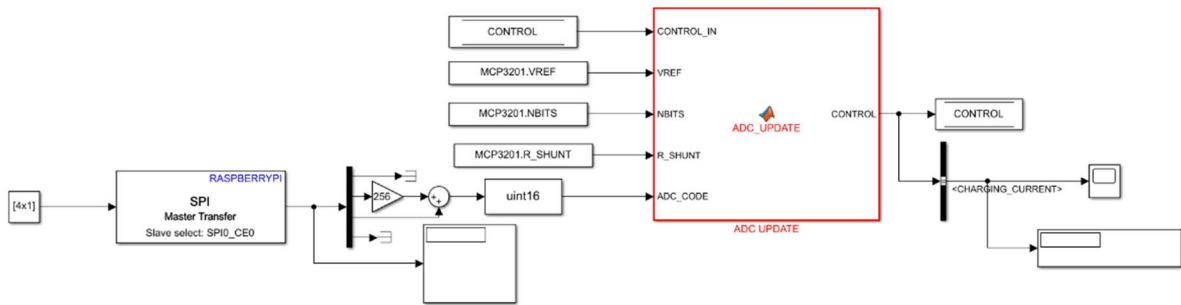


Figura 5.4.2 Diagrama de bloques del sensor de corriente en Simulink (Fuente: Elaboración propia)

Capítulo 6. SISTEMA DE CONTROL Y NAVEGACIÓN

6.1 DESCRIPCIÓN DEL SISTEMA DE CONTROL

El sistema de control y navegación aplicado en este trabajo es un control de la velocidad y orientación del vehículo para realizar el recorrido necesario para completar el acoplamiento a la estación de recarga basándose en un *waypoint* o punto de referencia, el damero en este caso. Dicho sistema se basa en un control en cascada con dos lazos de control, interno y externo (Figura 6.1.1).

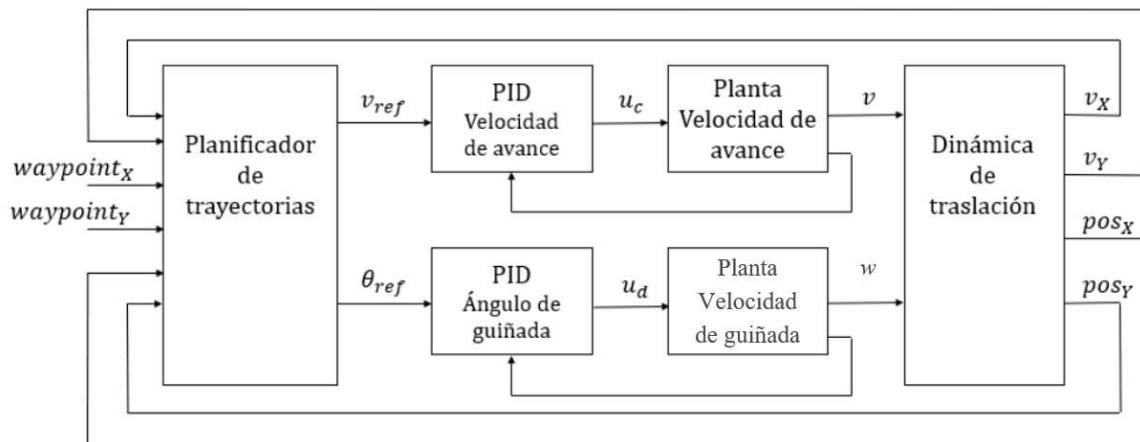


Figura 6.1.1 Diagrama del lazo de control del vehículo (Fuente: Departamento de Electrónica)

El lazo de control externo mantiene como referencia los parámetros de mando de velocidad v_{ref} y ángulo de guiñada θ_{ref} . Las variables de posición X e Y, las variables de velocidad lineal v_x y v_y , y los ejes de los *waypoint* intervienen controlando el lazo. En cambio, el lazo de control se refiere a los controles PID para la velocidad de avance y el ángulo de guiñada. Este lazo controla las variables de velocidad lineal v y velocidad de giro del vehículo w para gobernar el funcionamiento de los dos motores DC, tanto la tensión común y diferencial. A través de la velocidad de avance y la velocidad de guiñada en el lazo

se obtiene la descomposición vectorial del movimiento del vehículo en el plano horizontal v_x y v_y , y la orientación del vehículo θ .

Las plantas y los controles utilizados para dirigir el vehículo fueron caracterizadas y diseñados en trabajos anteriores, respectivamente. Puede utilizarse un control proporcional (P), un control proporcional integral (PI), control proporcional derivativo (PD) o un control proporcional integral derivativo (PID). En este caso, se ha decidido la implementación de un control PID, debido al rápido seguimiento de la referencia sin error, para la velocidad de avance, la velocidad de guiñada y un control P para el ángulo de giro, ya que la planta tiene un polo en el origen. El tipo de control y parámetros deseados se puede seleccionar en el fichero CONFIG_CONTROL.m en Matlab.

6.1.1 CONTROLES DE VELOCIDAD DE AVANCE Y GUIÑADA

Los dos controles tienen como objetivo común seguir la referencia establecida de velocidad de movimiento lineal y giro por el usuario mediante una emisora de radiofrecuencia manualmente o por el sistema de navegación. El control PID se ha basado en el diseño por margen de fase para respuesta en frecuencia. Los parámetros de diseño de los controles PID seleccionados se muestran en la Tabla 6.1.1.

Parámetro	Avance	Guiñada
Margen de fase (φ_m)	65°	60°
$\frac{w_o}{w_o P}$	1.5	1.2
Retraso de fase integral (φ_{PI})	-10°	-10°
Factor de filtrado (f)	0.1	0.1
Ponderación a la referencia (b)	1	1

Tabla 6.1.1 Parámetros de diseño control PID de velocidades (Fuente: Elaboración propia)

6.1.2 CONTROL DE ÁNGULO DE GIRO

Este control debe corregir el ángulo de giro del vehículo cuando se vea modificado volviendo a la orientación establecida inicialmente. Al igual que los controles de velocidad

se ha procedido a utilizar un control P diseñado por margen de fase. Los parámetros de diseño de los controles se muestran en la Tabla 6.1.2

Parámetro	Ángulo de giro
Margen de fase (φ_m)	60°
$\frac{w_o}{w_o P}$	1
Retraso de fase integral (φ_{PI})	-
Factor de filtrado (f)	-
Ponderación a la referencia (b)	-

Tabla 6.1.2 Parámetros de diseño control PID de ángulo de giro (Fuente: Elaboración propia)

6.2 CONTROL DE VEHÍCULO MEDIANTE SIMULINK

El fichero de Simulink CAR_CONTROL_SYSTEM.slx se encarga de ejecutar todas las instrucciones y comunicaciones del vehículo y permite monitorizar toda esta información. Este fichero se divide tres buses de almacenamiento de información y cuatro secciones de trabajo (Figura 6.2.1).

Los buses se denominan CONTROL, MSG y MODEL y sus funciones principales son:

- CONTROL: se encarga de recopilar y almacenar todas las variables relacionadas con el control del vehículo, como los mandos, entradas, medidas, estimaciones, salidas, etc.
- MSG: contiene todas las variables relacionadas con las comunicaciones establecidas por el vehículo y el resto de los dispositivos del proyecto.
- MODEL: se encarga de guardar las variables de modelado del vehículo, actuadores y sensores implementados.

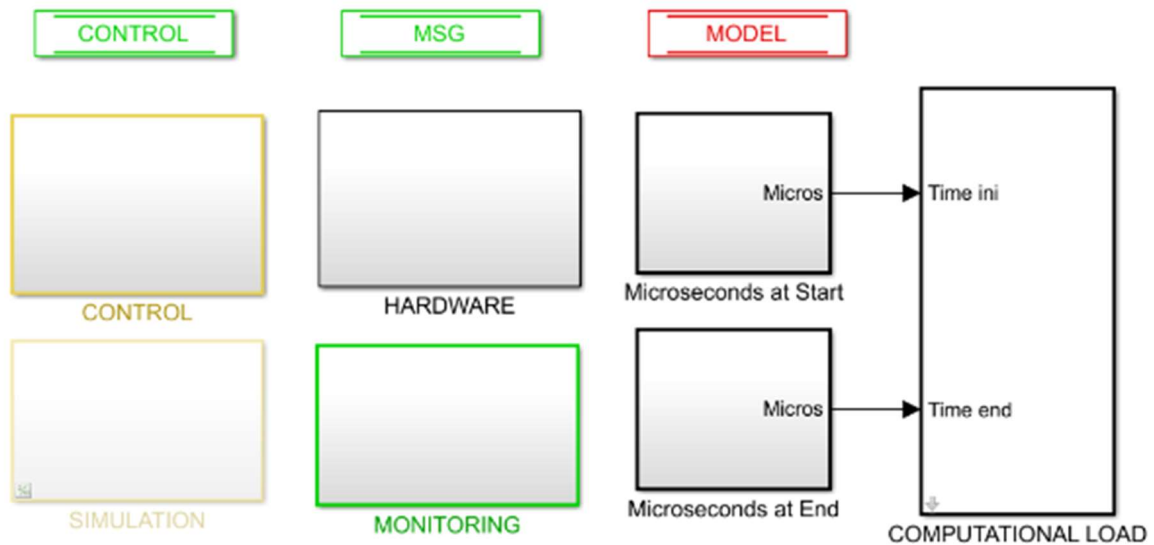


Figura 6.2.1 Diagrama de CAR_CONTROL_SYSTEM.slx (Fuente: Elaboración propia)

Por otro lado, los cuatro subsistemas se ejecutan mediante las variables de los buses sus funciones establecidas en Simulink.

Control: Todos los bloques están relacionados con los aspectos del sistema de control del vehículo, tanto la máquina de estados como los estimadores de estado y actualizaciones del control para calcular los valores de las salidas del sistema (

- Figura 6.2.2).
- Simulation: Los bloques de este subsistema (Figura 6.2.3) contienen los modelos del vehículo y actuadores para poder realizar simulaciones dentro de Simulink sin necesidad de utilizar el vehículo físicamente. Además, establece el funcionamiento de los botones de marcha de la Figura 7.1.2.
- Hardware: En la Figura 6.2.4 se muestra los 3 bloques (sensores, actuadores y comunicaciones) en los que se divide este apartado del fichero. Se almacenan todas las instrucciones referentes a los componentes y los protocolos de comunicación del vehículo. En este subsistema se encuentra el diagrama del sensor de corriente de la Figura 5.4.2 y los bloques de comunicación TCP/IP de la Figura 4.2.4.

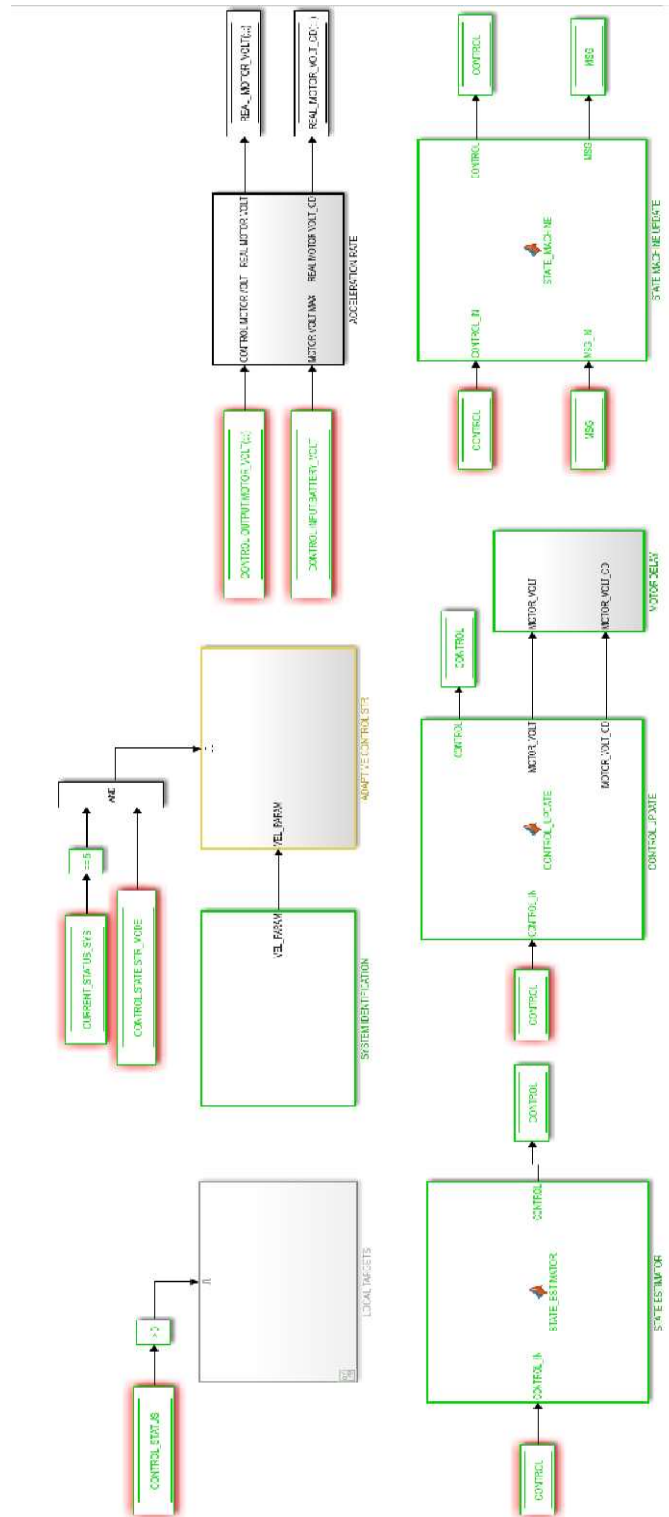


Figura 6.2.2 Subsistema Control (Fuente: Elaboración propia)

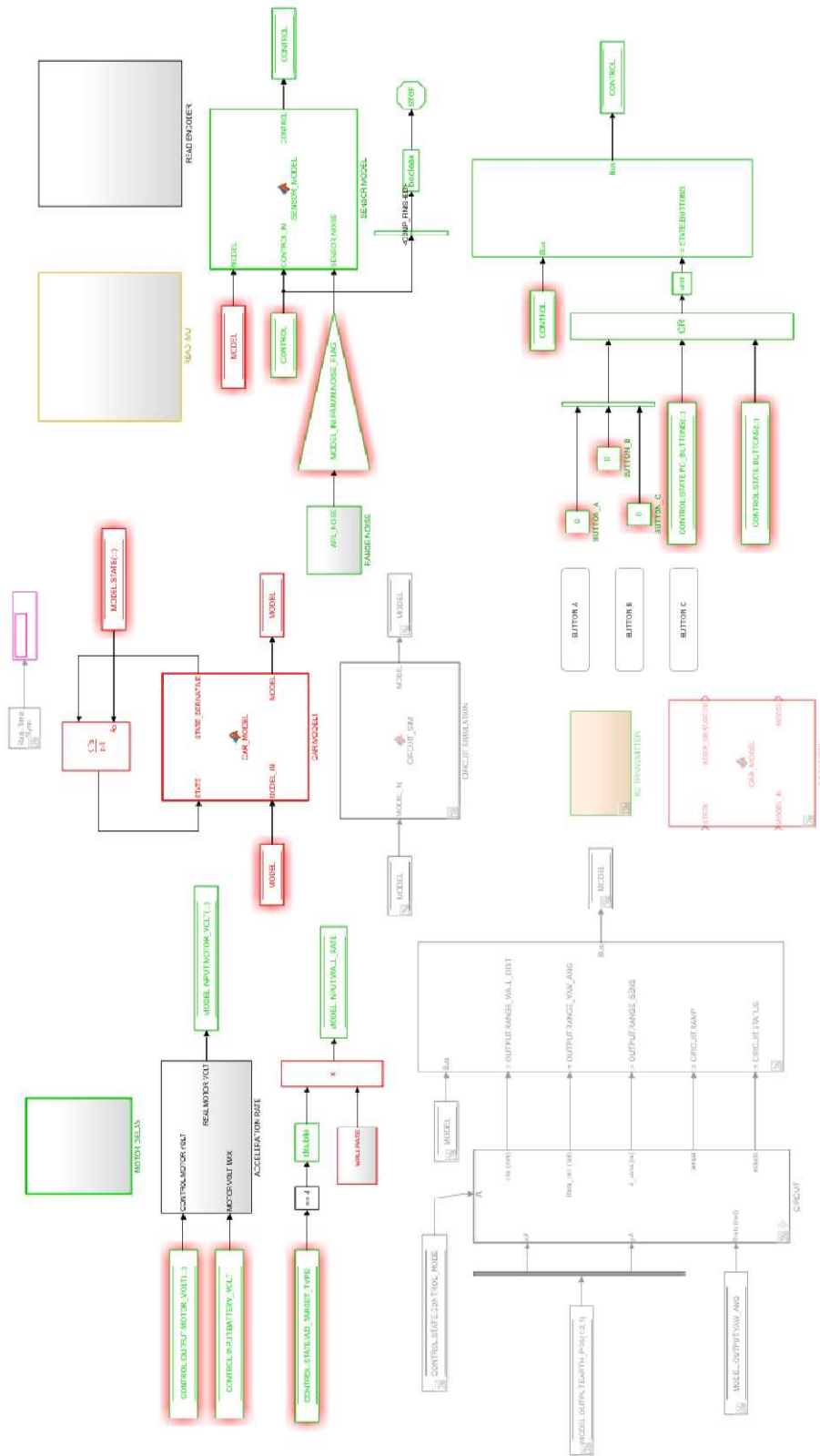


Figura 6.2.3 Subsistema Simulation (Fuente: Elaboración propia)



Figura 6.2.4 Subsistema Hardware (Fuente: Elaboración propia)

- Monitoring: este subsistema (Figura 6.2.5) contiene todos los osciloscopios o *scopes* necesarios para monitorizar cualquier variable del vehículo, por ejemplo, tensión de los motores o la velocidad de avance, entre otros.

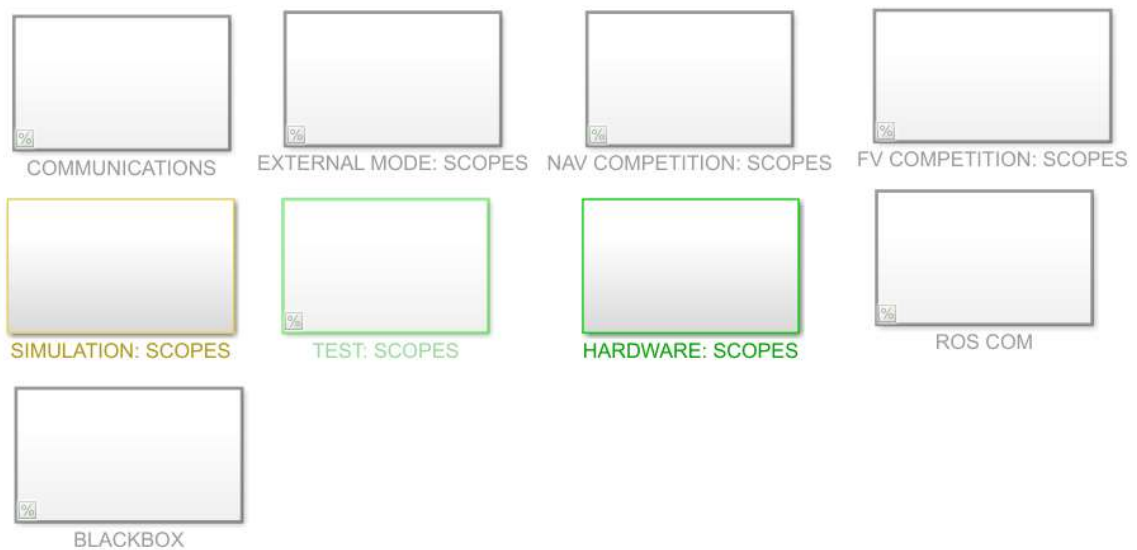


Figura 6.2.5 Subsistema Monitoring (Fuente: Elaboración propia)

Capítulo 7. ENSAYOS Y RESULTADOS

7.1 ENTORNO DE TRABAJO

En la sección 3.4 se comentó brevemente el entorno de trabajo donde se localiza el sistema de cámaras externo y las dimensiones de este espacio. Se debe conocer con exactitud las direcciones de los ejes del sistema de referencia tanto del sistema de cámaras externo OptiTrack y Motive como del módulo de cámara con OpenCV. En este caso, el eje Y de la cámara interna es equivalente con signo positivo al eje Z determinado por Motive, y viceversa. En cambio, los ejes X de cada sistema de referencia coinciden, pero con sentido opuesto. En la Figura 7.1.1 se muestra el entorno de trabajo donde se realizan los ensayos junto con la representación del sistema de coordenadas de Motive que se determina como absoluto en este proyecto.

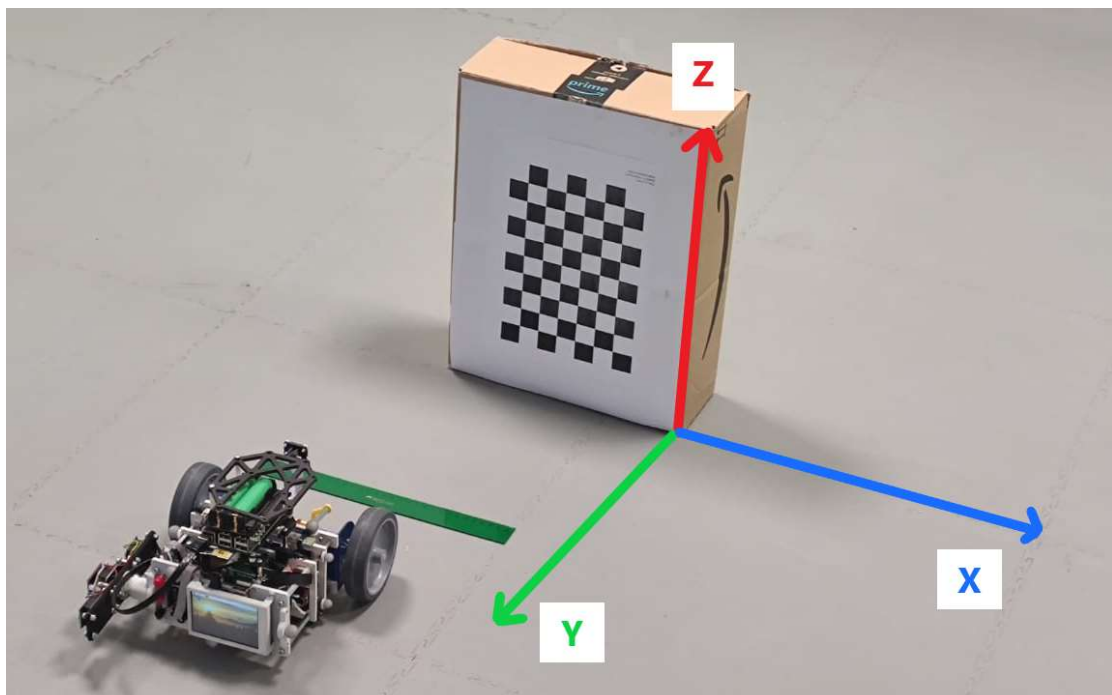


Figura 7.1.1 Entorno de trabajo con sistema de referencia (Fuente: Elaboración propia)

El damero se sitúa a 30cm de la estación de carga para que la cámara de la RPi secundaria pueda tener una visión completa del mismo al realizar el acoplamiento, puesto que si el área del damero visto fuese parcial no lo detectaría, siendo incapaz de determinar su posición. La disposición de botones y LED de marcha se presentan en la Figura 7.1.2. El procedimiento de los ensayos consta de los siguientes pasos:

1. Ejecutar el fichero de comunicación del vehículo desde Matlab para establecer conexión entre los dispositivos.
2. Comprobar que se los sistemas de posicionamiento están midiendo la referencia correctamente antes de establecer el origen de coordenadas.
3. Pulsar el botón negro del vehículo para establecer la posición actual como origen de coordenadas e igualar las medidas de Motive y OpenCV.
4. Esperar hasta que se ilumine el LED verde y pulsar el botón negro de nuevo para comenzar a recopilar información sobre la posición y activar el movimiento del vehículo.
5. El vehículo se detiene al pulsar el botón intermedio rojo.



Figura 7.1.2 Disposición de botones y LED de marcha del vehículo (Fuente: Elaboración propia)

Los ensayos que se han determinado como claves para recopilar información sobre el uso de esta tecnología y su implementación son el ensayo de acoplamiento a la base de carga y el ensayo de corrección de ángulo de giro.

7.2 ENSAYO DE ACOPLAMIENTO A BASE DE CARGA

El ensayo de acoplamiento busca establecer la precisión de las medidas calculadas por el algoritmo de OpenCV respecto a las medidas del sistema de cámaras OptiTrack, tomando como referencia las últimas. Este ensayo se divide en la caracterización de las medidas para el eje X y el eje Y de manera independiente, es decir, en cada ensayo se modifica el valor de un solo eje.

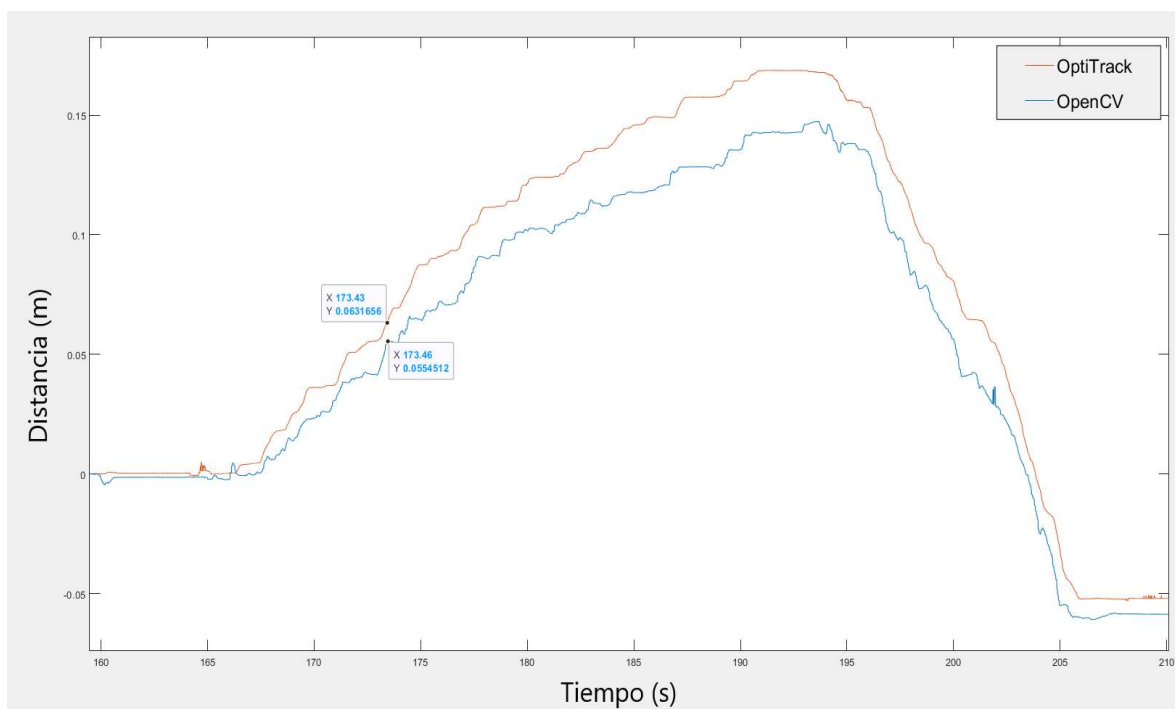


Figura 7.2.1 Ensayo de acoplamiento en el eje X

El primer ensayo que se realiza es un movimiento a lo largo del eje X, es decir, un movimiento lateral. La caracterización de esta coordenada se realiza a 30cm de distancia del damero respecto al eje Y. Se observa cómo ambos sistemas de localización detectan correctamente y al unísono el recorrido que realiza el vehículo sobre el eje X (Figura 7.2.1). Sin embargo, es evidente que la gráfica de OpenCV refleja siempre un rango entre 1-2cm de diferencia menor que las cámaras OptiTrack. Esto puede corregirse añadiendo dicho desvío en el fichero de Python. Aunque se detecte esta diferencia, se destaca que la forma de las gráficas es cuasi idéntica, con algunos picos de ruido menores, pero con un buen resultado a corta distancia.

Por otro lado, se debe comprobar el desempeño de la cámara cuando se realiza una ruta recta con una velocidad constante a través del eje Y, ya que el problema principal que reside en este sistema de posicionamiento es el hecho de perder de vista el damero. Se han estudiado dos situaciones posibles. Realizar el alejamiento y acoplamiento de la estación de carga sin sobrepasar demasiado el límite de visión de la cámara y otro ensayo en el que se deja de detectar el tablero durante un intervalo mayor de tiempo y distancia.

En el ensayo de acoplamiento cercano se busca establecer el límite de visión de la cámara con OpenCV para conocer cuánto puede alejarse el vehículo sin perder noción de su posición correcta. Los resultados se muestran en la Figura 7.2.2, y se observa como al alejarse más de 90cm el programa de OpenCV comienza a recopilar distancias demasiado alejadas de la realidad, como se puede comparar con la gráfica de OptiTrack. También existen escalones para medidas cercanas a 1.5m cuando en realidad no se había superado 1m de distancia. Esto se debe a que cuando el vehículo se encuentra en movimiento es inevitable que la imagen de la cámara se desenfoque al no tener una lente, por lo que en ocasiones detecta el tablero, pero no consigue determinar correctamente la posición. No obstante, al realizar el recorrido de regreso y atraviesa de nuevo la línea de los 90 cm retoma correctamente las medidas de posición y determinando con una exactitud notable el recorrido del vehículo.

En cambio, el ensayo de acoplamiento lejano pretende poner a prueba al algoritmo de OpenCV tras cierto periodo sin detectar el tablero. En el caso del ensayo de acoplamiento cercano, el programa tenía serios problemas para determinar la posición cuando se encuentra en el rango límite de operación. Observando la Figura 7.2.3, se corrobora el límite de operación del programa en 90cm y como, una vez sobrepasado este límite, el algoritmo no puede devolver más información, resultando en una gráfica que refleja que el vehículo no está en movimiento cuando la realidad representada en la gráfica de OptiTrack es que el vehículo llega al doble de distancia. Por tanto, el vehículo no puede alejarse más de 90 cm

del damero sin resultar en una pérdida del posicionamiento para el sistema de navegación.

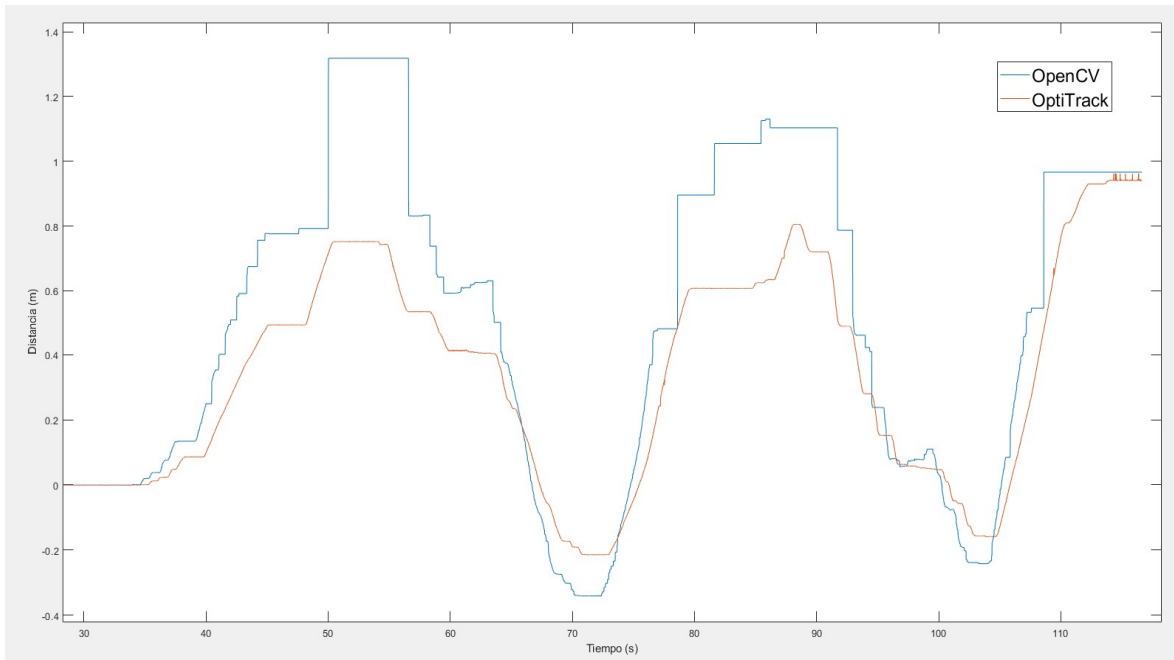


Figura 7.2.2 Ensayo de acoplamiento cercano en el eje Y

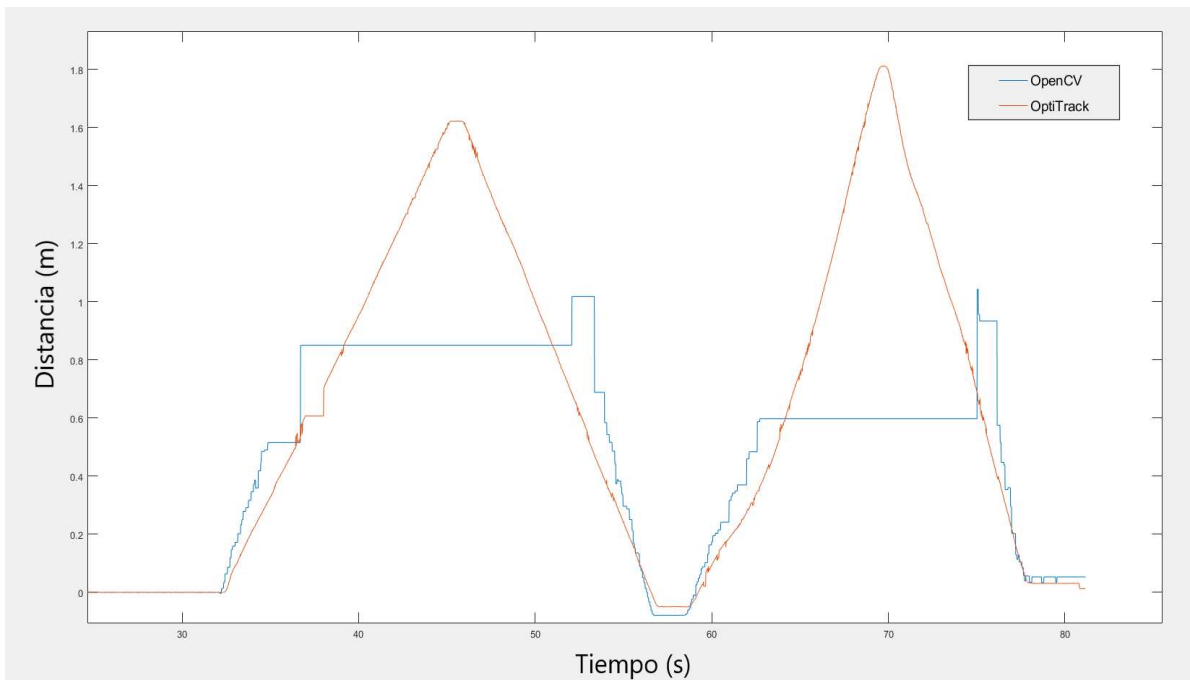


Figura 7.2.3 Ensayo de acoplamiento lejano en el eje Y

7.3 ENSAYO DE CORRECCIÓN DE ÁNGULO DE GIRO

Este ensayo se plantea introduciendo perturbaciones al ángulo de guiñada como referencia y el propósito del control es seguir dicha referencia buscando volver siempre al ángulo de giro nulo. Como se comentó en la sección 5.3, el límite de ángulo de guiñada permitido para conseguir un acoplamiento a la base de carga es de 12° por lo que se busca conseguir como resultado un ángulo del control menor para que el diseño sea confiable.

En este ensayo el control solo obtendrá información sobre el ángulo de giro del módulo de cámara de la RPi secundaria, ya que OptiTrack solamente se estaba utilizando como referencia. Para obtener un resultado óptimo, este ensayo se ha realizado con la cámara apuntando hacia el techo y el damero a menos de 1m de altura respecto al vehículo. En la Figura 7.3.1 se representa la respuesta del control P, definido en la sección 6.1.2, ante una referencia que obliga al vehículo a variar su ángulo de guiñada de manera considerable, superando en ocasiones el cambio de sentido por el giro.

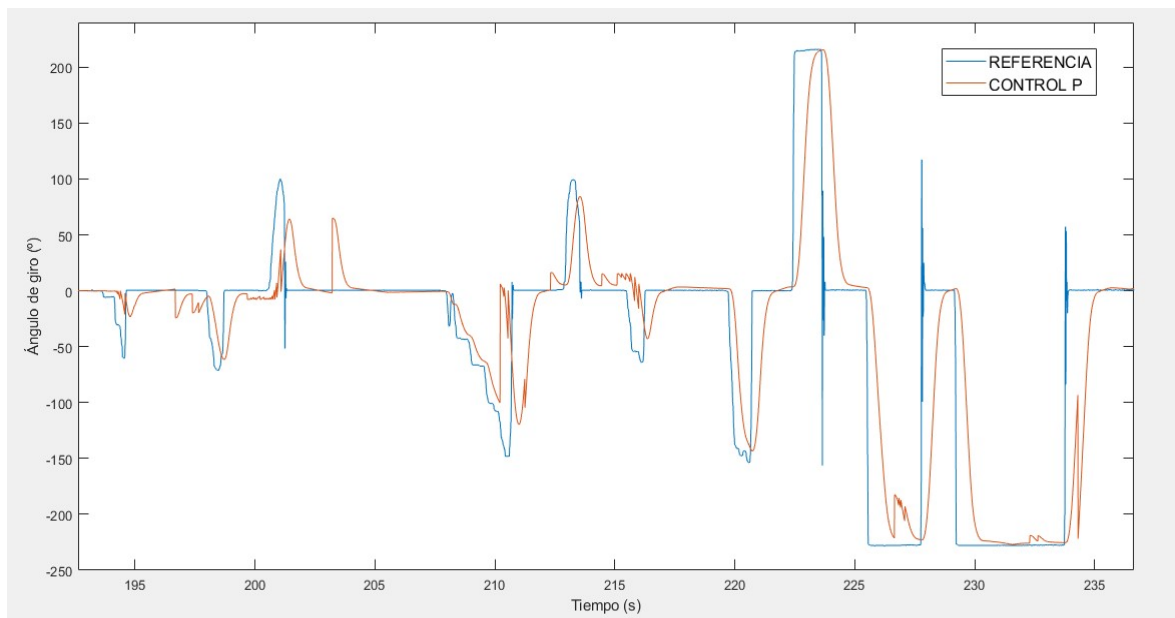


Figura 7.3.1 Respuesta del control P para corrección de ángulo de giro

El seguimiento de la referencia es correcto, con una respuesta de menos de 10ms a los escalones de referencia, poco ruido y aparentemente con un error nulo en régimen

estacionario. Observando en detalle el intervalo de 215s a 220s (Figura 7.3.2), que amplía la gráfica anterior para mostrar el error en el seguimiento. El error en este caso es de 3.3° de variación respecto al ángulo inicial, por lo que se encuentra dentro del rango deseado, y se alcanza este estado en un intervalo de tiempo óptimo. Por tanto, el control proporcional diseñado ha resultado ser robusto, rápido y con un error correcto, incluso con grandes variaciones en el ángulo de guiñada, por lo que se podría utilizar en un ensayo de movimiento libre en futuros trabajos.

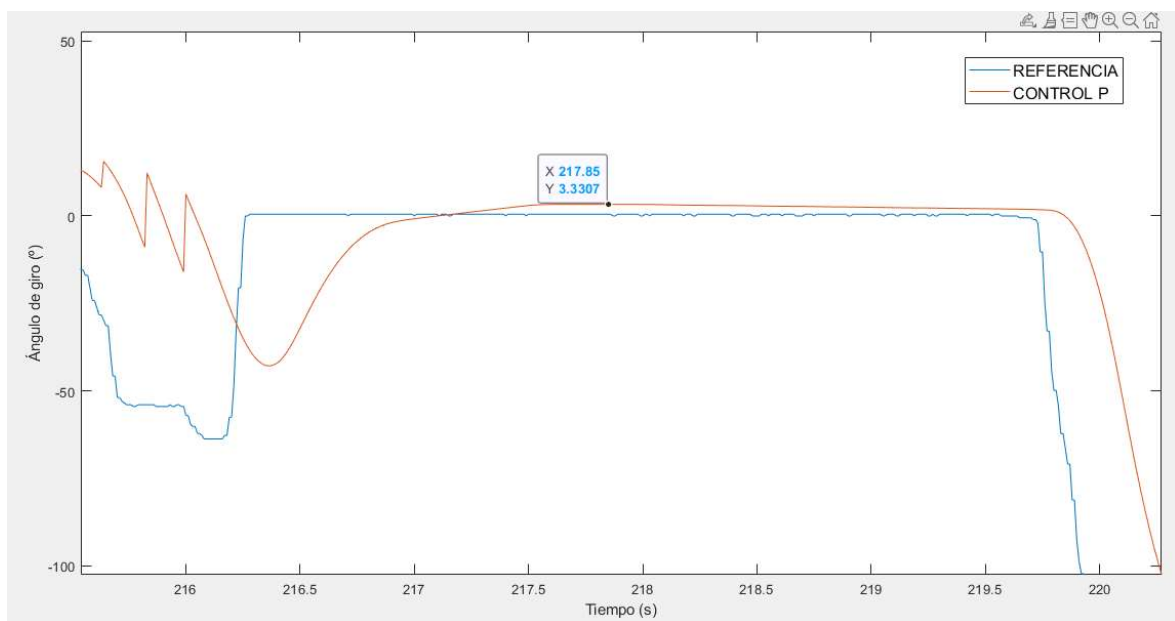


Figura 7.3.2 Detalle en respuesta del control P para corrección de ángulo de giro

Capítulo 8. CONCLUSIONES Y FUTUROS PROYECTOS

8.1 CONCLUSIONES

Las conclusiones obtenidas tras el desarrollo del trabajo fin de grado y la elaboración de la memoria son las siguientes:

La detección del damero ha resultado insuficiente en distancias largas

El desempeño del módulo de cámara para detectar el damero ha resultado inferior al esperado antes de la consecución del trabajo. Al compararlo con el sistema de cámaras OptiTrack la medida recopilada de distancias menores a 1m es correcta, aunque es necesaria más precisión para poder realizar el acople a la estación de carga con el margen de 10mm comentado en el apartado 5.3. Se ha comprobado que la calibración y el funcionamiento general es correcto, por lo que simplemente se trata de una limitación propia de la combinación de esta cámara y damero. Sin embargo, la recopilación de distancias y su precisión pueden considerarse correctas como punto de partida para futuros proyectos.

Cargar inalámbricamente es lento y costoso energéticamente

La tecnología de carga inalámbrica es realmente cómoda para este tipo de aplicaciones basadas en el funcionamiento independiente de un robot o vehículo. El problema reside en que, en ciertas ocasiones, el vehículo consume más energía de la que la base de carga puede suministrar, incluso con una eficiencia del 75% en el caso del kit TDK. Asimismo, en las situaciones en las que la diferencia de consumo es positiva y se procede a recargar el vehículo, resulta en un intervalo de tiempo hasta dos veces superior al método por cable. No obstante, se han conseguido suficientes evidencias para incentivar el desarrollo e implementación de esta tecnología como contendiente real a los métodos tradicionales.

El movimiento del vehículo ha resultado problemático

Teniendo en cuenta todos los factores decisivos en el funcionamiento del proyecto, es evidente que el resultado final no sea el esperado en ciertas ocasiones. Se ha comprobado que todos los dispositivos y programas trabajasen correctamente, pudieran transportar y recibir información, y combinar las tareas para llevar a cabo la meta del proyecto. Sin embargo, al tratarse de diversas partes funcionando de forma independiente puede resultar crítico el criterio incorrecto de algún parámetro clave. En el caso de procesar los datos para realizar el trazado de ruta libre, ha sido complicado concentrar todos los factores comentados para conseguir un resultado correcto.

8.2 FUTUROS PROYECTOS

Tras comentar las diferentes conclusiones y consecución de objetivos, se proponen ciertas mejoras y consejos para continuar con el desarrollo de este proyecto con mayor profundidad, ya que se pueden aplicar diversas variaciones para un futuro proyecto que parta de la base de este trabajo.

Optimizar el uso de dispositivos y comunicaciones

En este proyecto eran necesarios distintos dispositivos que realizasen de forma síncrona tareas clave mientras mantienen una comunicación continua. Esto puede resultar abrumador para el uso cotidiano, por lo que se propone implementar este trabajo disminuyendo el número de dispositivos utilizados, sin comprometer la eficacia de la aplicación final, puesto que requiere una costosa cantidad de recursos. En este caso, podría condensarse todas las tareas de ambas Raspberry Pi en una sola, aunque esto no se ha hecho por no obligar a modificar en exceso la RPi principal que tiene suficiente carga aplicada. Lo mismo se propone para los ordenadores, que en este proyecto han sido tres independientes y es perfectamente posible ejecutar todos los programas en un único dispositivo. De esta manera se requerirán menos líneas de comunicación y se disminuirá la latencia en el transporte de la información.

CONCLUSIONES Y FUTUROS PROYECTOS

Estudiar e implementar el kit Renesas

Como se ha comentado en el apartado 5.3 el kit Renesas podría suponer una mejora para la base de carga seleccionada finalmente en este proyecto, por lo que se propone estudiar con mayor detalle la documentación sobre el conexionado de las bobinas e implementar el protocolo de comunicación de los circuitos integrados para enviar instrucciones directamente a las bobinas desde el ordenador o el vehículo.

Experimentar distintos sistemas de localización

Los resultados obtenidos en el apartado 7.2 muestran que el método de detección de damero funciona correctamente en distancias cortas, pero puede resultar ineficiente para aplicaciones que requieran un movimiento más libre o de mayor distancia. Es por esto por lo que se presenta la opción de utilizar otros sistemas de localización relativa, como por ejemplo los marcadores ArUco, que se plantean como una alternativa más polivalente al tratarse de diversos marcadores que el algoritmo reconoce para distinguir distintas posiciones dentro de un mismo entorno. Otra opción es implementar una cámara de mayor resolución que no se vea afectada por el desenfoque de movimiento. En última instancia se propone cambiar el tipo de sistema de localización, es decir, sustituir la cámara por un sistema basado en otra tecnología como, por ejemplo, un sensor LiDAR, que utiliza la emisión de láseres para determinar las distancias.

Realizar ensayos de movimiento libre

Ha resultado inviable dentro de las capacidades y límites del proyecto realizar un ensayo de movimiento totalmente libre que se ejecutase de forma exitosa. Por tanto, se alienta a continuar el trabajo del sistema de navegación para conseguir combinar el control de corrección de ángulo con la rutina de movimiento lineal del vehículo para acoplarlo a la base de carga. En un futuro se podrá implementar dicho sistema de navegación para poder realizar una ruta completa sin intervención del usuario y poder aplicarlo a otros vehículos para montar una flota de vehículos independientes.

Capítulo 9. BIBLIOGRAFÍA

Bertozzi, M., Broggi, A., Cellario, M., Fascioli, A., Lombardi, P., & Porta, M. (2002). Artificial vision in road vehicles. *Proceedings of the IEEE*, 90(7), 1258-1271.

Bradski, G., & Kaehler, A. (3 de noviembre de 2000). OpenCV. *Dr. Dobb's journal of software tools*, 120.

Diebel, J. (20 de octubre de 2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 1-35.

Geekworm. (8 de mayo de 2020). *geekworm.com*. Recuperado el 27 de junio de 2022, de *geekworm.com*: <https://geekworm.com/products/raspberry-pi-x728-max-5-1v-8a-18650-ups-power-management-board>

Ghimire, D. (5 de mayo de 2020). Comparative study on Python web frameworks: Flask and Django. Helsinki, Finlandia: Metropolia University of Applied Sciences. Obtenido de https://www.theseus.fi/bitstream/handle/10024/339796/Ghimire_Devndra.pdf?sequence=2&isAllowed=y

GlobalData. (28 de febrero de 2020). *Verdict*. Recuperado el 18 de julio de 2022, de *Verdict*: <https://www.verdict.co.uk/computer-vision-timeline/>

Jalled, F., & Voronkov, I. (2016). *Object detection using image processing*. arXiv:1611.07791.

Kukelova, Z., & Pajdla, T. (17-22 de junio de 2007). A minimal solution to the autocalibration of radial distortion. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, (págs. 1-7). Minneapolis. doi:10.1109/CVPR.2007.383063.

Li, J., & Zhang, D. (febrero de 2011). Camera calibration with a near-parallel imaging system based on geometric moments. *Optical Engineering - OPT ENG*, 50(2), 1-12. doi:10.1117/1.3533032.

Li, L., Wen, D., Zheng, N., & Shen, L. (7 de julio de 2011). Cognitive cars: A new frontier for ADAS research. *IEEE Transactions on Intelligent Transportation Systems*, 13(1), págs. 395-407.

Litman, T. (2017). *Autonomous vehicle implementation predictions*. Victoria, Columbia Británica, Canadá: Victoria Transport Policy Institute. Obtenido de <https://www.vtpi.org/avip.pdf>

MikroElektronika. (24 de abril de 2017). *Mikroe*. Recuperado el 29 de junio de 2022, de Mikro: <https://www.mikroe.com/current-click>

Naciones Unidas. (2013). *Sistemas mundiales de navegación por satélite*. Oficina de asuntos del espacio ultraterrestre, Nueva York. Recuperado el 14 de febrero de 2022, de https://www.unoosa.org/pdf/icg/2013/Ed_GNSS_S_ebook.pdf

OpenCV. (1 de agosto de 2013). *OpenCV.org*. Recuperado el 19 de abril de 2022, de docs.opencv.org: https://docs.opencv.org/3.4/d4/d94/tutorial_camera_calibration.html

OpenCV. (2013, August 1). *OpenCV.org*. Retrieved April 19, 2022, from docs.opencv.org: https://docs.opencv.org/3.4/d4/d94/tutorial_camera_calibration.html

OptiTrack. (19 de febrero de 2015). *OptiTrack*. Recuperado el 2022 de julio de 18, de OptiTrack: <https://optitrack.com/cameras/flex-13/>

OptiTrack. (2015, February 19). *OptiTrack*. Retrieved July 2022, 18, from OptiTrack: <https://optitrack.com/cameras/flex-13/>

Panchal, C., Stegen, S., & Lu, J. (2018). Review of static and dynamic wireless electric vehicle charging system. *Engineering science and technology, an international journal*, 21(5), 922-937.

Parziale, L., Liu, W., Matthews, C., Rosselot, N., Davis, C., Forrester, J., & Britt, D. (2006). *TCP/IP tutorial and technical overview*. Poughkeepsie, Nueva York: IBM Redbooks.

Recuperado el 2022 de julio de 12, de <https://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf>

Peng, D., Cao, L., & Xu, W. (1 de diciembre de 2011). Using JSON for Data Exchanging in Web Service Applications. *Journal of Computational Information Systems*, 7(16), 5883-5890.

Perzow, J. (16 de junio de 2016). Measuring Wireless Charging Efficiency in the Real World. Taipei, Taiwán: WPC Trade Conference. Recuperado el 2022 de febrero de 12, de <https://www.wirelesspowerconsortium.com/data/downloadables/1/6/2/9/john-perzow-wpc-efficiency.pdf>

Petersen, R. (2020). *Ubuntu 20.04 LTS Desktop: Applications and administration*. Alameda, CA, Estados Unidos: Surfing Turtle Press.

Raspberry Pi. (25 de abril de 2016). www.raspberrypi.com. Recuperado el 27 de junio de 2022, de www.raspberrypi.com: <https://www.raspberrypi.com/products/camera-module-v2/>

Raspberry Pi. (2016, April 25). www.raspberrypi.com. Retrieved June 27, 2022, from www.raspberrypi.com: <https://www.raspberrypi.com/products/camera-module-v2/>

Raspberry Pi. (14 de marzo de 2018). www.raspberrypi.com. Recuperado el 27 de junio de 2022, de www.raspberrypi.com: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>

Renesas Electronics Corporation. (1 de diciembre de 2020). *Renesas*. Recuperado el 30 de enero de 2022, de Renesas: <https://www.renesas.com/us/en/products/power-power-management/wireless-power/wireless-power-transmitters/wp15wbd-rk-15w-wireless-charging-solution-bi-directional-data-communication>

TDK Corporation. (31 de marzo de 2019). *product.tdk*. Recuperado el 30 de enero de 2022, de [product.tdk](http://product.tdk.com): https://product.tdk.com/info/en/catalog/datasheets/wlc_rx_wrm483265-10f5-12v-g_en.pdf

Thorpe, C., Hebert, M., Kanade, T., & Shafer, S. (mayo de 1988). Vision and navigation for Carnegie-Mellon Navlab. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3), 362-373.

Ylonen, T., & Lonvick, C. (enero de 2006). *The Secure Shell (SSH) Protocol Architecture*. Recuperado el 28 de junio de 2022, de www.rfc-editor.org: <https://www.rfc-editor.org/rfc/pdf/rfc4251.txt.pdf>

ANEXO A: INSTALACIÓN DE UBUNTU EN RASPBERRY PI

Este anexo incluye una descripción detallada sobre la instalación del sistema operativo Ubuntu Server versión 20.04 LTS en una Raspberry Pi. Para ello es necesario el uso de una tarjeta de memoria Micro SD compatible con la Raspberry Pi, puesto que el OS se instalará en dicha unidad de almacenamiento. Se utiliza el software *Raspberry Pi Imager*, diseñado específicamente para montar imágenes de OS en una Raspberry Pi.

Se deben seguir tres pasos para llevar a cabo la instalación:

1. Entrar a la primera sección denominada “Sistema operativo” y seleccionar el OS Ubuntu Server 20.04 LTS 64-bit como se muestra en la Figura A.1.

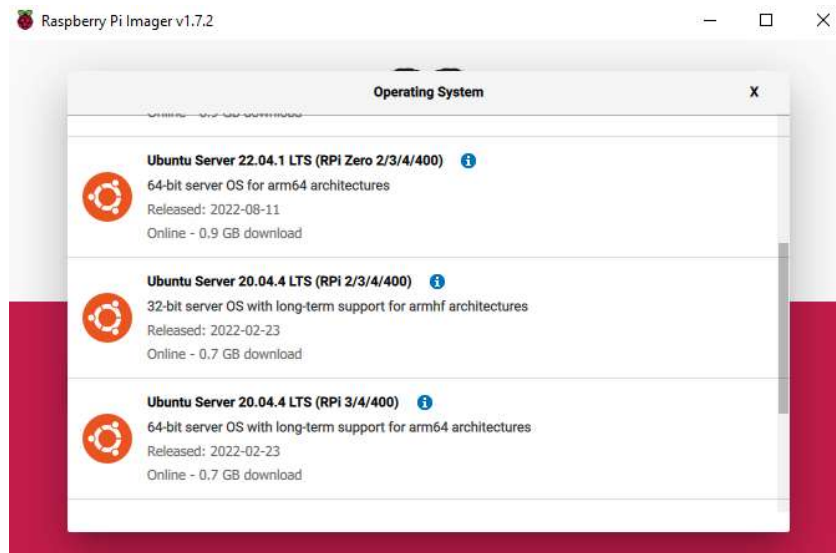


Figura A.1 Selección de OS

2. Elegir la tarjeta de almacenamiento introducida en el ordenador.
3. Rellenar los apartados como *Hostname*, usuario y contraseña, red WLAN y activar el protocolo de conexión SSH como se muestra en la Figura A.2

ANEXO A: INSTALACIÓN DE UBUNTU EN RASPBERRY PI

Image customization options to always use ▼

Set hostname: .local

Enable SSH

Use password authentication

Allow public-key authentication only

Set authorized_keys for 'pi':

Figura A.2 Configuración de instalación

4. Instalar el sistema operativo con las configuraciones del usuario y extraer la tarjeta MicroSD cuando finalice el proceso. Una vez extraída puede insertarse en una Raspberry Pi.

ANEXO B: WTM905525-12F2-12V-G Y WRM483265-10F5-12V-G DATASHEET

ANEXO B: WTM905525-12F2-12V-G Y WRM483265-10F5-12V-G DATASHEET

Nota: En este Anexo se incluye un extracto de las partes relevantes a este proyecto de los documentos originales. Si se desea consultar dichos documentos de forma íntegra, puede dirigirse a los siguientes enlaces.

Módulo de bobina transmisora

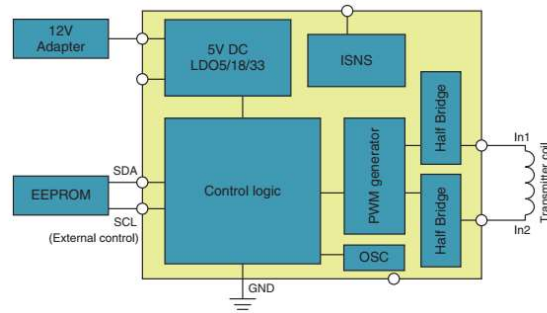
https://product.tdk.com/system/files/dam/doc/product/wireless-charge/wireless-charge/tx-coil-module/catalog/wlc_tx_wtm905525-12f2-12v-g_en.pdf

Wireless Power Transfer		TDK	
WTM905525-12F2-12V-G			
■ MAXIMUM RATINGS			
Operating Temperature / Humidity		Storage Temperature / Humidity	
-30 to +70°C / 10 to 95% R.H.		-40 to +85°C / 10 to 95% R.H.	
■ ELECTRICAL CHARACTERISTICS			
Tx Module			
WPC Compliant	Input Voltage (V)	Transfer Power (W)	Efficiency (%) typ.
Yes	12	15	75
Test Environment : Test Instrument : Chroma 8000 system or EQUIVALENT Ambient Temperature : 20 to 30 °C Relative Humidity : 25 to 75 %R.H. Rx module : WRM483265-10F5-12V-G Tx & Rx coil distance : 3.4mm			
■ IC INFORMATIONS			
IC	Manufacturer		
P9242-R	IDT (Integrated Device Technology)		
* Please contact us for more detail.			
■ PACKAGING : BOX TYPE			
Dimensions (mm)	Quantity (pcs)		
Tray	365x275x10	Qty per tray	10
Small carton	388x281.5x40	Qty per small carton	30
Outer carton	404x298x108	Qty per outer carton	60
■ OTHERS			
Weight (g) typ.			
50			

ANEXO B: WTM905525-12F2-12V-G Y WRM483265-10F5-12V-G DATASHEET

WTM905525-12F2-12V-G

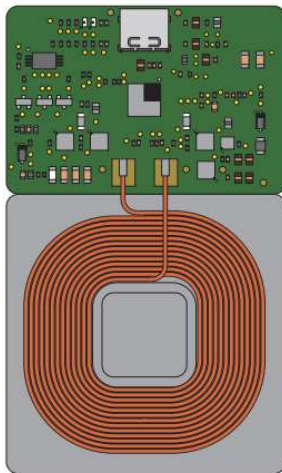
■ BLOCK DIAGRAMS



■ PIN LAYOUTS

Pin Configuration

(1) Type C connector input (CN1)



ANEXO B: WTM905525-12F2-12V-G Y WRM483265-10F5-12V-G DATASHEET

Módulo de bobina receptora

https://product.tdk.com/en/system/files?file=dam/doc/product/wireless-charge/wireless-charge/rx-coil-module/catalog/wlc_rx_wrm483265-10f5-12v-g_en.pdf

Wireless Power Transfer

WRM483265-10F5-12V-G

■ MAXIMUM RATINGS

Operating Temperature / Humidity	Storage Temperature / Humidity
-30 to +70°C / 10 to 95% R.H.	-40 to +85°C / 10 to 95% R.H.

■ ELECTRICAL CHARACTERISTICS

Rx Module			
Output Voltage (V)	Current Rating (A)	Output Power (W)	Efficiency (%) typ.
12	1.25	15	75

Test Environment :
 Test Instrument : Chroma 8000 system or EQUIVALENT
 Ambient Temperature : 20 to 30 °C
 Relative Humidity : 25 to 75 %R.H.
 Tx module : WTM905525-12F2-12V-G
 Tx & Rx coil distance : 3.4mm

■ IC INFORMATIONS

IC	Manufacturer
P9221-R	IDT (Integrated Device Technology)

* Please contact us for more detail.

■ PACKAGING : BOX TYPE

Dimensions (mm)		Quantity (pcs)	
Tray	365x275x10	Qty per tray	20
Small carton	388x281.5x40	Qty per small carton	100
Outer carton	404x298x108	Qty per outer carton	200

■ OTHERS

Weight (g) typ.
8.0

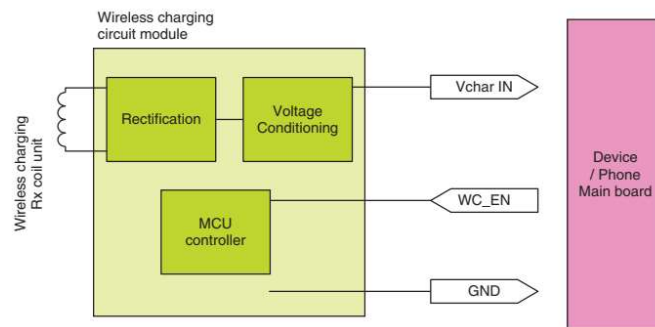
ANEXO B: WTM905525-12F2-12V-G Y WRM483265-10F5-12V-G DATASHEET

Wireless Power Transfer

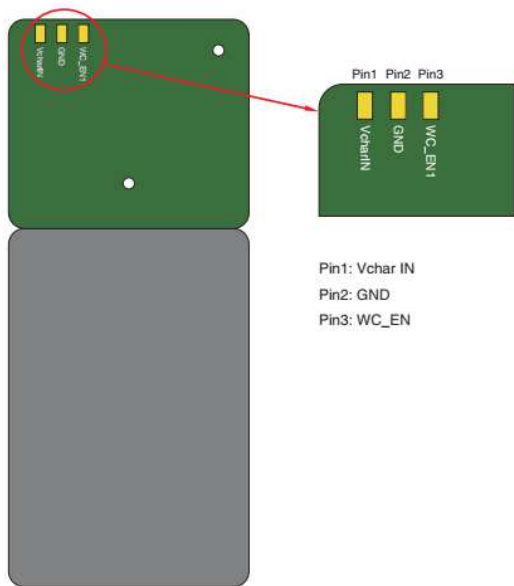


WRM483265-10F5-12V-G

■ BLOCK DIAGRAMS



■ PIN LAYOUTS



ANEXO C: CONTRIBUCIÓN A LOS OBJETIVOS PARA EL DESARROLLO SOSTENIBLE DE LA ONU

ANEXO C: CONTRIBUCIÓN A LOS OBJETIVOS PARA EL DESARROLLO SOSTENIBLE DE LA ONU

Este anexo comenta los Objetivos para el Desarrollo Sostenible (ODS) planteados por la Organización de las Naciones Unidas (ONU) que se busca apoyar con el proyecto. La Agenda 2030 de la ONU presenta 17 objetivos relacionados con aspectos económicos, sociales y medioambientales para garantizar un futuro para todos. Al desarrollar este proyecto se pretende el cumplimiento y compromiso con los objetivos 7 y 9 de los ODS, puesto que se relacionan directamente con el progreso y disponibilidad de la tecnología y la sostenibilidad.

Objetivo 7: Energía sostenible y no contaminante

Este proyecto apuesta por el uso de energía eléctrica de origen no contaminante para el consumo y alimentación de los dispositivos asequible para toda la población. El impulso de la sociedad digital debe ocurrir a través de energías completamente renovables y seguras para el medioambiente y el abandono de los combustibles fósiles.

Objetivo 9: Industria, innovación e infraestructura

Se propone un desarrollo tecnológico e industrial sostenible para garantizar igualdad y seguridad para todas las personas. Mediante los procesos industriales respetuosos con el medioambiente, las infraestructuras de calidad, la investigación e innovación se invita a las empresas a invertir en un futuro en el que la sociedad esté completamente desarrollada. En este caso, la industria robótica y la implementación de sistemas inteligentes en la rutina diaria pueden mejorar la calidad de vida de la población enormemente.